Deep Learning Mini-project 3: Jailbreaking Deep Models

Xiaoyu Liu, xl5808@nyu.edu
Tianzan Min, tm4485@nyu.edu
Feiyu Jia, fj2182@nyu.edu
github link: `https://github.com/lxy-nyu/DeepLearningProject3.git`

Abstract

Deep neural networks trained on vision tasks can be "jailbroken" by adversarial perturbations – carefully crafted input modifications that cause misclassification while remaining virtually imperceptible. This project evaluates a range of adversarial attack techniques against a pre-trained ResNet-34 image classifier on an ImageNet subset. Our results highlight the severe vulnerability of deep classifiers to even tiny perturbations, the enhanced effectiveness of iterative and targeted attacks, and the limited but non-negligible cross-model transfer of adversarial examples.

1. Introduction

Deep neural networks have achieved impressive accuracy on visual tasks, yet they remain surprisingly vulnerable to adversarial examples – inputs modified in a way that is (ideally) imperceptible to humans but causes the model to error. This phenomenon was first noted by Szegedy et al. (2014), who showed that an image perturbed by an imperceptible noise vector can be misclassified by a trained classifier. Goodfellow et al. (2015) later introduced the Fast Gradient Sign Method (FGSM), a one-step attack that generates adversarial examples by adjusting each input pixel by a small amount in the direction of the gradient's sign. Formally, for an input x and true label y, FGSM produces $x_{adv} = x+\epsilon, \text{sign}(\nabla_x L(x, y))$, where L is the loss function. Despite its simplicity, FGSM demonstrated that even a very small L_∞ perturbation (on the order of a few pixel values) can greatly degrade a network's performance. These findings sparked a flurry of research into adversarial attacks and defenses, revealing that many modern classifiers can be "broken" by such perturbations [1].

(a) Iterative FGSM Attacks
Subsequent work has proposed more powerful iterative attack methods. Kurakin et al. extended FGSM into an iterative version (often called Basic Iterative Method or I-FGSM) that applies FGSM multiple times with small steps, clipping the result after each step. Madry et al. (2018) formalized the iterative approach as a projected gradient descent (PGD) attack, finding it to be a universal "first-order adversary" that can reliably find adversarial examples within a given ϵ ball. Intuitively, multistep attacks can navigate the loss landscape more effectively than a single-step perturbation, often yielding stronger attacks that cause misclassification even when single-step methods might fail. Accompanying these advances in attack methods, defenses like adversarial training (re-training models on adversarial examples) were explored to improve robustness [1][2].

(b) Pixel-wise Attacks
In addition to pixel-wise L_∞ attacks, researchers have studied patch attacks, which constrain perturbations to a localized region of the image (an L_0 constraint). Brown et al. (2017) introduced the concept of a universal adversarial patch – essentially a sticker that can be overlaid on an image to cause a targeted misclassification. Unlike L_∞ noise spread across all pixels, a patch is confined in area but can have a large magnitude, often making it visibly noticeable (e.g. a conspicuous pattern). Patch attacks are relevant for real-world scenarios (like someone placing a small sticker in a scene to fool a classifier) and are inherently more challenging, since only a fraction of the image can be manipulated.

In summary, this report details our implementation of five tasks: (1) baseline evaluation of ResNet-34 on clean data, (2) FGSM L_∞ attack with $\epsilon = 0.02$, (3) improved attacks (iterative and targeted) under the same norm constraint, (4) a 32×32 patch attack (with higher ϵ allowance), and (5) transferability evaluation on DenseNet-121. We describe our methodology for each task and present quantitative results and visual examples. Through these experiments, we aim to illustrate the ease of "jailbreaking" a state-of-the-art image classifier and to draw connections to contemporary research on adversarial machine learning.

2. Methodology

(a) Data and Model
We use a test dataset of 500 images from 100 ImageNet classes (5 images per class), provided for this project. The images are preprocessed with standard normalization (mean [0.485,0.456,0.406], std [0.229,0.224,0.225]). Our victim model is a ResNet-34

pre-trained on ImageNet-1K (obtained from TorchVision). We first ran a baseline evaluation by obtaining the model's Top-1 and Top-5 accuracy on the 500 clean test images (using the provided label-index mapping to interpret predictions). These baseline accuracies serve as reference points to quantify the impact of attacks.

