AI Approach for Autonomous vehicles to Defend from Adversarial Attacks

Final Year Project Review

Members :
Prachee Gupta (BT17ECE008)
Kritika Dhawale (BT17ECE042)

Guide : **Dr. Tapan Kumar Jain**

Department of Electronics and Communication Engineering

Contents

1.

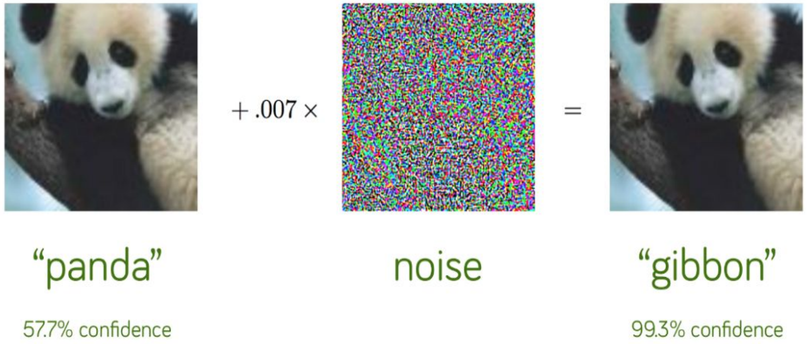
Are Neural Networks Really worth the Hype?

Let’s start our discussion on Neural Networks

Fooling a Machine Learning Model

In 2014, a group of researchers at Google and NYU found that it was far too easy to fool ConvNets with an imperceivable, but carefully constructed nudge in the input.

They added some carefully constructed noise to the input and the same neural network now predicts the image to be that of a gibbon!



Source: [Explaining and Harnessing Adversarial Examples](), Goodfellow et al, ICLR 2015.

Adversarial Attacks

Trying to Fool a Sign Recognition Model?

- Imp for an automated vehicle to recognize the sign correctly.
- What if somebody tries to manipulate the sign board with adversarial attacks?
- Intentionally just to fool the ML model
- So that the model will not predict the sign correctly.

classified as

Stop Sign

Imagine Self Driving Cars misclassifying “Stop” sign to “Speed Limit 100”

classified as
Stop Sign

+

Noise

=

classified as
Max Speed 100

Adversarial
Attacked
Sign Board

What we want to achieve?

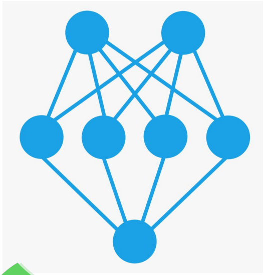
- Design a model that is robust and can handle multiple type of attacks on traffic signs.
- Despite of those attacks, our model should not be fooled and identify the sign correctly.
- Train the model with a defense mechanism to remove the adversarial noise.

Thought Process (Normal DL Models)

