Autonomous Driving in Adverse Weather Conditions

Rosa Ku[1], Adithya Sairamachandran[2]

Department of Computer Science

Wayne State University

Detroit, MI, USA

[1]gu7925@wayne.edu

[2]go2977@wayne.edu

***Abstract*— This report explores simulating autonomous driving within an open source simulator called CARLA. Primarily, we are concerned with simulating driving successfully in different weather conditions such as rain or snow. We take a neural network approach to filter out noise caused by weather within the images taken from the camera sensor in CARLA.**

***Keywords*— Autonomous driving, Convolutional Neural Networks, ResNet, CARLA**

I. Problem Description

Autonomous vehicles are vehicles capable of sensing objects in its environment and are able to operate safely without human involvement. They handle a variety of driving tasks regardless of external conditions. One major factor to consider in order to successfully produce an autonomous vehicle is the effect of adverse weather conditions on the autonomous system. Adverse weather conditions dramatically affect driving conditions in that there is reduced visibility in the environment on the cameras and they prevent LiDAR sensors from producing high quality data as lasers bounce off rain or snow particles. It is crucial to resolve these issues in order to improve reliability of autonomous systems, and prevent improper decision-making that can cause a fatal mistake on the road.

In this paper, we will be focusing on the issue of low visibility on the camera specifically within CARLA, an open source simulator for autonomous vehicle research. CARLA is useful because it provides open digital assets such as building, vehicles, and urban layouts while also supporting numerous sensor suites, environmental conditions, full control of all static and dynamic actors and map generation.

II. Solution Description

Our solution to this problem is to develop and implement a filtering algorithm to remove added noise in a single image such as the rain droplets picked up by a camera. This algorithm is applied to a RGB camera within the CARLA simulator to test if an autonomous vehicle can drive in harsh weather conditions. Different weather and time combinations are tested such as the base case of a clear day, rainy afternoon, dusk, and heavy rain. Performance is measured in various weather and daytime conditions by analyzing how well the rain streaks are filtered from the image.

A. Filtering Algorithm

For our filtering algorithm, we utilized and expanded on the method that Fu et. al [1] used to remove rainy streaks from single images. They use a deep learning approach that consists of first decomposing the images to its detailed components (only rain streaks and object structures), and using a residual neural network (ResNet) structure [2] with negative residual mapping (neg-mapping). Neg-mapping involves subtracting the rainy image X from the clean image Y in order to compress the mapping range to reduce the solution space. This residual is useful because it isolates the rain streaks and can be used to estimate the final de-rained image. This neg-mapping is combined with the detailed layer as the input to the parameter layers of the ResNet.

B. Network Architecture

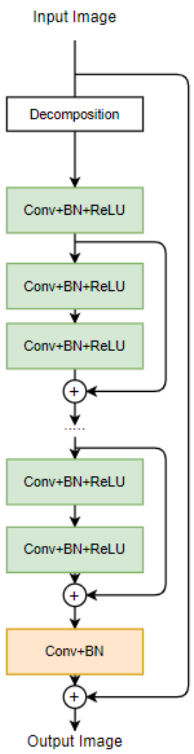


Figure 1: Neural Network Architecture

Figure 1 above shows the structure of the neural network used by Fu et al. and also applied to our project. The first layer decomposition of the rainy input image into its detailed components. The middle layers are the parameter layers which are made up of several residual blocks containing two layers each consisting of three components: a convolution, batch normalization, and feeding the output to a ReLU function. The last layer is the neg-mapping. The input image is added to the neg-mapping to produce the final filtered output image. Fu et al.'s model consisted of 26 layers total but we expanded on this to include 30 layers in order to see if it would improve performance.

C. Training

The training dataset comes from the rainy dataset provided in [3]. 100 rainy/clean image pairs were randomly selected for training. The parameter settings were as follows: number of feature maps was 16, number of image channels was 3, patch size of 64, kernel size of 3, while the batch size was 20. The total number of iterations was 110,000 and the learning rate was initially 0.1, then it was reduced to 0.01 after the 50,000 iterations, and finally 0.001 after 100,000 iterations.

