International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | December - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Dissecting Adversarial Attacks: A Comparative Analysis of Adversarial Perturbation Effects on Pre-Trained Deep Learning Models

Rekha Kumari[1], Tushar Bhatia[2], Peeyush Kumar Singh[3], Kanishk Vikram Singh[4]

[1]*Assistant Professor, Department of Computer Science and Engineering, HMR Institute of Technology and Management, Delhi, India*

[2,3,4] *Student, Department of Computer Science and Engineering, HMR Institute of Technology and Management, Delhi, India*

Abstract - It is well known that the majority of neural networks widely employed today are extremely susceptible to adversarial perturbations which causes the misclassification of the output. This, in turn, can cause severe security concerns. In this paper, we meticulously evaluate the robustness of prominent pre-trained deep learning models against images that are modified with the Fast Gradient Sign Method (FGSM) attack. For this purpose, we have selected the following models: InceptionV3, InceptionResNetV2, ResNet152V2, Xception, DenseNet121, and MobileNetV2. All these models are pre-trained on ImageNet, and hence, we use our custom 10-animals test dataset to produce clean as well as misclassified output. Rather than focusing solely on prediction accuracy, our study uniquely quantifies the perturbation required to alter output labels, shedding light on the models' susceptibility to misclassification. The outcomes underscore varying vulnerabilities among the models to FGSM attacks, providing nuanced insights crucial for fortifying neural networks against adversarial threats.

Key Words: Adversarial Perturbations, Deep Learning, ImageNet, FGSM Attack, Neural Networks, Pre-trained Models

1. INTRODUCTION

The proliferation of neural networks across various domains has revolutionized the landscape of artificial intelligence, enabling remarkable advancements in tasks ranging from image recognition to natural language processing. Despite their widespread adoption, neural networks are not impervious to adversarial attacks—sophisticated manipulations of input data designed to deceive the model and induce misclassifications. This vulnerability has raised critical concerns about the robustness and reliability of neural network-based system.

1.1 Neural Network and Their Vulnerabilities

Neural networks, inspired by the human brain, consist of interconnected layers of nodes that process information hierarchically. Trained on vast datasets, these networks demonstrate an impressive ability to generalize patterns and make accurate predictions. However, their reliance on complex mathematical functions leaves them susceptible to adversarial perturbations—subtle alterations to input data that lead to unpredictable and often incorrect outputs

1.2 The Menace of Adversarial Attacks

Adversarial attacks, a well-documented challenge in the field of machine learning, exploit the sensitivity of neural networks to imperceptible changes in input data. These attacks can manifest in various forms, aiming to manipulate models into making incorrect predictions. Among these, the Fast Gradient Sign Method (FGSM) stands out as a powerful and computationally efficient technique The FGSM attack capitalizes on the gradients of a neural network's loss function to perturb input data strategically, inducing misclassifications with remarkable efficiency. Understanding and mitigating such attacks are crucial for ensuring the reliability and security of neural network-based systems, especially in applications where erroneous decisions could have significant consequences.

1.3 Significance of Evaluating Model Vulnerability

As neural networks continue to permeate critical domains such as healthcare, finance, and autonomous systems, ensuring their robustness against adversarial attacks becomes paramount. This research delves into the vulnerability of widely employed pre-trained deep learning models when subjected to the FGSM attack. The selected models—InceptionV3, InceptionResNetV2, ResNet152v2, Xception, DenseNet121, and MobileNet2—represent a spectrum of architectures commonly utilized in real-world applications. The inclusion of these models in our comparative analysis aims to capture a broad spectrum of responses to adversarial perturbations. A detailed exploration of these architectures will be presented in Section 3, offering insights into the diverse structures that underpin our comparative analysis.

1.4 Objectives of the Study

Unlike traditional evaluations that focus solely on prediction accuracy, our study extends its gaze to the nuanced realm of

adversarial attacks. The primary objectives of this study include:

1. Evaluating the vulnerability of widely employed pre-trained deep learning models to adversarial attacks, specifically focusing on the Fast Gradient Sign Method (FGSM).
2. Selecting a diverse set of pre-trained models, including InceptionV3, InceptionResNetV2, ResNet152v2, Xception, DenseNet121, and MobileNet2, to represent a spectrum of architectures commonly used in real-world applications.
3. Extending the evaluation beyond traditional accuracy assessments to quantify the degree of perturbation required to alter each model's output label, leading to misclassification.
4. Meticulously curating a 10-animal test dataset to provide a controlled environment for scrutinizing model responses to both pristine and perturbed inputs.
5. Unraveling the intricacies of model robustness in the face of adversarial challenges, contributing insights to the dynamic landscape of deep learning and adversarial attacks.

2. Related works

[1] In "Adversarial Attacks and Defenses", a research group led by Anirban Chakraborty at the Indian Institute of Technology, Kharagpur (2018) observed that Support Vector Machines (SVMs) are supervised learning models capable of constructing a hyperplane or a set of hyperplanes in high-dimensional space. SVMs can be employed for classification, regression, or outlier detection. The paper also discusses Artificial Neural Networks (ANNs), encompassing both supervised models like Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs), as well as unsupervised network models and their associated learning rules. The study explores the vulnerability of machine learning models to adversarial attacks, presenting simple yet effective attacks on popular model classes such as logistic regression, neural networks, and decision trees. Notably, the introduction of substitute model learning is highlighted as a method to alleviate the need for attackers to infer architecture, learning models, and parameters in typical black-box attacks.

