Adversarial Attacks on Autonomous Driving Vehicles

Aman Aryan (20111009), Jay Kothdiya (20111027),
Moksha Vora (20111035)

IIT Kanpur

April 20, 2021

Objective

- Traffic sign recognition is an integral part of autonomous driving vehicles. Any misclassification of traffic signs can potentially lead to a multitude of disastrous consequences, ranging from a life-threatening accident to even a large-scale interruption of transportation services relying on autonomous vehicles. We, here, present one such attack on the autonomous driving vehicles.
- Our aim is to generate real-world like images by which the Autonomous driving vehicle can be deceived.

Figure 1: Adversarial effect on Autonomous driving vehicle

Components of the system

- Object Detector + Classifier (Car Simulator)
- Adversarial Network

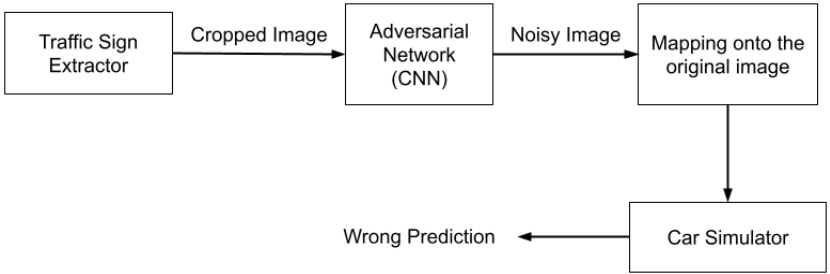

Figure 2: General Architecture of the system

Dataset: The dataset is manually generated using the Carla image dataset by extracting the traffic signs using the object detector module.

Car Simulator

For our objective to deceive the autonomous vehicle, we can consider an object detector and classifier as a car simulator.

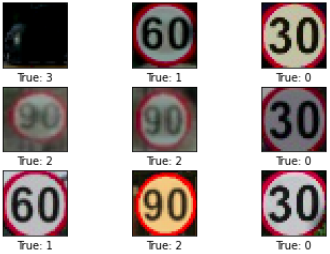

Figure 3: Sample of Training dataset

- **Object Detector:**

  The object detector is designed using OpenCV and pretrained Yolo v3 model which detects the traffic sign in the image.

- **Classifier:**

  The classifier is designed using Convolution Neural Network (CNN) based on traffic signs detected from Carla image dataset.

Object detection using Yolo v3

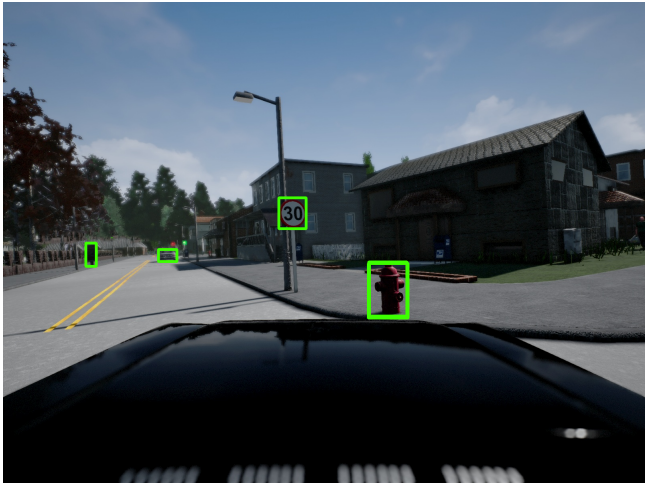

Figure 4: Object detection using Yolo v3

Adversarial Network Using CNN architecture

- The adversarial network is designed using CNN which takes cropped image from the car simulator and adds noise such that the classifier fails to predict the sign.
- For this, we use a targeted adversarial attack.
- We also generate a generalized loss for each class (different traffic sign class).

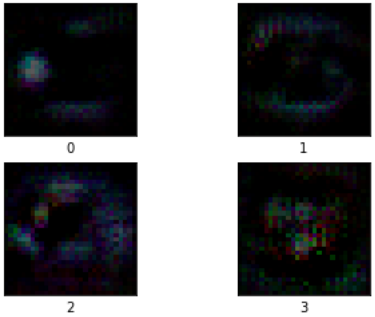

Figure 5: Noise learned by adversary for different classes

Adversarial Network Using GAN architecture

Training of Adversarial GAN Model:

- Train the discriminator to distinguish between fake and real traffic signs.
- Train the adversarial network (Extractor + Generator + Remap + Simulator) to generate the noisy traffic sign leading the misclassification by car simulator.
- We trained the GAN (Generator + Discriminator) to generate real-world like traffic sign using two optimization technique : ALT-OPT and Joint-optimisation.
- The architecture to train the adversarial GAN model is shown in Figure 6.

