Defense on Robust Physical Attack Against Road Sign Classifier

*

Nushrat Humaira
School of Computing
Clemson University
Clemson, SC, USA
nhumair@clemson.edu

Reetayan Das
School of Computing
Clemson University
Clemson, SC, USA
reetayd@clemson.edu

Urvi Patel
School of Computing
Clemson University
Clemson, SC, USA
urvip@clemson.edu

***Abstract*—Deep Learning is a subset of Machine Learning in artificial Intelligence(AI) with networks that are capable of learning from unstructured data. With this, we have seen Deep Neural Networks (DNNs) to be vulnerable to adversarial examples. These are inputs the attacker intentionally designs for machine learning models to cause them to make a mistake. Specially, classifier networks fell prey to perturbed examples and were mislead into targetted or non-targetted misclassification attacks. The effect of adversarial attack in physical world is different, however, as physical aspects of environment are brought into consideration.We have chosen *Robust Physical Perturbation* attack for physical world,termed as RP2 as foundation for our project. RP2 generates adversarial example for road sign images captured in practical driving scenario. Our proposed approach is to evaluate the robustness of physical world attack on road sign benchmark dataset and employ various defensive methods against the attack. Defending attacks lessens vulnerability of classifier and is useful to gain insights into future attacks of the same nature. We propose few novel strategies of defense as well as apply established defense methods to make it work for physical road sign perturbations. We analyze the effectiveness of methods, problems encountered and possible improvements.**

***Index Terms*—DNN,classification, Perturbations, Adversarial examples, robustness**

INTRODUCTION

Recent work has demonstrated that DNNs are very vulnerable to adversarial perturbations. These carefully crafted modifications to the (visual) input of DNNs can cause the systems they control to misbehave in unexpected and potentially dangerous ways. This is one of the major reasons Eykholt et al. [1] chose the Road sign images dataset as their input. They had several other reasons to choose the road sign classification as their target domain. One of them being the relative visual simplicity of road signs making it challenging to hide perturbations. The second reason was that the road signs exist in a noisy unconstrained environment with changing physical conditions such as the distance and angle of the viewing camera, this in turn implied that the physical adversarial perturbations would need to be robust against considerable environmental instability. The third reason being the importance of road signs in transportation safety. Also that an attacker may not be able to have control over the vehicle's system but they can modify the objects (road signs) in the physical world that the vehicle depends on to make safety crucial decisions.

Keeping all the above considerations in mind, Eykholt et al. [1] proposed Robust Physical Perturbations(RP2) to add perturbations to road sign object by sampling from camera distance and angle distribution. Perturbations created using RP2 was constrained only to road sign and was printed to emulate graffiti, which is common form of vandalism, and thus hide in the human psyche. To create the robust perturbations, the algorithm drew samples from a distribution that modelled physical dynamics like varying distances and angles using experimental data and synthetic transformations. RP2 can be extended to other physical objects as well, sticker attack was successful in deceiving Inception_V3 classifier to misclassify a mug as cash machine. They proposed an evaluation methodology to study the effectiveness of physical perturbations using two standard architecture classifiers that they built: LISA-CNN that achieved 91 percent accuracy on that dataset and GTRSB-CNN with 95.7 percent accuracy on the GTRSB test set.

Fig. 1: Shows real graffiti on a Stop sign

However, evaluation methods and experimental results were collected in certain set up that may need to be reproduced exactly every time to achieve the desired attack success rate. That leads to the uncertainty that attack may not as full proof as it claimed. Robustness of attacks against trained classifier

Fig. 2: Shows designed perturbations to mimic graffiti to hide in the human psyche

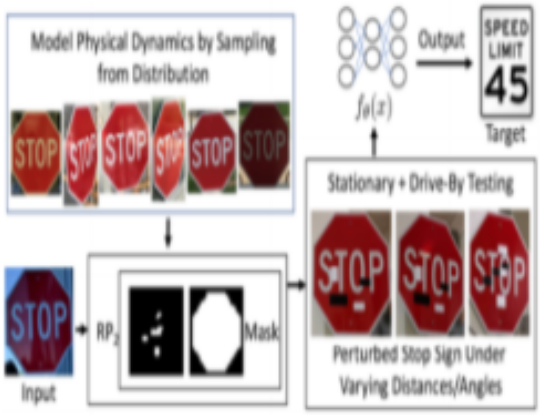


Fig. 3: RP2 pipeline overview [1]. The input is the target Stop sign. The adversary sticks out the resulting perturbations and sticks them to the target Stop sign.

