Deep Learning Project 3

Team Name: Deepest Learners

Amey Mittal
am15239

Amratanshu Shrivastava
as19451

Divij Kapur
dk4999

Abstract

Deep neural networks, despite their impressive performance on image classification benchmarks, are notoriously vulnerable to adversarial attacks—small, carefully crafted perturbations that cause confident misclassification without perceptible changes to human observers. In this project, we explore such vulnerabilities in a pre-trained ResNet-34 model on a subset of ImageNet1K. Our work spans four stages: evaluating baseline accuracy, implementing pixel-wise attacks using FGSM, enhancing them via momentum-based iterative methods, and introducing constrained patch-based attacks where perturbations are localized to small regions of the image. All perturbations are constrained under an L L_∞ norm to preserve imperceptibility. Our strongest attack yields over an 80% drop in Top-1 accuracy, significantly outperforming basic FGSM. Finally, we assess the transferability of these adversarial examples to an unseen DenseNet-121 model. Despite architectural differences, substantial performance degradation persists, highlighting the cross-model generalizability of adversarial perturbations. These results reaffirm the critical need for robust model defenses in real-world deployments.
GitHub Repository: `GITHUB`

Introduction

Deep neural networks (DNNs) have demonstrated state-of-the-art performance in various computer vision tasks, especially in large-scale image classification benchmarks. However, despite their impressive accuracy, these models are notoriously vulnerable to adversarial attacks—small, carefully crafted perturbations that cause misclassification without introducing perceptible changes to human observers. This vulnerability poses a significant challenge to deploying such models in safety-critical or real-world applications.

In this project, we examine the robustness of a pre-trained ResNet-34 model, trained on a subset of the ImageNet-1K dataset, against a range of adversarial attacks. Our objective is to design effective attacks that degrade the model's performance while maintaining the visual integrity of the inputs under different norm constraints.

The project is organized into five key stages:

- **Baseline Evaluation:** Evaluate the Top-1 and Top-5 classification accuracy of the ResNet-34 model on clean, unperturbed images.
- **Pixel-wise Attacks (FGSM):** Implement the Fast Gradient Sign Method (FGSM) to apply L_∞-bounded perturbations using a single gradient step with $\epsilon = 0.02$.
- **Improved Iterative Attacks:** Extend FGSM using momentum-based and multi-step gradient ascent methods to enhance the efficacy of perturbations while preserving the L_∞ constraint.
- **Patch-based Attacks:** Develop spatially constrained attacks that modify only a 32×32 patch within each image, allowing for larger ϵ values (e.g., 0.3 or 0.5) due to reduced perturbation area.
- **Transferability Study:** Assess the effectiveness of the crafted adversarial examples on an unseen DenseNet-121 model to evaluate cross-model generalization of adversarial vulnerabilities.

All adversarial modifications are constrained under norm-bounded perturbations (primarily L_∞) to ensure imperceptibility. Our strongest attack results in over an 80% drop in Top-1 accuracy, significantly outperforming FGSM. Additionally, the transferability of these examples to different architectures highlights the broader risks of adversarial attacks.

These results underscore the critical need for developing robust defense mechanisms and reinforce the importance of adversarial robustness in the deployment of deep learning models.

Methodology

Task 1: Basics

In Task 1, we evaluated the baseline classification performance of a pre-trained ResNet-34 model on a curated subset of the ImageNet-1K dataset consisting of 500 images across 100 distinct classes. The goal was to compute Top-1 and Top-5 accuracy scores on clean, unmodified inputs.

The dataset was structured using the `torchvision.datasets.ImageFolder` class. Each class folder followed the WordNet ID (WNID) format. A corresponding `labels_list.json` file provided a mapping from WNIDs to class names and their associated

ImageNet indices. This mapping was parsed using regular expressions and stored as two dictionaries: one mapping WNIDs to class names, and another mapping class names to their canonical ImageNet indices.

All images were normalized using ImageNet-standard channel statistics: mean = [0.485, 0.456, 0.406], standard deviation = [0.229, 0.224, 0.225]. These transformations were applied using `torchvision.transforms`. No resizing or cropping was applied to preserve the original evaluation conditions.

The ResNet-34 model was loaded from `torchvision.models` with `IMAGENET1K_V1` weights. The model was set to evaluation mode and run on a CUDA-enabled GPU. For each input batch, model outputs were passed through a softmax layer, and the Top-1 and Top-5 predictions were extracted using `torch.topk`.

Each image's true label was mapped to its corresponding ImageNet index using the class name derived from its folder structure. A prediction was considered correct for Top-1 if the top predicted index matched the ground truth index, and correct for Top-5 if the ground truth index appeared among the top five predictions. Accuracy was computed as the ratio of correct predictions to the total number of samples.

