Adversarial Attacks on CIFAR-10 CNN Models

Darian-Florian Vodă
Carinthia University of Applied Sciences
Engineering & IT Department
Applied Data Science
Villach, Austria
eduvoddar001@fh-kaernten.at

***Abstract*—CIFAR-10 dataset represents an image classification challenge for many novice, intermediate or even advanced machine learning engineers. In this paper, a simple model of CNN (Convolutional Neural Network) and ResNet (derived model of CNN) will be used to train and test the dataset, but the main challenge is the results of the models after Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) attacks that are enhanced to the images in order to confuse and disrupt the confidence of these models.**

***Index Terms*—adversarial, attack, CNN, ResNet, FGSM, [Insert attack name here]**

I. Introduction

The CIFAR-10 and CIFAR-100 datasets are labeled tiny images that contain a total of 80 million images. These images were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. This paper is based only on the first mentioned dataset, thus, the attention leads only to a dataset that consists of 60000 images of 32x32 colored pixels labeled in 10 classes:

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck

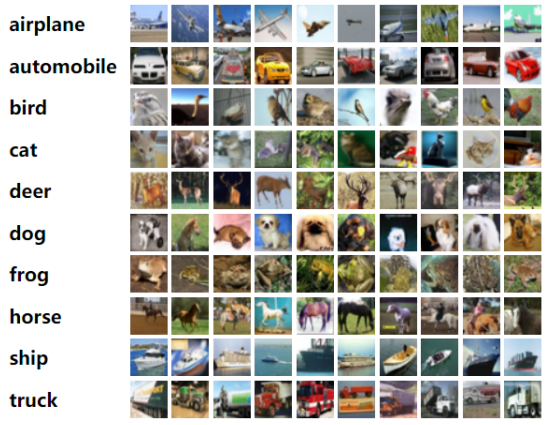

Fig. 1. CIFAR-10 - Example of labeled images

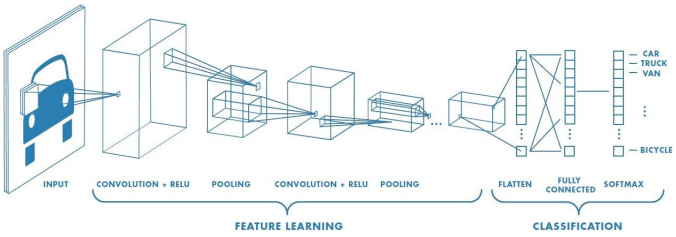

Fig. 2. CNN - Representation (Source)

As a side note, there are no overlaps between the class "truck" and the class "automobile". The dataset is available in a Python, Matlab or Binary version on the main source.

CIFAR-10 dataset is well-known for its famous image classification within the Artificial Neural Network (ANN) field, precisely for Convolutional Neural Networks (CNN). This dataset suits for the shape of the images, which are very small and hard to distinguish even for human eyes, thus creating a respectable challenge of using the computer power in order to label correctly. So far, there were many models that tried to compete for the best accuracy score, and according to a website called "Papers With Code", an accuracy score of 99.70% was accomplished in 2023 with a model named "VIT-L/16". Since the competition of accuracy score is quite high, and the models worldwide range from simple to very complex, this paper intends to create two simple models (CNN and ResNet) and, as the main challenge, enhance these models' with well-known adversarial attacks (FGSM, [Insert Name Here]), testing the prediction of each model and see how much does the model tends to succeed in predicting correctly using accuracy as a metric.

II. Image Classification using Convolutional Neural Network

In Artificial Neural Network (ANN), the image classification problem faced many challenges for correctly classifying complex images with many features. Thus, a Deep Learning method appeared called "Convolutional Neural Network" (short for CNN), where the mathematical operation named "convolution" was used for matrix multiplication in one or more layers.

