




RESEARCH

Adversarial Attacks on Visual Objects Using the Fast Gradient Sign Method

Syed Muhammad Ali Naqvi ·
Mohammad Shabaz ·
Muhammad Attique Khan · Syeda Iqra Hassan 

Received: 30 April 2023 / Accepted: 27 August 2023 / Published online: 22 September 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract Adversarial attacks exploit vulnerabilities or weaknesses in the model's decision-making process to generate inputs that appear benign to humans but can lead to incorrect or unintended outputs from the model. Neural networks (NNs) are widely used for aerial detection, and increased usage has highlighted the vulnerability of DNNs to adversarial cases intentionally designed to mislead them. The majority of adversarial attacks now in use can only rarely deceive a black-box model. We employ the fast gradient sign technique (FGSM) to immediately enhance the position of an adversarial area to identify the target. We

employ two open datasets in extensive experiments; however, the findings demonstrate that, on average, only 400 queries may successfully perturb at least one erroneous class in most of the photos in the test dataset. The proposed method can be used for both untargeted and targeted attacks, leading to incredible query efficiency in both scenarios. The experiment manipulates input images using gradients or noise to generate misclassified outputs. It is implemented in Python using the TensorFlow framework. The experiment optimizes performance by using an algorithm with an initial learning rate of 0.1 and adjusting the learning rate based on the number of training samples using different epoch values. Compared to other studies, our technique outperforms them in crafting adversaries and provides high accuracy. Moreover, this technique works effectively, needs a few lines of code to be implemented, and functions as a solid base for upcoming black-box attacks.

S.M.A Naqvi (✉)
School of Computer Science and Engineering, Central
South University, Lushan Road, Changsha 410083, Hunan,
China
e-mail: muhammadali1019@yahoo.com

M. Shabaz
Model Institute of Engineering and Technology, Jammu
180001, J&K, India
e-mail: bhatsab4@gmail.com

M. Attique
Department of Computer Science, HITEC University,
Taxila 47080, Punjab, Pakistan
e-mail: attique.khan@ieee.org

S.I Hassan
Department of Electronics and Electrical Engineering,
British Malaysian Institute, University of Kuala Lumpur,
53100 Gombak, Malaysia

Ziauddin University, 74600 Karachi, Pakistan
e-mail: syeda.iqra@s.unikl.edu.my
e-mail: iqra.hassan@zu.edu.ok

Keywords FGSM · Adversarial attack · Visual objects · Deep neural network

1 Introduction

Machine learning and deep neural networks are widely used in various applications due to their excellent performance. However, despite being designed to be robust, they are still susceptible to adversarial attacks. Adversarial attacks refer to the deliberate manipulation of input data to mislead a machine learning model. These attacks are designed to be imperceptible to the

human eye, but can cause the model to incorrectly classify the input with high confidence

Adversarial attacks can be a major problem in applications like image or speech recognition, as well as in autonomous driving systems. Recent studies have shown that deep neural networks are vulnerable to these attacks, where even tiny, imperceptible changes can cause them to make incorrect predictions. However, despite the success of deep learning-based object-tracking algorithms, it's important to evaluate how effective and robust they really are. [1,2].

Adversarial attacks against deep learning models in computer vision have become a topic of growing interest in recent years. Several successful adversarial attacks against deep networks have been developed, which can effectively mislead image classifiers and object detectors. One notable example is the research by Szegedy et al. [3] which demonstrated that even small perturbations in images that are nearly imperceptible to humans can result in deep learning models misclassifying the image. The addition of adversarial perturbations on the target patch of a free model in a specific frame can lead to the loss of the target in subsequent frames for state-of-the-art trackers. This malicious behavior poses a serious threat to surveillance systems, highlighting the need for further research on adversarial attacks in visual object tracking algorithms. Such research can aid in the development of preventive measures to mitigate the potential risks posed by adversarial attacks [4]. The data's factors impact the judgment decision of the deep neural network. As a result, deep neural networks can be attacked by data modification. Figure 1 shows how making little changes to the original image makes them undetectable to the human eye, but when added to the neural network model, they have a significant impact on how well the model recognizes objects.

The fast gradient sign method (FGSM) is a popular technique for creating adversarial examples that can fool machine learning models. It works well against white-box attacks where the attacker has full knowledge of the model's architecture and parameters. However, it's difficult to use FGSM for black-box attacks where the attacker only has access to the input and output of the model. Researchers are still studying FGSM's effectiveness against black-box attacks and potential defenses against them.

The FGSM's goal in altering speech is to exploit vulnerabilities in speech recognition systems and high-

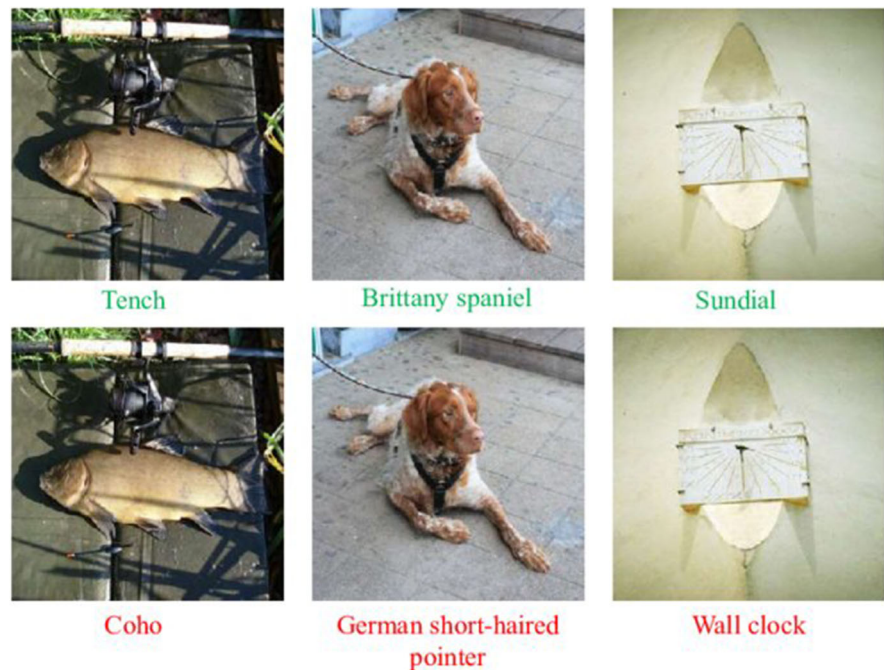
light their susceptibility to adversarial attacks. By generating adversarial examples, researchers can assess the robustness and vulnerability of speech recognition models and develop defense mechanisms to mitigate the impact of such attacks. The ultimate objective is to improve the reliability and security of speech recognition systems in real-world applications. The Fast Gradient Sign Method (FGSM) is not typically used to enhance an adversarial area, but rather to generate adversarial examples to exploit vulnerabilities in machine learning models. The FGSM is a simple and efficient technique used to craft adversarial examples by adding imperceptible perturbations to the input data. It leverages the gradients of the model's loss function with respect to the input to determine the direction and magnitude of the perturbations.

Adversarial attack methods often struggle to effectively target black-box models, especially those that have robust defense mechanisms like ensemble adversarial training. This is due to the trade-off between attack ability and transferability, where optimization-based and iterative methods generate adversarial examples that have poor transferability to other models, making black-box attacks less effective. One proposed solution by Papernot et al. [1] is to use adaptive queries to train a surrogate model and fully understand the behavior of the target model, turning black-box attacks into white-box attacks. However, this approach requires a large number of queries and the full prediction confidences of the target model, making it impractical for large datasets like ImageNet. To address this challenge, research is needed on how to effectively attack black-box models without knowledge of their architecture or parameters, and without querying them excessively.

In this paper, we study adversarial attacks against visual object tracking. This research conducts the attacks in a "black box" environment, where the only probabilities that are known are those projected by the target model. The task is defined as a constraint-based optimization problem. The goal is to determine the best attribute adjustments that will cause classifiers to predict incorrect labels.

Along with the speedy advancement of deep learning (DL) and artificial intelligence (AI) approaches, it is essential to guarantee the security and resilience of the implemented algorithms. The security weakness of DL algorithms to hostile samples has recently gained widespread recognition. The fake samples, however, appear harmless to humans and might cause numer-

Fig. 1 The bottom row of the image shows the adversarial examples and predicted labels generated by NP-Attack-R with a constraint of $L=0.05$, while the top row displays the original photos and actual labels[5]



ous errors in DL models. Adversarial attacks are successfully used in real-world situations to further illustrate their applicability. As a result, in the current age, the examination of combative attack and protection approaches gained a lot of attention from both the machine learning and security communities.

