Jailbreaking Deep Models: Adversarial Attacks on ImageNet Classifiers

ECE-GY 7123 Deep Learning
Amarnadh Reddy Mettu , DevaKumar Katta , Jeel Patel

New York University Tandon School of Engineering
6 MetroTech Center
Brooklyn, New York 11201 USA

Abstract

This report focuses on evaluating the vulnerability of deep learning models to adversarial attacks through the task of jailbreaking image classifiers. We employ a pretrained ResNet-34 model on ImageNet and implement three types of attacks: FGSM, iterative gradient-based, and localized patch attacks, each constrained under ℓ_∞ norms. Performance degradation is measured using Top-1 and Top-5 accuracy, with iterative attacks showing the most significant impact. We also assess the transferability of adversarial examples to DenseNet-121, confirming model-agnostic weaknesses. This work highlights the practical risks of adversarial perturbations and contributes to a better understanding of deep model robustness.

Code Availability

The source code for this project is available at:
https://github.com/amarnadh145/deeplearning_final

1 Introduction

Deep learning models have achieved remarkable performance in image classification tasks, especially with large-scale datasets like ImageNet. However, their success has revealed an important vulnerability: susceptibility to adversarial examples carefully crafted inputs that cause the model to misclassify with high confidence despite appearing visually similar to clean data.

This project investigates the robustness of deep neural networks through the lens of adversarial attacks, focusing on "jailbreaking" pretrained image classifiers. Using a ResNet-34 model trained on ImageNet, we implement and compare multiple attack strategies including Fast Gradient Sign Method (FGSM), iterative multi-step attacks, and localized patch-based perturbations. Each attack respects an L_∞ constraint to ensure visual similarity, while aiming to maximally degrade classification performance.

We further evaluate the transferability of these attacks by applying adversarial examples generated from ResNet-34 to a DenseNet-121 model. Our experiments reveal significant accuracy drops across all attack types, with iterative attacks showing the strongest impact. Visualizations of perturbed images highlight the subtlety of the manipulations and their surprising effectiveness.

This work provides a practical framework for generating and analyzing adversarial attacks, emphasizing the need for more robust architectures and defenses in safety-critical applications of deep learning.

2 Methodology

This section outlines the implementation of our adversarial attack pipeline targeting pretrained ImageNet classifiers. We detail the dataset, preprocessing techniques, model setup, and the attack strategies used to generate and evaluate adversarial examples against the ResNet-34 architecture.

2.1 Dataset

For this project, we utilize a held-out subset of 500 images sampled from the ImageNet 1K benchmark, spanning 100 unique synsets (object categories). The data are organized into class-specific directories named using WordNet synset IDs, with each folder containing raw RGB JPEG images. A `labels_list.json` file maps each folder to its corresponding ImageNet class index. Each sample includes:

- An RGB image resized and cropped to 224×224 pixels
- A ground truth ImageNet integer label

2.2 Data Preprocessing

To prepare inputs for the pretrained ResNet-34 model, we follow the preprocessing pipeline used during ImageNet training, implemented using `torchvision.transforms`:

1. **ToTensor:** Images are converted from $H \times W \times C$ format with pixel values in $[0, 255]$ to PyTorch float tensors scaled to $[0.0, 1.0]$.
2. **Normalization:** Channel-wise normalization is applied using the ImageNet training distribution:
$$\mu = [0.485,\ 0.456,\ 0.406], \quad \sigma = [0.229,\ 0.224,\ 0.225]$$
3. **DataLoader:** Processed data are wrapped using `ImageFolder` and `DataLoader` (batch size = 32, no shuffle during evaluation) for efficient batched inference.

2.3 Model Setup

We use a pretrained ResNet-34 model from the `torchvision.models` package, with weights initialized on the ImageNet 1K dataset. For transferability analysis, we also evaluate attacks on a pretrained DenseNet-121 model. Both models are set to evaluation mode during inference to disable dropout and batch normalization updates.

