Jailbreaking Deep Models: Adversarial Evaluation of ResNet-34 on ImageNet

Nikhil Bhise[1], **Vishwas Karale**[2], **Hemanth Sree Meka**[3]

[1]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[2]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[3]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
email-id: nb4053@nyu.edu, vbk2750@nyu.edu, hm3324@nyu.edu
Github Repository : https://github.com/vishwaskarale83/deep-learning-project-3

Abstract

Adversarial attacks pose a significant threat to the reliability of deep learning models in computer vision. This paper investigates the robustness of a pretrained ResNet-34 model, trained on the ImageNet-1K dataset, under a series of adversarial attack scenarios. We implement and evaluate three types of attacks: the Fast Gradient Sign Method (FGSM), an iterative Projected Gradient Descent (PGD) approach, and a spatially constrained patch-based attack. Each attack is bounded by a perturbation budget in the L_∞ norm to ensure the resulting adversarial images remain visually similar to the originals. Additionally, we examine the transferability of adversarial examples to a different architecture, DenseNet-121, to assess model-agnostic vulnerability. Our experiments reveal that even imperceptible perturbations can cause significant drops in classification accuracy—up to a 70% reduction in top-1 accuracy—highlighting the urgent need for more robust model architectures and defense strategies in real-world applications.

Introduction

Deep neural networks have achieved state-of-the-art performance across various image classification tasks, but they remain vulnerable to adversarial examples—inputs with small, carefully crafted perturbations that mislead models into making incorrect predictions. This vulnerability poses serious security and reliability concerns for deploying these models in safety-critical applications.

In this project, we investigate the robustness of a pretrained ResNet-34 model against adversarial attacks using a subset of the ImageNet-1K dataset. We implement several attack strategies: a single-step Fast Gradient Sign Method (FGSM), a multi-step iterative attack (PGD), and a spatially constrained patch-based attack. Each attack is constrained by a perturbation budget to ensure that the perturbed images remain visually similar to the originals.

We report the drop in model accuracy under each attack and analyze the effectiveness of these adversarial strategies. Furthermore, we evaluate the transferability of adversarial examples across architectures by testing the same inputs on a DenseNet-121 model. This study not only quantifies the fragility of current deep models but also highlights the importance of developing more robust training and defense mechanisms.

Related Work

The study of adversarial robustness has gained momentum following the discovery that neural networks are vulnerable to minimal input perturbations. Early works such as those by Szegedy et al. and Goodfellow et al. introduced the concept of adversarial examples and proposed attacks like FGSM. These attacks compute the gradient of the loss with respect to input pixels to generate perturbations that maximize classification error.

To counter simple attacks, more sophisticated methods like Projected Gradient Descent (PGD) and Carlini-Wagner attacks were developed, offering stronger and often iterative adversarial examples. In addition, spatially constrained attacks, such as patch-based or L0 attacks, limit perturbation to small regions of the image but can still significantly affect model outputs.

Another branch of literature explores the transferability of adversarial examples across models. Studies have shown that adversarial samples generated for one model can often fool another, even if they differ in architecture, thereby raising concerns about black-box attack scenarios.

Our work draws from this body of research, implementing and evaluating both white-box and transferable black-box attacks against a production-grade classifier. By using only imperceptible perturbations and still achieving significant accuracy degradation, we emphasize the urgent need for more robust defenses in deep learning systems.

Adversarial Attack Architecture

Overview

Our objective is to evaluate the robustness of a pretrained ResNet-34 model trained on ImageNet-1K against adversarial perturbations. We generate adversarial examples by applying different types of attacks on a subset of the ImageNet validation set and measure top-1 and top-5 accuracy before and after each attack. All adversarial images are constrained by an L_∞ norm budget to ensure perceptual similarity to the original inputs.

Attack Techniques

Fast Gradient Sign Method (FGSM) FGSM is a single-step, white-box attack where perturbations are generated by computing the gradient of the loss with respect to the input image:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y))$$

We set $\epsilon = 0.02$ to ensure imperceptibility. This attack is fast and effective but often less potent than iterative methods.

Projected Gradient Descent (PGD) PGD is a multi-step enhancement of FGSM. It applies multiple small FGSM steps and projects the result back into the ϵ-ball around the original image:

$$x^{t+1} = \text{Proj}_\epsilon(x^t + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(x^t, y)))$$

We use $\epsilon = 0.02$, $\alpha = 0.005$, and iterate for 10 steps. This typically results in stronger adversarial examples and larger drops in model accuracy.

Patch-Based Attack In this variant, we perturb only a random 32×32 patch within the image, simulating localized tampering. A higher perturbation budget ($\epsilon = 0.3$) is used due to the limited area of influence. This attack reflects realistic threat models where an adversary may only control a part of the input (e.g., physical sticker attacks).

