LNMIIT
The LNM Institute of Information Technology

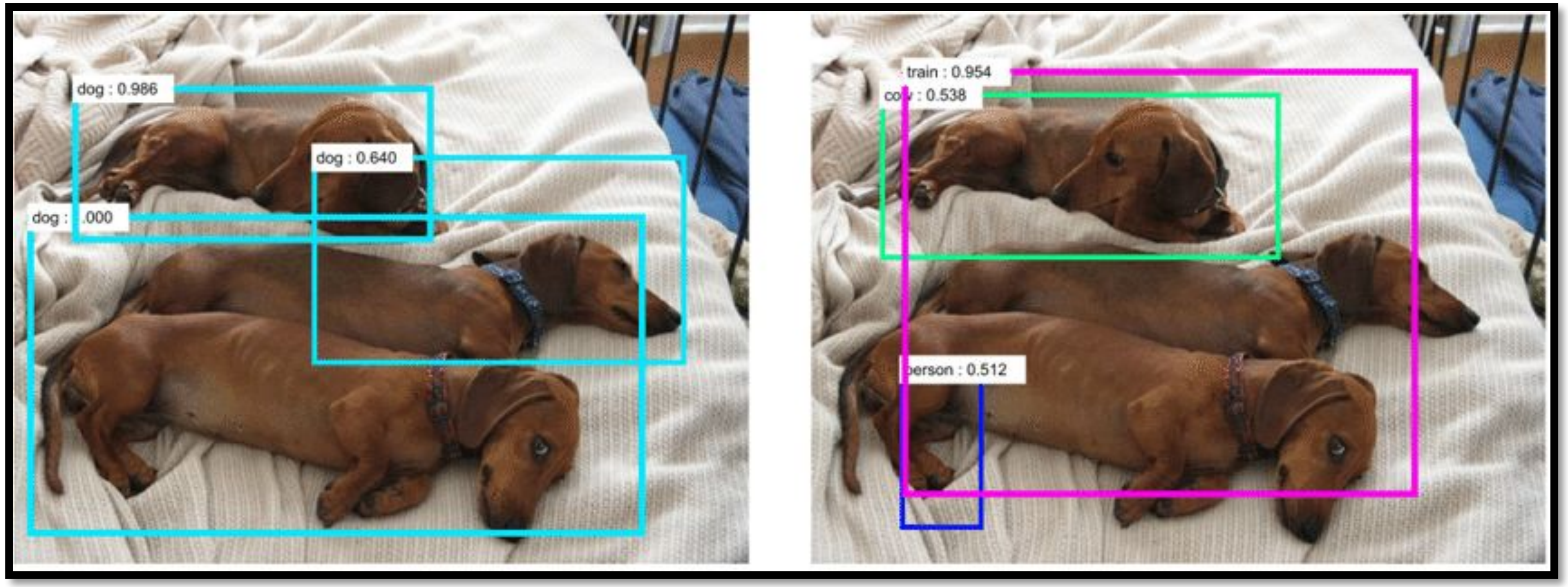
Defense for the Adversarial Attacks on Object detection

Supervisor :Dr. Indra Deep Mastan

- Introduction

Object Detection

- Computer vision technology that involves detecting and localizing objects of interest within an image or video using Deep Neural Network and Machine learning algorithm .



