Enhancing Security in Autonomous Vehicles:

Adversarial Attack Mitigation for Image-Based Neural Networks

Javaad Akhtar & Rishabh Jain

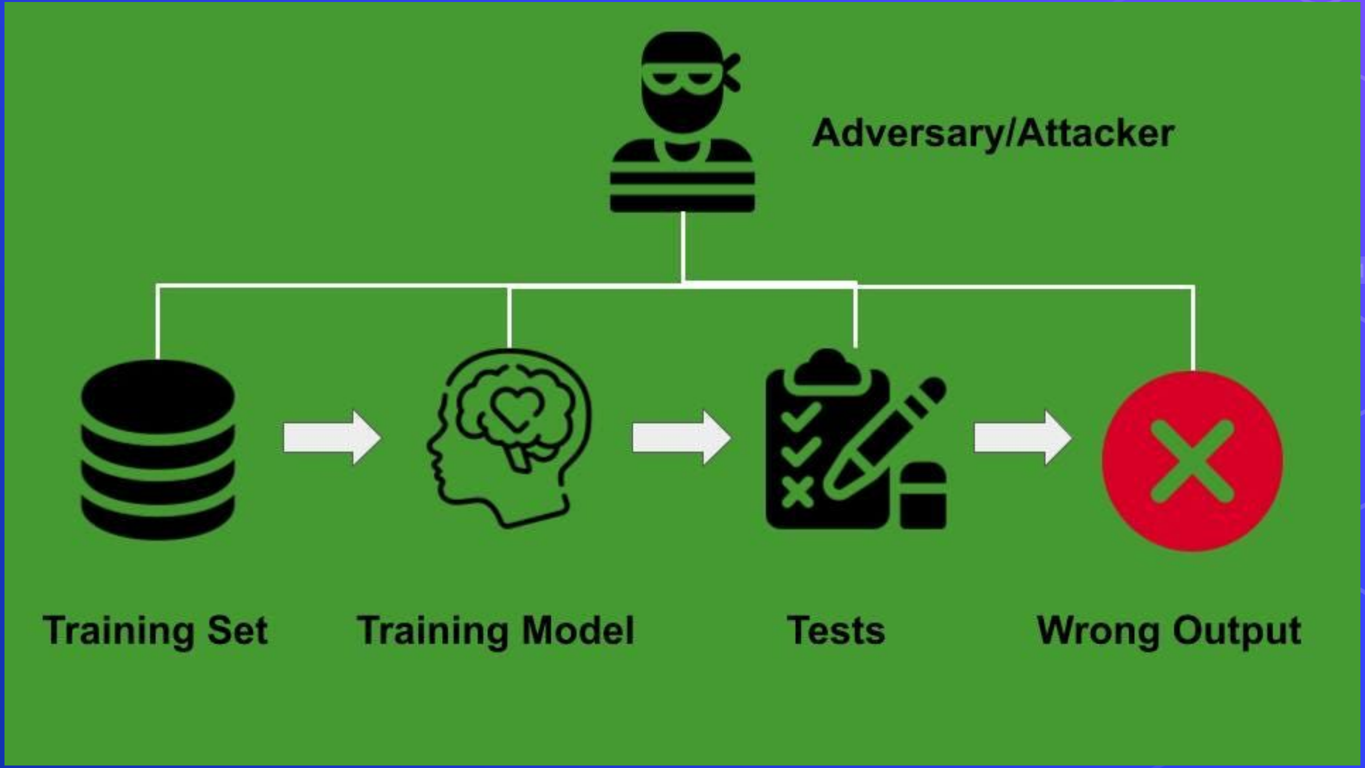
Key Words

- **Adversarial** - Techniques intentionally designed to manipulate models
- **MNIST** - Dataset of grayscale images of handwritten digits (0-9)
- **OCR** (Optical Character Recognition) - Interprets text from images/documents
- **Perturbation** - Small change applied to deceive machine learning models
- **Noise** - Unwanted or random disturbances in data affecting model accuracy
- **Autonomous Vehicle** - Camera reliant self-driving cars

Introduction to Adversarial Attacks

- Manipulate and deceive machine learning models
  - Especially neural networks.
- Exploiting vulnerabilities in models with input data
  - Known as adversarial examples.
- Model makes incorrect predictions or classifications through subtle manipulations.
  - Complex and nonlinear decision boundaries learned by neural networks.

Adversarial Attacks



Types of Adversarial Attacks

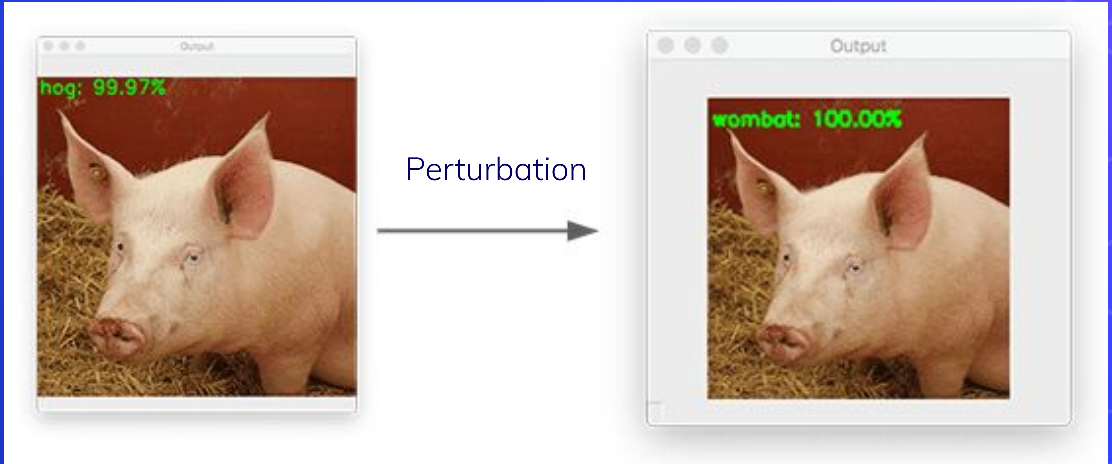
Targeted Attack

- Attacker intentionally tries to misclassify data
- Misguiding the model to a **particular class** as opposed to true class
- *f(x + e) = incorrect class*
  - *x: original image*
  - *e: perturbation*

