Analysis for Adversarial Attack on Image on CNN – Check Point

Lupin Cai
CS Department
Emory University
Atlanta, Georgia
lupin.cai@emory.edu

Helen Jin
CS Department
Emory University
Atlanta, Georgia
helen.jin@emory.edu

Jinghan Sun
CS Department
Emory University
Atlanta, Georgia
jinghan.sun@emory.edu

***Abstract*—This study investigates the effectiveness of different physical adversarial attack strategies on Convolutional Neural Networks (CNNs), with a special focus on landscape recognition. We will develop a CNN for classifying images from the Intel Image Classification dataset, which features diverse natural scenes. Once the model is trained, we will introduce adversarial patterns to the images and analyze their impact on classification accuracy. Our goal is to identify which adversarial perturbations are most effective in reducing model performance, contributing insights into the vulnerabilities of CNNs in natural scene classification, and informing strategies for enhancing model robustness.**

I. Introduction

Convolutional Neural Networks (CNNs) have significantly advanced the field of computer vision, particularly in image classification tasks. However, their vulnerability to adversarial attacks remains a critical area of research. This study aims to explore the robustness of CNNs in natural scene classification using the Intel Image Classification dataset, which contains diverse images of natural scenes from around the world. Our research has two primary objectives: to develop and train a CNN model for effectively classifying images from this dataset and to systematically introduce adversarial patterns into the images to analyze their impact on classification accuracy. By examining the relationship between adversarial perturbations and CNN performance, we seek to identify the types of patterns most effective in deceiving the network and contribute to the ongoing effort to develop more robust image classification systems. This research advances our understanding of adversarial attacks in the context of natural scene classification and informs strategies for improving the resilience of CNNs in real-world applications, such as autonomous navigation and environmental monitoring.

II. Related Work

In their project about generating robust physical adversarial perturbations, researchers proposed an algorithm called RP2 to create visual adversarial examples effective under various physical conditions[1]. This work underscores the importance of accounting for real-world variables in adversarial attacks, inspiring our exploration of robust generalization in landscape recognition. Unlike their focus on stop signs, our work extends these principles to the broader and more complex domain of landscape recognition.

In the another project about crafting physical adversarial perturbations to fool object detectors, researchers proposed ShapeShifter, an attack targeting image-based object detectors like Faster R-CNN[3]. They addressed the added complexity of misleading multiple bounding boxes at different scales and adapted the Expectation over Transformation technique to enhance robustness against real-world distortions. Their success in generating adversarial stop signs mis-detected by object detectors informs our understanding of physical adversarial attacks in more complex recognition tasks. While their work concentrates on object detection of specific targets like stop signs, our research focuses on landscape recognition, which involves broader scene understanding and unique challenges.

III. Intended proposed approaches and system design

We plan to analyze the performance of models under adversarial attacks by benchmarking how accurately the model can classify the labels given in the picture after Robust Physical Perturbation[1]

The data set we planned to use featured a number of images of various objects. However, we plan to implement a different perturbation technique because the Robust Physical Perturbation introduced earlier is for image of traffic signs only. In order to increase success rates for adversarial patterns for landscape images, the original methods will need to be modified.

We plan to train a CNN model trained with unperturbed data first. Then we will mask the images for adversarial attack for this specific model. We will use robust physical

perturbation and our own. For each of the techniques, we will compare the hyper-parameter tuning for each of them including patch epochs, clipping threshold, evaluation criteria, noise-level added, and similarity between the attack picture.

Finally, we will test our adversarial attack against one of the existing defense mechanism[4] to evaluate the success rate of the adversarial attack under defense to improve our implementation.

IV. Software and Dataset

We plan to use Python 3.11.5 as our programming language. It is widely used in machine learning and provides excellent libraries for image processing and deep learning, such as Tensorflow and PyTorch. These are Python libraries that provide comprehensive tools for building and training neural networks.

The CIFAR-10 dataset is a widely-used computer vision benchmark comprising 60,000 color images, each sized at 32x32 pixels. The dataset is organized into 10 mutually exclusive classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), with 6,000 images per class. It is divided into a training set of 50,000 images and a test set of 10,000 images, structured as five training batches and one test batch of 10,000 images each. While the test batch contains exactly 1,000 images from each class, the training batches may have varying class distributions but collectively contain 5,000 images per class. An important distinction in the dataset is the clear separation between the automobile class (which includes sedans and SUVs) and the truck class (which only includes large trucks), with neither category including pickup trucks.