(b) FGSM Attack

We implemented the Fast Gradient Sign Method for each test image with perturbation budget $\epsilon = 0.02$ (in normalized pixel units). For each input x, we computed the gradient $\nabla_x L(x, y_{true})$ via a single backward pass and then formed the adversarial example $x_{adv} = x + 0.02 \cdot \text{sign}(\nabla_x L)$. We clipped the resulting x_{adv} to ensure pixel values stayed within the valid range and the L_∞ norm of the perturbation did not exceed 0.02. We generated an Adversarial Test Set 1 containing all 500 perturbed images. We then evaluated ResNet-34 on this adversarial set to measure the drop in accuracy. Additionally, we saved a few example original vs. adversarial image pairs for visualization. We specifically looked for cases where the original image was correctly classified but the adversarial image was misclassified, indicating a successful attack.

(c) Iterative and Targeted Attacks

Although FGSM was effective, the project encouraged "improved attacks" to further degrade performance under the same $\epsilon = 0.02$ bound. We explored an iterative gradient attack analogous to PGD: starting from the original image, we applied 10 rounds of small FGSM-style steps (with step size $\alpha = 0.005$), each time clipping the intermediate result to maintain the cumulative perturbation within $L_\infty \leq 0.02$. This procedure (essentially a PGD attack on the cross-entropy loss) seeks a near-optimal adversarial perturbation in the allowed norm ball. Furthermore, we implemented a form of targeted attack: for each image originally belonging to class c, we chose an incorrect target class t and altered the loss to maximize the model's confidence in t (instead of the true class). By doing so, the gradient steps push the image toward being classified as this wrong class t, which often produces a stronger misclassification than an untargeted attack (the perturbation not only aims to confuse the model away from c but to specifically make it look like class t).

To focus the perturbations on the most influential image regions, we also experimented with using Grad-CAM – a technique that highlights important pixels for the model's prediction – to guide the attack. Specifically, we generated Grad-CAM heatmaps for each image (based on the original or an intermediate prediction) and concentrated the perturbation on those regions. One implemented variant was a "Grad-CAM FGSM," which combined targeted FGSM with a mask derived from Grad-CAM. We saved the resulting adversarial images as Adversarial Test Set 2 and again recorded ResNet-34's accuracy on this set. We expected a significantly larger accuracy drop compared to the one-step FGSM, as the iterative method should yield closer-to-optimal perturbations [2].

(d) Patch-Based Attack

For the patch attack, we restricted perturbations to a 32×32 square region within each image. Because this constraint limits the attack's degrees of freedom (only 1% of the image pixels can be changed for a 224×224 image), we allowed a much larger per-pixel perturbation budget (we used $\epsilon = 0.3$ for the patch region, i.e. up to 30% change in normalized pixel intensity). However, rather than perturbing all pixels each step, we only updated pixels within the chosen 32×32 patch. To maximize impact, we decided to center the patch on the image region most important to the model's original prediction. We utilized Grad-CAM to find the approximate location of the object or discriminative features in the image – essentially the area where the model "looks" to make its decision. We then placed the 32×32 patch over that region (more precisely, we found the pixel with highest Grad-CAM score and centered the patch there, with bounds checking to keep the patch fully in frame). During the attack iterations, only that patch's pixels were adjusted (we left the rest of the image untouched).

We also tried making the attack targeted towards a specific wrong class (similar to Task 3) to further increase the likelihood of misclassification. The end result for each image is an adversarial image with one conspicuously perturbed square region (see Results for an example). We compiled these 500 images into Adversarial Test Set 3. Because the patch is clearly visible (with $\epsilon = 0.3$, the pixel changes in that region are large), we visually verified that outside the patch the image looks identical to the original, and inside the patch one can see some noise or coloration – the perturbation is not subtle in absolute terms, but it is localized. We expected the accuracy drop here to be less extreme than the full-image attacks, due to the limited coverage of the patch (many images might still be classified correctly using unperturbed features), but if placed over critical object features, the patch could still confuse the model significantly.