2.4 Attack Techniques

Three types of adversarial attacks are implemented:

- **FGSM (Fast Gradient Sign Method):** A one-step untargeted pixel-wise attack using $\epsilon = 0.02$, designed to quickly perturb the image in the direction of the loss gradient.
- **Iterative Attack (Improved):** A stronger pixel-wise attack using multiple gradient steps ($\alpha = 0.002$, steps = 5) under the same ϵ constraint, yielding significantly more effective perturbations.
- **Patch-Based Attack:** A localized attack where only a 32×32 patch is perturbed using $\epsilon = 0.5$, $\alpha = 0.05$, and 15 steps. The patch location is randomly sampled per image, making it more challenging due to its spatial constraint.

All adversarial examples are saved and reused across evaluations to enable both in-model and cross-model testing. Accuracy metrics (Top-1 and Top-5) are recorded for clean and attacked inputs to quantify the effectiveness and transferability of the adversarial methods.

3 Model Architecture

This section outlines the architecture of the pretrained models used as adversarial targets, along with the design and implementation of three gradient-based attack modules. Our setup freezes all model weights and isolates the effect of input-space perturbations on classification performance. We test both intra-model vulnerability and cross-architecture transferability.

3.1 Classifier: ResNet-34

Our primary victim model is a pretrained ResNet-34 from the TorchVision library, chosen for its well-known residual architecture and moderate depth. The architecture consists of:

- **Initial Convolutional Stem**
  - 7×7 convolution, followed by BatchNorm and ReLU
  - 3×3 max pooling for early downsampling
- **Four Residual Stages**
  - Stage 1: 3 BasicBlocks operating on 64-channel feature maps (no downsampling)
  - Stage 2: 4 BasicBlocks increasing depth to 128 channels (first block uses stride2)
  - Stage 3: 6 BasicBlocks on 256 channels, further downsampling via stride
  - Stage 4: 3 BasicBlocks with 512 channels, capturing high-level semantics
- **Final Classifier**
  - Global average pooling
  - Fully connected layer mapping 512 features to 1000 ImageNet class logits

The model comprises approximately 21.8 million parameters and is used in frozen mode to ensure that only the inputs are modified during adversarial evaluation.

3.2 Adversarial Attack Modules

We implement three differentiable, parameter-free adversarial attack pipelines. Each operates directly in input space under clearly defined perturbation budgets:

- **FGSM (Fast Gradient Sign Method)**
  - Threat model: L_∞
  - Perturbation budget: $\epsilon = 0.02$
  - Update rule: $x_{\text{adv}} = \text{clip}(x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}))$
  - Description: A single-step, efficient method producing imperceptible but effective perturbations
- **Iterative Attack (Improved PGD Variant)**
  - Threat model: L_∞
  - Perturbation budget: $\epsilon = 0.02$
  - Hyperparameters: Steps = 5, Step size $\alpha = 0.002$
  - Update: Iterative gradient ascent projected within the ϵ-ball
  - Description: Stronger multi-step variant of FGSM, improving success rate at cost of runtime
- **Patch-Based Attack**
  - Threat model: L_0 (spatially constrained)
  - Perturbation region: 32×32 patch (randomly placed)
  - Perturbation budget: $\epsilon = 0.5$ within the patch
  - Hyperparameters: Steps = 15, Step size $\alpha = 0.05$
  - Description: Applies localized perturbations to a confined region of the image to simulate stealthy or targeted attacks

3.3 Attack Integration & Evaluation

For each minibatch of input images, one of the above attack modules is applied to generate adversarial versions. These perturbed images are then passed through the frozen ResNet-34 to compute performance metrics:

- **Accuracy Metrics:** We compute Top-1 and Top-5 accuracy for clean and adversarial sets.
- **Relative Degradation:** We analyze the accuracy drop under each attack pipeline.
- **Transferability:** All three adversarial datasets are subsequently evaluated on a DenseNet-121 backbone to assess cross-architecture vulnerability.

