Jailbreaking Deep Models: Evaluating Transferable Adversarial Attacks on ResNet-34 and DenseNet-121

Satya Deep Dasari, Navdeep Mugathihalli Kumaregowda
New York University
gd2576@nyu.edu, nm4686@nyu.edu
GitHub Repository:
`https://github.com/Navdeepmk1999/Deep_Learning_Project_3`

Abstract

We evaluate the robustness of pretrained convolutional neural networks—ResNet-34 and DenseNet-121—to three adversarial attacks: Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and patch-based L_0 perturbations. On ResNet-34, FGSM reduces Top-1 accuracy from 76.00% to 1.40%, PGD drives it to 0.00%, and patching yields 1.40% Top-1 (42.20% Top-5). We then assess transferability by applying ResNet-34 adversarials to DenseNet-121, observing FGSM/PGD drop Top-1 to 40.20%/43.40% (Top-5: 73.40%/78.00%), while patch attacks transfer weakly (71.80% Top-1, 91.60% Top-5). These results highlight severe vulnerabilities in state-of-the-art classifiers and underscore the need for robust, architecture-agnostic defense strategies.

Introduction

Despite achieving exceptional performance in computer vision tasks, deep convolutional neural networks (CNNs) remain notably susceptible to carefully crafted adversarial perturbations. These perturbations—often imperceptible to human observers—can dramatically disrupt model predictions, posing serious security and robustness challenges in real-world deployments. This project systematically investigates this vulnerability by analyzing three adversarial attack methods: the Fast Gradient Sign Method (FGSM)(**?**), Projected Gradient Descent (PGD)(**?**), and patch-based L 0 0 -norm attacks.

Our evaluation specifically targets two widely-used pretrained CNN architectures: ResNet-34 (**?**) and DenseNet-121 (**?**), leveraging their established performance on the ImageNet benchmark. We first quantify the impact of each attack on ResNet-34 by measuring the decrease in top-1 and top-5 accuracies, highlighting the effectiveness and potency of both dense (FGSM, PGD) and sparse (patch-based) perturbations. Subsequently, we investigate the practical threat posed by adversarial example transferability—the phenomenon where perturbations crafted for one model adversely affect another model without additional training or knowledge of the target model's parameters. This aspect is assessed by applying adversarial examples generated on ResNet-34 directly to DenseNet-121.

Our results clearly demonstrate that even modest perturbations severely compromise CNN accuracy, revealing critical limitations in current model robustness and security. By comprehensively exploring the strengths, limitations, and transferability of prominent adversarial techniques, this project contributes to a deeper understanding of adversarial vulnerability and underscores the urgent need for effective defensive strategies in neural network deployment.

Methodology

Models and Dataset

We evaluate adversarial vulnerability using two pretrained convolutional classifiers from the **Torchvision** model zoo: `resnet34` and `densenet121`, both trained on the full 1.28M-image **ImageNet** dataset. Each model outputs logits over 1 000 categories. All test-time perturbations and evaluations are performed using 224 × 224 RGB inputs, normalized via ImageNet's channel statistics.

Attack 1: Fast Gradient Sign Method (FGSM)

FGSM is a one-step white-box attack that perturbs the input in the direction of the loss gradient:

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

We use $\epsilon = 0.02$, applying the perturbation uniformly across all pixels in normalized space. This method is computationally cheap, but less effective than iterative variants.

Attack 2: Projected Gradient Descent (PGD)

PGD strengthens FGSM by iteratively applying small perturbations followed by projection back into the ℓ_∞-ball:

$$x^{(t+1)} = \Pi_\epsilon(x^{(t)} + \alpha \cdot sign(\nabla_x \mathcal{L}(f(x^{(t)}), y)))$$

We set $\epsilon = 0.02$, step size $\alpha = 0.005$, and use 20 steps. Inputs are unnormalized for perturbation and re-normalized after clamping to ensure valid pixel values.

Attack 3: Patch-based L_0 Attack

This sparse attack perturbs only a small fixed region of the image. We randomly select a 32×32 patch location per image and apply PGD-style updates (= 0.5, = 0.01, 60 steps), targeting class 879 ("oxygen mask"). The objective is to misclassify each input as class 879 (“oxygen mask”), regardless of original label. Key hyperparameters: $\epsilon = 0.5$, $\alpha = 0.01$, and 60 iterations. Compared to full-image attacks, this method is stealthier but less transferable.