were not compared with classifiers used by Weymo, Uber for their autonomous driving test. Training set of images was not very large to draw any conclusive strength of attack for all kinds of inputs. Furthermore, the paper didn't illustrate how any of state-of-the-art defensive method for adversarial attack fares against RP2. Lack of proof of robustness against defense motivated us for the project. We proposed several novel and existing defense strategies and applied it to counter the attacks successfully and provide insight on improving attack success rate. In this paper, we discuss the approaches we took to defend the attacks against the classifier and demonstrate our findings.

Related Work

Carlini and Wagner's C&W attack [3] consisted of three types of attack for adversarial example to defeat defensive distillation. They devised attacks for all three norms L1, L2 and L which are used to measure the deviation of the adversarial perturbation from the original sample. Zhao et al. [4] founded their study on CW attack. They generated examples based on ADMM(Alternating Direction Method of Multipliers) and adapted easily to L2 and L1 attacks. Their framework claimed to require less distortion compared to C&W, better transferable, immune to most common defenses such as distillation, adversarial retraining. While CW needed to run L2 attack for L0 attack in each iteration, this method is more generalized and adapted to both attacks separately.ADMM decomposes the initial problem of finding minimum distortion to the input for a given target label, into two sub problems. This strategy made it easier to optimize and converge faster. Attack success rates for ADMM based L2 attack achieved same as CW attack for best, worst and average target attack scenario. RP2 by Eykholt et al. [1] was built on CW. Lu et al. [5] argued that adversarial examples are less of security threat for object detection in Autonomous Vehicles as assumed because of the varying surrounding physical conditions in a moving car. Noisy physical environment destroy digitally formed perturbations. Later Lu et al. [6] illustrated that strong object detectors such as YOLO or Faster R-CNN are not deceived by RP2. Zeng et al. [7] claimed that digital image perturbations doesn't work well in real world. Athalye et al. [2] showed that 3D printed objects work very well for adversarial attacks in the physical world. This paper model a space of physical and digital transformation across different angles and viewpoints. 'Expectation Over Transformation' (EOT) framework constructs adversarial examples over distribution of image and object transformations. Compared to RP2 who uses combination of physical and synthetic transformations, EOT uses mostly simulation.

Defenses and Counter measures

Defenses against adversarial perturbations have been developed based on application field, type of data and the desired outcome. An attack is considered robust if it is successful despite defensive measures taken. Targeted attack caused LISA-CNN classifier to choose the target adversarial class instead of the true class. Since the classifier model architecture, input/output, hyperparameters are available to the attacker, Robust Physical Perturbation(RP2) can be considered as black box attack to LISA-CNN and GTRSB-CNN classifier. Due to time constraint, we worked on defending LISA-CNN classifier only. We briefly describe the directions and categories of defensive methods that we have considered to defend RP2 attack.

Based on our research, we decided on three approaches for the defense against the RP2.

- Modifying training or input.
- Modifying the network.
- Adding external layers to the network.

The first approach was trying to modify the inputs to try and train the classifier to work with input variations and still be able to predict the inputs given into their correct classes. An example of such a strategy is the adversarial training. Adversarial training is when the adversarial examples created are used during training to reduce misclassification. Each sample is re-trained with original classifier along with different perturbed inputs for the sample. Goodfellow et al. [11] and Huang et al. [12] included adversarial examples in the training stage as a countermeasure to make neural networks more robust, provide regularization for deep neural networks and improve precision as well.

The second approach was modifying the network. In this approach, we applied Gradient regularization or masking. It was applied as a part of a preliminary investigation. From universal perturbations proposed by Moosavi-Dezfooli et al. [10], it is evident that a very small perturbation value added to the image is almost possible to be detected by the human eye. Gradient masking is one of the approaches where we can see it is possible to revert back to. For that, anyone can take a value less than the maximum value of RGCC, that is, 255. This can be applied as the "mask" on the image as seen in 4.

