FGSM, PGD and Patch Attacks: A Parameter-Driven Adversarial Evaluation on ResNet-34 and DenseNet-121

Xiaoyan Ouyang, Jingwen Lu, Chenlin Liang
https://github.com/Stephenieoo/Deep-Learning-Project-25-Spring-NYU
New York University, Tandon School of Engineering

Abstract

This project investigates the vulnerability of state-of-the-art image classifiers to adversarial attacks. We focus on ResNet-34 trained on ImageNet-1K and conduct a sequence of adversarial manipulations, including L_∞ pixel-wise (FGSM), multi-step gradient methods, and patch-based attacks. The best perturbation result on **ResNet-34** achieved **0.0% Top-1** and **9.6% Top-5** accuracy using the **PGD** attack. In comparison, the best perturbation result on **DenseNet-121** achieved **62.8% Top-1** and **88.6% Top-5** accuracy using the **FGSM** attack. We also examine the transferability of these attacks to other architectures, such as DenseNet-121. The study demonstrates how minimal perturbations can substantially impact model reliability and emphasizes the importance of adversarial robustness in model deployment.

Introduction

Deep neural networks achieve state-of-the-art performance on visual classification tasks but remain susceptible to adversarial examples—inputs carefully perturbed to cause misclassification. In this project, we implement and compare multiple adversarial strategies against a pre-trained ResNet-34 model on the ImageNet dataset. Our goal is to degrade classification performance while ensuring perturbations are imperceptible or localized.

Dataset

To evaluate adversarial robustness, we used a clean subset of the ImageNet-1K dataset consisting of 100 classes, organized into class-specific folders. We implemented a custom PyTorch dataset loader that maps WordNet IDs (WNIDs) to class indices using a pre-defined dictionary. Each image was loaded with the PIL library and converted to RGB.

Before feeding images into the model, we applied standard ImageNet normalization with the following channel-wise statistics:

- Mean: $[0.485,\ 0.456,\ 0.406]$
- Standard deviation: $[0.229,\ 0.224,\ 0.225]$

The transformation pipeline was defined using `torchvision.transforms`, consisting of:

1. Conversion to PyTorch tensors using `ToTensor()`;
2. Normalization using the above ImageNet statistics.

The data set (Figure 1) was loaded using our custom ImageNetSubsetDataset class and wrapped in a DataLoader with a batch size of 25 for efficient testing.

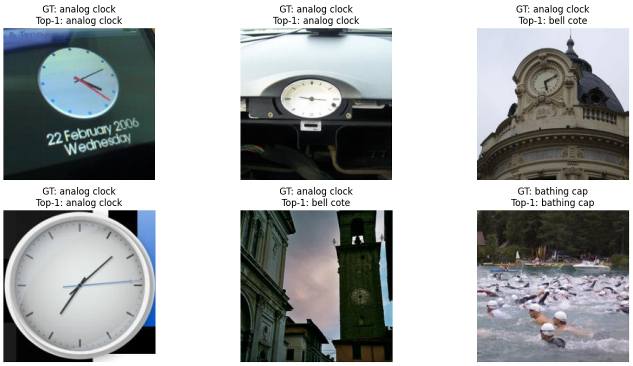

Figure 1: Original datasets.

Model

ResNet-34

We begin by establishing a baseline performance using a pretrained **ResNet-34** model on a clean subset of the ImageNet-1K dataset (100 classes). As shown in Table 1, ResNet-34 is a convolutional neural network composed of 34 layers, structured into four stages of residual blocks with skip connections. It includes an initial 7×7 convolution, max pooling, followed by:

- 3 residual blocks with 64 filters (`Conv2_x`),
- 4 blocks with 128 filters (`Conv3_x`),
- 6 blocks with 256 filters (`Conv4_x`),
- 3 blocks with 512 filters (`Conv5_x`),

ending in a global average pooling layer and a 1000-class fully connected layer.

DenseNet-121

DenseNet-121 is a convolutional neural network architecture characterized by **dense connections** between layers.

Table 1: ResNet-34 Architecture (Simplified)

Layer	Output Size	Configuration
Conv1	112×112	7×7, 64, stride 2
MaxPool	56×56	3×3, stride 2
Conv2_x	56×56	$3 \times (3 \times 3,\ 64)$
Conv3_x	28×28	$4 \times (3 \times 3,\ 128)$
Conv4_x	14×14	$6 \times (3 \times 3,\ 256)$
Conv5_x	7×7	$3 \times (3 \times 3,\ 512)$
AvgPool	1×1	Global AvgPool
FC	1×1	1000-d (ImageNet classes)

Unlike traditional CNNs where each layer has its own set of inputs, in DenseNet every layer receives input from all previous layers, improving information flow and gradient propagation.

The DenseNet-121 model comprises:

- An initial convolution and pooling layer.
- **Four dense blocks**, each containing several densely connected convolutional layers.
- **Three transition layers** that perform down-sampling between dense blocks.
- A final classification layer with global average pooling followed by a fully connected layer.

Each dense block ensures that the l^{th} layer receives feature-maps from all preceding layers, i.e.,

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])$$

where $[x_0, \dots, x_{l-1}]$ denotes the concatenation of feature-maps produced by layers 0 to $l - 1$.