[2] Machine learning models, particularly neural networks, face susceptibility to adversarial examples owing to their inherently linear nature, as quantified by recent findings. Goodfellow and collaborators (2014) extensively explored this vulnerability, revealing that state-of-the-art neural network, among other models, exhibit susceptibility to adversarial perturbations. Notably, adversarial examples align closely with model weight vectors, indicating shared learning functions among diverse models. The effectiveness of adversarial training in enhancing model robustness, akin to a regularization technique exceeding dropout, was highlighted. However, Radial Basis Function (RBF) networks demonstrated resistance to adversarial and irrelevant class examples. These findings unveil inherent blind spots in training algorithms and underscore the linearity of models, prompting questions about their true understanding of assigned tasks. Goodfellow emphasized the need for optimization procedures fostering local stability in model behavior. This study consolidates prior research, affirming that adversarial training not only generates rapid adversarial examples but also provides additional regularization benefits, contributing to a nuanced understanding of the intricacies in adversarial vulnerability within machine learning models.

[3] The article discusses various machine learning (ML) techniques used in Intrusion Detection Systems (IDS), focusing on Deep Neural Networks (DNN), Support Vector Machines (SVM), and Generative Adversarial Networks (GAN). It highlights the role of DNN in understanding complex cyber-attacks and the efficiency of SVM with small datasets, despite their sensitivity to noise. GANs are noted for data augmentation in attack detection. Adversarial Machine Learning (AML) is explored, where adversaries create inputs to mislead ML models, emphasizing the importance of adversarial samples and attack techniques. The paper also covers adversarial game-theoretic and threat models, detailing adversaries' capabilities, challenges, and potential threats. Benchmark datasets for IDS and defense strategies against adversarial attacks are also mentioned.

3. Selected Pre-Trained Model Architecture

Understanding the architectural intricacies of pre-trained models is crucial for comprehending their vulnerabilities to adversarial attacks. We meticulously analyze six prominent pre-trained deep learning models—Xception, ResNet152v2, Inception, InceptionResNetV2, DenseNet169, and MobileNetV2. Each model brings unique characteristics and architectural nuances that contribute to its overall performance. The following sections provide a detailed overview of the architecture for each selected pre-trained model:

3.1 Xception:

Xception, a creation of Google, distinguishes itself with depth-wise separable convolutions, striking a balance between model complexity and computational efficiency. The architecture draws inspiration from an extreme version of an Inception module.

Entry Flow:

- Initial Convolution Block: Convolutional layer with batch normalization and ReLU activation.
- Entry Flow Modules: Series of depth wise separable convolutions with skip connections.

Middle Flow:

- Repeated Middle Flow Modules: A stack of depth wise separable convolutions with skip connections.

Exit Flow:

- Exit Flow Modules: Combination of depth wise separable convolutions and global average pooling.
- Fully Connected Layer: Produces the final predictions.

3.2 ResNet152v2:

ResNet152v2, part of the ResNet family, prioritizes residual connections to address the vanishing gradient problem. Boasting 152 layers, it comprises specific components for optimal performance.

Initial Convolution Block:

- Convolutional layer with batch normalization and ReLU activation.
- Residual Blocks: Multiple blocks with identity and projection shortcuts.
- Bottleneck architecture: 1x1, 3x3, and 1x1 convolutions within each block.
- Final Fully Connected Layer: Global average pooling followed by a dense layer for predictions.

3.3 Inception:

Developed by Google, Inception leverages factorized convolutions and a multi-branch architecture for efficient learning of spatial hierarchies.

Inception Blocks:

- Multiple blocks with parallel convolutional branches of different kernel sizes.
- Efficiently captures features at different scales.
- Pooling Layers: Max pooling and average pooling layers for down-sampling.
- Fully Connected Layer: Global average pooling followed by a dense layer for predictions.

3.4 InceptionResNetV2:

An extension of Inception, InceptionResNetV2 integrates residual connections to enhance feature propagation, combining the strengths of Inception and ResNet architectures.

Inception Blocks:

- Similar to Inception but with additional residual connections.
- Reduction Blocks: Introduces additional 1x1 convolutions to reduce spatial dimensions.
- Final Fully Connected Layer: Global average pooling followed by a dense layer for predictions.

3.5 DenseNet169:

DenseNet169 adopts the DenseNet architecture, emphasizing dense connectivity patterns between layers, promoting feature reuse and efficient gradient flow.

Initial Convolution Block:

- Convolutional layer with batch normalization and ReLU activation.

Dense Blocks:

- Dense connectivity between layers, ensuring direct connections from each layer to every subsequent layer within a block.
- Consists of densely connected blocks with bottleneck structures.

Transition Blocks:

- 1x1 convolution and compression through pooling to manage feature map size.

Final Fully Connected Layer:

- Global average pooling followed by a dense layer for predictions.

3.6 MobileNetV2:

Designed for mobile and edge computing, MobileNetV2 prioritizes lightweight architectures without compromising performance, utilizing depth-wise separable convolutions and linear bottlenecks.

Initial Convolution Block:

- Convolutional layer with batch normalization and ReLU activation.

Inverted Residual Blocks:

- Lightweight blocks with depth-wise separable convolutions and linear bottlenecks.

