Adversarial Attack Detection and Defense with High-level Image Property

Atul Prakash
EECS
University of Michigan
Ann Arbor, US
aparkash@umich.edu

Ryan Feng
EECS
University of Michigan
Ann Arbor, US
rtfeng@umich.edu

Jiaming Zeng
Data Science
University of Michigan
Ann Arbor, US
zjiaming@umich.edu

I. Abstract

Recent work have shown Deep Neural Networks (DNNs) to be vulnerable toward small perturbation with images present test time know as adversarial examples. Our contribution is to show how high-level image property including image structures can be used for ruling out adversarial examples from incorrect classes. Our main technical innovation is by applying high level algorithms and property for detecting adversarial attack toward DNNs, instead of further training the models for defending. The model is through pre-processing the training datasets and using a single model for filtering the possible classes.

II. Introduction

The Deep Neural Networks (DNNs) have been proven to be solid and been applied to multiple technique fields including the area of computer vision, machine learning and artificial intelligence. However, the DNNs has been shown to be vulnerable toward adversarial attacks which make small perturbation toward the images without human notice. Thus, the defending and detection of the adversarial examples are becoming extra important.

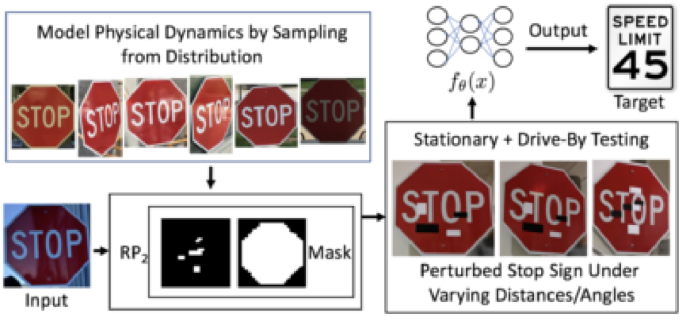

Fig. 1. An image of adversarial example.

Our contributions: 1. We introduce high-level models by taking advantage of k-means clustering and DISTS [2] image different metrics for detecting adversarial examples from distance differences with their true classes. 2. Given the results achieved from high-level methods, we propose a method for filtering the number of classes the adversarial examples can be classified with.

Fig. 2. An image of adversarial example.

III. Related Work

IV. Datasets

Currently, we have only experiment our algorithm with the GTSRB dataset, which is an image classification dataset consists of photos of traffic signs. There are a total of 43 classes with 39209 training images and 12630 testing images. We choose this dataset as a starting point for evaluating our adversarial examples filter.

V. Methodology

As a general guideline, we will use DISTS metrics to evaluate the distance between test images with our training images. K-means clustering would be applied for getting sub-classes from the original dataset. The specific algorithm for defending which contains two steps are explained below:

A. Preprocess Training Class

Our model requires an extra preprocessing on the image classes. We will use the GTSRB classes as an example. We first calculate the distance for all the training images within the same classes using the DISTS metrics. This would gives us a total of 43 matrix with the number of each classes as their length and width. We can also view the matrix as each images with a vector of distances between all other

images in this class. We then apply k-means clustering within each classes into new subclasses. The motivation for this extra preprocessing is to reduce the huge difference within each classes. By clustering the original classes into more subclasses, we are able to implement our algorithms better. Taking GTSRB as an example, we apply k-means clustering to each classes and get 8 subclasses for each class. We get a total of 344 subclasses for later training.

B. Training

The training step is through calculating the distance between held out dataset and training subclasses. We first choose three random images from each subclasses and use them as model images for each class. Then, we calculate the distances between the target held-out image with the three model images and get the mean. After calculating 100 test images with all 344 subclasses, we can get the lowest 10 value for each subclass, and use the lowest 10 as the threshold. Later in the filter process, this threshold value would be the filter. By calculating the distances between test images and model images, we can filter out all the classes for each image which have a distance larger than the threshold value.

VI. Evaluation Metrics

Currently, we will use Exact Match as our evaluation metrics. F1 could be applied later for a softer metrics.

Exact Match The EM measures how many actual results are included in the filtered predictions.

VII. Result

The current result at this point are quite small. We trained a neural network model with batch size of 64 and three convolution layers with 40 epochs. Then, we apply PGD attack toward the nerual network model and receives an accuracy relatively low. However, with applying the same adversarial examples toward the filters using distance metrics, the final results is around 55 percent.

References

[1] Goodfellow, I. J., Shlens, J., Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[2] Ding, K., Ma, K., Wang, S., Simoncelli, E. P. (2020). Image quality assessment: Unifying structure and texture similarity. arXiv preprint arXiv:2004.07728.