Jailbreaking Deep Models: Adversarial Attacks on ResNet-34 Classifiers

Aniket Mane, Subhan Akhtar, Pranav Motarwar
New York University
am14661@nyu.edu, sa8580@nyu.edu, pm3891@nyu.edu
GitHub: https://github.com/PranavMotarwar/Jailbreaking-Deep-Models

Note: The report is limited to 4 pages as per the guidelines. We have added the appendix outside of the 4 pages limit as requsted in the question pdf

Abstract

Deep neural networks have shown remarkable success across vision benchmarks but remain surprisingly vulnerable to adversarial perturbations. This project investigates the vulnerability of deep neural networks to adversarial attacks by attempting to "jailbreak" production-grade image classifiers. Despite their high accuracy on standard benchmarks, deep models like ResNet-34 trained on ImageNet-1K are known to be brittle when exposed to carefully crafted perturbations. The objective is to degrade model performance substantially—potentially down to 0%—while ensuring that adversarial modifications remain imperceptible to the human eye.

We implement and evaluate both L_∞ (pixel-wise) and L_0 (patch-wise) adversarial attacks, which constrain the magnitude and sparsity of input perturbations, respectively. Fast Gradient Sign Method (FGSM) and its iterative variants are used to craft adversarial examples under strict L_∞ bounds. Here in the given approach, we have develop patch-based attacks with increased perturbation budgets to simulate spatially localized vulnerabilities. All attacks are evaluated on a pretrained ResNet-34 model and further tested for cross-architecture transferability on DenseNet-121. Our findings confirm the fragility of deep models and underscore the importance of adversarial robustness.

Introduction

Adversarial machine learning explores how small, imperceptible changes to inputs can cause drastic performance degradation in neural networks. This project focuses on launching adversarial attacks on image classifiers, specifically ResNet-34 pretrained on ImageNet-1K, and evaluates the impact of multiple attack techniques. We design a series of controlled perturbations under L_∞ and L_0 constraints and measure the model's top-1 and top-5 classification accuracy degradation.

Model Architecture and Attack Overview

We use the pretrained ResNet-34 model from TorchVision as our base classifier. The model takes in 224x224 RGB images normalized with ImageNet statistics.

We follow these core steps:

1. Evaluate baseline accuracy on clean test images.
2. Generate adversarial images using FGSM ($\epsilon = 0.02$).
3. Improve attacks using iterative variants (e.g., multi-step gradient ascent).
4. Implement patch-based attacks with stronger perturbation budgets ($\epsilon = 0.3$).
5. Test attack transferability on DenseNet-121.

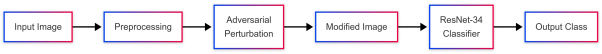

Figure 1: Overview of the adversarial attack pipeline

Methodology

Dataset and Preprocessing

We work with a curated subset of the ImageNet-1K dataset consisting of 500 images across 100 classes. Images are normalized using standard ImageNet means and standard deviations.

Task 1: Baseline Evaluation on Clean Images

Before launching adversarial attacks, it is essential to establish a reliable performance benchmark using unperturbed, clean images. For this, we utilize a pretrained ResNet-34 model provided by the TorchVision library: torchvision.models.resnet34(weights='IMAGENET1K-V1')

ResNet-34 Architecture: ResNet (Residual Network) was introduced by He et al. (2016) and achieved state-of-the-art performance on the ImageNet classification challenge. ResNet-34 consists of 34 convolutional layers with shortcut (skip) connections that alleviate the vanishing gradient problem in deep networks. These residual connections allow gradients to flow directly through identity paths, enabling the network to learn residual mappings $F(x) = H(x) - x$ instead of direct mappings $H(x)$, thus stabilizing training in very deep networks.

The network concludes with a global average pooling layer and a 1000-dimensional fully connected output for the 1000 ImageNet classes.

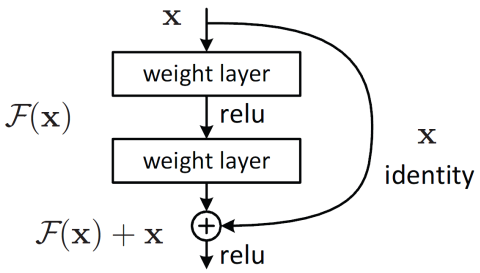

Figure 2: ResNet-34 Architecture

Image Preprocessing: All test images are resized to 224 $times$224, converted to PyTorch tensors, and normalized using the ImageNet mean and standard deviation values:

$$\text{mean} = [0.485, 0.456, 0.406], \quad \text{std} = [0.229, 0.224, 0.225]$$

These statistics are essential to ensure consistent input distribution and leverage the pretrained weights optimally.