Final Fully Connected Layer:

- Global average pooling followed by a dense layer for prediction

4. Methodology

4.1 Data Collection and Pre-Processing

We meticulously crafted a custom set of testing images to rigorously evaluate the models' performance. This set is a subset of the Animals-10 dataset [8], which itself is a subset of the larger ImageNet [9] classes. The rationale behind this customization is threefold:

- Close Animal Grouping: The selected animals represent a group with fine-grained distinctions, challenging the models to accurately predict subtle differences between closely related classes.

- Challenging Predictions: By choosing animals in proximity, we aim to test the models on a more intricate task where the margin of error is smaller, demanding a higher level of precision in label prediction.

- Enhancing FGSM Efficacy: The efficacy of the Fast Gradient Sign Method (FGSM) becomes more apparent when visualized using these carefully chosen images and labels.

We preprocessed all images using the following steps:

- Normalization: All images were normalized to the range of [-1, 1] to align with the input requirements of the models.

- Image Resizing: To maintain uniformity, all images were resized to a standard size. While most images were resized to 224 x 224 x 3, the subset with a larger size was adjusted to 299 x 299 x 3. This variation in image size allows us to investigate the impact of spatial information granularity on model performance.

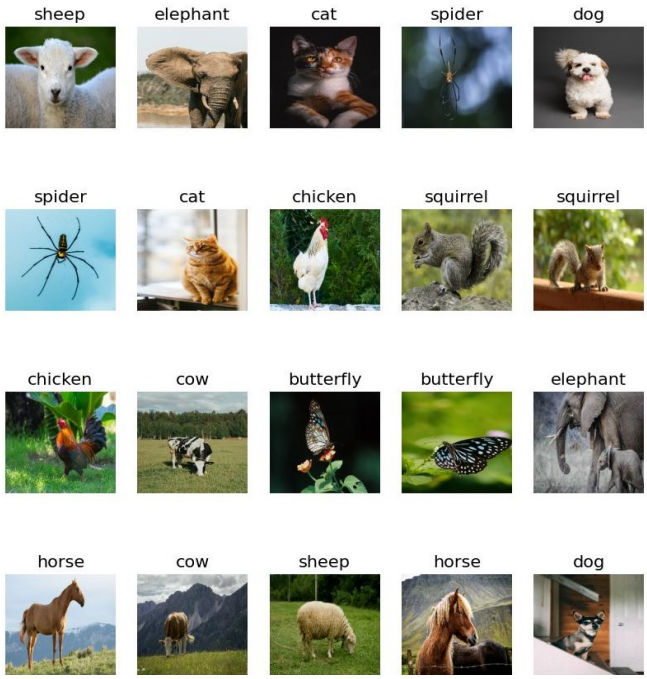

Fig -1: Test Set of 10 Images

4.2 Model Training

4.2.1 Motivation for Model Building

The primary objective in training models from scratch was to assess the efficacy of the Fast Gradient Sign Method (FGSM) across different architectures. To achieve this, we initially constructed a set of basic classifiers with varying architectures. This preliminary phase allowed us to establish the feasibility of applying FGSM on larger pre-trained models.

The proposed model follows a CNN (Convolutional Neural Network) architecture. The model is implemented using the TensorFlow Keras API. The model's architecture comprises an input layer with dimensions (224, 224, 3), followed by convolutional layers and max-pooling layers to capture hierarchical features and reduce spatial dimensions. The final layers include a flatten layer to reshape the output into a one-dimensional array and a dense layer for classification. The model is optimized using the Adam optimizer with categorical cross entropy as the loss function. The training duration is carefully chosen to ensure model convergence.

4.2.2 Transition to Pre-Trained Models

Upon confirming the suitability of FGSM through scratch-built models, we transitioned to larger, pre-trained architectures. The models selected for evaluation were Xception, ResNet152v2, Inception, InceptionResNetV2, DenseNet169, and MobileNetV2. This transition allowed us to explore FGSM across diverse complexities, providing valuable insights into model robustness and vulnerabilities.

4.3 Adversarial Pattern Generation

Adversarial Pattern Generation is a crucial component of our methodology, focusing on creating perturbations in input data to mislead neural network models. We employ the Fast Gradient Sign Method (FGSM), a well-established technique in adversarial machine learning.

Fast Gradient Sign Method (FGSM):

FGSM operates by perturbing input data based on the gradient information of the loss function with respect to the input. The perturbation is calculated to maximize the loss, leading to misclassifications. Mathematically, the perturbed image, $X(\text{adv})$, is generated as follows:

$$X(\text{adv}) = X + \epsilon \cdot \text{sign}\left(\nabla X J\left(X, Y(true)\right)\right)$$

Here:

- X represents the clean input image.
- $Y(\text{true})$ is the true label of the clean image.
- $J(X, Y(\text{true}))$ is the loss function based on the true label.
- ∇X denotes the gradient with respect to the input.
- ϵ controls the magnitude of perturbation.

The sign function ensures that the perturbation is added in the direction that increases the loss, aiming to induce misclassification. By adjusting ϵ, we control the strength of the attack. Smaller ϵ values result in subtle perturbations, while larger values lead to more pronounced changes.

This process is applied to each pixel in the input image, generating an adversarial image that, when fed into the neural network, is likely to be misclassified. The efficacy of FGSM lies in its simplicity and efficiency, making it a valuable tool for evaluating model robustness against adversarial attacks.

4.4 Pre-trained Model Evaluation

The Pre-trained Model Evaluation phase is a critical step in our methodology, where we assess the susceptibility of widely adopted pre-trained models to adversarial attacks using the Fast Gradient Sign Method (FGSM). We select six prominent pre-trained models for evaluation: Xception, ResNet152v2, Inception, InceptionResNetV2, DenseNet169, and MobileNetV2.