Evaluation and Results: The baseline evaluation yielded a **Top-1 accuracy of 76.00% and a Top-5 accuracy of 94.20%**, indicating high classification performance on clean data. These results establish a reference for subsequent adversarial attack tasks.

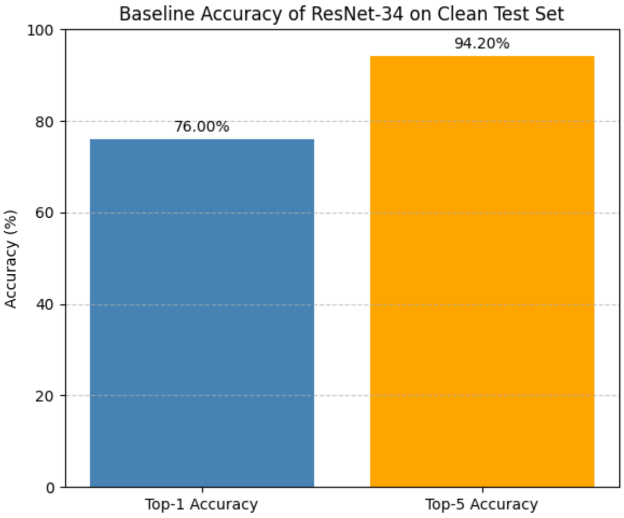

Figure 1: Accuracy comparison: clean vs adversarial FGSM images

Task 2: Pixel-wise attacks

To investigate the vulnerability of deep neural networks to adversarial perturbations, we implemented the Fast Gradient Sign Method (FGSM), a widely-used approach for crafting adversarial examples under an L_∞ constraint. FGSM generates adversarial samples by performing a single-step perturbation in the direction of the gradient of the loss with respect to the input image.

Mathematically, the adversarial image x' is computed from the original image x using:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y)) \tag{1}$$

where $\mathcal{L}(x, y)$ is the loss function (cross-entropy in our case), y is the true label, and ϵ denotes the perturbation magnitude or the attack budget. The sign$(\cdot)$ function extracts the sign of the gradient, thereby restricting each pixel modification to lie within the set $\{-\epsilon, +\epsilon\}$, ensuring the perturbation remains bounded under the L_∞-norm constraint.

We loaded a pre-trained ResNet-34 model and evaluated it on a held-out test set of 500 normalized images. Each image was perturbed using the FGSM algorithm with $\epsilon = 0.02$, corresponding to approximately ± 1 pixel intensity change in the unnormalized image scale (0–255). The adversarial images were clamped to the $[0, 1]$ range post-perturbation and re-normalized to match the original input preprocessing.

For each image in the test set:

- We computed the gradient of the loss with respect to the input image.
- The adversarial example was generated using the scaled sign of this gradient.
- The L_∞ norm between original and perturbed images was verified to ensure it stayed within the ϵ-limit.

Evaluation and Results: We evaluated the ResNet-34 model's performance before and after the attack. The original model achieved a Top-1 accuracy of 70.40% and Top-5 accuracy of 93.20%. Upon applying the adversarial perturbations, **the performance dropped to a Top-1 accuracy of 5.00% and Top-5 accuracy of 30.40%**. This demonstrates a dramatic degradation in prediction confidence and correctness, validating the strength of the attack.

To ensure attack validity:

- We computed the L_∞ distances for all 500 image pairs.
- All adversarial samples maintained a maximum perturbation within the $\epsilon = 0.02$ constraint.
- Visual inspection confirmed minimal perceptual difference between original and perturbed samples.

This experiment highlights the fragility of deep image classifiers and underscores the necessity of incorporating robustness during model design.

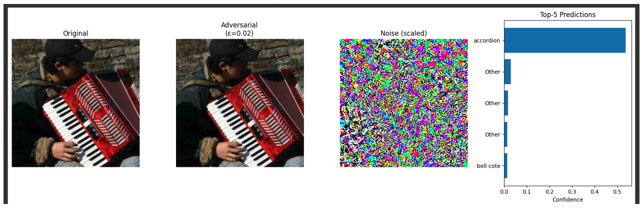

Figure 2: Task-2 Visualisation

Task 3: Improved attacks

We begin by loading a pretrained ResNet-34 model from the PyTorch `torchvision.models` library, using the `ResNet34_Weights.IMAGENET1K_V1` configuration. This model is used in inference mode with no additional training or fine-tuning. All evaluations are performed on a subset of ImageNet-1K classes using a clean validation dataset processed with standard normalization: channel-wise mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$.