By decoupling model weights from input-space optimization and systematically evaluating attack efficacy, we ensure reproducibility and emphasize the fragility of modern image classifiers under bounded perturbations.

4 Training and Attack Execution

In this project, no additional training of the classifier is performed. All experiments are conducted on a fixed, pretrained ResNet-34 model from the TorchVision model zoo. This design choice ensures that performance degradation is solely due to adversarial perturbations in the input space, and not influenced by model retraining or fine-tuning.

4.1 Data Handling

We utilize a balanced subset of 500 images from the ImageNet validation set, preprocessed and normalized using ImageNet-standard statistics (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]). The images are loaded using PyTorch's `ImageFolder` and `DataLoader` utilities with a batch size of 32 for both clean and adversarial evaluations. No data augmentation is applied to preserve consistency during perturbation analysis.

4.2 Loss Function and Gradients

All attack methods are gradient-based and use the standard *Cross-Entropy Loss* function to compute the classification error with respect to the true (or targeted) labels. The loss gradients are backpropagated through the frozen model to identify sensitive directions in input space. This approach enables the crafting of high-impact adversarial examples without updating any model parameters.

4.3 Attack Execution Strategy

Each attack pipeline is executed on a batch-wise basis. For a given clean batch, we compute the adversarial batch using one of three methods (FGSM, Iterative, Patch-based). After perturbation, the adversarial images are evaluated using the same ResNet-34 model to assess changes in Top-1 and Top-5 accuracy.

4.4 Transfer Evaluation

To evaluate cross-architecture transferability, all generated adversarial datasets are passed to a DenseNet-121 model (also pretrained on ImageNet). This allows us to analyze whether adversarial perturbations crafted for ResNet-34 generalize to unrelated architectures—highlighting their real-world threat potential.

4.5 Implementation Details

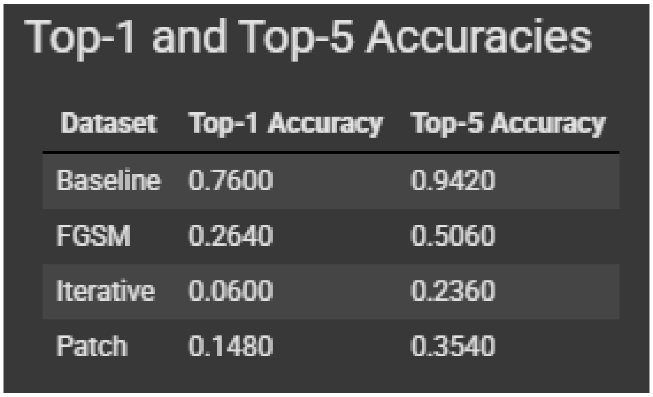
- All experiments were conducted using PyTorch on GPU-accelerated hardware.
- Adversarial sets were generated and saved as PyTorch `.pt` files for reuse in transfer evaluations.
- Gradient calculations, clipping, and perturbation constraints were implemented using low-level PyTorch tensor operations for reproducibility.

5 Results and Discussion

To assess the effectiveness of our adversarial attack pipelines, we evaluate model accuracy on clean and adversarial test sets. The frozen ResNet-34 classifier serves as the primary model under attack, while DenseNet-121 is used for evaluating cross-architecture transferability. Results are reported using Top-1 and Top-5 accuracy metrics.

5.1 Classification Performance Degradation

Figure 1 summarizes the Top-1 and Top-5 accuracy values for each dataset variant. The baseline model achieves 76.00% Top-1 accuracy on unperturbed inputs, with 94.20% Top-5 accuracy. Adversarial attacks significantly degrade this performance, demonstrating the model's susceptibility to small or localized perturbations.



