mathematics

MDPI

Article

# Enhancing the Transferability of Adversarial Examples with Feature Transformation

**Hao-Qi Xu [1,2], Cong Hu [1,2,*] and He-Feng Yin [1,2]**

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
[2] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China
* Correspondence: conghu@jiangnan.edu.cn

**Abstract:** The transferability of adversarial examples allows the attacker to fool deep neural networks (DNNs) without knowing any information about the target models. The current input transformation-based method generates adversarial examples by transforming the image in the input space, which implicitly integrates a set of models by concatenating image transformation into the trained model. However, the input transformation-based methods ignore the manifold embedding and hardly extract intrinsic information from high-dimensional data. To this end, we propose a novel feature transformation-based method (FTM), which conducts feature transformation in the feature space. FTM can improve the robustness of adversarial example by transforming the features of data. Combining with FTM, the intrinsic features of adversarial examples are extracted to generate transferable adversarial examples. The experimental results on two benchmark datasets show that FTM could effectively improve the attack success rate (ASR) of the state-of-the-art (SOTA) methods. FTM improves the attack success rate of the Scale-Invariant Method on Inception_v3 from 62.6% to 75.1% on ImageNet, which is a large margin of 12.5%.

**Keywords:** adversarial example; feature transformation; black-box attack; ensemble attack; deep neural network

**MSC:** 68T10



**Citation:** Xu, H.-Q.; Hu, C.; Yin, H.-F. Enhancing the Transferability of Adversarial Examples with Feature Transformation. *Mathematics* **2022**, *10*, 2976. https://doi.org/10.3390/math10162976

Academic Editors: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 26 July 2022
Accepted: 17 August 2022
Published: 18 August 2022



## 1. Introduction

DNNs have been shown to perform well in many fields, for example, image classification [1–3], human recognition [4], image segmentation [5], image fusion [6], visual object tracking [7,8], super-resolution [9], etc [10]. The ultimate goal of these studies is to make DNN-based applications more practicable and efficient. However, the existence of adversarial examples presents a concern for security of many applications, such as autonomous driving [11], face recognition [12–14], etc.

Adversarial examples [15], generated by adding indistinguishable perturbations to the raw images, can lead the DNNs to make completely different predictions. They can even take effect for completely unknown models, which is called the transferability of adversarial examples. In addition to this, there are several studies on universal adversarial perturbations [16,17], which are able to take effect on any image. Some studies are devoted to the application of adversarial examples to real-world scenarios, such as face recognition, autonomous driving, etc. [18–22]. Studying both adversarial attack and defense [23–26] is of significance, not only in revealing the vulnerability of DNNs, but also in improving the robustness of DNNs.

Many white-box attack methods have been proposed, such as Fast Gradient Sign Method (FGSM) [27], Basic Iterative Method (BIM) [28], etc. However, it is difficult for an attacker to obtain the structure and other parameters of the target model in the real-world situation. Therefore, various approaches have emerged to enhance the transferability of

adversarial examples for black-box attack. Ensemble Attack [29] is an effective method to enhance the transferability of adversarial examples. Lin et al. [30] proposed Scale-Invariant Method (SIM), which utilizes input transformation to obtain a new model. A set of models can be obtained by using different transformations several times. With this approach, they can perform an ensemble attack with only one trained model, which is an implicit ensemble attack. Input transformation-based methods are successfully used for an adversarial attack, such as Diverse Input Method (DIM) [31], Translation-Invariant Method (TIM) [32], Admix Attack Method (Admix) [33], etc. However, these methods ignore the manifold structure of adversarial examples and few works focus on feature transformation. To this end, this work proposes a feature transformation-based method (FTM) to improve the transferability of adversarial examples. Compared with the input transformation, our approach transforms the intrinsic features of data instead of the input images. FTM is an implicit ensemble attack that can simultaneously attack multiple models that extract different features. It can improve the robustness of the adversarial example at the feature level. This work proposes several feature transformation strategies. FTM could effectively improve the performance of the SOTA adversarial attacks. Our contributions can be summarized as follows.