4.4.1 Model Initialization and Prediction on Clean Images:

Initially, each pre-trained model is initialized, and predictions are made on the clean images from our curated 10-animal test dataset. This step establishes a baseline for the models' performance on pristine inputs.

4.4.2 Tabulation and Analysis:

Results from the predictions on clean images are tabulated, including confidence levels and correct classifications. This tabulation provides insights into the models' initial accuracy and their behavior on the unaltered dataset.

4.4.3 Adversarial Generation using FGSM:

Next, we apply the FGSM algorithm to generate adversarial images for each model. The previously calculated gradients are utilized to perturb the input images strategically. The perturbed images, known as adversarial examples, are then created for each class in the dataset.

4.4.4 Testing on Perturbed Images and Comparison with Clean Ones:

The models are subjected to predictions on both perturbed and clean images. The predictions on adversarial examples help us evaluate the models' vulnerability to adversarial attacks. We compare these results with predictions on clean images to gauge the impact of adversarial perturbations.

4.4.5 Tabulation of Results, Including Epsilon Values:

Results from the evaluation, including confidence levels, misclassification rates, and epsilon values used for perturbations, are tabulated comprehensively. Epsilon values play a crucial role in understanding the intensity of the applied perturbations. This detailed tabulation facilitates a nuanced analysis of each model's robustness and provides a basis for comparison across different architectures.

This thorough evaluation process allows us to discern the resilience and vulnerabilities of pre-trained models under the FGSM attack, contributing valuable insights to the broader discourse on adversarial attacks in deep learning.

5. Results and Discussion

In this section, we meticulously detail the outcomes of our experiments, offering an in-depth examination of adversarial attacks on six prominent pre-trained deep learning models—Xception, ResNet152v2, Inception, InceptionResNetV2, DenseNet169, and MobileNetV2. Leveraging our carefully curated 10-animal test dataset, we systematically assessed the vulnerabilities of each model under the Fast Gradient Sign Method (FGSM) attack.

(I) Xception Model

- Image Size: Utilizing a larger image size of 299 x 299 x 3.
- Epsilon Values: Predominantly low epsilon values to misclassify.
- Adversarial Confidence: Consistently exhibited confidence levels below 50%, indicating successful but low confidence misclassifications.
- Clean Input Confidence: Clean inputs displayed a mixed range of confidence levels which may explain the low epsilon value required to misclassify and low confidence of adversarial outputs.

Table-1: Xception Results

Name	Xception (Output and confidence)	Epsilon	Adversarial (Output and confidence)
Butterfly 1	Admiral: 44.27%	0.0132	Monarch: 36.92%
Butterfly 2	Monarch: 52.21%	0.0068	Lycaenid: 88.84%
Cat 1	Tiger cat: 41.97%	0.0075	Egyptian cat: 64.16%
Cat 2	Tiger cat: 50.24%	0.0475	Tabby: 31.53%
Chicken 1	Cock: 89.78%	0.0052	Hen: 56.44%
Chicken 2	Cock: 82.63%	0.0018	Hen: 42.35%
Cow 1	Ram: 23.76%	0.0001	Ox: 21.32%
Cow 2	Ox: 48.57%	0.0007	Plow: 34.36%
Dog 1	Lhasa: 88.00%	0.0023	Shih-Tzu: 38.75%
Dog 2	Chihuahua: 9.12%	0.0012	German Shepherd: 4.77%
Elephant 1	African elephant: 40.8%	0.0002	Tusker: 34.43%
Elephant 2	African elephant: 32.51%	0.0002	Tusker: 32.27%
Horse 1	Sorrel: 69.38%	0.0039	Hartebeest: 8.36%
Horse 2	Sorrel: 92.67%	0.0062	Bighorn: 7.33%
Sheep 1	Ram: 67.73%	0.0042	Ice bear: 3.70%
Sheep 2	Ram: 93.45%	0.0109	Airedale: 4.29%
Spider 1	Garden spider: 24.72%	0.0012	Barn spider: 15.05%
Spider 2	Harvestman: 39.94%	0.9299	Black and gold garden spider: 29.84%
Squirrel 1	Fox squirrel: 93.73%	0.0182	Mongoose: 7.62%
Squirrel 2	Fox squirrel: 94.64%	0.0312	Wood rabbit: 3.09%

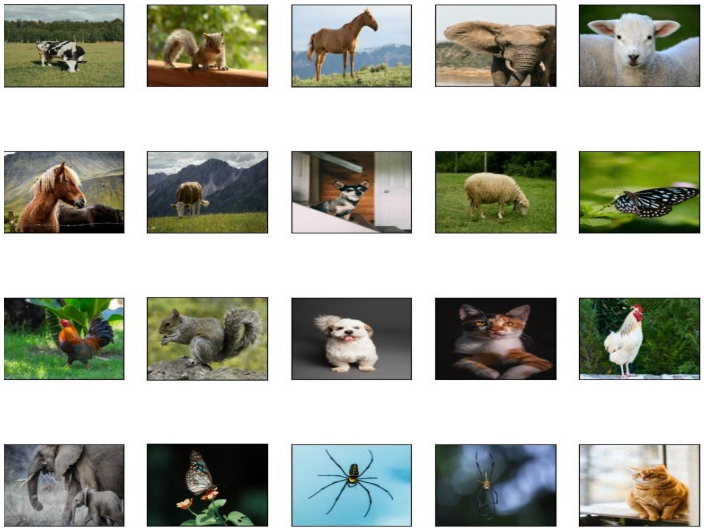

Fig -2: Perturbed images of Xception Model

(II) ResNet152v2 Model

- Image Size: Slightly smaller image size of 224 x 224 x 3.
- Epsilon Values: Low epsilon values dominated for misclassification.
- Adversarial Confidence: Frequently achieved over 50% confidence for adversarial outputs, showing a notable weakness to perturbations.
- Clean Input Confidence: Clean inputs consistently demonstrated high confidence levels.

