Improving Robustness in Deep Fusion Modeling via Adversarial Training

Amy Nguyen
Halıcıoğlu Data Science Institute
University of California, San Diego La Jolla, CA, 92093
atn001@ucsd.edu

Ayush More
Halıcıoğlu Data Science Institute
University of California, San Diego La Jolla, CA, 92093
amore@ucsd.edu

Abstract

Autonomous vehicles rely heavily on deep fusion modeling, which utilize multiple inputs for its inferences and decision making. By using the data from these inputs, the deep fusion model benefits from shared information, which is primarily associated with robustness as these input sources can face different levels of corruption. Thus, it is highly important that the deep fusion models used in autonomous vehicles are robust to corruption, especially to input sources that are weighted more heavily in different conditions. We explore a different approach in training the robustness for a deep fusion model through adversarial training. We train the model on adversarial examples and evaluate its robustness against single source noise and other forms of corruption. Our experimental results show that adversarial training was effective in improving the robustness of a deep fusion model object detector against adversarial noise and Gaussian noise while maintaining performance on clean data. We believe that this is relevant given the risks that autonomous vehicles pose to pedestrians - it is important that we ensure the inferences and decisions made by the model are robust against corruption, especially if it is intentional from outside threats.

1 Introduction

Deep fusion modeling has been used in many applications, specifically in autonomous vehicles. The key advantage in this approach is utilizing multiple input sources, in which they provide shared and complementary information. For instance, in different environmental conditions such as nighttime or rain, some sensors would be weighted more heavily than others and can complement the shortcomings of other input sources. This can be seen in a variety of input sources for autonomous vehicles such as LIDAR (Light Detection and Ranging) radars and RGB cameras, which serve different information about the environment such as distance and detection of other objects. Specifically, LIDAR sensors are more effective at nighttime in comparison to RGB cameras. Thus, it is important to ensure that the model can still make robust predictions when facing single source corruption, especially in the sensors that are weighted more heavily.

Single source corruption could be the result of physical damage done to a particular sensor instead of the overall inputs themselves, which emphasizes the importance of guaranteeing some robustness from the shared information of the sensors to compensate for the corruption. If this is not accounted for, there would be serious consequences for allowing a robust-poor autonomous vehicle to drive on the streets with actual civilians. Kim, Taewan, and Joydeep Ghosh's [1] work addresses this issue through implementing two efficient training algorithms for minimizing their novel loss to ensure robustness without affecting the performance of the model on clean data.

While these are effective approaches to improving the robustness of the model, the motivation behind these solutions was to handle random single source corruption, as the authors generated random noise through sampling from a Gaussian distribution as well as downsampling. However, these models are susceptible to intentional corruption, either through a third party source or a malfunction within the system, in which the objects are classified as something else. This poses a particular threat to safety-critical applications of ML, notably self-driving cars, as the noise can be intentionally optimized on the inputted data to control the decisions made by the model. In order to explore this motivation in protecting the deep fusion model systems, we examine adversarial training as a method to train for robustness against single source corruption.

We train three 3D object detection models, which are trained on clean data, fine-tuned on adversarial data, and fine-tuned on random noise data. Then, we compare the results of these models on clean, adversarial, and random noise data and evaluate their performance on robustness.