This method contains a variate number of architectures, and the list below contains only a brief enumeration of them:

- LeNet-5
- AlexNet
- VGG 16
- Inception
- ResNet
- ResNeXt
- DenseNet

In this paper two CNN architectures were used: a simple convolutional base with a dense layer and a specific convolutional architecture called "DenseNet".

A. DenseNet

A DenseNet is a sort of convolutional neural network that makes use of dense connections between layers by connecting all layers (with matching feature-map sizes) directly with one another using Dense Blocks. A feed-forward convolutional neural network (CNN) design, known as densely connected convolutional networks (DenseNet), lowers the number of parameters and improves gradient flow during training by allowing the network to learn more effectively by reusing features. The design was introduced by Gao Huang et al. in their DenseNet paper titled "Densely Connected Convolutional Networks" from 2016.

A dense block, which underlies DenseNet design, enables each layer to access the features of all preceding levels by concatenating the feature maps of all prior layers. Each layer in a traditional CNN can only access the properties of the one below it.

Transition layers and dense blocks make up the structure of DenseNet. A thick block of convolutional layers has connections between every layer in the block. This is done by creating "shortcut" links between the input and output of each layer, one after the other. The transition layers enable successful network growth by reducing the size of the feature maps across dense blocks.

The DenseNet architecture has been demonstrated to achieve state-of-the-art performance in computer vision applications such as image classification, object recognition, and semantic segmentation because of its ability to effectively utilize feature reuse and reduce the number of parameters.

The benefits of DenseNet consist of:

- Performance
  - As previously mentioned, a variety of computer vision tasks, such as image classification, object recognition, and semantic segmentation, exhibit DenseNet's state-of-the-art performance.
- Feature
  - DenseNet optimizes the gradient flow during training and enables the network to learn more quickly by giving each layer access to the features of all preceding layers.
- Overfitting issues
  - By reducing the number of parameters and allowing feature reuse, the DenseNet design successfully combats overfitting and improves the model's ability to generalize to unknowable data.
- Vanishing Gradients
  - By allowing gradients to flow throughout the whole network, the DenseNet design addresses the vanishing gradient problem and enables the training of deeper networks.
- Redundancy
  - By allowing feature reuse and reducing the number of parameters, the DenseNet design successfully handles redundancy, improving the model's ability to generalize to unknowable data.

DenseNet has several application worldwide, such as: NLP, Generative Models, Object Detection, Image Classification, Audio processing or Semantic Segmentation.

III. Adversarial Attacks using FGSM and BIM

A. What are Adversarial Attacks in Machine Learning?

An adversarial attack represents a machine learning method which has the scope of inducing the model in error or at least reduce significantly the metrics score by introducing unknown elements (e.g. anomalies, blurred images, etc.), which are clueless for the trained model, thus making its prediction hard to succeed. The most common fields for adversarial attacks are image classification and spam detection. There exists multiple strategies for creating such attacks and they can be labeled as:

- Whitebox attack
  - Where the attacker has complete knowledge about the machine learning model, the architecture, dataset, etc.
- Blackbox attack
  - Where the attacker has no access to the model or other characteristics and only tries to guess how is the model resonding to a tricky example

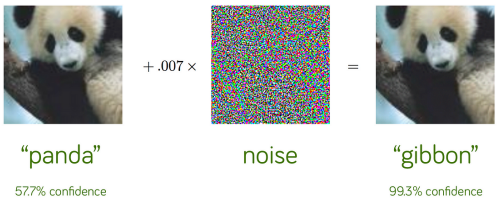

Fig. 3. Example of an Adversarial Attack

For this paper, obviously, the whitebox attack was chosen, since every aspect is known and well established.