Masking.PNG

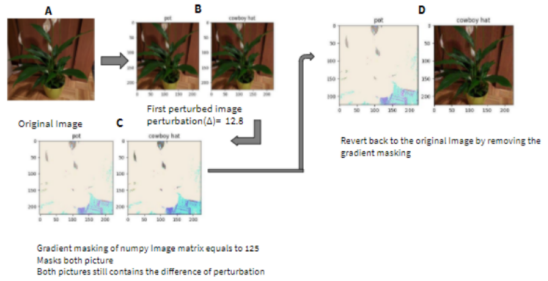


Fig. 4: Gradient Masking Result

Modifying the network approach involves adjusting network architecture/parameters during training phase. Unlike this, third approach appends external model to existing architecture. We applied Siamese network as add-on to the original network as pre-processing of inputs. For a set of images, siamese network would group images based on their feature similarity who belong to same class label. Based on the predicted similarity, perturbed images would be recognized.

IMPLEMENTATION

Robust Physical Perturbation attack on LISA-CNN, GTRSB-CNN and Imagenet Classifier have been implemented and published along with experimental images and optimized model output. Repository includes stop sign images taken by the researchers themselves inside their laboratory and around campus streets. We replicated the subliminal poster attack and sticker attack using one mask to confine the region of perturbation to the octagon boundary of road sign. Dataset used for both attacks consisted of stop sign images taken from different angles and distances and resized to 32 x 32 dimensions. Experiments were conducted through Anaconda environment using batch jobs on Palmetto Cluster GPU nodes. Original source code used tensorflow version 1.4.1 and keras version 1.2.0., which we upgraded to work with Tensorflow core 2.0 API as well. Optimization run for subliminal poster attack on 300 epoch showed that perturbation caused average 99.9 percent times LISA-CNN misclassified images. Adversarial loss was calculated as the l2-norm distance between target attack class and adversarial predictions by the model, combined with regularization loss. Adversarial loss was measured at 0.2 for initial run.

Defense techniques that we applied were implemented by reproducing the original attack in separate modules and using adversarial images created by optimized output provided by the authors. Following subsections describe those techniques and experimental results.

A. Modifying input to network

This kind of defense manipulates the input and target class used for training the network to recognize adversarial examples and makes the classifier more robust. A classifier learns to minimize the cross entropy loss and learns the target label. For a image that has been perturbed, where y is the true label and y* is the target label, a misclassification leads to targetted attack if y* is chosen. It is considered successful attack when clean image would be correctly classified otherwise. RP2 considered a fixed class label as its target class. For our approach, we wanted to decrease the attack success rate by manipulating adversarial target and adversarial loss.

1) Randomize targeted attack on LISA-CNN: Subliminal poster attack recreated the attack by Kurakin et al. [13] with the exception of keeping the region of adversarial perturbation to the road sign without any modification to background. The method used was Fast Gradient Sign method. FGSM perturbs the gradient direction of features of input image to the point that classifier would choose incorrect target. if loss was defined by $J(X, y_t)$ where X is input image, y_t is target class, adversarial image would be generated as (1).

$$X_{adv} = X - \epsilon sign(\Delta_X J(X, y_t)) \tag{1}$$

One-step target class' FGSM uses label of the least likely class of image instead of true label. Inspired by this, we modified the source code to choose randomly sampled adversarial target class as label to predict, rather than a fixed class. For each epoch, random class label chosen from distribution of 17 road sign categories, were fed into the model. This was successful bringing down the targeted attack success rate, percent of misclassified images, average of misclassified image rate and increased adversarial loss. We observed percent average misclassified images came down to 93.9 and percent average adversarial loss on random sampling went 4.2, that indicated adversarial target was not completely successful in fooling the network as before with fixed adversarial target class. Result of 20 runs of subliminal poster attack with randomly sampled adverasarial target class is shown in Table. I

B. Image retrieval and matching

Our novel idea originated from the concept to retrieve original images removing the perturbations. Model trained on the pixel distribution of image would be able to reconstruct it given noisy image. By transforming images into continuous value of pixel distribution we tried to distinguish among perturbed and clean pixels. Our initial choice was linear regression model which didn't produce quite satisfactory unperturbed images.

