Deep Learning - Project 3

Rahil Singhi[1], Alper Mumcular[2], Divya Srinivasan[3]
New York University
[1]rs9174@nyu.edu [2]am14533@nyu.edu [3]ds7852@nyu.edu

Abstract

In this project, we investigate the vulnerability of deep convolutional neural networks to adversarial attacks by implementing and analyzing various perturbation strategies on the *ResNet-34* model. We applied the Fast Gradient Sign Method (FGSM), Iterative FGSM, Projected Gradient Descent (PGD), and PGD-based patch attacks to generate adversarial examples. Our experiments show that even imperceptible perturbations can drastically reduce model accuracy. We also evaluated the transferability of these adversarial examples by testing them on a different model, *DenseNet-121*. Interestingly, *DenseNet-121* retained higher robustness, with accuracy ranging from 63.2% to 70.2% across adversarial datasets. Additionally, we analyzed the computational cost of each attack, finding a substantial increase in runtime for iterative and patch-based methods.
You can access the GitHub repository from this link.

Overview

ImageNet-1K Dataset

The ImageNet-1K Dataset is one of the widely used benchmark data sets in computer vision. This set includes ∼1.2 million RGB-formatted images, and 1000 object classes. Because the models trained on this data set often generalize well later on, it is widely used for deep learning in computer vision. Our project used a subset of this data set with only 100 object classes, each containing 5 images sized 224×224.

Generating Adversarial Examples

Three methods were used and mixed together to generate the perturbed images.

- **Fast Gradient Sign Method (FGSM):** Introduced by Ian Goodfellow et al. back in 2014 [1], FGSM is a light and fast method to generate adversarial examples. It alters the image in the direction that increases loss the most.

$$\eta = \epsilon \, \text{sign}(\nabla_x J(\theta, x, y)) \tag{1}$$

  In Eq. 1, perturbation η is proportional to the gradient of the loss function with respect to the input x. ϵ is a small scalar to control the size of the alteration, its smallness makes the alteration undetectable by the human eye. θ is the model parameters, and y is the true label of the input.
- **Iterative Fast Gradient Sign Method (I-FGSM):** Alexey Kurakin et. al formally described I-FGSM in 2016 [3] even though its concept was described in the original FGSM paper [1]. This method applies FGSM in smaller steps as it is the most powerful when ϵ remains smaller. As seen in Eq. 2 below, I-FGSM takes the original input and perturbs it by applying FGSM n many times. Each time the perturbation is added to the adversarial input, it is clipped to make sure to have the perturbation in expected sizes.

$$\begin{aligned} x_0^{adv} &= x, \\ x_{n+1}^{adv} &= \text{Clip}_{x,\epsilon}\{x_n^{adv} + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_n^{adv}, y))\} \end{aligned} \tag{2}$$

  This way, I-FGSM creates stronger and more precise attacks than the single-step FGSM.
- **Projected Gradient Descent (PGD):** Introduced as a building block on FGSM in 2018 [2], PGD is more robust than its predecessor. While FGSM lacked in larger ϵ scalars, PGD uses its strength of small scalars and iterates over small FGSM-like steps. This way, PGD is able to be robust and rigorous. It is able to uncover any weaknesses might have been missed by smaller attacks.

Findings

- The FGSM attack, despite being the fastest (9.77s), drastically dropped ResNet-34's Top-1 accuracy from 76.0% to 6.0%.
- Iterative attacks (I-FGSM, PGD) completely broke ResNet-34, reducing Top-1 accuracy to 0.0%, but took over 165 seconds to run.
- PGD Patch attack had the highest runtime (267.21s), but still retained 27.8% Top-1 accuracy, likely due to its spatially localized nature.
- Adversarial examples transferred partially to DenseNet-121, which maintained much higher accuracy (63.2–70.2% Top-1).
- Top-5 accuracy consistently remained higher than Top-1 across all attacks, indicating that adversarial perturba-

tions often displace the correct label rather than completely eliminating it from the top predictions.

- Visualization showed that even imperceptible perturbations can lead to severe misclassifications.

Methodology

Preprocessing and Evaluation Metrics

Normalization Standard preprocessing was applied using `torchvision.transforms`. Each image was converted to a tensor and normalized using ImageNet's per-channel statistics: mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225].

Evaluation Metrics We measured model performance using Top-1 and Top-5 accuracy. Each model was evaluated in inference mode using a batch size of 32. Top-k accuracy was computed using `torch.topk()` on softmax-normalized predictions.