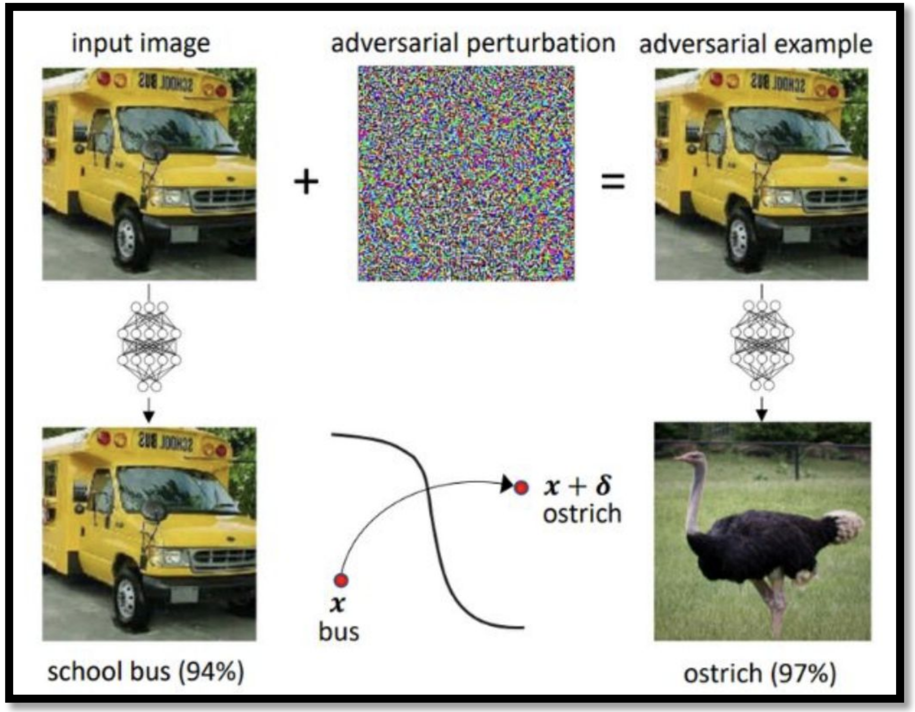
Adversarial Attacks

- Technique to deliberately perturb the input data to deceive machine learning model.
- Aim is to make the model misclassify the input data.

img source : https://www.researchgate.net/figure/An-adversarial-example-for-object-detection-task-22-Left-object-detection-on-a-clean_fig5_323267590

Adversarial Attacks On Object Detection

- Adversarial attacks on object detection involve making subtle and undetectable alterations to the input data to cause the detection algorithm to misidentify objects.
- Types of adversarial attacks:
  - Targeted
  - Untargeted



img source : https://imerit.net/blog/four-defenses-against-adversarial-attacks-all-una/

- Motivation

Motivation

Financial Costs-
A successful adversarial attack on an AI-powered system can have catastrophic financial repercussions for both individuals and businesses.

Safety Risks-
Hostile object detection assaults AI has severe safety implications and the potential to endanger human life in applications like self-driving automobiles.

Privacy Concerns-Face detection system adversarial attacks create severe privacy issues since they can be used to get around security measures and collect personal data.

UCF-Crime Dataset

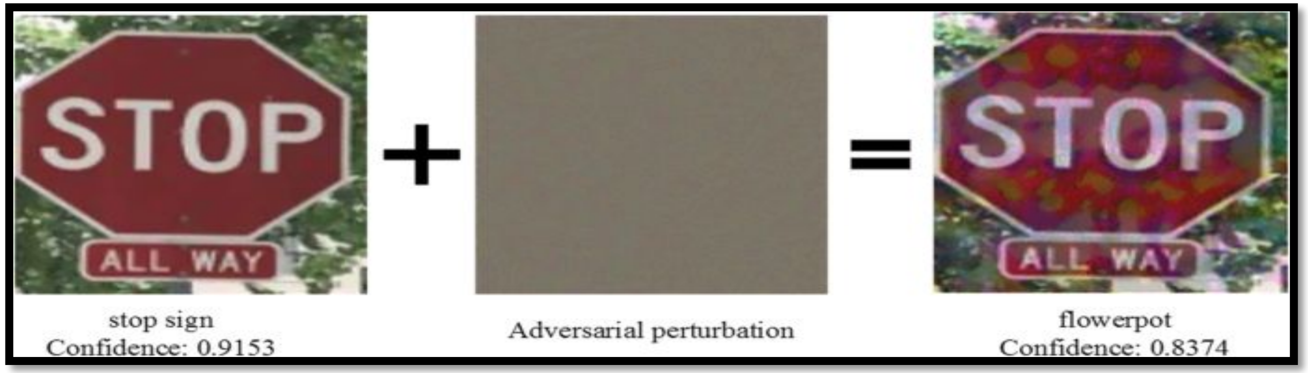
- The UCF-Crime Dataset is a comprehensive database of criminal activity that offers unique insights into the world of law and order.
- We will use the dataset in our project to test robustness of our defense model by detecting the crime after performing adversarial attacks.

https://www.crcv.ucf.edu/projects/real-world/

Fast Gradient Sign Method

What is FGSM?