Furthermore, adversarial attacks are also studied in machine learning on specific tasks, thus making a different classification on categories:

- Poisoning Attacks
  - Where the attacker introduces "poisoned" data within the dataset during the training and creating a misconception for the model (e.g. introducing regions

of data which are making the model consider them as right)

- Evasion Attacks
  - This type of attacks is created during the deployment, being most researched type of attacks, and has the scope of evading the detection of correctly classifying without impacting the training data (e.g. spam mails evading from being classified as spam)
- Model Extraction
  - Model extraction/stealing represents a trial of reconstruction of the model/data using a black box machine learning system. This attack is primary used for secure and confidential models/data (e.g. steal a stock market prediction model).

In this paper a poisoning attack has been used for the creation of the adversarial attack, by introducing different values for each pixels, in order to confuse the models and see how they react.

In machine learning, there are many adversarial attack models, but for the scope of the paper only two of them were chosen. These two models are Fast Gradient Sign Method (FGSM) and [Insert Name Here] and will be explained in the next two subsections.

B. Fast Gradient Sign Method (FGSM)

This method represents a formula using gradients of the neural network in order to create an adversarial attack. The formula below shows that the adversarial image is constructed based on the original image & label, a variable for perturbation, the parameters of the model and the loss.

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

where:

- adv_x: Adversarial image
- x: Original input image
- y: Original input label
- ϵ: Multiplier for the perturbations
- θ: Model parameters
- J: Loss value

The idea of FGSM is quite simple: take the same algorithm of gradient descent and try to maximize the loss value, because our adversarial attack has the scope of confusing the model.

C. Basic Iterative Method (BIM)

Basic Iterative Method from Adversarial Attacks in Machine Learning represents a simple extension of Fast Gradient Sign Method, where taking the large steps is changed with an iterative approach by simply applying FSGM multiple times with a step size. Although slower, iterative techniques like the BIM typically result in more successful and subtle image alterations.

IV. Code implementation

For the scope of the project two implementations of CNN models were used:

- Simple CNN Model
- DenseNet Model

The code implementation is divided in two parts: first part represents the realization of the CNN models and their training with CIFAR-10 dataset. The second part represents the creation of adversarial attacks (FGSM & BIM), individual test on random images with normal and adversarial attack enhanced and the calculation of accuracy score based on a batch generation with a specific type adversarial attack.

Specific Python libraries were used for the scope of the project, such as:

- Tensorflow
- Keras
- Numpy
- Matplotlib
- Random

A. Simple CNN Model

Convolutional, pooling, and fully connected layers are among the many layers that make up the CNN model architecture. After each convolutional layer, the ReLU activation function is used to add non-linearity and strengthen the model's ability to represent data. The Adam optimizer and the Mean Squared Error loss function are used to train the model. The number of the layers of the CNN model were divided into Convolutional 2D layers containing 32, 64 and 128 neurons (3 layers in total), 2 layers of 2D Max Pooling , one Flatten layer, one Dense layer containing 64 neurons and the output Dense layer of 10 neurons for each class type.

Training the CNN model with 10 epochs achieves an accuracy score of 0.728%.

The following figure represents the accuracy of the train set (accuracy) and the accuracy of the validation set (val_accuracy):