Table-2: Resnet152v2 Results

Name	Resnet152v2 (Output and confidence)	Epsilon	Adversarial (Output and confidence)
Butterfly 1	Lycaenid: 99.73%	0.2658	Flatworm: 65.78%
Butterfly 2	Monarch: 99.99%	0.0240	Limpkin: 61.8%
Cat 1	Egyptian Cat: 55.66%	0.0089	Tabby: 83.54%
Cat 2	Tabby: 62.71%	0.0094	Tiger cat: 93.28%
chicken 1	Cock: 99.76%	0.0065	Hen: 53.63%
chicken 2	Cock: 94.32%	0.0079	Hen: 97.30%
cow 1	Llama: 62.78%	0.0058	Ostrich: 98.17%
Cow 2	Ox: 72.09%	0.0042	Dalmatian: 97.93%
Dog 1	Shih-Tzu: 61.43%	0.0039	Lhasa: 98.24%
Dog 2	Miniature pinscher: 92.77%	0.0048	Chihuahua: 99.10%
Elephant 1	African elephant: 79.70%	0.0052	Tusker: 79.25%
Elephant 2	African elephant: 85.76%	0.0069	Tusker: 94.53%
Horse 1	Sorrel: 100%	0.0079	Hartebeest: 82.00%
Horse 2	Sorrel: 99.88%	0.0798	Ibex: 54.84%
Sheep 1	Ram: 98.41%	0.0081	Hog: 99.72%
Sheep 2	Ram: 100%	0.2595	Komondor: 60.14%
Spider 1	Black and gold garden spider: 80.3%	0.1898	Ant: 53.65%
Spider 2	Black and gold garden spider: 67.23%	0.1288	Harvestman: 67.12%
Squirrel 1	Fox squirrel: 94.01%	0.0028	Mongoose: 99.65%
Squirrel 2	Fox squirrel: 100%	0.4988	Porcupine: 21.93%

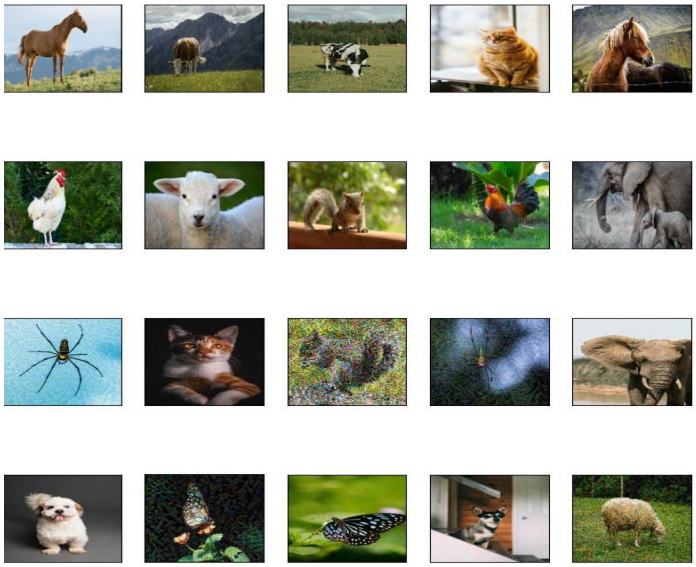

Fig -3: Perturbed images of ResNet152v2 Model

(III) InceptionV3 Model

- Image Size: Larger image size of 299 x 299 x 3.
- Epsilon Values: A preference for mostly high epsilon values (>0.1).
- Adversarial Confidence: The adversarial outputs predominantly registered confidence levels below 50%.
- Clean Input Confidence: In contrast, clean inputs consistently exhibited mixed to high confidence.

Table-3: InceptionV3 Results

Name	InceptionV3 (Output and confidence)	Epsilon	Adversarial (Output and confidence)
Butterfly 1	Monarch: 45.55%	0.2099	Mask: 4.56%

Butterfly 2	Monarch: 90.01%	0.2682	Lycaenid: 27.92%
Cat 1	Egyptian cat: 57.30%	0.0437	Tabby: 32.59%
Cat 2	Tiger cat: 53.30%	0.2564	Tabby: 44.63%
Chicken 1	Cock: 83.97%	0.5021	Jigsaw puzzle: 12.21%
Chicken 2	Cock: 93.68%	0.6590	Plastic bag: 6.55%
Cow 1	Ox: 33.55%	0.0032	Ram: 8.24%
Cow 2	Ox: 90.06%	0.2830	Plow: 39.41%
Dog 1	Lhasa: 79.95%	0.0972	Maltese dog: 39.38%
Dog 2	Chihuahua: 44.04%	0.0037	Egyptian cat: 17.34%
Elephant 1	African elephant: 54.80%	0.3717	Tusker: 13.01%
Elephant 2	African elephant: 37.89%	0.4562	Oxygen mask: 5.93%
Horse 1	Sorrel: 78.02%	0.2388	Airedale: 7.86%
Horse 2	Sorrel: 81.94%	0.3911	Cougar: 21.29%
Sheep 1	Ram: 50.66%	0.3658	Sealyham terrier: 5.76%
Sheep 2	Ram: 89.05%	0.2089	Water buffalo: 39.20%
Spider 1	Barn spider: 53.65%	0.0043	Garden spider: 41.03%
Spider 2	Harvestman: 84.08%	0.1265	Barn spider: 35.78%
Squirrel 1	Fox squirrel: 88.51%	0.3239	Grey fox: 22.77%
Squirrel 2	Fox squirrel: 94.73%	0.4688	Wombat: 32.47%

