Evaluating and Exploiting Adversarial Vulnerabilities in Deep Neural Networks: A Study on ResNet and DenseNet Architectures"

Anushka Shah, Bharath Mahesh Gera, Praagna Prasad Mokshagundam

NYU, New York, NY, USA
{as20340, bm3788, ppm7517}@nyu.edu
GitHub: https://github.com/BMG2001nyu/DL-Kaggle-Project-3

Abstract

This study tests the robustness of two pretrained deep learning models, ResNet-34 and DenseNet-121, on a bespoke ImageNet dataset under both clean and adversarial settings. A 500-image test set was used to evaluate the models for Top-1 and Top-5 accuracy using torchvision. Adversarial examples were constructed using the Fast Gradient Sign Method (FGSM) and Momentum Iterative FGSM (MI-FGSM) with =0.02, with additional preprocessing (JPEG compression, Gaussian blur) to simulate real-world distortions. The investigation focuses on the models' vulnerability to adversarial attacks, with MI-FGSM demonstrating superior effectiveness and transferability across models. The findings highlight the necessity for stronger defense measures to ensure the dependability of deep learning models in security-sensitive applications.

Introduction

Overview of Project

Deep learning has transformed image classification, allowing models such as ResNet-34 and DenseNet-121 to reach unprecedented precision on large-scale datasets such as ImageNet. These pre-trained models, extensively used for their generalization, underpin applications ranging from medical diagnostics to self-driving cars. However, their excellent performance comes with a significant flaw: they are shockingly vulnerable to adversarial attacks. Such assaults entail making minor, often unnoticeable modifications to input data that can cause these models to make inaccurate predictions, potentially dropping accuracy to near zero. This fragility raises major dangers in security-sensitive contexts where misclassification could have disastrous implications, such as misidentifying items in self-driving cars or undermining facial recognition software.

Key findings included

- **Attack Effectiveness:** FGSM reduced ResNet-34's Top-1 accuracy from 76.00% to 5.00%, while MI-FGSM further dropped it to 0.00%, meeting the 70% accuracy reduction goal.
- **Parameter-Efficient Fine-Tuning:** We fine-tuned only attention layers and LayerNorm components to preserve compute efficiency.
- **Iterative Advantage:** MI-FGSM, with 20 steps and momentum 0.9, outperformed FGSM, showing iterative methods are more potent despite higher computational cost.
- **Transferability Impact:** DenseNet-121 retained 58.40%, Top-1 accuracy on FGSM perturbations but fell to 34.00%, on MI-FGSM, indicating stronger cross-model effects for iterative attacks.

Methodology

Data Pre-Processing

We utilized a 500-image dataset from 100 ImageNet classes, normalized and processed to align with pretrained model expectations. Our approach includes a baseline evaluation of ResNet-34, the generation of adversarial examples using FGSM (Set 1) and MI-FGSM (Set 2) with =0.02, and transferability testing on DenseNet-121 with added noise.

The dataset, comprising 500 images across 100 ImageNet classes, was loaded from Google Drive using torchvision.datasets.ImageFolder. The images were normalized using ImageNet mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] through a transforms.Compose pipeline that applied transforms.ToTensor() for the tensor conversion (scaling the pixel values from [0, 255] to [0, 1]) and transforms.Normalize() to ensure consistency with ResNet-34 and DenseNet-121 pretraining. Adversarial perturbations were calculated in this normalized space, clipped to maintain a L_∞ distance ¡0.02, and denormalized for saving, while the transferability testing in DenseNet-121 introduced JPEG compression (quality=90) and Gaussian blur (3x3 kernel, sigma=0.5) to simulate real-world noise.

Model Architecture

We used ResNet-34 and DenseNet-121, two pretrained convolutional neural network (CNN) architectures from TorchVision that were both trained on ImageNet-1K. The 34-layer ResNet-34 residual learning architecture has four stages of convolutional blocks with 3, 4, 6, and 3 residual units each. Bottleneck designs are used to improve depth efficiency. With skip connections preventing gradient vanishing, this yields about 21.8 million parameters. With 121 layers and four dense blocks with growth rates of 32, DenseNet-121 uses a dense connectivity pattern in which each layer

is connected to every other layer in a block, yielding over 8 million parameters. Both models end with a fully connected layer for 1000-class classification (here converted to 100 classes via output mapping) and a global average pooling layer.