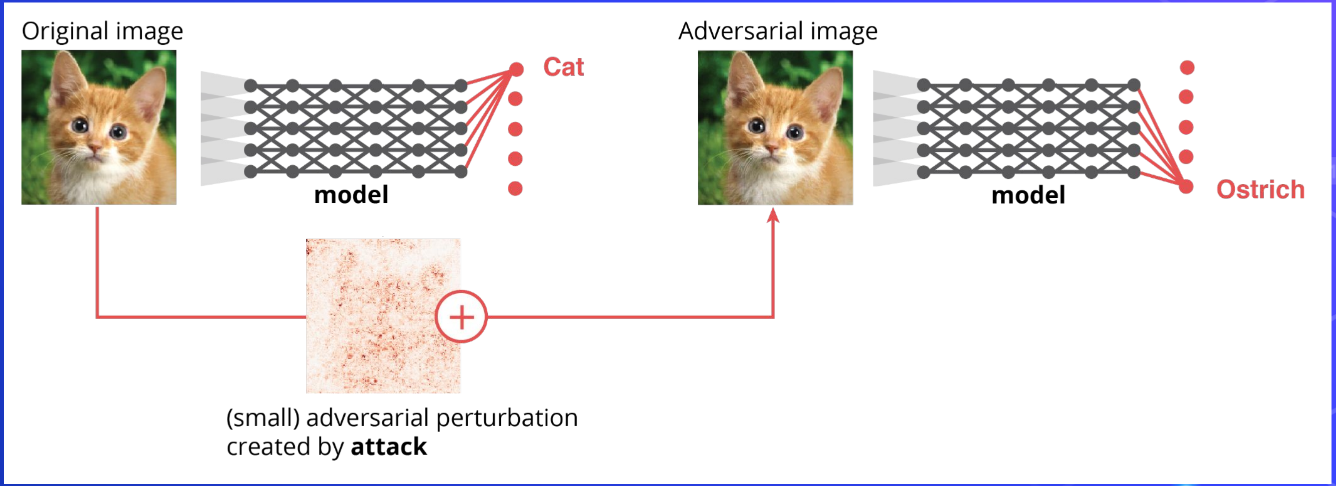
Non-Targeted Attack

- Attacker just wants to misclassify to an incorrect class
- *f(x + e) = correct class*
  - *x: original image*
  - *e: perturbation*