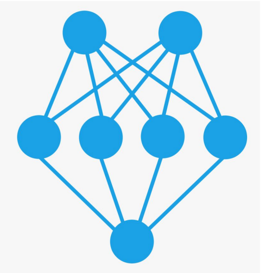
Raw Image

Attacked with FGSM or PGD

Attacked image which appears similar to original Image

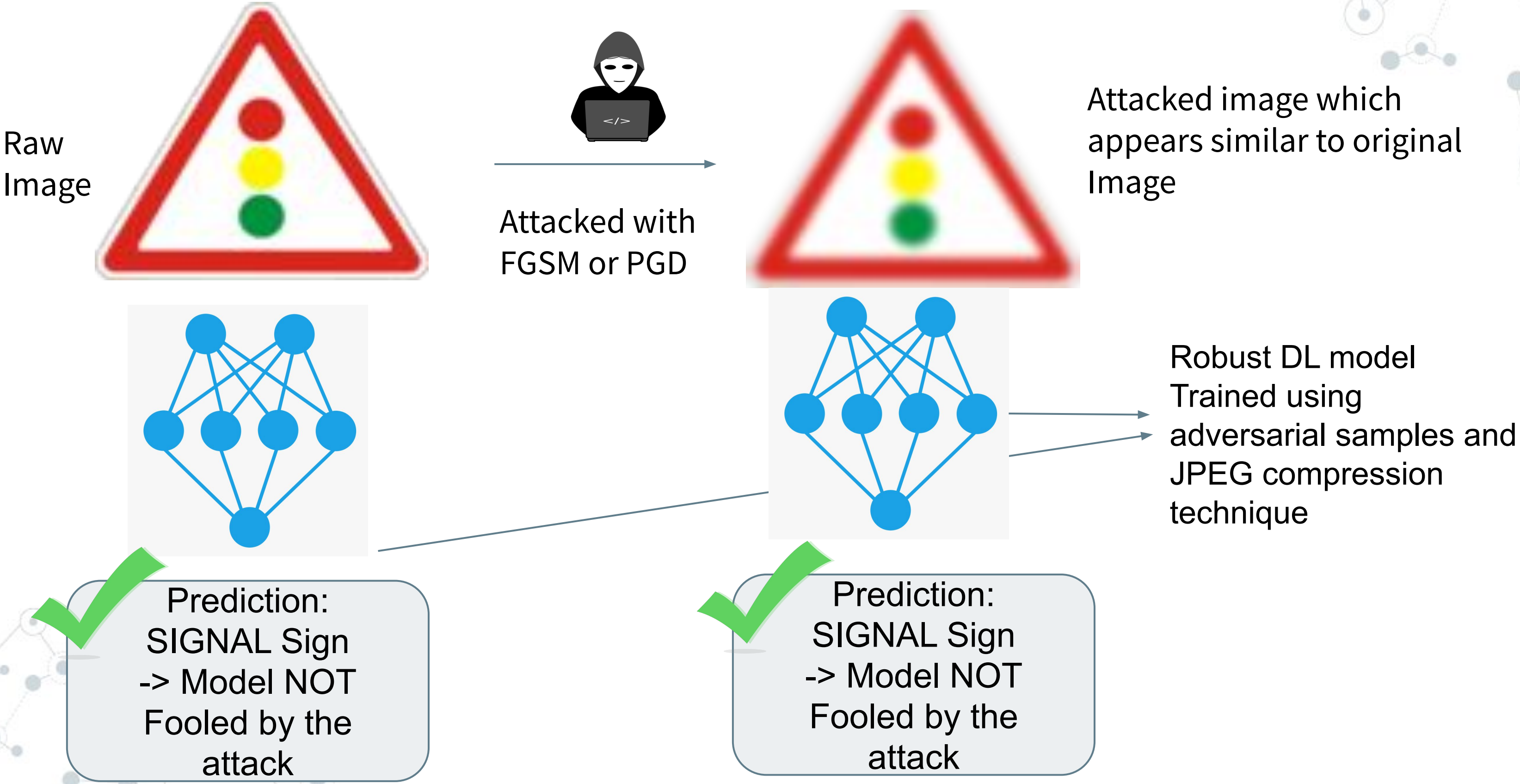


Prediction:
SIGNAL Sign
-> Model NOT Fooled by the attack



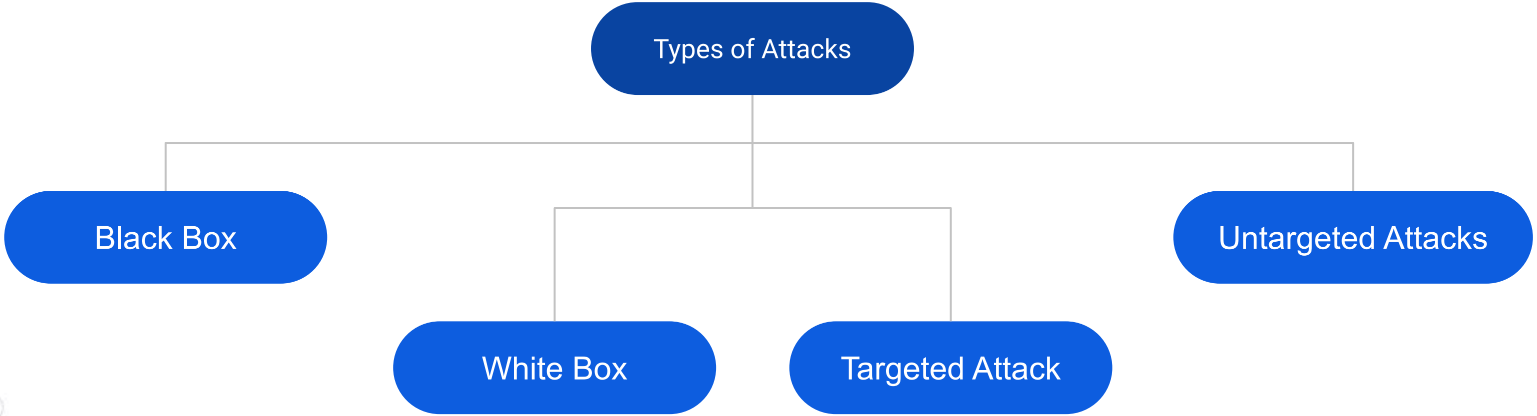
Prediction:
STOP Sign
-> Model is Fooled by the attack