Figure 1. Sample Image from KITTI Dataset

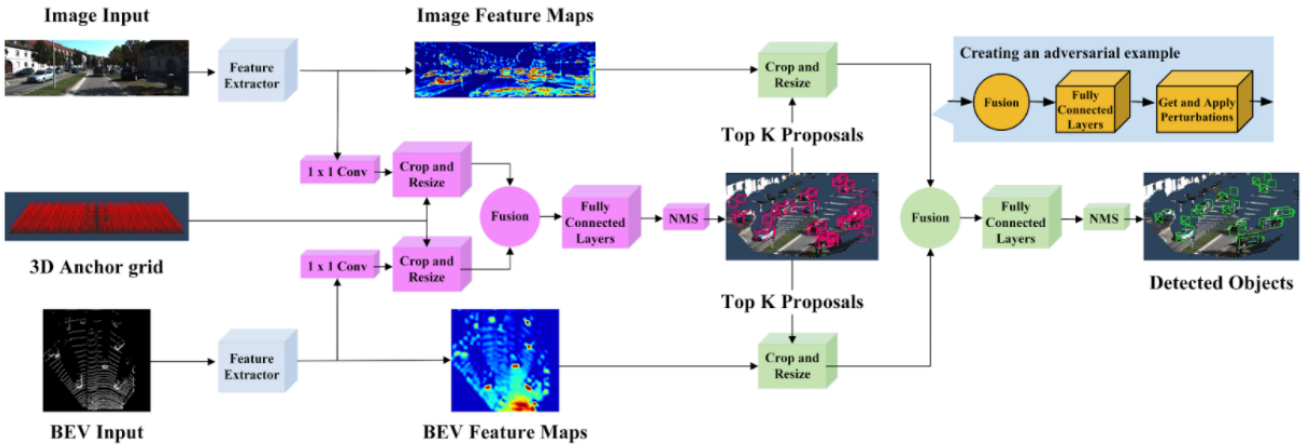

Figure 2. AVOD Model Architecture [6]

2 Data

For our research, we will use the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) dataset, a popular benchmark dataset for autonomous driving research. This contains six hours of traffic scenarios, which were recorded using various modalities such as color stereo cameras and a Velodyne 3D laser scanner. The scenarios recorded range between different locations such as rural streets, freeways, and city roads [5]. For our purposes of the experiment, we utilize the benchmarks for object detection tasks, which provides accurate bounding boxes in both 3D and BEV (Bird's Eye View) for object types such as cars, cyclists, and pedestrians. Figure 1 showcases an example of an RGB image input from this dataset.

3 Model

We use the AVOD (Aggregate View Object Detection) model, which is a neural network that uses LIDAR point clouds and RGB images to deliver real-time object detection in the form of bounding boxes and labels for objects in an image [6]. It is structured by two subnetworks, a region proposal network (RPN) and a second stage detector network, the former generating 3D object proposals for multiple object classes and the latter creating accurate oriented 3D bounding boxes and category classifications for predictions.

The AVOD model has state of the art results on the KITTI object detection benchmark, making it a great candidate for our baseline model. Using the same setup as Kim, Taewan, and Joydeep Ghosh [1], we will train the model solely on the car class for the object detection tasks and use the feature pyramid network for feature extraction. Figure 2 highlights the structure of the AVOD model, in which the blue components represent the feature extractors, pink components represent the region proposal network (RPN), green components represent the second stage detector network, and the yellow components representing the adversarial examples generation process.

4 Methods

As our motivation for pursuing this research is to handle intentional single source corruption, we explore adversarial training as an approach to developing robustness. Adversarial training focuses on deceiving the model into mislabeling an image by altering the pixel values so that the changes made to the image are indistinguishable to the human eye, but recognizable by a model. These mislabeled images through small perturbations are called adversarial examples. Figure 3 showcases an example of an image of a pig perturbed to be misidentified as an airliner, despite visually appearing the same.

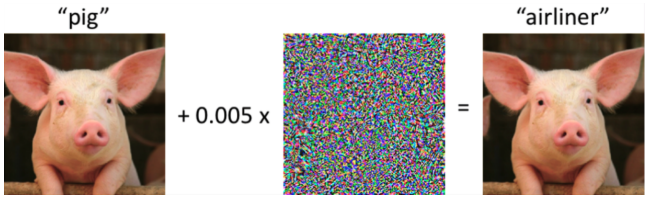

Figure 3. Adversarial Example of a Pig Image [4]

4.1 Adversarial Examples

Since an adversarial example should predict the wrong label, creating an adversarial example differs from the typical way of training a classifier. The usual way would be to minimize the loss of the input's predicted output to the true output as represented by

$$\underset{\theta}{minimize}\ \ell(h_\theta(x), y)$$

where ℓ is the loss function, h_θ is the model, and x is the image input. Given the structure of the AVOD model, we add the perturbation to the image ROIs (Region of Interest), which are represented as a feature map instead of directly manipulating the specific pixel values. This is generated from the region proposal network (RPN) component of the model. For the model to make its predictions, it converts the different input sources information into feature maps, which can be interpreted and passed through the convolutional layers of the model.

