On the Susceptibility of Deep Neural Networks to Natural Perturbations

Sabaina Haroon
University of Central Florida, Orlando FL
sabainaharoon@knights.ucf.edu

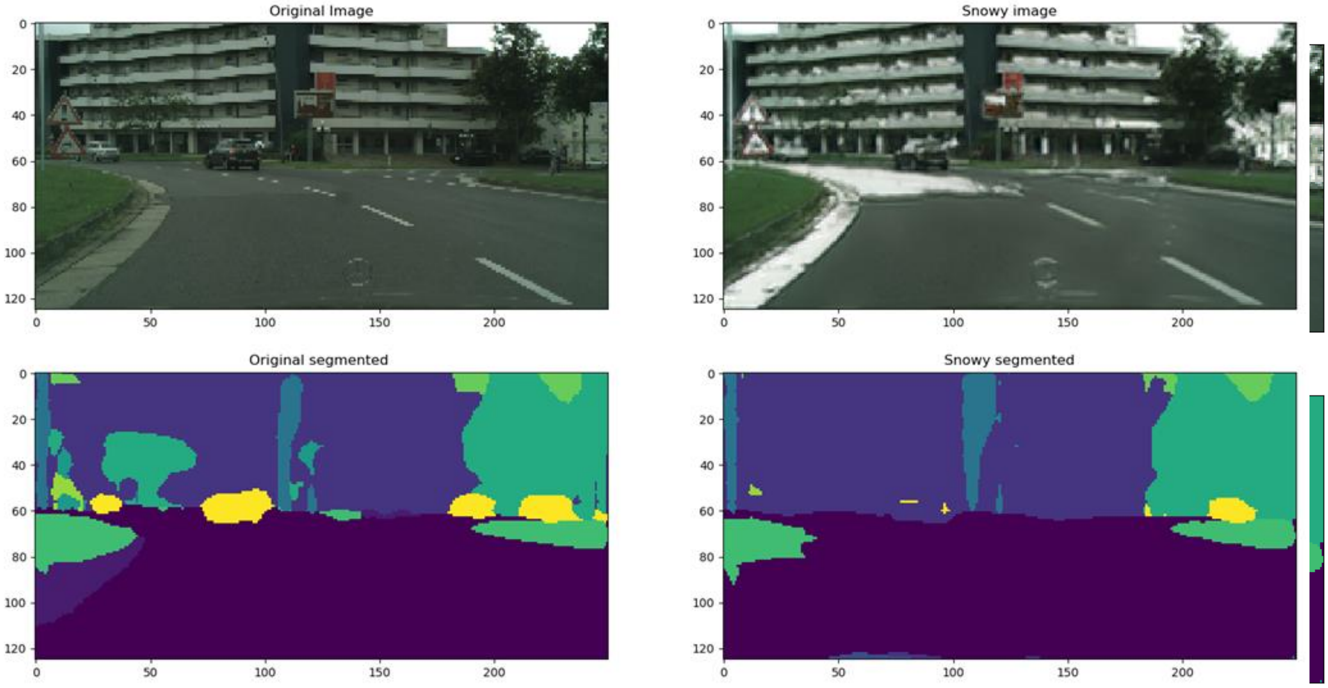

Figure 1: Snow is added to the original image, when tested car closer to snow is not detected

Abstract

Everything is moving towards automation and deep learning systems are playing a key role in this change. In one of the domains these systems are trained such that to classify and detect objects in real world scenario by an autonomous agent. Over the time, tremendous success has been seen in training such deep networks, but not much focus is given to the adversarial attacks these systems might face, and how they will respond to such adversaries. In this paper we are going to observe behavior of deep neural networks classified for autonomous vehicles using Cityscape dataset. Adversarial attacks are considered here in form of natural perturbations that can occur through rain, snow or hail.

1 Introduction

Deep neural networks are when exposed to different kind of adversaries, it poses multiple security concerns. There are different kind of adversarial attacks that a deep neural network can face, in one type of adversarial attack an image can be perturbed in such a way that human observer won't be able to detect any change, but classifier fails to classify the image correctly, whereas for this paper we are going to check the robustness of a given deep neural network against different natural perturbations that can occur in environment due to weather change. Since change in weather would mostly affect autonomous cars, for this project therefore I have picked street views dataset CityScape and have added synthetic perturbations to it that look closer to reality. I have added three different kind of noises against the original input image and after the addition of perturbations, state of the art trained model on this dataset is tested for the perturbed images. Since this classifier is strong, it still is able to detect different regions in the image correctly, but some of the regions are misclassified in an image, and for an autonomous car driving in the field without human intervention, misclassification of even a single object, with 80 percent other objects still classified correctly, would have dire consequences. A car won't slow down if it would not detect another vehicle in front of it and would cause an accident.