Thought Process (Robust DL Models)



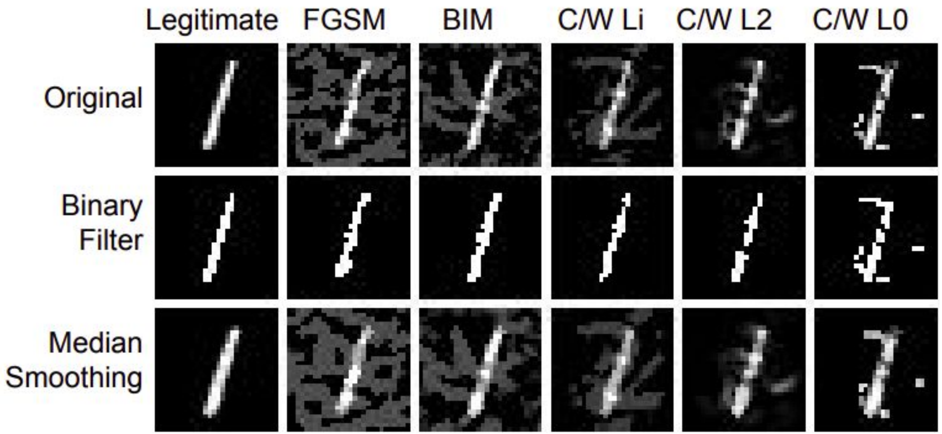
Types of Attacks

Attacks can be Classified Into 4 different Types[1]:



Attack Methods: [2]

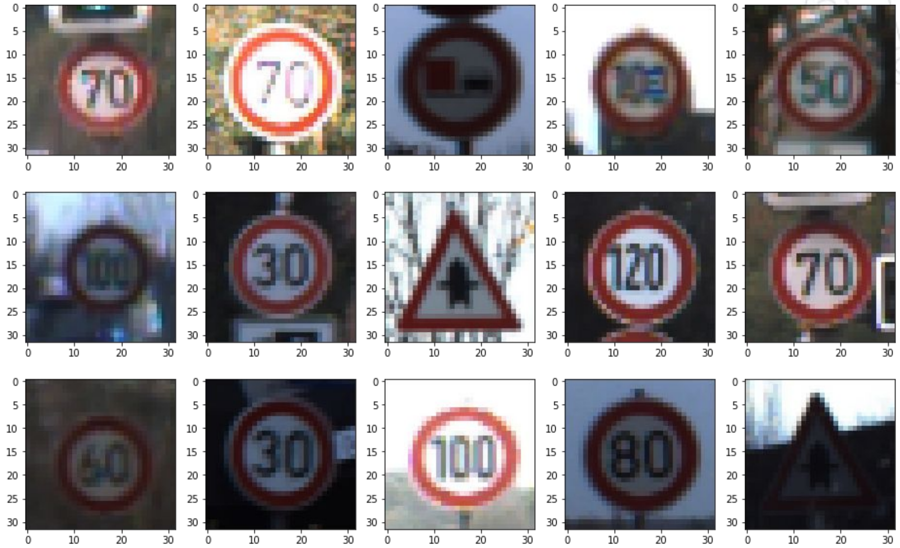
- Fast Gradient Sign Method
- Projected Gradient Descent (PGD)
- Target Class Gradient Method
- DeepFool Method
- Carlini and Wagner Method



