CS 199: Torchattacks and ResNet50:
A Comparative Analysis of Adversarial Attack Performance

ENCARNACION, Stephen
RAMIREZ, Dean
RECUERDO, Diane

Computer Security Group
Department of Computer Science
University of the Philippines

1 Introduction

With the advent of technology, data are being produced at an unprecedented rate. Extracting insights from these data is crucial for the creation of intelligent applications (Sarker, 2021). To process the sheer amount of information available, innovative solutions have emerged, with automation taking center stage. Among these innovations, image classification stands out as a crucial component of computer vision, allowing for the automatic classification of images based on their visual content. This serves many purposes and is fundamental for various applications such as facial recognition, medical imaging, autonomous vehicles, and security systems.

While people may think that computers see images the same as them, this is not the case— the way computers perceive images is fundamentally different from human vision. Boesch (2024) describes this process: instead of seeing objects and shapes, computers interpret images through a matrix of numbers, with each number representing a specific color value of its corresponding pixel. To enable a computer to make sense of these numerical values, the image must first undergo preprocessing, as stated by Sanghvi (2020). This critical stage improves the quality of the data and standardizes the images, ensuring that the computer can proceed to the succeeding stages and make accurate assumptions about the contents of the image. In this stage, various adjustments are made to the image, such as resizing, cropping, rotation, and normalization. Afterward, the computer can start the process of feature extraction, wherein it can determine the image's edges, corners, textures, and other significant attributes. The detected patterns are typically unique to certain classes of images (SuperAnnotate, 2023), leading to distinct classifications. Once the computer has learned to identify these key image features from the training data, it can use this knowledge to classify new, previously unseen images.

Initially, manual techniques were employed for feature extraction. Researchers aimed to handcraft specific algorithms to detect and describe features within the images. While these manual techniques were effective to some extent, they had limitations in terms of flexibility and adaptability to different types of images. To address these limitations, Convolutional Neural Networks (CNNs) were developed. CNNs represent a significant advancement in the field of image recognition as they automate the process of

feature extraction and classification. Instead of manually defining features, CNNs learn to identify them through training on very large datasets. CNNs consist of multiple layers that progressively extract higher-level features from the raw pixel values. The initial layers might detect simple edges and textures, while deeper layers can identify more complex patterns, such as shapes or even specific objects like faces or cars. This hierarchical approach allows CNNs to build a detailed and abstract representation of the image, making them highly effective for tasks that involve the classification of images.

As image classifiers become more prominent in various real-world applications, their security and robustness have to then be considered. One of the most significant threats to these models is adversarial attacks. These types of attacks introduce perturbed versions of the input image that seem normal to human observers but have the capacity to trick the classifier into labeling it as something else entirely. Most adversarial attacks happen on the digital platform, with the perturbations being imperceptible, following a certain algorithm to create noise that is designed to maximize the chance of misclassification.

This study aims to investigate the complex discipline of adversarial machine learning, including its background, adversarial attacks, and related defensive strategies designed to strengthen machine learning models' resilience.

2 Review of Related Literature

The fourth Industrial Revolution has paved the way for an era rich in data where nearly every aspect of human activity can be quantified. In order to understand the data effectively, the application of artificial intelligence, particularly machine learning (ML), is imperative. Machine learning encompasses various algorithms such as supervised, unsupervised, semi-supervised, and reinforcement learning, with deep learning standing out as a powerful tool for large-scale data analysis. The selection of appropriate ML algorithms is crucial, as their performance can vary significantly based on data characteristics and the specific requirements of different application domains (Sarker, 2021).

The advent of big data has transformed how we work and think, enhanc-

ing decision-making processes. However, the rapid expansion and complexity of big data present challenges for traditional machine learning tools, which often struggle in analyzing sheer amounts of information efficiently. As data continue to grow in volume, velocity, and variety, new approaches are necessary to manage and extract valuable insights. Effective big data analysis requires overcoming issues related to computational complexity, classification accuracy, and data heterogeneity. Techniques like deep learning and decision tree learning have shown promise in addressing these challenges, enabling more robust and scalable data analysis solutions. Future research must focus on developing innovative methods and frameworks to improve the ability of machine learning tools to handle big data, thereby enhancing its utility in various sectors such as healthcare, transportation, and businesses (Vishnu & Rajput, 2019).

Image classification has many applications. In particular, it has its uses in the field of agriculture, wherein it can be utilized to classify fruits and vegetables. Chaudhari & Waghmare (2022) discuss feature extraction as a crucial step in image processing, particularly in describing and classifying fruits. Feature extraction involves identifying attributes such as color, size, and shape, which are essential for various applications like fruit classification and quality assessment.

Adversarial machine learning focuses on creating deceptive input data to manipulate the predictions of machine learning models to create confusion. According to Boesch (2024), this field encompasses both attack mechanisms, such as generating adversarial examples, and defense strategies aimed at detecting these manipulative inputs. These techniques find significant applications in image classification and spam detection with extensive studies conducted on the former. Hashemi-Pour (2023) further emphasizes that these adversarial attacks are designed to be effective across different model architectures and datasets. By introducing subtle perturbations or noise to images, attackers can cause classifiers to misclassify them, with the alterations remaining indistinguishable from humans.