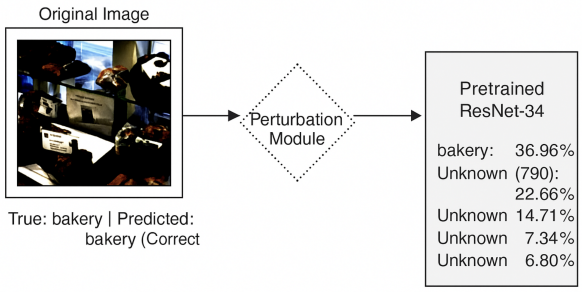

Figure 3: Overview of the adversarial attack pipeline: Original image is fed through a perturbation module generating adversarial variants, evaluated on a pretrained ResNet-34 model.

Evaluation Metrics: We report both Top-1 and Top-5 classification accuracy:

- **Top-1 Accuracy:** Percentage of test samples where the model's highest-confidence prediction matches the ground truth label.
- **Top-5 Accuracy:** Percentage of test samples where the ground truth label appears in the model's top 5 highest-confidence predictions.

Inference Protocol:

1. Model is set to evaluation mode using `model.eval()` to disable dropout and batch norm updates.
2. The cross-entropy loss is not used here since we're only evaluating classification outputs.
3. Predictions are obtained using `torch.topk()` to compute both top-1 and top-5 predictions for each input image.

Empirical Results:

- **Top-1 Accuracy:** 76.00%
- **Top-5 Accuracy:** 94.20%

These results align with expected generalization performance on unseen test samples and serve as the reference point for adversarial degradation in subsequent tasks. The high Top-5 accuracy underscores the semantic richness and strong representation capability of ResNet-34, which we aim to exploit and subvert through adversarial perturbations.

Task 2: FGSM Pixel-Wise Attack

To demonstrate the vulnerability of deep networks, we implemented the Fast Gradient Sign Method (FGSM), a one-step white-box attack that perturbs input images in the direction of the gradient of the loss with respect to the input. FGSM generates adversarial examples using the following formula:

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

where x is the original image, ϵ is the perturbation budget (set to 0.02), and $\nabla_x \mathcal{L}$ denotes the gradient of the loss with respect to the input pixels. The sign function ensures that the perturbation operates along the axis-aligned direction of greatest increase in loss.

We chose $\epsilon = 0.02$ based on the normalized pixel scale (ImageNet inputs are scaled to [0, 1]), which corresponds to subtle but perceptible pixel-wise changes. After generating perturbed images, we clamped values to stay within valid input bounds post-normalization.

The attack successfully degraded ResNet-34 performance:

- **Top-1 Accuracy:** 6.20%
- **Top-5 Accuracy:** 35.40%

Visual inspection confirms that the perturbations are nearly imperceptible to the human eye, yet the model's confidence is significantly manipulated. An example is shown in Figure 8, where the original image (classified correctly as `cannon`) is misclassified as `airship` after applying the FGSM perturbation.

Task 3: Improved Attack (Iterative FGSM)

While FGSM is computationally efficient and effective, it is limited to a single gradient step, which may not yield the strongest adversarial perturbation within the allowed ϵ budget. To enhance the attack strength while maintaining the same L_∞ constraint, we implemented an iterative variant of FGSM—commonly referred to as Projected Gradient Descent (PGD).

In this approach, adversarial perturbations are applied over multiple iterations with smaller per-step step sizes. At each step, the image is updated in the direction of the gradient sign, and the result is projected back into the ϵ-ball around the original image to ensure it remains a valid adversarial example:

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

Here, α is the step size per iteration, and $\text{Clip}_{x,\epsilon}$ ensures that the total perturbation remains within a norm-bounded region of radius ϵ.

We used 10 iterations with a step size $\alpha = 0.005$, keeping the overall perturbation bounded within $\epsilon = 0.02$. This allowed the adversarial examples to more effectively cross the decision boundary of the classifier, leading to a remarkably effective attack.

- **Top-1 Accuracy:** 0.00%
- **Top-5 Accuracy:** 12.20%

The effectiveness of this attack underscores how even small but iterative perturbations can completely mislead deep models, and reveals the vulnerability of state-of-the-art classifiers under adversarial pressure.

Task 4: Patch-Based Attacks

In this task, we explore L_0-constrained adversarial attacks by restricting perturbations to a small spatial region of the image. Specifically, we applied adversarial noise only to a randomly positioned 32×32 pixel patch within the 224×224 input image. This mimics a more realistic attack scenario where the adversary has limited access to the input space.