However, to create an adversarial example, we instead want to maximize the loss. The optimization problem to solve would be

$$\underset{\hat{x}}{maximize}\, \ell(h_\theta(\hat{x}), y)$$

where $\hat{x}$ is the adversarial example we want to maximize the loss of.

The adversarial example is an image x with noise δ added to it. This noise, more formally referred to as perturbation, is a mask of values over each pixel value and is specific to a given image. Rewriting the optimization problem again to include δ, we get

$$\underset{\delta \in \Delta}{maximize}\, \ell(h_\theta(x + \delta), y)$$

where δ indicates the valid set of perturbations to add. We keep δ within Δ to ensure the perturbed image is still recognizable to the human-eye. Solving this equation yields an adversarial example to use for an untargeted attack.

We will implement the Fast Gradient Sign Method (FGSM) as our primary method of solving for the optimization objective. For FGSM, we take the largest step possible to maximize the loss so that δ is updated to be as large as possible. To ensure is still a valid perturbation, δ is constrained to be within $\pm\epsilon$. For our approach, we select δ to be a fixed value as a maximum perturbation. This means the magnitude of δ is maxed to be ϵ in FGSM. We then With this constraint, the equation for updating δ in FGSM becomes the following.

$$\delta = \epsilon \cdot sign(g)$$

This process creates an adversarial example on one image. For computation speed, we propose using FGSM to produce an adversarial example. To train a model adversarially, multiple examples need to be created and the overall loss on the prediction of all these images need to be minimized.

4.2 Adversarial Training

To train a model against adversarial attacks, we create adversarial examples and include them into the training set. The loss for predicting all the adversarial examples would need to be minimized. Formally, this optimization problem can be written as

$$\underset{\theta}{minimize}\, \frac{1}{|S|} \sum_{x,y \in S} \underset{\hat{x}}{maximize}\, \ell(h_\theta(\hat{x}), y)$$

where S represents the input and output pairs and the inner maximization is the same as the previous section.

This optimization problem, also known as the outer minimization problem, can be solved using standard gradient descent. For our experiment, the model is fine-tuned using adversarial examples, as it is initially constructed using clean data. This allows us to focus on a standard baseline model and compare the effects of introducing adversarial examples as a means of fine-tuning the initial parameters.

4.3 Proposed Training Algorithm

We will implement our adversarial training algorithm through developing adversarial examples from the input sources and optimizing the maximum perturbation added to the data. Specifically, we plan on perturbing the image input of our model due to difficulties in translating the noise over to the LIDAR input [2]. Although we recognize that we can add perturbation to both input sources, we understand that there would be a different range of perturbations added. For instance, the noise added to the LIDAR input would be of a different magnitude and for our exploration, we plan on only focusing on images.

Initially, we train the model normally given clean data, but fine-tune the parameters based on the adversarial examples we provide later. We calculate the best perturbation to add by using FGSM to the image input for each image ROI (Region of Interest). We repeat this procedure after initially training on the clean data so that we can fine-tune the parameters in the training procedure with adversarial examples. We illustrate our procedure below:

Proposed Algorithm

1. Initially train on clean data for first 80% of iterations
2. For last 20% of iterations
   - A. Select (x_i, y_i) in mini-batch B
     - a. Find best attack perturbation $\delta^* = argmax_\delta \ell(f(x_i + \delta), y)$
     - b. Compute and add the gradient at δ^* g := g - $\alpha \nabla \ell(f(x + \delta^*), y)$
     - c. Perform backpropagation to fine-tune model parameters

Figure 4. Pseudocode for Training Algorithm

5 Experimental Results

We test our training algorithm for the 3D and BEV object detection tasks on the car class of the KITTI dataset and compare our results to the previous work done by Taewan Kim and Joydeep Ghosh [1]. These results are based on the difficulty levels within the dataset, ranging between easy, medium, and hard. We follow the standard metric of using an Average Precision (AP) score and reporting the minimum AP score across all input sources to assess robustness.