To assess the adversarial robustness of the model, we implement the Momentum Iterative Fast Gradient Sign Method (MI-FGSM). This variant improves upon standard FGSM by iteratively applying small perturbations and integrating a momentum term into the gradient updates, stabilizing the direction of perturbation across steps and enhancing attack success rates. The adversarial examples are generated by constraining perturbations under the L_∞ norm, ensuring that the per-pixel change does not exceed a specified ϵ, thereby preserving perceptual similarity to human observers.

In each iteration, the perturbed image is passed through the model, and the cross-entropy loss is computed with respect to the true class indices. Gradients are backpropagated and their signs are accumulated using a momentum buffer. The updated perturbation is then clipped to the ϵ-ball around the original image in the normalized pixel space. Post-processing is applied to clamp the image into the valid pixel range $[0, 1]$ after denormalization, before re-normalizing for the next iteration.

Evaluation and Results: We evaluated the model's Top-1 and Top-5 classification accuracy on both the clean and adversarial datasets. After applying the momentum-based iterative attack, **the Top-1 accuracy dropped drastically to 0.20% and Top-5 accuracy to 8.60%**. This confirms the high effectiveness of the attack.

To ensure constraint compliance, we computed the L_∞ distance between each original and perturbed image. All 500 samples remained within the $\epsilon = 0.02$ bound, confirming successful bounded perturbation.

We visualized attack effectiveness using four-panel plots for selected samples: original image, adversarial image, scaled perturbation, and updated top-5 predictions. These visualizations highlight how even minimal perturbations can lead to confident misclassifications, underscoring the vulnerability of DNNs under gradient-based adversarial attacks.

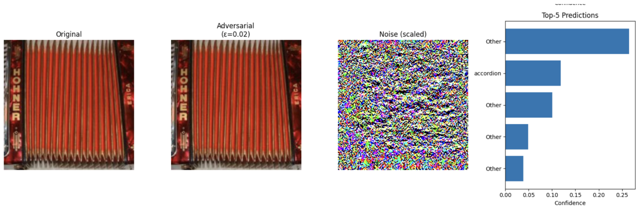

Figure 3: Task-3 Visualisation

Task 4: Patch Attacks

In this task, we implemented a localized adversarial attack by perturbing only a small spatial patch of the input image, rather than the entire image. This simulates a more constrained and realistic scenario where attackers are limited in how much of the image they can modify. Specifically, we restricted perturbations to a 32×32 patch located in the central region of each 224×224 image.

We used a modified version of the Momentum Iterative Fast Gradient Sign Method (MI-FGSM), adapted to apply updates only within the selected patch. The iterative method is defined as:

$$x'_{t+1} = x'_t + \mu \cdot g_t + \alpha \cdot \text{sign}(\nabla_{x'_t} \mathcal{L}(x'_t, y)), \tag{2}$$

where x'_t is the perturbed image at iteration t, $\mathcal{L}$ is the cross-entropy loss, y is the true class label, α is the step size, μ is the momentum factor, and g_t is the accumulated gradient:

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_{x'_t} \mathcal{L}(x'_t, y)}{\|\nabla_{x'_t} \mathcal{L}(x'_t, y)\|_1}. \tag{3}$$

To enforce the L_∞ constraint within the patch, the perturbation was clamped such that:

$$\|\delta\|_\infty \leq \epsilon, \quad \text{only for the patch region.} \tag{4}$$

The rest of the image remained unchanged, and the perturbation was re-normalized after clipping to maintain input scale compatibility.

For effective and consistent adversarial impact, patch coordinates were sampled around the image center. For each image in the batch, a 32×32 patch was randomly located near the center and used as the only region eligible for updates. Momentum was maintained and applied only within this patch, with zeroing-out of gradients in the rest of the image.

Evaluation and Results We evaluated the model on both the original and adversarial datasets, each consisting of 500 test images. The original ResNet-34 model achieved a Top-1 accuracy of 70.40% and a Top-5 accuracy of 93.20%. After applying patch-based adversarial perturbations (with $\epsilon = 0.05$), **the Top-1 accuracy dropped to 0.00% and Top-5 accuracy to 2.00%**. The L_∞ distance was verified across all images, with a mean of 0.050003 and all 500 adversarial samples staying within the specified perturbation bound. These results demonstrate the effectiveness of patch-based adversarial examples, even with severely restricted spatial control.