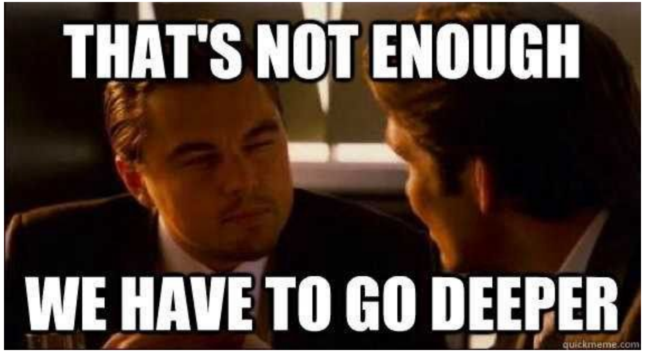
Dataset

We have used German Traffic Sign Recognition Benchmark Dataset (GTSRB)

- Single-image, multi-class classification problem
- 43 classes, 32*32 img size
- Approx 40,000 images for training
- We used 6535 images after preprocessing/cleaning the data



Samples from the dataset



2.

Literature Review

Related Work

Necessity

Attacks

Defense

SNo.	Paper Title	Work Done	Limitations
1	Deep Learning for Large-Scale Traffic-Sign Detection and Recognition [3]	Deep learning method for the detection of traffic signs with large-intra-category appearance variation	Losses by classification network. Still room for improvement.
2	A Hierarchical Deep Architecture and Mini-Batch Selection Method For Joint Traffic Sign and Light Detection [4]	The deep hierarchical architecture that allows a network to detect both traffic lights and signs from training on the separate traffic light and sign datasets	Unable to hit a good accuracy with some network.
3	Traffic Sign Detection under Challenging Conditions: A Deeper Look Into Performance Variations and Spectral Characteristics [5]	A Deeper Look Into Performance Variations and Spectral Characteristics	

SNo.	Paper Title	Work Done	Limitations
4	Fooling a Real Car with Adversarial Traffic Signs [6]	How to utilize adversarial attacks to attack real-life systems in the physical world.	Sometimes shows unexpected behaviours.
5	Rogue Signs: Deceiving Traffic Sign Recognition with Malicious Ads and Logos [7]	Generation of adversarial samples which are robust to the environmental conditions and noisy image transformations .	Defensive Measures, not trained a robust model on adversarial images
6	DARTS: Deceiving Autonomous Cars with Toxic Signs [8]	Out-of-Distribution attacks, Lenticular Printing attack	Defensive Measures, not trained a robust model on adversarial images
7	Building Robust Deep Neural Networks for Road Sign Detection (**SOTA**) [9]	Complete model including attacks and defensive approaches, used defensive distillation with 91.46% accuracy.	The test accuracy cannot reach the original test accuracy on non-adversarial samples.

SNo.	Paper Title	Work Done	Limitations
8	Defending against Adversarial Images using Basis Functions Transformations [10]	Experiment with low-pass filtering, PCA, JPEG compression, low resolution wavelet approximation, and soft-thresholding.	
9	Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples [11]	JPEG-based defensive compression framework	Further Improvements are possible.
10	Local Gradients Smoothing: Defense against localized adversarial attacks [12]	Focuses on frequency changes in the attacked images. Further adoption of Local Gradients Smoothing (LGS) scheme.	More effective for localized adversarial attacks
11	Image Super-Resolution as a Defense Against Adversarial Attacks [13]	Proposes a computationally efficient image enhancement approach that provides a strong defense mechanism	Complex model