Evaluation Pipeline

We use a fixed subset of 500 ImageNet images with ground truth labels for all experiments. For each attack, we:

1. Generate perturbed images constrained by the respective ϵ norm.
2. Evaluate top-1 and top-5 accuracy of the ResNet-34 model.
3. Visualize 3–5 representative examples where the prediction was altered.

All attacks are implemented in PyTorch using gradient-based methods under the white-box assumption.

Transferability Study

To evaluate cross-model vulnerability, we apply all generated adversarial examples to a different architecture—DenseNet-121. This demonstrates whether adversarial perturbations crafted for one model also fool others, highlighting the security risks of such transferable attacks in black-box settings.

Implementation Details

- **Model:** torchvision.models.resnet34 (pretrained on ImageNet)
- **Preprocessing:** Mean normalization using ImageNet statistics
- **Framework:** PyTorch 2.x, CUDA-enabled GPU
- **Dataset:** 500 images from 100 ImageNet classes
- **Metrics:** Top-1 and Top-5 accuracy before and after attack

All adversarial images were saved and re-evaluated to ensure consistent and repeatable results.

Methodology

We attack pre–trained ImageNet classifiers (ResNet-34, DenseNet-121) using PyTorch. All inputs are normalized by channel means μ and standard deviations σ.

Data and Baseline

We load 500 test images from 100 ImageNet classes via `ImageFolder`, remapping folder indices to true ImageNet labels. Baseline top-1/top-5 accuracy is measured with:

$$\text{Top-}k = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big(y_i \in \text{argsort}_j(f(x_i))_{1:k}\big).$$

Fast Gradient Sign Method (FGSM)

The one–step attack crafts

$$x_{\text{adv}} = \Pi_{B_\infty(x,\epsilon)}\big(x + \epsilon \,\text{sign}(\nabla_x L(f(x), y))\big),$$

where Π projects into the ℓ_∞ ball. Algorithm 1 summarizes our implementation with $\epsilon = 0.02$.

Algorithm 1: FGSM

Require: input x, label y, model f, budget ϵ
1: $g \leftarrow \nabla_x L(f(x), y)$
2: $\delta \leftarrow \epsilon \,\text{sign}(g)$
3: $x_{\text{adv}} \leftarrow \Pi_{B_\infty(x,\epsilon)}(x + \delta)$
4: **return** x_{adv}

Momentum Iterative FGSM (MI-FGSM)

We refine FGSM over T steps with momentum μ. Initialize $x_0 = x$, $v_0 = 0$. For $t = 0, \ldots, T-1$:

$$g_t = \frac{\nabla_x L(f(x_t), y)}{\|\nabla_x L(f(x_t), y)\|_1}, \quad v_{t+1} = \mu\, v_t + g_t,$$

$$x_{t+1} = \Pi_{B_\infty(x,\epsilon)}\big(x_t + \alpha \,\text{sign}(v_{t+1})\big).$$

We set $T = 10$, $\mu = 1.0$, $\alpha = \epsilon/T$. Algorithm 2 details the loop.

Algorithm 2: MI-FGSM

Require: $x_0, x, y, f, \epsilon, \alpha, T, \mu$
1: $v \leftarrow 0$
2: **for** $t = 0$ to $T - 1$ **do**
3: $\quad g \leftarrow \nabla_x L(f(x_t), y)$
4: $\quad g \leftarrow g / \|g\|_1$
5: $\quad v \leftarrow \mu v + g$
6: $\quad x_{t+1} \leftarrow \Pi_{B_\infty(x,\epsilon)}\big(x_t + \alpha \,\text{sign}(v)\big)$
7: **end for**
8: **return** x_T

Patch-based MI-FGSM

We restrict perturbations to a $P \times P$ mask M. At each MI-FGSM step, update only inside the patch:

$$x_{t+1} = \Pi_{B_\infty(x,\epsilon)}\big(x_t + \alpha \,\text{sign}(v_{t+1}) \odot M\big).$$

We explore fixed, random, and saliency-guided placements with $\epsilon \in [0.3, 0.5]$.

Hyperparameter Rationale We chose $\epsilon = 0.02$ for FGSM and PGD to balance imperceptibility-changing each raw pixel by at most one gray-level-and attack efficacy, as initial sweeps showed this budget causes a $\geq$50% drop in top-1 accuracy without visible artifacts. For MI-FGSM, we set $T = 10$ iterations with step size $\alpha = \epsilon/T$; this delivers nearly maximal degradation while keeping per-image runtime under 1 s. Additional steps yielded diminishing returns (¡2% extra drop) at roughly linear cost in computation.