It is an adversarial attack algorithm used to generate adversarial examples for deep learning models. Idea is to add a small perturbation to the input data to misclassify it.



- In FGSM we add large amount of noise to the input image to perform the adversarial attack.
- By giving the perturbation a value that most nearly resembles the weight vector w, it is possible to maximize the perturbation that results in an inaccurate prediction.

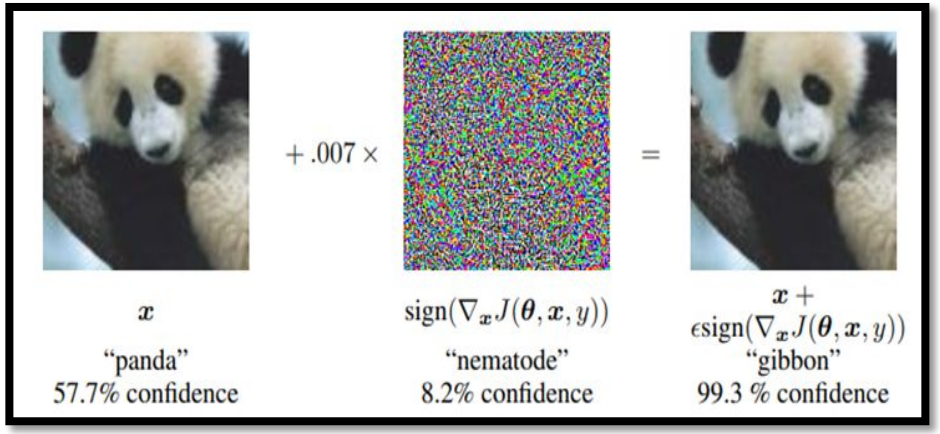
$$w^\top \tilde{x} = w^\top x + w^\top \eta.$$

Explaining And Harnessing Adversarial Examples by Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy , ICLR 2015

How it works?

- Weight vector refers to a vector of numerical values that are used to represent features of an image.
- In an adversarial attack scenario, the value of epsilon (ε) indicates the magnitude of the disturbance added to the input image. It is usually a small value.
- Larger confidence rate = Larger perturbation to fool My Network

$$\boldsymbol{\eta} = \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$



Explaining And Harnessing Adversarial Examples by Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy , ICLR 2015