We compare three different algorithms and assess their performance based on the data provided: the AVOD model trained on (i) clean data, (ii) single source randomly generated noisy data, and (iii) adversarial examples. For our training purposes, we opted to use the metrics recorded by Taewan Kim and Joydeep Ghosh [1] for the following: AVOD model trained on (i) clean data and (ii) single source random noise and the inference on both of these data. Hence, we focus on the following experimental set-up to generate our results:

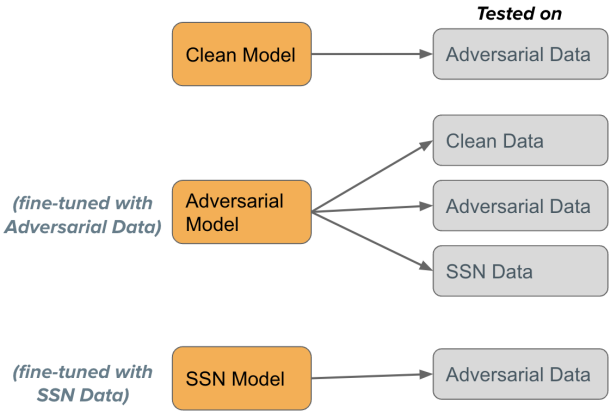

Figure 5. Experimental Set-Up

5.1 Results

From our results, we observed that on the clean test data, the adversarial model performed slightly worse than the clean and SSN models but managed to perform slightly better than the SSN model on SSN data (Figure 6). We find this interesting as the SSN model was trained specifically to handle random single source generated noise, but our adversarial model proved to be as robust in handling random noise. Although our adversarial model performed slightly worse on the clean dataset compared to the other two models, its performance is comparable in that there is not a significant drop as seen with the adversarial test data for the clean and SSN model.

We also observed that the adversarial model performed significantly better than the other two models on adversarial

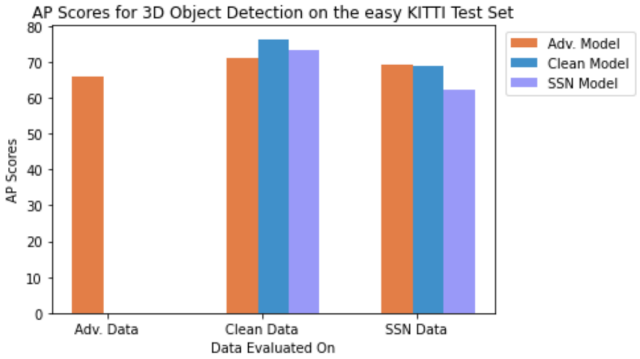

Figure 6. AP Scores of Models on Easy KITTI Validation Data

data. Specifically, we observe a comparable performance of the adversarial model to the other types of data, clean and SSN, but when comparing the adversarial inferences of the clean model and the SSN model, their performances dramatically dropped close to zero. For instance, the clean model dropped from a performance of 89.33 to 0.0536 from a switch from clean to adversarial data. Although the clean model had the highest performance on clean data, it is concerning that it was not robust at all to handle adversarial corruption. This indicates that the adversarial attacks were successful against the deep fusion models and proved their lack of robustness against adversarial attacks. However, the adversarial model proved to be robust against these attacks while maintaining comparable performance on the other benchmarks.

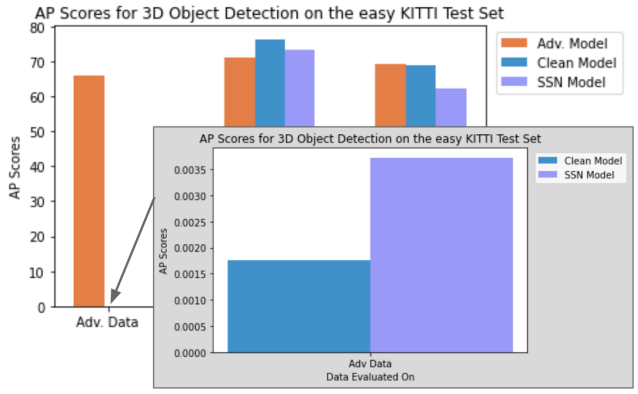

Figure 7. AP Scores of Models (Clean, SSN) on Easy KITTI Validation Data

6 Future Work

For further development within this research, we would like to experiment with different values of $\pm\epsilon$ to understand how we can best define the maximum perturbation. For our approach, we decided to choose a value that added a