To achieve severe damage achievement in the black-box situation, we use the fast gradient sign method (FGSM) to instantaneously enhance the situation and distress for a combative patch. Two open datasets are used in extensive experiments. The findings demonstrate that, on average, only 400 queries successfully perturb at least one erroneous class in most of the photos confidencet dataset. This procedure is difficult to implement and creates an adverse strike, but our contribution is to implement the full FGSM function in just a few lines of code.

The majority of adversarial attacks struggle to deceive a black-box model, which means that these attacks have limited success when the attacker has only partial knowledge or no access to the internal workings of the model. Black-box models are typically more challenging to manipulate due to their restricted visibility to the attacker. As a result, crafting effective adversarial examples that can consistently fool black-box models is a difficult task. This highlights the complexity and limitations associated with conducting suc-

cessful attacks in scenarios where the attacker has limited knowledge or access to the target model. Some of the commonly known adversarial attack methods that struggle to fool black-box models include many adversarial attacks rely on the transferability property, where an adversarial example crafted for one model can also deceive another model. However, in the case of black-box models, transferability is often reduced, making it difficult for these attacks to achieve the desired misclassification. Adversarial attacks that rely on querying the black-box model to obtain its responses can face limitations due to the limited number of queries allowed. The attacker's access to the model's input-output interface is restricted, and this lack of knowledge poses challenges in generating effective adversarial examples. Zeroth-order optimization attacks, which aim to optimize the adversarial perturbations without access to the model's gradients or internal information, often struggle to achieve significant misclassifications in black-box scenarios. The lack of gradient information makes it challenging to efficiently manipulate the model's decision boundaries. Some adversarial attacks are designed specifically for certain models or architectures, leveraging their vulnerabilities. These model-specific attacks may not generalize well to black-box models, which have different architectures and decision boundaries, limiting their success.

One type of malicious behavior that poses a serious threat to surveillance systems is the manipulation or tampering of surveillance data. This can involve various actions intended to deceive or manipulate the system's perception of events, leading to inaccurate or misleading information.

The organization of the article begins with the conceptualization of the research in Section 1. Section 2 provides a state-of-the-artwork, which highlights recent related articles. Considering the related work, the methodology is provided along with the experimental setups and their observations in Section 3. The outcomes of the research are further discussed in Section 4 in detail. The final words are concluded in the last section which leads toward the future direction of this research.

Adversarial machine learning is a method of artificial intelligence that aims to trick machine learning models by feeding them fake data [6]. It, therefore, covers both the construction of hostile samples and inputs intended to mislead classifiers as well as their detection. Numerous areas, including spam detection and picture classification, have seen extensive research on these so-called adversarial machine learning approaches [7]. White box attacks differ from black box attacks in that the attacker has access to the model's parameters rather than using the target model while creating adversarial images in the hopes that they will transfer to it.

There has been a lot of focus on this issue since the landmark publication by Szegedy et al. [3], which demonstrated that the most cutting-edge neural networks are susceptible to adversarial attacks. The research has resulted in investigations into several adversarial threat models and situations [8], as well as attacks that are both computationally and perturbational efficient [9], etc.

A white box attack in adversarial machine learning occurs when we are entirely concerned with the implemented model, including its inputs, structural design, and particular internals like weights or coefficient values. [10, 11] A procedure for generating unfavorable instances is an adversarial attack. An adversarial example is one that, despite appearing to be true to a human, is designed to lead a machine learning model to predict inaccurately [12].

It's important to note that the field of adversarial attacks is evolving, and new vulnerabilities are discovered over time. Here are a few examples of neural networks that have demonstrated susceptibility to adver-

sarial attacks: Convolutional Neural Networks (CNNs): CNNs, which are widely used for image classification tasks, have been shown to be vulnerable to adversarial attacks. Techniques like the Fast Gradient Sign Method (FGSM) and its variants can generate adversarial examples that can fool CNNs into misclassifying images. Recurrent Neural Networks (RNNs): RNNs, commonly used for sequence-based tasks like natural language processing and speech recognition, can also be susceptible to adversarial attacks. Adversarial examples crafted to exploit vulnerabilities in the learning process or input representations can lead to incorrect predictions or misinterpretations of sequences. Generative Adversarial Networks (GANs): GANs, which consist of a generator and a discriminator network, have been shown to be vulnerable to adversarial attacks. Attacks on GANs can involve manipulating the generator to produce fake samples that can deceive the discriminator or perturbing the input to generate adversarial samples. Transformers: Transformers have gained significant attention in natural language processing and have demonstrated impressive performance. However, recent research has shown that they are also susceptible to adversarial attacks. Adversarial examples crafted to exploit weaknesses in attention mechanisms or input representations can lead to erroneous predictions or incorrect language understanding. It's worth noting that the susceptibility of neural networks to adversarial attacks depends on various factors, including the specific architecture, the training process, and the attack methodology. Researchers are actively working on developing robust architectures, training techniques, and defenses to mitigate the vulnerability of neural networks to adversarial attacks.

There has been an increase in research in explainable ML in latest years in response to the major difficulties posed by black-box models [13–18]. By reducing the temporal fluctuation of speech components, modulation spectrum smoothing accomplishes the goal of altering speech [19–22]. Instead of making black-box models automatically interpretable, most of this research is focused on developing clear justifications for model operations and descriptions of its individual predictions [23–27].

A combination of factors such as vulnerability of DNNs, limited success of black-box attacks, the effectiveness of the FGSM technique, query efficiency, outperforming existing studies, and ease of implementa-

tion, which contribute to the novelty of the approach discussed.

Highlighting vulnerability of DNNs: The vulnerability of Deep Neural Networks (DNNs) used for aerial detection to adversarial attacks intentionally designed to mislead them. This research highlights the need for addressing the security concerns associated with these networks. **Fast Gradient Sign Technique (FGSM):** The research introduces the FGSM as a technique employed to enhance the position for an adversarial area and identify the target. This technique is widely used in crafting adversarial examples and is known for its efficiency and effectiveness. **Query efficiency:** The research proposed the method achieves high query efficiency in both untargeted and targeted attacks. This indicates that the technique requires fewer queries to successfully perturb the target model, making it more efficient compared to other approaches. **Outperforming existing studies:** The research claims that the proposed technique outperforms other studies in crafting adversarial examples and provides high accuracy. This suggests that the technique offers advancements and improvements over existing methods in generating effective adversarial attacks. **Ease of implementation:** The statement emphasizes that the proposed technique is easy to implement, requiring only a few lines of code. This indicates that the technique offers a practical and accessible solution for crafting adversarial attacks.

The unique factor is our novelty which is discussed as:

- A combination of factors such as vulnerability of DNNs, limited success of black-box attacks, the effectiveness of the FGSM technique, query efficiency, outperforming existing studies, and ease of implementation, which contribute to the novelty of the approach discussed.
- **Highlighting vulnerability of DNNs:** The vulnerability of Deep Neural Networks (DNNs) used for aerial detection to adversarial attacks intentionally designed to mislead them. This research highlights the need for addressing the security concerns associated with these networks.
- **Fast Gradient Sign Technique (FGSM):** The research introduces the FGSM as a technique employed to enhance the position for an adversarial area and identify the target. This technique is widely used in crafting adversarial examples and is known for its efficiency and effectiveness.

- **Query efficiency:** The research proposed the method achieves high query efficiency in both untargeted and targeted attacks. This indicates that the technique requires fewer queries to successfully perturb the target model, making it more efficient compared to other approaches.
- **Outperforming existing studies:** The research claims that the proposed technique outperforms other studies in crafting adversarial examples and provides high accuracy. This suggests that the technique offers advancements and improvements over existing methods in generating effective adversarial attacks.
- **Ease of implementation:** The statement emphasizes that the proposed technique is easy to implement, requiring only a few lines of code. This indicates that the technique offers a practical and accessible solution for crafting adversarial attacks.

In this study, the main focus is on black-box attacks, where it is assumed that the attacker has no knowledge of the network's architecture and can only use it as an oracle [5,28]. The attacker can create fake images that can fool a machine learning model. To achieve this by training a new model based on the original model's output labels obtained through a process. The new model is then used to generate adversarial images that can deceive the original model [29–31]. The transferability property between the original and replacement networks must be held for the attack to be successful [32–35].

The research contributions are the following:

- Regularization of FGSM techniques that helps the existing algorithms to overcome catastrophic overfitting problems. Although adversarial resilience is a desired quality that raises the reliability of machine learning models.
- Demonstrate how to create adversarial images for modern deep convolutional neural networks (CNNs) without any knowledge of the network's architecture or parameters.
- We conducted thorough experiments to compare our method with existing techniques and found that: 1) our approach generates adversarial perturbations more accurately and efficiently than other methods, and 2) incorporating adversarial examples into the training data significantly enhances the model's resilience against adversarial attacks.