Next we implemented Deep Siamese network for image retrieval. Trained model extracts features from both perturbed

TABLE I: Percent Average misclassification(20 run of 300 epoch each

Percent Average misclassifcied images
percent average misclassified images 93.4
percent average misclassified images 94.7
percent average misclassified images 91.8
percent average misclassified images 87.6
percent average misclassified images 94.5
percent average misclassified images 91.5
percent average misclassified images 90.9
percent average misclassified images 90.0
percent average misclassified images 91.6
percent average misclassified images 92.4
percent average misclassified images 90.6
percent average misclassified images 93.6
percent average misclassified images 89.5
percent average misclassified images 89.8
percent average misclassified images 92.8
percent average misclassified images 91.5
percent average misclassified images 95.6
percent average misclassified images 94.5
percent average misclassified images 89.4

and clean images and learns similarity among those. Both of these models are described in detail below.

1) Linear Regression Model: Linear Regression Model was trained with the pixel values of the perturbed images and tested against clean images of the LISA CNN Data Set. Our expectation was that to retrieve a clean image and if the classifier detect it as a Stop sign then to show our conjecture that a good defense created on their proposed attack. In time of training, we had to convert each training and testing image to numpy array which holds the continuous pixel value in range 0 to 1.After reshaping the numpy array we got a Data-Frame of three columns were holding "R","G","B" values. Then we had to encode the pixel value to categorical "0" and "1" based on a particular checker function depends on a particular threshold value. Pipeline is showed in Fig. 5. The

Fig. 5: Image retrieval with Linear regression

Checker Algorithm in Fig. 6 threshold value was not sure primarily, so we applied trial and error many times to get the highest accuracy. The Best accuracy was 99.86 %, which was possible due to manipulation of the threshold value to 0.55. So any pixel value that came below 0.55 was encoded as 0, else encoded as 1. Even the accuracy proved higher with a visual representation of precision and recall in confusion matrix (see Fig. 7. From the model we learned that, the linear regression is a very good classifier after training a decent amount of perturbed images, but the predicted image is hard to recognize as a stop sign in human eye.

2) Siamese network: Siamese network takes two input images in parallel where either one can be perturbed or clean and find the similarity score based on features. For

```
In [557]: cutoff = 0.55
          y_pred_classes = np.zeros_like(Y_pred2)
          y_pred_classes[Y_pred2 > cutoff] = 1

          y_pred_classes
          #y_pred_classes.shape

Out[557]: array([[1., 1., 1.],
                 [0., 0., 0.],
                 [0., 1., 1.],
                 ...,
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.]], dtype=float32)
```

Fig. 6

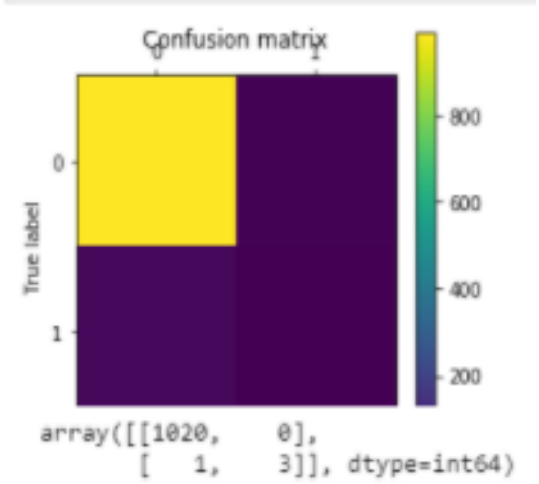


Fig. 7

road sign images that belong to same category i.e stop or speedLimit45 we can naively assume they would naturally have high similarity. But perturbed images or images taken from different angle and distances would appear different and classifier would predict them being in different classes. Thus convolution neural network was used to generate fixed length features vectors from images and siamese network then find the similarity score. Similarity score is squished between 0 and 1 using a sigmoid function,where 0 denotes no similarity and 1 denotes full similarity.The goal is to learn the similarity function. Siamese network architecture(source: One-Shot) is shown in Fig. 8. We implemented two Siamese