- DeepFool

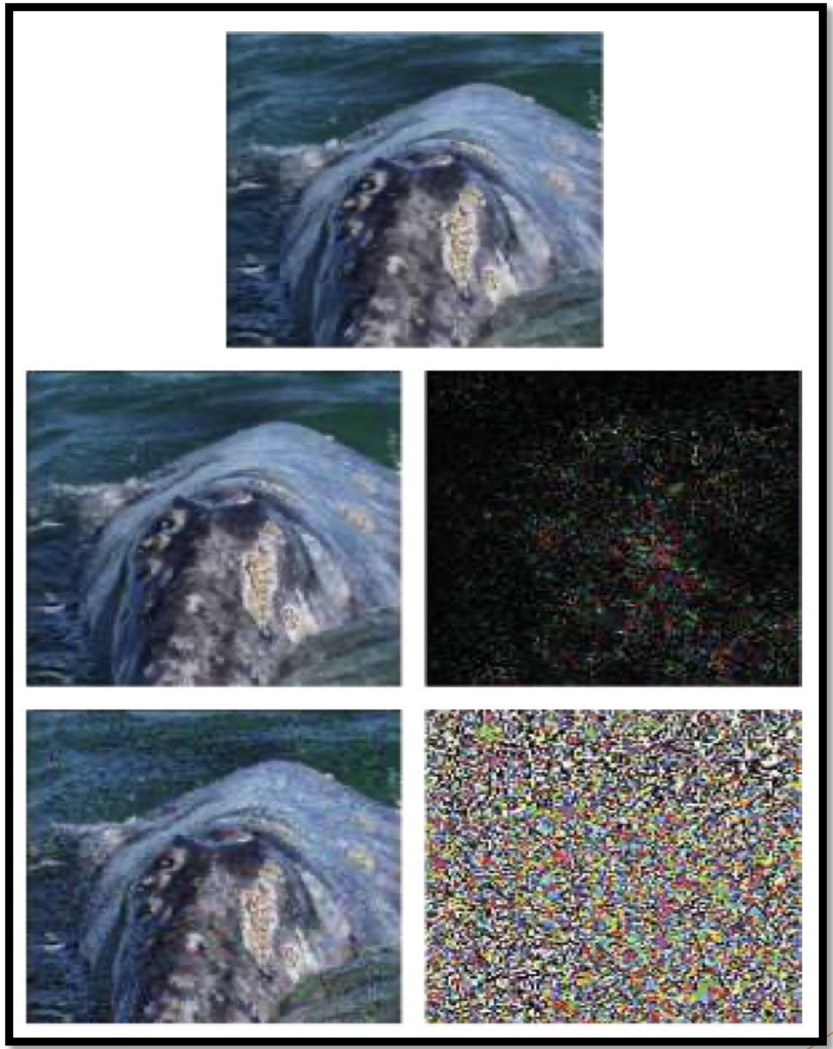
What is DeepFool?

- DeepFool is an algorithm for generating adversarial examples for deep neural networks (DNNs).
- It is simple, computationally efficient, and works across different model architectures.

DeepFool: a simple and accurate method to fool deep neural networks by Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard , CVF_2016

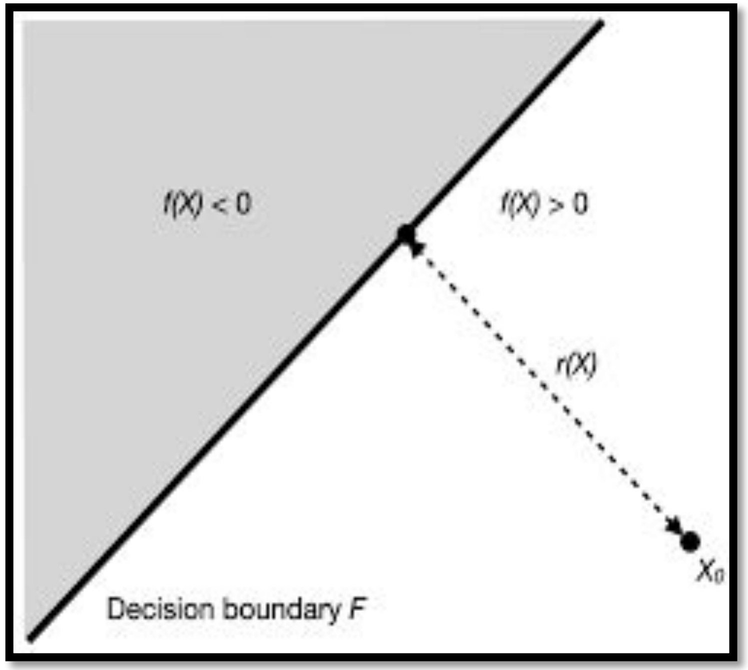
How it works?

- The algorithm iteratively calculates the direction of the decision boundary and finding the smallest perturbation that crosses that boundary.
- The process is repeated until the image is classified as a different class.



DeepFool: a simple and accurate method to fool deep neural networks by Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard , CVF_2016

Deepfool for Binary classifier



$$r_*(x_0) = -\frac{f(x_0)}{\|w\|_2^2} w$$

input: Image x, classifier f.
output: Perturbation $\hat{r}$.
Initialize $x_0 \leftarrow x$, $i \leftarrow 0$.
while $\operatorname{sign}(f(x_i)) = \operatorname{sign}(f(x_0))$ **do**
    $r_i \leftarrow -\frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \nabla f(x_i)$,
    $x_{i+1} \leftarrow x_i + r_i$,
    $i \leftarrow i + 1$.
end while
return $\hat{r} = \sum_i r_i$.