D. CARLA Integration

CARLA integration begins with combining the example manual control [5] and automatic control [6] examples provided in the source code in order to give autonomous agents driving control while also allowing a user to change certain parameters (e.g. weather). Next, based on the weather settings defined in [7], we determined that the weather presets indexed from 3-7 and 11-14 were ideal for our testing scenarios. Then, with regards to creating the autonomous driving agent, sample code for a BasicAgent that follows a cyclic path [8] was integrated with the camera blueprint defined in [9] with the intent of allowing images picked up by the car to be processed using this model. Finally, to tie everything together, when the camera feed receives a new image, the_process_image function is called, where the raw bytes of the image are converted into a tensor before the model is called.

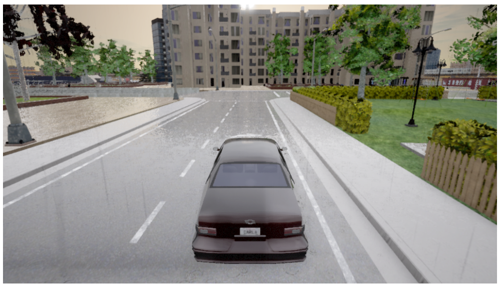
III. Results

Original capture on top, followed by the de-rained image.

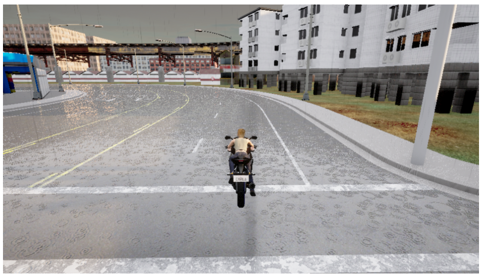
1. Clear Day

2. Dusk

3. Moderate Rain (Afternoon)



4. Heavy Rain



As shown by all these examples, the filtering algorithm doesn't change properties of the image in clear conditions, but does reduce the effect of rain droplets if they are present. Please view the project page [10] for enlarged images.

IV. Milestones and Outcomes

A. *Milestones*

1. Setting up AWS with CARLA
2. Creating the rain droplet filtering algorithm
3. Creating a driving agent that allows for autonomous driving with the CARLA Python API
4. Connecting the driving agent with the filtering algorithm

B. *Outcomes*

1. Creating a filtering algorithm to remove the effect of rain droplets when driving
2. Integrating this algorithm with a driving agent that can be used for autonomous driving simulations
3. Improving the performance of autonomous driving agents in adverse weather conditions

V. List of Hardware/Software Needs

- Python 3
- Tensorflow 2
- A CARLA instance running on AWS as shown in [4]
    - Note: CARLA version 0.9.10.1 works best
    - Note 2: CPU + GPUs could be scaled up as final framerate is ~2 FPS
- NVIDIA GPU
- This papers' codebase as listed in [10]

VI. Team Contributions

Rosa worked on the filtering algorithm and trained the model while Adithya worked on the CARLA integration.

References

[1] X. Fu, J Huang, D. Zeng, Y. Huang, X. Ding and J. Paisley, “Removing rain from single images via a deep detail network,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017), pp. 1715-1723, Honolulu, HI, USA, July 21-26, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.

[3] https://xueyangfu.github.io/projects/cvpr2017.html

[4] https://github.com/jbnunn/CARLADesktop

[5] https://github.com/carla-simulator/carla/blob/master/PythonAPI/examples/manual_control.py

[6] https://github.com/carla-simulator/carla/blob/master/PythonAPI/examples/automatic_control.py

[7] https://carla.readthedocs.io/en/stable/carla_settings/

[8] https://github.com/carla-simulator/carla/blob/master/PythonAPI/carla/agents/navigation/basic_agent.py

[9] https://github.com/carla-simulator/carla/blob/master/PythonAPI/examples/client_bounding_boxes.py

[10] https://github.com/AdiSai/Driving-in-the-Rain