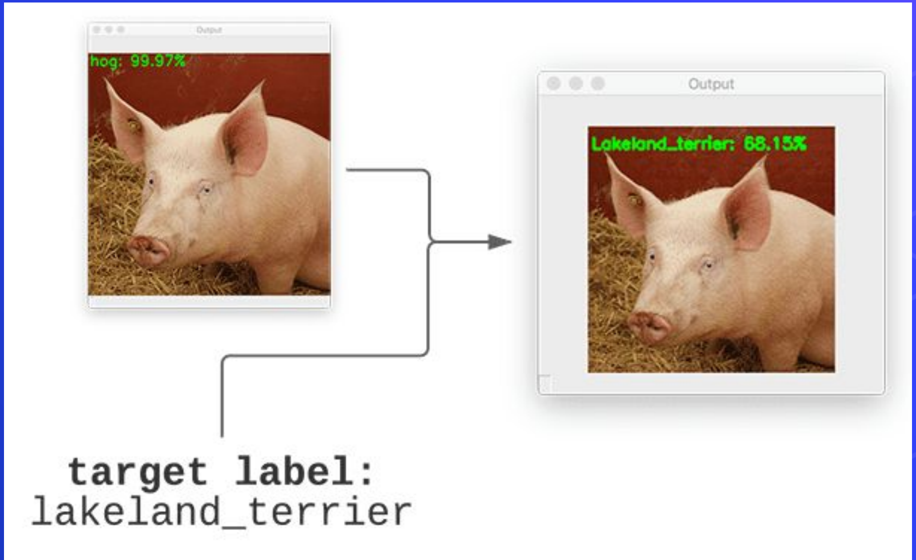
Non-Targeted Attack



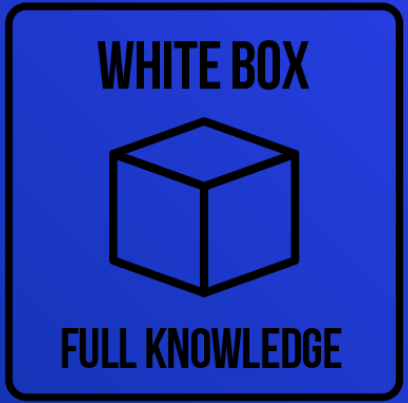
Non-Targeted Attack

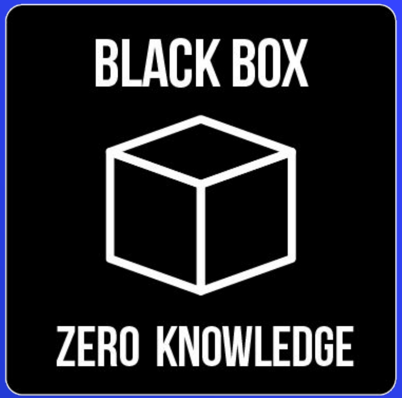


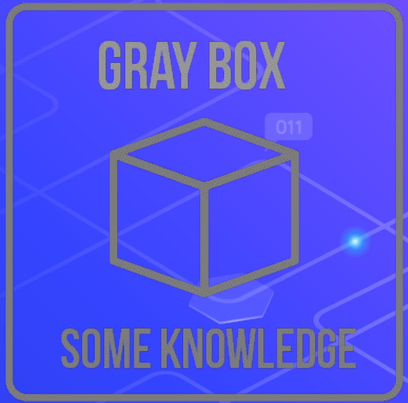
Targeted Attack



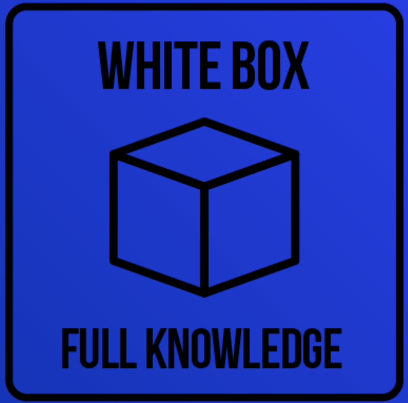
Types of Adversarial Attacks







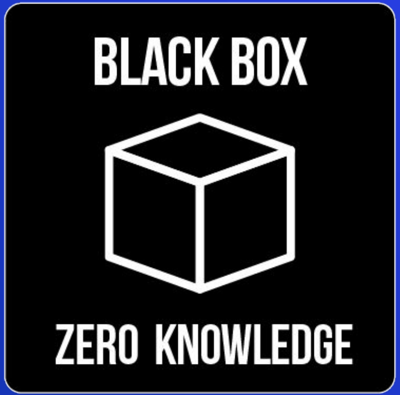
White Box Attack



White-Box Attacks:

- Completely aware of target model
  - Network architecture
  - Weight
  - Biases

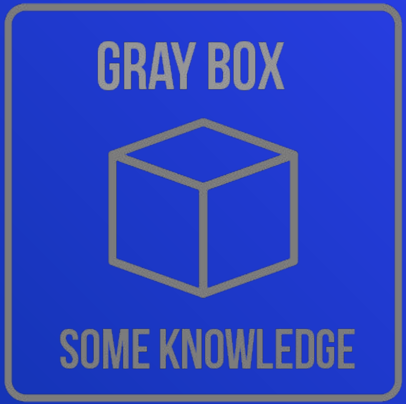
Black Box Attack



Black-Box Attacks:

- Unaware of internal architecture of neural network
- No information on weight or biases
- Attacker resorts to creating adversarial samples to use on the network

Gray Box Attack



Gray-Box Attacks:

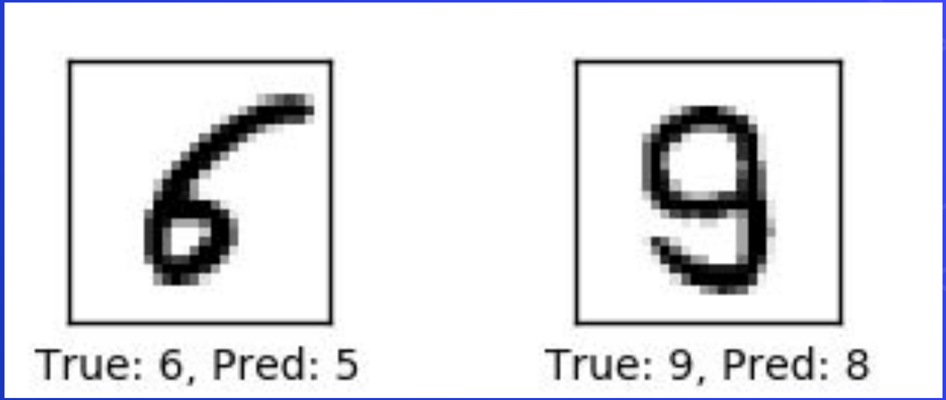
- Limited knowledge of network and parameters
- Used to create targeted adversarial samples
- Ex. Just knowledge of weights

Potential Real-Life Problems

Impact on MNIST data & OCR Systems

OCR (Optical Character Recognition)

Adversarial attacks may lead to misinterpretation of characters in images or documents.

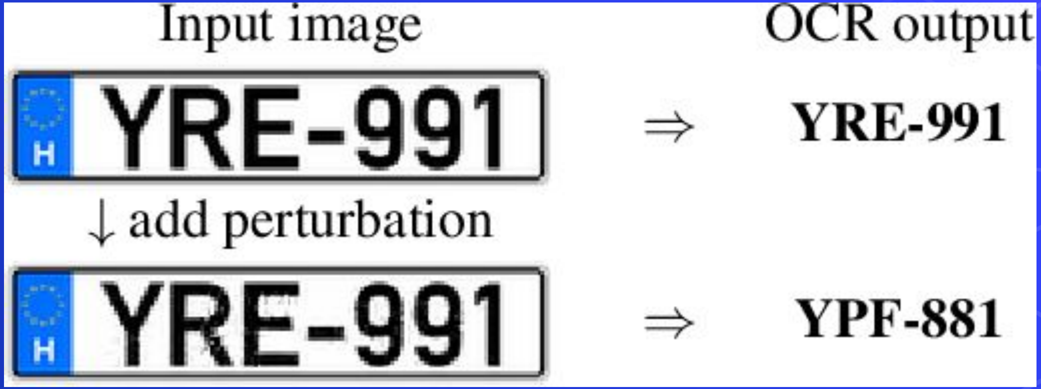


What’s the BIG deal?

Think about...