- This work proposes a novel feature transformation-based method (FTM) for enhancing the transferability of adversarial examples.
- We propose several feature transformation strategies and comprehensively analyze the hyper-parameters of them.
- The experimental results on two benchmark datasets show that FTM could effectively improve the attack success rate of the SOTA methods.

The structure of the paper is organized as follows. Section 2 introduces related work. Section 3 details the proposed FTM. Section 4 shows the experimental results. Section 5 gives a summary of this work.

## 2. Related Work

### 2.1. Adversarial Example and Transferability

It is firstly pointed out by Szegedy et al. [15] that DNNs are vulnerable to adversarial examples, which are generated by adding imperceptible noises to raw images.

Let $x$ be a clean image, $y = f(x;\theta)$ be the output label predicted by the model with parameters $\theta$, and $||\cdots||_p$ denotes the $p$-norm. The adversarial example is an image $x^{adv}$ whose output label $f(x^{adv},\theta) \neq f(x,\theta)$, and the $L_p$ norm of the adversarial perturbation $x^{adv} - x$ is smaller than a threshold $\epsilon$ as $||x^{adv} - x|| \leq \epsilon$. $p = \infty$ is used to limit the distortion. Many methods are proposed to improve the attack success rate (ASR) of adversarial examples. These methods can be divided into two branches: advanced gradient calculation and input transformations.

### 2.2. Advanced Gradient Calculation

This branch exploits better gradient calculation algorithms to enhance the performance of adversarial examples in both white-box settings and black-box settings.

**Fast Gradient Sign Method (FGSM)**: Szegedy et al. [27] make the point that linear behavior in high-dimensional spaces is sufficient to cause adversarial examples. According to this point, they propose the FGSM, which generates an adversarial example $x^{adv}$ by maximizing the loss function $J(x^{adv}, y;\theta)$ with a one-step update:

$$x^{adv} = x + \epsilon \cdot sign(\nabla_x J(x, y, \theta)) \tag{1}$$

where $J(x, y : \theta)$ denotes the loss function of classifier $f(x : \theta)$, $\nabla_x J(x, y, \theta)$ is the gradient of loss function with regard to $x$ and $sign(\cdot)$ is the sign function to make the perturbation meet the $L_p$ norm bound.

**Basic Iterative Method (BIM)**: Kurakin et al. [28] extend FGSM to an iterative version by iteratively applying gradient updates multiple times with a small step size $\alpha$. BIM can be expressed as:

$$x_{t+1}^{adv} = Clip_x^{\epsilon}\{x_t^{adv} + \alpha \cdot sign(\nabla_x J(x, y, \theta))\} \tag{2}$$

where $x_0 = x$ and $Cilp_x^{\epsilon}(\cdot)$ restricts generated adversarial examples to be within the $\epsilon$-ball of $x$.

**Momentum Iterative Fast Gradient Sign Method (MI-FGSM)**: To reduce the variation in update direction and avoid local minima, Dong et al. [34] introduce momentum into the BIM. The update procedure is formulated as follows:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x, y, \theta)}{||\nabla_x J(x, y, \theta)||_1} \tag{3}$$

$$x_{t+1}^{adv} = Clip_x^{\epsilon}\{x_t^{adv} + \alpha \cdot sign(g_{t+1})\} \tag{4}$$

where $g_t$ gathers the gradient of the first $t$ iterations with a decay factor $\mu$.

**Nesterov Iterative Fast Gradient Sign Method (NI-FGSM)**: NI-FGSM [30] adopts Nesterov's accelerated gradient to improve the transferability of MI-FGSM. This method replaces $x_t^{adv}$ in Equation (4) with $x_{nest}$, while $x_{nest}$ can be formulated as follows:

$$x_{nest} = x_t^{adv} + \alpha \cdot \mu \cdot g_t \tag{5}$$

### *2.3. Input Transformations*

Various input transformation-based methods, such as DIM, TIM, SIM, and Admix, are proposed to generate transferable adversarial examples.