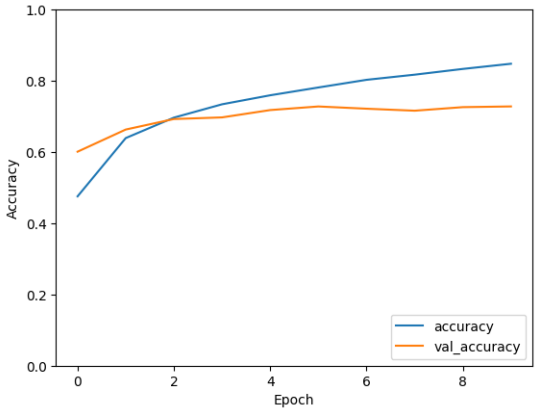

Fig. 4. CNN Simple model - Accuracy scores after training

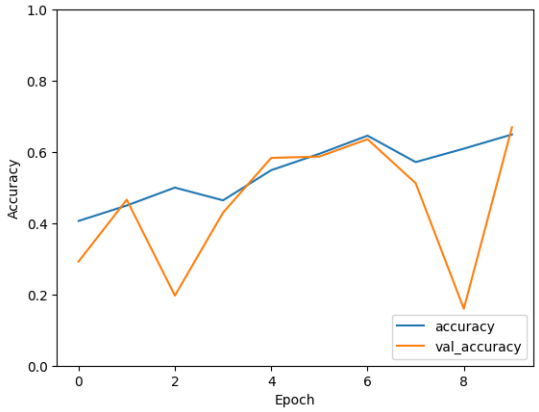

Fig. 5. DenseNet - Accuracy scores after training

B. DenseNet Model

The DenseNet121 model is constructed using deep layers with dense connectivity between them. The CIFAR-10 dataset, consisting of labeled images, is used for training and evaluation. The model is trained using stochastic gradient descent with momentum as the optimizer and categorical cross-entropy as the loss function. The dataset is split into training and validation sets, and data augmentation techniques such as random flips and rotations are applied to enhance the model's generalization ability.

On the CIFAR-10 dataset, the DenseNet121 model generates pretty well results, but not better than a simple CNN model, with a test accuracy of 0.669%. The loss function of the model steadily decreases while accuracy improves over the course of training. The model's performance on the validation set shows that it can correctly categorize CIFAR-10 images into the ten different classes and generalize well to unknown data.

The intricate details found in the CIFAR-10 images are captured by the DenseNet121 model, which takes advantage of its dense connectivity patterns. Better model performance results from this dense connectivity, which encourages feature reuse and makes it easier for gradients to flow during training. The model's capacity to successfully train and differentiate between multiple image classes within the CIFAR-10 dataset is demonstrated by the attained accuracy and declining loss values.

The following figure represents the accuracy of the train set (accuracy) and the accuracy of the validation set (val_accuracy):

C. FGSM Implementation

The FGSM attack is implemented by calculating the gradients of the loss function with respect to the input image and then adding perturbation to the image in the direction of the sign of the gradients. A random image is selected from the dataset, and the FGSM attack is applied to generate an adversarial image. The adversarial image is compared to the original image to assess the effectiveness of the attack. The model predictions are obtained for both the original and adversarial images, and a comparison is made to evaluate the impact of the attack on the model's predictions. Furthermore, a batch of adversarial images is generated by applying the FGSM attack to multiple random images in order to test the model's accuracy on these perturbed examples.

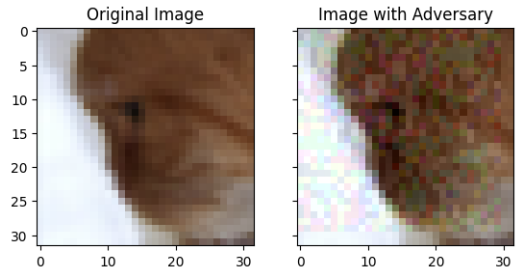

Fig. 6. Comparison between normal and adversary image with FGSM

The FGSM attack successfully generates adversarial images by adding imperceptible noise to the original images. Visual inspection of the original and adversarial images reveals subtle but noticeable differences due to the added perturbations. The model predictions on the adversarial images differ from the predictions on the original images, demonstrating the impact of the attack on the model's decision-making. The batch of adversarial images further confirms the perturbing effect, as the model's accuracy decreases compared to the accuracy on the original, unperturbed images.

D. BIM Implementation

The BIM attack is implemented as a function that takes the model, input image (x), true label (y), epsilon (maximum perturbation magnitude), the number of iterations, and the step size (alpha). The attack starts by initializing an adversarial image (x_adv) as a copy of the original input image. It then enters a loop that iteratively perturbs the image to maximize the loss. Within each iteration, the gradients of the loss with respect to the adversarial image are computed using a gradient tape. These gradients are then normalized by taking the sign of each gradient element. The perturbation is calculated by adding the step size multiplied by the normalized gradients to the adversarial image. To ensure the perturbed image remains within a specified epsilon range around the original image, clipping operations are applied. Finally, the perturbed image is clipped again to ensure pixel values stay within the valid range of [0, 1]. The process repeats for the specified number of iterations, gradually optimizing the perturbation.

