# Deep Learning Effective Adversarial Attacks

**Jesse Noppe-Brandon and Weijun Huang**

New York University
CS 6923
{jn2934, wh2531}@nyu.edu

## Abstract

This study investigates the effectiveness of adversarial attacks designed to deceive deep learning models while remaining subtle or imperceptible to human observers. We evaluate the robustness of ResNet-34 on a 100-class subset of the ImageNet-1k dataset (class indices 401–500) using three types of attacks: Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and patch-based PGD attacks with a 32×32 patch size, both in targeted and untargeted forms. Attack performance is measured using top-1 and top-5 accuracy metrics. Results show that PGD is the most effective, reducing ResNet-34's top-1 accuracy to 0.2% and top-5 to 2%, significantly outperforming other attacks. Targeted patch attacks demonstrate strong effectiveness depending on the target class and patch location. We test the same adversarial examples on a separate DenseNet-121 model to evaluate transferability. Unlike ResNet-34, DenseNet-121 exhibits strong resistance to all attacks, with even the most effective (FGSM) maintaining over 65% top-1 accuracy. This highlights the impact of architectural design on model robustness and the need for architecture-aware attack strategies.

https://github.com/jessenb16/Adversarial_Attack

## Introduction

Deep neural networks have achieved state-of-the-art performance on many computer vision tasks, yet they remain surprisingly fragile to carefully crafted, human-imperceptible perturbations known as adversarial examples. This vulnerability poses serious risks for safety-critical applications such as autonomous driving and medical imaging. In this work, we systematically study three families of $L_\infty$-bounded attacks—pixel-wise single-step (FGSM), iterative multi-step (PGD), and localized patch attacks—against a ResNet-34 model on a 100-class subset of ImageNet (indices 401–500).

We evaluate FGSM and PGD under the same budget ($\varepsilon = 0.02$), measuring drops in Top-1 and Top-5 accuracy, verifying the perturbation magnitude, and visualizing representative failures. For patch attacks, we restrict perturbations to a $32 \times 32$ region and allow a larger budget (e.g. $\varepsilon = 0.5$) to ensure efficacy. To assess transferability, we also test all adversarial sets on a held-out DenseNet-121. Our main contributions are:

- **Baseline**: report clean Top-1/Top-5 accuracy on 500 test images.
- **FGSM**: generate "Adversarial Test Set 1" and demonstrate a substantial accuracy drop.
- **PGD**: craft "Adversarial Test Set 2" via multi-step attacks, driving Top-1 accuracy close to zero.
- **Patch Attacks**: apply adversarial patches ($32 \times 32$) to create "Adversarial Test Set 3," and quantify its impact.
- **Transferability**: evaluate all adversarial sets on DenseNet-121, revealing consistent cross-model vulnerabilities.

## Methodology

In this section we describe the three families of $L_\infty$-bounded attacks we employ: single-step FGSM, multi-step PGD, and localized PGD patch attacks.

### FGSM Attack

The Fast Gradient Sign Method (FGSM) generates a one-shot perturbation in raw pixel space while matching the network's training normalization. Let

$$x \in [0,1]^{3\times H\times W}$$

be the original image and $y$ its true label. Inside the forward pass we normalize

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

compute the loss gradient

$$g_{\text{norm}} = \nabla_{x_{\text{norm}}} \mathcal{L}\big(f(x_{\text{norm}}), y\big),$$

and recover the raw-pixel gradient by

$$g = \frac{g_{\text{norm}}}{\sigma}.$$

We then take a single FGSM step in pixel space:

$$x_{\text{adv}} = \text{clip}\big(x + \varepsilon\,\text{sign}(g),\, 0,\, 1\big),$$

with $\varepsilon = 0.02$. This guarantees $\|x_{\text{adv}} - x\|_\infty \le \varepsilon$ and, by clamping into $[0,1]$, produces valid PNG images that remain visually indistinguishable from the originals.

### PGD Attack

Projected Gradient Descent (PGD) extends FGSM by iterating many small steps under the same total budget. PGD is harder to defend against because it repeatedly "refines" the attack instead of attacking in a single step. At each iteration $t = 0, \ldots, T-1$, we:

Normalize for the model's input as we did in FGSM, compute the loss gradient in respect to the input and recover the gradient by dividing out, $\sigma$.

We then take a small FGSM-style step of size $\alpha$:

$$x' = x^{(t)} + \alpha \operatorname{sign}\Big(\tfrac{1}{\sigma} \nabla_{x^{(t)}_{\text{norm}}} \mathcal{L}\Big).$$

To enforce the $L_\infty$ budget, we project $x'$ back into the $\ell_\infty$ ball of radius $\varepsilon$ centered at the original image,

$$x^{(t+1)} = \operatorname{clip}\big(x',\ x - \varepsilon,\ x + \varepsilon\big),$$

and finally clamp each pixel into $[0, 1]$ to guarantee valid image values. We then save these perturbed images directly as PNGs in raw-pixel space.