Fig 4. Algorithm to calculate the Adversarial Image for Binary Classifiers.

DeepFool: a simple and accurate method to fool deep neural networks by Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard , CVF_2016

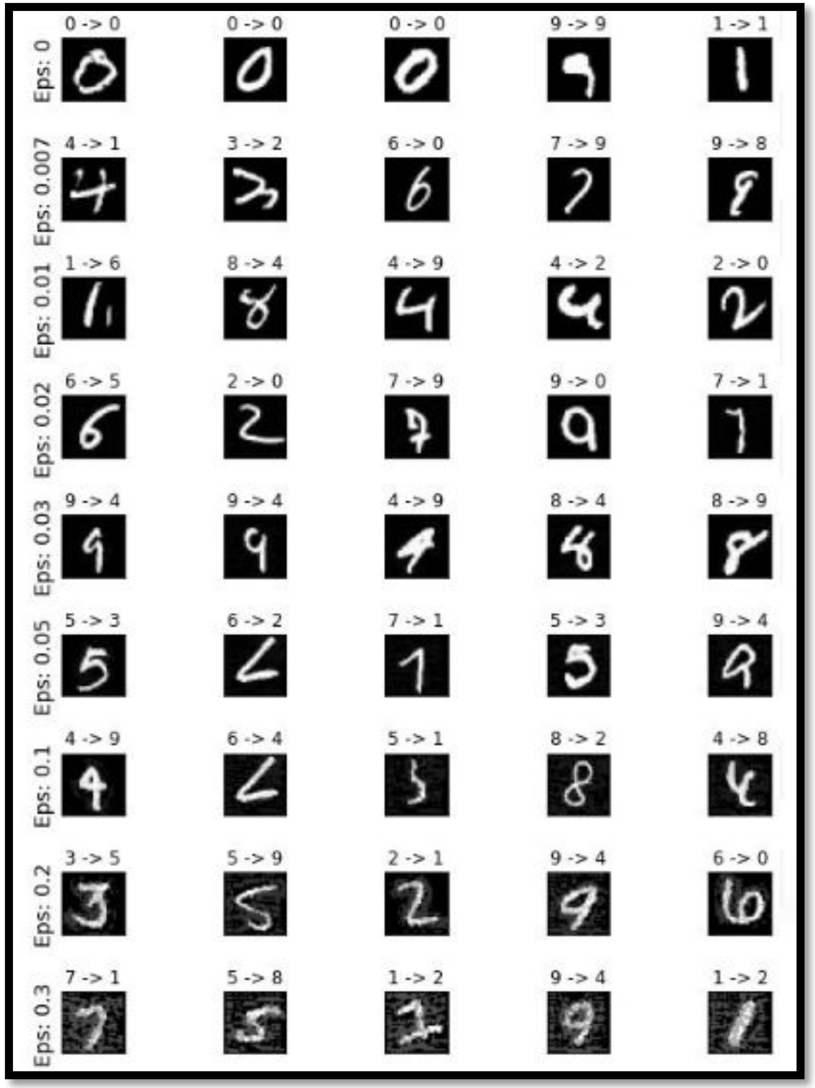
Model: FGSM and DeepFool

FGSM Model: FGSM Function

```
def fgsm_attack(input,epsilon,data_grad):
  pert_out = input + epsilon*data_grad.sign()
  pert_out = torch.clamp(pert_out, 0, 1)
  return pert_out
```

https://colab.research.google.com/github/as791/Adversarial-Example-Attack-and-Defense/blob/master/Adversarial_Example_%28Attack_and_defense%29.ipynb?authuser=3

