Examining the Efficacy of FGSM and Adversarial Patches in Fooling Neural Networks: Insights from CIFAR-10 and ImageNet Experiments

Ishrat Patel
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
iipatel@uwaterloo.ca

Jaspreet Bhinder
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
jk3bhind@uwaterloo.ca

***Abstract*—In recent years, the vulnerability of neural networks to adversarial attacks has raised significant concerns regarding their reliability in safety-critical applications. This study investigates the efficacy of two prominent adversarial attack methods, Fast Gradient Sign Method (FGSM) and Adversarial Patches, in fooling neural networks. Our study focuses on evaluating these attacks on two distinct datasets: CIFAR-10 and ImageNet. We trained a Convolutional Neural Network (CNN) on CIFAR-10 and utilized a pre-trained ResNet model for ImageNet experiments. Furthermore, we investigated the transferability of Adversarial Patches across Imagenet to assess the generalizability of adversarial attacks. This exploration aims to determine the transferability of white-box attacks within the same dataset, evaluating whether adversarial patches crafted for one dataset can deceive different neural networks trained on the same dataset. Through extensive experimentation, we analyzed the effectiveness of FGSM and Adversarial Patches in compromising the accuracy of the neural networks on both datasets. Our findings reveal insights into the robustness of neural networks against these attacks. Additionally, our results also reveal that while some attack methods may be more effective than others, neural networks, in general, exhibit vulnerabilities to adversarial attacks across different methodologies. These findings emphasize the importance of developing robust defense mechanisms to bolster the security and reliability of neural networks in real-world applications.**

***Index Terms*—Adversarial Attacks, Adversarial Patch, Fast Gradient Sign Method, Neural Networks**

I. Introduction

Adversarial attacks are characterized by subtle perturbations applied to input data with the intent of causing misclassification by the neural network, pose significant threats to the reliability, security, and trustworthiness of deep learning systems. Understanding the efficacy of different types of adversarial attacks and their impact on neural network models is essential for developing robust and resilient machine learning algorithms capable of withstanding real-world adversarial manipulation.

This research endeavors to investigate the effectiveness of two prominent adversarial attack methods: the Fast Gradient Sign Method (FGSM) and adversarial patches, in deceiving neural networks. The Fast Gradient Sign Method perturbs input data by taking a small step in the direction of the gradient of the loss function with respect to the input, thereby generating adversarial examples with minimal perturbations. Adversarial patches, on the other hand, involve strategically placing imperceptible patches in the input image to induce misclassification by the neural network. By examining the performance of these attacks on two distinct neural network architectures, namely a custom Convolutional Neural Network (CNN) trained on the CIFAR-10 dataset and a pre-trained ResNet34 model trained on the ImageNet dataset (customized), we aim to gain insights into the robustness of these models against adversarial attacks and the transferability of adversarial patches across different network structures.

The primary objective of this study is twofold. Firstly, we seek to evaluate the susceptibility of the custom CNN and ResNet34 models to adversarial attacks, comparing the effectiveness of FGSM and adversarial patches in causing misclassification. This evaluation involves assessing the impact of adversarial attacks on model accuracy and performance metrics under various attack scenarios. Secondly, we aim to investigate how the structural characteristics and complexity of neural network architectures influence the transferability of adversarial patches. To achieve these objectives, we conducted rigorous experiments using standard evaluation metrics, including accuracy, and transferability measures, to quantify the effects of adversarial attacks on model behavior and performance.

This research holds significant implications for the development of secure and reliable machine learning systems. By uncovering the vulnerabilities of neural networks to adversarial attacks and analyzing the transferability of adversarial perturbations across different network architectures, we contribute to a deeper understanding of the challenges and limitations of adversarial machine learning. Ultimately, our findings can inform the design of more resilient models and defense mechanisms against adversarial threats, advancing the progress towards trustworthy AI systems capable of operating effectively in adversarial environments.

The remainder of this paper is organized as follows: Sec-

tion 2 delves into the background of adversarial attacks and the neural network architectures used in this study, Section 3 presents the related work, summarizing previous studies, methodologies, and findings. Section 4 details the methodology employed in our experiments, including the architecture of the models, dataset preparation, and adversarial attack techniques. In Section 5, we explain the experimentation setup and present the results and analyze the findings, elucidating the effectiveness of adversarial attacks and their implications for model robustness. Finally, Section 6 concludes the paper with a summary of key insights and contributions, reaffirming the significance of our research in advancing the understanding and development of adversarially robust machine learning systems.

II. Background

Adversarial attacks on neural networks have become a pressing concern in the field of machine learning and artificial intelligence [1]. These attacks exploit vulnerabilities in the underlying mechanisms of deep learning models, causing them to produce incorrect outputs in response to carefully crafted input data. Understanding the nature of adversarial attacks, their impact on neural network models, and the strategies employed to defend against them is crucial for building trustworthy and resilient AI systems.

In the realm of computer vision, Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid-like data, such as images [2]. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. These layers extract hierarchical features from input images and learn to classify objects or patterns present in the data. CNNs have demonstrated remarkable performance in various computer vision tasks, including image classification, object detection, and segmentation.

Residual Networks (ResNets) are a type of CNN architecture introduced by He et al [3]. in 2015. ResNets are characterized by the use of residual connections, which allow information to flow through the network more efficiently by bypassing certain layers. This architecture mitigates the vanishing gradient problem and enables the training of very deep neural networks with hundreds or even thousands of layers. ResNets have achieved state-of-the-art performance on image classification tasks and have been widely adopted in both research and industry.