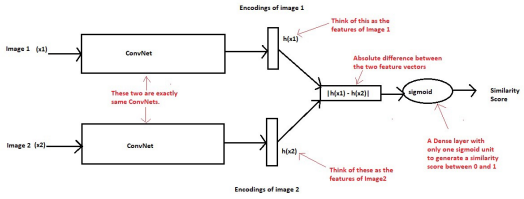


Fig. 8: Siamese Network Architecture

network architecture, primarily we used a separate feature generation model to extract from image and was trained for both images in pair. Then those features were merged and propagated through the rest of the network , finally connected to FCN layer with sigmoid activation to produce similarity score.Fig. 9 shows our primary siamese network architecture. Our secondary architecture uses a more compact form to generate feature vectors and uses different convolutional kernels to extract the features. Most importantly, in Lambda layer this network calculates the element wise absolute difference between two features using cosine similarity and then pass that on to sigmoid activated FCN layer to generate similarity score as shown in 10. The model was compiled using the adam

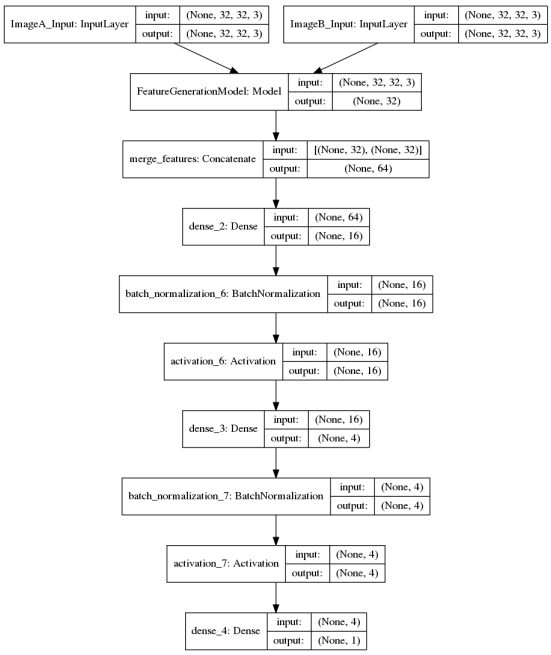


Fig. 9: Siamese Network Architecture 1

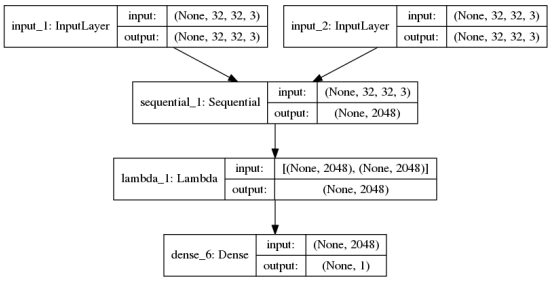


Fig. 10: Siamese Network Architecture 2

optimizer and binary cross entropy loss function with very low learning rate as higher rate increased convergence time. Pretrained model was tested on randomly selected images from different categories. Each pair of stop sign images along with actual similarity based on same class in first row and predicted similarity in second row is shown in Fig. 11. We can observe that noisy images created from sticker perturbation attack and clean images sampled from angle and distance distribution has higher similarity score.

C. Detection only approaches

D. Sticker segmentation with UNet

For semantic segmentation, we label each pixel of an image with a corresponding class of what is being represented. The output itself is a high resolution image (typically of the same size as input image) in which each pixel is classified to a particular class. Thus it is a pixel level image classification. Inspired by this, we wanted to segment perturbed stickers on the image by semantic segmentation with UNet. Model architecture is shown in Fig. 14. The UNET was developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation. The architecture contains two paths. First path is the contraction path (also called as the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called as the decoder) which is used to enable precise localization using transposed convolutions. Thus it is an end-to-end fully convolutional network (FCN), i.e. it only contains Convolutional layers and does not contain any Dense layer because of which it can accept image of any size.Training result of Unet on pixel segmentation is showed on 12. Fig. 13 shows sticker clearly segmented by UNet based on the L1 based mask. For each pixel we get a value between 0 to 1. 0 represents no sticker and 1 represents sticker.We take 0.5 as the threshold to decide whether to classify a pixel as 0 or 1. However deciding threshold is tricky and can be treated as another hyper parameter.