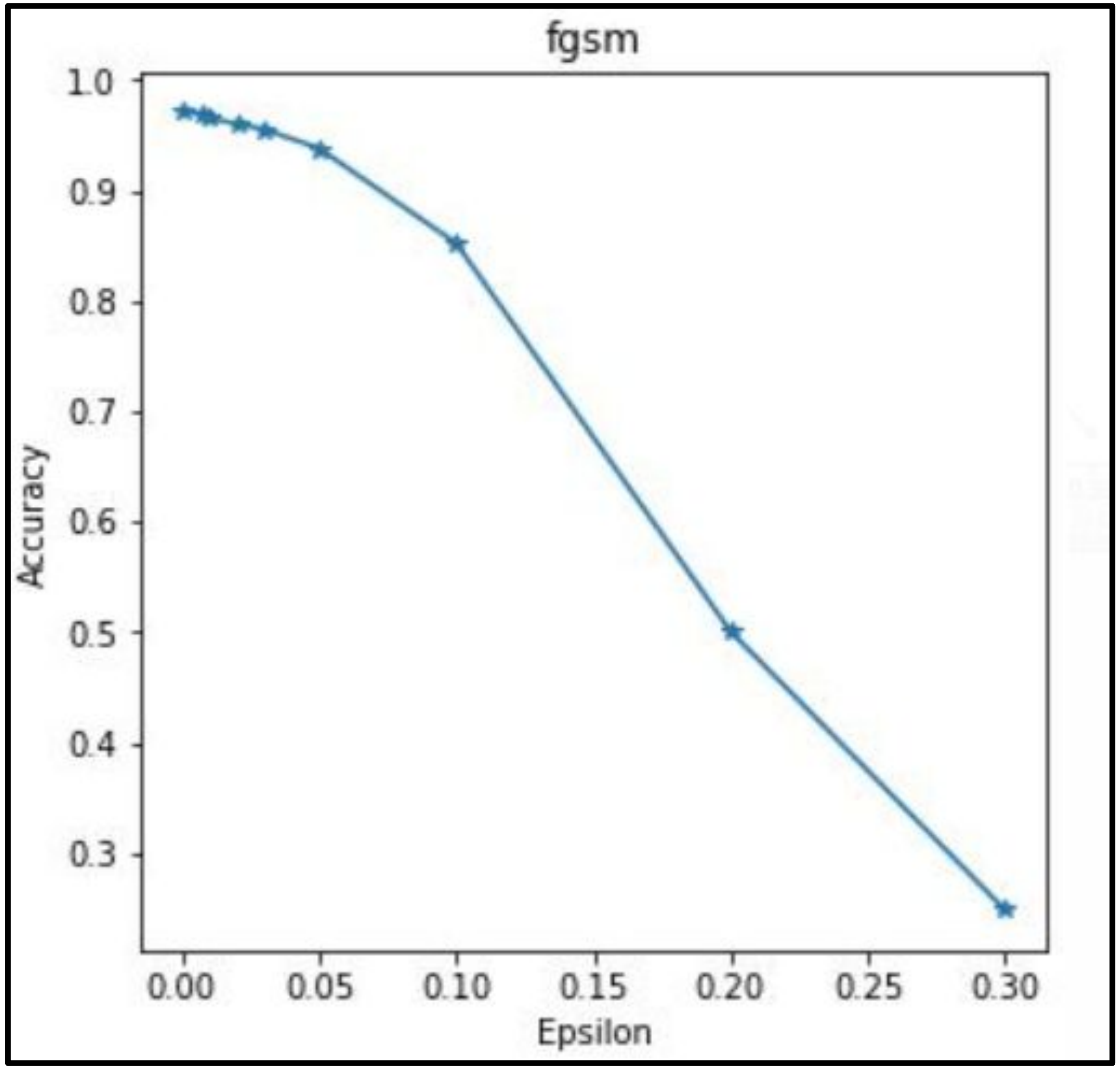
FGSM Model: Results




```
Epsilon: 0      Test Accuracy = 9708 / 10000 = 0.9708
Epsilon: 0.007  Test Accuracy = 9701 / 10000 = 0.9701
Epsilon: 0.01   Test Accuracy = 9650 / 10000 = 0.965
Epsilon: 0.02   Test Accuracy = 9602 / 10000 = 0.9602
Epsilon: 0.03   Test Accuracy = 9552 / 10000 = 0.9552
Epsilon: 0.05   Test Accuracy = 9380 / 10000 = 0.938
Epsilon: 0.1    Test Accuracy = 8520 / 10000 = 0.852
Epsilon: 0.2    Test Accuracy = 5006 / 10000 = 0.5006
Epsilon: 0.3    Test Accuracy = 2484 / 10000 = 0.2484
```