The goals of these adversarial attacks are diverse, as stated by Chakraborty, Alam, Dey, Chattopadhyay, and Mukhopadhyay (2021). They include confidence reduction, where the adversary reduces the confidence level of the model's predictions; misclassification, which involves altering the output clas-

sification to a different class; targeted misclassification, where inputs are crafted to produce a specific incorrect class; and source/target misclassification, which aims to classify a particular input into a predefined incorrect class.

Adversarial machine learning attacks can be categorized into three (3) main types (Hashemi-Pour, 2023). The most common, evasion attacks, occur during the deployment phase wherein the attacker manipulates data to deceive previously trained classifiers. These are achieved by introducing subtle perturbations to cause misclassification without altering the training data. Data poisoning attacks then occur during the training phase, where attackers introduce malicious samples into the training dataset, thereby compromising the learning process and reducing the accuracy of the model. Lastly, model extraction attacks, as Boesch (2024) describes it, involve the probing of a black-box system to either reconstruct the model or extract sensitive training data. This is significant when the training data or the model itself is confidential.

Hashemi-Pour (2023) discusses that these attacks can be further categorized based on the attacker's knowledge of the model into white-box and black-box attacks. In the former, the attacker has access to the model's vital information, such as the data it was trained on, allowing it to identify vulnerabilities and exploit these weaknesses. Conversely, in the latter, the attacker has no knowledge of the model's internal workings and relies on probing the system with carefully crafted inputs to observe outputs and make assumptions. While protecting a model from external threats can mitigate the risk of white-box attacks, black-box attacks remain a threat as attackers can still extract sensitive data by analyzing the outputs.

Duan et al. (2020) expanded the scope of adversarial attacks by categorizing them into digital and physical-world settings. In the digital setting, attackers feed perturbed images directly into the classifiers, while in the physical-world setting, adversarial images are presented to cameras, from which classifiers can take its input from. They identified three properties of adversarial attacks: strength, which is the ability of the attack to fool neural networks; stealthiness, which is about the imperceptibility of the changes to human observers; and flexibility, which is the degree to which the attack can change its form. Digital attacks often utilize small perturbations to avoid detection by human observers. For physical world attacks, perturbations

need to be larger and unrestricted, as subtle changes may not be picked up by the camera. This differentiation underscores the varying challenges and techniques needed for adversarial attacks in different environments.

Brown et al. (2018) introduced a specific type of adversarial attack in the physical world known as adversarial patches. Unlike typical adversarial examples that require minimal and inconspicuous perturbations, adversarial patches are easily detected by human observers. These are designed to be the most salient features in an image, leading to misclassification by the classifiers. It was demonstrated that these patches could be universally applied to various images, maintaining effectiveness despite changes in location, rotation, and scale.

In the digital world, attackers create adversarial images by injecting malicious noise into clean digital images to deceive machine learning models. This process is confined entirely within the computer and does not involve real-world objects. There are two main methods for crafting perturbations, as described by Bajaj and Vishwakarma (2024), namely one-shot and iterative. In the one-shot method, perturbations are computed using the gradient of the loss with respect to the input images. When this perturbation is added to a legitimate input, it forms an adversarial sample, which crosses the decision boundary of its true class with a single perturbation. On the other hand, the iterative method involves adding small perturbations iteratively to the legitimate input to craft a more optimal adversarial example. Although this iterative process has higher computational costs, it is preferred by attackers for generating more optimal adversarial examples.

Various digital attacks have already been created, with each algorithm differing from another. Torchattacks is a PyTorch library developed by Hoki Kim (2021) to generate adversarial examples from different types and test the robustness of deep learning models. It includes a variety of adversarial attack algorithms such as including Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Carlini & Wagner's (CW) attack, Random+FGSM (R+FGSM), Projected Gradient Descent (PGD), EOT+PGD (EOTPGD), PGD in TRADES (TPGD), FGSM in fast adversarial training (FFGSM), and MI-FGSM (MIFGSM), among others.

Adversarial examples are also important in evaluating the robustness of

machine learning models. It is imperative that they correctly classify the given inputs, especially in sensitive tasks such as medical imaging and autonomous driving. Given that these adversarial examples fool machine learning models despite having very small perturbations to them from their original input this is a significant field of study. Several studies have evaluated the robustness of these models against adversarial examples. Villegas-Ch et al. (2024) assessed the robustness of VGG16, an image classification model, against adversarial attacks namely the FGSM, PGD, and CW attacks. Results showed a 25% drop in the model's accuracy when presented with adversarial images generated using FGSM and PGD and an even more significant decline of 35% with CW attack-generated adversarial images. In 2021, Buzhinsky et al. evaluated the robustness of generative models concerning "natural" perturbations as opposed to the calculated perturbations that do not exist in nature. Natural perturbations include rotations, changes in brightness, or more high-level changes. In 2016, a study on the adversarial defense, defensive distillation, an approach that can increase the robustness of an arbitrary neural network and reduce the success rate of certain attacks' ability to find adversarial examples from 95% to 0.5%, shows that the Carlini and Wagner attack is still effective against the said defense with 100% probability of success (Carlini, 2016).

ResNet50 is a state-of-the-art image classification model developed by Microsoft Research in 2015. It has been trained on large datasets and achieved promising results on the ImageNet dataset. This dataset contains more than 14 million images and 1000 classes (Kundu, 2023). On this, ResNet50 attained an error rate of 22.85%, comparable to human performance with an error rate of 5.% (Kundu, 2023), and a 92.2% top-5 classification accuracy (Mostafid, 2023). ResNet is one of the typical models adversaries use to perform adversarial attacks (Bajaj & Vishwakarma, 2024).

