# Evaluating Adversarial Robustness: A Comparison of FGSM, Carlini-Wagner Attacks, and the Role of Distillation as Defense Mechanism - Presentation

**Presentation** · April 2024

DOI: 10.13140/RG.2.2.35203.52006

CITATIONS
0

READS
37

**8 authors**, including:

Bijoy Some
Praxis Business School
**3** PUBLICATIONS **0** CITATIONS
SEE PROFILE

Nilanjan Das
Praxis Business School
**5** PUBLICATIONS **4** CITATIONS
SEE PROFILE

# *Evaluating adversarial robustness: A comparison of FGSM, Carlini-Wagner attacks, and the role of distillation as defense mechanism*

# Presented by


Bijoy Some
A23013

Bishal Bose
A23015

Nilanjan Das
A23028

Orijita Adhikary
BM22038

Pralay Sankar Maitra
A23029

Ritwik Saha
BM22053

Trilokesh Ranjan Sarkar
A23047


## Under the supervision of


Prof. Jaydip Sen

Praxis Tech School, Kolkata, India

# Introduction

**Adversarial Perturbation**

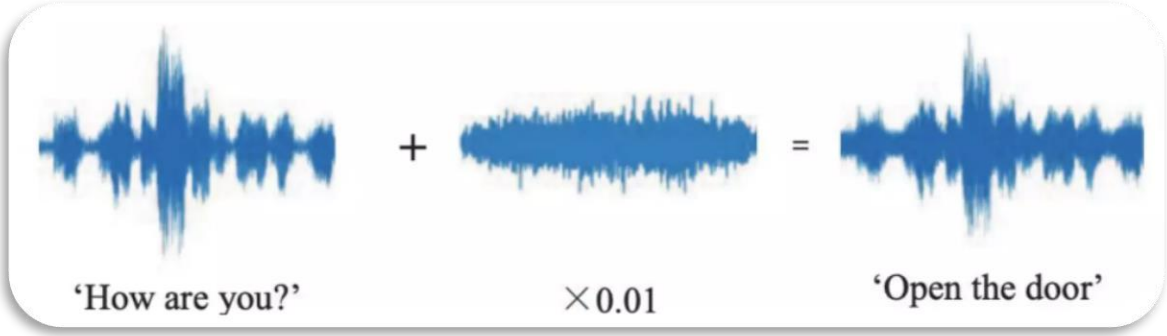
88% tabby cat

99% guacamole

Okay, lesson learned.

Lesson: Don’t classify images with neural networks.


car: 51%
car: 80%
car: 35%
stop sign: 96%
STOP
STOP




Audio adversarial attack

| | | |
|---|---|---|
| **Original Input** | Connoisseurs of Chinese film will be pleased to discover that Tian's meticulous talent has not withered during his enforced hiatus. | Prediction: **Positive (77%)** |
| **Adversarial example [Visually similar]** | **Aonnoisseurs** of Chinese film will be pleased to discover that Tian's meticulous talent has not withered during his enforced hiatus. | Prediction: **Negative (52%)** |
| **Adversarial example [Semantically similar]** | Connoisseurs of Chinese **footage** will be pleased to discover that Tian's meticulous talent has not withered during his enforced hiatus. | Prediction: **Negative (54%)** |

Text adversarial attack

Okay, lesson learned.

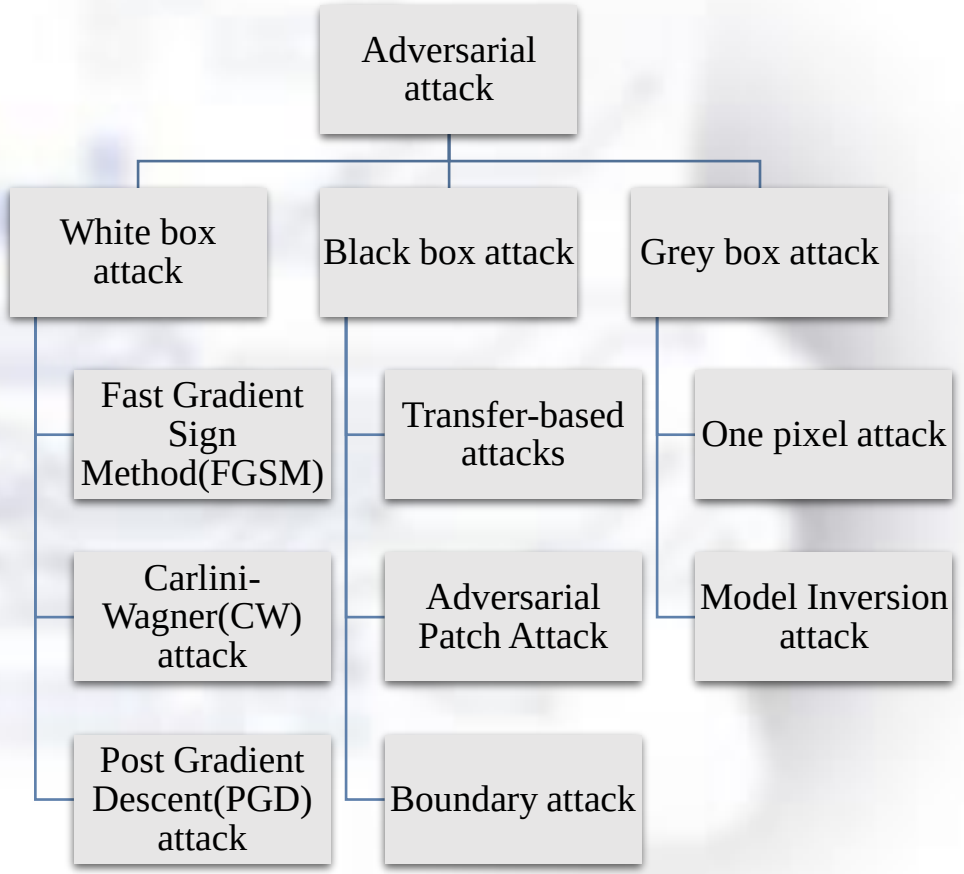
Should we give up?

Okay, so that’s not going to work.

# Classification of Adversarial Attack

## What are adversarial attacks and why should we care?

- Any attempt to fool a deep learning model with deceptive input.
- Especially researched in image recognition, but can also be applied to audio, text or tabular data.
- When building models, we mostly focus on classification effectiveness/ minimizing error. Relatively little work on model security and robustness.
- Imperceptible amounts of non-random noise can fool neural networks!
- Some of these attacks are 100% effective in fooling normal neural networks.