2 Related Work

Synthetic perturbations can be added to the dataset using multiple approaches, one of them is sign gradient method that adds perturbations to an image based on the optimization of numerical functions, these perturbations are calculated to fool a deep neural network for a given input.

Most relevant related work to this paper is the baseline research paper to which this work is an extension. Work done in baseline method adds fog to an input image using disparity maps created from left and right stereo images using *Cityscapes* dataset. The fog is added manually using computer graphics techniques to attack Inception deep learning model. OpenCV inbuilt function is used to calculate depth of each pixel in an image for disparity maps generation. These disparity maps are then smoothed for adding natural fog affect to an image and is controlled using two factors, thickness of fog and brightness of image. With more fog in the weather scene is blurred and its brightness decreases. Whereas to add more fog disparity maps are smoothed more. Fog is added with different thicknesses to generate a new image against an input image. Both the input image and newly generated image are tested against a trained classifier and change in their results are reported along with the Peak Signal to Noise Ratio between two images.

3 Our Approach

Perturbation to mimic effects of natural weather in an image are added using only classical computer vision techniques in previous works. However natural perturbations such as rain, hail or snow, generated using these classical methods look too digital and away from reality, to make these perturbation look natural and closer to reality I have used Generative adversarial network to transfer the content and style of one image to another. This automated method generates image that look closer to reality. There are still some limitations to our method such as selection of templates for perturbations and time needed to learn representations through generative adversarial networks, but these are worthwhile since they generate images very much closer to reality.

Generative adversarial networks (GAN) work with two deep networks learning against one another i.e. we have two modules in our network, generator and discriminator, generator generates a fake image and discriminator must detect if an image generated is fake or real. Models are trained side by side until discriminator becomes unable to distinguish