**Diverse Input Method (DIM)**: Inspired by the facts that data augmentation is effective to prevent networks from overfitting, Xie et al. [31] apply random resizing and random padding to the inputs to improve the transferability of adversarial examples.

**Translation-Invariant Method (TIM)**: Dong et al. [32] propose to replace the gradient on the original image with the average value of multiple translated images for the update. Inspired by the translation-invariant property, they approximate this process by convolving the gradient with a predefined kernel matrix to avoid introducing much more computations.

**Scale-Invariant Method (SIM)**: Lin et al. [30] discover the scale-invariant property of deep learning models and introduce the definition of loss-preserving transformation and model augmentation. Accordingly, they present SIM that calculates the average gradient on the scaled copies of the original image for the update.

**Admix Attack Method (Admix)**: Admix is proposed by [33] to enhance the transferability of the adversarial examples. It integrates gradient information of different categories of images for the update. Specifically, Admix randomly selects a number of different categories of images and then admix the sampled image with a minor weight to the original input image. It calculates the gradient on the mixed image for update.

### *2.4. Adversarial Defense*

In addition to adversarial attacks, many works on adversarial defense have been proposed to improve the robustness of the classifiers. The current defense methods can be divided into two categories.

One category aims to improve the robustness of the classifier itself, such as adversarial training [27,35]. It adds adversarial examples to the training set during the training of the model, making it immune to the adversarial examples. This is a popular and effective defense method and has many great following works [36,37]. However, its effectiveness is largely limited by the method of generation of the added adversarial examples.

Another category of defense methods reduces the impact of adversarial perturbations by modifying the input images, such as adding noises and compressing the images [38,39]. Xie et al. [40] propose to perform randomized resizing and padding to inputs at inference

time, which is the top-1 defense solution in the NIPS competition. Nips-r3 fuse multiple adversarial trained models and perform several input transformations at inference time. These methods require no additional training computational overhead and are effective against various attack approaches.

## 3. Our Approach

A DNN model could be formulated as $f(x) = \text{lin}(\text{con}(x))$, where $\text{con}(\cdot)$ and $\text{lin}(\cdot)$ denote the convolutional part and the fully connected part, respectively. $\boldsymbol{p} = \text{con}(x)$ denotes the feature extracted by the convolutional part.

To obtain an ensemble of models that extract different features, we propose the feature transformation denoted as $\text{FT}(\cdot)$. Through introducing feature transformation, we can obtain a new model $f'(x) = \text{lin}(p') = \text{lin}(\text{FT}(\text{con}(x)))$ extracting different features from the original model during every iteration. FTM optimizes the adversarial perturbations over several different transformed features:

$$\arg\min_{x^{adv}} \frac{1}{m}\sum_{i=0}^{m} \text{J}(\text{lin}(\text{FT}_\text{i}(\text{con}(x^{adv}))), y_{true}), \tag{6}$$

$$s.t., \ \|x^{adv} - x\|_\infty \leq \epsilon, \tag{7}$$

where $m$ denotes the number of iterations and $\text{FT}(\cdot)$ denotes the feature transformation. Thus, FTM is an implicit ensemble attack that simultaneously attacks $m$ models. The illustration of the FTM is shown in Figure 1.