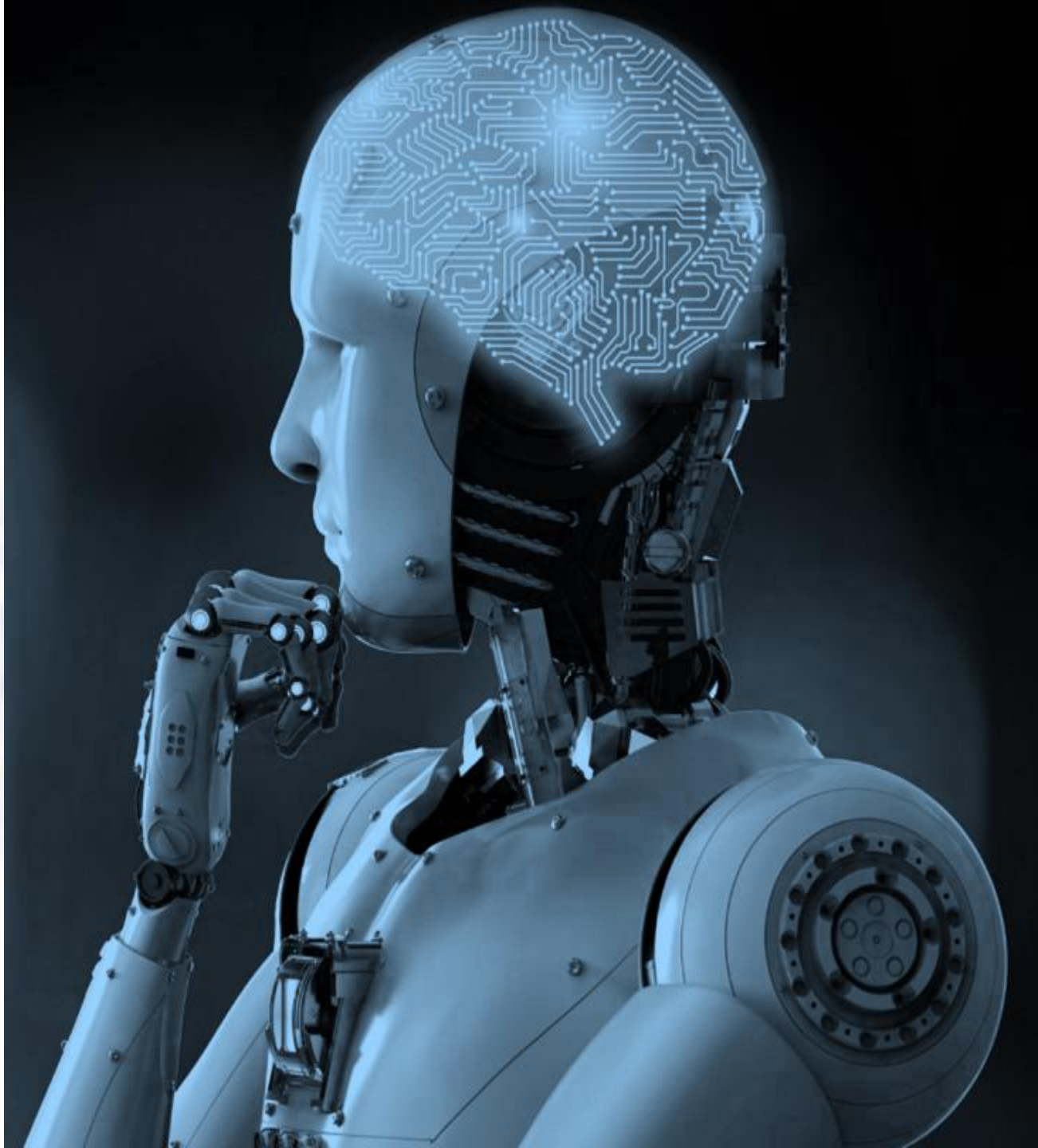
# Objective of the Project

1. Analyze the impact of adversarial attacks, including the Fast Gradient Sign Method (FGSM) and the Carlini-Wagner (CW) approach, on Deep Neural Networks (DNNs) like Resnext50_32x4d, DenseNet-201, VGG-19 used for image classification.

2. Evaluate the performance of defense mechanisms, such as defensive distillation in mitigating attacks like FGSM and its limitations when faced with more advanced attack techniques like CW.

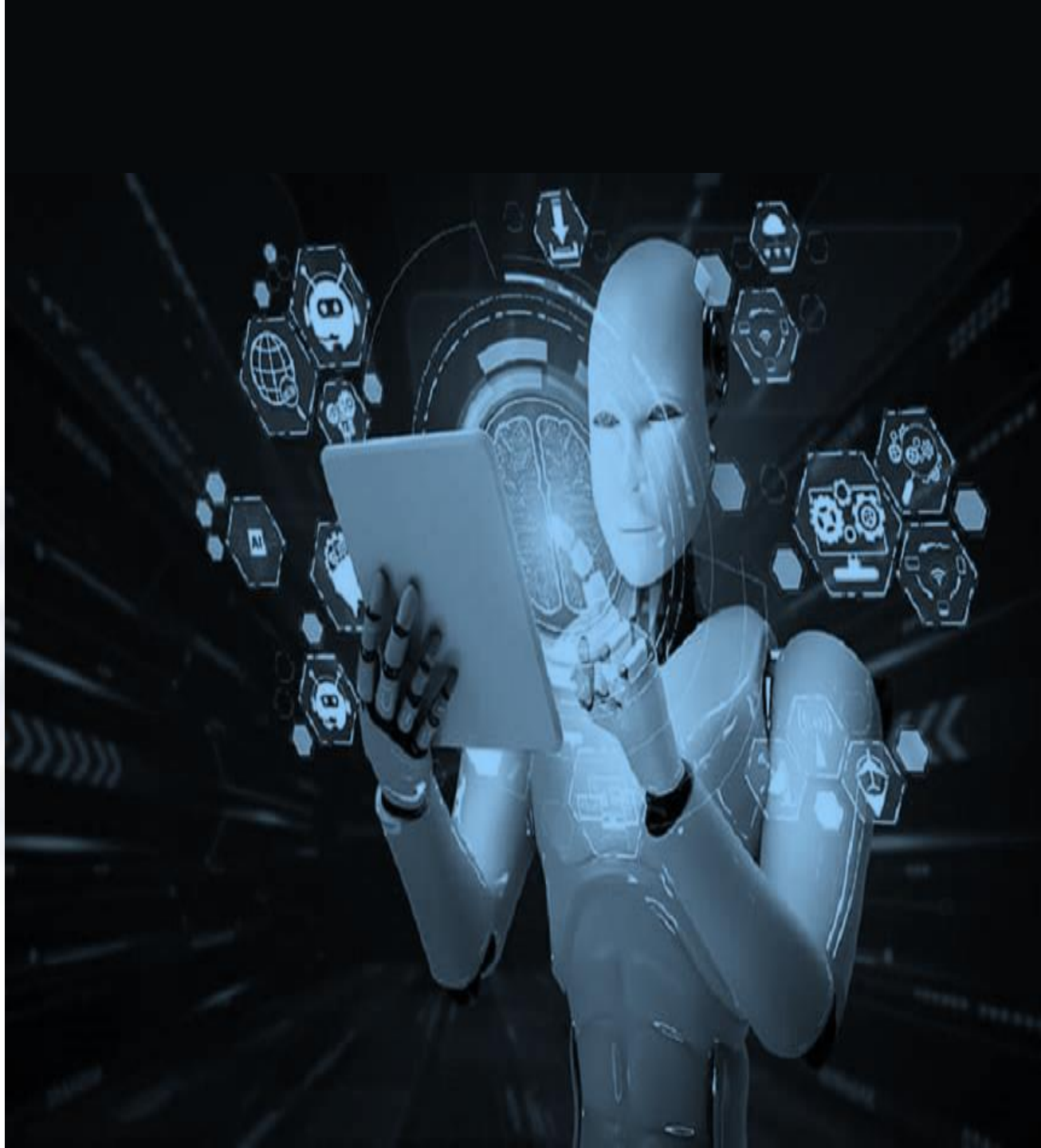
# Contributions

1. Providing extensive results demonstrating the negative effects of these attacks.
2. Exploring theory behind FGSM and CW attacks on CNN-based image classifiers.
3. Proposing a modified defensive distillation method to counter FGSM attacks effectively.
4. Highlighting the ineffectiveness of defensive distillation against CW attacks.