Dense Convolutional Networks (DenseNet) are another variant of CNN architecture proposed by Huang et al. in 2017 [4]. DenseNets introduce dense connections between layers, where each layer receives feature maps from all preceding layers and passes its own feature maps to all subsequent layers. This dense connectivity pattern facilitates feature reuse, enhances gradient flow, and encourages feature propagation throughout the network. DenseNets have demonstrated superior performance in image classification and object detection tasks compared to traditional CNN architectures.

SqueezeNet is a lightweight CNN architecture designed for efficient model deployment on resource-constrained devices, such as mobile phones and embedded systems [5]. Proposed by Iandola et al. in 2016, SqueezeNet achieves a comparable level of accuracy to larger CNNs while significantly reducing model size and computational complexity. SqueezeNet achieves this efficiency through various architectural design choices, including the use of 1x1 convolutions, downsampling via stride, and aggressive compression.

Moreover, the CIFAR-10 dataset is a benchmark dataset commonly used for training and evaluating machine learning models for image classification [6]. CIFAR-10 consists of 60,000 32x32 color images across 10 classes, with 6,000 images per class. CIFAR-10 is widely used in research due to its relatively small size and diverse set of classes, making it suitable for rapid experimentation and benchmarking.

Additionally, ImageNet dataset is one of the largest and most widely used datasets in computer vision research. It contains over 14 million labeled images spanning thousands of object categories. ImageNet serves as a crucial resource for training and evaluating image classification algorithms, enabling researchers to develop and benchmark state-of-the-art models. With its vast diversity and scale, the ImageNet dataset facilitates the exploration of complex visual recognition tasks, ranging from identifying common objects to fine-grained categorization.

III. Related Work

Several studies have investigated the vulnerability of neural networks to adversarial attacks and proposed various defense mechanisms to mitigate their impact [7]. This section provides an overview of relevant research in the field of adversarial machine learning, focusing on studies that explore the effectiveness of different attack methods and defense strategies.

Szegedy et al., illuminated these intriguing properties of neural networks through meticulous experimentation and analysis [8]. They systematically explored the semantic meanings of individual units within deep neural networks and examined their discontinuities. Through a series of experiments, they meticulously crafted adversarial negatives—inputs that appear indistinguishable from regular examples but lead to misclassifications when presented to the network. By observing the network's responses to these adversarial negatives, Szegedy et al. revealed the network's vulnerability to subtle perturbations and its seemingly counterintuitive behavior despite achieving high generalization performance. Through rigorous experimentation and insightful analysis, they shed light on these unexpected characteristics, sparking further inquiry into the mechanisms underlying neural network behavior.

Goodfellow et al. [9] introduced the Fast Gradient Sign Method (FGSM) as a straightforward yet powerful technique for generating adversarial examples. FGSM crafts adversarial examples by perturbing the input data in the direction of the gradient of the loss function. FGSM's simplicity and effectiveness have made it a foundational method in the field

of adversarial attacks, serving as a benchmark for subsequent research.

Tom B. Brown et al. introduced a groundbreaking approach to adversarial attacks with their paper on creating universal, robust, and targeted adversarial image patches [10]. Unlike traditional methods that focus on imperceptible changes to inputs or fixed-location attacks, the adversarial patch method generates image-independent patches that are highly salient to neural networks. This method exhibits universality, as the patches are scene-independent and can be used across different scenes without requiring knowledge of lighting, camera angle, or classifier type.

Various defense strategies have been proposed to enhance the robustness of neural networks against adversarial attacks. Madry et al. (2018) [1] introduced adversarial training with Projected Gradient Descent (PGD), a robust optimization method that generates adversarial examples during training to improve model robustness. Zhang et al. (2019) [11] proposed Feature Denoising Networks (FDNs), which incorporate noise regularization into neural network architectures to enhance resilience against adversarial attacks.

Pre-trained models, particularly those trained on large-scale datasets like ImageNet, are also vulnerable to adversarial attacks. Carlini and Wagner (2017) [12] proposed the Carlini-Wagner L2 attack, which aims to generate imperceptible perturbations that maximize the model's classification error. Engstrom et al. (2019) [13] investigated the vulnerability of pre-trained models to adversarial patches, highlighting the importance of robustness testing beyond standard image classification tasks.

Papernot et al. were among the first to highlight the notable phenomenon of adversarial sample transferability within the realm of machine learning. Their seminal work revealed that adversarial samples not only confound models trained using the same machine learning technique but also traverse across different methodologies [14]. Moreover, they enhanced the accuracy and diminished the computational burden of an existing algorithm for constructing model substitutes for machine learning classifiers.

In a pivotal experiment, they showcased the practical implications of their findings by targeting online classifiers hosted by major platforms like Amazon and Google, achieving staggering misclassification rates, respectively, without requiring any insight into the model architecture or parameters. These revelations underscore the urgent need for input validation mechanisms in machine learning algorithms, presenting an ongoing challenge in the field. Papernot et al. advocate for continued efforts to enhance substitute learning techniques to optimize accuracy and the transferability of adversarial samples across targeted models.

In summary, previous research has provided valuable insights into the nature of adversarial attacks, their impact on neural network models, and strategies for defending against them. Our study builds upon this body of work by conducting empirical experiments to evaluate the efficacy of adversarial attacks on different neural network architectures and datasets, with a focus on understanding the transferability of adversarial perturbations across models.

IV. Methodology

A. Implementation Details

The experiments were implemented using Python programming language with the PyTorch deep learning framework [15]. Data visualization was performed using Matplotlib [16] and Seaborn [17] libraries. Google Colab [18] and Kaggle [19] was used as the computing environment for running the experiments, providing access to GPU resources for faster model training and evaluation.

B. Models and Dataset used

Custom CNN Model

A custom CNN model was designed for image classification on the CIFAR-10 dataset. It consists of three convolutional layers, followed by max-pooling layers, and three fully connected (linear) layers.