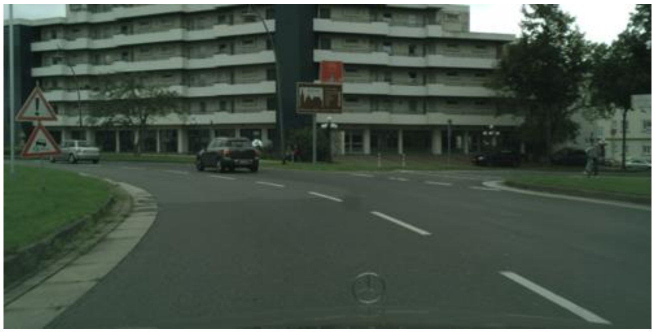

Figure 2: Original Image from anchen city

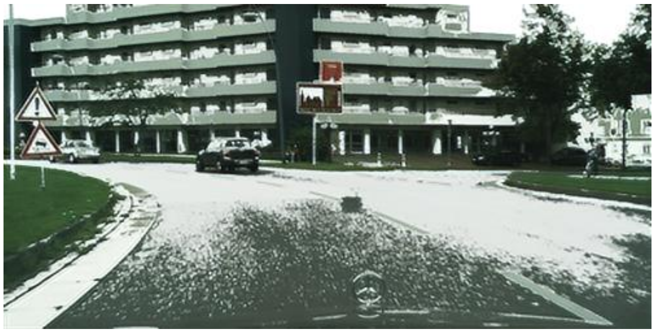

Figure 3: Snow added to Original Image using SinGAN

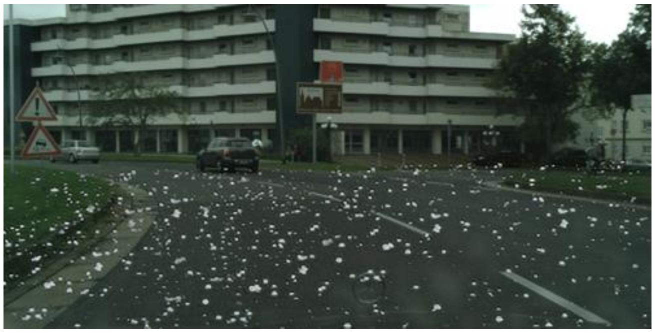

Figure 4: Hail added to the image using SinGAN

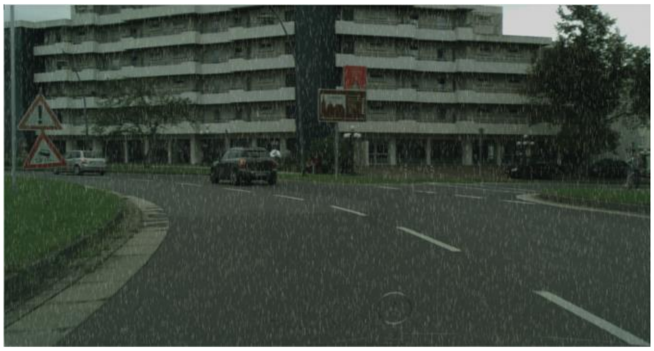

Figure 5: Rain generated using computer vision

fake images from real. I generate a naïve version of adversarial noise using a classical computer vision technique and blend it with input original image using GAN. This allows me to mimic natural affects closer to reality. SinGAN has been used to generate these adversarial images, It is a recent technique presented in ICCV 2019 and won best paper award. This paper uniquely outperformed all other GAN networks presented earlier in such a way that although it's a deep network, but it does not need any prior training data to train its network and can train GAN using only one input image. Which makes it perfect choice of generative adversarial network for our work.

3.1 Snow Perturbation

For neural style transfer we must provide a template of the image whose content we want to transfer, I generate this template for snow using classical computer vision approach. Snow mask is created for an image using HLS color space. Our algorithm detects RGB pixels having smaller values, which mostly are part of roads, trees etc. these values are converted into HLS color space and their lightness is adjusted to generate snowy affect. Brightness coefficient is used to control the brightness of image with snow and a threshold is used over the RGB pixel value, above threshold value lightness of pixels is adjusted for snow. A mask is generated as in figure 7. This mask is naively pasted on to original image by replacing values in original image with mask image values. After this SinGAN is trained for original image. After that harmonization is performed using naïve image and mask image using SinGAN which blends the affect with the environment of the input image to make it realistic.

Figure 7 and 8 show snow masks and naively pasted image with snow

3. 2 Hail Perturbation

Hail is generated using template of hail for a random image picked from results of google search. Mask for the hail is generated once and is added onto all the images. In case of snow we were generating snow image adaptive to each input image, whereas hail has the tradeoff with time that it uses a single template for all the images and does not require extra time to customize it for each image. Mask is naively pasted onto original images and is fed to SinGAN for harmonization. Figure 9 and 10 show template and mask selected from google search.

3.3 Rain Perturbation

Due to limited time constraint associated with this project, it was not possible to generate all the perturbations using SinGAN since it takes considerable amount of time around 25 minutes for training of one image. Rain is added using classical algorithm where random points are generated in an image, along these random points raindrops are added using line function. These raindrops can be slanted. Brightness is less on a rainy day so brightness of an image is also controlled. Raindrops are colorless therefore I have added opacity to raindrops.