ResNet50 has 50 layers (Mostafid, 2023). Its architecture is partitioned into four main parts: the convolutional layers, the identity block, the convolutional block, and the fully connected layers. The convolutional layers extract the features from the input image. The identity and convolutional blocks then process and transform these features. Lastly, the fully connected layers make the final decision. It makes use of residual blocks allowing the network to bypass layers and mitigate vanishing gradients. Vanishing gradients is when gradients in the deeper layers become very small and training

those layers becomes difficult (Kundu 2023).

3 Methodology

3.1 Research Design

3.1.1 Research Objective

The study aims to perform a comparative analysis of the performance of the convolutional neural network (CNN) image classification model ResNet50 when classifying adversarial images generated using various adversarial attack types from the torchattacks library. The study uses several metrics such as accuracy, precision, recall, F1 score, and attack execution time to evaluate the model's performance, gain insights into the robustness of ResNet50 against adversarial attacks, and identify potential areas for improving its resilience against such attacks.

3.1.2 Scope and Delimitations of the Study

The researchers utilized the ResNet50 model, the torchattacks library, and nine adversarial attack types from the library: FGSM, PGD, CW, PGD L2, NIFGSM, Pixle, SINIFGSM, VMIFGSM, and VNIFGSM. These adversarial images were generated from a subset of a Kaggle dataset of fruits and vegetables), where 50 images were randomly selected (25 images of fruits and 25 images of vegetables). Thereafter, the performance of the model was assessed based on various metrics. This included measuring the model's accuracy, precision, recall, F1 score, and attack execution time.

As the researchers lack time and exceptional hardware resources, they decided to limit the number of images used to 50, use only ResNet50 as the classification model, and utilize just torchattacks as the adversarial attack library. These delimitations should be kept in mind as they can affect the results and should be considered for future studies that may be similar to or based on this paper.

3.2 Data Collection

3.2.1 Fruits and Vegetables Image Recognition Dataset

The researchers utilized a dataset from Kaggle that contains images of fruits and vegetables used for training, testing, and validating machine learning models. The images in this dataset were scraped from Bing Image Search (Seth, 2020). The dataset contains 36 image classifications:

a. **Fruits**: banana, apple, pear, grapes, orange, kiwi, watermelon, pomegranate, pineapple, and mango.

b. **Vegetables**: cucumber, carrot, capsicum, onion, potato, lemon, tomato, raddish, beetroot, cabbage, lettuce, spinach, soy bean, cauliflower, bell pepper, chili pepper, turnip, corn, sweetcorn, sweet potato, paprika, jalepeño, ginger, garlic, peas, and eggplant.

The dataset contains three folders:

a. **train** (100 images per classification)

b. **test** (10 images per classification)

c. **validation** (10 images per classification)

3.2.2 Preprocessing

To ensure consistency in the input data, the sampled images were first converted to the RGB color space. This was done if they were not already in the required format, as the model expects three color channels (red, green, and blue). Aside from converting the images, they were also resized to a standard size of 224x224 pixels. This ensured that all images have the same size and dimensions before being used as the basis of the adversarial images. The images were also then converted to tensors (multi-dimensional arrays) as the model processes them in that form. Lastly, the images were normalized using predefined mean and standard deviation values for each color channel. This ensured that the input values (pixel intensities) have a similar data distribution, making the learning process more efficient.

3.2.3 Sampling Method

This study only uses the test folder as it only requires a relatively smaller number of images due to previously discussed limitations. To avoid biases, the researchers use random sampling on the fruit and vegetable image dataset. The study uses 50 out of 359 test images, which consisted of 25 fruit images and 25 vegetable images. These were stored in two folders (fruits and vegetables). Inside these folders are also folders that contain the test images and are named after the actual classification of the images inside them.

3.3 Model

3.3.1 ResNet50

ResNet is a deep convolutional neural network (CNN) architecture that stands for "Residual Network," (Kundu, 2023). ResNet50 is a variant of the said architecture, with "50" in the name referring to the number of layers in the network. The researchers selected this model because it is accessible, easy to use as it is already pre-trained, and uses 1000 classifications, including those that are used by the image dataset from Kaggle. Mathworks (n. d.) mentioned, "You can load a pre-trained version of the neural network trained on more than a million images from the ImageNet database. The trained neural network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the neural network has learned rich feature representations for a wide range of images."