Architecture of Custom CNN:

- **Convolutional Layers:**
  - **conv1:** 32 filters, kernel size 3x3.
  - **conv2:** 64 filters, kernel size 3x3.
  - **conv3:** 128 filters, kernel size 3x3.
- **Fully Connected Layers:**
  - **fc1:** 512 neurons.
  - **fc2:** 256 neurons.
  - **fc3:** 10 neurons (output layer for CIFAR-10 classes).

Pretrained ResNet34

ResNet34 is a convolutional neural network (CNN) architecture with 34 layers. It belongs to the ResNet (Residual Network) family of architectures, known for its depth and efficiency in image classification tasks.

- **Depth:** 34 layers (including the initial convolution and the final fully connected layer).
- **Building Block:** Basic block with two convolutional layers and a residual connection.
- **Skip Connections:** Residual connections that allow gradients to flow more easily during training.
- **Batch Normalization:** Utilizes batch normalization after convolutional layers.
- **Global Average Pooling:** Ends with a global average pooling layer.

Both the custom CNN and ResNet34 are architectures suitable for image classification tasks. ResNet34 is deeper and more complex, while the custom CNN is simpler, designed specifically for CIFAR-10.

In practice, ResNet34 is often preferred for challenging image classification tasks or when higher accuracy is required. Custom CNNs can be more lightweight and easier to train with limited computational resources.

CIFAR-10: The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 50,000 training images and

10,000 test images [6]. We used a Convolutional Neural Network (CNN) architecture tailored for CIFAR-10 classification tasks.

Customized ImageNet: The customized ImageNet dataset is a curated subset derived from the original ImageNet dataset, consisting of a smaller set of pre-processed images. This specialized dataset retains the structure and labeling of the original ImageNet, comprising five images for each of the 1000 distinct labels in the dataset.

C. Adversarial Attack Techniques

The adversarial attacks that we explored were: FGSM and adversarial patches. We focused on assessing the vulnerability of a custom Convolutional Neural Network (CNN) trained on the CIFAR-10 dataset and Resnet34 trained on ImageNet dataset to these attacks.

1) FGSM: The Fast Gradient Sign Method (FGSM) is a popular and efficient technique for generating adversarial examples. The FGSM generates adversarial examples by perturbing the input data (image) using the gradients of the loss function with respect to the input. The adversarial perturbation is calculated as the sign of the gradient multiplied by a small constant (ϵ) [9].

Mathematically, an adversarial example x_{adv} is generated from a benign input x as follows:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

Where:

- ϵ is a small perturbation magnitude.
- $\nabla_x J(\theta, x, y)$ is the gradient of the loss function J with respect to the input x.
- θ represents the model parameters.
- y is the true label of image x.

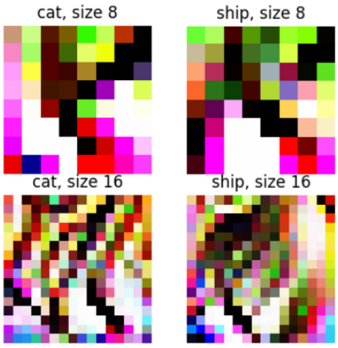

Fig. 1. Patch generated of cat and ship from CIFAR10 against Custom CNN

2) Adversarial Patch: The adversarial patch was developed following the method from [10]. These patches are universal, meaning they can attack any scene, robust to different transformations, and can target any desired output class.

The strategy for finding a targeted adversarial example is as follows: given some classifier $\mathbb{P}[y \mid x]$, some input $x \in \mathbb{R}^n$, some target class $\hat{y}$ and a maximum perturbation ϵ, we want to find the input $\hat{x}$ that maximizes $\log(\mathbb{P}[\hat{y} \mid \hat{x}])$, subject to the constraint that $\|x - \hat{x}\|_\infty \leq \epsilon$. When $\mathbb{P}[y \mid x]$ is parameterized by a neural network, an attacker with access to the model can perform iterated gradient descent on x in order to find a suitable input $\hat{x}$. This strategy can produce a well camouflaged attack, but requires modifying the target image.

This study created the attack by completely replacing a part of the image with our patch, then masked the patch to allow it to take any shape, and then train over a variety of images, applying a random translation, scaling, and rotation on the patch in each image, optimizing using gradient descent. More formally, we have

The final image $\hat{x}$ must be realizable using the fixed mask m applied on the trained patch $\hat{(z)}$ and some transformation $t \in T$. More formally, we define a patch function p corresponding to every transformation $t \in T$ that applies the transformed patch onto the image:

$$p_t(x, z)t(m \odot z) + [t(1 - m) \odot x]$$

where $\odot$ refers to the pixel-wise Hadamard product. Then the final adversarially perturbed image $\hat{x}$ must satisfy $\hat{x} = p_t(x, \hat{(z)})$ for the final trained patch $\hat{z}$ and some $t \in T$.

To obtain the trained patch $\hat{z}$, we use a variant of the Expectation over Transformations (EOT) In this setting, we have a family of affine transformations T, a distance metric d in the transformed space, and the objective is to find a perturbed image $\hat{x}$ satisfying:

$$\hat{x} = \underset{x'}{\text{argmax}}\, \mathbb{E}_{t \sim T}[\log \Pr(\hat{y}|t(x')]$$
$$\text{s.t. } \mathbb{E}_{t \sim T}[d(t(x', t(x))] < \epsilon$$

i.e., The final image should be within ϵ-ball in expectation over all the transformations in T. Instead of having an ϵ bound on the amount of perturbation we are allowed to add, our attack instead finds the solution to the following unconstrained optimization problem:

$$\hat{z} = \underset{z' \in \mathbb{R}^n}{\text{argmax}}\, \mathbb{E}_{t \sim T}(\mathbb{P}[\hat{y}|p_t(x, z')])$$

The final perturbed image $\hat{x}$ can then be obtained by choosing any $t \in T$ and applying the patch function to $\hat{z}$, i.e. $\hat{x} = p_t(x, \hat{z})$

Because the above attack technique leaves the transformation t as an open variable, the attacker is free to choose a t that has desirable properties for the attack. For example, in the real world, an attacker could choose to place a trained patch $\hat{z}$ in an inconspicuous place in the background of an image.