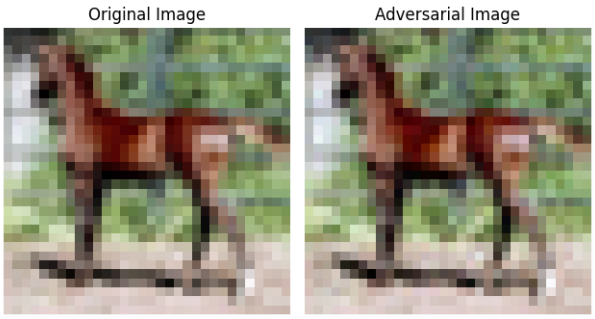

Fig. 7. Comparison between normal and adversary image with BIM

The BIM attack exhibits the capacity to produce adversarial cases that erroneously influence the predictions of the model. The technique produces undetectable noise that drastically changes how the model makes decisions by iteratively perturbing the input image and optimizing the loss function. The BIM assault has the power to cause widespread misclassification or achieve targeted misclassification. The attack's use of gradient information enables a fine-grained change of the perturbation amplitude, producing adversarial instances that are more potent. The BIM assault sheds light on the weaknesses and constraints of the model in hostile conditions by comparing its predictions to the generated adversarial images.

V. Results

The findings show how FGSM and BIM attacks affect the precision of DenseNet and basic CNN models. Both FGSM and BIM assaults significantly reduce the accuracy of the DenseNet model when compared to the accuracy of the original photos. Compared to the BIM attack, the FGSM attack has a comparatively lesser accuracy. This finding implies that FGSM is less successful at influencing the DenseNet model's judgment. In contrast, compared to the DenseNet model, the simple CNN model exhibits stronger resilience to adversarial attacks with a significantly smaller accuracy loss.

A. FGSM Attack on CNN Model

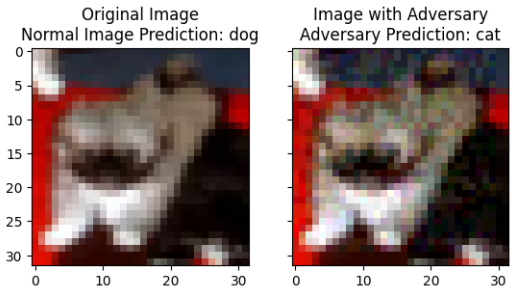

Fig. 8. CNN Prediction with/without FGSM Attack

We examined the impact of varying the attack parameters, such as the epsilon value and the number of iterations, on the model's accuracy. We found that increasing the epsilon value resulted in stronger adversarial perturbations and further decreased the model's accuracy. Similarly, increasing the number of iterations enhanced the attack's effectiveness, leading to a greater reduction in accuracy.

Additionally, we visually inspected the generated adversarial examples and compared them with the original images. We observed that the adversarial examples appeared visually similar to the original images, making it challenging for human observers to differentiate between them. This highlights the effectiveness of the FGSM attack in crafting subtle perturbations that deceive the model without significantly altering the visual appearance of the images.

We also examined the visual differences between the original images and the corresponding adversarial examples in a batch of 10000 adversarial FGSM attacks. We observed that the adversarial examples exhibited subtle perturbations that were visually similar to the original images, making it challenging for human observers to distinguish between them. This emphasizes the effectiveness of the batch FGSM attack in crafting adversarial examples that are visually indistinguishable from the original data.