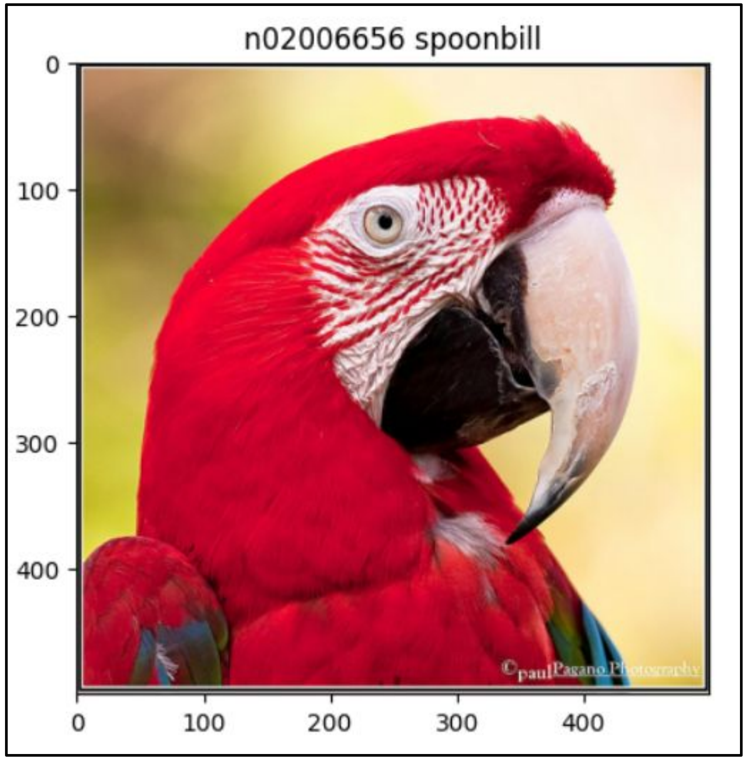

https://colab.research.google.com/github/as791/Adversarial-Example-Attack-and-Defense/blob/master/Adversarial_Example_%28Attack_and_defense%29.ipynb?authuser=3

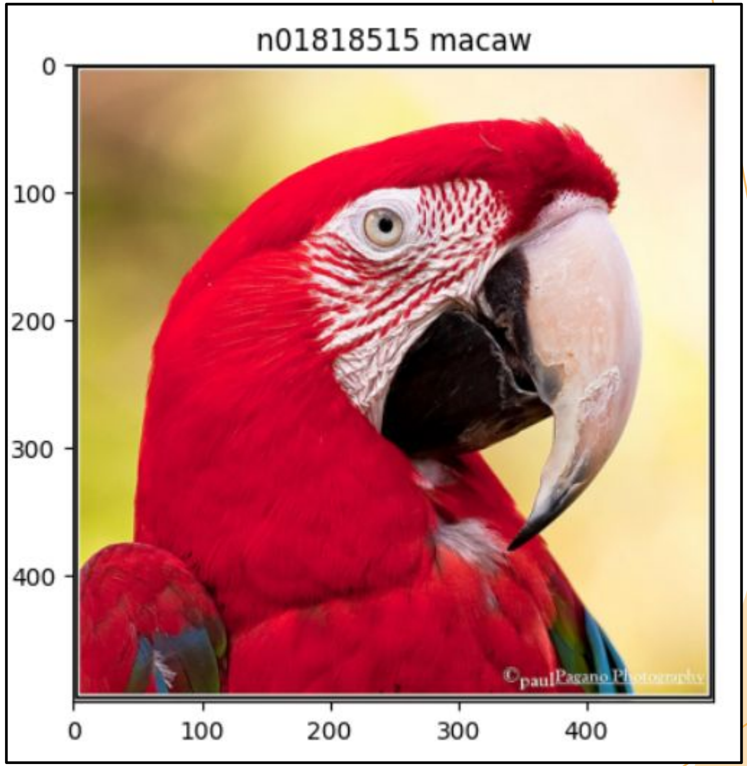
FGSM Model: Accuracy v/s Epsilon



https://colab.research.google.com/github/as791/Adversarial-Example-Attack-and-Defense/blob/master/Adversarial_Example_%28Attack_and_defense%29.ipynb?authuser=3