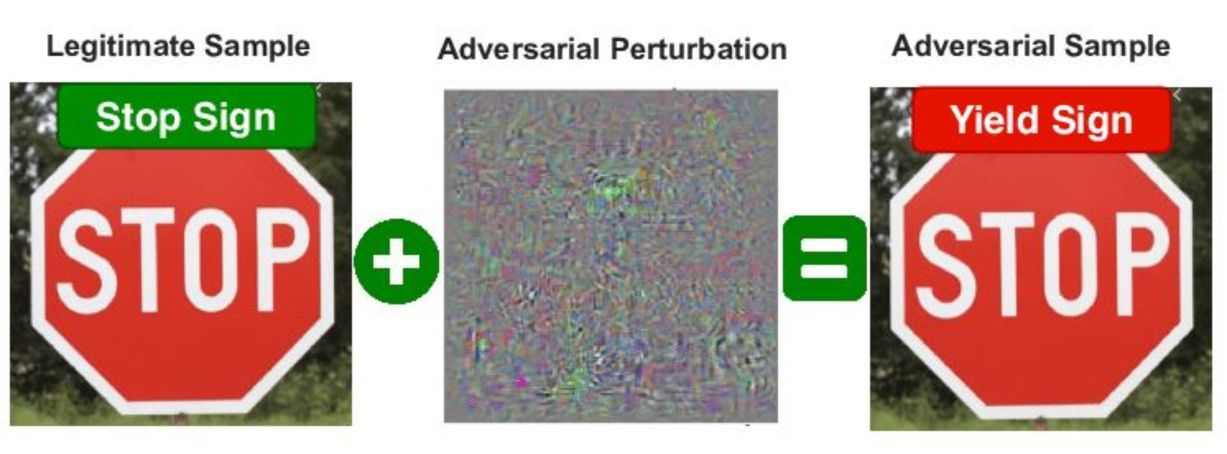
License plates

- Speeding fines
- Parking tickets



What about...

Self driving cars



Autonomous Vehicles...