2 Related Work

2.1 Black-box Attacks vs. White-box Attacks

In a black-box attack scenario, we have limited or no knowledge about the targeted model's internal architecture, parameters, or gradients and can only query the model and observe its outputs. The goal is to generate adversarial examples that can deceive the model without exploiting its internal information. In a white-box attack scenario, we have complete knowledge of the targeted model, including its architecture, parameters, and gradients, and can directly access and manipulate the model's internals to generate adversarial examples.

2.2 Targeted vs. Untargeted Adversarial Attacks

In targeted attacks, the aim is to generate adversarial examples that force the model to classify them into specific target classes chosen. The goal is to achieve a specific misclassification. In untargeted attacks, the objective is to generate adversarial examples that cause misclassification without targeting any specific class. The focus is on inducing any form of misclassification, regardless of the particular class it is misclassified as.

2.3 One-shot vs. Iterative Adversarial Attacks

In a one-shot attack, a single perturbation or modification is applied to the input data to create an adversarial example, generated once, and tested against the model. In iterative attacks, the adversarial example generation process involves multiple iterations and applies small perturbations repeatedly to the input, updating the perturbations in each iteration to gradually optimize the adversarial example. Iterative attacks often yield more effective and stronger adversarial examples compared to one-shot attacks.

2.4 Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a simple and fast adversarial attack technique. It uses the gradients of the model's loss function with respect to the input data to generate adversarial examples. FGSM perturbs the input data by taking a small step in the direction of

the sign of the gradients. It is a one-shot attack and is particularly effective against models without adversarial training or specific defenses.

2.5 Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) is an iterative variant of the FGSM attack. It applies multiple iterations of FGSM with small step sizes, gradually updating the perturbations in each iteration. PGD is more powerful and robust than FGSM and can overcome certain defenses, such as gradient masking or input preprocessing techniques. It is widely used to evaluate the robustness of models against adversarial attacks and is often employed as a benchmark attack method.

Adversarial attacks have been extensively studied in the field of deep learning, with researchers continuously investigating new techniques and defenses to enhance their understanding of model vulnerabilities and improve their robustness. In [36], the image space of legit and adversarial images is compared, along with the predicted class, to check whether the image has been potentially perturbed. The obtained results are useful with a 95% accuracy in detecting adversaries with different levels of perturbations, but only raw perturbations are studied and only the PGD attack method is tested. Using the internal parameters and layer activations to detect adversarial examples [37] provides an interesting study that achieves a certifiable detection rate under certain restrictions in the L_∞ norm. Five different common attacks are tested although the method is able to achieve good results only on the MNIST dataset. In [38], an autoencoder is built for detection, where the reconstruction of the candidate image determines the probability of an adversarial origin. Moreover, the approach is reused to build an adversarial, robust network with up to 76% accuracy for the Sparse attack. Finally, a comparable approach to ours is given by [39], where the image data is studied to classify an image as potentially adversarial. In this case, through feature alignment with an external model, the method is able to detect the adversarial image and fix the classifier decision. However, the employed dataset is a customized version of PASCAL VOC, ImageNet, and scraped images from the internet, which results in a difficult framework for comparison. However, the studied attacks do not include the latest and most powerful proposals in the state of the art. Table 1 shows a

Table 1 Comparison of different studies in the field

Method	Datasets	Attacks	Approach
Yin et al., 2019 [36]	MNIST, CIFAR10	PGD	Image space
Shumailov et al., 2020 [37]	MNIST	FGSM, PGD, BIM, CW	Layer activations
Vacanti et al., 2020 [38]	MNIST, CIFAR10	CW, SPARSE, FGSM	Autoencoder
Vacanti et al., 2020 [39]	ImageNet	PGD, MI-FGSM	Feature alignment

comparison of their approaches, studied datasets, and tested attacks or methods.

3 Methodology

Open datasets play a crucial role in conducting extensive experiments, as highlighted in [40]. These datasets are specifically related to misclassified dog breeds, providing valuable insights into this common issue. To facilitate the experimentation process, the data is divided into two sets: training and testing samples. This division is based on an 80:20 ratio, as demonstrated in Table 2. By using these datasets, can explore and develop effective methods for addressing misclassification of dog breeds.

The Fast Gradient Sign Method (FGSM) is a popular and widely used attack method in adversarial machine learning research. There are several reasons for choosing FGSM in this research. FGSM is relatively simple to implement compared to more complex attack methods like PGD or CW. It involves just one step of perturbation based on the sign of the gradient, making it computationally efficient. Due to its simplicity, FGSM attacks can be performed quickly, which can be advantageous when evaluating the robustness of machine learning models against adversarial examples or when conducting large-scale experiments. FGSM is often used as a baseline attack method to assess the vulnerability of a machine learning model. It provides a starting point for evaluating model robustness and

comparing it against more sophisticated attacks. FGSM leverages the gradient information of the model, perturbing the input in the direction that maximizes the loss and induces misclassification. This intuitive approach helps researchers understand the vulnerability of models to small input perturbations.

3.1 Normalization

In the process of sending an image to the network, it is usually normalized. The dimensions of the normalized image remain the same as those of the original image, the coordinate values are altered. Since normalization methods are commonly used and standardized, we assume that the adversary is capable of performing these operations.[30,41].

3.2 Model Architecture

The FGSM technique used in this research works by taking the gradients of the loss function with respect to the input data and using the sign of those gradients to perturb the input data. By adding or subtracting a small fraction of the sign of the gradients from the original input, FGSM generates adversarial examples that can cause the targeted model to produce incorrect outputs. Select the MobileNetV2 model as our neural network architecture and use MobileNetV2 as a baseline model to compare its vulnerability to adversarial attacks with other more complex and computationally intensive models. While MobileNetV2 has been widely used for image classification tasks, exploring its vulnerability to adversarial attacks using FGSM can reveal insights into the robustness of the mode and potential security vulnerabilities. The MobileNetV2 model provides a practical and realistic context for investigating adversarial attacks in image classification in a black box environment as well as in a white box. One of the

Table 2 Data set

Data Set	Training	Testing
Bearcat, cat bear	80%	20%
Eskimo dog	80%	20%
Husky	80%	20%

key features of MobileNetV2 is its use of depth-wise separable convolutions, which separate the spatial and channel-wise convolutions. This reduces the number of parameters and computations required, resulting in a more lightweight model. MobileNetV2 also incorporates linear bottlenecks and inverted residuals, which help improve the efficiency and accuracy of the network.

3.3 Adversarial Image Generation

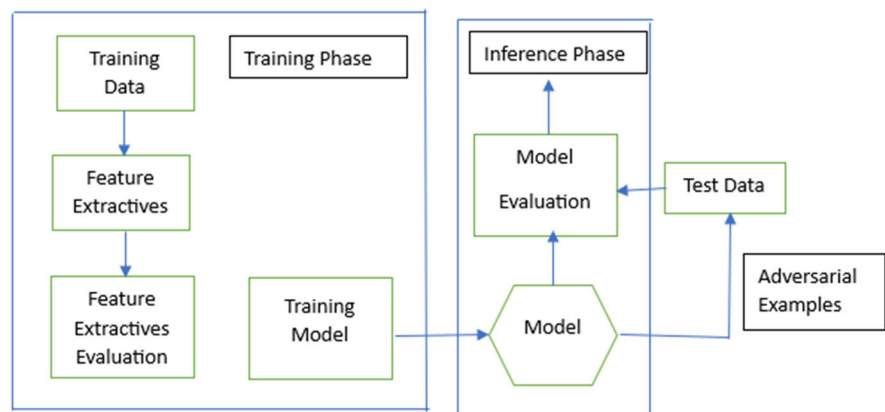
Designing adversarial attacks involves crafting specifically designed inputs to fool or manipulate machine learning models. Figure 2 shows the design flow of an adversarial attack. These attacks exploit vulnerabilities in the model's decision-making process to induce misclassification or produce undesirable outputs. Adversarial attacks can be classified into various types, including evasion attacks and poisoning attacks. Here's a general process for designing adversarial attacks:

- **Define the Attack Objective:** Determine the goal of your attack. Do you want to cause misclassification, manipulate the output, or achieve some other outcome?
- **Select a target model:** Identify the machine learning model you want to attack. This could be an image classifier, a natural language processing model, or any other type of model.
- **Gather Information:** Obtain information about the target model, such as its architecture, training data, and any available documentation. This information will help you understand the model's weaknesses and potential vulnerabilities.
- **Choose an Attack Method “Gradient-Based Attacks”:** These attacks involve perturbing the input data based on the gradients of the target model. Examples include the Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), and Projected Gradient Descent (PGD).
- **Generate Adversarial Examples:** Using the chosen attack method, create adversarial examples by perturbing the input data. This could involve modifying pixels in images, adding or altering words in the text, or manipulating other input features.
- **Test and Evaluate Adversarial Examples:** Apply the generated adversarial examples to the target model and observe the model's response. Evaluate the success of the attack by measuring the misclassification rate or the extent to which the output matches the attack objective.