D. Evaluation Method

We evaluated the performance of the models under adversarial patch attacks using top-1 accuracy, which measures the percentage of correctly classified images. Additionally, we computed the top-5 accuracy, which assesses how often the true label falls within the model's top 5 predictions. As models typically perform well on these predictions, we report the error (1 - accuracy) instead of the accuracy to provide a more informative metric. To conduct the evaluation, we

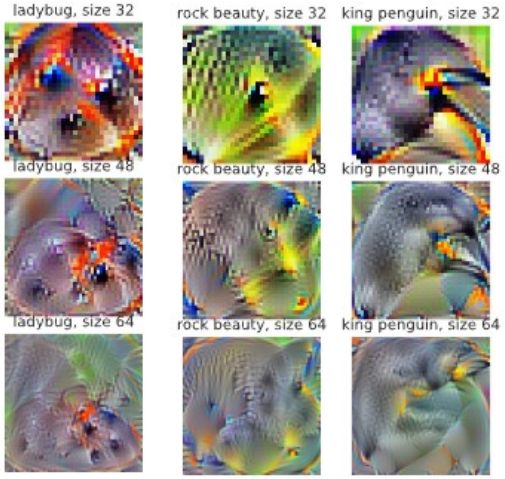

Fig. 2. Patch generated of ladybug, rock beauty and penguin from ImageNet against Resnet34

implemented a validation function that computes both top-1 and top-5 accuracy metrics on the validation dataset.

V. Experiment

The parameters and hyperparameters used in the experiments are summarized as follows:

For CIFAR-10 CNN:

- **Training Epochs**: 50
- **Batch Size**: 64
- **Optimizer**: Adam
- **Learning Rate**: 0.001

For Pre-trained ResNet-34

- **Training Epochs**: Not applicable (pre-trained)

A. Training, Testing, and Evaluation Procedures

The training and testing procedures were as follows:

- **CIFAR-10 CNN**:
  - **Training**: The CNN was trained on the CIFAR-10 training dataset for 50 epochs with a batch size of 64 using the Adam optimizer with a learning rate of 0.001.
  - **Testing Accuracy**: 92.5%
  - **Epsilons**: The model was evaluated using FGSM adversarial attacks with epsilon values of 0, 0.01, 0.02, 0.03, 0.04, and 0.05. The results are presented in Table I for Top-1 error and Table II for Top-5 error.
  - **Patches**: The model was evaluated using adversarial patches with size 8x8 pixels and 16x16 pixels.
- **Pre-trained ResNet-34 on ImageNet**:
  - **Test Accuracy**: 93%
  - **Epsilons**: The model was evaluated using FGSM adversarial attacks with epsilon values of 0, 0.01, 0.02, 0.03, 0.04, and 0.05. The results are presented in Table I for Top-1 error and Table II for Top-5 error.
  - **Patches**: The model was evaluated using adversarial patches with size 32x32 pixels, 48x48 pixels and 64x64 pixels.

Fig. 3. Patch attack specifically targeting the 'cat' class on our custom Convolutional Neural Network (CNN)

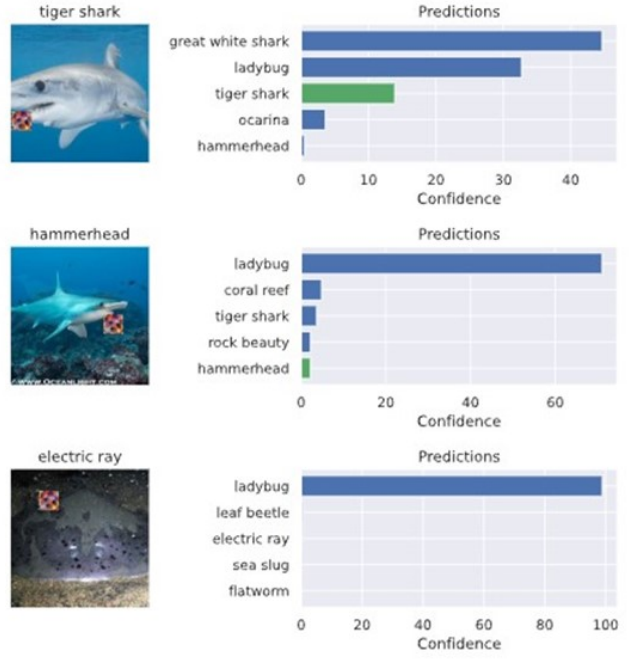

Fig. 4. Patch attack specifically targeting the 'ladybug' class on our ResNet34

B. FGSM Attacks

In this experiment, FGSM attacks were performed on both the custom CNN trained on CIFAR-10 and the ResNet-34 model trained on ImageNet. The attacks were conducted with changing epsilon values: 0, 0.01, 0.02, 0.03, 0.04, and 0.05. For each epsilon value, we evaluated the top-1 error and top-5 error rates.