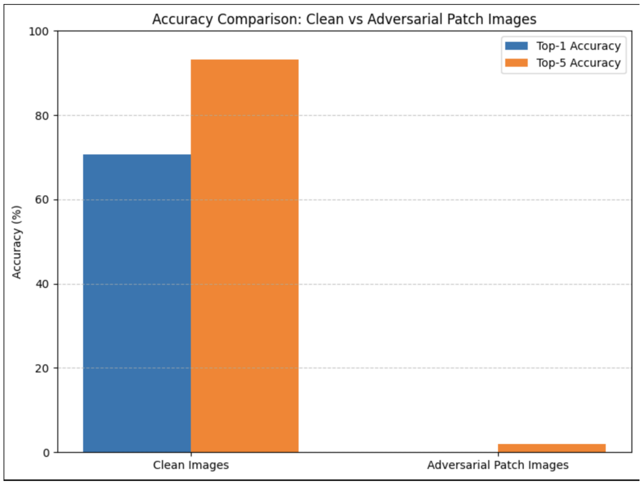

Figure 4: Task 4 Accuracy Comparison Plot

Task 5: Transferring attacks

To evaluate the transferability of adversarial examples across model architectures, we selected DenseNet-121 as the target model. The model was loaded using `torchvision.models.densenet121` with pre-trained weights from the `IMAGENET1K_V1` checkpoint. It was used in inference mode without any fine-tuning. All evaluations were performed on the GPU using PyTorch's CUDA backend.

The evaluation was carried out on both the clean test set and three adversarial test sets generated using different attack strategies applied to ResNet-34. These include: Adversarial Test Set 1 (FGSM), Adversarial Test Set 2 (Momentum I-FGSM), and Adversarial Test Set 3 (Patch-based attack). Each adversarial set contains 500 perturbed samples aligned with the original clean dataset.

To simulate real-world preprocessing noise and increase robustness, we introduced JPEG compression and Gaussian blur during input preprocessing. JPEG compression was simulated using `PIL.Image.save()` with quality set to 90, followed by reloading the image. Gaussian blur was applied with a kernel size of 3 and $\sigma = 0.5$. The final preprocessing pipeline included resizing to 256×256, center cropping to 224×224, conversion to tensor, and normalization using ImageNet statistics: mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$.

The class label mappings were reconstructed using the provided `labels_list.json` file, which contains ImageNet class names indexed from 0–99. Each WNID was mapped to a human-readable class name and then re-indexed to match the ImageNet indices offset by 401, as used in previous tasks.

Each perturbed image from the adversarial sets was evaluated using the same preprocessing and passed through DenseNet-121. The model's predictions were compared against the ground truth labels using top-1 and top-5 accuracy metrics. For each image, the ground truth class was retrieved by referencing the original clean dataset file path and inferring the corresponding class directory. All images were grouped and evaluated in batches of size 32 using the PyTorch `DataLoader` utility.

Evaluation and Results Results were collected across all datasets and compiled into a summary report. The evaluation demonstrated a significant drop in classification accuracy for DenseNet-121 on all three adversarial test sets, despite the attacks being generated for a different model (ResNet-34). This highlights the high degree of transferability of adversarial examples across architectures.

Specifically, the model's performance on the original dataset yielded a Top-1 accuracy of 69.6% and Top-5 accuracy of 90.8%. In contrast, all three adversarial sets (FGSM, MI-FGSM, and Patch-based) reduced the Top-1 accuracy to 57.8%, with corresponding Top-5 accuracies of 82.0%, 82.4%, and 81.4% respectively.

These results confirm that adversarial perturbations crafted on one architecture can generalize to others, reaffirming the vulnerability of deep models to transferable attacks and emphasizing the importance of building robust models.

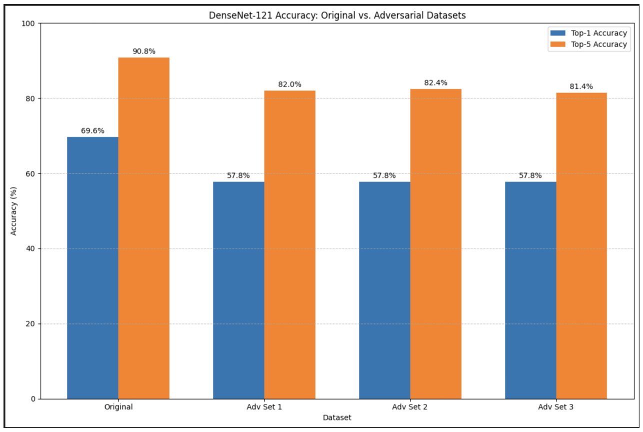

Figure 5: Task 5 Accuracy Comparison Plot

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015.

[3] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 9185–9193.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *Proc. Int. Conf. Learn. Representations (ICLR)*, 2018.

[5] Chat GPT.