Adversarial Attack Methods

Fast Gradient Sign Method

To evaluate the robustness of the model against single-step adversarial attacks, we implemented the Fast Gradient Sign Method as originally proposed by Goodfellow et al. (Goodfellow, Shlens, and Szegedy 2014). The FGSM attack perturbs each pixel of an input image in the direction of the sign of the gradient of the loss with respect to the input. The perturbation is scaled by a small constant ϵ, which controls the attack strength.

The adversarial image x_{adv} is generated as:

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

where f denotes the model, $\mathcal{L}$ is the cross-entropy loss, x is the original input, and y is the ground-truth label.

We set $\epsilon = 0.02$ and applied FGSM to the entire test set. To evaluate the magnitude of the perturbation introduced, we computed the L_∞ norm of the perturbation (i.e., the maximum absolute change across all pixels per image). This was done by comparing the adversarial images and their corresponding original inputs:

$$\delta = |x_{\text{adv}} - x|, \quad L_\infty(\delta) = \max |\delta|$$

Projected Gradient Descent

To further evaluate model robustness under stronger adversarial conditions, we implemented the Projected Gradient Descent (PGD) attack—a multi-step extension of FGSM (Madry et al. 2018). Unlike FGSM, which performs a single perturbation step, PGD iteratively updates the adversarial image and projects it back into the allowed ϵ-ball around the original input to enforce a perturbation constraint.

The adversarial image at each iteration i is computed as:

$$x_{\text{adv}}^{i+1} = \Pi_{\mathcal{B}_\epsilon(x)} \left(x_{\text{adv}}^{i} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(f(x_{\text{adv}}^{i}), y)) \right)$$

where:

- $\mathcal{L}$ is the cross-entropy loss,
- α is the step size,
- $\Pi_{\mathcal{B}_\epsilon(x)}$ is the projection operator to the L_∞ ball of radius ϵ around the original image x.

In our experiment, we set $\epsilon = 0.02$, $\alpha = 0.005$, and used 10 iterations. The attack was applied to all test images. The L_∞ norm of each perturbation was computed to evaluate the distortion level.

Experiments and Results

We evaluated the robustness of two convolutional neural network architectures—**ResNet-34** and **DenseNet-121**—under various adversarial attack scenarios on a subset of the ImageNet-1K dataset. For each model, we report Top-1 and Top-5 classification accuracy on the clean (unaltered) test set, as well as after applying three attack methods: **FGSM**, **PGD**, and **Patch-PGD**.

ResNet-34 Results

As shown in Table 2, PGD yielded the most severe accuracy degradation, completely collapsing ResNet-34's classification ability under the $\epsilon = 0.02$ constraint. FGSM also significantly reduced performance, though to a lesser extent. The Patch-PGD attack, despite being limited to a 32×32 region, also caused a notable drop in accuracy. These results show a significant degradation in performance, confirming that the model is highly vulnerable to iterative adversarial attacks even under small, imperceptible perturbations. The runtime for Patch-PGD was substantially higher due to localized iterative updates.

We visualize the effects of different adversarial attacks in Figure 2, Figure 3, and Figure 4, corresponding to FGSM, PGD, and a localized 32×32 patch attack, respectively. Each figure shows side-by-side comparisons of the original image, the adversarial image, and the perturbation itself.

To facilitate human interpretation, the perturbation is amplified using a scaling factor (×20) to make the normally imperceptible changes visible. These visualizations reveal the nature and intensity of perturbations under different attack strategies.

Table 2: ResNet-34 Accuracy Under Adversarial Attacks

Condition	Top-1 Acc.	Top-5 Acc.	Time (s)	Notes
Clean	75.60%	93.60%	2.90	Baseline performance
FGSM ($\epsilon = 0.02$)	6.00%	35.20%	4.63	One-step attack
PGD ($\epsilon = 0.02$, $\alpha = 0.005$, 10 iters)	**0.00%**	**9.60%**	27.82	Strong multi-step attack
Patch-PGD ($\epsilon = 0.5$, $\alpha = 0.04$, 100 iters, 32×32)	11.60%	48.20%	266.11	Localized perturbation

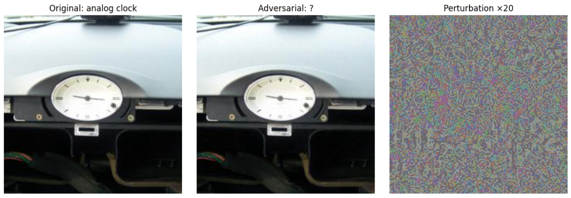

Figure 2: Visualization of ResNet-34 predictions under clean input (left), FGSM adversarial example (center), and perturbation (right).

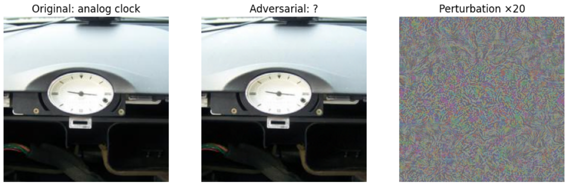

Figure 3: Visualization of PGD adversarial.