Adversarial Attack Implementation

Fast Gradient Sign Method (FGSM) FGSM is a one-step attack that perturbs input pixels in the direction of the loss gradient, computed using cross-entropy loss between the model's output and the true label. The adversarial image is computed as:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

We used $\epsilon = 0.02$ to ensure the perturbation was imperceptible.

Iterative FGSM (I-FGSM) I-FGSM extends FGSM by applying multiple small updates. With $\alpha = 0.008$ and 25 iterations, we clipped each update to stay within $\epsilon = 0.02$ using Eq. 2.

Projected Gradient Descent (PGD) PGD builds on I-FGSM by explicitly projecting the perturbed input back into the ϵ-ball around the original input after each step. It uses the same ϵ and α values as I-FGSM but reinitializes gradients at every step for greater attack rigor.

PGD Patch Attack This is a localized variant of PGD that modifies a 32×32 patch within the image, rather than the full image. It uses $\epsilon = 0.5$ and $\alpha = 0.03$, focusing stronger perturbations on a smaller region to analyze spatial sensitivity.

Performance Evaluation

Accuracy Measurement Top-1 and Top-5 accuracy were computed after each attack using the same evaluation pipeline. Results were compared to baseline performance on the clean dataset.

L_∞ Distance We computed the maximum pixel-wise absolute difference between original and adversarial images using the L_∞ norm to quantify perturbation intensity.

Runtime Tracking Total attack time was measured using Python's `time.time()` function to evaluate computational cost across attack methods.

Visualization We visualized adversarial examples, perturbation maps, and top-5 prediction bars for misclassified images to qualitatively assess attack strength and model confidence shifts.

Design Lessons and Trade-offs During the design and testing of adversarial attacks, we explored various hyperparameter configurations and observed key trade-offs:

- **FGSM Pros:** Fast and easy to implement; good for quick attack baselines.
- **FGSM Cons:** Less effective; higher ϵ values needed to succeed, which risk visual detectability.
- **I-FGSM / PGD Pros:** Significantly more effective at fooling models even with low ϵ; allows precise control via α and iteration count.
- **I-FGSM / PGD Cons:** Much slower to compute; may overfit to source model and reduce transferability.
- **PGD Patch Pros:** Explores spatially constrained vulnerabilities; easier to hide in real-world settings.
- **PGD Patch Cons:** Highest runtime; less transferable due to model-specific receptive fields.

Lessons Learned:

- Small ϵ values (0.02) were sufficient if paired with enough iterations.
- Larger α values led to quicker degradation but increased risk of exceeding the perturbation budget.
- Transferability was limited, reminding us that overfitting to one model's gradients can weaken attacks on others.

Transferability Testing

Observations DenseNet-121 demonstrated higher robustness to transferred attacks, with Top-1 accuracy ranging from 63.2% to 70.2%. The PGD Patch attack showed the least transferability, likely due to its spatial localization. Differences in architecture, such as dense connectivity and gradient flow, likely contributed to reduced cross-model attack effectiveness.

Results

Perturbation Size

Perturbation size played in important role in this project. It determined the success of an attack and the imperceptibility of a perturbed image. Pixel-wise attacks FGSM and I-FGSM, updated the whole image size 224×224 (Table 1) while perturbing the pixels. On the other hand, patch-wise attacks focused on one 32×32 sized area (Table 1). While this allowed for a larger perturbation constant, up to $\epsilon = 0.5$, the perturbation might be visible to the keen human eye, as seen in Fig. 1.

Runtime

FGSM, with its single gradient step, was the fastest method at 9.77 seconds (Table 2). In contrast, iterative attacks involving multiple optimization steps such as I-FGSM and PGD took approximately 165–170 seconds (Table 2), reflecting their higher computational complexity. PGD Patch,

Task	Method	Perturbation Size
Task 1	Original	224×224
Task 2	FGSM	224×224
Task 3	I-FGSM / PGD	224×224
Task 4	PGD Patch	32×32

Table 1: **Task and Perturbation Details** The perturbation size created by the methods used in every task.

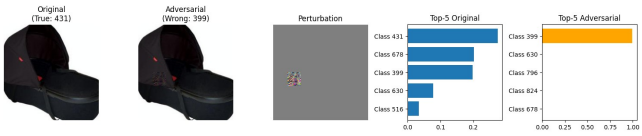

Figure 1: **Example of Misclassifications with PGD-patch**

despite modifying only a small region, had the highest runtime at 267.21 seconds (Table 2) due to larger iterations to get reasonable results.