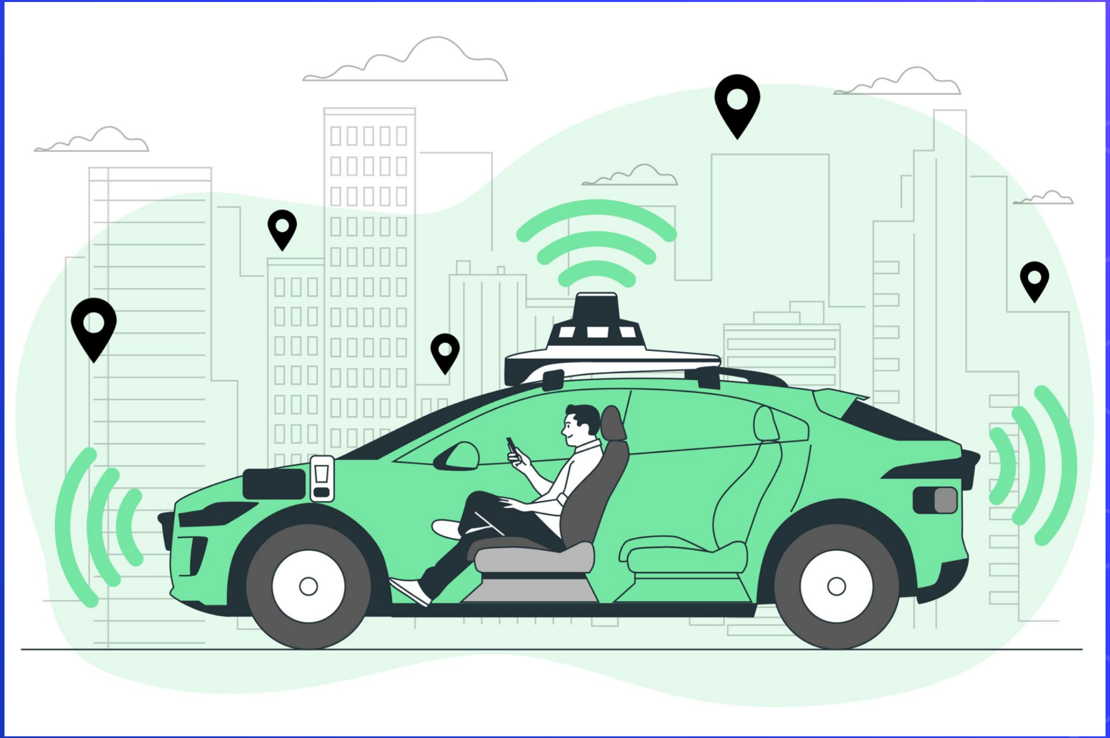
Lidar

GPS

Cameras

Radar

Ultrasonic



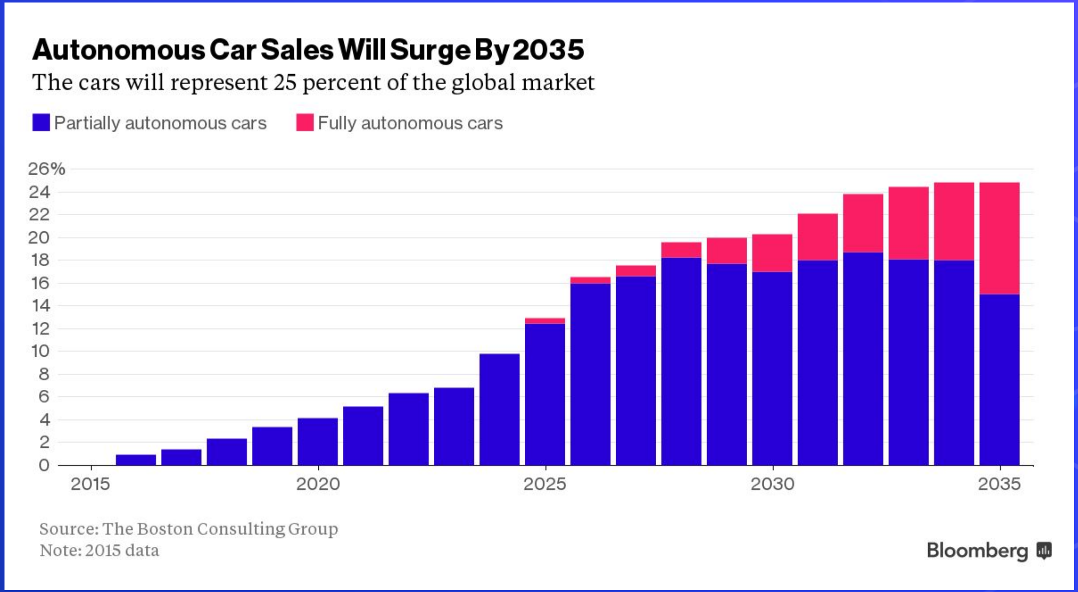
> 30 Million

Self driving cars on the road…

Consider the impact this could have

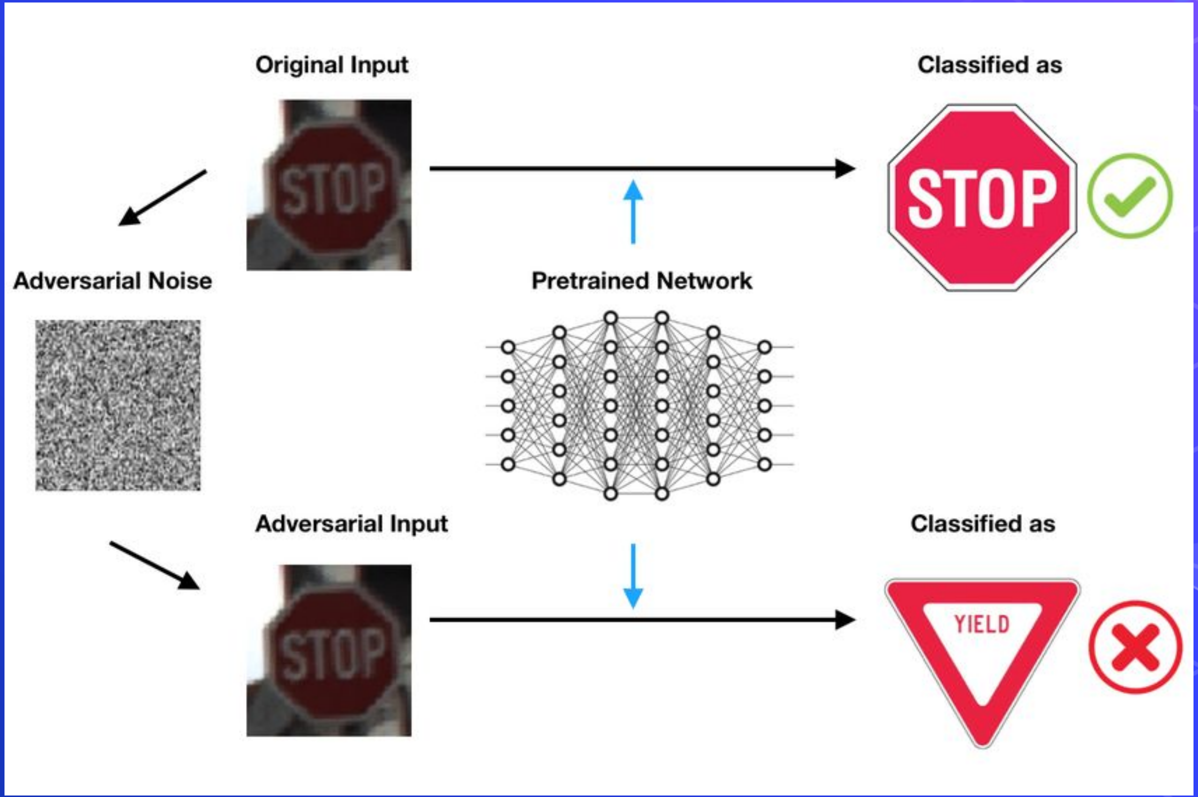
Relevance

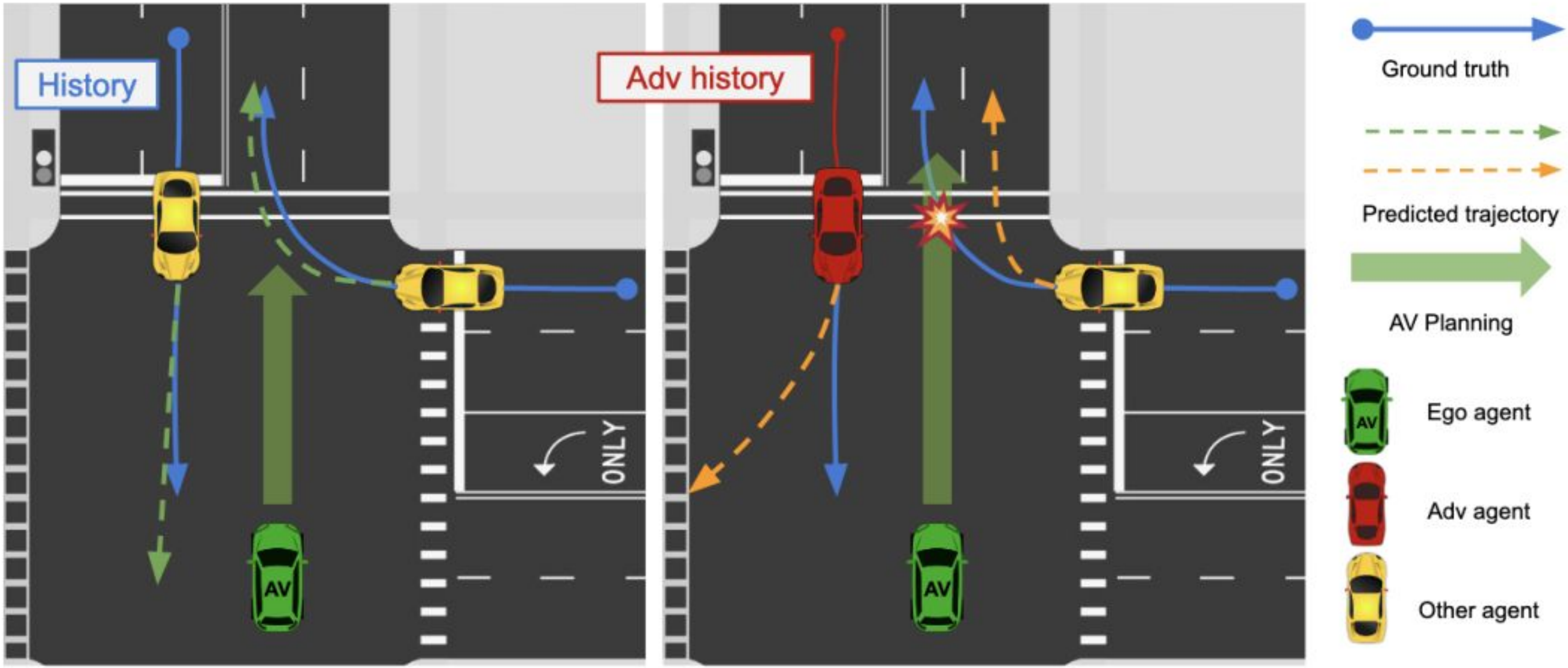
- Growth of autonomous vehicle technology
- Reliance on image-based neural networks