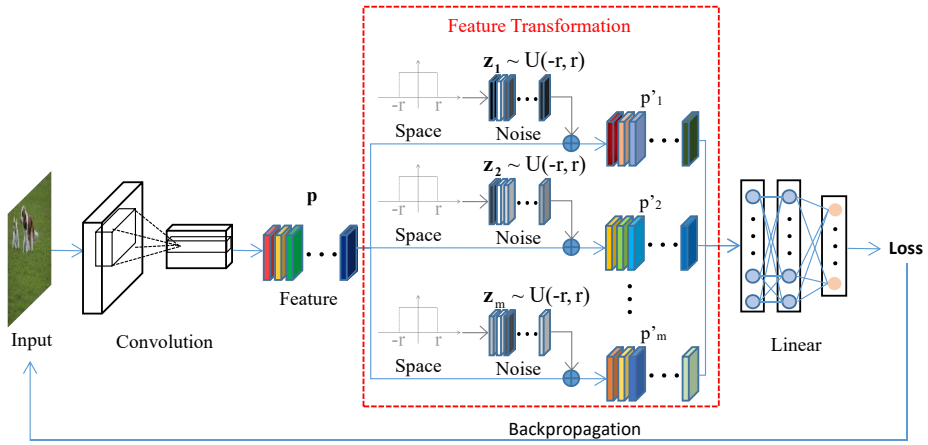


**Figure 1.** Illustration of the proposed FTM. The feature transformation shown in the illustration is the Strategy I. The random noise vectors $\boldsymbol{z_i}$ sampled from the uniform distribution are added to the feature $\boldsymbol{p}$. The average loss of the transformed features is calculated to update the input image.

In this paper, we consider five strategies of feature transformation as follows:

Strategy I: Fixed threshold random noise: Add a random vector $\boldsymbol{z}$ sampled from the uniform distribution $\mathcal{U}(-r, r)$:

$$\text{FT}(p) = \boldsymbol{p} + \boldsymbol{z} \tag{8}$$

Strategy II: Mean-based threshold random noise: $\boldsymbol{z}$ is a random vector sampled from the uniform distribution $\mathcal{U}(-r, r)$ and $\overline{p}$ is the mean value of feature $\boldsymbol{p}$. Adding $\overline{p} \cdot z$ to feature $p$:

$$\text{FT}(p) = \boldsymbol{p} + \overline{p} \cdot \boldsymbol{z} \tag{9}$$

Strategy III: Feature overall scaled: Multiply the features $\boldsymbol{p}$ by a random number $k$ sampled from the uniform distribution $\mathcal{U}(-r, r)$:

$$\mathrm{FT}(p) = k \cdot \boldsymbol{p} \tag{10}$$

Strategy IV: Each value of feature scaled separately: Multiply feature $\boldsymbol{p}$ by a random vector $\boldsymbol{z}$ sampled from the uniform distribution $\mathcal{U}(-r, r)$:

$$\mathrm{FT}(p) = \boldsymbol{z} \cdot \boldsymbol{p} \tag{11}$$

Strategy V: Offset mean random noise: Add a random vector $\boldsymbol{z}$ sampled from the uniform distribution $\mathcal{U}(-r+s, r+s)$ to feature $\boldsymbol{p}$:

$$\mathrm{FT}(p) = \boldsymbol{p} + \boldsymbol{z} \tag{12}$$

The feature transformation should also be an accuracy-preserving transformation. We define the accuracy-preserving feature transformation as follows:

**Definition 1** (Acc-preserving Feature Transformation)**.** *Given a test set $X$ and a classifier $f(x) = \mathrm{lin}(\mathrm{con}(x))$, $Acc(\mathrm{lin}(\mathrm{con}(x)), X)$ denotes the accuracy of model $f(x)$ on data set $X$. If there exists an feature transformation $\mathrm{FT}(\cdot)$ that satisfies $Acc(\mathrm{lin}(\mathrm{con}(x)), X) \approx Acc(\mathrm{lin}(\mathrm{FT}(\mathrm{con}(x))), X)$, we say $\mathrm{FT}(\cdot)$ is an accuracy-preserving feature transformation.*

We experimentally study the acc-preserving feature transformation strategies in Section 4.1.2. We determine the magnitude $r$ of uniform distribution to ensure that our feature transformations are accuracy-preserving transformations. The algorithm of the FTM attack is summarized in Algorithm 1.