C. Adversarial Patch Attacks

We conducted experiments for custom CNN, trained on CIFAR-10, by employing adversarial patches of varying sizes—8x8 pixels and 16x16 pixels (Fig. 1)—strategically embedding them within input images during inference. We extended our investigation to the vulnerability of a pre-trained ResNet34 model, trained on the custom ImageNet dataset, to adversarial patch attacks. Adversarial patches of various sizes—32x32 pixels, 48x48 pixels, and 64x64 pixels (Fig. 2)—were meticulously crafted to exploit vulnerabilities in the model's decision boundaries. The training regimen for these patches involved precise gradient calculations focused solely on the patch rather than every pixel in the input image. Employing stochastic gradient descent (SGD) optimization, we ensured the patches' efficacy in deceiving the model across diverse input images(Fig. 3, 4).

D. Transferability of Adversarial Patches

In addition to evaluating the effectiveness of adversarial patches on the original model, we investigated their transferability across different neural network architectures. Initially, we crafted a 32x32 adversarial ladybug patch utilizing the ImageNet dataset, specifically trained to deceive a ResNet34 model. Subsequently, we rigorously evaluated the transferability of this adversarial patch by subjecting it to ResNet50, SqueezeNet1.0, and DenseNet121 architectures.

E. Results

1) Effect of Epsilon on Accuracy: The robustness of both the CNN and the pre-trained ResNet-34 against adversarial attacks was evaluated first using the Fast Gradient Sign Method (FGSM) with varying epsilon values. The results indicate varying degrees of vulnerability to adversarial attacks between the two models.

While both models exhibited a significant decline in accuracy when subjected to the same adversarial attack, the CNN showed a more pronounced drop compared to the ResNet-34, as highlighted in Fig. 9 and Fig. 10 .

2) Visualizing Model Predictions Before and After Adversarial Perturbation: Visual representation of model predictions provides valuable insights into the susceptibility of models to adversarial attacks. Fig. 5 and Fig. 6 depict the predictions of the CNN model on CIFAR-10 images before and after perturbation with an epsilon value of 0.02, respectively. Similarly, Fig. 7 and Fig. 8 represent the predictions of the pre-trained ResNet-34 model on custom ImageNet images under the same conditions.

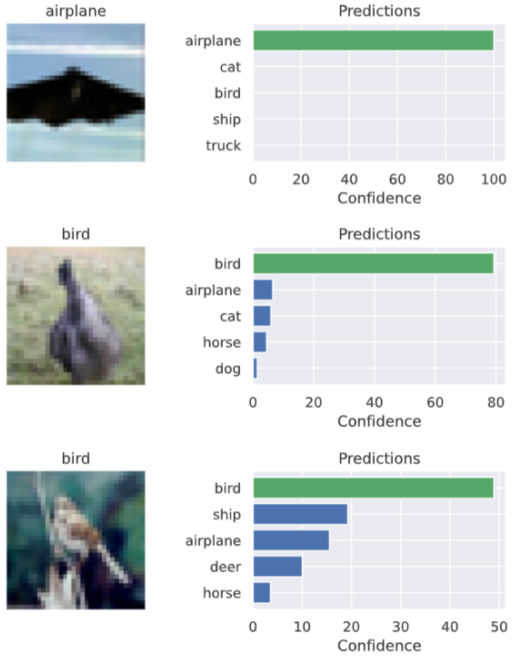

Fig. 5. CNN predictions on CIFAR-10 images before perturbation. High confidence predictions of *bird* and *airplane* are evident.

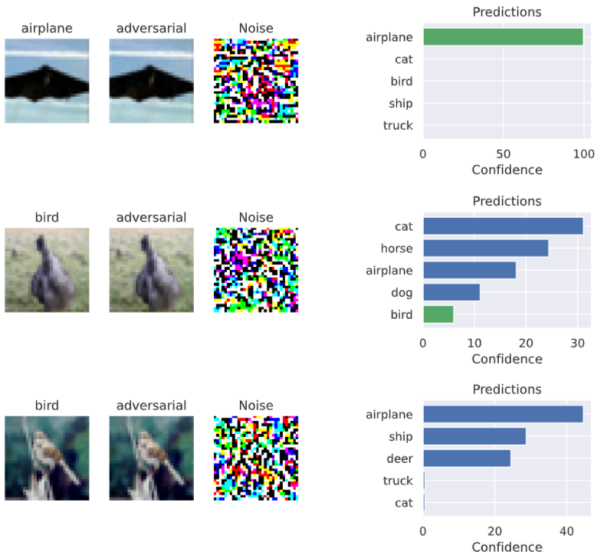

Fig. 6. CNN predictions on CIFAR-10 images after perturbation with epsilon=0.02. Despite minimal noise, the model misclassifies *bird* as *cat* and airplane but retains the correct prediction for *airplane*.

From the visual observations, it's clear that both models exhibit a vulnerability to adversarial perturbations. While the CNN model misclassifies the *bird* as *cat* and retains the correct prediction for *airplane*, the ResNet-34 model misclassifies both *goldfish* and *tench*. Interestingly, even with minimal perturbations that are imperceptible to the human eye, the models' predictions are significantly altered, underscoring the importance of adversarial robustness in deep learning models.

The vulnerability of the models to FGSM attacks is further

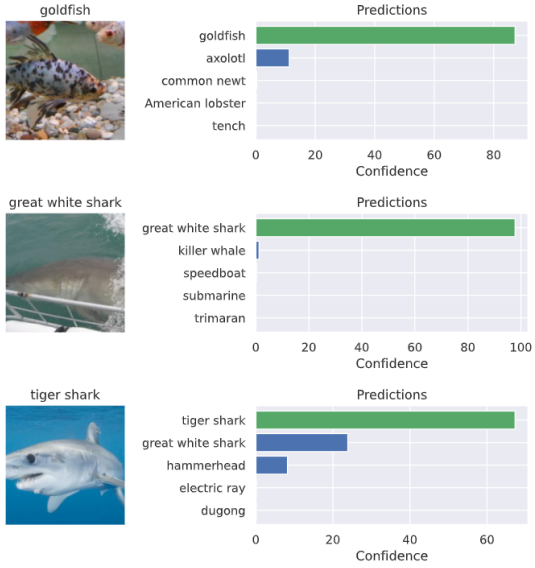

Fig. 7. ResNet-34 predictions on custom Tiny ImageNet images before perturbation. High confidence predictions of *goldfish*, *great white shark* and *tiger shark* are evident.

