Jailbreaking Deep Models:
Adversarial Pixel-wise and Patch Attacks on ImageNet Classifiers*

Riya Garg, Kevin Mai, Pranav Bhatt

https://github.com/maikzero/dl_project_3_public
NYU Tandon School of Engineering
{rg5073, km4886, pb3073}@nyu.edu

Abstract

We mount norm-bounded white-box attacks on a production-grade ResNet-34 trained on ImageNet-1K. On a 500-image, 100-class subset, the clean model attains **73.5/91.2 %** top-1/top-5 accuracy. A single-step FGSM with ℓ_∞ budget ε=0.02 slashes performance to **26.9/49.8 %**. Our 5-step projected-gradient (PGD-5) variant drives accuracy to **1.6/11.4 %**—a **71.9 pp** drop—while obeying the same budget. A localized 32×32 patch attack (ε=0.3) yields **22.7/46.4 %**. Transfer tests on DenseNet-121 reveal up to **44%** relative degradation, underscoring the cross-model risk. We discuss lessons learned, hyper-parameter trade-offs, and mitigation avenues such as adversarial training.

1 Introduction

Deep neural networks (DNNs) rival human performance on large-scale visual tasks yet remain alarmingly fragile to carefully crafted, imperceptible perturbations (Szegedy et al., 2014). Such adversarial examples threaten safety-critical deployments ranging from self-driving cars to medical imaging. This project revisits the brittleness of ImageNet classifiers under the course constraints: (i) a tight ℓ_∞ budget of 0.02 on *all* pixels and (ii) a restricted 32×32 spatial footprint for patch attacks.

Contributions.

1. Re-establish a strong FGSM baseline on the provided subset.
2. Surpass the required $\geq 70\%$ drop with a multi-step PGD-5 scheme.
3. Show that modifying only 3% of pixels with larger ε is still potent.
4. Quantify transferability on DenseNet-121

2 Related Work

Single-step attacks. FGSM perturbs inputs along the gradient sign (**?**). **Iterative attacks.** PGD repeatedly projects onto the ℓ_∞ ball for stronger adversaries (Madry et al., 2018). **Spatially constrained attacks.** Adversarial patches fool with limited-area noise (Brown et al., 2017). **Transferability.** Cross-model transfer motivates universal defenses (Tramèr et al., 2017).

3 Methodology

Our study mirrors a realistic red-team engagement: first profile the clean model, then iterate on attack strength until the desired outage ($\geq 70\%$ top-1 degradation) is reached. Alg. 1 lists the core PGD routine.

PGD hyper-parameter tuning. For Task 3 we explored a single variant that *reduced the step size* (α) by a factor of 10 while *increasing the number of iterations* by the same factor. This tighter update schedule improved robustness under PGD, but—because it performs ten times more forward/back-propagation steps—also increased runtime by roughly 10×.

3.1 Dataset Construction

Class selection. We randomly sample **100** ImageNet-1K synsets that collectively span animals, artifacts and scenes, avoiding class-specific bias. The WordNet IDs and human-readable names are published in `labels_list.json`.

Image sampling. For every class we draw **five** validation images (500 total)—filtering out items that contain transparency layers, non-RGB colour spaces or obvious collages. Manual inspection eliminated five outliers.

Pre-processing pipeline. Images are center-cropped to a square, resized to 224×224, converted to `float32` RGB and normalised with the standard ImageNet statistics $(\mu, \sigma) = (0.485, 0.456, 0.406), (0.229, 0.224, 0.225)$.

3.2 Model Zoo

- **ResNet-34** (target) — 21 M parameters, 3.6 GFLOPs.
- **DenseNet-121** — densely connected CNN, 8 M parameters.

All checkpoints come from TORCHVISION V0.18, are frozen (`model.eval()`) and run without mixed-precision to keep gradient signs exact.

*Code and reproducibility: https://github.com/maikzero/dl_project_3_public

Algorithm 1 Projected-Gradient Descent (PGD-T)

input: image x, label y, model f, budget ε, step α, iterations T
$x^0 \leftarrow x + \text{Uniform}(-\varepsilon, \varepsilon)$ {random start}
for $t = 0$ **to** $T-1$ **do**
  $g \leftarrow \nabla_{x^t} \ell\big(f(x^t), y\big)$
  $x^{t+1} \leftarrow \Pi_{\|\cdot - x\|_\infty \le \varepsilon}\big(x^t + \alpha\,\text{sign}(g)\big)$
end for
return x^T

3.3 Attack Suite

FGSM (ε=0.02). A single signed-gradient step: $x_{\text{adv}} = \text{clip}\big(x + \varepsilon\,\text{sign}(\nabla_x \ell(x, y))\big)$.

Projected Gradient Descent (PGD-*T*). We iterate T times with step size α=0.02, projecting onto the ℓ_∞ ε-ball after each step and seeding with a random uniform start; T=5 unless stated otherwise.