3.4 Adversarial Attacks and Image Generation

In order to choose which attack type to use, the researchers looked at the torchattack repository and searched for the types that were already studied and the most recently cited, specifically in the year 2022. Below are some untampered images from the sampled dataset:

The study used the following adversarial attack types:

a. **Fast Gradient Sign Method (FGSM):** This uses the neural network's gradients to create an adversarial sample. This method uses the gradients of the loss with respect to the image to make a new image that maximizes the loss. Here are some sample FGSM-affected images from the sampled dataset:

b. **Projected Gradient Descent (PGD):** This is an iterative version of FGSM. Considered one of the strongest gradient-based attacks, this process consists of performing multiple FGSM attacks with small step sizes and clipping pixels that get modified beyond an epsilon ball. Here are some sample PGD-affected images from the sampled dataset:

c. **Carlini & Wagner (CW) Attack:** This generates adversarial images by formulating them as optimization problems. It seeks to discover the

smallest perturbation to the image, causing a misclassification by the target model. Here are some sample CW-affected images from the sampled dataset:

d. **Projected Gradient Descent with L2 Norm Constraint (PGD L2):** A variant of PGD that uses the L2 norm constraint on the adversarial perturbation. This generates examples using different Lp norms. Here are some sample PGD L2-affected images from the sampled dataset:

e. **Nesterov Iterative Fast Gradient Sign Method (NIFGSM):** The transferability of adversarial examples is improved by incorporating Nesterov's accelerated gradient into the iterative attacks. It looks ahead and enhances the transferability of examples. Here are some sample NIFGSM-affected images from the sampled dataset:

f. **Pixle:** A black-box attack that can correctly attack a high percentage of dataset samples. This is done by rearranging a small number of pixels within the input image. Here are some sample Pixle-affected images from the sampled dataset:

g. **Scale-Invariant Nesterov Iterative Fast Gradient Sign Method (SINIFGSM):** Leverages the scale-invariant property of deep learning models. Aside from this, it also optimizes the adversarial example through multiple images as its inputs. Here are some sample SINIFGSM-affected images from the sampled dataset:

h. **VMIFGSM:** Enhances the class of iterative gradient-based attack methods and enhances their attack transferability. For every gradient calculation's iteration, it considers the variance of the previous iteration to determine the current gradient. This is opposite to attacks that directly use the current gradient momentum accumulation. Here are some sample VMIFGSM-affected images from the sampled dataset:

i. **VNIFGSM:** Similar to VMIFGSM. Unfortunately, the available information regarding this attack method is limited. Here are some sample VNIFGSM-affected images from the sampled dataset:

These attacks were used to generate adversarial images to be fed into the ResNet50 model. Thereafter, the model classified them while the researchers gathered information regarding their performance.

3.5 Evaluation Metrics

To effectively assess the performance of the ResNet50 model against adversarial attacks, several metrics were calculated. The researchers also considered the type of machine learning task, specifically multi-class classification, in the selection of the following parameters:

a. **Accuracy:** Measures the ratio of correctly classified cases to the total number of objects in the dataset (Evidently AI Team, n.d.). This is given by the following formula:

$$\text{accuracy} = \left(\frac{\text{correct predictions}}{\text{all predictions}} \right) \times 100$$

Accuracy is inversely proportional to the effectiveness of the adversarial attack.

b. **Precision:** In a multi-class classification problem, precision is the percentage of cases that, out of all those that the model predicted to belong to a given class, were properly classified as such. Ultimately, it measures the ability of the model to correctly identify instances of a specific class (Evidently AI Team, n. d.). This is given by the formula:

$$\text{precision}_A = \left(\frac{TP_A}{TP_A + FP_A}\right) \times 100$$

c. **Recall:** In multi-class classification, recall refers to the percentage of a class's instances that the model successfully classified out of all the class's occurrences. Ultimately, recall measures the model's ability to identify all instances of a particular class (Evidently AI Team, n. d.). This is given by the formula:

$$\text{recall}_A = \left(\frac{TP_A}{TP_A + FN_A}\right) \times 100$$

d. **F1-score:** This is a measure of a model's performance, representing the harmonic mean of precision and recall. By giving both precision and recall equal weight, the balance between these two can be indicated. A higher F1 score generally signifies a better balance between the two metrics, denoting better model effectiveness (Baeldung, 2024). This is given by the formula:

$$F1_{ClassA} = \left(\frac{2 \times \text{precision}_A \times \text{recall}_A}{\text{precision}_A + \text{recall}_A}\right) \times 100$$

e. **Attack execution time:** To measure how fast and efficiently adversarial attacks generate images, the attack execution time is measured.

3.6 Procedure

By utilizing various adversarial attack types from the torchattacks library, the researchers were able to evaluate and compare the performance of the classification model ResNet50 using several performance metrics. This

study consisted of two steps: evaluating the model's performance without using torchattacks, and with torchattacks. All of these steps were done using Google Colab, GitHub, and Python. This process was executed on a machine with the following specifications:

- CPU: 11th Generation Intel(R) Core(TM) i5-11300H @3.10 GHz
- Memory: 16 GB
- Storage: 477 GB SSD NVMe Micron MTFDHBA512QFD
- GPU 0: Intel(R) Iris(R) Xe Graphics
- GPU 1: NVIDIA GeForce RTX 3050 Laptop GPU

3.7 Analysis Methods

Various measures of center (mean, median, and mode) and a measure of variation (standard deviation) were used to interpret the results. This gave the researchers a sense of the central tendency and variability of each metric. Aside from calculating the descriptive statistics, the attacks were ranked based on each performance metric to help identify which attacks perform best and worst on average.

4 Results and Discussion

This section evaluated and discussed the results of the study. Results came from testing the ResNet50 model's robustness against images generated using adversarial attacks from the torchattacks library.