DeepFool model: Results





https://colab.research.google.com/drive/1f-6jaa3oB3U4gBJn0Dc88bhJdtidYmKN?authuser=3

FGSM vs DeepFool

FGSM

- *Simple to implement and fast to produce*
- *Effective in producing small perturbations that can cause misclassification*
- *Can produce high-frequency noise that can be detected by human eyes*

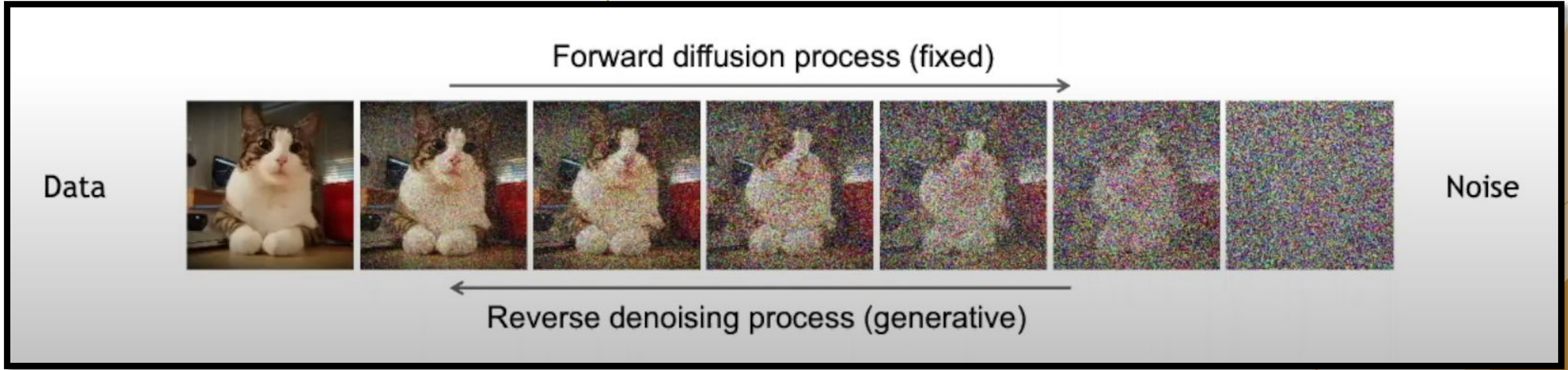
DeepFool

- *Time-consuming and resource-intensive.*
- *Effective in generating hard-to-detect adversarial examples that can be used to evaluate the robustness of a model.*
- *Produces a misclassified image with low frequency noise*

- Future Work

Guided Diffusion Model for Purification (GDMP)

- The basic idea behind this technique is to reduce the amount of noise present in an image while preserving the important details or edges in the image.



https://www.youtube.com/watch?v=cS6JQpEY9cs

Bibliography

- Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
- Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- Wang, Jinyi, et al. "Guided diffusion model for adversarial purification." arXiv preprint arXiv:2205.14969 (2022).
- Chakraborty, Anirban, et al. "A survey on adversarial attacks and defences." CAAI Transactions on Intelligence Technology 6.1 (2021): 25-45.
- https://colab.research.google.com/github/as791/Adversarial-Example-Attack-and-Defense/blob/master/Adversarial_Example_%28Attack_and_defense%29.ipynb?authuser=3
- https://colab.research.google.com/drive/1f-6jaa3oB3U4gBJn0Dc88bhJdtidYmKN?authuser=3

- Thank You