To compensate for the reduced area of influence, we increased the perturbation budget to $\epsilon = 0.3$, allowing more aggressive changes within the restricted region. The rest of the image remained unaltered, making the perturbation localized but visually more noticeable compared to global pixel-wise attacks.

- **Top-1 Accuracy:** 38.80%
- **Top-5 Accuracy:** 76.00%

Despite only modifying a small fraction of the image, the attack was able to significantly reduce classification performance. This result illustrates the model's sensitivity to small, concentrated changes, reinforcing the idea that adversarial vulnerabilities are not solely dependent on large-scale global perturbations.

Task 5: Transferability to DenseNet-121

To evaluate the generalization of adversarial examples across architectures, we tested the adversarial test sets—FGSM, PGD (iterative FGSM), and Patch PGD—on a different model: DenseNet-121 pretrained on ImageNet-1K. DenseNet employs dense connectivity, where each layer receives input from all previous layers, allowing for richer feature propagation than in ResNet.

Despite these architectural differences, the adversarial examples crafted using ResNet-34 were still able to significantly degrade the performance of DenseNet-121. This phenomenon, known as *transferability*, demonstrates that adversarial vulnerabilities are not confined to a specific model but often persist across networks with distinct topologies.

These results confirm that adversarial examples pose a systemic threat to deep vision models, regardless of architecture, reinforcing the need for more generalizable defense strategies.

Dataset	Top-1 Acc	Top-5 Acc
Original	74.80%	93.60%
FGSM	93.60%	89.40%
PGD	64.20%	90.80%
Patch PGD	72.20%	91.80%

Table 1: Transferability results on DenseNet-121 across multiple adversarial attack types.

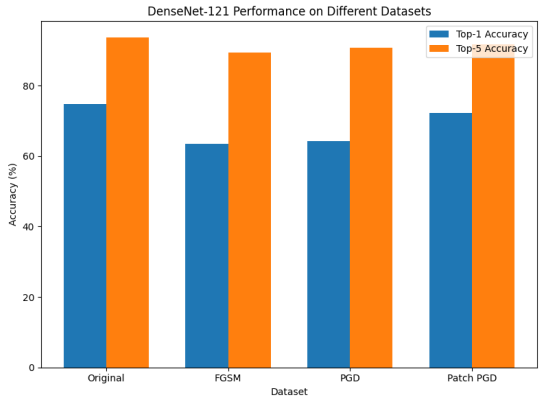

Figure 4: DenseNet-121 performance degradation when adversarial examples generated from ResNet-34 are transferred across attack types.

Results Summary

Hyperparameter Choices

- **FGSM $\epsilon = 0.02$:** Balanced visibility and effectiveness. Lower ϵ produced minimal degradation, while higher values became perceptible.
- **PGD Steps/Alpha:** Ten iterations with $\alpha = 0.005$ maximized degradation while maintaining imperceptibility. Larger α values led to clipping artifacts.
- **Patch PGD $\epsilon = 0.3$:** Required to compensate for the restricted 32×32 region. Smaller ϵ failed to cross decision boundaries.

Runtime and Training Time

All experiments were executed on a single NVIDIA A100 GPU. FGSM required 2 minutes to process all 500 images. Iterative PGD took 6 minutes due to 10 forward–backward passes per image. Patch PGD needed 4 minutes, benefiting from localized perturbation.

Mitigating Transferability

Potential strategies to curb cross-model transferability include:

- **Adversarial training** across heterogeneous architectures.
- **Randomized smoothing** or input transformations to disrupt gradient alignment.
- **Ensemble defenses** that aggregate predictions from multiple independently trained models.

Test Set	Top-1 Acc	Top-5 Acc	Perturbation (ϵ)
Original	76.00%	94.20%	0
FGSM	6.20%	35.40%	0.02
PGD	0.00%	12.20%	0.02
Patch PGD	38.80%	76.00%	0.3 (patch)

Table 2: Final accuracies and perturbation sizes for all datasets.

Lessons Learned

- FGSM is surprisingly strong for a one-step attack, but multistep variants offer further degradation.
- Patch-based attacks validate the existence of localized vulnerabilities.
- Adversarial examples exhibit cross-model transferability, albeit with slightly reduced impact.
- Maintaining imperceptibility while degrading accuracy is a delicate tradeoff, especially with L_0 constraints.