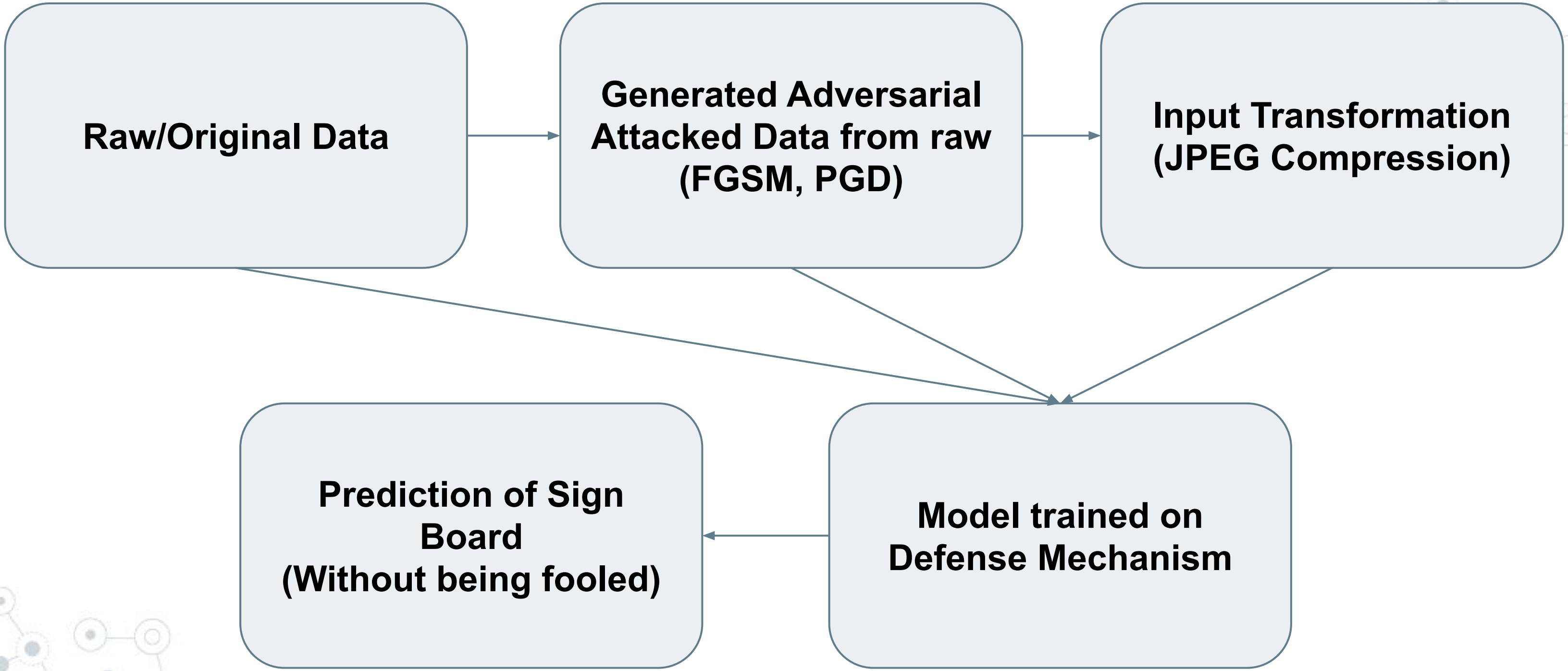
3.

Methodology

Novel Approach/ Improvements

- Idea is to build an end to end pipeline which will correctly identify the attacked images as well.
- We propose a workaround, by building a model which is robust to adversarial attacks with the help of **JPEG compression as defense mechanism**.
- Improved accuracy w.r.t SOTA [9]

Pipeline


Raw/Original Data
Generated Adversarial Attacked Data from raw (FGSM, PGD)
Input Transformation (JPEG Compression)
Model trained on Defense Mechanism
Prediction of Sign Board (Without being fooled)

4.

Work Done

Generated Attacked Images

We generated Adversarially Attacked Images with FGSM and PGD attack methods with changing epsilon (Ep) values.