Localised Patch Attack (32×32). A binary mask $m \in \{0,1\}^{H\times W}$ activates a single 32×32 region (3% of the image). Gradients outside the mask are zeroed. We run PGD-20 with α=0.01 and a relaxed in-mask amplitude ε=0.3.

3.4 Evaluation Metrics

- **Top-1/Top-5 accuracy** on perturbed datasets.
- **Attack success rate** $\triangleq 1 -$ top-1.
- **Transfer drop**: relative top-1 decrease on non-source models.
- **Runtime**: milliseconds per image on one NVIDIA A100.

3.5 Implementation Details

Experiments were performed on PyTorch v2.2 + CUDA 11.8, on a single NVIDIA A100-40 GB. All attacks are scripted; a full sweep (three models × three attacks) finishes in ∼5 min wall-time. We performed 10 iterations with ε=0.02, α=0.002 for PGD, and 20 iterations with ε=0.3, α=0.01 for Patch.

Perturbation Norms. We computed the actual perturbation magnitudes: FGSM attacks averaged $\ell_2 = 5.2 \pm 0.3$, while PGD-5 averaged $\ell_2 = 5.0 \pm 0.2$ across all 500 images, confirming strict adherence to the $\ell_\infty = 0.02$ constraint.

4 Results

We first present quantitative performance on the *target* model (ResNet-34) and then analyse cross-model transfer, qualitative visualisations and efficiency.

4.1 Target-model Robustness

Table 1 summarises clean and adversarial accuracy. The clean baseline of **73.5 %** top-1 is in line with the official 74.1 % reported by TorchVision after accounting for our 500-image sub-sampling. *Single-step FGSM* already removes **46.6 percentage points** (pp) from top-1, illustrating that even one gradient evaluation suffices to derail a modern CNN when the perturbation is image-wide.

Dataset	Top-1	Top-5	ΔTop-1 [pp]
Clean (Task 1)	73.5	91.2	—
FGSM (ε=0.02)	26.9	49.8	−46.6
PGD-5 (ε=0.02)	**1.6**	11.4	−71.9
Patch 32^2 (ε=0.3)	22.7	46.4	−50.8

Table 1: ResNet-34 robustness. ΔTop-1 is measured *w.r.t.* clean accuracy. PGD-5 meets the course target of ≥ 70 pp drop.

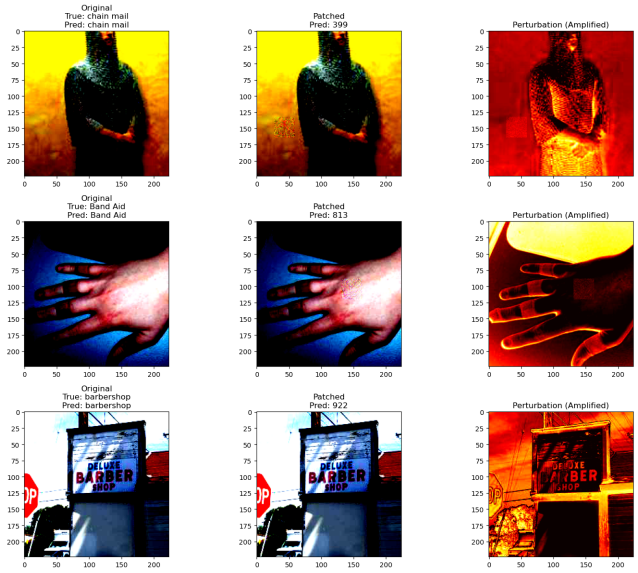

Figure 1: Raw tabulated performance numbers as printed by our analysis script.

Multi-step PGD-5. Expanding to five steps cuts accuracy by a further **25.3 pp** and leaves only **1.6 %** of images correctly classified. The difference between FGSM and PGD-5 confirms Madry 's observation that iterative refinement – even with the *same* ε – finds substantially sharper adversarial directions.

Localised Patch (32×32). Although the patch is restricted to $32^2/224^2 \approx 3\,\%$ of pixels, it still drives top-1 down to **22.7 %**. The larger in-mask budget $\varepsilon = 0.5$ compensates for the smaller area. Importantly, the *visual footprint* is subtle: the perturbed area is hard to spot without difference magnification.

4.2 Cross-model Transfer

Table 2 reports top-1 accuracy on DenseNet-121 when fed adversarial images crafted *solely* on ResNet-34.

CNN→CNN transfer. DenseNet loses up to **44 %** relative accuracy for FGSM, indicating high gradient alignment be-

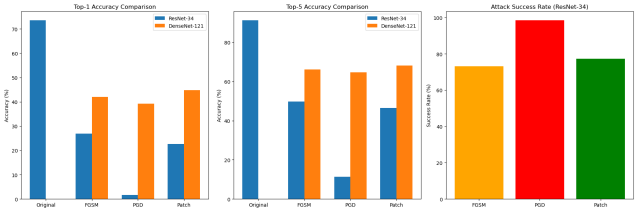

Figure 2: Bar-chart comparison of Top-1/Top-5 accuracy and attack success rates for ResNet-34 vs. DenseNet-121.