Task	Runtime (s)
FGSM	9.77
Iterative FGSM	168.61
PGD	164.52
PGD Patch	267.21

Table 2: Runtime comparison for different adversarial attack methods.

Top-1 Accuracy

While the attacks were very effective, patch-wise attacks were less catastrophic than pixel-wise attacks, seen as PGD Patch that still has 27. 8% accuracy (Fig. 2). We also observe that transferability reduces the effectiveness the attacks. Nevertheless, the accuracies obtained on DenseNet-121 (Fig. 2) are still threats, especially in black-box situations.

Top-5 Accuracy

Top-5 accuracy resulted in more nuanced view of the models. As seen in Fig. 3, Top-5 accuracy remained non-zero across the attacks, indicating that the correct class label always remained in the top 5 choices the model made. Similar to Top-1 accuracy, adversarial transfer made the Top-5 accuracy to be very high.

Discussion

In this section, we will discuss the implications of our findings, particularly analyzing the runtime-performance trade-off, the role of perturbation size and iteration counts, differences between Top-1 and Top-5 accuracies across tasks, and the partial transferability of adversarial examples to a different model architecture.

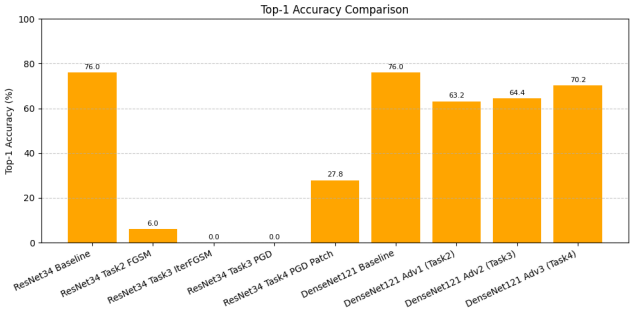

Figure 2: **Comparison of Top-1 Accuracies**

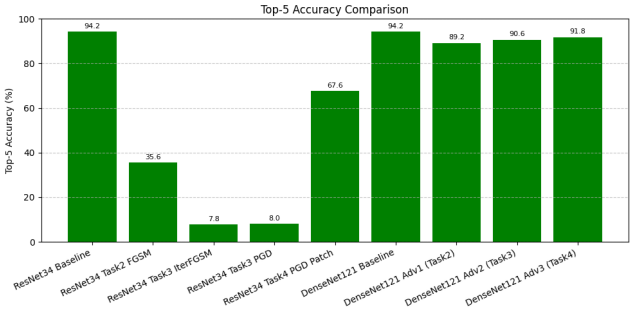

Figure 3: **Comparison of Top-5 Accuracies**

Runtime vs. Attack Strength

Our results demonstrate a clear trade-off between attack runtime and effectiveness. As shown in Table 2, the FGSM attack, being a single-step gradient-based method, completes in under 10 seconds. However, its performance degradation is limited in comparison to more sophisticated attacks. Iterative FGSM, while significantly slower (approximately 165–170 seconds), completely incapacitate the ResNet-34 model, reducing Top-1 accuracy to 0%. The PGD Patch attack is even more computationally demanding, requiring around 267 seconds, but it manages to maintain some model accuracy due to its localized nature.

This demonstrates that runtime and attack efficacy are inherently linked: stronger attacks typically require more computational resources.

Perturbation Size and Iterative Attacks

Another important observation from our study is the relationship between perturbation size (ϵ), step size (α), and the number of iterations in multi-step attacks. As we aimed to maintain a small ϵ (0.02) to ensure imperceptibility, achieving successful attacks required increasing the number of iterations or fine-tuning α.

For instance, we found that setting $\alpha = 0.008$ with 25 iterations gave the best results for both Iterative FGSM and PGD attacks. This highlights the importance of balancing perturbation strength and iteration granularity: with smaller perturbation budgets, fine-grained, repeated updates are necessary to effectively navigate the model's decision boundary.