Top-1 and Top-5 Accuracies

Dataset	Top-1 Accuracy	Top-5 Accuracy
Baseline	0.7600	0.9420
FGSM	0.2640	0.5060
Iterative	0.0600	0.2360
Patch	0.1480	0.3540

Figure 1: Top-1 and Top-5 classification accuracies across clean and adversarial datasets.

5.2 Quantitative Comparison Across Attacks

Figure 2 visualizes the accuracy drop caused by each attack. The iterative attack is the most damaging, reducing Top-1 accuracy to just 6.00% and Top-5 accuracy to 23.60%. Patch-based attacks, though spatially constrained, still result in significant performance degradation. FGSM, being single-step, is weaker than the iterative variant but still demonstrates the model's fragility.

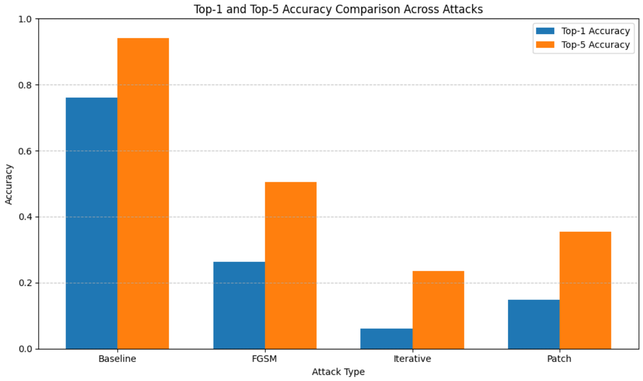


Figure 2: Bar chart comparing Top-1 and Top-5 accuracy across attack types.

5.3 Discussion of Attack Efficacy

- **FGSM:** Quick to compute but relatively less effective, with a Top-1 accuracy of 26.40%. The attack perturbs in the direction of maximum loss gradient but lacks iterative refinement.

- **Iterative:** Significantly more effective due to multi-step updates and tighter control of perturbation within the ϵ-ball.
- **Patch:** Despite operating on only a small region of the image, this attack demonstrates moderate effectiveness and simulates stealthy, localized corruption.

5.4 Transferability to DenseNet-121

All adversarial datasets were also evaluated on DenseNet-121 to assess transferability. The attacks caused consistent accuracy drops, confirming that adversarial perturbations are not model-specific and pose risks in black-box scenarios. This highlights the need for robust defenses that generalize across architectures.

6 Conclusion

In conclusion, our work demonstrates the ease with which modern deep learning models can be compromised through simple gradient-based input perturbations. By leveraging a frozen ResNet-34 classifier and integrating three adversarial attack pipelines FGSM, iterative (PGD-style), and patch based we systematically degraded Top-1 and Top-5 classification performance on ImageNet inputs.

Our experiments confirm that even small L_∞-bounded perturbations can cause catastrophic misclassifications, with accuracy dropping from 76.00% to as low as 6.00%. Additionally, the transferability of these attacks to a DenseNet-121 model underscores the systemic vulnerability of deep models across architectures.

This study highlights the critical need for adversarial robustness in real-world deployment scenarios. Future directions include adversarial training, certified defenses, and black-box attack extensions using surrogate models.

7 Acknowledgments

We would like to thank Professor Chinmay Hegde for their guidance throughout this project. We also express our appreciation to the teaching staff and peers at New York University for their feedback and collaborative discussions. Finally, we acknowledge the computational resources that enabled large-scale testing and adversarial evaluations on pretrained ImageNet models.

References

[1] I. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, arXiv preprint arXiv:1412.6572, 2015.

[2] A. Kurakin, I. Goodfellow, and S. Bengio, *Adversarial Machine Learning at Scale*, arXiv preprint arXiv:1611.01236, 2017.

[3] PyTorch Vision Models, *TorchVision Documentation*, https://pytorch.org/vision/stable/models.html.

[4] Papers with Code, *Adversarial Attacks Benchmark*, https://paperswithcode.com/task/adversarial-attacks.