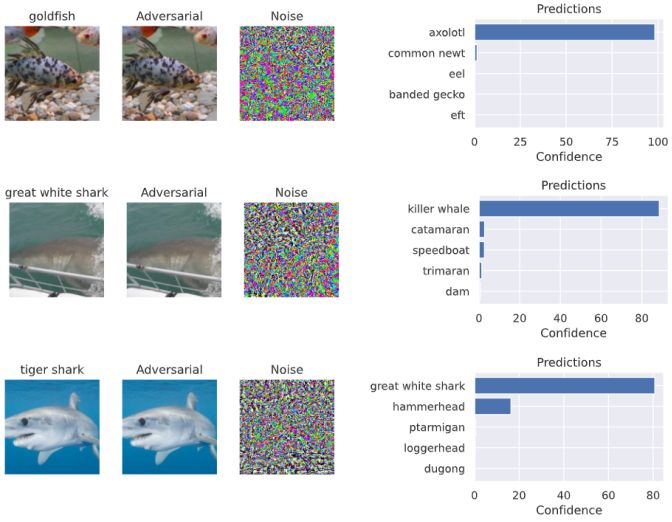

Fig. 8. ResNet-34 predictions on custom Tiny ImageNet images after perturbation with epsilon=0.02. Despite minimal noise, the model misclassifies *goldfish* as *axolotal*, *great white shark* as *killer whale* and *tiger shark* as *great white shark*.

corroborated by the error tables I, II for both models.These tables highlight the escalation of top errors as epsilon values increase, further emphasizing the models' susceptibility to adversarial attacks.

3) Effect of Patch Size on Accuracy: From the observations of Table III, it is evident that as the size of the adversarial patch increases, the effectiveness of fooling the neural network also increases. For instance, in the CIFAR-10 dataset, for classes like Airplane, Bird, Cat, Horse, and Ship, we observe a significant improvement in accuracy when transitioning from a patch size of 8x8 to 16x16 pixels. The accuracy jumps from around 30% to above 95% for most classes, indicating that

TABLE I
ERROR RATES FOR CUSTOM CNN ACROSS EPSILON VALUES

Epsilon	Top-1 Error (%)	Top-5 Error (%)
0	5.96	0.02
0.01	35.3	0.57
0.02	61.83	3.05
0.03	76.57	7.5
0.04	82.55	12.69
0.05	85.84	17.55

TABLE II
ERROR RATES FOR PRETRAINED RESNET-34 ACROSS EPSILON VALUES

Epsilon	Top-1 Error (%)	Top-5 Error (%)
0	19.38	4.38
0.01	83.66	44.20
0.02	93.74	60.84
0.03	95.50	68.30
0.04	96.36	72.30
0.05	96.90	74.74

larger patches are more successful in manipulating the model's predictions. This trend underscores the potency of adversarial patches and highlights their ability to exploit vulnerabilities in neural network architectures, particularly when they cover a larger portion of the input image.

On the other hand, while examining Table IV, focusing on the results obtained from the custom ImageNet dataset when patches were generated against ResNet34, we observe interesting trends. Despite increasing the patch size from 48x48 to 64x64 pixels, the accuracy improvements are relatively marginal for classes like King Penguin, Ladybug, Rock Beauty, Speedboat, and Toaster. For instance, the accuracy for the Toaster class only shows a slight increase from 90.48% to 98.58% when transitioning from a 48x48 to a 64x64 patch. This phenomenon suggests that beyond a certain threshold, enlarging the patch size may not yield significant improvements in fooling the neural network. It implies that the model's decision boundaries may already be effectively exploited by patches of smaller sizes, and further increasing the patch size does not substantially enhance their effectiveness in deceiving the network. This finding underscores the nuanced relationship between patch size and their efficacy in adversarial attacks, suggesting the importance of carefully considering the size of the adversarial patch in crafting effective attacks against neural networks.

TABLE III
AVERSARIAL PATCH RESULTS FOR CIFAR-10 CNN

Class Name	Patch size 8x8	Patch size 16x16
Airplane	33.07%	96.74%
Bird	26.34%	96.77%
Cat	32.82%	95.16%
Horse	32.45%	95.94%
Ship	33.28%	96.19%

TABLE IV
AVERSARIAL PATCH RESULTS FOR IMAGENET RESNET34

Class Name	32x32	48x48	64x64
King Penguin	73.01%	92.11%	97.79%
Ladybug	89.54%	95.96%	98.49%
Rock Beauty	72.70%	90.77%	96.63%
Speedboat	62.17%	89.33%	95.77%
Toaster	48.89%	90.48%	98.58%

4) Effect of Network Architecture on Transferability: Table V presents the transferability results obtained by testing a ladybug patch generated against ResNet34 on three different models: ResNet50, DenseNet121, and SqueezeNet1.0. The accuracies obtained for each model are as follows: 54.36% for ResNet50, 57.02% for DenseNet121, and 67.00% for SqueezeNet1.0.

Analyzing these accuracies, we can observe that the ladybug patch exhibits varying degrees of effectiveness in fooling different models. Notably, the highest accuracy of 67.00% is achieved when the patch is tested against SqueezeNet1.0, followed by 57.02% for DenseNet121 and 54.36% for ResNet50.