V. CNN Model Set Up

We built a CNN model for the CIFAR-10 dataset, using three convolutional layers with batch normalization and max-pooling after each layer to progressively reduce spatial dimensions while capturing features. Starting with an input of three channels, the first convolutional layer outputs 32 channels, followed by layers that increase output channels to 64 and then 128. After passing through these layers, the feature maps are processed with adaptive average pooling to reduce dimensionality, flattened, and passed through a dropout layer to mitigate overfitting before reaching the fully connected layer, which outputs 10 class predictions. The training process involves loading the data into batches of 64 and optimizing the model using Stochastic Gradient Descent (SGD) with momentum. A learning rate scheduler reduces the learning rate every 30 epochs to fine-tune the model's performance. At each epoch, cross-entropy loss is used to evaluate the model's performance on both training and test sets, with accuracy tracked to monitor progress. Additionally, an early stopping condition stops training if the test accuracy exceeds 70% and training loss falls below 0.18, ensuring efficient use of resources. Results are logged, metrics are visualized in a plot, and the trained model weights are saved for future use[5].

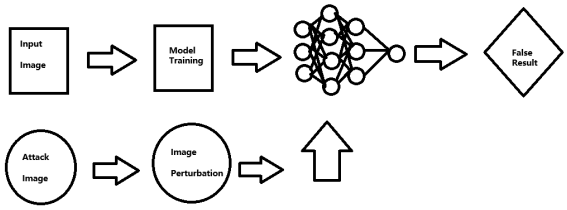

Fig. 1. Basic Flow Visualization

VI. Result Analysis

The training and evaluation of our CNN model on the CIFAR-10 dataset yielded promising results, as visualized in the graph:

Training Loss: The training loss (blue curve) decreases significantly within the initial epochs, showing that the model quickly learns the essential features from the dataset. The curve reaches a steady state around 0.2 after approximately 50 epochs, indicating stable convergence without signs of over fitting, as it remains low across the training duration.

Test Loss: The test loss (orange curve) follows a similar downward trend initially but stabilizes at a higher level compared to the training loss. This difference suggests that while the model generalizes reasonably well, there is some gap between training and testing performance. The test loss stabilizes around 0.6, which is relatively higher than the training loss, indicating potential areas for further tuning or regularization to improve generalization.

Test Accuracy: The test accuracy, represented by the green curve, shows a strong upward trend in the initial stages of training, quickly surpassing the 70% threshold and continuing to climb. This early rise in accuracy suggests that the model rapidly learns the fundamental features of the CIFAR-10 dataset, effectively distinguishing between the classes from the outset.

After around 20 epochs, the test accuracy stabilizes above 80%, indicating that the model achieves a high level of classification accuracy on unseen data. The plateau above 80% shows that the model has successfully generalized, capturing the core patterns and features across different classes without overfitting. This is especially notable for a challenging dataset like CIFAR-10, which contains

Fig. 2. Result Analysis

diverse images across 10 distinct categories, each with substantial variability in color, texture, and composition.

The steady test accuracy throughout the latter epochs also reflects the model's robustness in handling new data. Unlike models that exhibit sharp fluctuations or declines in test accuracy due to overfitting, our model maintains consistent performance, suggesting that it generalizes well across the dataset. This stability may be attributed to the regularization effects of dropout layers and batch normalization, which help prevent the model from relying too heavily on specific features of the training data.

Furthermore, reaching an accuracy over 80% positions this model as a competitive baseline for CIFAR-10 classification tasks, often comparable to the performance of more complex architectures. However, the plateau around this level also suggests that the model may be reaching its capacity within this architecture's design constraints. To further boost performance, we could consider techniques like data augmentation (to introduce more variability in the training data), additional fine-tuning, or exploring deeper architectures.

In summary, the test accuracy consistently surpassing 80% indicates that the model has effectively learned to differentiate between the CIFAR-10 classes, achieving reliable performance across epochs. This high level of accuracy demonstrates the model's readiness for potential real-world applications, where consistent and accurate classification is essential.

VII. Blank Paper and Black Dots

We write a program that generates a series of 32x32 white images with progressively added random black dots and classifies them using a CNN model to observe how the model responds to these abstract inputs. Initially, a plain white image is created as a baseline, with subsequent images having black dots added at random positions, simulating random noise. For each new image generated, the program records which class the CNN assigns to it. When two different images are classified into the same category, the program stops and we will analyze the pattern of dots on those images. This approach allows for an exploration of CNN's pattern recognition and classification sensitivity, aiming to uncover whether specific patterns of random dots trigger particular class predictions.

VIII. Future Steps

Our next steps focus on a thorough investigation of CNN classification behavior when presented with abstract patterns in the form of black dots on blank white images. Initially, we will create a diverse set of 32x32 white images with incrementally increasing numbers of black dots, carefully adding them at random or controlled positions to explore how the CNN responds to these variations. By systematically increasing the dot count, we aim to observe at what point certain patterns begin to produce consistent class predictions. For each generated image, we will record the CNN's classification output, paying close attention to any cases where different dot patterns are reliably classified into the same category. This