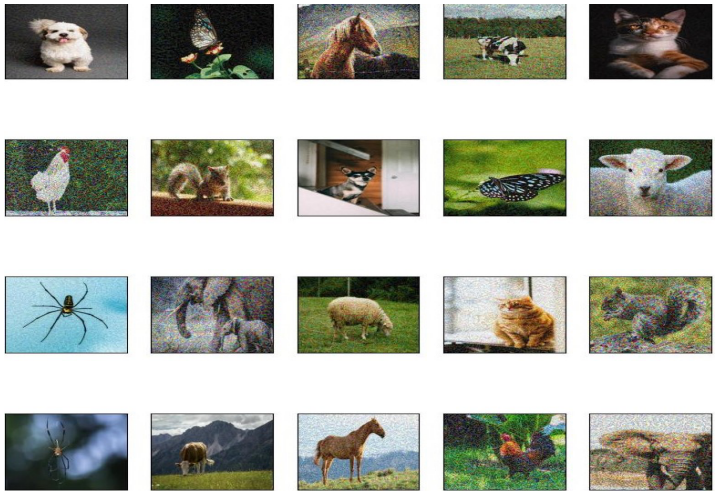

Fig-4: Perturbed images of InceptionV3 model

(IV) InceptionResNetV2 Model

- Image Size: Operating on the same larger image size of 299 x 299 x 3, InceptionResNetV2 showcased a deliberate exploration of perturbation space.
- Epsilon Values: A tendency to utilize higher epsilon values (>0.05) was noted, implying a deliberate exploration of perturbation space.
- Adversarial Confidence: The model primarily showcased confidence levels below 50% for adversarial outputs, indicative of successful misclassifications.
- Clean Input Confidence: Clean inputs consistently exhibited high confidence, showcasing the model's resilience under normal conditions.

Table-4: InceptionResNetV2 Results

Name	**InceptionResNetV2 (Output and confidence)**	**Epsilon**	**Adversarial (Output and confidence)**
Butterfly 1	Lycaenid: 71.71%	0.0101	monarch 26.72%
Butterfly 2	Monarch: 80.37%	0.3853	puffer 12.49%
Cat 1	Egyptian Cat: 55.26%	0.0272	tabby 35.33%
Cat 2	tiger cat: 43.28%	0.0760	tabby 29.40%
chicken 1	Cock: 92.40%	0.7364	Goldfish: 7.57%

chicken 2	Cock: 86.61%	0.9087	Hen: 17.73%
cow 1	Ox: 70.80%	0.0132	water buffalo: 46.16%
cow2	Ox: 92.55%	0.0885	Plow: 35.58%
dog 1	Lhasa: 56.52%	0.0958	Maltese dog: 31.23%
dog 2	Chihuahua: 87.34%	0.2923	Miniature pinscher: 46.57%
elephant 1	African elephant: 78.50%	0.2573	Tusker: 44.91%
elephant 2	African elephant: 62.37%	0.3221	Tusker: 44.47%
horse 1	Sorrel: 92.32%	0.0584	Hartebeest: 39.66%
horse 2	Sorrel: 90.94%	0.0553	Basenji: 5.92%
sheep 1	Ram: 86.40%	0.5832	Hog: 31.57%
sheep 2	Ram: 90.88%	0.3273	Armadillo: 39.76%
spider 1	Garden spider: 29.96%	0.0095	Black widow: 21.75%
spider 2	Harvestman: 89.01%	0.3939	Garden spider: 20.54%
squirrel 1	Fox squirrel: 89.42%	0.5089	Grey fox: 33.67%
Squirrel 2	Fox squirrel: 94.45%	0.3694	Mongoose: 11.66%

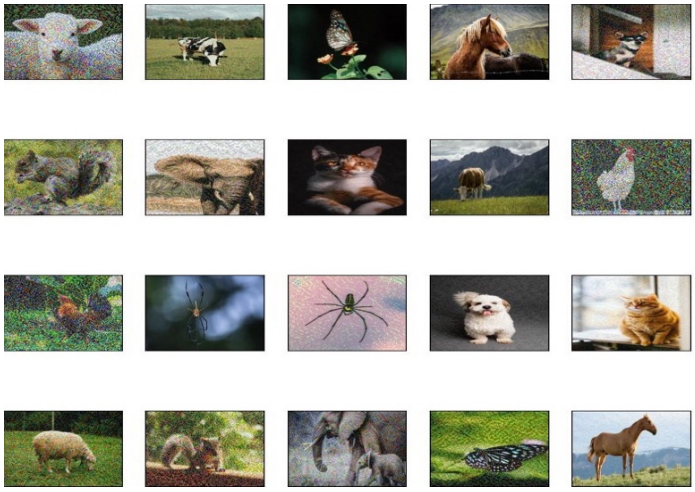

Fig-5: Perturbed Images of InceptionResNetV2 model

(V) DenseNet169 Model

- Image Size: Smaller image size of 224 x 224 x 3.
- Epsilon Values: Low epsilon values were observed.
- Adversarial Confidence: Mostly registered confidence levels below 50%.
- Clean Input Confidence: Displayed a mixed range of confidence levels.