4.1 Performance Analysis of Each Adversarial Attack Type

Table 1 shows the results of the experimentation. Each row represents an adversarial attack type (including the case where no attack was made), while each column represents their performance metrics, including accuracy, precision, recall, f1-score, and attack execution time. The results of the experiment concerning each adversarial attack type are listed below:

Table 1: Performance of various adversarial attack types

Attack type	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)
No attack	72.0	48.53	49.16	48.57	0.00
FGSM	14.0	18.18	7.32	9.55	38.37
PGD	52.0	23.21	18.49	20.09	44.32
CW	50.0	23.21	17.98	17.98	73.67
PGD (L2)	52.0	23.21	17.98	17.98	65.92
NIFGSM	54.0	23.81	20.11	21.00	236.56
Pixle	74.0	51.56	53.12	52.08	31.72
SINIFGSM	52.0	23.21	18.49	20.09	229.59
VMIFGSM	52.0	23.21	18.49	20.09	267.68
VNIFGSM	54.0	23.57	19.39	20.68	283.14

a. **No Attack:** As expected, without any adversarial attack, the model performs the best with an accuracy of 72.00%, precision of 48.53%, recall of 49.16%, and F1 score of 48.57%.

b. **Fast Gradient Sign Method (FGSM):** FGSM significantly reduces the model's performance, lowering the accuracy to 14.00%. This shows the effectiveness of FGSM in creating adversarial examples that can mislead the model.

c. **Projected Gradient Descent (PGD):** PGD also effectively reduces the model's performance, with an accuracy of 52.00%. It takes slightly longer than FGSM, indicating a trade-off between attack effectiveness and computational cost.

d. **Carlini & Wagner (CW) Attack:** The CW attack decreases the model's accuracy to 50.00%. It takes more time than both FGSM and PGD, suggesting that it might be a more complex attack method.

e. **Projected Gradient Descent with L2 Norm Constraint (PGD L2):** This variant of PGD has the same effect on the model's performance as the standard PGD and CW attack, but it takes slightly more time.

f. **Nesterov Iterative Fast Gradient Sign Method (NIFGSM):** NIFGSM slightly improves the model's accuracy compared to PGD,

CW, and PGD L2, but it takes significantly more time, indicating a higher computational cost.

g. **Pixle:** Pixle seems to be the least effective adversarial attack in this list, as the model's performance is only slightly reduced compared to the case where no attack was used. However, it takes less time than most other attacks.

h. **Scale-Invariant Nesterov Iterative Fast Gradient Sign Method (SINIFGSM):** SINIFGSM has the same effect on the model's performance as PGD, CW, and PGD L2, but it takes significantly more time.

i. **VMIFGSM:** VMIFGSM has the same effect on the model's performance as PGD, CW, and PGD L2, but it takes even more time than SINIFGSM.

j. **VNIFGSM:** VNIFGSM slightly improves the model's accuracy compared to PGD, CW, PGD L2, SINIFGSM, and VMIFGSM, but it takes the most time among all the attacks.

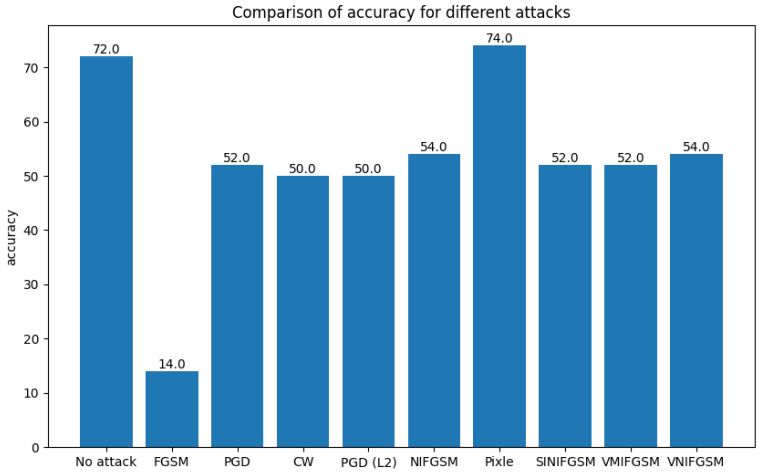
To better visualize this information, the researchers made a bar graph per performance metric. Each graph plots the values obtained for each case of attack.

4.2 Comparative Analysis of Performance Metrics Across Adversarial Attack Types

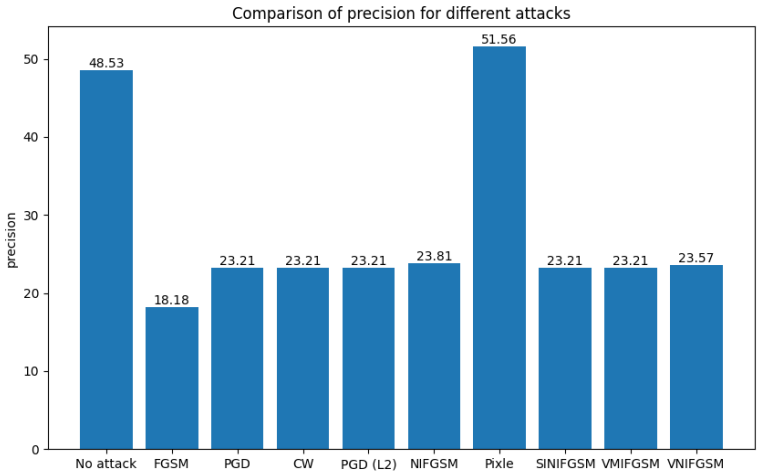
Various measures of center (mean, median, and mode) and a measure of variation (standard deviation) were used to interpret the results. This gave the researchers a sense of the central tendency and variability of each metric. Aside from calculating the descriptive statistics, the attacks were ranked based on each performance metric to help identify which attacks perform best and worst on average.

a. **Accuracy:** The cases where Pixle was used and no adversarial attack was used yielded the highest accuracy, with scores of 74.00% and 72.00%, respectively. This indicates that the model performs best when there is no adversarial attack or when the Pixle attack is used. On the other hand, the FGSM attack results in the lowest accuracy with a

score of 14.00%, suggesting that this attack significantly impairs the model's performance.