A subtle modification of an original image that renders the alterations virtually invisible to the human eye constitutes an adversarial attack. When presented to a classifier [31,42,43], the modified image known as an adversarial image is incorrectly classified while the original image is correctly classified [32,44].

Adversarial attacks that make use of gradient-based strategies have been the most successful. These attacks involve modifying an image in the direction of the gradient of the loss function with respect to the input image. There are two main methods for carrying out such attacks: one-shot attacks, where the attacker moves the image one step in the gradient direction, and iterative attacks, where multiple steps are taken. In this study, the MobileNetV2 model pre-trained on ImageNet was used.

Fig. 2 Design flow of an Adversarial Attack



The classification of photos, where making minute and frequently undetectable changes to the images can mislead deep classifiers. Given that deep learning has evolved into one of the cornerstones of many applications that deal with sensitive security issues, such as text-based spam detection, it naturally raises questions about how resilient deep learning systems are.

3.4 Workflow

One interesting aspect of this case is that the gradients used for adversarial image generation are calculated based on the input image. The aim is to create an image that minimizes the loss function, and one way to achieve this is to determine the contribution of each pixel to the loss and apply a perturbation accordingly. This approach is efficient because it involves straight-forward calculations of the gradients and the pixel-wise contributions to the loss. It also leaves the model parameters unchanged, as the model is not being retrained. The algorithm's flow is illustrated in Fig. 3.

3.5 Designing CNN Model

The design mode is shown in Fig. 4. Choose the architecture of your CNN. This typically involves deciding

the number and types of layers, such as convolutional layers, pooling layers, and fully connected layers. Consider the following:

- **Input Layer:** Specify the input shape based on the dimensions of your input data (e.g., image width, height, and number of channels).
 - **Convolutional Layers:** Decide the number of convolutional layers and the size and number of filters in each layer. You may also choose the stride, padding, and activation function for each convolutional layer.
 - **Pooling Layers:** Determine the type of pooling (e.g., max pooling or average pooling) and the pooling size.
 - **Fully Connected Layers:** Decide on the number and size of fully connected layers, also known as dense layers, which often appear towards the end of the network architecture.
5. **Output Layer:** Define the number of output units based on the number of classes in your classification task or the desired output shape for other tasks.

Configure the loss function, optimizer, and evaluation metrics for training the model. The choice of loss function depends on your task (e.g., categorical cross-entropy for classification). Select an optimizer, such as Adam or stochastic gradient descent (SGD), and define

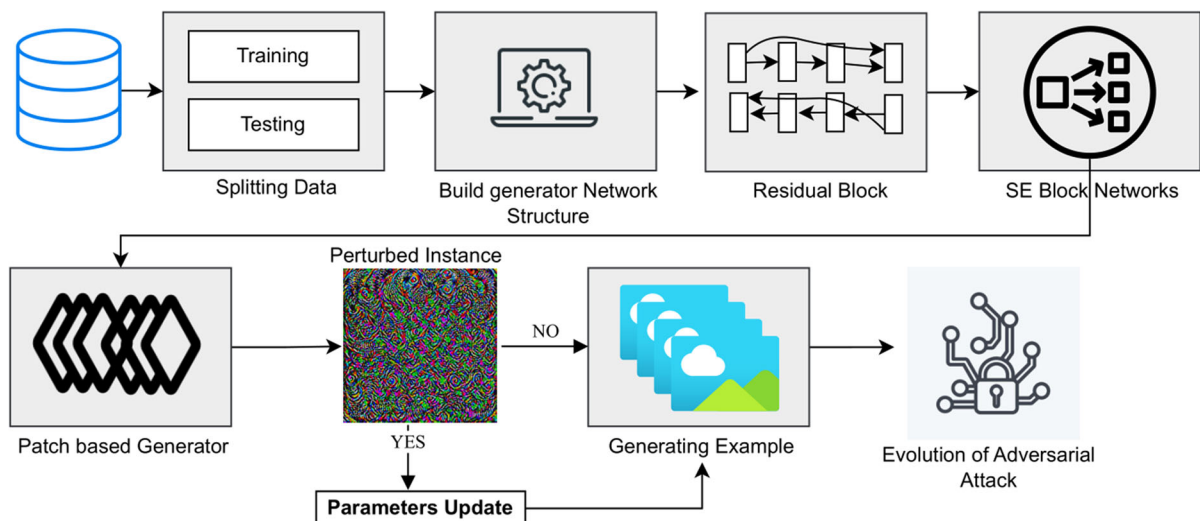
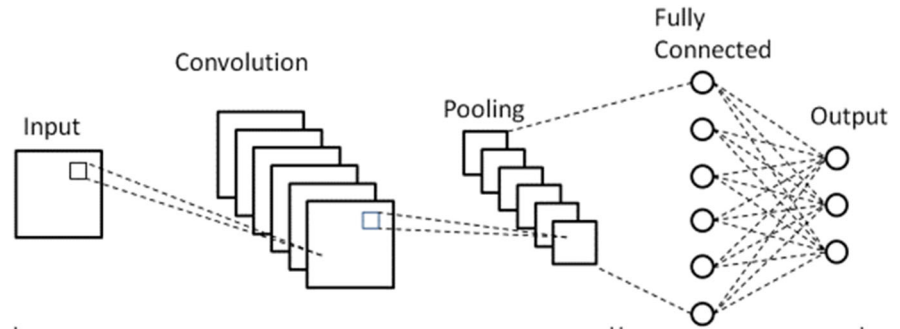
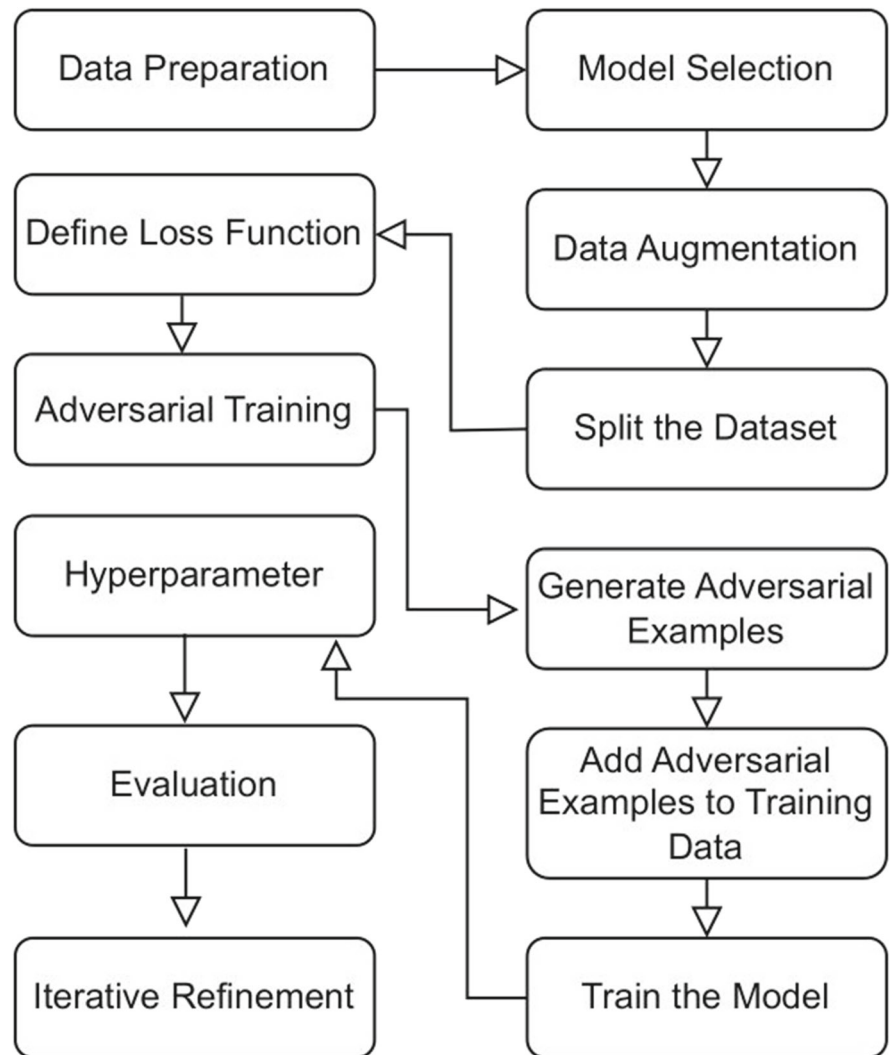


Fig. 3 Flow chart of Implementing Algorithm: Start with the dividing data set for testing and training then build a generator for network structure, residual, and SE block network then go

on the generator that created patches and add some instance of perturbed to generate and evaluate an adversarial attack

Fig. 4 Working of Adversarial Attack**Fig. 5** Training Protocol

any additional parameters like learning rate or momentum.