Model	Dataset/Task	Top-1 Accuracy	Top-5 Accuracy
ResNet-34	Baseline	76.0%	94.2%
ResNet-34	Task 2: FGSM	6.0%	35.6%
ResNet-34	Task 3: I-FGSM	0.0%	7.8%
ResNet-34	Task 3: PGD	0.0%	8.0%
ResNet-34	Task 4: PGD Patch	27.8%	67.6%
DenseNet-121	Baseline	76.0%	94.2%
DenseNet-121	Adversarial Dataset 1 (from Task 2)	63.2%	89.2%
DenseNet-121	Adversarial Dataset 2 (from Task 3)	64.4%	90.6%
DenseNet-121	Adversarial Dataset 3 (from Task 4)	70.2%	91.8%

Table 3: **ResNet-34 and DenseNet-121 Performance Summary**

In contrast, FGSM performs a single large update, which is faster but less precise and more detectable.

In the case of the PGD Patch attack, although the patch size was relatively small (32×32), we set a much higher $\epsilon = 0.5$ with $\alpha = 0.03$, allowing more drastic local changes. This shows that localized attacks can compensate for their limited spatial footprint with stronger per-pixel modifications.

Top-1 vs. Top-5 Accuracy Trends

A consistent pattern observed across all tasks is that adversarial attacks impact Top-1 accuracy much more severely than Top-5 accuracy. As shown in Table 3, FGSM reduced Top-1 accuracy to 6.0%, while still maintaining a relatively higher Top-5 accuracy of 35.6%. Similarly, Iterative FGSM and PGD reduced Top-1 to 0.0% but retained Top-5 accuracy around 8%.

This suggests that while the adversarial examples succeeded in displacing the correct class from the top position, the models still often ranked it within the top 5 predictions. Therefore, Top-5 accuracy may provide a more lenient robustness measure, especially in multi-class classification tasks.

Transferability to DenseNet-121

To explore the transferability of adversarial examples, we evaluated DenseNet-121 on datasets adversarially crafted using ResNet-34. Interestingly, DenseNet-121 exhibited much higher robustness to these inputs, with Top-1 accuracy ranging from 63.2% to 70.2% across all attack datasets. This suggests only partial transferability.

Several factors may contribute to this discrepancy:

- **Architectural Differences**: DenseNet's feature reuse and dense connectivity patterns may lead to smoother decision boundaries, making it less susceptible to perturbations crafted for other models.
- **Gradient Mismatch**: Since the attacks were generated using gradients from ResNet-34, they may not align with the loss surface of DenseNet-121, reducing effectiveness.
- **Overfitting of Attacks**: Iterative attacks in particular may overfit to the source model's weaknesses, making them less transferable.

The PGD Patch attack was the least transferable, which aligns with the hypothesis that localized, high-magnitude perturbations exploit model-specific receptive fields and spatial sensitivities. This further indicates that model robustness should be evaluated not just in isolation, but also under cross-model adversarial exposure.

Mitigating Transferability: To mitigate the transferability of adversarial examples across models, several defense strategies can be employed:

- **Adversarial training using diverse models:** Train the target model with adversarial examples generated from multiple source architectures to improve general robustness.
- **Ensemble methods:** Aggregate predictions from multiple heterogeneous models to reduce the impact of perturbations that are effective on a single model.

Employing these strategies may decrease vulnerability to cross-model adversarial examples.

Summary of Key Insights

- Increased iterations and precision in multi-step attacks drastically lower model accuracy, but come at a high computational cost.
- Smaller perturbation sizes require finer step sizes and more iterations to be effective.
- Top-1 accuracy is significantly more affected by attacks than Top-5, emphasizing its role as a sensitive robustness metric.
- Adversarial examples generated for one model may only partially affect others, depending on architecture and gradient similarity.

Conclusion

This project examined the robustness of convolutional neural networks to adversarial attacks using FGSM, I-FGSM, PGD, and PGD Patch on ResNet-34. Results showed that small perturbations significantly reduced accuracy, with multi-step attacks (PGD, I-FGSM) being especially effective but computationally costly.

We also tested cross-model transferability on DenseNet-121. While some adversarial examples transferred, DenseNet-121 showed greater robustness due to architectural differences. These findings show the need for models that perform well on both clean and adversarial input.

Note: This report was generated with some assistance from GPT and DeepSeek.

References

[1] Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy. Explaining and Harnessing Adversarial Examples In *ICLR'2015, arXiv:1412.6572*, 2014.

[2] A. Madry, A Makelov, L. Schmidt, D. Tsipras, A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks In *ICLR'2018, arXiv:1706.06083*, 2017.

[3] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio. Adversarial examples in the physical world In *ICLR'2017, arXiv:1607.02533*, 2016.