Fahad Ahmad

Personal Information

Status: Undergraduate Student

Program: Electrical and Electronics Engineering

School: New York University Abu Dhabi

Website: https://www.linkedin.com/in/fahad-ahmadb/

RA Period: From 2017-05 to 2017-10

Biography

I'm an associate consultant at Management Solutions. Before that, I was a research assistant in NYU Multimedia and Visual Computing Lab, advised by Professor Yi Fang. I am broadly interested in 3D Computer Vision, Pattern Recognition and Deep Learning.

Capstone Project: Deep Learning Approaches for Self-Driving Cars in an Adversarial Environment

1 Description

Our capstone project aims to improve the performance of self-driving cars in adversarial conditions. As such there are two main parts of the project – (1) Computer Vision (object detection and localization) and (2) Security (attacking the object detectors). In light of the criteria above, our team worked on two object detection algorithms, namely YOLO (2D) and VoxelNet (3D). Firstly, YOLO, a state- of-the-art 2D object detection algorithm was implemented from scratch. The YOLO algorithm was successfully implemented by using a 24-layer convolutional network. Training of the algorithm took about a week (135 epochs) with the PASCAL VOC dataset and resulted in an eventual loss of 3.427, indicating the relatively high accuracy of the object detector. After having successfully completed implementation of the 2D YOLO object detection algorithm, the next goal for the computer vision side of the project was to implement a 3D object detection algorithm. The algorithm chosen for this task was VoxelNet, which is a state-of-the-art 3D object detection algorithm recently published by Apple. The implementation was successfully completed. The security aspect of the project was started by executing one-pixel (black-box) attacks. Two categories of one-pixel attacks were implemented – (1) targeted attacks and (2) untargeted attacks. After successfully implementing the two types of one-pixel attacks on standard CNNs like ResNet and DenseNet, our team integrated the security attacks with object detection and implemented targeted attacks on the YOLO object detector. However, since YOLO is a state-of-the-art object detection algorithm, fooling the YOLO detector required perturbing around 10 – 15 pixels in contrast to the one-pixel attacks on ResNet and DenseNet. In the next part, we decided to test our security attacks using a well-known car simulator known as Udacity. Firstly, the Udacity car simulator was tested with modified input images (flipped, blurred, varied brightness). The results were successful, and we were able to affect the autonomous mode of the car and thus the car went off the track. To take a further step, we started the implementation of GANs focusing on DCGANs. We aimed to generate fake images and test them on Udacity simulator

and check if we were able to fool the simulator. However, due to time constraints, we were not able to design stable GANs which are difficult to train due to problems such as non-convergence, mode collapse and diminished gradients. After having a solid background in different detection and security methods, we finally moved to a specific problem in self-driving car industry: road lane line detection. Lane detection is a critical part of any autonomous driving system. Although lane detection appears to be a relatively uncomplicated task compared to most other computer vision applications, there are several factors that challenge lane detection in real world scenarios. In particular, the lack of any distinctive features makes lane detection difficult, readily resulting in the network being confused by other objects with similar appearance. Despite current lane detection algorithms performing well on standard lane detection tasks, curved lane detection has not yet been well addressed in the literature. In this project, we propose a novel multi-stage network CurveNet specifically for curved lane detection. There are 3 main components of CurveNet – Primary Convolutional Neural Network (P-CNN), Far-Near Convolutional Neural Network (FN-CNN) and Bend Network (BendNet). The first stage, P-CNN, encodes foundational visual features from the input image into a latent feature map, which is then fed separately to both FN-CNN and BendNet. FN-CNN then extracts horizontal and vertical feature maps, whereas BendNet generates horizontal and vertical orientation weights to weight the corresponding feature maps, allowing the network to learn the curvature of the lanes. Finally, the combined feature map is used to produce a probability map which in turn yields the desired instance segmentation results. Experiments on the CULane dataset show that CurveNet delivers outstanding performance on curved lane detection and is comparable to other existing methods on regular lane detection.