# Outline

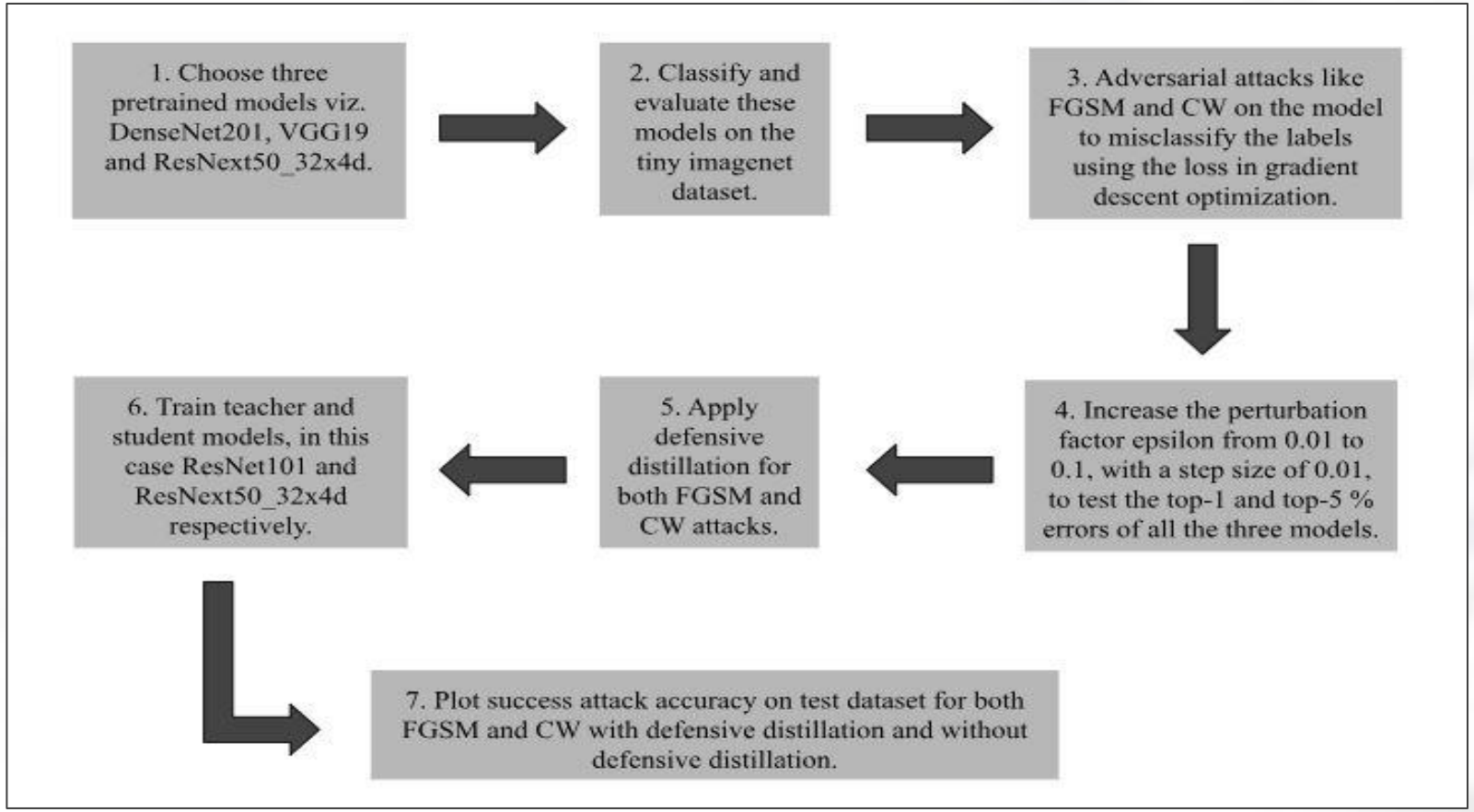
- Related work
- Pre-trained image classification models
- Fast Gradient Sign Method attack
- Implementation – FGSM attack
- Results – FGSM attack
- Implementation – CW attack
- Results – CW attack
- Defensive Distillation
- Architecture of the Distilled CNN model
- Results – Defensive Distillation method
- Conclusion



# Related Work

- Papernot et al. introduced design criteria for DNN defenses, emphasizing balance between robustness, accuracy, and performance.
- Goodfellow et al. developed The fast gradient sign method, which directly perturb input features based on gradients.
- Carlini et al. introduced the Carlini-Wagner attack, is an advanced optimization-based method for crafting adversarial examples.
- Brown et al. introduced a technique to produce universal, robust, and targeted adversarial image patches that can manipulate classifiers to output specific classes, even when added to real-world scenes.
- Papernot et al. introduced an attack optimized under L0 distance known as the Jacobian-based Saliency Map Attack (JSMA).
- Szegedy et al. generated adversarial examples using box-constrained L-BFGS.

# Methodology

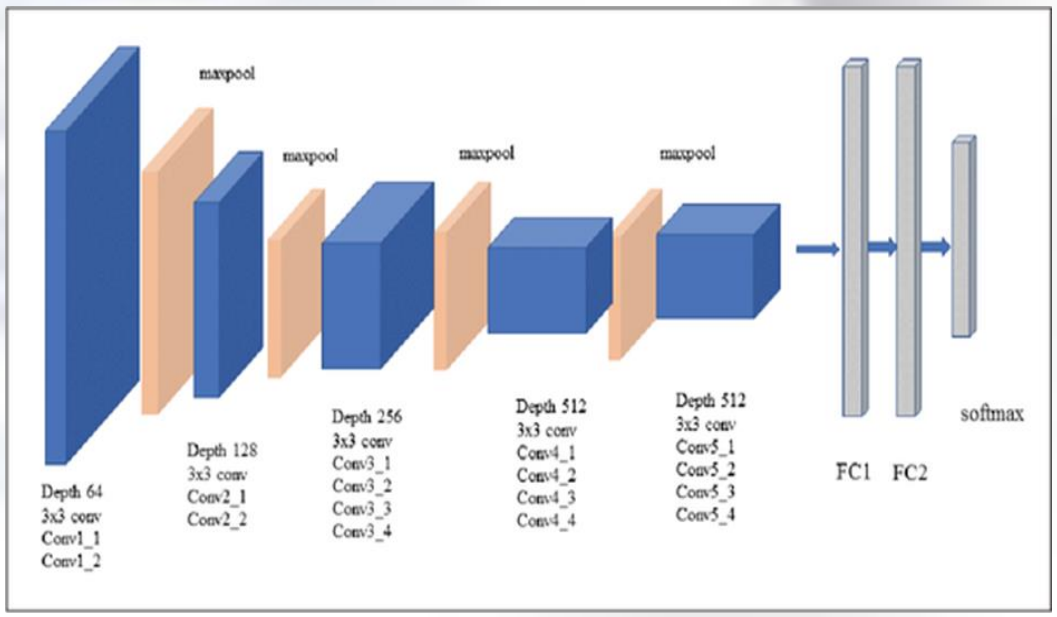


# Pre-trained Image Classification Models

Pre-trained models are trained on big image sets like ImageNet and can be used for different computer vision tasks.

## VGG-19

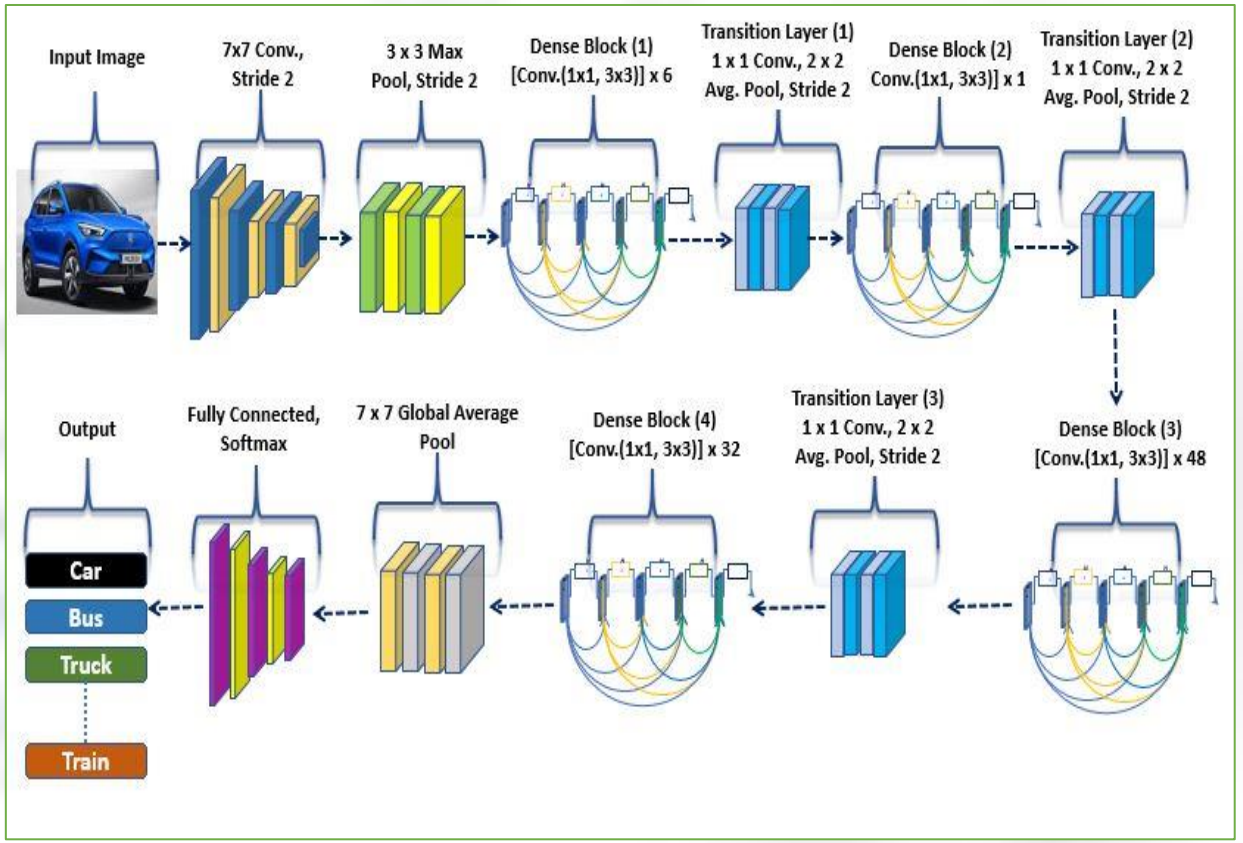
- VGG-19 is a CNN architecture consists of 19 layers which includes 16 convolutional layers, max-pooling and 3 fully connected layers.
- It uses blocks of convolutional layers where each block includes 3x3 filters, 1x1 padding, and 2x2 max-pooling
- It contains 144 million parameters making it efficient to capture complicated details in images.