Regarding the hypothesis that simpler architectures tend to learn more transferable features compared to deeper models, the results from Table 3 do not entirely support this notion. While SqueezeNet1.0, which is a simpler architecture designed for efficient model deployment, achieves the highest accuracy among the three models tested, DenseNet121, a deeper and more complex architecture, outperforms ResNet50 in terms of transferability.

These findings suggest that the transferability of adversarial patches is influenced by various factors beyond just the complexity of the model architecture. Factors such as the architectural design, feature representation, and decision boundaries of the models also play significant roles. Therefore, while simpler architectures may indeed exhibit favorable transferability properties in some cases, the relationship between model complexity and transferability is not straightforward and requires careful consideration of multiple factors.

TABLE V
TRANSFERABILITY RESULTS

Model	Resnet50	DenseNet121	SqueezeNet1.0
Results	54.36%	57.02%	67.00%

TABLE VI
ACCURACY OF MODELS UNDER FGSM AND PATCH ATTACK

Model	**FGSM (Epsilon=0.02)**	**Patch Attack**
Custom CNN	61.3%	95.16% (cat,8)
ResNet	93.74%	95.96% (ladybug, 48)

Table VI presents the accuracy comparison of FGSM (Fast Gradient Sign Method) and Patch Attack, on custom CNN trained and pretrained ResNet.

Under the FGSM (Epsilon=0.02) attack, the table showcases the accuracy of the models in correctly misclassifying images when subjected to perturbations introduced by the attack. For instance, the Custom CNN achieved an accuracy of 61.3%, indicating that it misclassified approximately 38.7% of the images when subjected to FGSM perturbations.

On the other hand, under the Patch Attack, the accuracies represent how many images were classified with the target class as the highest prediction. For example, the Custom CNN achieved an accuracy of 95.16% under the Patch Attack, with 'cat' being the target class, and the size of the patch being 8 pixels. This implies that the majority of the images were classified as 'cat' by the network when subjected to the adversarial patch.

Similarly, for the ResNet model, the FGSM accuracy is 93.74%, indicating its susceptibility to misclassification under FGSM perturbations. However, under the Patch Attack, the accuracy is 95.96%, with 'ladybug' being the target class and the patch size being 48 pixels. This suggests that the majority of the images were classified as 'ladybug' by the ResNet model when subjected to the adversarial patch.

Overall, the table provides insights into the comparative effectiveness of FGSM and Patch Attack methods in influencing the predictions of the models under consideration, highlighting the potential vulnerabilities of neural networks to adversarial attacks.

These results highlight the nuanced vulnerabilities each model presents against different adversarial techniques. This also underscores the necessity for a comprehensive evaluation of model robustness against a variety of adversarial techniques in real-world applications.

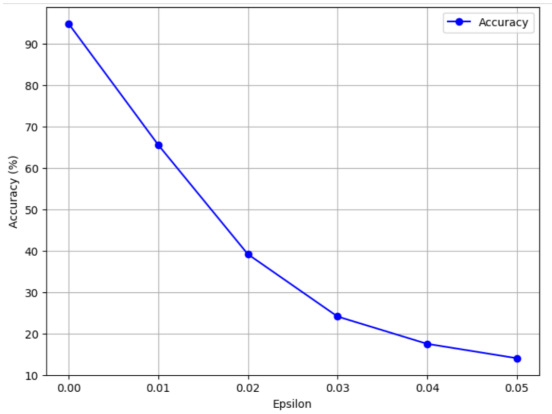

Fig. 9. Decline in Accuracy of custom CNN with increasing Epsilon values

VI. DISCUSSION

The results obtained from our experiments shed light on the susceptibility of deep learning models to adversarial attacks, specifically focusing on the effectiveness of FGSM (Fast Gradient Sign Method) and Patch Attack techniques. The experiments encompassed two primary models: a custom

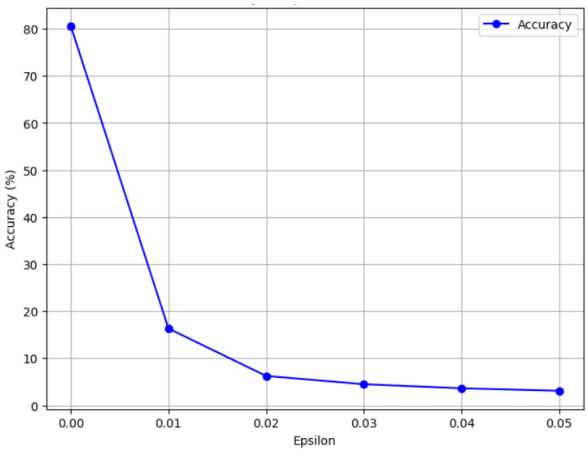

Fig. 10. Decline in Accuracy with Epsilon for ResNet-34 on Tiny ImageNet-400

Convolutional Neural Network (CNN) trained on the CIFAR-10 dataset and a pre-trained ResNet-34 model on Custom ImageNet.

The experiments conducted with FGSM attacks demonstrated that both the CNN and ResNet-34 models exhibited a notable decline in accuracy as the epsilon values increased. Notably, the CNN model showed a more pronounced susceptibility to adversarial perturbations compared to the ResNet-34 model. Visual representations of model predictions before and after perturbation highlighted the subtle yet impactful alterations in predictions induced by adversarial perturbations, underscoring the importance of robustness evaluation in deep learning models.

Furthermore, our investigation into adversarial patch attacks revealed intriguing insights into the relationship between patch size and model vulnerability. While larger patches were generally more successful in manipulating the predictions of the CIFAR-10 CNN model, the effectiveness of enlarging patch sizes beyond a certain threshold appeared to diminish for the ResNet-34 model. These observations underscore the nuanced dynamics between patch size and efficacy in adversarial attacks, emphasizing the need for tailored mitigation strategies based on the target model architecture and the dataset used.