### PGD Patch Attack

We restrict PGD to a $32 \times 32$ patch by:

- **Patch selection:** Randomly sample patch coordinates $(x, y)$ so that the $32 \times 32$ region lies fully inside the image.
- **Variants:**
  - *Untargeted*: optimize the patch to cause any misclassification, reducing confidence in the true class and pushing the model toward any other label.
  - *Targeted*: specify a fixed target class and optimize the patch to force the model to predict that class regardless of the true label.
- **Initialization:** Copy the full raw-pixel image $x$ to $x^{(0)}$ and record the patch location.
- **Iterative update (for $t = 0, \ldots, T-1$):**
  - Normalize $x^{(t)}$ and compute the loss gradient $g^{(t)}_{\text{norm}} = \nabla_{x^{(t)}_{\text{norm}}} \mathcal{L}$, where for targeted attacks the loss is taken with respect to the target label, and for untargeted attacks with respect to the true label.
  - Recover raw-pixel gradient $g^{(t)} = g^{(t)}_{\text{norm}} / \sigma$.
  - *Patch step:* update only the $32 \times 32$ region via
  $$x'_{\text{patch}} = x^{(t)}_{\text{patch}} + \alpha \operatorname{sign}\big(g^{(t)}_{\text{patch}}\big).$$
  - *Projection:* clip the patch perturbation to $[-\varepsilon, \varepsilon]$ around its original values.
  - *Clamp:* merge the patch back into $x^{(t)}$, then clip the full image to $[0, 1]$ and set $x^{(t+1)}$.
- **Saving:** after $T$ steps, save each perturbed image as a PNG in raw-pixel space.

## Experiments

In this section we describe the setup and results for each attack. All experiments use the *Cross-Entropy Loss* as the loss function.

### Data and Preprocessing

ImageNet-1K contains 1,000 object categories, but for this study, we evaluated our models on a focused 500-image test set drawn from 100 selected classes (indices 401–500). These images were organized using the standard ImageFolder directory structure. We chose this subset to reduce complexity in label handling and narrow our analysis's scope.

Each image is associated with a class label based on the ImageNet index. Still, rather than using the complete label mapping, we constructed two simplified dictionaries: one mapping ImageNet indices (401–500) to human-readable class names, and another mapping local class IDs (0–99) to their corresponding ImageNet indices.

The images were not preprocessed with the standard ImageNet normalization

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

before being loaded. Instead, they were converted to tensors and normalization was applied dynamically during the forward pass of each attack. Finally, the processed dataset was loaded into a PyTorch DataLoader with a batch size of 32 and no shuffling.

### Baseline Evaluation

We first evaluated ResNet-34's performance on the clean 500-image test set. For each batch of images, we:

1. Ran the network to obtain 1,000 logits per image.
2. Selected the logits corresponding to the 100 target classes (indices 401–500).
3. Computed Top-1 and Top-5 accuracy within this subset and mapped predictions back to the original ImageNet labels.

The resulting baseline accuracy was:

$$\text{Top-1 Accuracy} = 89.60\%, \quad \text{Top-5 Accuracy} = 99.40\%$$

Because this task involves only 100 classes (instead of the full 1,000), the evaluation is less challenging, which explains the higher accuracy compared to ResNet-34's typical performance on full ImageNet (around 76% Top-1 and 94.2% Top-5). This subset also simplifies label mapping and focuses the evaluation on a well-defined target domain, enabling more precise measurement of adversarial effects.

### FGSM Attack (Adversarial Test Set 1)

The images were all generated using $\epsilon = 0.02$ to create visually imperceptible differences. The images were evaluated the same way as the baseline images and on the 500-image subset, it achieves

$$\text{Top-1 accuracy} = 20.60\%, \quad \text{Top-5 accuracy} = 47.20\%.$$

| | Top-1 | Top-5 |
|---|---|---|
| Clean Test Set | 89.60% | 99.40% |
| FGSM ($\varepsilon = 0.02$) | 20.60% | 47.2% |
| Absolute drop | 69.00% | 52.20% |

**Table 1:** Clean vs. FGSM accuracies on ResNet-34 ($\varepsilon = 0.02$).

Table 1 shows that FGSM reduces accuracy dramatically.

Visual inspection of several failure cases revealed that FGSM often assigns high confidence to incorrect labels. When multiple classes are semantically similar, the true label may still appear in the top-5, but with much lower confidence.