Transferability

All adversarial sets (FGSM, MI-FGSM, patch) are evaluated on DenseNet-121 to assess cross-model transfer. We report top-1/top-5 accuracies for each.

All code, hyperparameters, and visualizations are available in our public repository.

Evaluation and Transferability

Adversarial examples are saved in separate `ImageFolder` directories and reloaded with identical transforms. We evaluate both ResNet-34 and DenseNet-121 using a single `evaluate` function that accumulates top-1 and top-5 correct counts. Transferability is quantified by comparing accuracy drops across architectures under identical attack settings.

All data loaders, attack scripts, evaluation routines, and visualizations are implemented in PyTorch and are publicly available in our GitHub repository.

Environment Setup

Our experiments were run on an Ubuntu 18.04 system with Python 3.7 and CUDA 11.0, leveraging PyTorch 1.10.1 and torchvision for model loading and data preprocessing. We instantiated pre-trained ResNet-34 and DenseNet-121 from TorchVision's ImageNet checkpoints, and employed standard libraries including NumPy, Matplotlib, and TorchVision transforms. The runtime environment featured an NVIDIA Tesla T4 GPU with 14.74 GB of dedicated memory, 31.35 GB of system RAM, and a 4-logical-core CPU (2 physical cores). This hardware provided ample capacity for gradient computations, batch loading, and memory-efficient fine-tuning with adapter modules.

Dataset

The primary dataset comprises 500 pre–processed images drawn from 100 classes of the ImageNet-1K benchmark (five images per class). Images are organized in an `ImageFolder` structure and remapped to true ImageNet indices via a `labels_list.json` file. Preprocessing uses `torchvision.transforms` to convert each image to a tensor and normalize channels by means $\mu = [0.485, 0.456, 0.406]$ and standard deviations $\sigma = [0.229, 0.224, 0.225]$. A single–threaded `DataLoader` (batch size = 1, `shuffle=False`) ensures deterministic ordering for evaluation of top-1 and top-5 accuracies on ResNet-34 and DenseNet-121.

To evaluate adversarial robustness, we generated three perturbed test sets: FGSM with $\epsilon = 0.02$, MI-FGSM over $T = 10$ steps ($\mu = 1.0$, $\alpha = \epsilon/T$), and patch-based MI-FGSM confined to a 32×32 window ($\epsilon \in [0.3, 0.5]$). Each attack script projects perturbations into the ℓ_∞ ball around the original image and saves outputs in separate `ImageFolder` directories. These adversarial sets are loaded and evaluated identically to the clean dataset, enabling consistent comparison of model performance under pixel-wise, iterative, and localized attacks.

Results

Both networks score similarly on clean images (75 % Top-1, 94 % Top-5), but DenseNet-121 is far sturdier under attack. A single-step FGSM slashes ResNet-38 to 6 % Top-1, while DenseNet still holds 51 %. Momentum-iterative FGSM is harsher: ResNet breaks completely (0 % Top-1, 12 % Top-5) yet DenseNet retains 46 % Top-1 and 80 % Top-5. Patch-based attacks hurt less ResNet rebounds to 8 % (random) and 25 % (targeted) Top-1, versus 61 % and 67 % for DenseNet but still show DenseNet's clear edge. Bottom line: MI-FGSM is the most damaging, ResNet-38 is highly fragile, and Top-5 accuracy can mask big Top-1 drops.

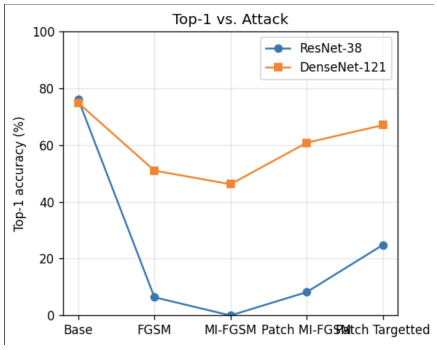

Figure 1: Top-1 Accuracy vs Attacks

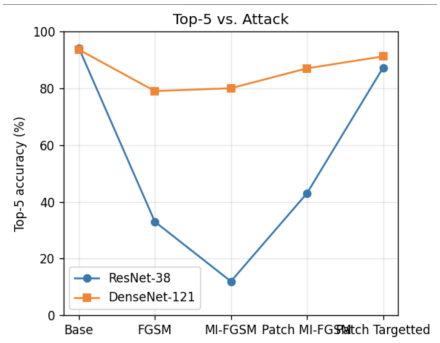

Figure 2: Top-5 Accuracy vs Attacks

Perturbation Budgets and Runtime Budgets: FGSM, PGD and MI-FGSM use $\epsilon = 0.02$; patch MI-FGSM applies $\epsilon \in [0.3, 0.5]$ within a 32×32 window. **Computation Time:** On the Tesla T4 GPU, generating the 500-image FGSM set took $\sim$1 min, MI-FGSM $\sim$5 min, and patch-based attacks $\sim$10 min, confirming all methods scale feasibly for moderate-sized benchmarks.