Additionally, our exploration of the transferability of adversarial patches across different neural network architectures highlighted the varying degrees of effectiveness in fooling different models. Contrary to the hypothesis that simpler architectures tend to exhibit more transferable features, our results demonstrated nuanced dependencies on factors beyond model complexity, such as architectural design and feature representation.

Moreover, several studies have highlighted the detrimental impact of adversarial attacks across various fields. For instance, research in medical imaging has shown how adversarial attacks can lead to misdiagnosis or incorrect medical predictions, potentially jeopardizing patient safety [20], [21]. Similarly, in the context of autonomous vehicles, adversarial attacks can manipulate sensor inputs or classification systems, leading to hazardous driving conditions and compromising passenger safety [22].

Also, in the realm of finance, adversarial attacks on fraud detection systems can lead to erroneous classification of transactions, resulting in financial losses for individuals and organizations [23]. Similarly, in the field of cybersecurity, adversarial attacks can exploit vulnerabilities in intrusion detection systems, allowing malicious actors to evade detection and compromise network security [24].

Moreover, in industrial settings, adversarial attacks on predictive maintenance systems can disrupt critical operations by providing false alerts or concealing genuine maintenance requirements, leading to equipment failures and production downtimes [25]. In the domain of natural language processing, adversarial attacks on sentiment analysis models can manipulate public opinion by generating deceptive content or biasing sentiment analysis results [26], thereby influencing decision-making processes and public discourse.

These examples underscore the multifaceted nature of adversarial attacks and their potential to undermine the integrity, reliability, and safety of AI systems across diverse domains. As such, there is an urgent need for robustness evaluation and mitigation strategies to safeguard against adversarial threats and ensure the trustworthiness and effectiveness of AI technologies in real-world applications.

Ethical considerations surrounding the deployment of adversarial techniques in real-world applications are paramount [27]. As demonstrated by our experiments, adversarial attacks pose significant threats to the reliability and trustworthiness of deep learning systems, potentially leading to erroneous outcomes with far-reaching consequences. In light of these risks, it is imperative for researchers and practitioners to prioritize the development of robust and resilient deep learning models through rigorous evaluation and mitigation strategies [28]. Additionally, efforts must be made to raise awareness about the vulnerabilities of AI systems to adversarial attacks and to foster interdisciplinary collaborations aimed at addressing these challenges.

In conclusion, our experiments underscore the importance of robustness evaluation and mitigation strategies in deep learning models to mitigate the risks posed by adversarial attacks. By elucidating the vulnerabilities of neural networks to adversarial techniques and emphasizing the need for ethical considerations in AI development, our research contributes to the broader discourse on AI safety and reliability.

ACKNOWLEDGMENT

We extend our sincere gratitude to Professor Elliot Creager for his invaluable guidance and support throughout the research process. Professor Creager's expertise and constructive feedback have been instrumental in shaping this study and enhancing its quality. We are deeply grateful for his unwavering commitment and dedication to our academic and research endeavors.

REFERENCES

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[2] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[5] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[6] A. Krizhevsky, "The CIFAR-10 dataset," Online, 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html

[7] M. Ozdag, "Adversarial attacks and defenses against deep neural networks: A survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918319884

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint*, 2013. [Online]. Available: https://arxiv.org/abs/1312.6199

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[10] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[11] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 501–509.

[12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[13] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *International conference on machine learning*. PMLR, 2019, pp. 1802–1811.

[14] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," https://pytorch.org/, 2019, accessed: 2024-04-19.

[16] J. D. Hunter, "Matplotlib: Python plotting," https://matplotlib.org/, 2007, accessed: 2024-04-19.

[17] M. Waskom *et al.*, "Seaborn: Statistical data visualization," https://seaborn.pydata.org/, 2020, accessed: 2024-04-19.

[18] Google, "Google Colaboratory," https://colab.research.google.com/, accessed: April 18, 2024.

[19] "Kaggle," https://www.kaggle.com/, accessed: April 19, 2024.

[20] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

[21] S. Kaviani, K. J. Han, and I. Sohn, "Adversarial attacks and defenses on ai in medical imaging informatics: A survey," *Expert Systems with Applications*, vol. 198, p. 116815, 2022.

[22] T. Wu, X. Ning, W. Li, R. Huang, H. Yang, and Y. Wang, "Physical adversarial attack on vehicle detector in the carla simulator," *arXiv preprint arXiv:2007.16118*, 2020.

[23] I. Fursov, M. Morozov, N. Kaploukhaya, E. Kovtun, R. Rivera-Castro, G. Gusev, D. Babaev, I. Kireev, A. Zaytsev, and E. Burnaev, "Adversarial attacks on deep models for financial transaction records," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2868–2878.

[24] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[25] O. Gungor, T. Rosing, and B. Aksanli, "Stewart: Stacking ensemble for white-box adversarial attacks towards more resilient data-driven predictive maintenance," *Computers in Industry*, vol. 140, p. 103660, 2022.

[26] G. A. de Oliveira Júnior, R. T. de Sousa Jr, R. de Oliveira Albuquerque, and L. J. G. Villalba, "Adversarial attacks on a lexical sentiment analysis classifier," *Computer Communications*, vol. 174, pp. 154–171, 2021.

[27] L. Adomaitis and R. Oak, "Ethics of adversarial machine learning and data poisoning," *Digital Society*, vol. 2, no. 1, p. 8, 2023.

[28] P. Delobelle, P. Temple, G. Perrouin, B. Frénay, P. Heymans, and B. Berendt, "Ethical adversaries: Towards mitigating unfairness with adversarial machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 23, no. 1, pp. 32–41, 2021.