1) Perturbed pixel identification: We wanted to identify the sticker position of the perturbed image. We converted a perturbed image but this time we took images in JPEG format instead of PNG format. We discovered a JPEG Image contains the pixel value in RGB color Standard which ranges from 0 to 255 whereas PNG formatted image's pixel value lies in the range of "0" to "1". Unfortunately we haven't had sufficient amount of JPEG formatted stop sign perturbed images. We introduced our novel Sticker Identifier Algorithm. We wanted to identify where black or near black colored clearly visible stickers were added to the road sign. Once the images were transformed into numpy array of pixel value distribution, we looked for patterns matching the value of pixels that would indicate perturbation. The graph in Fig. 15 depicts the density plot of R,G and B Values based on the position we have identified in the Data-Frame. Sticker Identification Algorithm goes as follows:

(i) Convert the perturbed (filled with stickers) image into numpy array.
(ii) Check which of the values in the numpy array holds the particular RGB color where we applied np.where(array = 'value') . In our case the value was "0" for the black stickers.
(iii) Transpose the array to get the R,G,B values column wise.

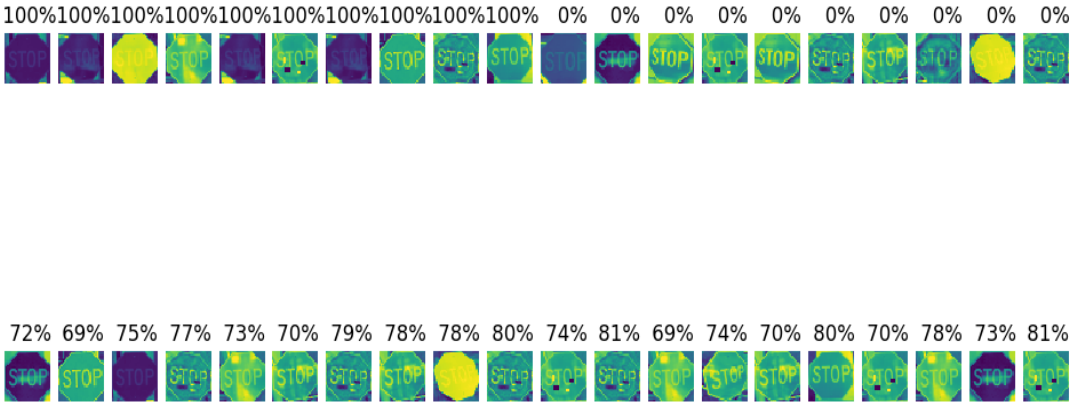


Fig. 11: Simese model output visualization

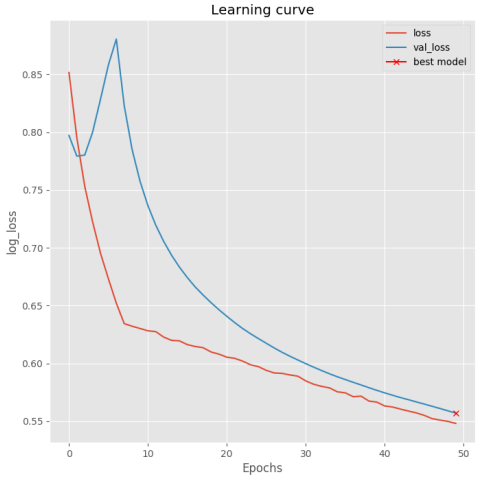


Fig. 12: Learning curve of UNet sticker segmentation

(iv) Form a Joint plot with matplotlib. The joint plot will show the color distribution of the stickers by pixel.
(v) From the joint plot, its easy to get the position of the stickers and its easy to make a boundary box which could be mapped into original image.

As we were successfully able to identify the stickers, according to the last step we were able to make to boundary box and mapped into our original perturbed image. In Fig. 16 its shown