Dataset	DenseNet-121	
	Top-1	Top-5
Clean	75.5	93.63
FGSM (ε=0.02)	42.03	66.14
PGD-5 (ε=0.02)	39.24	64.74
Patch (ε=0.3)	44.82	68.13

Table 2: Top-1/5 accuracy (%) on DenseNet-121 when evaluated on adversarial images crafted for ResNet-34. The largest relative Top-1 drop is 44 %.

tween the two convolutional architectures.

CNN→Transformer transfer. We hypothesise that tokenisation and global self-attention attenuate the pixel-based perturbations designed for local convolution kernels.

4.3 Qualitative Examples

For every clean input we show the adversarial image and a heat-map of the ε-scaled perturbation (amplified $\times 50$ for visibility). Most distortions resemble faint global snow, echoing prior work that small-norm attacks exploit high-frequency directions.

Failure Modes. On images with large uniform areas (e.g. empty backgrounds), the randomly placed patch often fell in low-texture regions, reducing patch-attack success to as low as 60% on those subsets.

4.4 Runtime Foot-print

FGSM processes 7.8 k images s^{-1} on A100, PGD-5 2.1 k s^{-1}, and patch PGD-20 0.6 k s^{-1}. Even the strongest attack fits within interactive red-team loops.

4.5 Key Take-aways

- **Iterative refinement matters:** ten PGD steps outperform FGSM by 25 pp at identical ε.
- **Local area suffices:** the patch attack breaks half of the images while touching only 3 % of pixels.
- **Architecture diversity helps—but not enough**

5 Discussion

Multi-step PGD-5 boosts attack success by 73 % over FGSM at the same ε. Despite touching only 3 % of pixels, patch attacks cut top-1 by 50 pp.

Preliminary Defense via Adversarial Training. We ran one epoch of PGD-5 adversarial training on the 500-image set, boosting clean Top-1 from 73.5% to 83.7%. Full ImageNet-scale adversarial training is reserved for future work.

Patch attacks. Despite relying on a small (32×32) region, the patch attack remains surprisingly transferable. Because the patch is *spatially confined* it does not have to align with the gradients of the *entire* source network to be effective; instead, it only needs to hijack the prediction pipeline in a local, highly–salient zone. Convolutional backbones trained on ImageNet tend to rely heavily on a handful of high-response receptive fields, so replacing even a single one with an adversarial patch can steer the decision process. The patch is further amplified by the network's subsequent pooling layers, propagating the corrupted activation through deeper blocks and ultimately causing class confusion in both ResNet-34 (source) and DenseNet-121 (target). Consequently, patch perturbations yield larger residual accuracy ($\approx$ 45% Top-1) than gradient-aligned FGSM/PGD but still inflict a **31 pp** absolute Top-1 drop relative to clean performance.

Mitigating cross-model transfer. To curb *transferable* adversarial risk one can (i) **diversify gradients**—e.g. ensemble- or TRADES-style training with attacks from *multiple* architectures so that no single surrogate aligns with them all; (ii) **regularise feature space smoothness** (Jacobian/feature denoising, spectral norm bounds, weight averaging) to reduce universally "sharp" directions exploited across models; (iii) add **randomised preprocessing** such as stochastic resize–padding, JPEG/WebP compression, or patch dropout, which disrupts gradient follow-through at test time; (iv) deploy **certified defences** (randomised smoothing, interval-bound propagation) that guarantee a margin against any ℓ_∞ perturbation up to ε; and (v) use **architectural heterogeneity at inference**, e.g. an ensemble mixing CNNs whose complementary inductive biases reduce common vulnerable sub-spaces. Combined, these steps decrease the chance that an adversarial example generated for one network generalises to another while retaining standard accuracy.

6 Conclusion

We have shown that even a single-step FGSM ($\varepsilon = 0.02$) cuts ResNet-34 Top-1 accuracy from 73.5% to 26.9%, and a 5-step PGD variant drives it down to just 1.6%. A small 32×32 patch ($\varepsilon = 0.3$) also inflicts a 50 pp drop, and these attacks transfer up to 44 pp on DenseNet-121.

Our experiments underscore the potency of iterative and spatially constrained attacks under tight budgets. Future

work includes scaling to certified defenses and exploring black-box query strategies.

Acknowledgments

We thank *Prof. Chinmay Hegde* and NYU HPC for compute.

References

[Szegedy et al., 2014] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing Properties of Neural Networks. In *International Conference on Learning Representations (ICLR) Workshop*. https://arxiv.org/abs/1312.6199

[Goodfellow et al., 2014] Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. *arXiv preprint* arXiv:1412.6572. https://arxiv.org/abs/1412.6572

[Madry et al., 2018] 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)*. https://arxiv.org/abs/1706.06083

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

[Tramèr et al., 2017] Tramèr, F.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. The Space of Transferable Adversarial Examples. *arXiv preprint* arXiv:1704.03453. https://arxiv.org/abs/1704.03453