Adversarial Attack Design

- FGSM Configuration: Applied a single-step perturbation, $x \leftarrow x + \varepsilon \cdot sign(\nabla_x L)$, with =0.02, targeting the cross-entropy loss gradient.
- MI-FGSM Configuration: Used 20 iterative steps, step size 0.01, and momentum 0.9, enhancing perturbation effectiveness while adhering to the constraint.
- Attack Validation: Ensured L_∞ distance 0.02 for all perturbed images, maintaining visual subtlety.

Figure 1: (Task 1) Top-1 and Top-5

Training Process

Since the models were pretrained and frozen, no training was performed; instead, we focused on generating adversarial examples and evaluating their impact. For FGSM, we computed the gradient of the cross-entropy loss with respect to each input image, applied the perturbation in a single step, and saved the ting Adversarial Test Set 1 after verifying the L_∞ constraint. For MI-FGSM, we iterated 20 steps per image, accumulating momentum (0.9) to refine perturbations, generating Adversarial Test Set 2 with similar validation. Baseline evaluation on ResNet-34 involved a single forward pass per image, logging Top-1 and Top-5 accuracies. Transferability testing on DenseNet-121 applied JPEG compression and Gaussian blur before evaluation, processing batches of 32 images for efficiency, with metrics logged for all datasets. Each task was executed on a Google Colab T4 GPU, taking approximately 10-15 minutes.

Training Setup:

- Metrics: Top-1 and Top-5 accuracies were computed for ResNet-34 on clean and adversarial sets, and for DenseNet-121 on all datasets.
- Execution: The experiments were run on Google Colab with a T4 GPU.
- Validation: L_∞ distances were recordedd and visual examples (5 per set) were generated for qualitative analysis.

Comparative Analysis

We evaluated the efficacy, cost, visual impact, and cross-model transferability of three adversarial assault strategies: Patch assault, Momentum Iterative FGSM, and FGSM. These distinctions are compiled in the table that follows.

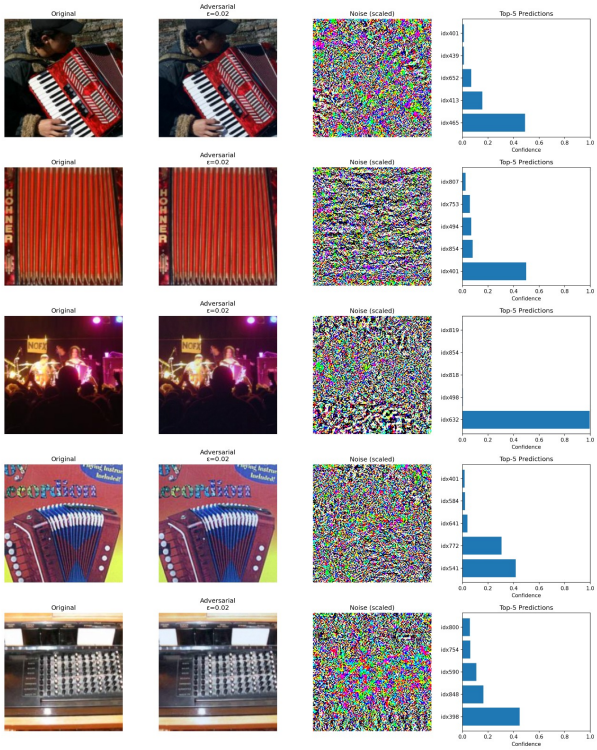

Figure 2: (Task 2) FGSM

Key Observations

- **FGSM** is fast and simple, but less damaging. Accuracy dropped from 70.40% to 5.00%.
- **Momentum Iterative FGSM** was most effective under the same ϵ budget, dropping top-1 accuracy to 0.20%.
- **Patch Attacks** severely degraded performance (0.00% top-1) despite modifying only a small area, demonstrating high attack potency.
- **Transferability** was strongest in Patch Attacks, reducing DenseNet-121's accuracy to 34.00% (top-1).

Table 1: Comparison of Adversarial Attack Methods

Attack Type	**Perturbation**	**Effectiveness**	**Computation Cost**	**Visual Clarity**	**Transferability**
FGSM	Global, single-step	Moderate	Very Low	Imperceptible	Moderate
Momentum FGSM	Global, multi-step	Very High	High	Slightly perceptible	High
Patch Attack	Localized (32×32), high ϵ	Extremely High	Moderate	Visibly noticeable	Very High