3.6 Training Protocol

The Fast Gradient Sign Method (FGSM) is an adversarial attack method that generates adversarial examples by perturbing the input data based on the gradients of the target model. However, to refer to the training protocol for defending against FGSM attacks. The training protocol that aims to enhance the robustness of a model against FGSM attacks is shown in Fig. 5:

3.7 Adversarial Attack using FGSM and Output Generation

The adversarial attack to misidentify the data is performed by FGSM. Initially, the data set is divided into training and testing data sets then the preprocessing performs for a patch-based generator which generates the example of attack if successfully done the program end, and if not its updates the parameters for adversarial attacks. The original image is attacked by the notorious input or bug which generates the attack for the original image which is unidentified as the original. Figure 6 shows the attack of noise in the original image which gives you unidentified results from the original.

The fast gradient sign method is a technique that leverages the gradients of a neural network to generate an adversarial example. Specifically, it uses the gradients of the loss function with respect to the input image to create a new image that maximizes the loss given the input image. This new image is called the adversarial image. Figure 7 shows the epsilon range and achieves an accuracy of adverse attacks.

The graph depicts the attack success rate against different epsilon values. The x-axis represents the epsilon

Algorithm 1 Crafting an Adversarial Example

Input: A classifier f with loss function J ; a real example x and ground-truth label y ;

Input: The size of perturbation ϵ ; iterations T and decay factor μ .

Output: An adversarial example x^* with $\|x^* - x\|_\infty \leq \epsilon$.

```

1:  $\alpha = \epsilon / T$ ;
2:  $g_0 = 0$ ;  $x_0^* = x$ ;
3: for  $t = 0$  to  $t - 1$  do
4:   Input  $x_t^*$  to  $f$  obtain the gradient  $\nabla_x J(x_t^*, y)$ ;
5:   Update  $g_{t+1}$  by accumulating the velocity vector in the
      gradient direction as  $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1}$ ;
6:   Update  $x_{t+1}^*$  by applying the sign gradient as  $x_{t+1}^* = x_t^* + \alpha \cdot \text{sign}(g_{t+1})$ ;
7: end for
8: return  $x^* = x_T^*$ .
```

values, which range from 0.025 to 0.20 with an increment of 0.025. These epsilon values determine the magnitude of the perturbation applied to the input images during the attack. The y-axis represents the attack success rate, which is calculated as the percentage of successfully generated adversarial examples that are misclassified by the targeted model. A higher attack success rate indicates a greater ability of the adversarial examples to deceive the model and achieve the desired misclassification. As shown in the graph, the attack success rate generally increases as the epsilon values increase. Observe that this aligns with the intuition that larger epsilon values result in more significant perturbations, making it easier to generate adversarial examples that can fool the model.

$$Adv_x = x + \epsilon \times \text{sign}(\nabla_x J(\theta, x, y)) \quad (1)$$

where Adv_x = Adversarial Image

x = Original input image

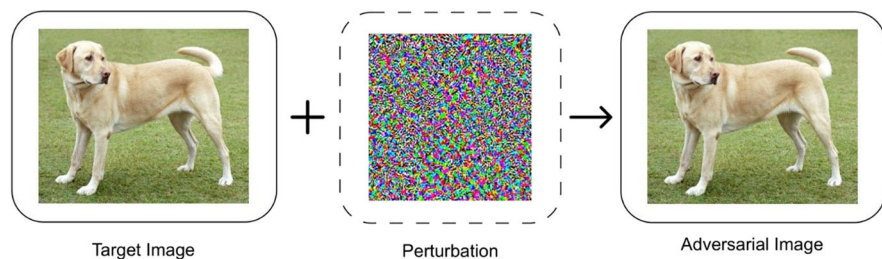
y = Original input label

ϵ = Multiplier to ensure the perturbations are small

θ = Model parameters

J = Loss

Fig. 6 Working of Adversarial Attack



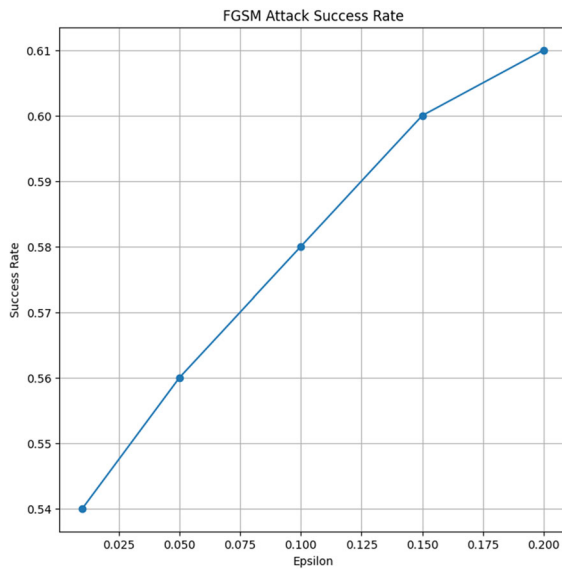


Fig. 7 Adversarial attack FGSM Epsilon

$$x \times 0 = x \quad (2)$$

$$x \times i = \text{clip } x, e(x \times i - 1 + e \text{ sign}(\nabla x \times i - 1 J(\theta, x \times i - 1, y))) \quad (3)$$

Here, $\text{clip } x, e$, denotes a clipping of the adversarial sample's values so that they are in the vicinity of the original sample x . This strategy is convenient because it gives the attacker more attack control.

For example, one can choose to stop the loop on an iteration when x is incorrectly categorized for the first time or to add more noise after that. This allows one to choose how far a sample is pushed past the classification boundary.

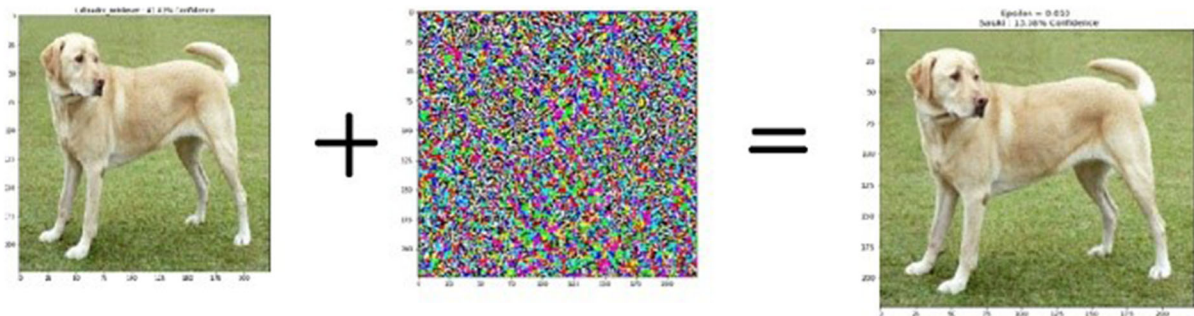


Fig. 8 Output of the research

4 Results

In this research, the data set related to dog types is used. The original image is a Labrador retriever having a confidence of 41.82%.

In the original image, the FGSM is implemented. Creating perturbations is the initial step in the process since they desire to be used to warp the initial image and produce an argumentative image. The gradients are measured in the picture. Figure 8 shows the gradient image. Epsilon can be raised, then the resulting image is observed. It has been observed that it is simpler to trick the network as epsilon's value rises. The disturbances do become easier to identify because of this trade-off, though.

The output is generated when the epsilon is increased to 0.01, it is not detected the same as the original image of a dog. The name of the output dog is Saluki. The results are generated and defined as the input image(data) which is attacked by adversarial noise and produces output in an unrecognized form of the data as shown in Fig. 8.

4.1 Evaluation Criteria

The Fast Gradient Sign Method (FGSM) is primarily designed to generate adversarial examples and evaluate the vulnerability of models against such attacks. FGSM is a simple yet effective attack method, but it may not provide a robust defense against more advanced and adaptive attack techniques. Let's discuss the performance and sustainability of an FGSM-trained model against different attacks in different scenarios:

4.1.1 FGSM Attack Performanc

FGSM-generated adversarial examples can significantly impact the performance of models trained without specific defenses. The attack is successful in causing misclassification by perturbing the input data along the direction of the model's gradients. The success rate of FGSM attacks depends on the chosen perturbation size (epsilon). Smaller epsilon values may result in subtle perturbations and a lower success rate, while larger epsilon values can cause more noticeable perturbations and higher success rates. FGSM attacks are generally effective against models trained without adversarial training or specific defenses against adversarial attacks.