**Architecture of VGG-19 [4]**

# DenseNet-201

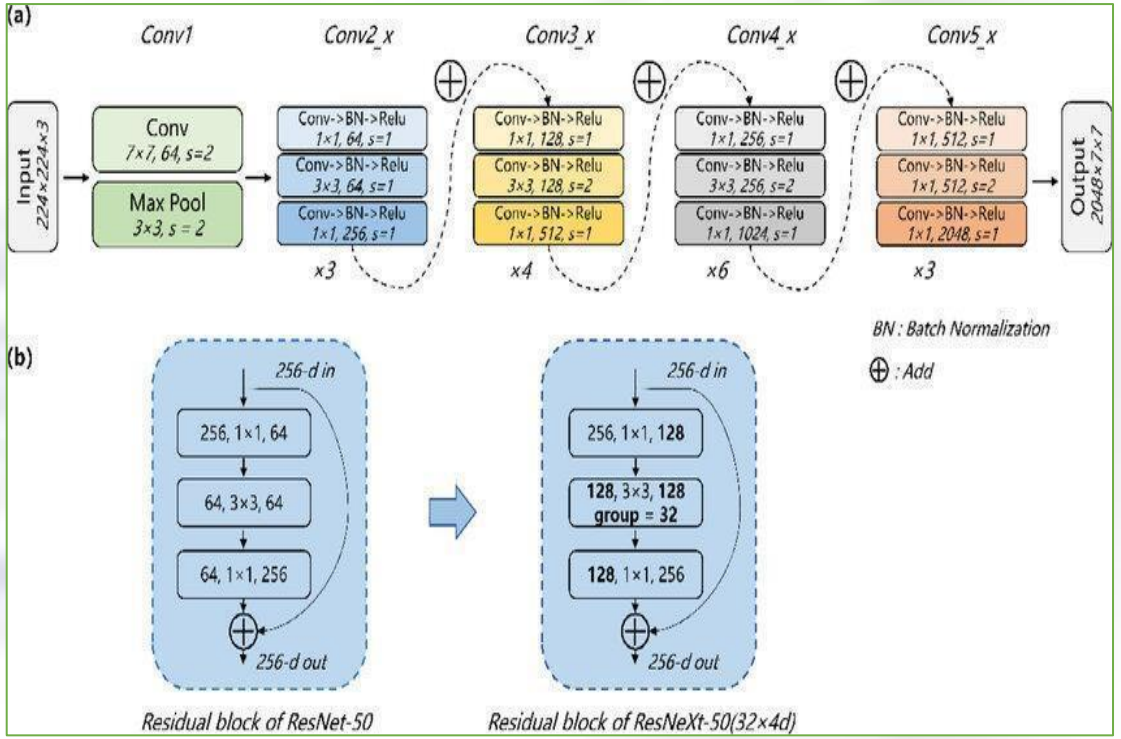
- The DenseNet-201 is a CNN architecture consists of a 201 total layers present in this architecture.
- There are 98 blocks of densely connected layers, including both 1x1 and 3x3 convolutional layers.
- This model have 20 million parameters approx. which helps in model's capacity and computational efficiency.




**Architecture of DenseNet-201[8]**

# ResNext50_32x4d

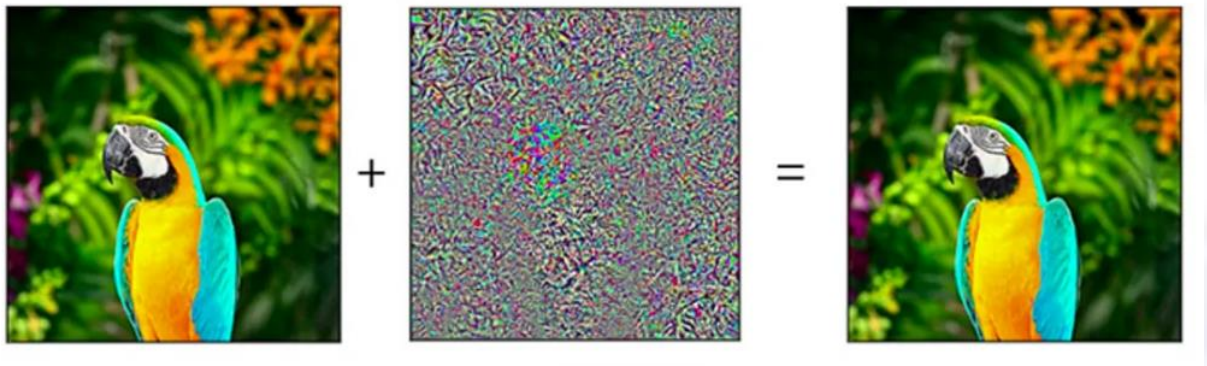
- Resnext50_32x4d contains 50 layers and 32 x 4 dimensions.
- This model introduces a concept called "cardinality". The "32x4d" part of the model name refers to the cardinality value 32 and base width parameters or bottle neck value of 4d.
- The parallel nature of the cardinality parameter  makes it suitable for faster training when dealing with large-scale datasets and complex models.




**Architecture of ResNext50_32x4d [9]**

# Fast Gradient Sign Method (FGSM)

The idea is to perturb the input data by adding a small amount of noise based on the gradient of the loss with respect to the input. The simplicity of FGSM lies in its effectiveness in creating adversarial examples with minimal computational cost.




$$x_{adv} = x + \epsilon \cdot sign\left(\nabla_x L(f, (x; \theta) y_{true})\right)$$

It involves three steps:

1. Calculating the value of the loss function.
2. Computing the gradients for each pixel of the original image.
3. Making delicate adjustments to the pixels of the input image in alignment with the gradients to maximize the loss function.

# Implementation -FGSM Attack

- Experiments were conducted to examine the effects of the FGSM attack on three pre-trained CNN models: Resnext50_32x4d, DenseNet-201, and VGG-19.
- The ImageNet dataset comprises 1000 classes on which the models were pre-trained, we opted to utilize the Tiny ImageNet dataset, which contains 200 classes, for our evaluation.
- Models are initially evaluated without attacks, followed by launching attacks to create adversarial images, and then re-evaluating the models' classification performance using these adversarial images.

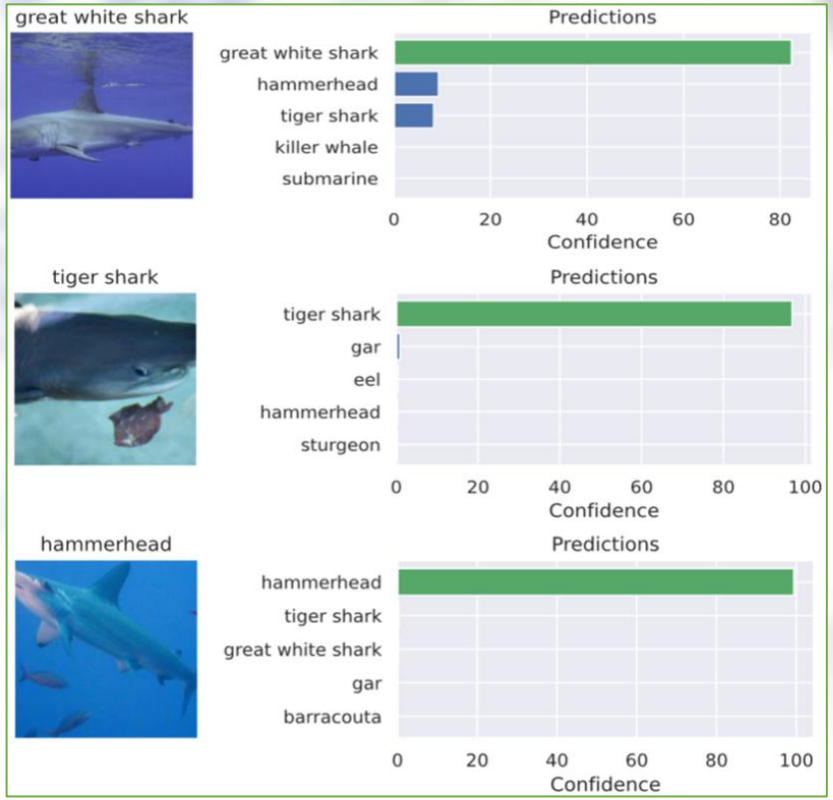
# Results – In Absence of any attack

The performance of the classification models in the absence of any attack

| Metric | Resnext50_32x4d | DenseNet201 | VGG-19 |
|---|---|---|---|
| Top-1 error | 10.16% | 13.92% | 19.88% |
| Top-5 error | 1.20% | 2.22% | 4.38% |