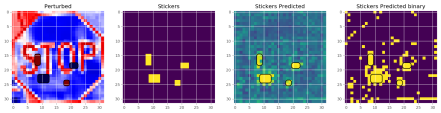
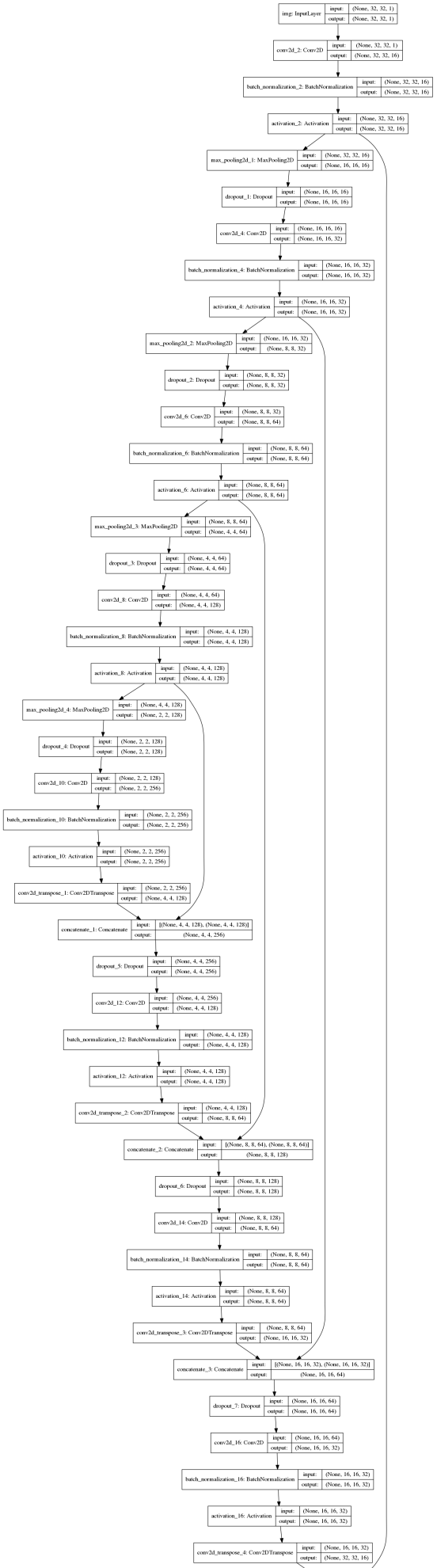


Fig. 13: Stickers predicted by UNet

that two stickers were correctly identified by two red colored boundary boxes. We have come up with few hypothesis of our approach. User/Experimenter should know the color code of the stickers. The stickers should contain monotone color. Our Algorithm processed each pixel of the image we have converted the whole image into numpy array. So for identification of any perturbation/stickers, it is actually possible to identify them even in the background also. The author's limitation was that there RP2 algorithm didn't work on the background. In this case, we have successfully overcome their limitation.

2) Detect adversarial perturbation using PCA: Unlike clean images, adversarial images created using FGSM method has larger coefficients or weights on principal components. This was discovered by [14]. They used variance and coefficient to detect adversarial images. Original method was tested on MNIST. We observed it couldn't detect perturbations on perturbed road sign images given in paper by Evtimov



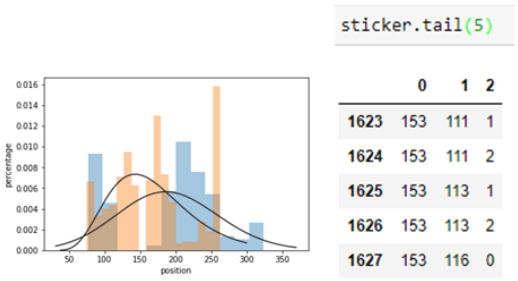


Fig. 15: Density plot of R,G,B values

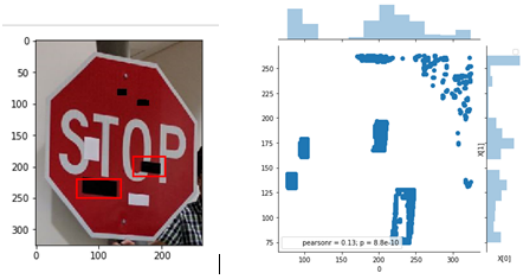


Fig. 16: Boundary box and joint plots

et al. For original and adversarial images, PCA components remained same for our implementation. Further analysis is needed to discover the issues.