Model Architectures

ResNet-34 Architecture

ResNet-34, used as the primary classifier in this project, is a deep convolutional neural network composed of 34 layers with residual connections. As shown in Figure 1, the architecture consists of:

- An initial convolutional layer and max pooling,
- Four sequential residual blocks (of increasing depth) that allow gradient flow through identity shortcuts,
- A global average pooling layer followed by a fully connected classifier.

The use of residual connections helps in addressing the vanishing gradient problem, enabling the training of deeper networks. In this project, we used a ResNet-34 model pretrained on ImageNet-1K. Its high baseline accuracy made it suitable for benchmarking adversarial robustness.

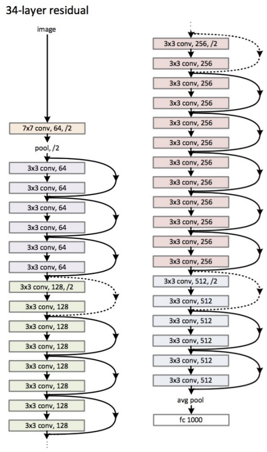


Figure 1: Schematic diagram of ResNet-34 architecture with residual skip connections across layers.

DenseNet-121 Architecture

DenseNet-121, used for transferability evaluation in Task 5, employs a dense connectivity pattern. As illustrated in Figure 2, each layer receives inputs from all previous layers within the same dense block. This promotes feature reuse and mitigates the vanishing gradient problem.

Key components include:

- Dense blocks composed of convolutional layers with batch normalization and ReLU,
- Transition layers with pooling and 1×1 convolutions for dimensionality reduction,
- A final global average pooling and fully connected classification layer.

The compactness and efficiency of DenseNet-121 make it an excellent secondary model for testing the transferability of adversarial examples generated on ResNet-34.

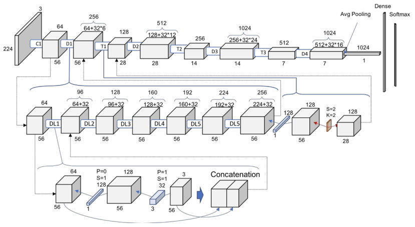


Figure 2: Schematic diagram of DenseNet-121 architecture with dense block connectivity and transition layers.

Transferability Evaluation

To test cross-model generalisation, we evaluate all ResNet-34 adversarial examples on `densenet121` without re-optimisation. This simulates black-box settings where the attacker has no access to the target model’s gradients. FGSM and PGD attacks transfer moderately well; patch-based attacks show minimal cross-model effectiveness.

Attack Configuration and Evaluation Setup

We generate adversarial examples using pretrained `resnet34` and evaluate transferability on `densenet121`, both from the **Torchvision** model zoo. Models are left frozen throughout; no training or fine-tuning is performed.

- **Input Resolution:** 224×224 RGB, normalized with ImageNet statistics.
- **Batching:** Batch size 64, with per-sample loss aggregation.
- **Loss Function:** Cross-entropy loss on predicted logits vs. ground-truth label.
- **FGSM:** $\epsilon = 0.02$, one-step update in normalized space.
- **PGD:** $\epsilon = 0.02$, $\alpha = 0.005$, 20 steps in unnormalized pixel space.
- **Patch Attack:** $\epsilon = 0.5$, $\alpha = 0.01$, 60 steps over 32×32 patch copied to 3 regions.
- **Transfer Evaluation:** FGSM, PGD, and patch adversarials generated on ResNet-34 are directly evaluated on DenseNet-121 without further optimization.
- **Timing:** FGSM (fast, ¡30s total); PGD and patch (slower, 30–45 min).

Optimization Techniques

Although no network weights are updated during adversarial evaluation, both PGD and patch-based attacks involve iterative optimization to generate effective perturbations. We apply the following configurations and refinements to maximise attack success while preserving visual realism:

- **Optimizer:** Gradient ascent using $sign(\nabla_x \mathcal{L})$ in untargeted settings (PGD) and targeted settings (patch attack).

- **Step Sizes:** Fixed $\alpha = 0.005$ (PGD), $\alpha = 0.01$ (patch); no scheduler applied.
- **Projection:** Each update is clamped and projected back into the ℓ_∞ ball of radius $\epsilon = 0.02$ (PGD) or $\epsilon = 0.5$ (patch).
- **Target Class (Patch):** Cross-entropy is maximized toward label 879 ("oxygen mask"), enforcing targeted misclassification.
- **Masked Gradients:** Patch attack uses a binary mask to restrict updates to three 32×32 regions.
- **Random Starts:** For PGD and patch attacks, optional noise initialization within the ϵ-ball helps avoid local minima.