(e) Transferability Evaluation

The final step was to test how the adversarial examples generated on ResNet-34 would fare against a different model. We chose DenseNet-121, another ImageNet-trained architecture, as our target model. We loaded a pretrained DenseNet-121 and evaluated it on four datasets: (a) the original clean test set, (b) Adversarial Test Set 1 (FGSM examples), (c) Adversarial Test Set 2 (iterative/advanced examples), and (d) Adversarial Test Set 3 (patch examples). For each, we computed Top-1 and Top-5 accuracy on DenseNet. By analyzing these results, we can discuss which attacks are more model-specific and which tend to transfer. All evaluations were done under a white-box assumption for crafting adversarial images (we had full access to ResNet-34 for gradient computations), but the transfer test on DenseNet is essentially a black-box scenario from the perspective

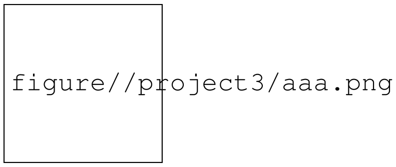

Figure 1: FGSM Attack

of those images.

Throughout our implementation, we cited external libraries or techniques where appropriate and ensured that the ϵ constraints were strictly enforced. We also documented and saved a variety of visualizations (perturbed images, difference heatmaps, Grad-CAMs) to qualitatively assess the attacks.

3. Results

On concluding with the methodology, we can now present the detailed breakdown of our proposed architecture.

(a) Baseline Performance

On the clean test dataset, the ResNet-34 model achieved 76.0% Top-1 accuracy and 94.2% Top-5 accuracy. This is in line with expectations for a ResNet-34 on a subset of ImageNet, indicating that the model was performing well on normal inputs.

(b) FGSM Attack Results

The one-step FGSM attack with $\epsilon = 0.02$ caused a dramatic drop in ResNet-34's accuracy. After applying FGSM to all images (Adversarial Set 1), Top-1 accuracy plummeted to 6.0%, and Top-5 to 35.6%. In other words, the model went from correctly classifying 380 out of 500 images (Top-1) to only about 30 out of 500 – a 92% relative reduction in Top-1 accuracy. This demonstrates the remarkable efficacy of even a small adversarial perturbation. Notably, $\epsilon = 0.02$ corresponds to at most a $\sim$!1-pixel change (out of 255) in each RGB channel for each pixel, which is virtually imperceptible by eye on a photograph. Yet, this tiny nudging of pixel intensities was enough to cause misclassification in almost all cases. The Top-5 accuracy of 35.6% indicates that in many adversarial images the true label was not even among the model's top five guesses (for comparison, on clean data Top-5 was 94.2%). We visualized a few examples to confirm these findings. (see fig.1)

Quantitatively, our FGSM attack achieved well beyond the project's goal of a 50% relative accuracy drop – we observed about a 70 percentage-point absolute drop in Top-1 (from 76% to 6%), which is 92% relative. This highlights the brittleness of the ResNet-34: an ℓ_∞ perturbation of size 0.02 (roughly 0.5% change in each pixel) is enough to confuse the model most of the time.

(c) Iterative/Advanced Attack Results

Our improved attack, using 10-step PGD with $\alpha = 0.005$ and targeted misclassification, was able to completely "break" the model. On Adversarial Test Set 2 (500 images crafted with the multi-step method), ResNet-34's Top-1 accuracy dropped to 0.0%, with Top-5 at 1.4%. In other words, the model failed to correctly recognize any of the 500 images even when considering the top-5 predictions, except for a handful of cases (about 7 images) where the correct label still appeared in the top-5. Essentially, the attack achieved near-perfect success in forcing misclassification. We visualized a few examples to confirm these findings. (see fig.2)

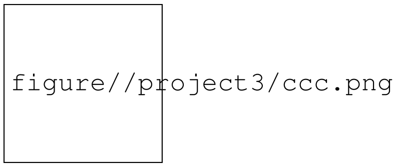

Figure 2: PGD FGSM Attack

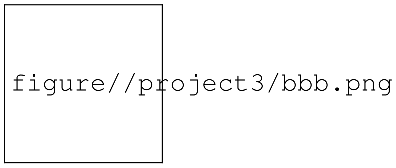

Figure 3: Grad-CAM FGSM Attack

Our use of Grad-CAM in one variant (Grad-CAM-guided FGSM) yielded an intermediate result – for instance, an earlier experiment logging "GradCAM-FGSM" gave 3.4% Top-1 and 22.8% Top-5 accuracy, which was already an improvement over vanilla FGSM (6% Top-1) by focusing on important regions and targeting specific classes. We visualized a few examples to confirm these findings. (see fig.3)