## PGD Attack (Adversarial Test Set 2)

In this task, we experimented with different values of the number of iterations $T$ and step size $\alpha$, while keeping the perturbation budget fixed at $\varepsilon = 0.02$. We observed that increasing the number of iterations consistently reduced model accuracy, with diminishing returns beyond a certain point. The lowest accuracy was achieved with $T = 100$ and $\alpha = 0.005$, indicating that this combination balances attack strength and computational efficiency. We tested $\alpha$ values in the range of 0.003 to 0.008, and found that $\alpha = 0.005$ provided the most effective perturbation without overshooting or underutilizing the perturbation budget.

Table 2 highlights the difference between FGSM and PGD attacks. While FGSM applies a single-step update, PGD uses multiple iterative steps to generate stronger perturbations. As a result, PGD is significantly more effective, reducing Top-1 accuracy to just 0.2% and Top-5 accuracy to 2%, leading to a near-complete collapse in model performance.

We observed that PGD often assigns a near 100% confidence on it's Top-1 prediction with the correct label rarely even being in the Top-5.

| | Top-1 | Top-5 |
|---|---|---|
| Clean Test Set | 89.60% | 99.40% |
| PGD ($\varepsilon = 0.02,\ \alpha = 0.005,\ T = 100$) | 0.20% | 2.00% |
| Absolute drop | 89.40% | 97.40% |

**Table 2:** Clean vs. PGD accuracies on ResNet-34.

## PGD Patch Attack (Adversarial Test Set 3)

To further examine the robustness of the model, we applied a $32 \times 32$ patch on each image and varied the attack budget $\varepsilon$, step size $\alpha$, and number of iterations $T$. In particular, we compared:

- **Untargeted patch** with $\varepsilon = 0.50,\ \alpha = 0.06,\ T = 200$,
- **Targeted patch** (all images forced toward class 401) with $\varepsilon = 0.50,\ \alpha = 0.06,\ T = 200$.

| | Top-1 | Top-5 |
|---|---|---|
| Clean Test Set | 89.6% | 99.4% |
| PGD-Patch Untargeted | 34.6% | 70.6% |
| PGD-Patch Targeted | 11.8% | 86.6% |
| Absolute drop (Untargeted) | 55.0% | 28.8% |
| Absolute drop (Targeted) | 77.8% | 12.8% |

**Table 3:** Clean vs. PGD-Patch accuracies on ResNet-34.

From Table 3, we found that increasing the patch budget from $\varepsilon = 0.02$ to $\varepsilon = 0.50$ produced a dramatic additional drop in Top-1 accuracy: down to 34.6% for the untargeted patch and to 11.8% for the targeted version. Although only 2% of each image's pixels were modified, the larger budget and directional constraint significantly strengthened the attack. The attacks performed better on decreasing Top-1 predictions than FGSM, which perturbed the entire image.

We also tuned the step size and chose $\alpha = 0.06$—roughly scaling $\alpha$ in proportion to the larger $\varepsilon$—to balance convergence speed and perturbation stability. A larger number of iterations ($T = 200$) were required compared to standard PGD to fully exploit the patch budget.

Unlike full-image PGD (which perturbs every pixel), our patch attacks leave most of the image intact. Consequently, untargeted-patch examples often still retained the true label among their Top-5 predictions, distributing confidence across multiple plausible classes. In contrast, targeted-patch examples concentrated almost all probability mass on the single (incorrect) target class—usually relegating the true label to second place with only a few percent confidence.

Because we randomized the patch location, some placements (e.g. over background) were less effective if they missed salient object features. In early trials we also varied the target class and observed that forcing all images toward very dissimilar classes (e.g. "cliff dwelling") resulted in much higher residual accuracy (≈59%) compared to using "accordion" (≈11.8%). This suggests that semantic proximity between source and target labels can dramatically influence targeted-patch success.

## Transferability to DenseNet-121

DenseNet-121 differs from ResNet-34 in its architectural design: Each layer in DenseNet receives the feature maps from all preceding layers via concatenation. This dense connectivity promotes feature reuse and improves gradient flow, enhancing the model's robustness to small, iterative perturbations (Huang et al. 2018).

To assess transferability, we evaluated all adversarial test sets on a DenseNet-121 model pretrained on ImageNet, using the same 100-class subset (indices 401–500) and the evaluation protocol described earlier.