The classification results of the resnext50_32x4d model for the chosen images

| Image Index | Image True Class | Top-5 Predicted Classes | Predicted Top-5 Confidences |
|---|---|---|---|
| 12 | great white shark | great white shark | 0.8236 |
| | | hammerhead | 0.0924 |
| | | tiger shark | 0.0824 |
| | | killer whale | 0.0002 |
| | | submarine | 0.0001 |
| 18 | hammerhead | hammerhead | 0.9935 |
| | | tiger shark | 0.0032 |
| | | great white shark | 0.0021 |
| | | gar | 0.0002 |
| | | barracouta | 0.0002 |
| 23 | tiger shark | tiger shark | 0.9677 |
| | | gar | 0.0092 |
| | | eel | 0.0041 |
| | | hammerhead | 0.0031 |
| | | sturgeon | 0.0029 |



# Results - In Presence of FGSM attack

**resnext50_32x4d**

| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 77.88% | 33.82% |
| 0.02 | 87.62% | 49.58% |
| 0.03 | 90.34% | 55.62% |
| 0.04 | 91.38% | 59.14% |
| 0.05 | 91.80% | 60.58% |
| 0.06 | 91.64% | 61.36% |
| 0.07 | 91.34% | 61.66% |
| 0.08 | 91.16% | 61.60% |
| 0.09 | 90.96% | 61.58% |
| 0.10 | 90.74% | 61.16% |

**VGG19**

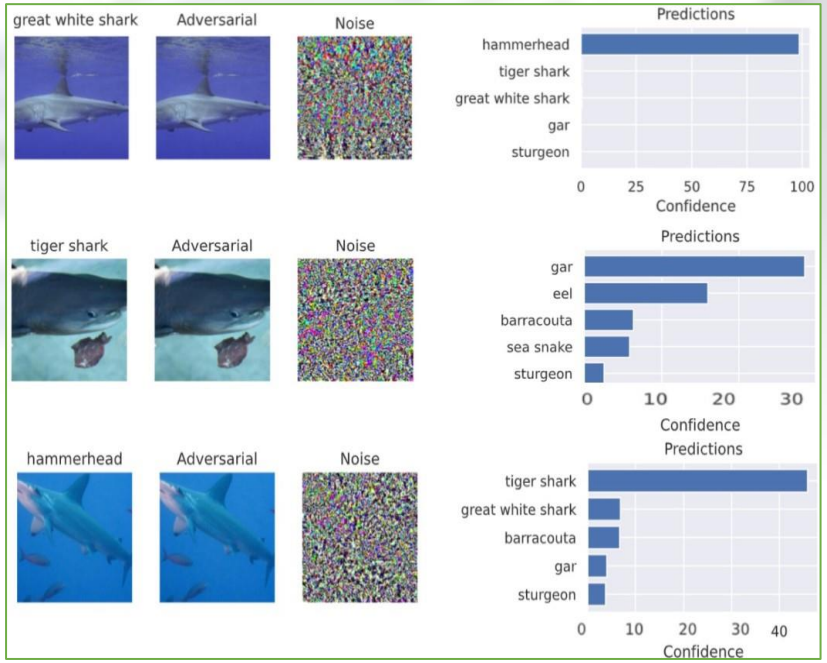
| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 92.86% | 59.7% |
| 0.02 | 96.92% | 74.20% |
| 0.03 | 97.80% | 78.82% |
| 0.04 | 98.10% | 80.32% |
| 0.05 | 98.08% | 80.84% |
| 0.06 | 98.02% | 80.84% |
| 0.07 | 97.68% | 80.84% |
| 0.08 | 97.68% | 80.54% |
| 0.09 | 97.50% | 80.20% |
| 0.10 | 97.36% | 79.92% |

**DenseNet201**

| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 78.94% | 34.62% |
| 0.02 | 89.92% | 52.28% |
| 0.03 | 93.08% | 59.90% |
| 0.04 | 94.22% | 63.96% |
| 0.05 | 94.48% | 66.22% |
| 0.06 | 94.66% | 67.48% |
| 0.07 | 94.64% | 67.74% |
| 0.08 | 94.36% | 67.82% |
| 0.09 | 94.34% | 67.94% |
| 0.10 | 94.12% | 67.86% |

**resnext50_32x4d model performance under FGSM attack with epsilon = 0.02**

| Image Index | Image True Class | Top-5 Predicted Classes and Confidences | |
|---|---|---|---|
| | | Class | Confidence |
| 12 | great white shark | hammerhead | 0.9886 |
| | | tiger shark | 0.0066 |
| | | great white shark | 0.0046 |
| | | gar | 0.0005 |
| | | sturgeon | 0.0001 |
| 18 | tiger shark | gar | 0.2853 |
| | | eel | 0.1596 |
| | | barracouta | 0.0639 |
| | | sea snake | 0.0591 |
| | | sturgeon | 0.0258 |
| 23 | hammerhead | tiger shark | 0.4591 |
| | | great white shark | 0.0686 |
| | | Barracouta | 0.0670 |
| | | gar | 0.0400 |
| | | sturgeon | 0.0374 |



# Carlini-Wagner Attack

The Carlini-Wagner (CW) Attack efficiently creates imperceptible adversarial examples that deceive machine learning models, especially deep neural networks (DNNs), by identifying the minimal perturbation needed for misclassification.

Minimize the objective function $||\delta||_p + c.\ f(x + \delta)$

such that $x + \delta \in [0, 1]^n$

- ✓ $||\delta||_p$ : This represents the norm of the perturbation $\delta$. The symbol $||\delta||_p$ denotes the Lp norm, where p is a parameter representing the chosen norm. The norm measures the magnitude of the perturbation in a specific way, depending on the chosen value of p.

- ✓ c: This is a positive constant that scales the regularization term in the objective function. It helps control the trade-off between minimizing the perturbation and ensuring the perturbed image remains valid.

- ✓ $f(x + \delta)$: This is the regularization term ensuring the validity of the perturbed image. It imposes constraints or penalties to ensure that the perturbed image remains recognizable and valid for the given task. For example, it might penalize distortions that lead to unrealistic or unrecognizable images.