Testing Adversarial GAN on Car Simulator:

- The original image is passed to the traffic sign extractor.
- The extracted traffic sign is fed to generator to generate an adversarial traffic sign.
- The adversarial traffic sign is mapped onto the original image.
- The whole adversarial image is given to the car simulator which predicts the wrong traffic sign class.
- The architecture to test the adversarial GAN model is shown in Figure 7.

Training of Adversarial GAN

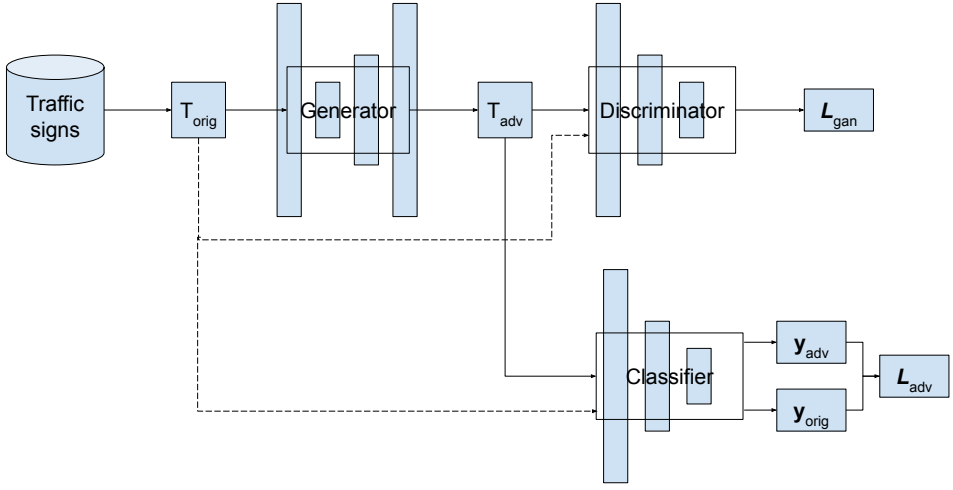

Figure 6: Architecture to train the Adversarial GAN

Objective (Loss) functions

- The L_{gan} is a binary cross-entropy loss function used for GAN (Generator + Discriminator).
- The L_{adv} is a custom loss function based on the categorical cross-entropy loss function used for adversarial network (Generator + Classifier).
- The objective is to minimize the GAN loss and maximize the adversarial loss. To maximize the adversarial loss, we are minimizing the negative of the categorical cross-entropy loss function.
- In **ALT-OPT**, we are optimizing the above two losses one-at-a-time.
- In **Joint-optimisation**, we are optimizing both the loss functions simultaneously.

Testing of Adversarial GAN

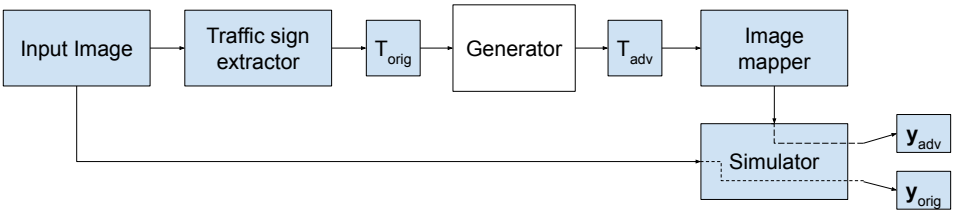

Figure 7: Architecture to test the Adversarial GAN

Results

- This step is used to map the adversarial traffic sign onto the real image. This is done by using the bounding box size obtained from the object detector.
- The Figure 8 and Figure 9 show the adversarial image generated by CNN and GAN models, respectively.
- In both figures, the image on the left is the original image and that on the right is the adversarial image. The simulator predicts class 0 (i.e. Speed Limit 30km/h) on the original image and predicts class 2 (i.e. Speed Limit 90km/h) on the adversarial image.