What can happen?

- Misreading text-based signs
  - Stop signs
  - Speed limits
- Distinguishing between the lines of different lanes
- Underestimating distance between other vehicles
- Etc.


Original Input
Classified as
STOP
Adversarial Noise
Pretrained Network
Adversarial Input
STOP
Classified as
YIELD



Without attack   Under attack

Adversarial Attacks with regards to autonomous driving

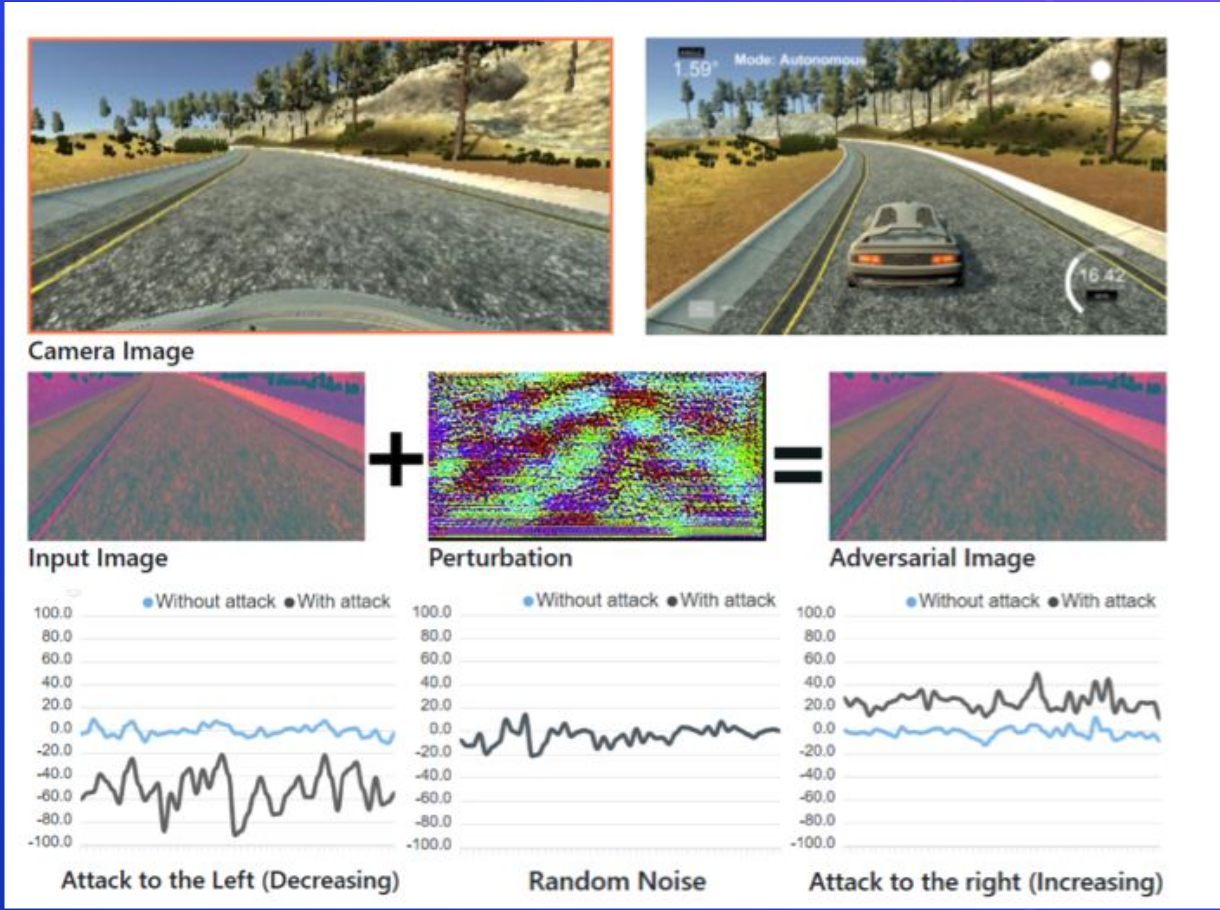
Research

Paper: Adversarial Driving: Attacking End-to-End Autonomous Driving

Details:

- Examination of vulnerabilities within neural network models designed for image processing in autonomous vehicles
- Analysis of white-box targeted attacks against advanced autonomous driving model
- Exploration of potential manipulations in autonomous driving models leading to oversteer or understeer during vehicular turning maneuver.