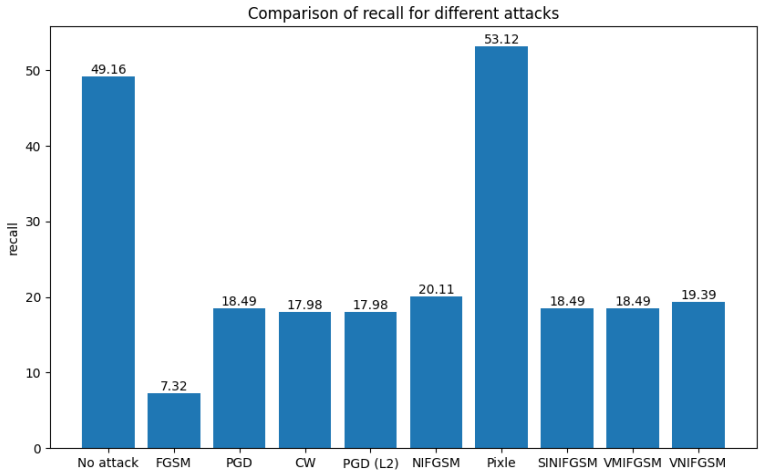


b. **Precision:** Again, the cases where Pixle was used and no adversarial attack was used yielded the highest precision, with scores of 51.56% and 48.53%, respectively. This indicates that the model's positive predictions are most reliable in these scenarios. On the other hand, the FGSM attack results in the lowest precision (18.18%), suggesting that this attack causes the model to make a large number of false positive predictions.

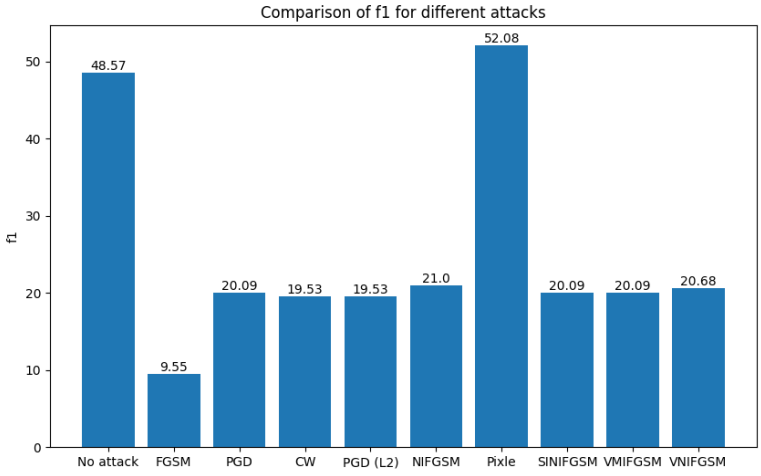


c. **Recall:** The cases where Pixle was used and no adversarial attack was used yielded the highest recall with scores of 49.16% and 53.12%,

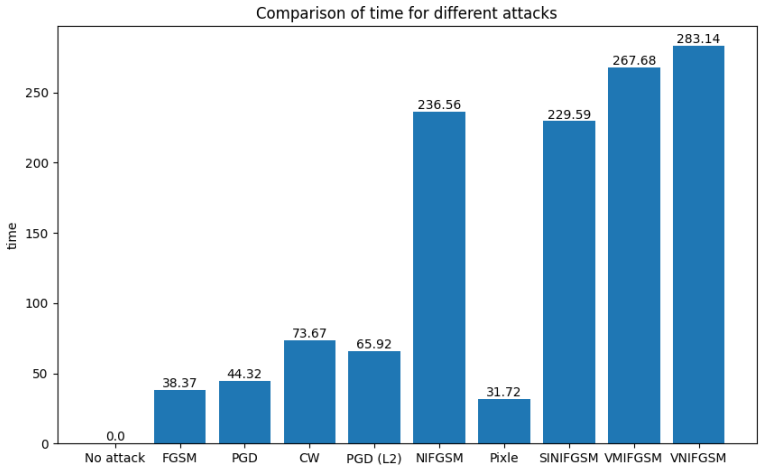
respectively. This indicates that the model is best able to identify all positive instances in these scenarios. On the other hand, the FGSM attack results in the lowest recall (7.32%), suggesting that this attack causes the model to miss a large number of positive instances.



d. **F1 Score:** The cases where Pixle was used and no adversarial attack was used yielded the highest F1 with scores of 52.08% and 48.57%, respectively. This indicates that the model achieves the best balance between precision and recall in these scenarios. On the other hand, the FGSM attack results in the lowest F1 score (9.55%), suggesting that this attack significantly disrupts the balance between precision and recall.



e. **Time:** Pixle takes the least time of 31.72 seconds (excluding the case where no adversarial attack was used), while the VNIFGSM attack takes the most time of attack execution at 283.14 seconds. This suggests that some adversarial attacks may be more computationally intensive than others.