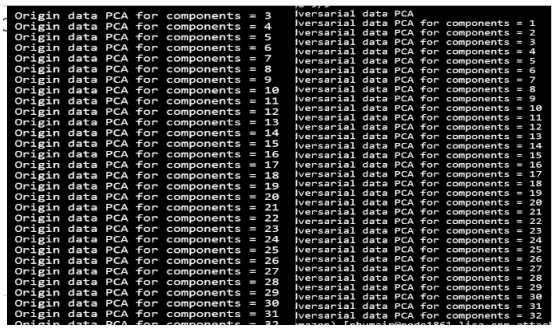


Fig. 17: Detection of perturbations using PCA

EVALUATION

We analyzed defense method along with applicability. We have observed that detection based approaches work well in countering adversarial attack on road sign images. The reason is early detection stops the classifier from learning misleading and adversarial target for perturbed images. We focused on detecting perturbed stickers on images before they were fed into the classifier. Our novel approach for sticker identification using pixel distribution worked well when assumption on perturbed pixel values were nearly accurate. Siamese network worked quite well on detecting perturbation on images with various angles and distances.

Conclusion

Adversarial attacks cause visual classifiers to produce incorrect prediction. For physical world visual images, adversarial perturbation lead to serious consequences. We implemented several defense strategies to counter the robust physical perturbations and reduce the degree of incorrect classifier prediction. We encountered several challenges with dataset and implementation as source repository released by authors had resized images in only one category, stop sign. To run our defense experiments, we collected the original LISA U.S traffic sign dataset and extracted the annotated signs from the video frames for all categories. We used a short subset of the images for our experiments. Due to preprocessing involved, we couldn't complete as many defense strategies we would have preferred, still we tried our best to defend the attacks with our novel and existing defense techniques. Our project code is available on the github repository https://github.com/nushrathumaira/8580_final_project.git

Project Contributions

Table. II shows timeline of our project with all milestones, tasks within each milestone and person responsible for task.

Acknowledgement

We are grateful to Dr. Hongxin Hu for providing us with an opportunity to work on the project as well as giving us constant guidelines and advice on how to better the projects at each step. Special thanks to Nishant Vishwamitra, TA for providing us with extra guidelines and opinions on how to go about the project methods and correct strategies to accomplish the goal of the project.

References

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," arXiv preprint arXiv:1707.07397, 2017.

[3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 39–57.

[4] P. Zhao, K. Xu, T. Zhang, M. Fardad, Y. Wang, and X. Lin, "Reinforced adversarial attacks on deep neural networks using admm," in 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2018, pp. 1169–1173.

[5] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," arXiv preprint arXiv:1707.03501, 2017.

[6] ——, "Standard detectors aren't (currently) fooled by physical adversarial stop signs," arXiv preprint arXiv:1710.03337, 2017.

[7] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, and A. L. Yuille, "Adversarial attacks beyond the image space," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4302–4311.

[8] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 1528–1540.

[9] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in Thirty-second AAAI conference on artificial intelligence, 2018.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1765–1773.

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

[12] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," arXiv preprint arXiv:1511.03034, 2015.

[13] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world." arXiv preprint arXiv:1607.02533 ,2016.

[14] Hendrycks, Dan, and Kevin Gimpel. "Early methods for detecting adversarial images." arXiv preprint arXiv:1608.00530,2016.

Milestone	Task	Handler	Start Date	End Date
Project Proposal	Related work research	All	October 2	October 7
	Gradient Masking	Reetayanl	October 2	October 7
	Install environment and run source code	Nushrat	October 2	October 7
	Write Proposal	All	October 2	October 7
Defense strategy development	Run and produce result from source code	Nushrat	October 8	October 16
	Linear regression model development	Reetayan	October 8	October 16
	Research defense strategies	Urvi	October 8	October 16
Midterm Presentation	Reproduce attacks	Nushrat	October 24	October 30
	linear regression model improvement and image retrieval	Reetayan	October 17	October 30
	Training regression model	Urvi	October 17	October 30
Final presentation	Randomized target attack and PCA detection	Nushrat	November 5	December 2
	Developed novel perturbed sticker pixel detection	Reetayan	November 5	December 2
	Regression model new result	Urvi	November 5	December 2
Final report	Siamese network and UNet complete work	Nushrat	December 3	December 11
	Completed Sticker attack detection	Reetayan	December 3	December 11
	Training and evaluation	Urvi	December 3	December 11
	Final report	All	December 3	December 11

TABLE II: This shows a detailed breakdown of the schedules, including milestone, tasks and timeline