Effect of Attack



Architecture

Neural Network

Layer	Output Shape	Parameters
Input	(None, 160, 320, 3)	0
Conv2D	(None, 78, 158, 24)	1824
Conv2D	(None, 37, 77, 36)	21636
Conv2D	(None, 17, 37, 48)	43248
Conv2D	(None, 15, 35, 64)	27712
Conv2D	(None, 13, 13, 24)	36928
Dropout	(None, 13, 13, 24)	0
Flatten	(None, 27456)	0
Dense	(None, 100)	2745700
Dense	(None, 50)	5050
Dense	(None, 10)	510
Dense	(None, 1)	11

Attack example

Input: The regression model $f(\theta, x)$, input images in a driving record X, the target direction $I \in \{-1, 1\}$.
Parameters: the number of iterations n, the learning rate α, the step size ξ , and the strength of the attack ϵ measured by the l_∞ norm.
Output: Image-agnostic perturbation η.
Initialization: $\eta \leftarrow 0$
for each iteration **do**
  for each input image x in the driving record X **do**
    Inference: $y = f(\theta, x + \eta)$
    if sign$(y) \neq I$ **then**
      $x' = x + \eta$
      $\eta_t \leftarrow 0$
      while sign$(y) \neq I$ **do**
        Gradients: $\nabla = \frac{\partial J(y)}{\partial x'}$
        Perturbation: $\eta_t = \eta_t + proj_2(\nabla,\ \xi)$
        Inference: $y = f(\theta, x + \eta_t)$
      end while
      $\eta = proj_\infty(\eta + \frac{\alpha}{\xi}\eta_t,\ \epsilon)$
    end if
  end for
end for

Demo 1

Our Model

Details:

- Added policies for mitigation for adversarial attacks
- Pre-processing: Incorporated data augmentation techniques to enhance the pre-processing phase
- Model and Training: Implement the addition of random gaussian noise subsequent to each convolutional step within the neural network model. This aims to robustify the model against the overfitting and improve generalization
- Post-processing: Introducing course correction algorithm for prior to transmitting steering information to the car

Data Set

- In-House Dataset Creation:
  - Utilized the advanced capabilities of the Udemy driving simulator to generate a custom dataset. This tool enables users to record their driving sessions, thereby facilitating the creation of a unique and comprehensive dataset tailored for autonomous driving research.

- Dataset Integration:
  - Successfully merged our proprietary dataset with the one referenced in the research paper. This integration was executed to enrich the training data, enhancing the robustness and diversity of the dataset employed for our autonomous driving model development.

Pre-processing (Data Augmentation):

- Random Resizing
- Random Padding

Architecture

Ours

```
Layer (type)                 Output Shape              Param #
=================================================================
lambda (Lambda)              (None, 160, 320, 3)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 78, 158, 24)       1824
_________________________________________________________________
gaussian_noise (GaussianNois (None, 78, 158, 24)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 37, 77, 36)        21636
_________________________________________________________________
gaussian_noise_1 (GaussianNo (None, 37, 77, 36)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 17, 37, 48)        43248
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 35, 64)        27712
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 13, 33, 64)        36928
_________________________________________________________________
dropout (Dropout)            (None, 13, 33, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 27456)             0
_________________________________________________________________
dense (Dense)                (None, 100)               2745700
_________________________________________________________________
dense_1 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_2 (Dense)              (None, 10)                510
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 11
=================================================================
Total params: 2,882,619
Trainable params: 2,882,619
```

Paper's

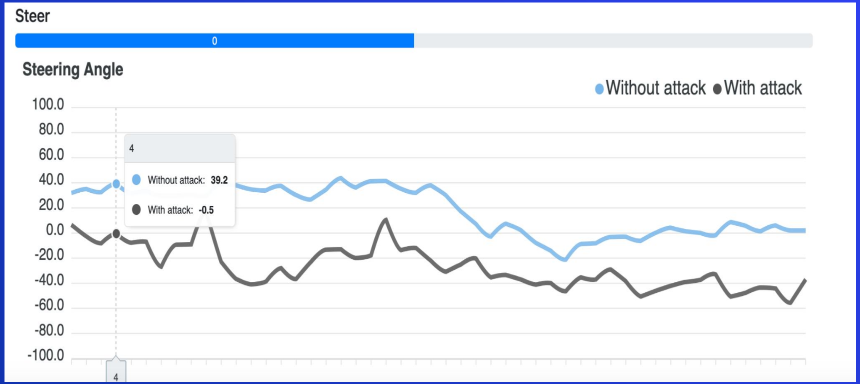
Layer	Output Shape	Parameters
Input	(None, 160, 320, 3)	0
Conv2D	(None, 78, 158, 24)	1824
Conv2D	(None, 37, 77, 36)	21636
Conv2D	(None, 17, 37, 48)	43248
Conv2D	(None, 15, 35, 64)	27712
Conv2D	(None, 13, 13, 24)	36928
Dropout	(None, 13, 13, 24)	0
Flatten	(None, 27456)	0
Dense	(None, 100)	2745700
Dense	(None, 50)	5050
Dense	(None, 10)	510
Dense	(None, 1)	11

Demo 2

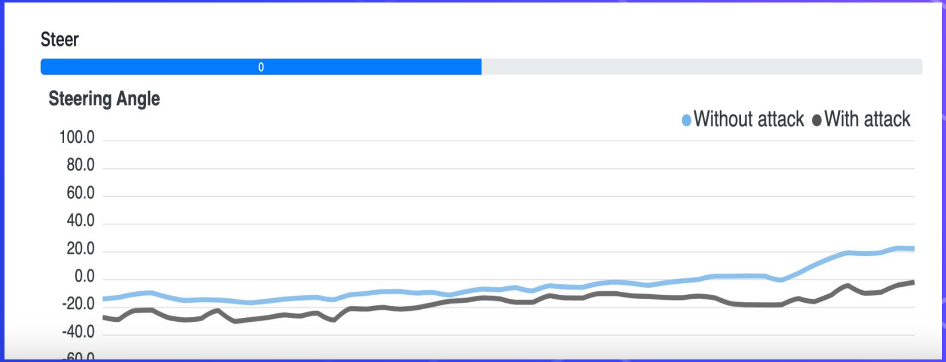
Comparison and Analysis

Steering Angle (Post-Attack)