Ep = 0.01

Ep = 0.05

Ep = 0.1

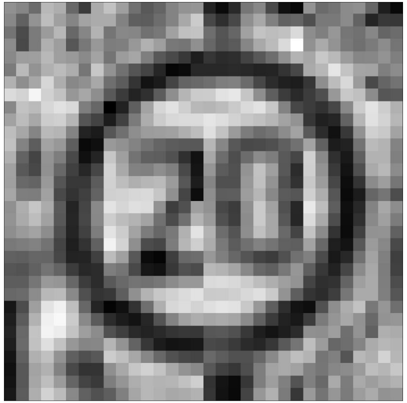
Calculated Accuracy on Attacked Images

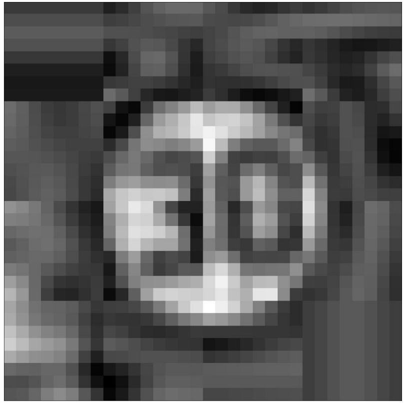
Accuracy on adversarial samples tested with ordinary ML Model (trained only on raw images):

- Ep = 0.01 → Acc =**13.64%**
- Ep = 0.05 → Acc =**10.15%**
- Ep = 0.1 → Acc =**7.39%**

Applied JPEG Compression on Attacked Images

JPEG compression helps to minimize the attack success rate while further improving the defense efficiency.





Trained a Robust Model

Trained a new model on adversarially generated images and jpeg compressed images.

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 32, 32, 3)]	0	
conv2d_40 (Conv2D)	(None, 32, 32, 16)	448	input_4[0][0]
batch_normalization_31 (BatchNo	(None, 32, 32, 16)	64	conv2d_40[0][0]
activation_31 (Activation)	(None, 32, 32, 16)	0	batch_normalization_31[0][0]
conv2d_41 (Conv2D)	(None, 32, 32, 16)	272	activation_31[0][0]
batch_normalization_32 (BatchNo	(None, 32, 32, 16)	64	conv2d_41[0][0]
activation_32 (Activation)	(None, 32, 32, 16)	0	batch_normalization_32[0][0]
conv2d_42 (Conv2D)	(None, 32, 32, 16)	2320	activation_32[0][0]
batch_normalization_33 (BatchNo	(None, 32, 32, 16)	64	conv2d_42[0][0]
activation_33 (Activation)	(None, 32, 32, 16)	0	batch_normalization_33[0][0]
conv2d_44 (Conv2D)	(None, 32, 32, 64)	1088	activation_31[0][0]
conv2d_43 (Conv2D)	(None, 32, 32, 64)	1088	activation_33[0][0]
add_10 (Add)	(None, 32, 32, 64)	0	conv2d_44[0][0] conv2d_43[0][0]
batch_normalization_34 (BatchNo	(None, 32, 32, 64)	256	add_10[0][0]
activation_34 (Activation)	(None, 32, 32, 64)	0	batch_normalization_34[0][0]
conv2d_45 (Conv2D)	(None, 16, 16, 64)	4160	activation_34[0][0]
batch_normalization_35 (BatchNo	(None, 16, 16, 64)	256	conv2d_45[0][0]
activation_37 (Activation)	(None, 16, 16, 128)	0	batch_normalization_37[0][0]
conv2d_49 (Conv2D)	(None, 8, 8, 128)	16512	activation_37[0][0]
batch_normalization_38 (BatchNo	(None, 8, 8, 128)	512	conv2d_49[0][0]
activation_38 (Activation)	(None, 8, 8, 128)	0	batch_normalization_38[0][0]
conv2d_50 (Conv2D)	(None, 8, 8, 128)	147584	activation_38[0][0]
batch_normalization_39 (BatchNo	(None, 8, 8, 128)	512	conv2d_50[0][0]
activation_39 (Activation)	(None, 8, 8, 128)	0	batch_normalization_39[0][0]
conv2d_52 (Conv2D)	(None, 8, 8, 256)	33024	add_11[0][0]
conv2d_51 (Conv2D)	(None, 8, 8, 256)	33024	activation_39[0][0]
add_12 (Add)	(None, 8, 8, 256)	0	conv2d_52[0][0] conv2d_51[0][0]
batch_normalization_40 (BatchNo	(None, 8, 8, 256)	1024	add_12[0][0]
activation_40 (Activation)	(None, 8, 8, 256)	0	batch_normalization_40[0][0]
average_pooling2d_4 (AveragePoo	(None, 1, 1, 256)	0	activation_40[0][0]
flatten_4 (Flatten)	(None, 256)	0	average_pooling2d_4[0][0]
dense_4 (Dense)	(None, 43)	11051	flatten_4[0][0]

