Deep Learning Project 3: Jailbreaking Deep Models

Murali Peddi[1], **Tejdeep Chippa**[1], **Vamsi Jonnakuti**[1]

[1]Computer Engineering Department, New York University, Tandon School of Engineering
mp6904@nyu.edu, tc4263@nyu.edu, vj2280@nyu.edu

Abstract

Deep neural networks are highly accurate yet vulnerable to adversarial examples—subtle input perturbations that cause misclassification. In this project, we attack a pretrained ResNet-34 model using FGSM, PGD, and patch-based PGD on a 100-class ImageNet subset. Under tight ℓ_∞ constraints, PGD reduces Top-1 accuracy from 76.00% to 0.00%, while even localized patch attacks drop it to 30.20%. These results highlight the fragility of deep models and the need for robust defenses.

Introduction

Deep neural networks excel at vision tasks but remain vulnerable to adversarial examples—subtle input perturbations that can cause severe misclassifications. These weaknesses pose challenges for deploying models in safety-critical applications.

This project investigates adversarial robustness by attacking a pretrained ResNet-34 model on a 100-class ImageNet subset. We evaluate three methods like FGSM which uses single-step pixel-wise perturbation, PGD which is an iterative variant with stronger impact and Patch PGD which is a localized attack on a 32×32 region.

Under tight ℓ_∞ or spatially localized ℓ_0 constraints, these attacks significantly degrade accuracy—PGD reduces Top-1 accuracy to 0%, and patch attacks cause over 60% drop. Our findings underscore the need for robust defenses in modern vision systems.

Dataset & Experimental Setup

We perform adversarial evaluations using a curated subset of the **ImageNet-1K** dataset comprising **500 RGB images across 100 distinct classes**. Each class corresponds to a WordNet synset ID (e.g., `n02808304`) and contains 5 randomly sampled validation images. Class-to-label mapping is defined in a `labels_list.json` file provided alongside the dataset. These labels are essential for evaluating classification accuracy under attack scenarios.

Preprocessing Pipeline

Images are first resized to **224×224** pixels to match the input resolution of standard ImageNet models. We apply the standard normalization used in pretrained torchvision models:

$$\mu = [0.485,\ 0.456,\ 0.406], \quad \sigma = [0.229,\ 0.224,\ 0.225]$$

Normalized inputs are computed using:

$$x_{\text{norm}} = \frac{x_{\text{raw}} - \mu}{\sigma}$$

This normalization was consistently applied across all tasks, including clean evaluation and adversarial inference.

Base Model

All attacks target the pretrained **ResNet-34** model provided via PyTorch's `torchvision.models` API. The model is loaded with `IMAGENET1K_V1` weights and evaluated in `eval()` mode on a CUDA-enabled GPU.

```
model = torchvision.models.resnet34(weights=
    'IMAGENET1K_V1').to("cuda").eval()
```

To map folder names to integer labels, we built a dictionary that assigns sequential class indices starting from 401:

```
folder_to_label = {folder: 401 + i for i,
    folder in enumerate(sorted(class_folders
    ))}
```

This consistent indexing ensures accurate label matching for Top-k accuracy evaluations.

Evaluation Protocol

We report **Top-1** and **Top-5 accuracy** on both the clean dataset and each adversarially perturbed version. These metrics are defined as:

- **Top-1 Accuracy**: The percentage of images where the top predicted label matches the true class.
- **Top-5 Accuracy**: The percentage of images where the true class appears in the top five predicted labels.

Methodology

We explored three primary adversarial attack strategies—Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and patch-based PGD—each designed

Project Codebase: Github Repository Link

to degrade the classification performance of a pretrained ResNet-34 model on ImageNet-like inputs. All attacks were implemented in **raw pixel space** to ensure precise control over the perturbation budget ϵ, while normalized inputs were used for model inference.

Task 1 – Clean Baseline Evaluation

To establish a baseline, we evaluated the pretrained ResNet-34 model on the unaltered dataset of 500 images. Inputs were normalized using standard ImageNet statistics, and accuracy was computed using Top-1 and Top-5 metrics. The model achieved a Top-1 accuracy of **76.00%** and Top-5 accuracy of **94.20%**, forming the reference for quantifying the degradation caused by subsequent attacks.