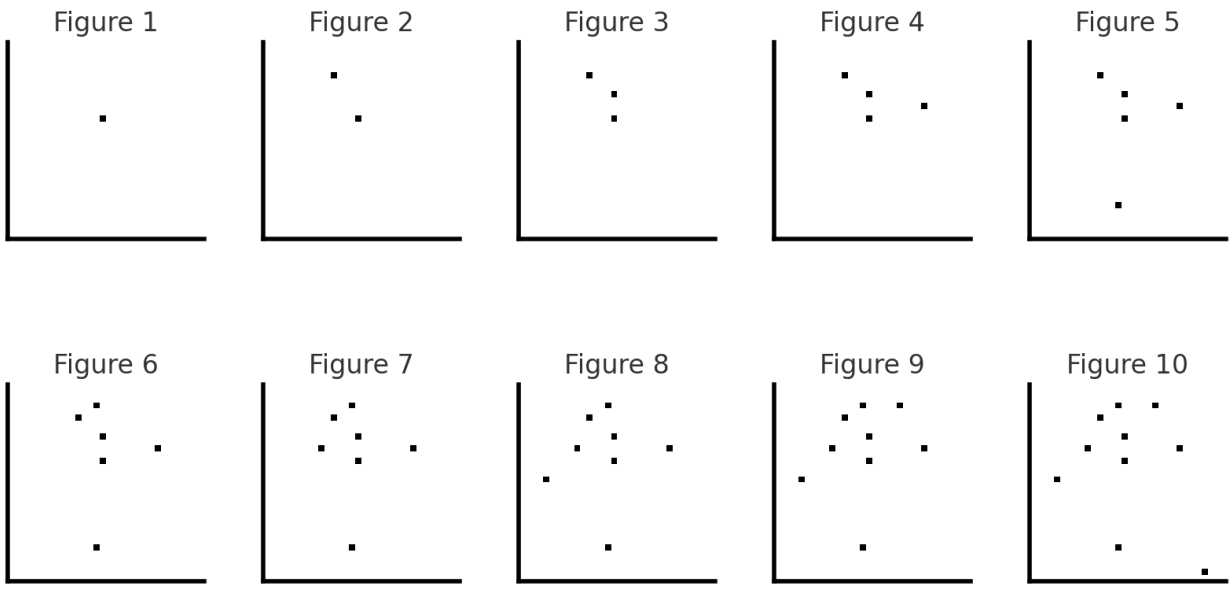

Fig. 3. Ten Random Images Generated

will help us understand whether specific configurations or densities of dots elicit particular responses from the CNN, potentially revealing abstract features that the model mistakenly associates with learned patterns from the training data.

In addition to simple increases in dot quantity, we will introduce a variety of spatial distributions and configurations to further examine CNN sensitivity to abstract noise. These configurations will include clustered dots, evenly distributed patterns, and more structured formations, allowing us to evaluate whether certain spatial arrangements are more likely to trigger consistent misclassifications. We will also explore variations such as randomized noise patterns, mixed dot sizes, and different grayscale intensities to simulate more complex types of visual noise. By testing these variations, we aim to determine whether these factors influence classification stability and contribute to the CNN's susceptibility to noise-based adversarial patterns.

Ultimately, our goal is to identify the specific dot configurations and noise patterns that are most effective at deceiving the CNN and causing consistent misclassification. By pinpointing patterns that reliably trigger the same erroneous class prediction, we can better understand the kinds of visual stimuli that disturb the CNN's recognition abilities. This knowledge will not only inform future studies on CNN vulnerability but also provide valuable insights into designing more robust image classification models. With a clearer understanding of how abstract patterns impact CNN decision-making, we can take steps toward developing defenses against adversarial attacks in practical applications, strengthening CNN performance in real-world scenarios where random noise and perturbations may interfere with classification accuracy.

REFERENCES

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 1625-1634, doi: 10.1109/CVPR.2018.00175.

[2] Feng, Weiwei; Xu, Nanqing; Zhang, Tianzhu; Wu, Baoyuan; Zhang, Yongdong. (2023). Robust and Generalized Physical Adversarial Attacks via Meta-GAN. IEEE Transactions on Information Forensics and Security. PP. 1-1. 10.1109/TIFS.2023.3288426.

[3] Chen, Shang-Tse; Cornelius, Cory; Martin, Jason; Chau, Polo. (2019). ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector: Recognizing Outstanding Ph.D. Research. 10.1007/978-3-030-10925-7_4.

[4] Chong Xiang and Arjun Nitin Bhagoji and Vikash Sehwag and Prateek Mittal. (2021). PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking. doi: 10.48550/arXiv.2005.10884

[5] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html. [Accessed: Nov. 9, 2024].

IX. Appendix

The following outlines our timeline for the rest of the semester. We have broken each week up into one objective, which we will aim to complete that week. Specific task descriptions can be found in the **Task** column.

TABLE I
PROJECT TIMELINE

Date	**Objective**	**Task**
10/8	Project Proposal	Write Proposal - all members
10/10	CNN Model Training	Find / Train existing CNN model capable of identifying categorizing landscapes. Train another model with existing defence mechanism. - all members
11/1	Adversarial Attack Implementation	Implement enhanced adversarial algorithms, compare and contrast with non-adversarial image. - all members
12/1	Benchmarking	Apply adversarial attack to existing defence mechanism, compare and contrast the results to the model without defence mechanism.