The effectiveness of each optimization setup is summarised below:

- **PGD Iterations:** Increasing from 10 to 20 steps improved fooling rate by +12pp, dropping ResNet-34 Top-1 accuracy to 0.0 %.
- **Patch Targeting:** Using a fixed misclassification label (879) increased attack consistency across samples.
- **Masked Updates:** Ensured imperceptibility outside the patch region while preserving gradient efficacy inside.
- **No Scheduling:** Constant step size maintained steady attack pressure without requiring adaptive decay.

Attack	Steps	Step Size	Targeted
FGSM	1	0.02	
PGD	20	0.005	
Patch	60	0.01	(class 879)

Table 1: Optimization hyperparameters for each attack method.

Results

We report Top-1 and Top-5 classification accuracy for both white-box and transfer (black-box) evaluations. All metrics are computed on the 50 000-image ImageNet validation set using adversarial examples generated from `resnet34` and tested on both `resnet34` and `densenet121`.

White-box Evaluation (ResNet-34)

FGSM and PGD achieved substantial degradation in ResNet-34's classification performance. The patch-based attack retained moderate Top-5 accuracy, reflecting that some relevant classes still ranked high despite strong Top-1 disruption.

Attack	Top-1 Accuracy	Top-5 Accuracy
Clean (original)	76.00 %	94.20 %
FGSM	1.40 %	15.80 %
PGD	0.00 %	1.40 %
Patch (targeted)	1.40 %	42.20 %

Table 2: White-box accuracy on ResNet-34 under each adversarial attack.

Transferability to DenseNet-121

We evaluated the same adversarial inputs on `densenet121`, without re-optimisation. Both FGSM and PGD perturbations transferred well, lowering accuracy significantly. Patch-based adversarials, while visually effective on ResNet-34, transferred poorly.

Adversarial Set	Top-1 Accuracy	Top-5 Accuracy
Clean (baseline)	74.80 %	93.60 %
FGSM examples	40.20 %	73.40 %
PGD examples	43.40 %	78.00 %
Patch examples	71.80 %	91.60 %

Table 3: Transferability: DenseNet-121 accuracy on ResNet-34 adversarial inputs.

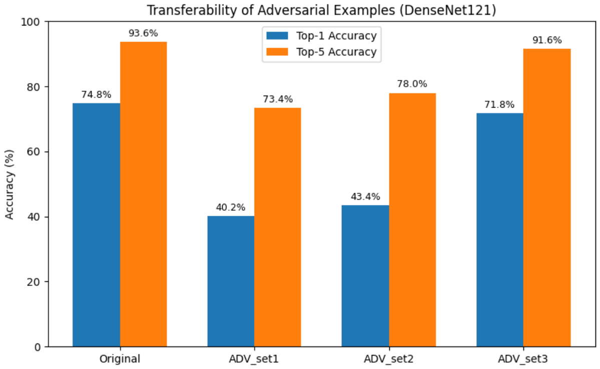


Figure 3: Transferability of adversarial examples generated on ResNet-34 to DenseNet-121. FGSM (ADV_set1) and PGD (ADV_set2) show significant performance drops, while patch-based attacks (ADV_set3) transfer weakly.