4.3 Evaluation of Central Tendency and Dispersion Measures for Each Performance Metric

Table 2: Performance metrics of various adversarial attack types

Metric	Mean	Median	Mode	Standard Deviation
Accuracy (%)	52.40	52.00	52.000	16.13
Precision (%)	28.17	23.21	23.21	11.66
Recall (%)	24.05	18.49	18.49	14.75
F1-score (%)	25.12	20.09	20.09	13.72
Attack execution time (s)	13.72	69.80	0.00	112.14

The researchers discuss below the central tendency and dispersion measures of the results of the experiment:

a. **Accuracy:** The mean, median, and mode of this metric are almost equal to each other (52.4, 52.0, and 52.0). This could signify that the distribution is nearly perfectly symmetrical and non-skewed (Statsdirect, n. d.). The mean of this metric shows that, on average, the model

classifies approximately half of the adversarial examples correctly. This is 27.22% less than the model's accuracy in the 'no attack' case at 72%. On the other hand, despite having the second highest standard deviation among all of the metrics at 16.13, this indicates a moderate spread of scores.

b. **Precision:** On average, the model's positive predictions are 28.17% correct. This is slightly higher than the median and mode, indicating a slight skew in the data. Furthermore, the precision's mean is 41.95% less than the model's precision in the 'no attack' case at 48.53%. The median and mode being equal at 23.21% suggests that these are the common values across different adversarial attacks. Having a standard deviation of 11.66 indicates a moderate spread of scores.

c. **Recall:** On average, the model correctly identifies 24.05% of all actual positive cases. This is slightly higher than the median and mode, indicating a slight skew in the data. Furthermore, the recall's mean is 51.08% less than the model's recall in the 'no attack' case at 49.16%. The median and mode being equal at 18.49% suggests that these are the common values across different adversarial attacks. Having a standard deviation of 14.75 indicates a moderate spread of scores.

d. **F1-score:** An F1 score is considered perfect at 100%, and worst at 0%. The mean F1 score of 25.12% suggests that the model has low effectiveness in classifying adversarial examples. Furthermore, the F1 score's mean is 48.28% less than the model's F1 score in the 'no attack' case at 48.57%. The median and mode being equal at 20.09% suggests that these are the common values across different adversarial attacks. Having a standard deviation of 13.72 indicates a moderate spread of scores.

e. **Attack execution time:** On average, the model identifies all adversarial images, regardless of whether they are correct or wrong, within 127.10 seconds. This is slightly higher than the median and mode, indicating a slight skew in the data. The standard deviation of 112.14 indicates a high variation in prediction time, which could be due to image complexity or hardware performance.

In summary, the results of the experiment suggest a significant decrease in the model's performance when classifying adversarial images compared to

untampered images. The model can correctly classify these images about 50% of the time (accuracy). It also showed low percentages of correctly identifying positive instances (low recall and precision). The F1 score's mean being low signifies a low balance in the model's precision and recall. In this case, both precision and recall are scored low, implying a decrease in accuracy and robustness.

5 Conclusions

As machine learning models continue to become more prevalent in studying big data, our understanding of adversarial attacks and examples need to keep up. This study provides significant insights into the robustness of the convolutional neural network classification model ResNet50 against various adversarial attacks. The results are already presented and signified a significant performance decrease (accuracy, precision, and recall).

Fast Gradient Sign Method (FGSM) was able to fool the model the most by achieving the lowest accuracy for all adversary attacks. On the other hand, the Pixle attack also displayed fascinating results by being the least effective, even displaying a slight performance improvement compared to when no attack was applied to the model. The study also showed potential for a trade-off between attack effectiveness and computation cost. A few adversarial attacks were relatively time-consuming, e.g. VNIFGSM took more time to create an adversarial example and marginally enhanced the accuracy results slightly compared to some other attacks.

It is stated that, in a wider sense, the research emphasizes the need to understand adversarial attacks in machine learning. In different applications where these models are indispensable, they have to be made resilient enough against future dangers. Also, even more advanced models like ResNet50 can occasionally be quite vulnerable to adversarial attacks.

The FGSM and Pixel Attack displaying varying degrees of effectiveness show that it is incorrect to say there is a one-size-fits-all approach for securing our models. Each attack has its challenges and requires unique countermeasures. Furthermore, this dilemma is compounded by the trade-off between the effect of an attack and computational cost. Some attacks

might be better but could also consume more resources which may not be available in some cases.

In summary, this work is another reflection of the continuing race between improving model performance on the one hand and ensuring model security on the other. Continuous research and development will be crucial as we move ahead to stay well ahead of potential threats and make our machine-learning models trustworthy.

6 Recommendations

Based on the results of the study, the researchers propose to consider several recommendations for future research. The main constraints of this study include the need for more time and limitations in hardware resource adequacy. Hence, allotting more time and optimizing hardware resources should allow for more extensive testing. This includes expanding the dataset, utilizing other adversarial attack libraries, considering different sampling methods, and investigating additional performance metrics. This, in turn, provides a more comprehensive assessment of the model's performance against adversarial attacks.