4.1.2 Transferability

FGSM-generated adversarial examples are often not transferable to other models or architectures that were not trained using FGSM. The perturbations applied by FGSM may be specific to the targeted model's decision boundaries and may not generalize well to other models. However, there may be cases where transferability exists to some extent, particularly when the attacked model and the targeted model have similar architectures or decision boundaries.

4.1.3 Robustness Against Other Attacks

While FGSM is effective against models without adversarial training, it may not provide sufficient robustness against more sophisticated attacks, such as the Carlini and Wagner attack, DeepFool, or iterative attacks like Projected Gradient Descent (PGD). Models trained using FGSM may still be vulnerable to more advanced attacks that can adapt and bypass the defense mechanisms employed by FGSM. These attacks often involve solving optimization problems or incorporating additional constraints during the adversarial example generation process.

4.1.4 Limitations and Defense Mechanisms

FGSM training has limitations and may not guarantee complete robustness against adversarial attacks. To improve the sustainability of models against various attacks, additional defense mechanisms can be employed, such as adversarial training with more

advanced attack methods like PGD, defensive distillation, feature squeezing, input denoising, or ensemble methods. Robustness can also be enhanced by combining multiple defense techniques or incorporating architectural changes in the model to better detect and resist adversarial examples.

4.2 Ablation Tests

4.2.1 Evaluation of Attack

The success rate is 99.25%, with 400 queries on average proposed. The success rate of adversarial attacks is a crucial metric for evaluating the effectiveness of our method in generating adversarial examples. Figure 9 presents the graph depicting the attack success rate against different epsilon values. The x-axis represents the epsilon values, which range from 0.025 to 0.20 with an increment of 0.025. These epsilon values determine the magnitude of the perturbation applied to the input images during the attack. The y-axis represents the attack success rate, which is calculated as the percentage of successfully generated adversarial examples that the target misclassifies. A higher attack success rate indicates a greater ability of the adversarial examples to deceive the model and achieve the desired misclassification. As shown in the graph, the attack success rate generally increases as the epsilon values increase.

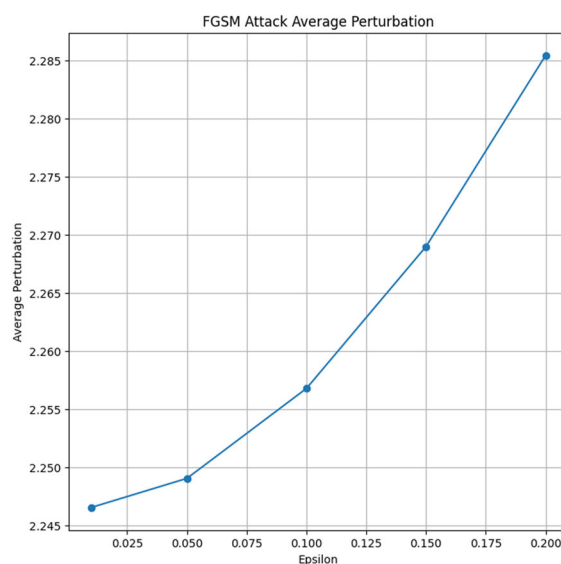


Fig. 9 Attack success rate

Observe that this aligns with the intuition that larger epsilon values result in more significant perturbations, making it easier to generate adversarial examples that can fool the model.

4.2.2 Accuracy vs Epsilon

The accuracy vs. epsilon graph illustrates the relationship between the robustness of the deep learning model and the magnitude of perturbations applied to the input image. Epsilon represents the level of perturbation or distortion applied to the original image to generate an adversarial example. The graph plots the accuracy of the model in classifying these adversarial examples against different epsilon values. Figure 10 shows that the x-axis of the graph represents the epsilon values, which range from low to high, and the y-axis represents the accuracy of the deep learning model in correctly classifying the adversarial examples. The first result is the accuracy versus the epsilon plot. As alluded to earlier, as epsilon increases, the trend in the curve is not linear, even though the epsilon values are linearly spaced. For example, the accuracy at $\epsilon = 0.01$, is only about 1% lower than $\epsilon = 0$. Notice that the accuracy of the model hits random accuracy for a 10-class classifier between $\epsilon = 0.15$ and $\epsilon = 0.2$. As the epsilon value increases, the accuracy of the model in classifying the adversarial examples decreases. The decline in accuracy indicates that the deep learning model becomes more vulnerable to adversarial attacks as the perturbations in the input image become more prominent. The decrease in accuracy signifies that even small changes

to the input image can cause the model to misclassify the adversarial examples. These findings indicate that the deep learning model being studied is not robust against adversarial attacks, as even small perturbations of 0.1 can lead to misclassifications.

4.3 Sample Adversarial Examples

The experiment involves altering the input image with either gradients or noise to create an output that is misidentified by the model. The experiment is implemented using Python and the TensorFlow framework. To optimize the experiment, an algorithm with a learning rate of 0.1 is used initially, and the learning rate is lowered depending on the number of training samples by setting various epoch values. Table 3 shows the outcomes of our successfully achieved research.

As epsilon increases, the test accuracy decreases, but the perturbations become more easily perceptible. There is a tradeoff between accuracy degradation and perceptibility. Here, the study shows some examples of successful adversarial strategies at each epsilon value. Figure 11 shows that each row of the plot has a different epsilon value. The first row is the $\epsilon = 0$ examples, which represent the original “clean” image with no perturbation as a true label of ‘Labrador retriever’. The title of each image shows the adversarial classification. Notice that the perturbations start to become evident at $\epsilon = 0.02$ and quite evident at $\epsilon = 0.05$. However, in all cases, humans are still capable of identifying the correct class despite the added noise but the model is misleading.

Figure 12 shows how the original image, which should be detected, is disturbed after the adversarial attack, resulting in a wrong detection that is not identified as the original.

In this research FGSM fast gradient Sign Method is used which is a quick and efficient way to create visuals that are hostile. Using an image as a source utilizing a trained CNN to make predictions about the image calculating the prediction’s loss based on the correct class label, calculating the loss gradients in relation to the input picture, and calculating the gradient’s sign.

The chosen model need not be resistant to hostile cases under standard supervised training. The training process must somehow incorporate this trait, which is where adversarial training comes in.

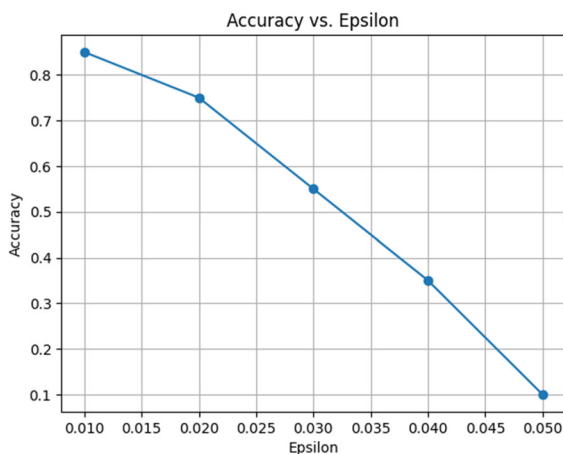


Fig. 10 Graph of Accuracy vs Epsilon

Table 3 Output of Proposed Algorithm

Original Image	Epsilon	Calculations (millions)	Parameters (millions)	Generated image
Labrador retriever	0.00	568	3.21	Labrador retriever
Labrador retriever	0.010	568	3.21	Saluki