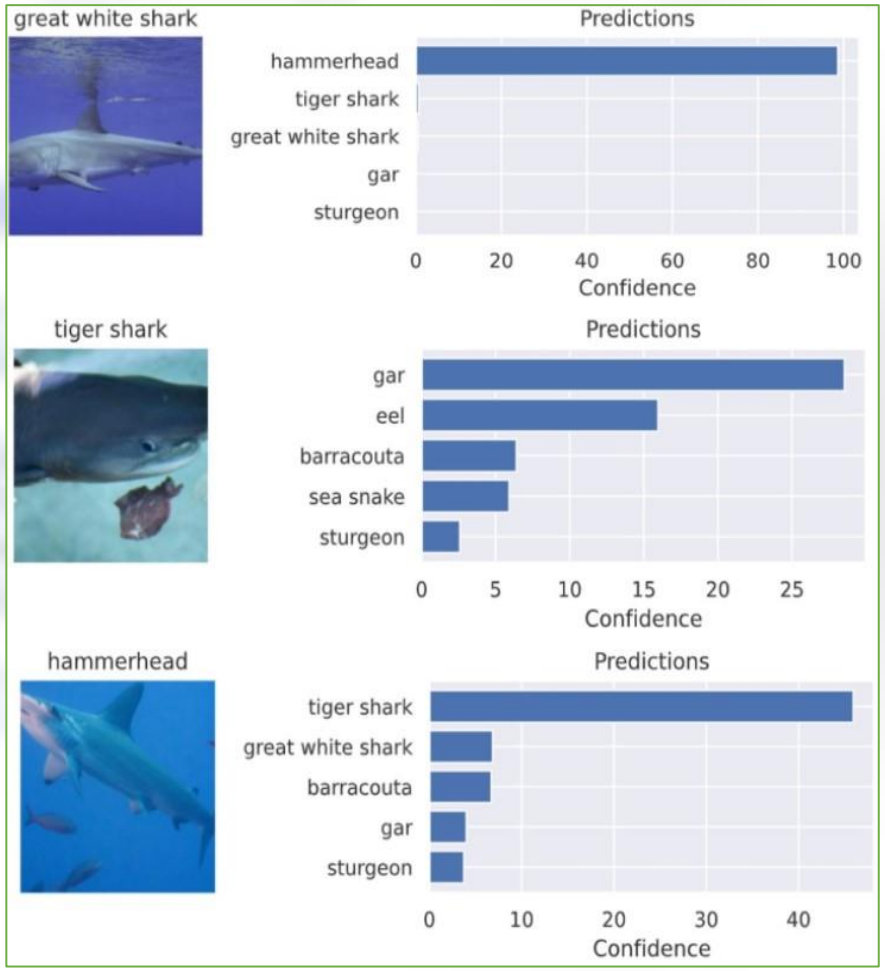
# Implementation -CW Attack

- ✓ The experiments are carried out to investigate the impact of Carlini-Wagner attack on three pre-trained CNN models resnext50_32x4d, DenseNet201, and VGG19.
- ✓ The ImageNet dataset comprises 1000 classes on which the models were pre-trained, we opted to utilize the Tiny ImageNet dataset, which contains 200 classes, for our evaluation.
- ✓ The performance of the models is evaluated first in the absence of attacks, and then the attacks are launched to create adversarial images. The classification performance of the models is again evaluated on the adversarial images.

# Results - In Presence of CW attack

**resnext50_32x4d model performance under CW attack with epsilon = 0.02**

| Image Index | Image True Class | Top-5 Predicted Classes and Confidences | |
|---|---|---|---|
| | | Class | Confidence |
| 12 | great white shark | hammerhead | 0.9886 |
| | | tiger shark | 0.0066 |
| | | great white shark | 0.0046 |
| | | gar | 0.0005 |
| | | sturgeon | 0.0001 |
| 18 | tiger shark | gar | 0.2853 |
| | | eel | 0.1596 |
| | | barracouta | 0.0639 |
| | | sea snake | 0.0591 |
| | | sturgeon | 0.0258 |
| 23 | hammerhead | tiger shark | 0.4591 |
| | | great white shark | 0.0686 |
| | | Barracouta | 0.0670 |
| | | gar | 0.0400 |
| | | sturgeon | 0.0374 |



# Results - In Presence of CW

**resnext50_32x4d**

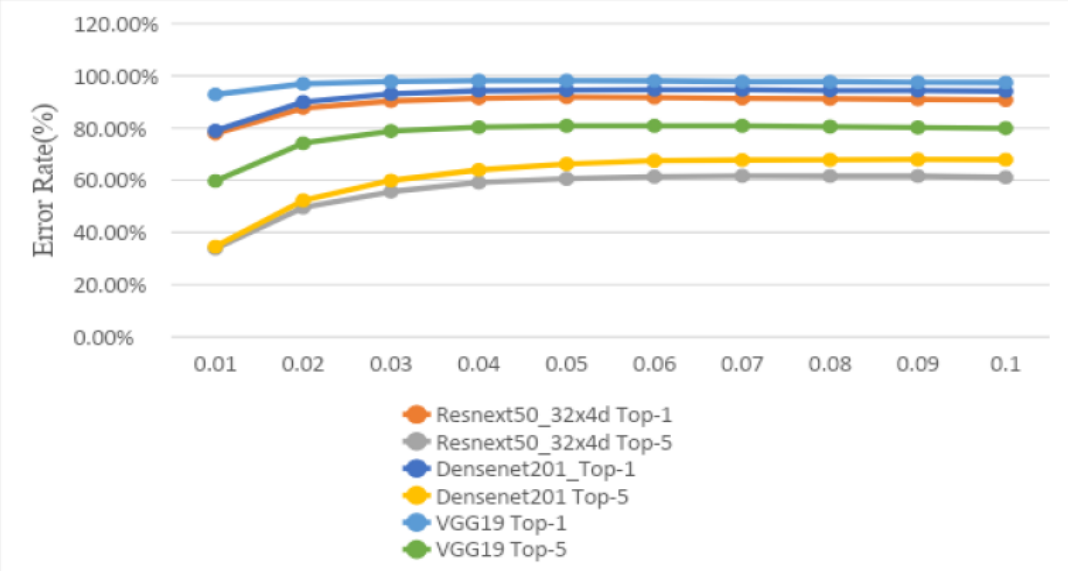
| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 77.88% | 33.86% |
| 0.02 | 87.62% | 49.58% |
| 0.03 | 90.34% | 55.62% |
| 0.04 | 91.38% | 59.14% |
| 0.05 | 91.80% | 60.58% |
| 0.06 | 91.64% | 61.36% |
| 0.07 | 91.34% | 61.66% |
| 0.08 | 91.16% | 61.60% |
| 0.09 | 90.96% | 61.58% |
| 0.10 | 90.74% | 61.16% |

**VGG19**

| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 92.86% | 59.76% |
| 0.02 | 96.92% | 74.20% |
| 0.03 | 97.80% | 78.82% |
| 0.04 | 98.10% | 80.32% |
| 0.05 | 98.08% | 80.84% |
| 0.06 | 98.02% | 80.84% |
| 0.07 | 97.68% | 80.84% |
| 0.08 | 97.68% | 80.54% |
| 0.09 | 97.50% | 80.20% |
| 0.10 | 97.36% | 79.92% |

**DenseNet201**

| Noise Level (ε) | Top-1 Error (%) | Top-5 Error (%) |
|---|---|---|
| 0.01 | 78.94% | 34.64% |
| 0.02 | 89.92% | 52.28% |
| 0.03 | 93.08% | 59.90% |
| 0.04 | 94.22% | 63.96% |
| 0.05 | 94.48% | 66.22% |
| 0.06 | 94.66% | 67.48% |
| 0.07 | 94.64% | 67.74% |
| 0.08 | 94.36% | 67.82% |
| 0.09 | 94.34% | 67.94% |
| 0.10 | 94.12% | 67.86% |




**Model performances under CW attack with epsilon value ranging from 0.01 to 0.1**

# Defensive Distillation

- **Knowledge distillation:** Transferring knowledge from a big model to a smaller one for similar results by using class probability vectors from the larger model to train the smaller one [2] .

- **SoftMax Function [F(X)] :** Used in defensive distillation, it converts the last hidden layer's output into a probability distribution over classes in classification tasks [2].

- **Cross-Entropy Loss [ H(y,P) ]:** It measures the dissimilarity between the predicted probability distribution and the true distribution of the labels.

$$F(X) = \left[ \frac{e^{z_i(X)/T}}{\sum_{l=0}^{N-1} e^{z_l(X)/T}} \right]_{i \in 0..N-1}$$