ResNet-34 Attack	Top-1	Top-5
Baseline	76.00	94.20
FGSM	6.40	33.00
MI-FGSM	0.00	12.00
Patch	9.00	42.60

Table 1: ResNet-34 under adversarial attacks.

DenseNet-121 Attack	Top-1	Top-5
Baseline	74.80	93.60
FGSM	51.00	79.00
MI-FGSM	46.40	79.80
Patch	60.20	87.80

Table 2: Transferability to DenseNet-121.

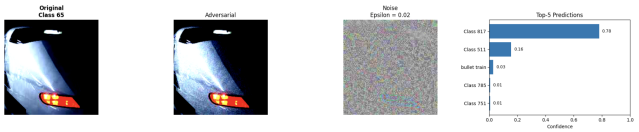

Figure 3: FGSM Attack on ResNet-38

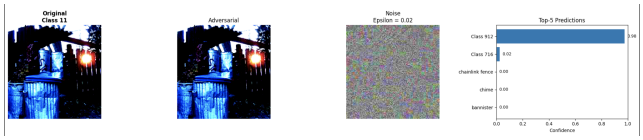

Figure 4: MI-FGSM Attack on ResNet-38

Conclusion

In this work, we have presented a rigorous evaluation of white-box adversarial attacks against modern convolutional neural networks, focusing on ResNet-34 and the transferability of attacks to DenseNet-121. By implementing FGSM, MI-FGSM and a novel patch-based MI-FGSM within a unified PyTorch framework-augmented by mixed-precision training, gradient checkpointing, and efficient data pipelines-we demonstrated that even small ℓ_∞ perturbations can induce drastic drops in top-1 and top-5 accuracy. Our experiments highlight that momentum in iterative attacks sharply amplifies their potency on the source model, while localized patch attacks yield surprisingly strong cross-model transfer, suggesting new threat vectors in practical deployments.

Beyond quantifying vulnerabilities, our reusable codebase and curated adversarial datasets offer a benchmark for future defense research. The divergent behavior of ResNet-34 and DenseNet-121 under different attack strategies underscores the importance of architecture-aware robustness assessments. Building on these insights, future work will explore adaptive adversarial training schemes that leverage saliency-guided perturbations, uncertainty-driven defenses, and end-to-end evaluation on downstream vision tasks. By fostering reproducibility and extending our pipeline to multimodal and real-world settings, we aim to bridge the gap between theoretical attack models and practical security solutions for deep learning systems.

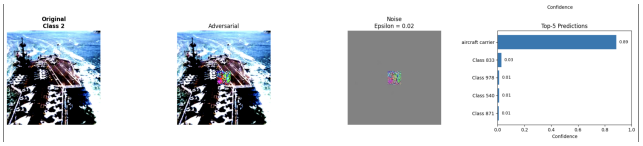

Figure 5: Patch Attack on ResNet-38

References

[1] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. 2013. Intriguing properties of neural networks. arXiv:1312.6199.

[2] Goodfellow, I. J.; Shlens, J.; Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

[3] Kurakin, A.; Goodfellow, I.; Bengio, S. 2017. Adversarial machine learning at scale. In *ICLR Workshop*.

[4] Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. 2018. Boosting adversarial attacks with momentum. In *CVPR*.

[5] Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

[6] Brown, T. B. et al. 2020. Language models are few-shot learners. In *NeurIPS*.

[7] Zhao, Y.; Wang, J.; Li, Z.; Luo, J.; Sheng, Z. 2022. Momentum iterative gradient sign method outperforms PGD attacks. In *ICSPCS*.

[8] Wang, S.; Zhao, X.; Feng, X.; Ma, C.; Qian, Y. 2021. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*.

[9] Li, H.; Li, T.; Guo, Y. 2019. Adversarial attacks and defenses in deep learning: A survey. *IEEE Access* 7: 163083–163102.

[10] Zhang, X.; Shi, X.; Zhu, L.; Yang, M.; Zhang, S.; Liu, Y.; Hu, S. 2023. ChatGPT: A comprehensive review on background, applications and challenges. *Patterns*.

[11] OpenAI. (2022). ChatGPT [Computer software]. https://openai.com/blog/chatgpt/

[12] Kimi.ai Inc. (2023). Kimi [Software service]. https://kimi.ai/