Task 2 – Fast Gradient Sign Method (FGSM)

We implemented FGSM by computing the sign of the gradient of the cross-entropy loss with respect to the input image, scaled by a perturbation budget $\epsilon = 0.02$. Critically, gradients were computed in the **raw pixel space**, using the chain rule to propagate through the normalization layer:

$$x_{\text{adv}} = x_{\text{raw}} + \epsilon \cdot \text{sign}\left(\frac{\partial \mathcal{L}}{\partial x_{\text{norm}}} \cdot \frac{1}{\sigma}\right)$$

where $x_{\text{norm}} = \frac{x_{\text{raw}} - \mu}{\sigma}$ and μ, σ are the normalization constants.

The perturbation was clipped to the $[0, 1]$ valid image range. FGSM generated the first adversarial dataset ("Adversarial Test Set 1"), causing a notable accuracy drop (Top-1: **3.60%**, Top-5: **20.80%**).

Task 3 – Projected Gradient Descent (PGD)

To improve upon FGSM, we extended it into a multi-step PGD attack under the same $\epsilon = 0.02$ constraint. For 10 steps, we iteratively applied gradient updates in raw space using:

$$x_{\text{adv}}^{(t+1)} = \Pi_{B_\infty(\epsilon)}\left[x_{\text{adv}}^{(t)} + \alpha \cdot \text{sign}\left(\nabla_{x_{\text{raw}}} \mathcal{L}\right)\right]$$

where $\alpha = 0.005$ is the step size and $\Pi_{B_\infty(\epsilon)}$ denotes projection back into the ϵ-ball around the original input.

The resulting adversarial dataset ("Adversarial Test Set 2") reduced model accuracy to Top-1: **0.00%**, Top-5: **1.40%**, showing PGD's effectiveness even under constrained perturbations.

Task 4 – Patch-Based PGD Attack

Lastly, we localized perturbations to a **32×32** random patch within each image while increasing the budget to $\epsilon = 0.5$. The patch-based PGD followed a similar 10-step loop as above, but restricted updates to a spatial slice:

$$x_{\text{adv}}^{(t+1)}[p] = \Pi_{B_\infty(\epsilon)}\left[x_{\text{adv}}^{(t)}[p] + \alpha \cdot \text{sign}(\nabla \mathcal{L})[p]\right]$$

where p denotes the randomly selected patch location. Despite modifying only a small portion of each image, this attack substantially degraded performance (Top-1: **30.20%**, Top-5: **67.60%**), highlighting the classifier's sensitivity to localized perturbations.

Task 5 – Transferability to DenseNet-121

To test cross-model robustness, we evaluated adversarial datasets generated on ResNet-34 against a structurally different model: DenseNet-121. This setup tests if adversarial examples transfer across architectures.

Evaluation & Results

To assess the impact of adversarial attacks on classification performance, we evaluated model accuracy across the original dataset and three perturbed variants generated via FGSM, PGD, and Patch-based PGD. Each attack adheres to strict ℓ_∞ constraints and aims to minimize perceptual differences while maximizing misclassification.

Quantitative Accuracy Comparison

Dataset	Top-1 Accuracy	Top-5 Accuracy
Clean Test Set	76.00%	94.20%
FGSM ($\epsilon = 0.02$)	3.60%	20.80%
PGD ($\epsilon = 0.02$, $\alpha = 0.005$)	0.00%	1.40%
Patch PGD ($\epsilon = 0.5$, 32×32)	30.20%	67.60%

Table 1: Top-1 and Top-5 accuracy after each attack.

As shown in Table 1, PGD was the most effective attack, degrading Top-1 accuracy from 76.00% to 0.00%, despite the perturbation budget being just $\epsilon = 0.02$. FGSM also performed well, reducing accuracy to 3.60%, but was outperformed by the iterative PGD attack. The patch-based attack, while modifying only a 32×32 region, achieved a significant drop in accuracy—demonstrating the vulnerability of convolutional architectures to localized perturbations.

Visual Examples from the Codebase

Figures below depict the original, adversarial, and perturbation-amplified images for all three attack types. These were generated using `visualize_fgsm_examples`, `visualize_pgd_examples`, and `visualize_patch_examples_simple` functions in the notebook.

FGSM Results Each image below shows the original image, the FGSM adversarial version, and the amplified perturbation. Even though the changes are imperceptible to the human eye, classification is drastically altered.

PGD Results PGD leads to nearly complete misclassification with slightly more structured noise compared to FGSM. Again, perturbations are hard to notice without amplification.

Patch PGD Results Even when constrained to a random 32×32 region, the patch attack causes over 47% Top-1 accuracy drop relative to the clean baseline.