The result from Table 4 shows DenseNet-121 proved far more resilient than ResNet-34. FGSM and PGD attacks only reduced Top-1 by about 15–19%, compared to over 60% on ResNet-34, and patch attacks caused at most a 2.6% drop. We attribute this to DenseNet's architecture: by concatenat-

| | Top-1 | Top-5 |
|---|---|---|
| Clean Test Set | 88.0% | 98.4% |
| FGSM ($\varepsilon = 0.02$) | 68.6% | 91.6% |
| PGD ($\varepsilon = 0.02$) | 72.4% | 93.0% |
| PGD-Patch (Untargeted) | 85.4% | 97.6% |
| PGD-Patch (Targeted) | 86.4% | 97.6% |
| Absolute drop (FGSM) | 19.4% | 6.8% |
| Absolute drop (PGD) | 15.6% | 4.6% |
| Absolute drop (Untargeted) | 2.6% | 0.8% |
| Absolute drop (Targeted) | 1.6% | 0.8% |

**Table 4:** DenseNet-121 transferability results (100 classes, 401–500).

ing low-level features into every subsequent block, it preserves unperturbed information even when some layers receive adversarial noise, mitigating the effect of single-step and iterative perturbations.

Interestingly, FGSM yielded a larger drop on DenseNet-121 (from 88.0 % to 68.6 %) than PGD (to 72.4 %), reversing the pattern observed on ResNet-34. We hypothesize that FGSM's single-step, large-magnitude perturbation can bypass DenseNet's feature-aggregation defenses more effectively. In contrast, the dense connectivity partially absorbs PGD's smaller, incremental updates, making them less disruptive to the network's prediction.

## Analysis

Below we offer our reflections on the behaviors we observed.

**Sensitivity to imperceptible noise** We noticed that even very small $\ell_\infty$ perturbations (on the order of a single gray-level change) often caused ResNet-34 to misclassify. One possible explanation is that the network's decision boundary in pixel-space can be quite close to natural images, so a tiny shift aligned with the loss gradient may push an input into a different region of feature space. In practice, these small changes—though imperceptible to humans—appear to amplify through the model's layers and alter its final prediction.

**Impact of a localized patch** It was somewhat surprising that modifying only 2% of the pixels (a $32 \times 32$ patch) could still degrade accuracy significantly. We speculate that classifiers often rely on a few highly discriminative local features; if an adversarial patch overlaps one of those critical receptive fields, it may disproportionately distort the model's internal representations. Even when the patch does not cover the most salient object parts, repeated gradient updates in the patch region seem to introduce enough artifact to confuse downstream layers.

**Transferability and potential defenses** Our experiments showed that adversarial examples crafted on ResNet-34 tended to transfer to DenseNet-121 with only modest degradation in effectiveness. This suggests that different architectures trained on the same data may learn similar vulnerable regions in input-space. In our view, defenses such as adversarial training (mixing in examples from multiple attack types) or simple input transformations (e.g. random resizing or compression) could help break this alignment (Madry et al. 2019).

As these are preliminary reflections, a more systematic study would be needed to confirm which of these mechanisms dominate in practice.

## Conclusion

This study investigated the effectiveness of various adversarial attacks—FGSM, PGD, and patch-based PGD (both targeted and untargeted)—on deep image classification models. Using a 500-image test set drawn from 100 ImageNet classes, we evaluated the robustness of ResNet-34 and examined the transferability of attacks to DenseNet-121.

Our results show that ResNet-34 is highly vulnerable to iterative attacks like PGD, reducing Top-1 accuracy to as low as 0.2%. Even small, localized patches affecting only 2% of the image were sufficient to cause significant drops in accuracy. This suggests a high sensitivity of convolutional models to subtle and spatially constrained perturbations.

In contrast, DenseNet-121 demonstrated greater resilience under the same attack settings. As discussed in our analysis, this is likely due to its dense connectivity, which helps preserve low-level features and absorb minor perturbations. The results underscore the impact of architectural design on adversarial robustness and suggest that transferability remains a practical concern, even across different network types.

| Attack | ResNet-34 | | DenseNet-121 | |
|---|---|---|---|---|
| | Top-1 | Top-5 | Top-1 | Top-5 |
| Clean | 89.6% | 99.4% | 88.0% | 98.4% |
| FGSM ($\varepsilon = 0.02$) | 20.6% | 47.2% | 68.6% | 91.6% |
| PGD ($\varepsilon = 0.02$, $\alpha = 0.005$, $T = 100$) | 0.2% | 2.0% | 72.4% | 93.0% |
| PGD-Patch Untargeted ($\varepsilon = 0.50$, $\alpha = 0.06$, $T = 200$) | 34.6% | 70.6% | 85.4% | 97.6% |
| PGD-Patch Targeted ($\varepsilon = 0.50$, $\alpha = 0.06$, $T = 200$) | 11.8% | 86.6% | 86.4% | 97.6% |

**Table 5:** Summary of final Top-1/Top-5 accuracies (%) on ResNet-34 and DenseNet-121.

## References

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083.