Furthermore, future studies can also employ multiple models, aside from the ResNet50 model, in order to compare various models' levels of robustness and accuracy against adversarial attacks. Different models are trained on different datasets, resulting in varied classification specializations. Lastly, new research can focus on training the model(s) using the images generated by multiple adversarial attacks. Doing so explores the possibilities of improving the robustness and accuracy of the model(s) against significant perturbations.

Considering these recommendations for future research can allow for possible breakthroughs or advancements in our understanding of the robustness of classification models against adversarial attacks. This could mean better and more resilient models in the future.

References

Baeldung. (2024, March). *F-1 score for multi-class classification*. Baeldung on Computer Science.
Retrieved from https://www.baeldung.com/cs/multi-class-f1-score

Bajaj, A., & Vishwakarma, D. K. (2024). A state-of-the-art review on adversarial machine learning in image classification.
Multimedia Tools and Applications, 83(3), 9351-9416.

Boesch, G. (2023, December). What is adversarial machine learning? Attack methods in 2024. viso.ai.
Retrieved from https://viso.ai/deep-learning/adversarial-machine-learning/

Boesch, G. (2024). *A Complete Guide to Image Classification in 2024.*
Retrieved from https://viso.ai/computer-vision/image-classification/

Brown, T. B., Man´e, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial Patch (Version 2). arXiv.
Retrieved from https://doi.org/10.48550/ARXIV.1712.09665

Buzhinsky, I., Nerinovsky, A., & Tripakis, S. (2021). Metrics and methods for robustness evaluation of neural networks with generative models.
Machine Learning. doi:10.1007/s10994-021-05994-9

Carlini, N., & Wagner, D. (2016). Towards Evaluating the Robustness of Neural Networks. arXiv [Cs.CR].
Retrieved from http://arxiv.org/abs/1608.04644

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., & Mukhopadhyay, D. (2021). A survey on adversarial attacks and defences.
CAAI Transactions on Intelligence Technology, 6(1), 25-45.

Chaudhari, D., & Waghmare, S. (2022, May). Machine vision based fruit classification and grading—a review. In *ICCCE 2021: Proceedings of the 4th International Conference on Communications and Cyber Physical Engineering* (pp. 775-781). Singapore: Springer Nature Singapore.

Duan, R., Ma, X., Wang, Y., Bailey, J., Qin, A., Yang, Y. (2020). Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles. Retrieved from https://arxiv.org/abs/2003.08757

Evidently AI Team. (n.d.). Accuracy, precision, and recall in multi-class classification. *Evidently AI - Open-Source ML Monitoring and Observability.* Retrieved from https://www.evidentlyai.com/classification-metrics/multi-class-metrics

Gupta, M. (2024, February). Calculating Precision & Recall for Multi-Class Classification. *Medium.*
Retrieved from https://medium.com/data-science-in-your-pocket/calculating-precision-recall-for-multi-class-classification-9055931ee229

Haldar, S. (2020, April). Gradient-based adversarial attacks: An introduction. *Medium.*
Retrieved from https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9

Hashemi-Pour, C., & Gillis, A. S. (2023, November). What is adversarial machine learning? *Enterprise AI.*
Retrieved from https://www.techtarget.com/searchenterpriseai/definition/adversarial-machine-learning

Kim, H. (2021). Torchattacks: A PyTorch Repository for Adversarial Attacks. arXiv [Cs.LG].
Retrieved from http://arxiv.org/abs/2010.01950

Kundu, N. (2023, January). Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation. *Medium.*
Retrieved from https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f

Mimma, N., Ahmed, S., Rahman, T., Khan, R. (2022). Fruits Classification and Detection Application Using Deep Learning. *Scientific Programming*, doi: 10.1155/2022/4194874

Mostafid, T. (2023, December). Overview of VGG16, ResNet50, Xcep-

tion and MobileNet Neural Networks. *Medium*.
Retrieved from https://medium.com/@t.mostafid/overview-of-vgg16-xception-mobilenet-and-resnet50-neural-networks-c678e0c0ee85

ResNet-50 convolutional neural network. *MATLAB*.
Retrieved from https://www.mathworks.com/help/deeplearning/ref/resnet50.html

Sanghvi, K. (2020). Image Classification Techniques. *Medium*.
Retrieved from https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3), 160.

Seth, K. (2022, February). Fruits and vegetables image recognition dataset. *Kaggle*.
Retrieved from https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition

Statsdirect. (n.d.). Central tendency. *StatsDirect*. Retrieved from https://www.statsdirect.com/help/basic_descriptive_statistics/central_tendency.htm

SuperAnnotate (2023). What is image classification? Basics you need to know.
Retrieved from https://www.superannotate.com/blog/image-classification-basics

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. arXiv [Cs.CV].
Retrieved from http://arxiv.org/abs/1312.6199

Villegas-Ch, W., Jaramillo-Alcázar, A., & Luján-Mora, S. (2024). Evaluating the Robustness of Deep Learning Models against Adversarial Attacks: An Analysis with FGSM, PGD and CW.
Big Data and Cognitive Computing, 8(1). doi: 10.3390/bdcc8010008

Vishnu, V. K., & Rajput, D. S. (2020). A review on the significance of machine learning for data analysis in big data. *Jordanian Journal of*

Computers and Information Technology, 6(1).