Adversarial Image by CNN

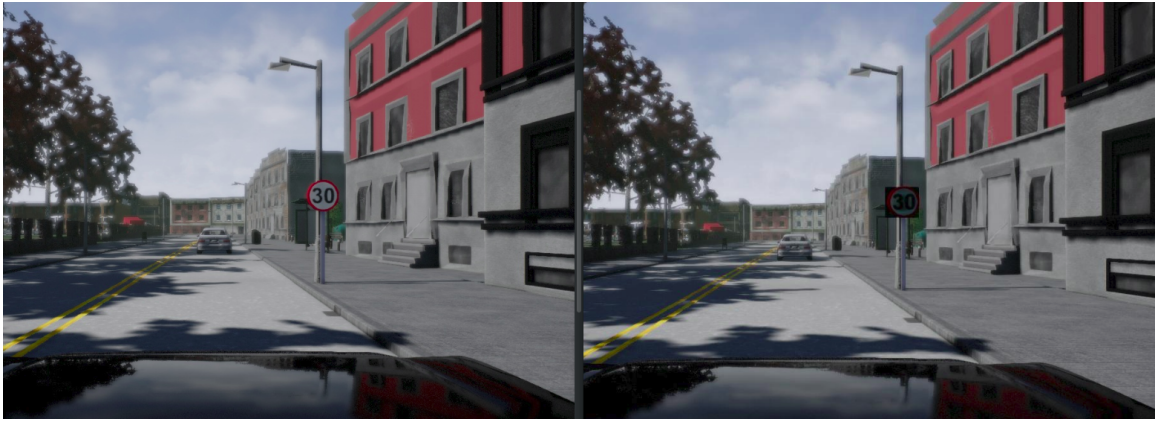

Figure 8: Comparison between original image and adversarial image by CNN

Adversarial Image by GAN

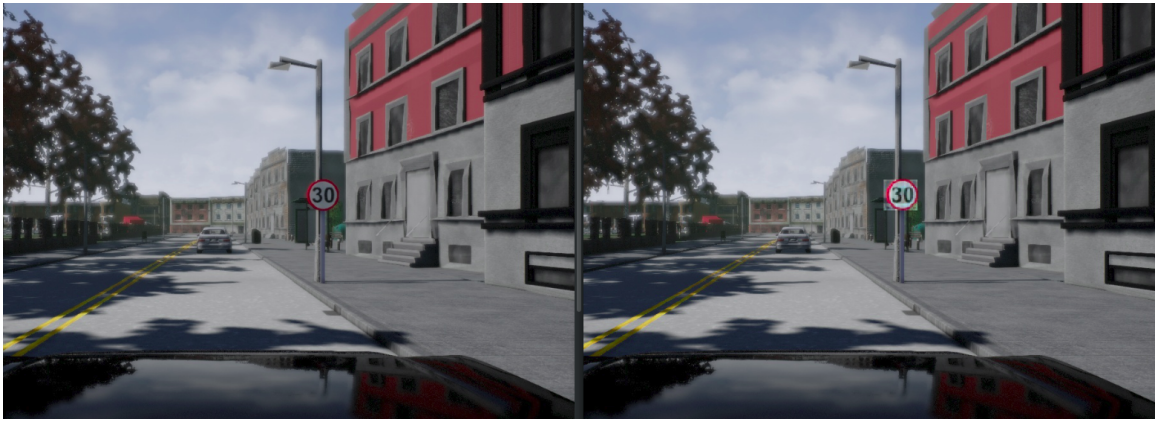

Figure 9: Comparison between original image and adversarial image by GAN

Observation

- Table 1 shows the results of train, validation, and test phase of adversarial GAN trained using 2 different techniques: ALT-OPT and Joint-optimisation.

	ALT-OPT	**Joint-optimisation**
Training Misclassification %	63.21	70.35
Validation Misclassification %	65.71	65.71
# Test images	1075	1075
# Traffic signs detected in original images	860	860
# Traffic signs detected in adversarial images	449	368
# Misclassified traffic signs	121	155

Table 1: Observations

- By comparing both Figure 8 and 9, we can see that GAN is able to deceive the simulator with lesser noise and the adversarial image generated by GAN is more similar to original image than adversarial image by CNN.

Future Works

- Currently, the shape of bounding box is identified to be square which leads to some boundary error. We are able to identify the circular boundary, but not able to reverse map it on the image. So, the possible future work can be to correctly map the circular sign on image.
- The pre-trained Yolo object detector is not very accurate. In future, we will try to implement our own object detector.
- In this project, image dataset is collected from Carla simulator. In future, we will try to train and test our model with the real world images.
- Another future direction can be to prevent the attack on autonomous driving vehicle using adversarial training.

Contribution and Work done after the presentation

Contribution:

Table 2 shows the tentative contribution of each member. However, each member helped each other in every task.

Name	Contribution
Aman	Worked on training the adversarial GAN model
Jay	Worked on data generation, object detection and classifier
Moksha	Worked on training the adversarial CNN model

Table 2: Contribution

Work done after the presentation:

- The adversarial model using CNN was completed before the presentation while the adversarial model using GAN is done after the presentation.
- The link to GitHub repo: `https://github.com/MokshaVora/Adversarial-Attacks-on-Autonomous-Driving-Vehicles`.

References

Darts: Deceiving autonomous cars with toxic signs

arXiv preprint arXiv:1802.06430, 2018

Physgan: Generating physical-world-resilient adversarial examples for autonomous driving

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14254–14263, 2020

Physically realizable adversarial examples for lidar object detection

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13716–13725, 2020