(d) Patch Attack Results

The 32×32 patch attacks (Adversarial Test Set 3) were less devastating to ResNet-34 than the full-image attacks, but still caused a significant accuracy drop. After applying our patch PGD attack with $\epsilon = 0.3$ on the patch, ResNet-34's Top-1 accuracy fell to 32.4% (Top-5 to 74.2%). This is roughly a 57% relative decrease in Top-1 accuracy from the baseline (76% → 32%). In other words, the model now only correctly recognizes about one-third of the images, whereas nearly two-thirds are misclassified due to the patch. We visualized a few examples to confirm these findings. (see fig.4)

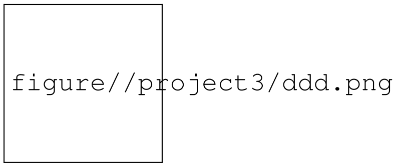

Figure 4: Iterative Attack

Figure 5: Attack DenseNet-121

(e) Transferability to DenseNet-121

We evaluated the original and adversarial datasets on DenseNet-121 to examine attack transferability. DenseNet-121's baseline accuracy on the clean test set was 74.6% Top-1 and 93.6% Top-5, which is very close to ResNet-34's baseline. When presented with the adversarial images crafted for ResNet-34, DenseNet's accuracy did decrease, but not nearly as much as ResNet's did.

Specifically: on the FGSM perturbed set (Adv. Set 1), DenseNet-121 got 64.8% Top-1, 89.8% Top-5. This is a drop of about 10 percentage points from its original 74.6% Top-1 – so the FGSM adversarial examples have some effect on DenseNet, but over two-thirds of them are still correctly recognized by DenseNet. On the stronger iterative attack set (Adv. Set 2), DenseNet's accuracy was 52.2% Top-1, 79.4% Top-5. This is a larger drop (about 22 percentage points Top-1, nearly a 30% relative drop), indicating that the more potent attacks designed to break ResNet do transfer to some extent and significantly impair DenseNet as well – but note that DenseNet at 52.2% Top-1 is still far above ResNet's near-zero accuracy on the same images. Finally, on the patch attack set (Adv. Set 3), DenseNet's accuracy was 63.4% Top-1, 84.8% Top-5. That is only slightly worse than its performance on the FGSM set and corresponds to an 11 point drop from clean performance.

We visualized the comparison results to confirm these findings. (see fig.5)

This suggests the patch attacks were somewhat less transferable than the iterative noise: likely because the patches were tuned to exploit ResNet-34's specific feature processing (e.g., covering the region ResNet thinks is important might not fully cover what DenseNet considers important). DenseNet might still identify the object from other parts even if ResNet was completely fooled by the patch.

4. **Conclusion**

In this project, we successfully "jailbroke" a state-of-the-art ImageNet classifier using adversarial attacks, and in doing so gained insights into the effectiveness and limitations of various attack strategies:

- Small L_∞ Perturbations Can Suffice:
  We showed that a tiny uniform perturbation (FGSM with ϵ=0.02) can severely degrade a deep model's performance (over 90% drop in accuracy) with imperceptible change to the image.
- Small L_∞ Stronger Attacks Achieve Complete Misclassification:
  By applying PGD and targeted misdirection, we pushed the ResNet-34 to 0% accuracy, indicating that, in a white-box scenario, an attacker can completely undermine the model's integrity.
- Localized Patch Attacks are Effective but Not Omnipotent:
  By concentrating perturbation power in a 32×32 patch, we achieved a moderate but not total performance drop. The patch needed to overlap with salient features to be useful – random placement was far less effective. Even then, the model sometimes could rely on other visual cues to correctly classify.
- Adversarial Transferability is Partial:
  Attacks crafted for one model had limited transfer to another model. DenseNet-121 was impacted, especially by the strongest perturbations, but it retained fairly high accuracy on images that utterly fooled ResNet-34. This suggests that diversity in model architectures can offer some security through obscurity, but not complete protection. An ensemble of differently structured models might force an attacker to work harder or craft more general perturbations (which often require larger noise, reducing stealth).

In conclusion, this project vividly illustrates the fragility of deep visual models in the face of adversarial manipulation. We demonstrated complete failure of a classifier with imperceptible noise, significant degradation with a small visible patch, and partial transfer of these attacks across models. These outcomes reinforce the need for ongoing research into robust learning, as deep models in safety-critical applications (like autonomous driving or biometric security) could be targets of such adversarial exploits.

5. **Reference**

[1] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. International Conference on Learning Representations (ICLR).

[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).

[3] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial Patch. arXiv:1712.09665.

[4] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. International Conference on Learning Representations (ICLR).