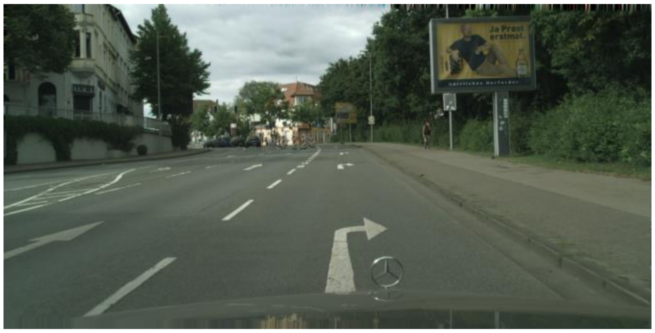

Figure 6: Original Image, bielefeld_000000_061975_leftImg8bit

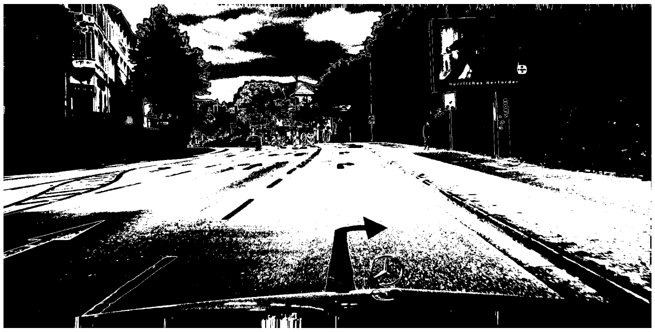

Figure 7: Snow Mask, threshold=210, brightness coefficient= 0.20

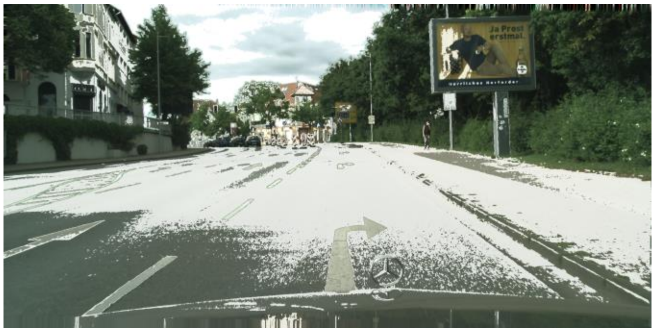

Figure 8: Naively pasted snow mask on original image

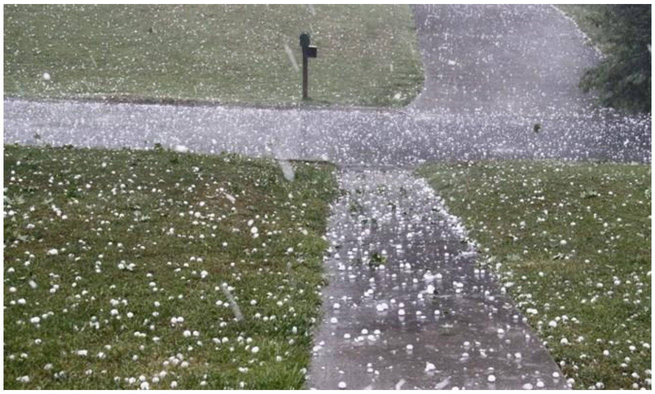

Figure 9: Hail Template image selected from internet

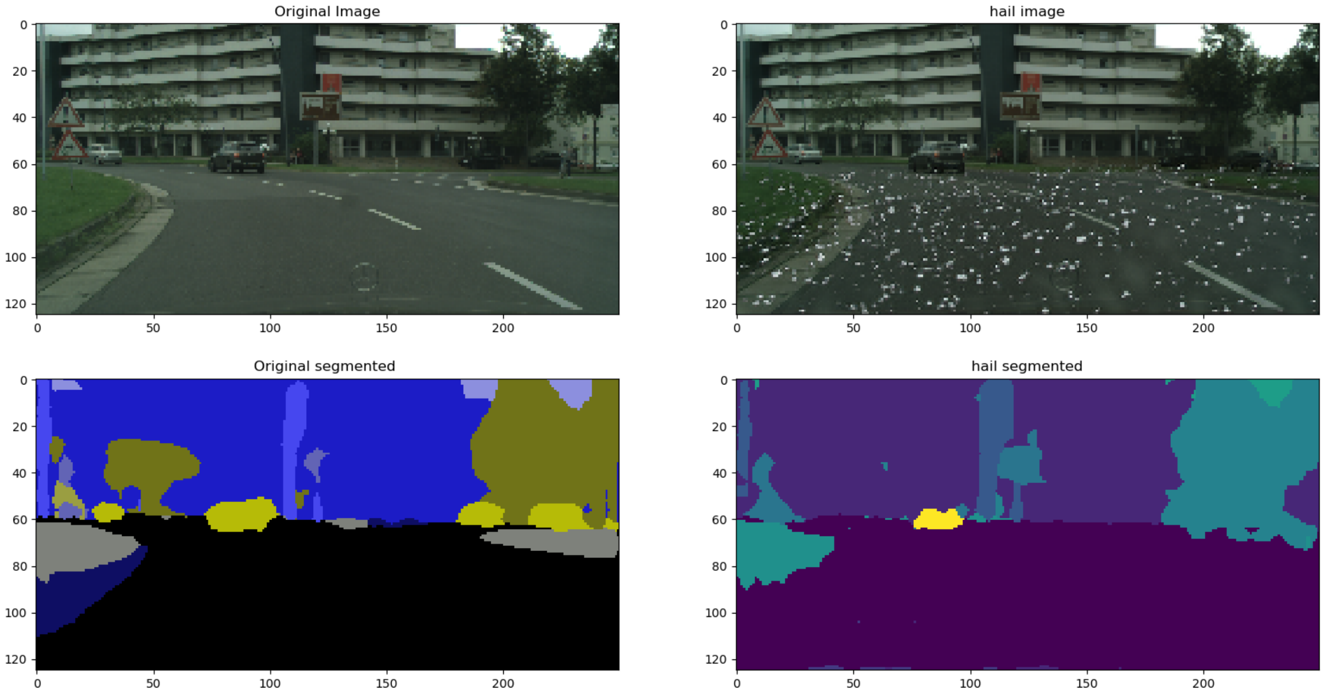

Figure 11: hail added on image image: aachen_000000_000019 , PSNR = 29.48137 db

3.4 SinGAN

In synthetic GAN model, unconditional generative model is used in pyramid structure to achieve different applications, such as paint to image, editing, harmonization, super-resolution and animation, by using single image instead of entire dataset as input. Their method, SinGAN, consists of pyramids of fully convolutional GAN networks that capture the patch distributions within an image at different scales (pyramid level). At each scale, GAN learns an image down sampled to that scale using noise and image generated from lower level GAN. GANs are trained from coarsest level to finest level of resolution, where at the starting coarsest level only noise is used as