Table-5: DenseNet169 Results

Name	**Densenet169 (Output and confidence)**	**Epsilon**	**Adversarial (Output and confidence)**
Butterfly 1	Lycaenid: 33.29%	0.0001	Monarch:26.95%
Butterfly 2	Monarch: 86.55%	0.0249	Lycaenid: 40.66%
Cat 1	Tiger cat: 39.02%	0.0134	Egyptian cat:30.02%
Cat 2	Tiger cat: 54.85%	0.0014	Screen: 17.26%
Chicken 1	Cock: 93.21%	0.0039	Hen: 50.15%
Chicken 2	Cock: 98.65%	0.0031	Hen: 51.10%
Cow 1	Ram: 42.49%	0.0012	Bighorn: 19.3%

Cow 2	Dalmatian: 53.29%	0.0003	Ox:42.61%
Dog 1	Lhasa: 64.37%	0.0005	Shih-Tzu: 44.5%
Dog 2	Cardigan:25.32%	0.0001	Miniature_pinscher: 26.15%
Elephant 1	African elephant:49.76%	0.0004	Tusker:41.27%
Elephant 2	African elephant:73.92%	0.0018	Tusker:41.42%
Horse 1	Sorrel:56.41%	0.0003	Hartebeest:43.01%
Horse 2	Sorrel:43.92%	0.0006	Ram:26.73%
Sheep 1	Ram:91.75%	0.0014	Wallaby:46.23%
Sheep 2	Ram:73.58%	0.0022	Komondor:26.84%
Spider 1	Garden spider:51.34%	0.0011	Black and gold garden spider:31.1%
Spider 2	Harvestman:43.57%	0.0003	Black and gold garden spider:30.96%
Squirrel 1	Fox squirrel: 92.61%	0.0017	Mongoose:48.48%
Squirrel 2	Fox squirrel: 99.79%	0.0068	Marmot:46.23%

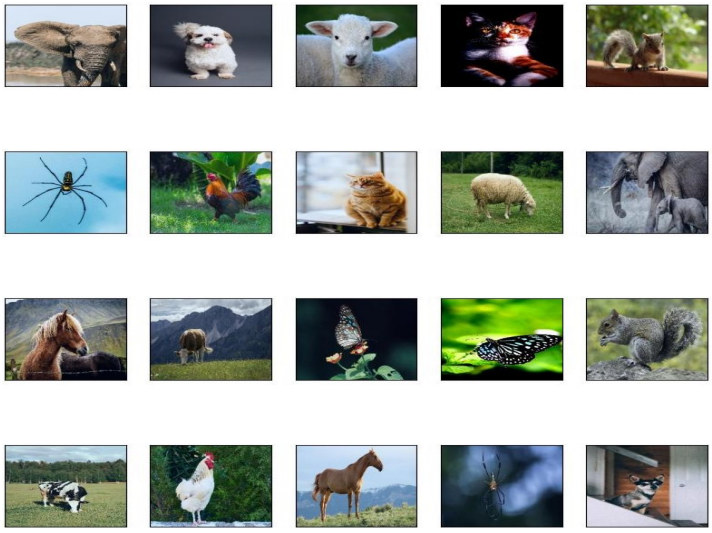

Fig-6: Perturbed Images of DenseNet169 model

(VI) MobileNetv2 Model

- Image Size: Smaller image size of 224 x 224 x 3.
- Epsilon Values: Mostly low epsilon values.
- Adversarial Confidence: The adversarial outputs consistently exhibited confidence levels below 50%.
- Clean Input Confidence: Clean inputs showcased a mixed range of confidence levels.

Table-6: MobileNetV2 Results

Name	**MobilenetV2 (Output and confidence)**	**Epsilon**	**Adversarial (Output and confidence)**
Butterfly 1	Lycaenid: 86.27%	0.0011	Admiral :29.7%
Butterfly 2	Lycaenid:19.19%	0.0003	Monarch :14.18%
Cat 1	Tiger cat :25.11%	0.0003	Egyptian cat :24.09%
Cat 2	Tiger cat :76.59%	0.0023	Laptop :16.19%
Chicken 1	Cock :81.73%	0.0065	Hen :13.04%
Chicken 2	Cock :72.72%	0.0008	Hen :49.44%
Cow 1	Ram :34.08%	0.0002	Ox :25.61%

Cow 2	Ox :20.3%	0.0003	Dalmatian :15.72%
Dog 1	Shih-tzu :70.03%	0.0012	Lhasa :29.56%
Dog 2	Egyptian cat :22.19%	0.0003	Carton :11.97%
Elephant 1	African elephant :79.3%	0.0009	Tusker :45.96%
Elephant 2	African elephant :70.03%	0.0012	Tusker :25.61%
Horse 1	Sorrel :38.39%	0.0006	Saluki :13.15%
Horse 2	Sorrel :38.43%	0.0008	brown bear:15.81%
Sheep 1	Ram :62.41%	0.0014	Hog :8.59%
Sheep 2	Ram :97.07%	0.0099	Hay :12.88%
Spider 1	Garden spider :45.12%	0.0007	Black and gold spider :28.03%
Spider 2	Garden spider :49.43%	0.0004	Black and gold spider :40.46%
Squirrel 1	Fox squirrel :80.7%	0.0038	Egyptian cat :7.67%
Squirrel 2	Fox squirrel :94.78%	0.00399	grey fox :8.49%

