Defending Convolutional Neural Network-Based Object Detectors Against Adversarial Attacks

Jeffrey Cheng[1] and Victor Hu[2]

[1]Bridgewater Raritan Regional High School

[2]Watchung Hills Regional High School

1. Introduction

Artificial intelligence and machine learning are constantly growing in prevalence in everyday affairs, including self-driving cars and facial recognition security systems. Neural networks are at the center of these developments and allow autonomous vehicles or facial recognition systems to "learn" to classify images. As these networks become more commonplace, these networks must be able to address any situation with a minimal chance of failure. While a minor mistake in speech or facial recognition may not be particularly harmful, a missed traffic sign could result in harm, damages, or death. Researchers have developed methods to specifically target and trick neural networks into skipping or misclassifying important objects that it would otherwise detect. One such method is an adversarial attack, which is the emphasis of our study.

2. Background

2.1 Object Detection

Convolutional neural networks (CNNs) are a type of deep learning neural network that are especially useful in classifying pictures. Object detectors use a CNN, but they additionally have to detect where objects are located in an image before classifying them. This makes their job exponentially more difficult, as they must locate and take into consideration thousands of possible regions of interest within a single frame. The YOLO (You-Only-Look-Once) object detector is able to infer the possible location of an object in one forward pass of a single convolutional neural network, significantly expediting the object detection process. [6].

2.2 Adversarial Attacks

Adversarial examples are inputs which look more-or-less "ordinary" to a human, but actually cause a neural network to produce the wrong classification. Figure 1 shows an example of an adversarial attack. By freezing the weights and biases of a neural network, adversarial examples can be trained by using gradient descent and backwards propagation to minimize the loss function. Creating adversarial attacks that succeed in a physical environment are a larger challenge due to the variety of distortions in the real world, such as changes to lighting, angle, scale, and rotation, as well as camera noise.

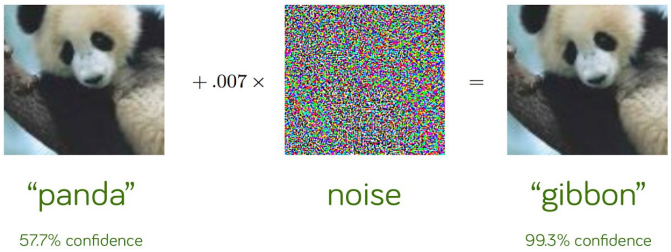

Figure 1: An example of an attack that generates adversarial pixel-noise atop an image to fool a classifier. The adversarial image looks seemingly identical to the original. These types of attacks are not robust enough to succeed in a physical environment. Image taken from [11].

3. Materials and Methods

Figure 2: Example of an adversarially perturbed stop sign we created designed to increase the probability of “Person” misclassifications.

3.1 Physical Adversarial Attack Against YOLOv3

We took on the scenario of placing an adversarial attack on a stop sign so that a neural network would no longer recognize the stop sign and an autonomous vehicle would fail to stop appropriately. To produce adversarial perturbations that would fool YOLOv3 in a physical environment, we used the Expectation over Transformation method [9], which generates random lighting, scalar, and rotational transformations of the attack. We modified the *ShapeShifter* attack [2], originally developed by Chen et al., to target YOLOv3 trained on the MS COCO dataset, which includes 80 different classes [6]. For our loss function, the classification probabilities across the 80 classes were fed into a softmax function to create mutually exclusive class probabilities, and cross-entropy loss was computed with the target goal of 100% Person.

3.2 Proposed Defenses

To defend against our adversarial attack, we propose two approaches: color thresholding, as well as haar-feature verification. Color thresholding is essentially the chromatic equivalent to

image binarization, since convolutional neural networks take color into account when making inferences. Since we want to maximize the uniform redness of a stop sign while trying to avoid altering the rest of the image, the pixel values were "snapped" to red if the red-green or red-blue ratio was sufficiently high. By inserting a color thresholding step into the image preprocessing pipeline, we hope to minimize the human-like features that cause the object detector to misclassify the adversarial stop sign as human.

Our second proposed defense relies on error-checking the inferences of the neural network using haar features. Haar features are based on finding different features based on the average whiteness and blackness of different areas in the image. We believe that since haar features are based off of average white and black areas, the adversarial perturbations would have little effect upon them. By implementing a haar cascade classifier as the second step in our classification pipeline, we hope to rectify "Person" misdetections outputted by YOLOv3.

3.3 Experimental Setup

We used Raspberry PI and camera mounted on an Arduino robot car to mimic the changing scale, lighting, and angle conditions that would occur as a car approaches a road sign, to further test the physical robustness of the adversarial attack.