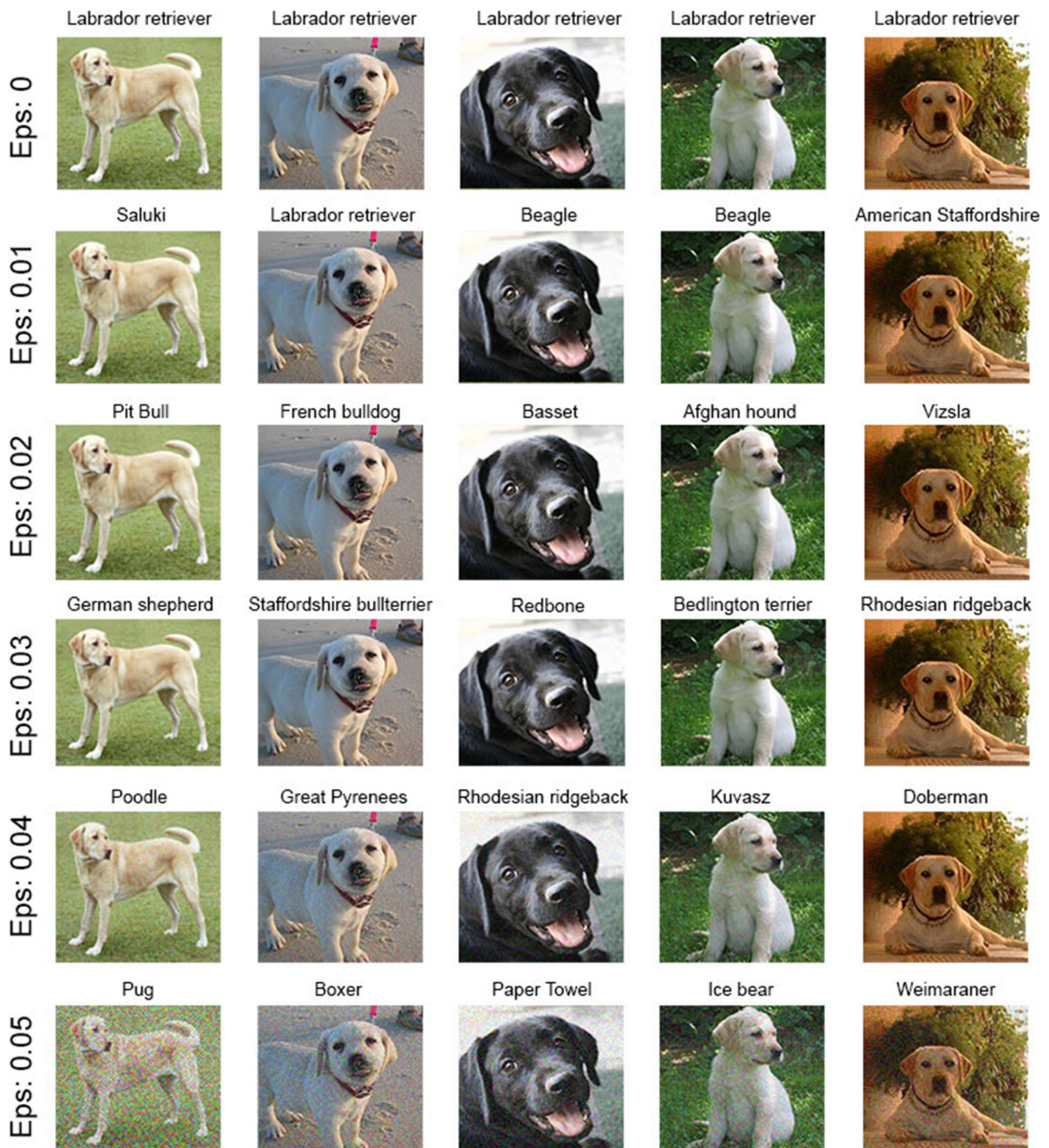
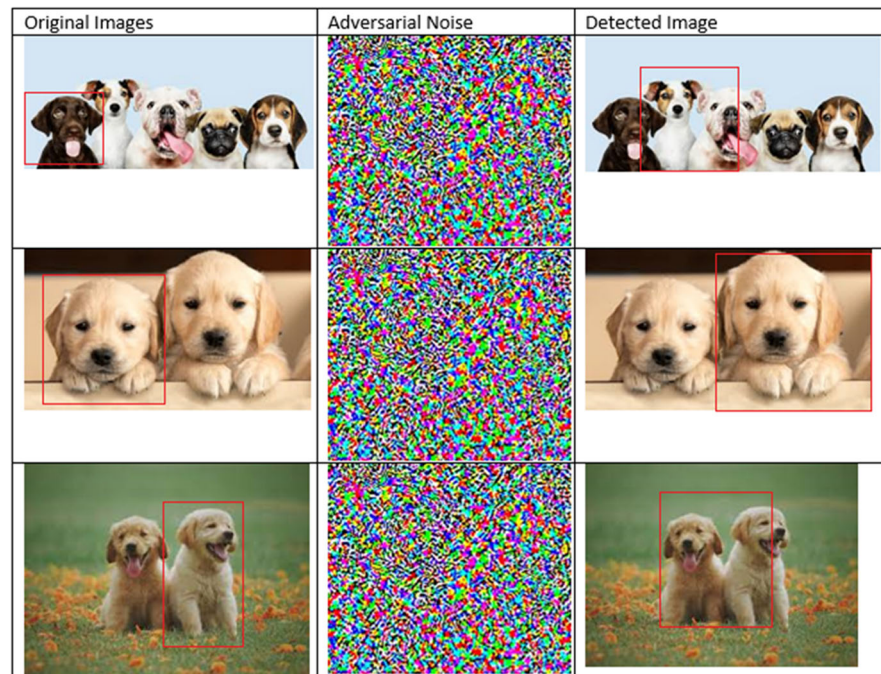
**Fig. 11** Adversarial samples at each epsilon

Fig. 12 Unidentified detection of an algorithm

It has been demonstrated that training a neural network on a combination of adversarial and clean samples can partially regularize it. Data augmentation uses methods like translation, rotation, etc. to create input samples that are comparable to what may be anticipated in the test set. The use of inputs that are unlikely to occur normally reveals problems with how the model conceptualizes its decision function making training on adversarial samples slightly distinct from other approaches. Hence, a diverse body of literature.

Our model is performing well at predicting digits, as evidenced by the script in Fig. 12, with accuracy scores of 99.25. This research focuses on a methodical investigation of the optimum strategy for adversarial training techniques that are computationally effective. We recommend using the FGSM regularization strategy to help existing algorithms overcome the problem of catastrophic overfitting. Although adversarial resilience is a desirable quality that improves the reliability of machine learning models, we have observed positive results from our research. Consequently, it is critical to be able to train resilient models effectively without restricting effective training to only small-scale perturbations. The comparison between our pro-

posed method and other existing techniques presents in Table 4.

Making the output adversarial image using the signed gradient. Although this procedure might seem difficult, the main contribution is achieved by being able to implement the full FGSM function in just a few lines of code.

5 Conclusion

The research to produce an adversarial attack has successfully achieved its goal of creating an attack that is smart enough to misidentify the original image. Such attacks have very dangerous real-world applications; for instance, one may alter a traffic sign such that an autonomous vehicle interprets it incorrectly and causes an accident. Another illustration is the possible risk of unsuitable or unlawful content being altered so that it cannot be detected by police web crawlers or the content moderation algorithms employed on well-known websites. The future of this research is to implement a video data set and identify the solution to block adversarial attacks.

Table 4 Comparative Analysis

Reference	Findings	Accessibility	Domain	Strategy
[32]	Targeted	Black Box	Text	Euclidean distance
[33]	Targeted	Black Box	Image	Gradient
[34]	Non targeted	Black Box	Text and image	WGAN-based
[35]	Non targeted	White and black box	Text	Gradient
Our Technique	Targeted	Black Box	Image	Full FGSM

5.1 Potential Risk

Mitigating the potential risks posed by adversarial attacks is an active area of research. While it is challenging to achieve complete robustness, several preventive measures can help mitigate the impact of such attacks. Here are some commonly employed techniques:

1. **Adversarial Training:** Adversarial training involves augmenting the training data with adversarial examples generated during the training process. By exposing the model to both clean and adversarial examples, it learns to be more resilient to adversarial perturbations. This approach helps the model generalize better and improves its ability to detect and classify adversarial inputs.
2. **Defensive Distillation:** Defensive distillation is a technique that involves training a model using the softened output probabilities of a pre-trained model. The idea is to make the model less sensitive to small changes in the input by smoothing the output probabilities. This technique can make the model more robust against adversarial attacks, particularly those based on gradient information.
3. **Input Preprocessing and Transformation:** Applying input preprocessing techniques, such as input normalization, can help reduce the effectiveness of adversarial attacks. Additionally, transformations like random resizing, cropping, or rotation during the training phase can make the model more robust to specific perturbations and increase its generalization capability.
4. **Ensemble Methods:** Ensembling involves combining multiple models or defenses to improve robustness. By training and aggregating the predictions of multiple models, the ensemble approach can reduce the impact of adversarial attacks. Different models may have varying vulnerabilities, making it harder for an attacker to craft adversarial examples that fool all models simultaneously.
5. **Feature Squeezing:** Feature squeezing is a method that reduces the dimensionality or granularity of the input features. It can help detect and remove adversarial perturbations by squeezing the input features into a lower bit-depth representation. By decreasing the space for potential perturbations, feature squeezing can make the model more resilient to adversarial attacks.
6. **Gradient Masking:** Gradient masking involves intentionally limiting access to gradient information during the training process. By controlling the flow of gradient updates, the attacker's ability to estimate gradients accurately is hindered, making it more challenging to craft effective adversarial examples. Techniques like defensive distillation and randomized gradient masking can be employed to achieve gradient masking.
7. **Robust Model Architecture:** Designing models with robust architectures can help alleviate the impact of adversarial attacks. Techniques like adversarial training, incorporating defensive layers (e.g., feature denoising, spatial smoothing), or using model architectures with inherent robustness properties (e.g., robust deep neural networks) can enhance the model's resistance to adversarial attacks.
8. **Regularization Techniques:** Applying regularization techniques such as L1 or L2 regularization, dropout, or weight decay during training can add a form of noise to the learning process. This regularization can make the model more robust to small perturbations in the input data and limit the effect of adversarial examples.