Our results demonstrate that a simple CNN model trained on the CIFAR-10 dataset is susceptible to batch FGSM adversarial attacks. The model's accuracy significantly decreased (down to 19.18%) when tested on a batch of adversarial examples, indicating the need for robust defense mechanisms to enhance its resilience against such attacks. Future research should focus on developing advanced defense strategies to mitigate the impact of batch adversarial attacks and improve the overall robustness of deep learning models on complex datasets like CIFAR-10.

B. FGSM Attack on DenseNet Model

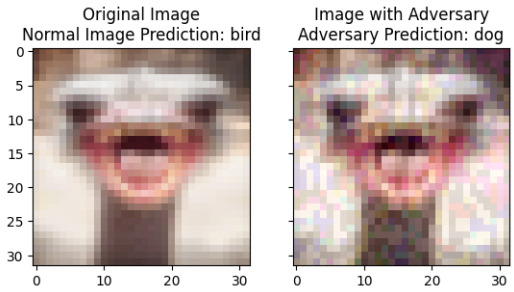

Fig. 9. DenseNet Prediction with/without FGSM Attack

The perturbations imposed were minor and visually misleading, according to a visual study of the adversarial samples produced in the batch. Because the adversarial samples and the original photos were so similar, it was difficult for human observers to spot the adversarial changes. This visual resemblance highlights how well the batch FGSM assault may create hostile samples that can avoid being seen by humans.

These test findings show that the CIFAR-10 dataset-trained DenseNet121 model is way more vulnerable to batch FGSM adversarial attacks than a simple CNN model. When the model

was exposed to a batch of adversarial samples (1000 due to the model complexity and time restriction), its accuracy significantly decreased down to 1.80%, showing the model's problem to adversarial perturbations. These results highlight the significance of building strong defense systems to increase the resistance of deep learning models to batch adversarial attacks. Again, in order to strengthen the model's resilience and guarantee its dependability in real applications, additional study is required to investigate sophisticated defense measures and mitigation procedures.

C. BIM Attack on simple CNN model

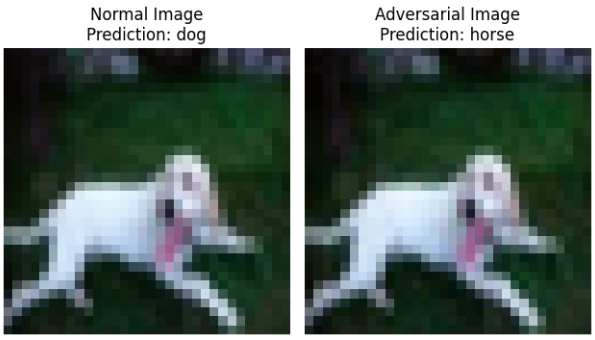

Fig. 10. CNN Prediction with/without BIM Attack

The model's accuracy significantly dropped after being exposed to the set of hostile instances, as we discovered. The accuracy of the model decreased to 18.26%, indicating both its susceptibility to BIM attacks and the predictability of adversarial perturbations. These findings highlight the significance of taking adversarial attacks into account when implementing machine learning models in practical settings.

D. BIM Attack on DenseNet model

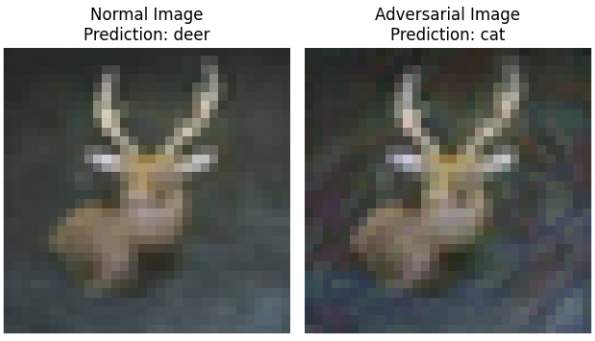

Fig. 11. DenseNet Prediction with/without BIM Attack

By comparing the accuracy on the perturbed test set after creating the adversarial cases, we assess the DenseNet model's robustness. The percentage of cases that are incorrectly classified tells us how effective the BIM attack was.