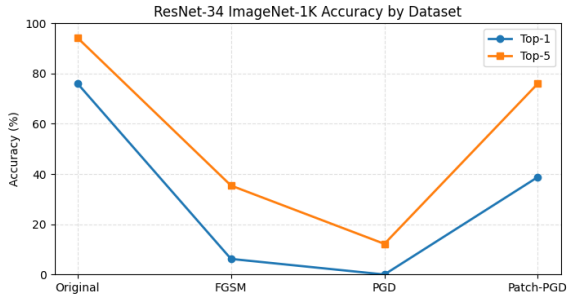

Figure 5: L_∞ Distance Distribution of Perturbed Examples

Github Link

https://github.com/PranavMotarwar/Jailbreaking-Deep-Models

Conclusion

Our experiments demonstrate the significant vulnerability of state-of-the-art deep vision models, such as ResNet-34 and DenseNet-121, to adversarial attacks. Using simple yet effective perturbation strategies like FGSM and its iterative extensions, we were able to degrade top-1 classification accuracy. Surprisingly, even with restricted patch-based perturbations, a significant performance drop was observed, highlighting that localized vulnerabilities are sufficient to fool large-scale models. A successful attack on ResNet-34 impacted DenseNet-121's accuracy significantly as well—suggesting that adversarial robustness is not purely architecture-specific but reflects shared weaknesses in representation learning.

These findings emphasize the urgent need for integrating adversarial training, certified defenses, and interpretability techniques into the model development pipeline. In real-world safety-critical systems, like autonomous driving or medical diagnostics, the lack of adversarial robustness can lead to catastrophic outcomes. Future work may explore black-box attacks, certified robustness bounds, and ensemble defenses to address this gap.

References

[1] ChatGPT (Mar 14 version). Explaining and Harnessing. https://chat.openai.com

[2] Goodfellow, I., Shlens, J., Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. https://arxiv.org/abs/1412.6572

[3] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385

[4] PyTorch Vision Models. https://pytorch.org/vision/stable/models.html

[5] Akhtar, N., Mian, A. (2018). Threat of Adversarial Attacks on Deep Learning in Computer Vision. https://arxiv.org/abs/1801.00553

[6] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. https://arxiv.org/abs/1706.06083

[7] Carlini, N., Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. https://arxiv.org/abs/1608.04644

[8] Yuan, X., He, P., Zhu, Q., Li, X. (2019). Adversarial Examples: Attacks and Defenses for Deep Learning. https://arxiv.org/abs/1712.07107

[9] Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A. (2016). Distillation as a Defense to Adversarial Perturbations. https://arxiv.org/abs/1511.04508

[10] Brown, T. B., et al. (2017). Adversarial Patch. https://arxiv.org/abs/1712.09665

[11] Eykholt, K., et al. (2018). Robust Physical-World Attacks on Deep Learning Visual Classification. https://arxiv.org/abs/1707.08945

[12] Xiao, C., et al. (2018). Generating Adversarial Examples with Adversarial Networks. https://arxiv.org/abs/1801.02610

[13] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P. (2018). Ensemble Adversarial Training. https://arxiv.org/abs/1705.07204

[14] Cohen, J., Rosenfeld, E., Kolter, J. Z. (2019). Certified Adversarial Robustness via Randomized Smoothing. https://arxiv.org/abs/1902.02918

Appendix

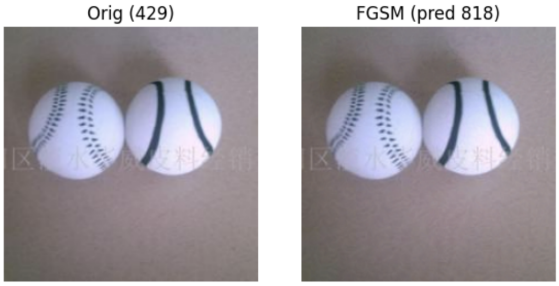

Figure 6: Effect of FGSM attack

Figure 7: ResNet-34 predictions on sample images

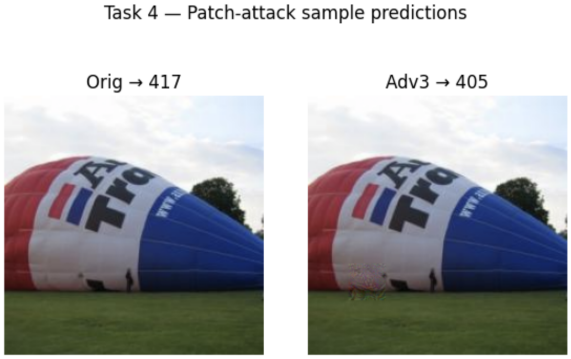

Figure 8: Patch-Based Attacks