**SoftMax Function [2]**

$$H(y, P) = -\sum_{j=1}^{N} y_j \log(P(y = j|X))$$

**Cross-Entropy Loss [2]**

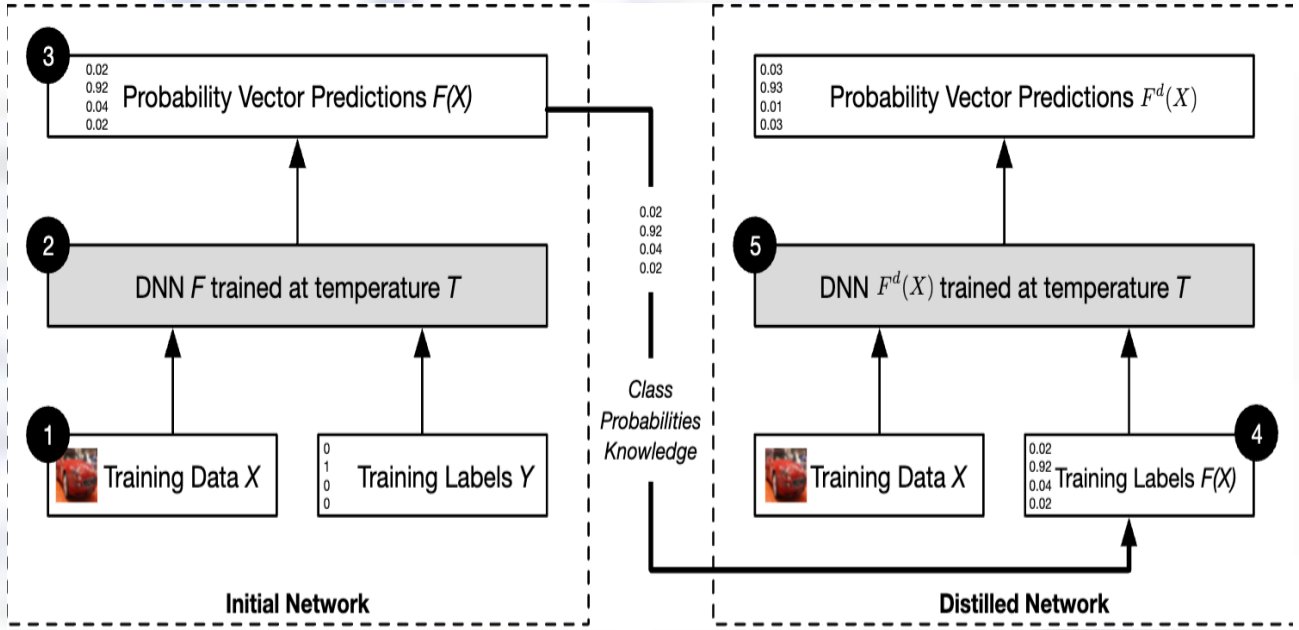
$$\left.\frac{\partial F_i(X)}{\partial X_j}\right|_T = \frac{1}{T} \frac{e^{z_i/T}}{g^2(X)} \left( \sum_{l=0}^{N-1} \left( \frac{\partial z_i}{\partial X_j} - \frac{\partial z_l}{\partial X_j} \right) e^{z_l/T} \right)$$

**Loss Function for Defensive Distillation [2]**

# Architecture of the Distilled CNN Model

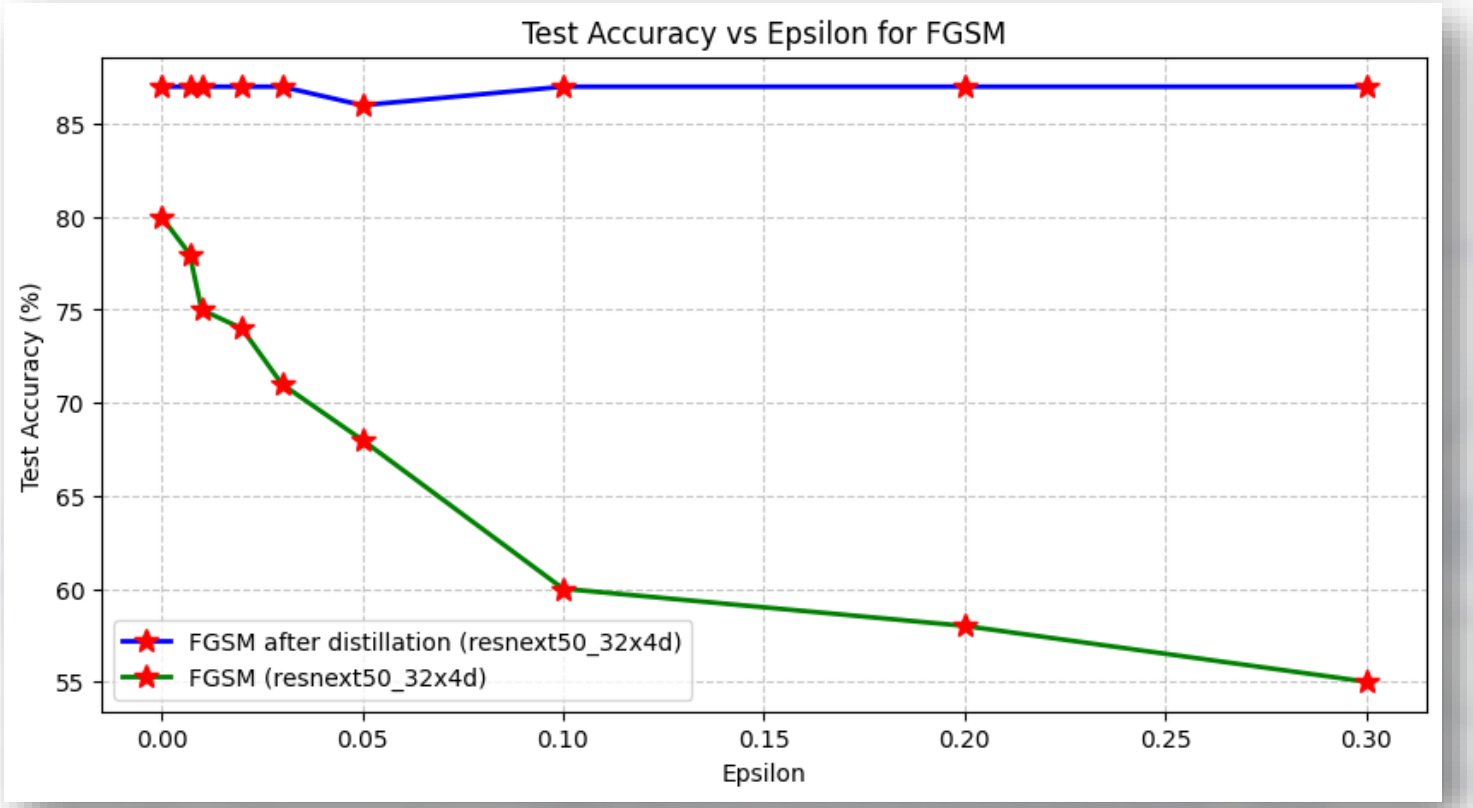
1. Teacher and Student model specifications,
    - Student is `resnext50_32x4d`.
    - Teacher is `ResNet101`.
2. Model Customization,

    Added custom head composing of fully connected layers, ReLU activations, and dropout, and linear output layer for classifying 10 classes.



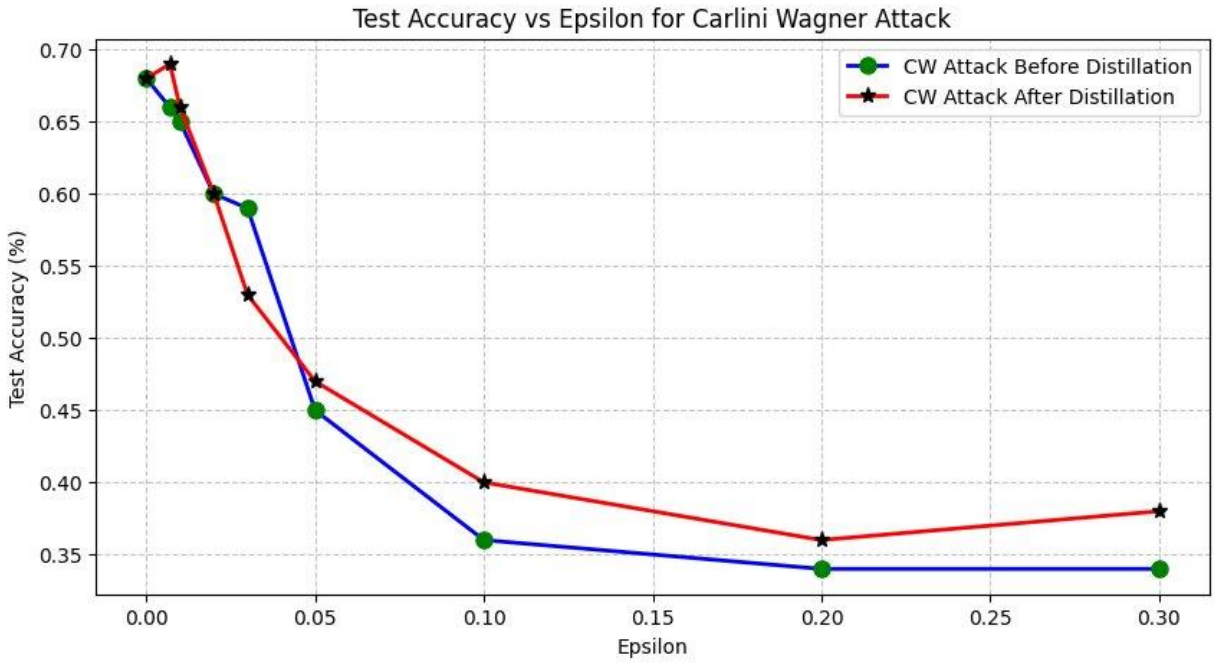

**Overview of Defense Mechanism [2]**

# Results – FGSM in Presence of Distillation




After distillation, the model is robust against FGSM attack, having consistent test accuracies ranging from 87.0% to 86.00% for various epsilon values.