input, at every preceding level noise and image from previous level is used as input. GAN at each level is independently trained and fixed before going to next scale/level. The structure of GAN is same for all scales consisting of 5 fully convolutional layers. User studies were conducted to evaluated the performance of their method, paired and unpaired form tests were conducted, in paired tests two images were presented to the user and they had select one as real and one as fake, whereas in unpaired tests users were presented with images and were asked to classify them as real and fake. Their results achieved around 42 percent confusion level for unpaired tests, where 50 percent means perfect confusion between real and fake images which shows their results were promising. This method won best paper award at ICCV 2019.

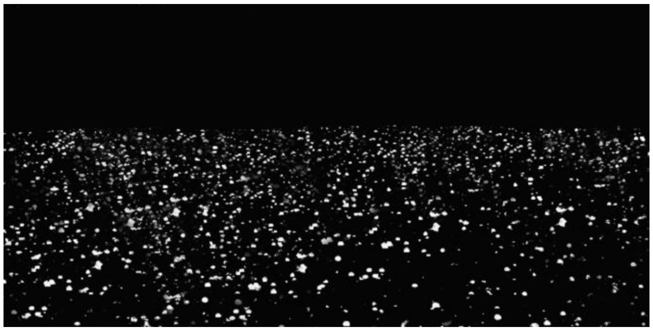

Figure 10: Hail Mask

3.5 PSNR

Peak signal to noise ratio is calculated between original image and perturbed image using the formula as follows

$$PSNR = 20\log_{10}\left(\frac{D}{RMSE}\right)$$

Where D is the maximum pixel intensity between two images and RMSE is the root mean square error between original image and perturbed image.