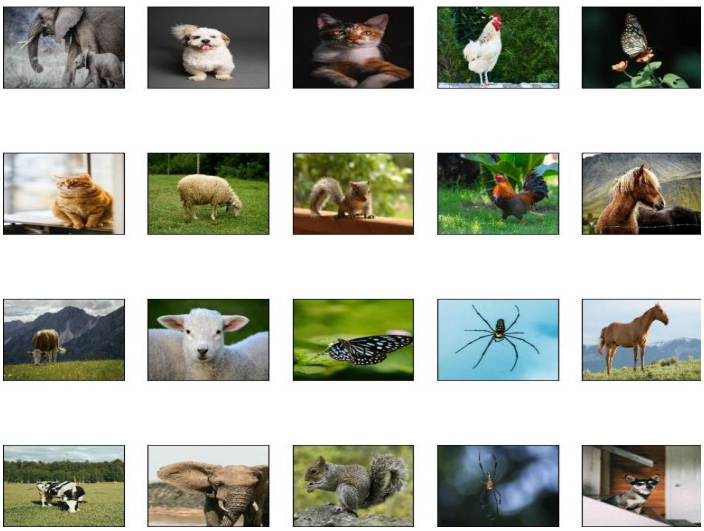

Fig-7: Perturbed images of MobileNetV2 model

Table-7: Model Comparison

Model	Image Size	Epsilon Values	Adversarial Confidence (in %)	Clean Input Confidence
Xception	299 x 299 x 3	Low	<50%	Mixed
ResNet152v2	224 x 224 x 3	Low	>50%	Mixed
Inception	299 x 299 x 3	High	<50%	High
Inception ResNetv2	299 x 299 x 3	High	<50%	High
DenseNet 169	224 x 224 x 3	Low	<50%	Mixed
MobileNetv2	224 x 224 x 3	Low	<50%	Mixed

6. Conclusion and Future Scope

In the realm of deep learning, our investigation into adversarial vulnerabilities within pre-trained models has revealed compelling insights. The scrutiny of six prominent models—Xception, ResNet152v2, Inception, InceptionResNetV2, DenseNet169, and MobileNetV2—under the Fast Gradient Sign Method (FGSM) attack has provided a nuanced understanding of their response to adversarial perturbations. Noteworthy findings include-

- The pre-trained models are susceptible to adversarial attacks such as FGSM.
- Some models notably Inception and InceptionResNetv2 demonstrated more resistance to FGSM as evident by the high epsilon values required to misclassify.
- Some models notably Xception, DenseNet169 and MobileNetv2 misclassified at low epsilon values but their degree of classification i.e. the output confidence was notably lower.
- The ResNet152v2 started misclassifying at low epsilon values with high confidence.

- The models that have high confidence for clean images resist FGSM to a greater degree than those who have low confidence for clean images.
- It is also seen that the model which takes the larger image size as input resists the FGSM to a greater degree than those who take smaller image size as input. This may be because a larger size image has much more pixels for the model to consider while making an inference and thus require a greater amount of change in the image to misclassify.

This study extends beyond conventional accuracy assessments, offering a detailed analysis of perturbation impacts on model outputs. The inclusion of epsilon values and confidence levels enhances the depth of our evaluation, setting a valuable benchmark for future research in adversarial robustness. By shedding light on the intricate interplay between model architecture and adversarial attacks, our work contributes to a more holistic understanding of model vulnerabilities.

As the field of adversarial machine learning continues to evolve, several promising avenues beckon for future exploration. The investigation of adversarial example transferability across diverse models and architectures holds the potential to unveil broader patterns in vulnerability. Real-world applications, particularly in domains like healthcare and autonomous systems, present intriguing challenges for mitigating adversarial impact. Future research endeavors may delve into the integration of defense mechanisms and robust training strategies to fortify models against adversarial threats. This study serves as a stepping stone, urging the research community to delve deeper into fortifying artificial intelligence against the ever-evolving landscape of adversarial challenges.

REFERENCES

[1] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., & Mukhopadhyay, D. (2018). Adversarial Attacks and Defences: A Survey. arXiv e-prints. doi:10.48550/arXiv.1810.00069.

[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. arXiv e-prints. doi:10.48550/arXiv.1412.6572.

[3] Alotaibi A, Rassam MA. (2023). Adversarial Machine Learning Attacks against Intrusion Detection Systems: A Survey on Strategies and Defense. Future Internet. 2023; 15(2):62. doi:10.3390/fi15020062

[4] Song, Y., Kim, T., Nowozin, S., Ermon, S., & Kushman, N. (2018). PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. In Advances in Neural Information Processing Systems (NeurIPS).https://arxiv.org/abs/1710.10766

[5] Wu, J., Zhou, M., Zhu, C., Liu, Y., Harandi, M., & Li, L. (2021). Performance Evaluation of Adversarial Attacks: Discrepancies and Solutions. arXiv e-prints. doi:10.48550/arXiv.2104.11103.

[6] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. arXiv e-prints. doi:10.48550/arXiv.1312.6199.

[7] Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP) (pp. 39-57).

[8] https://www.kaggle.com/datasets/alessiocorrado99/

animals10/ data

[9] https://www.image-net.org/