In future work, adversarial attacks will be used in more situations, such as semantic segmentation, visual object tracking, and fog-like rains. This will help us

improve our adversarial attack. Also, we will evaluate DNN using our adversarial attack's novel mutation. The adversarial attack's formulation is halfway between conventional additive noise attacks and fully non-additive noise attacks. Further investigation is required to explore the potential impact of the adversarial rain on other adversarial attack mechanisms. This could provide valuable insights into how different adversarial attacks interact and could help in developing more effective defense strategies.

The examination of combative attack and protection approaches against adversarial attacks has gained significant attention from both the machine learning and security communities. Adversarial attacks refer to deliberate manipulations of inputs to machine learning models in order to deceive or mislead them. It has witnessed collaborations between machine learning experts and security professionals to address the evolving landscape of adversarial attacks. Defense Mechanisms: In response to adversarial attacks, a wide range of defense mechanisms have been proposed. These mechanisms aim to enhance the robustness and resilience of machine learning models against adversarial manipulations. Some common defense techniques include adversarial training, defensive distillation, input sanitization, model ensemble, and gradient masking. These approaches attempt to detect, mitigate, or eliminate the effects of adversarial perturbations on the model's decision-making process. Adversarial Detection and Robustness Metrics: Researchers have also focused on developing metrics and detection techniques to identify and quantify adversarial examples. These methods aim to provide insights into the vulnerability of machine learning models and assess their robustness against adversarial attacks. Metrics such as robust accuracy, robustness bounds, and adversarial distance measures help evaluate the effectiveness of defense mechanisms and guide the development of more robust models.

Author contributions S.M.A: Conceptualization, Methodology, Formal analysis, Supervision, Writing - original draft, Writing - review & editing. M.S: Investigation, Data Curation, Validation, Resources, Writing - review & editing. M.A.K: Project administration, Investigation, Writing - review & editing. S.I.H: Writing - original draft, Writing - review & editing.

Funding This research received no specific grant from any funding agency.

Availability of Data and Materials Data is available upon request.

Declarations

Competing Interests The authors declare no competing interests.

References

- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017)
- Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1328–1338 (2019)
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
- Xu, H., Ma, Y., Liu, H.-C., Deb, D., Liu, H., Tang, J.-L., Jain, A.K.: Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* **17**, 151–178 (2020)
- Bai, Y., Zeng, Y., Jiang, Y., Wang, Y., Xia, S.-T., Guo, W.: Improving query efficiency of black-box adversarial attack. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, pp. 101–116 (2020). Springer
- Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(9), 2805–2824 (2019)
- Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
- Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. *Stat. Surv.* **16**, 1–85 (2022)
- Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* **23**(5), 828–841 (2019)
- Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: TextFool: fool your model with natural adversarial text (2019)
- Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1369–1378 (2017)
- Lee, L., Rose, R.: A frequency warping approach to speaker normalization. *IEEE Trans. Speech Audio Process.* **6**(1), 49–60 (1998)
- Gao, J., Yan, D., Dong, M.: On the robustness of speech emotion models to black-box adversarial attack (2022)
- Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial exam-

- ples with input diversity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2730–2739 (2019)
15. Zhang, M., Zhang, Y., Zhang, L., Liu, C., Khurshid, S.: Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 132–142 (2018)
 16. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5764–5772 (2017)
 17. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint [arXiv:1703.00410](https://arxiv.org/abs/1703.00410) (2017)
 18. Tian, S., Yang, G., Cai, Y.: Detecting adversarial examples through image transformation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
 19. Kwon, H., Lee, S.: Ensemble transfer attack targeting text classification systems. *Comput. Secur.* **117**, 102695 (2022)
 20. Ma, X., Li, B., Wang, Y., Erfani, S.M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M.E., Bailey, J.: Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint [arXiv:1801.02613](https://arxiv.org/abs/1801.02613) (2018)
 21. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
 22. Mao, G., Li, L., Wang, Q., Li, J.: Study on the method of adversarial example attack based on mi-fgsm. In: Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the IIH-MSP 2021 & FITAT 2021, Kaohsiung, Taiwan, Volume 1, pp. 281–288. Springer, 978-981-19-1057-9 (2022)
 23. Yu, M., Sun, S.: Fe-dast: Fast and effective data-free substitute training for blackbox adversarial attacks. *Comput. Secur.* **113**, 102555 (2022)
 24. Yang, W., Tan, R.T., Wang, S., Fang, Y., Liu, J.: Single image deraining: From model-based to data-driven and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(11), 4059–4077 (2020)
 25. Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(9), 2805–2824 (2019)
 26. Zhang, H., Patel, V.M.: Density-aware single image deraining using a multistream dense network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 695–704 (2018)
 27. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. *IEEE Trans. Circuits Syst. Video Technol.* **30**(11), 3943–3956 (2019)
 28. Liu, J., Zhang, Q., Mo, K., Xiang, X., Li, J., Cheng, D., Gao, R., Liu, B., Chen, K., Wei, G.: An efficient adversarial example generation algorithm based on an accelerated gradient iterative fast gradient. *Comput. Stand. Interfaces* **82**, 103612 (2022)
 29. Lu, S., Wang, M., Wang, D., Wei, X., Xiao, S., Wang, Z., Han, N., Wang, L.: Black-box attacks against log anomaly detection with adversarial examples. *Inf. Sci.* **619**, 249–262 (2023)
 30. Wang, J., Wang, C., Lin, Q., Luo, C., Wu, C., Li, J.: Adversarial attacks and defenses in deep learning for image recognition: A survey. *Neurocomputing* (2022)
 31. Wang, C., Wang, J., Lin, Q.: Adversarial attacks and defenses in deep learning: A survey. In: Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, August 12–15, 2021, Proceedings, Part I 17, pp. 450–461 (2021). Springer
 32. Grosse, K., Papernot, N., Manoharan, P., Backes, M., McDaniel, P.: Adversarial perturbations against deep neural networks for malware classification. arXiv preprint [arXiv:1606.04435](https://arxiv.org/abs/1606.04435) (2016)
 33. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: Artificial Intelligence Safety and Security, pp. 99–112. Chapman and Hall/CRC, [arXiv:1607.02533](https://arxiv.org/abs/1607.02533) (2018)
 34. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint [arXiv:1412.6572](https://arxiv.org/abs/1412.6572) (2014)
 35. Linzen, T., Chrupala, G., Belinkov, Y., Hupkes, D.: Proceedings of the 2019 acl workshop blackboxnlp: Analyzing and interpreting neural networks for nlp. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (2019)
 36. Yin, X., Kolouri, S., Rohde, G.K.: Divide-and-conquer adversarial detection. *CoRR*, abs/1905.11475 [arXiv:1905.11475](https://arxiv.org/abs/1905.11475) (2019)
 37. Shumailov, I., Zhao, Y., Mullins, R., Anderson, R.: Towards certifiable adversarial sample detection. In: Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security, pp. 13–24 (2020)
 38. Vacanti, G., Van Looveren, A.: Adversarial detection and correction by matching prediction distributions. arXiv preprint [arXiv:2002.09364](https://arxiv.org/abs/2002.09364) (2020)
 39. Freitas, S., Chen, S.-T., Wang, Z.J., Chau, D.H.: Unmask: Adversarial detection and defense through robust feature alignment. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 1081–1088 (2020). IEEE
 40. v2, M.: Dataset. https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224.h5. [MobileNet v2] (2018)
 41. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1357–1366 (2017)
 42. Xie, X., Ma, L., Wang, H., Li, Y., Liu, Y., Li, X.: Diffchaser: Detecting disagreements for deep neural networks. International Joint Conferences on Artificial Intelligence Organization (2019)
 43. Yang, W., Liu, J., Yang, S., Guo, Z.: Scale-free single image deraining via visibility-enhanced recurrent wavelet learning. *IEEE Trans. Image Process.* **28**(6), 2948–2961 (2019)
 44. Xie, X., Ma, L., Juefei-Xu, F., Xue, M., Chen, H., Liu, Y., Zhao, J., Li, B., Yin, J., See, S.: Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 146–157 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.