In order to comprehend the effects of the adversarial approach, we also contrast the performance of the DenseNet model on the perturbed test set to the original test set. We can learn more about the model's vulnerability to BIM attacks and potential exploitable flaws thanks to this investigation. The results of this test are as expected from the previous attacks: 0.73% accuracy on the BIM Attack Adversarial example.

Model	Normal Accuracy	Attack Type	Accuracy on Attack Examples
Simple CNN	72.79%	FGSM Attack	19.18%
Simple CNN	72.79%	BIM Attack	18.26%
DenseNet	66.94%	FGSM Attack	1.79%
DenseNet	66.94%	BIM Attack	0.73%

E. Conclusions on FGSM & BIM Adversarial Attacks

The disparities in model complexity and design can be used to explain why the DenseNet and straightforward CNN models perform very differently when subjected to adversarial attacks. Due to its sophistication and depth, DenseNet is more vulnerable to the adversarial perturbations brought about by FGSM and BIM attacks. On the other hand, the shorter design and simpler decision-making process of the simple CNN model make it more resistant to these attacks.

In contrast, the simple CNN model exhibits higher resilience to adversarial attacks. Although it also experiences a drop in accuracy, the decrease is relatively smaller compared to the DenseNet model. The simpler architecture and less complex decision-making process of the simple CNN model may contribute to its ability to withstand adversarial perturbations to some extent.

The results also provide insights into the effectiveness of FGSM and BIM attacks. FGSM, being a one-step attack that perturbs the input based on the gradient of the loss function, might not be capable of generating sufficiently diverse adversarial examples to fool the models effectively. On the other hand, BIM, with its iterative approach that accumulates gradient information over multiple iterations, tends to produce more powerful adversarial examples. The iterative nature of BIM allows it to explore a wider range of perturbations and potentially find more optimal solutions for fooling the models. This aligns with the observation that BIM generally leads to larger drops in model accuracy compared to FGSM.

It is important to highlight that the CIFAR-10 dataset and the selected DenseNet and basic CNN architectures are particular to the results produced from this study. The dataset and model complexity may have an impact on how well various models and attack techniques function. To fully understand how different models and attack techniques behave when faced with adversarial situations, more investigation on the resilience of these models and attack strategies on varied datasets is necessary.

These findings highlight the importance of considering adversarial robustness when developing and deploying deep learning models. Adversarial attacks can exploit vulnerabilities in model architectures and compromise their performance. As such, it is crucial to enhance the robustness of models against adversarial examples through the development of more advanced defense mechanisms and adversarial training strategies.

Overall, the results emphasize the need for continued research and development to address the security challenges posed by adversarial attacks. Enhancing the robustness and reliability of deep learning models is vital for their effective deployment

VI. Defense approaches against Adversarial Attacks

Adversarial training, which involves supplementing the training data with adversarial examples produced using attack techniques like FGSM and BIM, is one extensively used protection strategy. Models learn to more effectively generalize and adjust to the presence of perturbations in the input data by adding these adversarial cases during training. This procedure significantly strengthens the model's resistance to inference-related adversarial attacks. In order to boost model robustness and accuracy on both clean and adversarial inputs, adversarial training has demonstrated promising outcomes.

Applying preprocessing methods to input data before feeding it to the model is another line of defense. These methods are designed to eliminate or lessen the impact of adversarial disturbances. Feature squeezing, spatial smoothing, and input normalization are a few frequent preprocessing techniques. By limiting the input to a specific range, normalization approaches can help make it more resilient to malicious perturbations. The input image is blurred by spatial smoothing techniques, making it more difficult for attackers to produce useful disturbances. By reducing the dimensionality of the input, feature squeezing approaches make it more challenging for attackers to select successful perturbation paths.