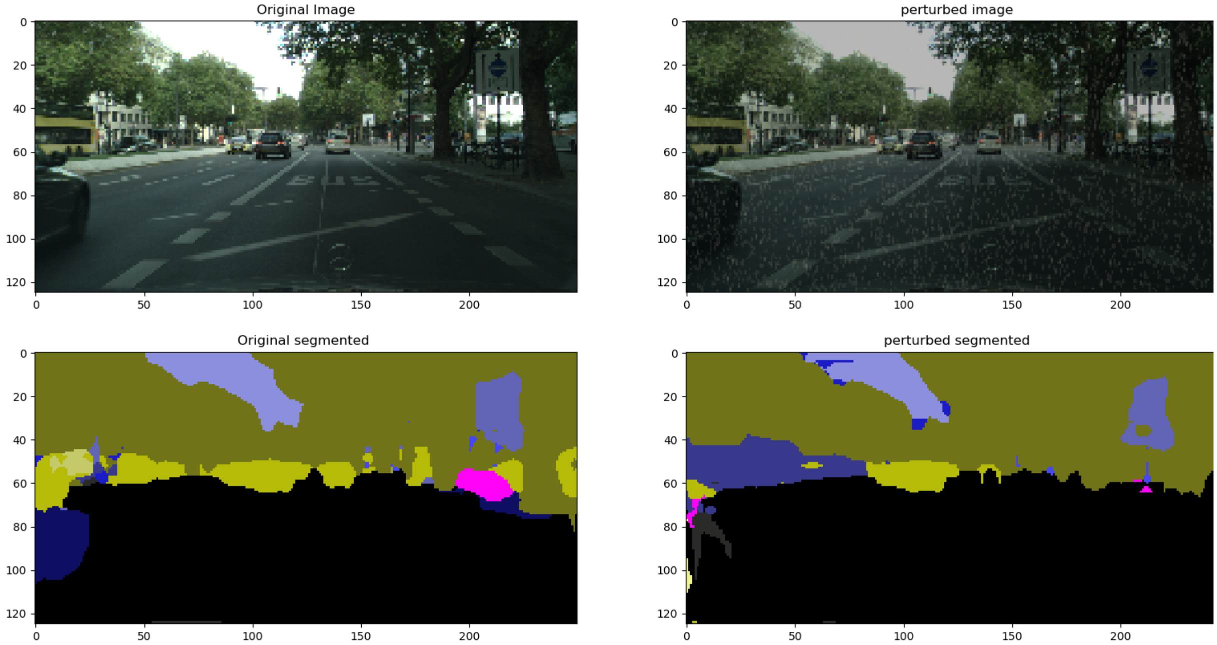

Figure 12: Rain impact observed on Image: berlin_000000_000019, PSNR: 28.507 db

4 Impact on Deep neural networks

For checking the susceptibility and robustness of a deep neural network towards natural perturbations I have selected model zoo’s inception pretrained model trained on cityscapes dataset. This model gave 80.31% when tested on validation set of cityscapes dataset. Since this was a semester project and I had limited time constrained for it, therefore I could not ran the algorithm for all the images in test set, but have tested it on all different streets on the testing dataset where the number is limited since testing it over full range would take time. Results of the behavior of the model with perturbed images is shown in fig 1, 11, 12

Insights on Figure 1. Results

Before snow	'road', 'sidewalk', 'building', 'pole', 'traffic sign', 'vegetation', 'terrain', 'sky', 'person', 'car'
After snow	'road', 'building', 'wall', 'pole', 'traffic light', 'traffic sign', 'vegetation', 'terrain', 'sky', 'person', 'car'

- Car that is closer to snow is not detected
- Sidewalk in leftmost corner of image is classified as road in perturbed image

Insights on Figure 11. Results

Before hail	'road', 'sidewalk', 'building', 'pole', 'traffic sign', 'vegetation', 'terrain', 'sky', 'person', 'car'
After hail	'road', 'building', 'wall', 'pole', 'traffic sign', 'vegetation', 'terrain', 'sky', 'bicycle'

- Car misclassified as bicycle
- Blue region at left most side of bottom left picture is sidewalk, after hailing it is being classified as road now
- No cars are detected after hail, this is dangerous as autonomous car would not stop moving at might cause an accident since it couldn’t see a car

Insights on Figure 12. Results

Before rain	'road', 'sidewalk', 'building', 'wall', 'fence', 'pole', 'traffic sign', 'vegetation', 'sky', 'person', 'car', 'truck', 'bicycle'
After rain	'road', 'building', 'wall', 'fence', 'pole', 'traffic sign', 'vegetation', 'sky', 'car', 'motorcycle', 'bicycle'

- Some of the cars on the left are misclassified as fence
- Not detecting most of the cars
- Sidewalk is misclassified as road after perturbation

Above tables show classification output before perturbations for original image and outputs for perturbed image. Though we might get same labels in the table but it does not necessarily means that these labels are correctly assigned to the region in segmentation mask for perturbed image.

5 Conclusion

I have tested susceptibility of a deep neural network against different kind of natural perturbations occurring in the field of operation of our autonomous agent using the deep neural network. Since this was a robustness check focused towards the testing of deep modules used in autonomous cars therefore it as only tested for different weather conditions. By testing the model against the best trained model zoo inception model, it was found that adding perturbations which were very close to reality the detector was still able to segment most of the parts in an image whereas it was misclassifying vehicle objects and sidewalks in some of the images, some of such instances have been reported in this paper. In future work rain can be generated using SinGAN also along with devising methods to reduce the time of generation of perturbations.

References

Shaham, Tamar Rott et al. "SinGAN: Learning a Generative Model From a Single Natural Image." *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019): 4569-4579.

Ozdag, Mesut et al. "On the Susceptibility of Deep Neural Networks to Natural Perturbations." *AISafety@IJCAI* (2019).

https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md

https://github.com/tamarott/SinGAN