Average Precision (AP) Score: 3D Object Detection			
Clean Data	Easy	Moderate	Hard
AVOD	76.41	72.74	66.86
+SSN	73.50	65.66	64.74
+ADV	71.09	63.75	63.73
SSN Data	Easy	Moderate	Hard
AVOD	69.04	55.08	54.63
+SSN	62.46	53.85	47.62
+ADV	69.12	54.89	54.55
ADV Data	Easy	Moderate	Hard
AVOD	0.0018	0.0035	0.0057
+SSN	0.0037	0.0105	0.0120
+ADV	66.11	60.40	61.04

Table 1. Car detection (3D) performance of AVOD with element-wise mean fusion layers against Gaussian SSN and Adversarial Examples on the KITTI validation set

Average Precision (AP) Score: BEV Object Detection			
Clean Data	Easy	Moderate	Hard
AVOD	89.33	86.49	79.44
+SSN	88.27	85.65	78.98
+ADV	86.97	78.47	78.29
SSN Data	Easy	Moderate	Hard
AVOD	87.77	78.38	78.41
+SSN	77.77	68.71	67.89
+ADV	87.83	78.40	78.33
ADV Data	Easy	Moderate	Hard
AVOD	0.0536	0.0890	0.1177
+SSN	0.0630	0.1264	0.1617
+ADV	83.85	76.84	76.43

Table 2. Car detection (BEV) performance of AVOD with element-wise mean fusion layers against Gaussian SSN and Adversarial Examples on the KITTI validation set

small perturbation to each value within the feature map but not drastic to the extent where the input data itself is completely manipulated. Additionally, we would like to certify the robustness of our model, specifically utilizing Chiang, Ping-yeh, et al.'s Certified Object Detection [3] approach for verifying robustness of object detectors for two categories of object detection: bounding-box and label. Their method involves smoothing based on Gaussian medians as opposed to Gaussian means and can ensure model robustness against all possible attackers and would be helpful in certifying that our model is robust against any generalized attack instead of just adversarial and random Gaussian noise.

7 Conclusion

We explored the importance of developing robustness in deep fusion modeling as seen in the area of autonomous vehicles. While there has been much research done in making these models as accurate as they can be, it is imperative that we focus on ensuring that the model can still make proper and reasonable inferences when faced with unforeseen circumstances. Through adversarial training, we were able to demonstrate that this is a viable approach in improving the robustness against single source corruption in addition to previous works. The adversarial model proved to be robust against both single source Gaussian noise as well as adversarial examples, whereas the other models performed extremely poorly against adversarial examples. While our models' performance was comparable on the other validation datasets, it is important that these models are robust against any attacks, especially those that are intentional like adversarial attacks from third parties. We hope our work inspires further exploration of using adversarial training in developing robustness.

References

[1] Kim, Taewan, and Joydeep Ghosh. "On single source robustness in deep fusion models." arXiv preprint arXiv:1906.04691 (2019). https://arxiv.org/pdf/1906.04691.pdf

[2] Park, Won, and Chen, Qi Alfred. “Crafting Adversarial Examples on 3D Object Detection Sensor Fusion Models.” arXiv preprint arXiv:2109.06363 (2021). https://arxiv.org/pdf/2109.06363.pdf

[3] Chiang, Ping-yeh, et al. "Detection as Regression: Certified Object Detection by Median Smoothing." arXiv preprint arXiv:2007.03730 (2020).https://arxiv.org/pdf/2007.03730.pdf

[4] Madry, Zico Kolter and Aleksander. “Adversarial Robustness - Theory and Practice.” Adversarial Robustness - Theory and Practice, https://adversarial-ml-tutorial.org/.

[5] Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237http://www.cvlibs.net/publications/Geiger2013IJRR.pdf

[6] Ku, Jason, et al. "Joint 3d proposal generation and object detection from view aggregation." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018. https://arxiv.org/pdf/1712.02294.pdf