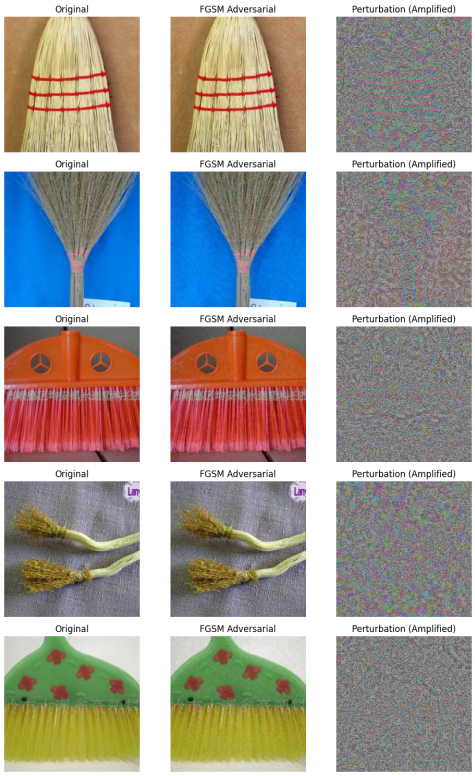

Figure 1: FGSM Results

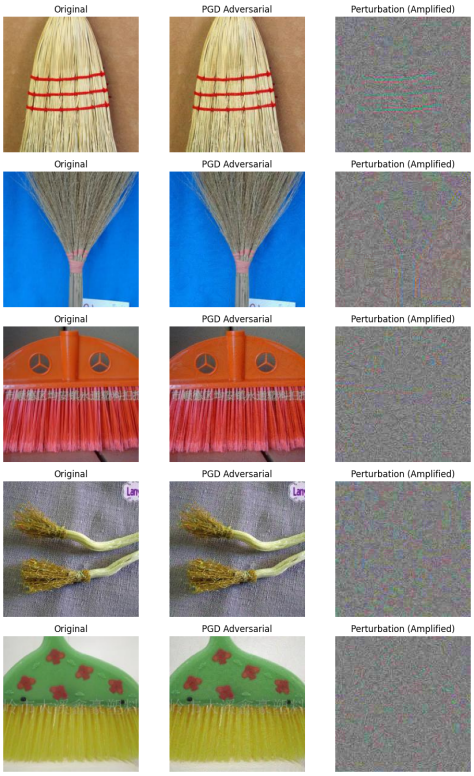

Figure 2: PGD Results

Perturbation Constraints

All attacks respected their corresponding perturbation budgets:

- **FGSM & PGD:** $\epsilon = 0.02$ applied globally
- **Patch PGD:** $\epsilon = 0.5$ applied locally within a 32×32 region

Runtime checks enforced the constraint via:

$$\max \|x_{\text{adv}} - x\|_\infty \leq \epsilon$$

No violations were observed across all 500 samples for any attack.

Evaluation of the transferability to DenseNet-121

To test cross-model robustness, we evaluated how well adversarial examples generated for ResNet-34 could fool DenseNet-121—a structurally different convolutional architecture. Table 2 summarizes the Top-1 and Top-5 accuracy of DenseNet-121 on each adversarial dataset.

Despite strong degradation on ResNet-34, the adversarial examples transferred poorly: DenseNet-121 maintained relatively high accuracy under all three attacks. This highlights the low transferability of gradient-based perturbations and suggests that attack effectiveness is tightly coupled to the target model's architecture.

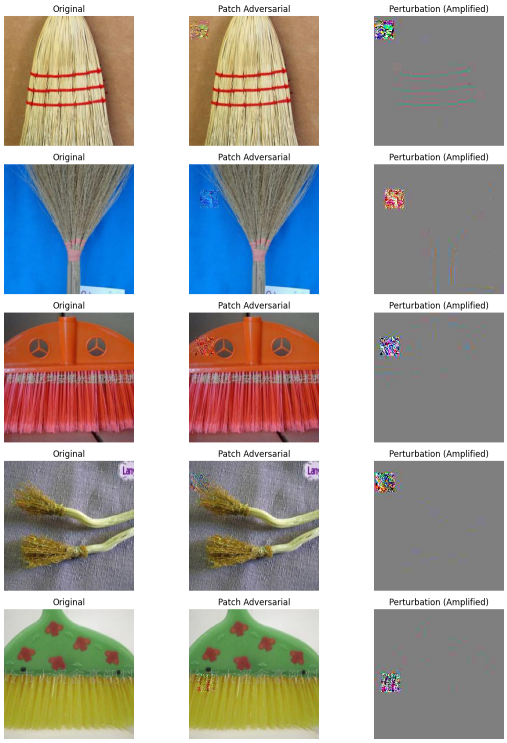

Figure 3: Patch PGD Results

Adversarial Dataset	DenseNet-121 Top-1 Accuracy	DenseNet-121 Top-5 Accuracy
Clean Test Set	74.60%	93.60%
FGSM ($\epsilon = 0.02$)	51.20%	80.00%
PGD ($\epsilon = 0.02$)	56.80%	85.40%
Patch PGD ($\epsilon = 0.5$, 32×32)	70.00%	91.40%