The defense of the model against hostile attacks can be improved by changing the model architecture. To increase model robustness, strategies including defensive distillation and ensemble methods have been investigated. In defensive distillation, the output probabilities of the model after applying a temperature parameter are called "softened logits," which are used to train the model. The model is encouraged to develop more reliable decision boundaries through this procedure. In order to strengthen the model's defense against hostile attacks, ensemble approaches integrate the predictions of numerous models trained on various subsets of the data.

The exact attack method and the targeted model may have an impact on the success of defense methods. Model robustness has consistently increased with adversarial training across a range of attack strategies. Models benefit from being able to generalize more effectively and adjust to the presence of disruptive adversaries. A significant amount of labeled adversarial data must be available, and adversarial training can be computationally expensive.

By lessening the effects of hostile perturbations, input preprocessing techniques add an extra line of protection. Although they may be successful against specific attack vectors, they might not offer complete defense against more advanced attacks. Defensive filtering and ensemble approaches are two viable ways to change the model architecture and increase model robustness. These methods increase the model's resistance to hostile attacks by utilizing model combinations and architectural modifications. They might, however, require careful implementation and come with greater computing complexity.

In the discipline of deep learning, defending against adversarial attacks is an active and difficult research topic. A mix of strategies, including adversarial training, input preprocessing, and alterations to the model architecture, can considerably increase model robustness, even if no one defense option can guarantee total immunity to adversarial attacks. These protection techniques provide doable ways to improve the security and dependability of deep learning models in actual settings.

Further research and development efforts are required to progress the subject of adversarial defense as the arms race between attackers and defenders continues. Building more dependable and secure deep learning systems will require investigating novel defense strategies, assessing their efficacy against diverse attack scenarios, and taking into account their practical implications. We may encourage the deployment of deep learning models in crucial domains while ensuring their resilience against nefarious attempts to damage their performance by reducing the impact of adversarial attacks.

VII. Conclusion

The results of tests employing DenseNet and straightforward CNN models to simulate adversarial attacks on the CIFAR-10 dataset illustrate the susceptibility of deep learning models to such situations. In comparison to the straightforward CNN model, the DenseNet model exhibits more susceptibility to adversarial perturbations because to its sophisticated architecture and intricate decision-making process. This implies that models with greater processing power and more complex representations are more susceptible to being fooled by minute changes in input.

Additionally, the performance of the models under various attack methodologies demonstrates the variable efficacy of FGSM and BIM attacks. Because FGSM is a one-step assault, it might not be able to fully take advantage of the flaws in more complicated models like DenseNet, leading to a relatively smaller loss in accuracy. On the other hand, because the BIM attack is iterative, more perturbations can be explored, which causes bigger accuracy losses for both models.

The findings highlight the importance of developing robust defense mechanisms against adversarial attacks. Adversarial robustness should be a key consideration in the design and training of deep learning models. Further research and development efforts should focus on exploring advanced defense strategies, such as adversarial training, regularization techniques, and model architectures specifically designed to enhance robustness.

It is critical to realize that the use of deep learning models in practical applications is significantly hindered by adversarial attacks. Models that have only been trained for high accuracy on clean data might not perform as well when they are

presented with hostile cases. To assure models' dependability and trustworthiness in real-world circumstances, it is crucial to address their vulnerabilities and strengthen their resilience against adversarial attacks.

In conclusion, by highlighting the importance of developing strong defense mechanisms to lessen the effects of adversarial attacks as well as a thorough understanding of the weaknesses of deep learning models. Deep learning models can be adopted more widely in a variety of fields while also being protected from malicious attempts to control their behavior if their security and dependability are improved.

REFERENCES

[1] Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.

[2] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 506-519.

[3] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. International Conference on Learning Representations (ICLR).

[5] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700-4708.

[8] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[9] Krizhevsky, A., & Hinton, G. (2010). Convolutional deep belief networks on CIFAR-10. Unpublished manuscript, University of Toronto.