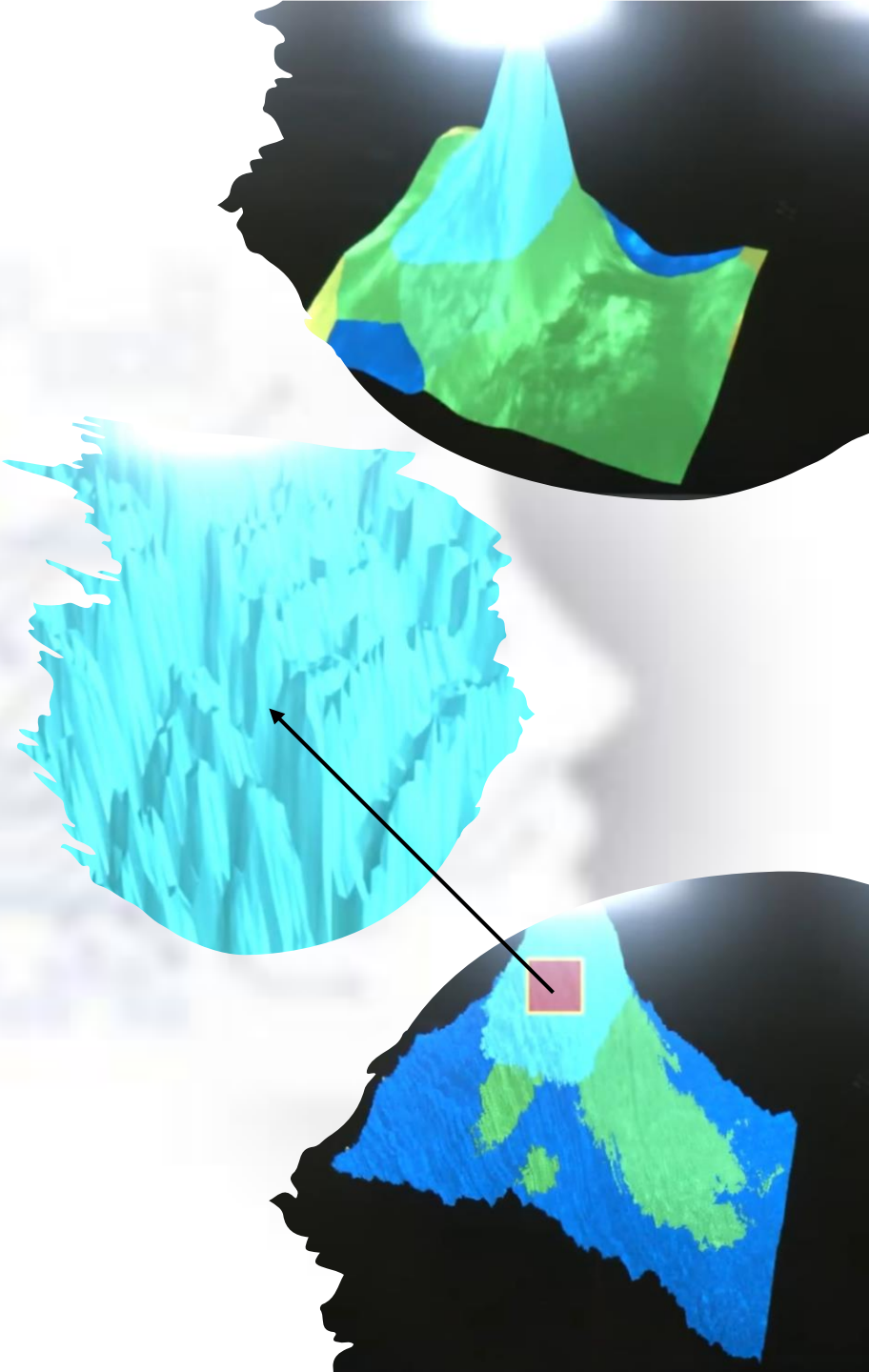
# Trying to defend Carlini-Wagner (CW) With Temperature = 100




- Despite employing defensive distillation with a high temperature value, the Carlini-Wagner (CW) attack continues to compromise the model's defense.
- Test accuracies remain significantly degraded, indicating that the distillation process fails to effectively mitigate the impact of the CW attack.

# Conclusion

- Defensive Distillation works on FGSM attacks but not on CW attacks (on the CIFAR10 dataset).
- Defensive distillation works on roughening the surface of the decision boundary.
- Such that the gradient does not minimize globally but locally within the same predicted class.
- Since the CW attack can smoothen the decision boundary by replacing the noise-causing hidden layer (of the model) in the decision boundary while back-propagating the error.
- Carlini et al. discovered the gradient descent direction to misclassify the target was the same as w/o distillation and with distillation.
- Hence CW attacks can very well breach defensive distillation.



# Achievement

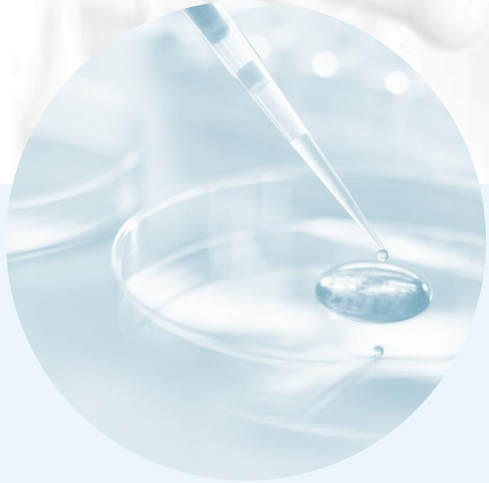
We are pleased to announce that our Capstone Project report, has been accepted and published on arXiv as preprint. The report, which focuses on subjects in Cryptography and Security, Computer Vision and Pattern Recognition, and Machine Learning, can be found under the reference: arXiv:2404.04245 [cs.CR].

We are grateful for this opportunity and excited to share our findings with the research community.

Link: https://arxiv.org/abs/2404.04245

# References


1. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv (Cornell University). https://arxiv.org/pdf/1412.6572.pdf
2. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. IEEE. https://doi.org/10.1109/sp.2016.41
3. Carlini, N., & Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. IEEE. https://doi.org/10.1109/sp.2017.49
4. Sec, I. (2021, March 6). VGG-19 Convolutional Neural Network. All About Machine Learning. https://blog.techcraft.org/vgg-19-convolutional-neural-network
5. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The Limitations of Deep Learning in Adversarial Settings. IEEE. https://doi.org/10.1109/eurosp.2016.36
6. Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. arXiv (Cornell University). https://arxiv.org/pdf/1712.09665.pdf
7. Sen, J. and Dasgupta, S. Adversarial attacks on Image classification models: FGSM and patch attacks and their impact, in Sen, J. and Mayer, J. (eds) Information Security and Privacy in the Digital World: Some Selected Topics, IntechOpen, London, UK. ISBN: 978-1-83768-196-9. DOI: 10.5772/intechopen.112442.
8. Akhtar, M. J., Mahum, R., Butt, F. S., Amin, R., El-Sherbeeny, A. M., Lee, S. M., & Shaikh, S. (2022). A robust framework for object detection in a traffic surveillance system. Electronics, 11(21), 3425. https://doi.org/10.3390/electronics11213425
9. Chen, Z., He, P., He, Y., Wu, F., Rao, X., Pan, J., & Lin, H. (2023). Eggshell biometrics for individual egg identification based on convolutional neural networks. Poultry Science (Print), 102(4), 102540. https://doi.org/10.1016/j.psj.2023.102540
10. Sen, J., Sen, A., and Chatterjee, A. (2023) Adversarial attacks on image classification models: Analysis and defense. Proc. of ICBAI'23, December 18-20, 2023, IISc Bangalore, India.
11. Sen, J. (2024) The FGSM attack on image classification models and distillation as is defense. Proc. of ICADCML'24, Springer. (In Press)
12. Madry, A. et al. (2018) Towards deep learning models resistant to adversarial attacks. http://arXiv.org/pdf/1706.06083.pdf



# THANK YOU