Table 2: DenseNet-121 accuracy on adversarial examples crafted for ResNet-34.

Discussion

Our experiments confirm that even small, imperceptible perturbations can drastically impair the performance of a state-of-the-art ResNet-34 classifier. Among all methods evaluated, PGD proved most effective, reducing Top-1 accuracy from 76.00% to 0.00%, closely followed by FGSM. Patch-based PGD, despite modifying only a 32×32 spatial region, still caused a notable degradation of nearly 50% in Top-1 accuracy.

To capture not just the raw accuracy but also the **relative degradation**, we include a new metric: **Top-1 Accuracy Drop (%)**, which measures the proportion of performance lost with respect to the baseline:

$$\text{Drop}_{\text{Top-1}} = \frac{\text{Baseline Top-1} - \text{Attack Top-1}}{\text{Baseline Top-1}} \times 100$$

$$\text{Drop}_{\text{Top-5}} = \frac{\text{Baseline Top-5} - \text{Attack Top-5}}{\text{Baseline Top-5}} \times 100$$

Dataset	**Top-1 Accuracy**	**Top-5 Accuracy**	**Top-1 Drop (%)**	**Top-5 Drop (%)**
Clean Test Set	76.00%	94.20%	0.00	0.00
FGSM ($\epsilon = 0.02$)	3.60%	20.80%	95.26	77.92
PGD ($\epsilon = 0.02$)	0.00%	1.40%	100.00	98.51
Patch PGD ($\epsilon = 0.5$, 32×32)	30.20%	67.60%	60.26	28.26

Table 3: Accuracy and degradation relative to the clean baseline.

While FGSM offers a lightweight, single-step attack, its sharp gradient direction often over-perturbs sensitive features. In contrast, PGD achieves tighter optimization within the perturbation budget via iterative refinement. Patch attacks are more localized but still effective—suggesting that critical visual cues are concentrated in small spatial regions.

Overall, these results highlight the fragility of vision models under adversarial settings and motivate future work on incorporating robustness during training (e.g., adversarial training or certified defenses).

In addition to single-model evaluations, we investigated the transferability of adversarial examples from ResNet-34 to DenseNet-121. Despite the attacks being highly effective on the source model, DenseNet-121 retained substantial robustness: for instance, Top-1 accuracy remained at 56.80% on PGD examples that fully broke ResNet-34. These results suggest that adversarial examples exhibit limited generalization across architectures and that model-specific gradients heavily influence the attack's effectiveness. This highlights the importance of evaluating robustness not only in isolation but also in cross-model scenarios, especially when considering real-world black-box threat models.

Conclusion

This project highlights the vulnerability of deep image classifiers to adversarial examples. Gradient-based attacks on a pretrained ResNet-34 model—especially PGD—were able to reduce Top-1 accuracy from 76.00% to 0.00% under small ℓ_∞ constraints. Even localized patch attacks, confined to 32×32 regions, caused significant degradation.

Among the methods, PGD was most effective, while FGSM provided a faster but weaker alternative. Patch-based PGD revealed the sensitivity of spatially localized features.

These results underscore the need for robust defenses in vision systems. Future work may explore attack transferability, adversarial training, and robustness under real-world conditions.

Acknowledgment

We consulted ChatGPT, to understand concepts, improve the writing quality and get assistance in formatting LaTeX. All final answers, code and analysis are original and written in own words.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. doi:10.1109/CVPR.2016.90

[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations (ICLR)*. arXiv:1412.6572

[3] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. *International Conference on Learning Representations (ICLR)*. arXiv:1706.06083

[4] Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F. S., & Porikli, F. (2021). Improving Transferability of Adversarial Examples with Ghost Networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5311–5320.

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing Properties of Neural Networks. *International Conference on Learning Representations (ICLR)*. arXiv:1312.6199

[6] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. doi:10.1109/CVPR.2009.5206848