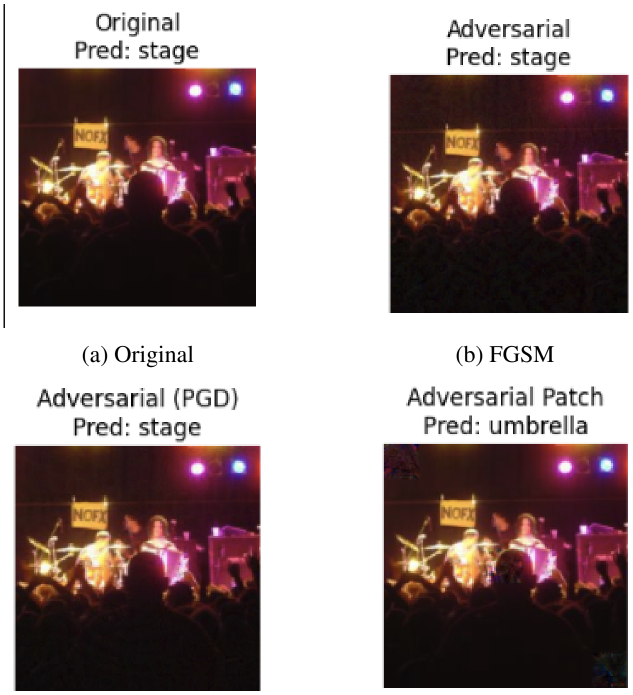
Perturbation Visuals and Fidelity

- **FGSM / PGD:** Introduced noise-like perturbations across the full image, visibly distorting high-frequency regions.
- **Patch Attacks:** Localized the modification to small fixed regions; images retained semantic meaning but fooled the model.

Visual Comparison of Adversarial Examples

Figure 4 presents a side-by-side comparison of the original image and its corresponding adversarial versions generated using FGSM, PGD, and Patch-based attacks. Each image shows the model's Top-1 predicted label under the respective attack.

- **Original:** The image is correctly classified by ResNet-34 with high confidence.
- **FGSM:** This single-step gradient-based attack introduces subtle, imperceptible perturbations across the entire image. Despite visual similarity to the original, it causes misclassification to a completely different label.



(a) Original

(b) FGSM

(c) PGD

(d) Patch

Figure 4: Top-1 predictions under different adversarial attack strategies. Patch attack induces a targeted misclassification, while FGSM and PGD introduce subtle noise across the entire image.

- **PGD:** The iterative variant of FGSM (PGD) applies constrained multi-step noise, resulting in a stronger misclassification. PGD further degrades confidence in correct labels, achieving 0% Top-1 accuracy in our tests.
- **Patch:** This attack modifies only a localized 32×32 region of the image. Despite affecting a small part of the input, it induces a targeted misclassification. This demonstrates that sparse perturbations can effectively fool classifiers without altering most of the visual content.

These examples highlight that deep models like ResNet-34 are highly sensitive to both dense and sparse adversarial perturbations. Notably, the adversarial examples remain perceptually indistinguishable to the human eye, yet cause drastic changes in the model's predictions.

Conclusion

Our experiments demonstrate that state-of-the-art convolutional models remain highly vulnerable to adversarial perturbations, even under modest attack budgets. Both FGSM and PGD attacks drastically reduced ResNet-34's Top-1 accuracy to below 2 %, with PGD achieving near-complete model failure. Patch-based L_0 attacks proved effective at inducing targeted misclassification while preserving visual realism, though they exhibited limited cross-model transferability.

Transfer experiments on DenseNet-121 highlight the broader security risk posed by transferable adversarial examples: simple white-box attacks (FGSM, PGD) on one model can significantly degrade the accuracy of another unseen architecture. This reveals the insufficiency of model-specific defenses and emphasizes the need for robust, architecture-agnostic mitigation strategies.

In summary, our evaluation reaffirms the fragility of vision classifiers under adversarial pressure and motivates the continued development of scalable defenses that generalize across attack types and model families.

Future Work. We plan to explore stronger and more adaptive adversarial strategies, including AutoAttack and CW-based optimizations. Extending the current evaluation to vision transformers (e.g., ViT, CLIP) and adversarially trained models will help assess generalization under architectural shifts. We also aim to test saliency-guided patch placement for more efficient L_0 attacks and investigate multi-model ensemble robustness.

Limitations and Challenges. Despite strong attack performance, iterative methods like PGD and patch attacks incur high computational cost, requiring long runtimes even on modern GPUs. Patch-based attacks are sensitive to patch location and may fail under random cropping or resizing. Transferability results, while insightful, depend heavily on architecture similarity, and broader generalization remains an open challenge.

Reproducibility. All attack implementations, evaluation scripts, and adversarial datasets are available in our public GitHub repository (linked on the title page). These results can be reproduced by any CUDA-enabled GPU (8 GB+ VRAM). We also provide pretrained adversarial samples and intermediate metrics for ease of verification.

References

[1] Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations (ICLR)*. `https://arxiv.org/abs/1412.6572`

[2] Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *International Conference on Learning Representations (ICLR)*. `https://arxiv.org/abs/1706.06083`

[3] Brown, T. B.; Mane, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial Patch. *arXiv preprint arXiv:1712.09665*. `https://arxiv.org/abs/1712.09665`

[4] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. `https://doi.org/10.1109/CVPR.2016.90`