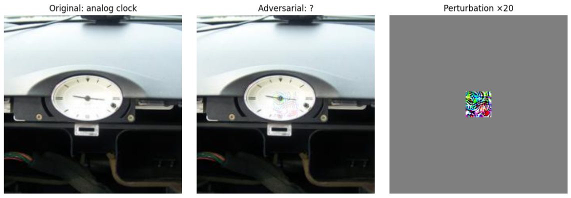

Figure 4: Visualization of Patch-PGD adversarial.

DenseNet-121 Results

Table 3: DenseNet-121 Accuracy Under Adversarial Attacks

Condition	Top-1 Acc.	Top-5 Acc.
Clean	74.40%	92.80%
FGSM ($\epsilon = 0.02$)	62.80%	88.60%
PGD ($\epsilon = 0.02$)	63.00%	90.00%
Patch-PGD	69.80%	90.80%

To evaluate the transferability of adversarial examples, we tested all three adversarial test sets—generated using FGSM, iterative FGSM (PGD), and patch-based PGD—on a different pretrained model: DenseNet-121. This allows us to assess whether attacks crafted on ResNet-34 remain effective against another architecture.

In contrast to ResNet-34, DenseNet-121 showed a much smaller accuracy drop across all attack methods under the same ϵ constraint, which aligns with prior findings on model-specific gradient sensitivity (Liu et al. 2017). As shown in Table 3, PGD and FGSM barely reduced Top-1 accuracy by more than 10%, suggesting that the DenseNet architecture may inherently offer better gradient smoothing or feature redundancy. Even the Patch-PGD attack resulted in only a minor decline in performance, highlighting its stronger robustness under standard attack settings.

Hyperparameter Exploration

We experimented with various values of ϵ, α, and iteration count to analyze their effect on attack effectiveness and runtime:

- **Smaller** ϵ (e.g., 0.01–0.02) preserved image quality but often failed to fool robust models like DenseNet-121.
- **Larger** ϵ (e.g., 0.3–0.5) ensured stronger attacks but resulted in visible perturbations or unrealistic modifications.
- **Higher iteration counts** (e.g., > 50) improved attack success, especially in Patch-PGD, but greatly increased computational cost (e.g., > 200 s per attack).

Ultimately, we selected values that offered a trade-off between attack strength and runtime, especially when working under time constraints for batch evaluation.

Summary and Insights

- **ResNet-34** is significantly more vulnerable to PGD and FGSM attacks, especially under small ϵ.
- **DenseNet-121** maintains relatively high accuracy under identical attack parameters, indicating stronger adversarial robustness.
- **Patch-PGD** attacks are computationally expensive but can degrade performance even when perturbations are limited to small image regions.
- **Transferability and model architecture matter**: identical adversarial examples have very different effects depending on the model.

Lessons Learned

Several insights emerged from the design and testing process, many of which were directly shaped by empirical observations and iterative attack refinements.

- **Attack effectiveness is highly sensitive to both model architecture and hyperparameter tuning.** Through experimentation, we found that ResNet-34 was extremely vulnerable under PGD: using $\epsilon = 0.02$, $\alpha = 0.005$, and 10 iterations was sufficient to reduce Top-1 accuracy from 75.60% to 0.00%. In contrast, DenseNet-121 remained relatively robust under the same PGD setting, retaining a Top-1 accuracy above 63%. This demonstrates how architectural differences such as dense connectivity in DenseNet can provide implicit gradient smoothing or feature redundancy that hinders attack effectiveness.
- **FGSM and PGD respond differently to ϵ changes.** FGSM achieved rapid degradation (Top-1 dropped to 6% on ResNet-34), but required precise ϵ tuning. PGD, on the other hand, allowed finer control via step size α and iteration count, and was more effective overall in bypassing local minima and causing misclassifications.
- **Patch attacks require higher ϵ and longer iterations to succeed.** Unlike full-image attacks, localized Patch-PGD required us to increase ϵ up to 0.5 and run 100 iterations to observe a significant drop in Top-1 accuracy (down to 11.60%). Despite being restricted to a 32×32 region, the attack was surprisingly effective on ResNet-34, highlighting the high vulnerability of models to small, focused changes. However, runtime was a major bottleneck, with each patch attack taking over 4 minutes.
- **Visual similarity does not imply robustness.** In multiple examples, adversarial images were visually indistinguishable from the originals, especially for FGSM and PGD. However, these minor perturbations were sufficient to completely change model predictions, illustrating the disconnect between human and machine perception.
- **Transferability of adversarial examples varies significantly across models.** Adversarial examples generated for ResNet-34 often failed to degrade DenseNet-121 to the same extent. This suggests that model-specific gradients and internal representations play a key role in how susceptible a network is to adversarial examples. Thus, robust evaluation should always consider multiple architectures.

These findings suggest that building truly robust models requires not just high clean-data accuracy, but thoughtful architecture selection, training strategies that include adversarial data, and evaluations that cover multiple types of attacks. In particular, localized and transferable attacks remain a critical threat that must be addressed in future work.

References

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2017. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations (ICLR)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.