2 Method

In this section, we introduce the final design of our project. Initially, we started with the implementation of different object detection algorithms followed by different security attacks on these algorithms. Based on this, we moved to our final goal of working on the security of autonomous vehicles. We worked on the problem of lane detection in self-driving cars and proposed a novel solution to lane detection in curved roads. In the initial stages of lane detection, our deep learning model was quite different from the final one. Numerous ideas and architectures were tried and tested before choosing the final model shown

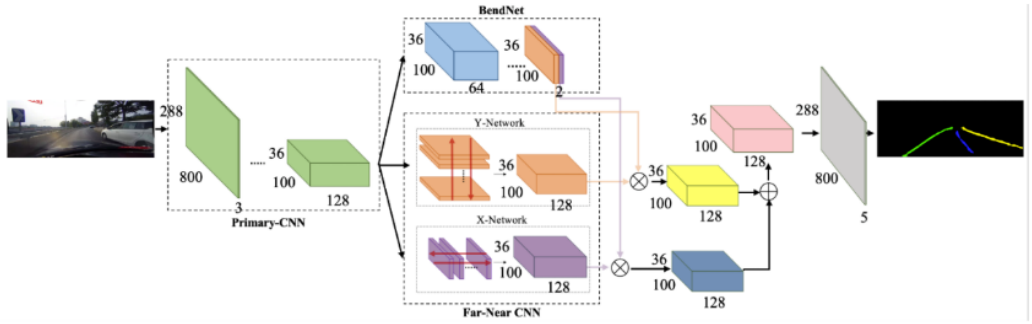

Figure 1: The final pipeline of the deep learning model for our lane detection method.

in Figure.1. After trying numerous architectures and understanding why these architectures did not yield the required results, we found the optimal architecture which is reproduced below for ease. For each of the iteration of architecture, we tuned the given network with optimal hyper parameters. Note that it was a time consuming process since after changing a few hyperparameters, we needed to train the network for several days. Thus, testing each of the networks took approximately about a week for testing.

3 Results

In this section, we conduct experiments to demonstrate the effectiveness of the proposed approach. We test our CurveNet on the CULane dataset, which is a large scale challenging dataset containing 88880 images for training set, 9675 images for validation set, and 34680 images for test set. CULane dataset consists of approximately 55 hours of videos and has 133235 frames. Each image in the dataset has a resolution of 1640x590. Furthermore, the test set is further divided into several challenging categories: curved, crowded, night, no line,

Approach	Normal	Curved
SCNN	90.6	64.4
ReNet	83.3	59.9
DenseCRF	81.3	57.8
ResNet-50	87.4	59.8
CurveNet	**84.3**	**61.1**

Table 1: Comparison of experimental results of CurveNet with other state-of-the-art methods on normal and curved categories of CULane test set. F1 scores (with an IoU threshold of 0.5) are reported.

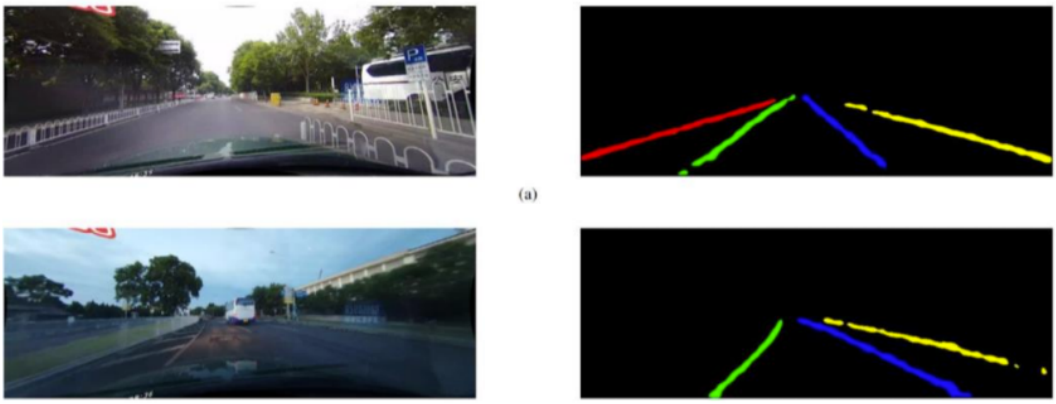

Figure 2: Shows qualitative results for adversarial attack on CurveNet.

shadow, arrow, dazzle light and crossroad. For the ground truth, all the images are annotated with cubic splines. Note that in CULane dataset even if the lane lines are not visible in the images due to occlusion, abrasion or low illumination, the lanes are still annotated. We provide our results for both the 'normal' and the challenging 'curved' lanes in Table 1 below. For evaluation, we follow the widely used evaluation metrics of True Positive Rate (TPR) and False Positive Rate (FPR). As shown in Figure.2, the attack was also implemented to attack CurveNet. The attack algorithm is still in the early stage and able to reduce the performance of CurveNet, however, the result is still conclusive whether to conclude it is a successful attack. Hence, the lane detector, CurveNet, was found to be robust under adversarial perturbation attack compare to normal object detection.