Old

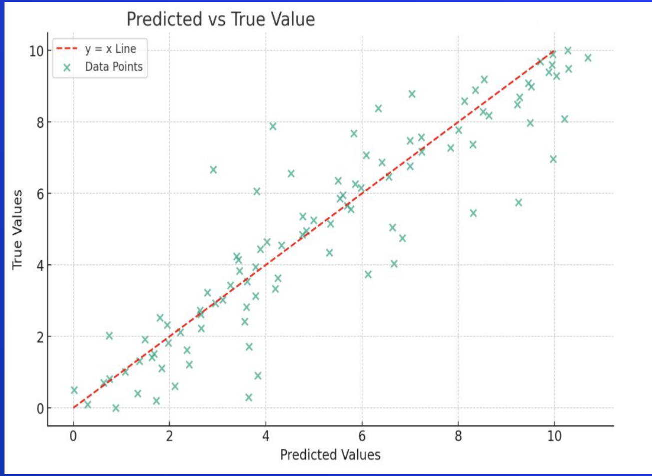


New

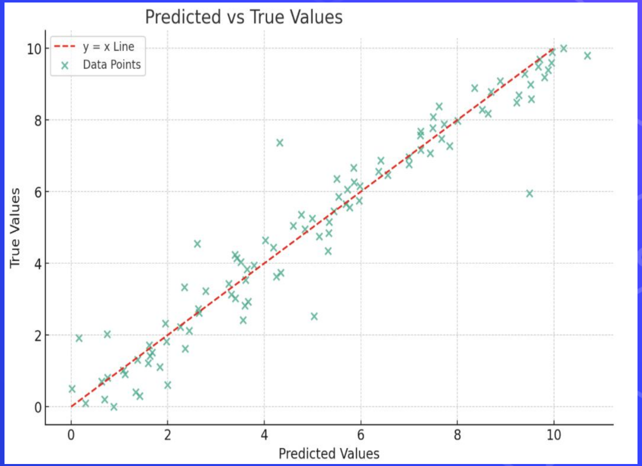


True vs Predicted (Random Perturbation added)

Old



New



Mitigation Policies

- Randomization
- Random Noising
- Denoising

Future Work

- Utilize additional mitigation policies regarding adversarial attack
- Continue improving simulator logic
- More adversarial training for CNN model
- Using ROS to simulate real world application
- Implement a real world test track

Thank you for listening!

Any questions?

References - Research

Singh, M.J, Ramachandra R., (2015) Deep Composite Face Image Attacks: Generation, Vulnerability and Detection. doi: https://arxiv.org/pdf/2211.11039.pdf

Wu H., Yunas S., Rowlands S., Ruan W., (2022) Adversarial Driving: Attacking End-to-End Autonomous Driving. doi: https://arxiv.org/pdf/2103.09151.pdf

Ren K., Zheng T., Qin Z., (2020) Adversarial Attacks and Defenses in Deep Learning.

Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014. arXiv:1412.6572.

Rathore P, Basak A., Nistala S.H., Untargeted, Targeted and Universal Adversarial Attacks and Defenses on Time Series. 2020 DOI - 10.1109/IJCNN48605.2020.9207272

Bran M., Naeh I., Adversarial Attack Against Image-Based Localization Neural Networks

References - Images

ResearchGate, Adversarial Attack Causing Misclassification in MNIST Dataset, https://www.researchgate.net/figure/Adversarial-attack-causing-miss-classification-in-MNIST-data-set_fig1_322950125

Semantic Scholar, Fooling OCR Systems with Adversarial Text Images, https://www.semanticscholar.org/paper/Fooling-OCR-Systems-with-Adversarial-Text-Images-Song-Shmatikov/266a06c96834bfd24dc3cb56c8ef191a0c8b5b7a/figure/0

ResearchGate, Common Adversarial Attack on Image Classification ML Model, https://www.researchgate.net/figure/An-illustration-of-a-common-adversarial-attack-on-image-classification-ML-model-The_fig4_347398615

TechAheadCorp, Future of Self-Driving Cars, https://www.techaheadcorp.com/blog/future-of-self-driving-cars/

ResearchGate, Adversarial Attack on Autonomous Driving for Computing Spectral Norm of DNNs, https://www.researchgate.net/figure/Example-adversarial-attack-on-autonomous-driving-for-computing-the-spectral-norm-of-DNNs_fig1_350397997

NVIDIA Research, Realistic Adversarial Attacks on Trajectory Prediction, https://research.nvidia.com/publication/2022-10_advdo-realistic-adversarial-attacks-trajectory-prediction

Medium, Easy Examples for Black, White, and Gray Box Testings, https://medium.com/@clarkjasonngo/easy-examples-for-black-white-and-gray-box-testings-fdceb2a8b664

Towards Data Science, Adversarial Machine Learning Attacks and Possible Defense Strategies, https://towardsdatascience.com/adversarial-machine-learning-attacks-and-possible-defense-strategies-c00eac0b395a

AIcrowd, NIPS 2018 Adversarial Vision Challenge - Untargeted Attack Track, https://www.aicrowd.com/challenges/nips-2018-adversarial-vision-challenge-untargeted-attack-track

PyImageSearch, Targeted Adversarial Attacks with Keras and TensorFlow, https://pyimagesearch.com/2020/10/26/targeted-adversarial-attacks-with-keras-and-tensorflow/

Forbes, The Most Revolutionary Thing About Self-Driving Cars Isn't What You Think, https://www.forbes.com/sites/worldeconomicforum/2017/06/20/the-most-revolutionary-thing-about-self-driving-cars-isnt-what-you-think/?sh=74558a1b5c57