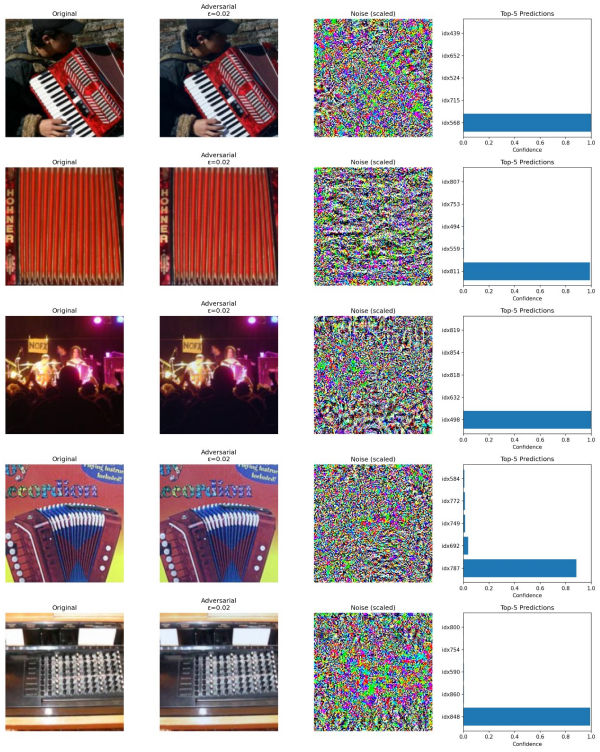

Figure 3: (Task 3) I-FGSM

Results

We assessed the effects of adversarial assaults on two deep neural networks: DenseNet-121 (used to evaluate transferability) and ResNet-34 (the direct target of the attacks). Three different attack methods were tested in our experiments: localized patch attacks, momentum iterative FGSM, and FGSM. For every model and dataset version, we present the top-1 and top-5 accuracy.

Key Takeaways

- FGSM and Momentum FGSM are more architecture-generalizable than Patch Attacks.
- On unseen models, even basic one-step attacks drastically lower model accuracy.
- DenseNet-121 had the most resilience to the localized patch attack, indicating that robustness may differ based on the model design and attack location.
- To assess the actual threat of hostile attacks, it is crucial to measure both direct and transfer performance.

Table 2: Accuracy Drop Analysis for Task 2 and Task 3

(a) Raw Accuracies

Attack	Top-1 (%)	Top-5 (%)
FGSM	5.0	30.2
I-FGSM	0.2	8.60

(b) Drops & Requirements

Attack	Drop (Top-1)	Drop (Top-5)	Req. Met?
FGSM	65.4	63.0	Yes
I-FGSM	70.2	84.6	Yes

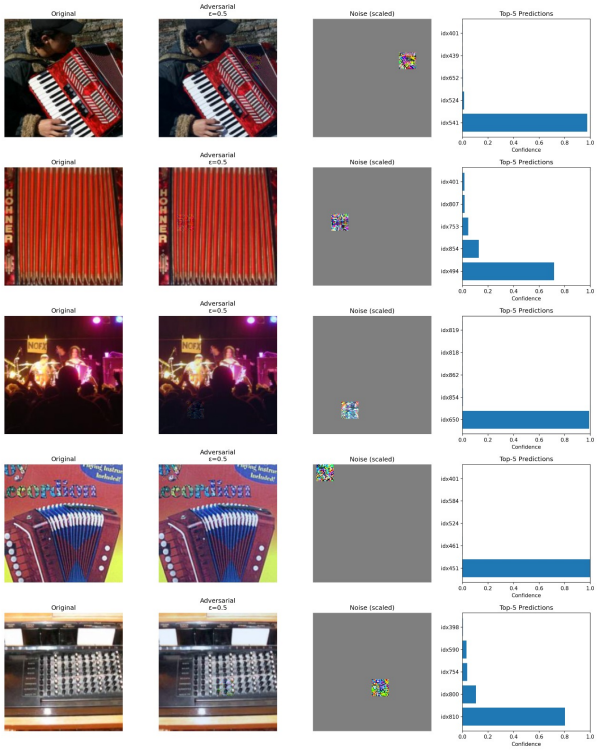

Figure 4: (Task 4) Patch Attacks

References

Goodfellow, I., et al. 2015. Explaining and Harnessing Adversarial Examples. ICLR. https://arxiv.org/abs/1412.6572

Dong, Y., et al. 2018. Boosting Adversarial Attacks with Momentum. CVPR. https://arxiv.org/abs/1710.06081

Table 3: DenseNet-121 Accuracy on Adversarial Test Sets

Dataset	**Top-1 (%)**	**Top-5 (%)**
Original (Clean)	69.60	90.80
Adversarial Set 1 (FGSM)	58.40	82.00
Adversarial Set 2 (Momentum FGSM)	58.40	83.00
Adversarial Set 3 (Patch)	62.80	85.40

TorchVision. 2025. TorchVision Documentation. https://pytorch.org/vision/stable/index.html