4. Results

The experiment was run with a regular stop sign, a stop sign with sticker graffiti, and the adversarial stop sign. The regular stop sign was used as a control group as a baseline for the object detector's performance. The independent variable would be the category of stop sign, while the dependent variables would be detection rates and confidence levels. Three trials were conducted for each experimental configuration, and the results were averaged together.

Experiment Configuration	**Average "Stop Sign" Frames**	**Average "Stop Sign" Conf.**	**Average "Person" Frames**	**Average "Person" Conf.**
Regular Stop Sign	100% (709)	0.999	0% (0)	0.000
Stop Sign with Sticker Graffiti	89.02% (605)	0.746	0% (0)	0.000
Adversarial Stop Sign	58.74% (358)	0.545	66.90% (407)	0.535

Table 1: Summary of data for the experimental trials. Predictions above a 50% confidence threshold were considered detections. For the detections in each frame, the highest confidence prediction was used to calculate the average confidence. Note "stop sign" and "person" detections are not mutually exclusive.

	Average "Stop Sign" Frames	Average "Stop Sign" Conf.	Average "Person" Frames	Average "Person" Conf.
Color Thresholding Defense	99.67% (607)	0.983	0% (0)	0.000

Table 2: Results after implementing our color thresholding defense. See Table 5 in Appendix A for complete data.

	Average "Stop Sign" Frames After Defense Checking	Average "Person" Frames After Defense Checking
Haar Classifier Defense	99.84% (608)	0.16% (1)

Table 3: Results after implementing our haar feature defense against the adversarial stop sign. See Table 6 in Appendix A for complete data.

5. Discussion and Conclusions

Looking at Table 1, the regular and graffitied stop signs were detected in at least 89% of the frames with high confidence and zero "Person" detections, demonstrating that YOLOv3 was exceptionally capable of detecting stop signs under normal conditions. When we added the adversarial attack, however, the successful "stop sign" detection rate plummeted to a meager 58% with 67% of the frames mis-detecting a person. To counteract this, we added the color thresholding and haar classifier defenses as proposed before, and successful detection rates shot back up to over 99% with only one "Person" misdetection between all three trials of both defenses.

Ultimately, the results show that our adversarial attack poses a realistic threat in a safety-critical situation like riding in an autonomous car. A stop sign detection rate and confidence level of only around 50% is nowhere near reliable enough for use in real life, in addition to the fact that the object detector would be actively confused with the faulty "person" classifications. However, our research demonstrates that with our proposed defenses against adversarial attacks, stop sign detection and confidence rates return to near-optimal levels.

References

[1] Thyst, S., Ranst W. V., and Goedemé, T. 2019. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. *arXiv preprint arXiv:1904.08653.*

[2] Chen, S., Cornelius, C., Martin, J., and Chau, D. H. 2019. ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. *arXiv preprint arXiv:1804.05810.*

[3] Liu, X., Yang, H., Liu, Z., Song, L., Li, H., and Chen, Y. 2019. DPatch: An Adversarial Patch Attack on Object Detectors. *arXiv preprint arXiv:1806.02299.*

[4] Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., and Chen, K. 2019. Seeing isn't Believing: Towards More Robust Adversarial Attack Towards Real World Object Detectors. *arXiv preprint arXiv:1812.10217.*

[5] Papernot, N., McDaniel, P., and Goodfellow, I. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv preprint arXiv:1605.07277.*

[6] Redmond, J., Divvala, S., Girshick, R., Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. *arXiv preprint arXiv:1506.02640.*

[7] Ren, S., He, K., Girshick, R., Sun, J. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv preprint arXiv:1506.01497.*

[8] Redmon, J., Farhadi, A. 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767.*

[9] Athalye, A., Engstrom, L., Ilyas, A., Kwok, K. 2018. Synthesizing Robust Adversarial Examples. *arXiv preprint arXiv:1707.07397.*

[10] Yuan, X., He, P., Zhu, Q., Li, X. 2018. Adversarial Examples: Attacks and Defenses for Deep Learning. *arXiv preprint arXiv:1712.07107.*

[11] Goodfellow, I., Shlens, J., Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572.*

[12] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramèr, F., Prakash, A., Kohno, T., Song, D. 2018. Physical Adversarial Examples for Object Detectors. *arXiv preprint arXiv:1807.07769.*

[13] Eykholt, K., Gupta, S., Prakash, A., Rahmati, A., Vaishnavi, P., Zheng, H. 2019 Robust Classification using Robust Feature Augmentation. *arXiv preprint arXiv:1905.10904.*