Total params: 307,659
Trainable params: 305,899
Non-trainable params: 1,760

Results

Legit Samples - Raw data, Adv samples - Adversarial data

	Without defense mechanism	
	On legit samples	On Adv samples
Testing Accuracy	98.765%	10.39%
F1 Score	0.98765	0.1039

	With defense mechanism	
	On legit samples	On Adv samples
Testing Accuracy	98.299%	93.56%
F1 score	0.98299	0.9356

5.

Deployment

Technology Stack

Frontend

Used tensorflow.js and simple js, html, css over the Skeleton framework to create a frontend.

It also helps in keeping the frontend code simple and implement states to provide the dynamic nature.

We loaded our model in tensorflow.js which predicts the traffic sign board images (normal as well as attacked) into 43 different categories.

Conclusion and Future Work

- Knowledge of various factors that can badly affect automated vehicles importantly Adversarials.
- Attempt to create a buckler to avoid adversarial attacks
- Employ Multiple defensive or use of augmented defensive layers
- Preprocessing like smoothing including other techniques can be adopted
- Extension of work for digital sign boards as well.
- **We have communicated our research paper to ICPS-2021 conference and soon it will be published.**

References

[1] Shan, S., Wenger, E., Wang, B., Li, B., Zheng, H., & Zhao, B. Y. (2020, October). Gotta Catch'Em All: Using Honeypots to Catch Adversarial Attacks on Neural Networks. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 67-83).

[2] Tan, M., Te, K., & Lai, N. (2019). CS 229: Milestone Learning Adversarially Robust and Rich Image Transformations for Object Classification.

[3] Tabernik, D., & Skočaj, D. (2019). Deep learning for large-scale traffic-sign detection and recognition. IEEE transactions on intelligent transportation systems, 21(4), 1427-1440.

[4] Pon, A., Adrienko, O., Harakeh, A., & Waslander, S. L. (2018, May). A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. In 2018 15th Conference on Computer and Robot Vision (CRV) (pp. 102-109). IEEE.

[5] Temel, D., Chen, M. H., & AlRegib, G. (2019). Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3663-3673.

[6] Morgulis, N., Kreines, A., Mendelowitz, S., & Weisglass, Y. (2019). Fooling a real car with adversarial traffic signs. arXiv preprint arXiv:1907.00374.

References

[7] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Mittal, P., & Chiang, M. (2018). Rogue signs: Deceiving traffic sign recognition with malicious ads and logos. arXiv preprint arXiv:1801.02780.

[8] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., & Mittal, P. (2018). Darts: Deceiving autonomous cars with toxic signs. arXiv preprint arXiv:1802.06430.

[9] Aung, A. M., Fadila, Y., Gondokaryono, R., & Gonzalez, L. (2017). Building robust deep neural networks for road sign detection. arXiv preprint arXiv:1712.09327.

[10] Shaham, U., Garritano, J., Yamada, Y., Weinberger, E., Cloninger, A., Cheng, X., ... & Kluger, Y. (2018). Defending against adversarial images using basis functions transformations. arXiv preprint arXiv:1803.10840.

[11] Liu, Z., Liu, Q., Liu, T., Xu, N., Lin, X., Wang, Y., & Wen, W. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In 2019 IEEE. In CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 860-868).

[12] Naseer, M., Khan, S., & Porikli, F. (2019, January). Local gradients smoothing: Defense against localized adversarial attacks. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1300-1307). IEEE.

Thanks!

Any questions?