**Algorithm 1** Algorithm of FTM.

**Input:** Original image $x$, true label $y^{true}$, a classifier $f = \mathrm{lin}(\mathrm{con}(x))$, loss function $J$, feature transformation $\mathrm{FT}(\cdot)$
**Hyper-parameters:** Perturbation size $\epsilon$, maximum iterations $T$, number of iterations of feature transformation $m$
**Output:** Adversarial example $x_{adv}$
1: perturbation size in each iteration: $\alpha = \epsilon / T$
2: **while** $0 \leq t < T - 1$.
3: **if** $k = 0$.
4: $x_0 = x$.
5: **end if**
6: $g = 0$
7: **while** $0 \leq i < m - 1$
8: feature:$\boldsymbol{p} = \mathrm{con}(x)$
9: transformed feature: $\boldsymbol{p'} = \mathrm{FT}(\boldsymbol{p})$
10: Get the gradients by $\nabla_x J(\mathrm{lin}(\boldsymbol{p'}), y^{true})$
11: Update $g = g + \nabla_x J(\mathrm{lin}(\boldsymbol{p'}), y^{true})$
12: **end while**
13: Get average gradients as $\overline{g} = \frac{1}{m} \cdot g$
14: Update $x_{i+1}^{adv} = \mathrm{Clip}_x^{\epsilon}\{x_i^{adv} + \alpha \cdot \mathrm{sign}(\overline{g})\}$
15: **end while**
16: return $x^{adv} = x_T^{adv}$

## 4. Experimental Results

### *4.1. Experiment on ImageNet*

#### 4.1.1. Experimental Setup

**Dataset.** We perform experiments on ImageNet, which is the most common and challenging image classification dataset. 1000 images from the ImageNet [41] are selected as our test set. The 1000 benign images belong to 1000 different categories and can be correctly classified by the tested models.

**Networks.** This work selects four mainstream models, including Inception_v3 (Inc_v3) [42], Inception_v4 (Inc_v4), Inception-Resnet_v2 (IncRes_v2) [43], and Xception(Xcep) [44].

**Attack setting.** We follow the setting in Lin et al. [30] with the maximum perturbation as $\epsilon = 16$, number of iteration $T = 16$, and step size $\alpha = 1.6$, which is a difficult and challenging attack setting. We adopt the decay factor $\mu = 1.0$ for MI-FGSM. The transformation probability is set to 0.5 for DIM. The number of scale copies is set to $m = 5$ for SIM. We set $m_1 = 5$ , and randomly sample $m_2 = 3$ images with $\eta = 0.2$ for Admix. The hyper-parameter settings of these attack methods are all consistent with the original papers.

#### 4.1.2. Accuracy-Preserving Transformation

To investigate accuracy-preserving transformations, we test the accuracy of the models integrated with the five strategies on the ImageNet dataset. We keep the magnitude $r$ of uniform distribution in the range of $[0, 10]$.

The magnitude of uniform distribution is an important hyper-parameter of FTM. A larger magnitude will increase the diversity of the implicit ensemble models and thus improve the transferability of the adversarial examples. However, too large a magnitude will make the model invalid and thus lose the ability to guide the generation of AE. As shown in Figure 2, the accuracy curves are smooth and stable for strategies I, II, and V when the magnitude is in the range of $[0, 4]$. They drop significantly after the magnitude exceeds 4. Moreover, the accuracies for strategy III and IV are extremely low when the magnitude is close to 0. They turn to remain stable after the magnitude exceeds 4. It can be seen that the feature transformation strategy with scaled operation is more sensitive to small magnitude, e.g., strategies III and IV. The feature transformation strategy of adding noise is more sensitive to a large magnitude, e.g., strategies I, II, and V. Based on the experimental results, the magnitude of uniform distribution is set to 4 in the following experiment to ensure that the feature transformations are accuracy-preserving transformations.

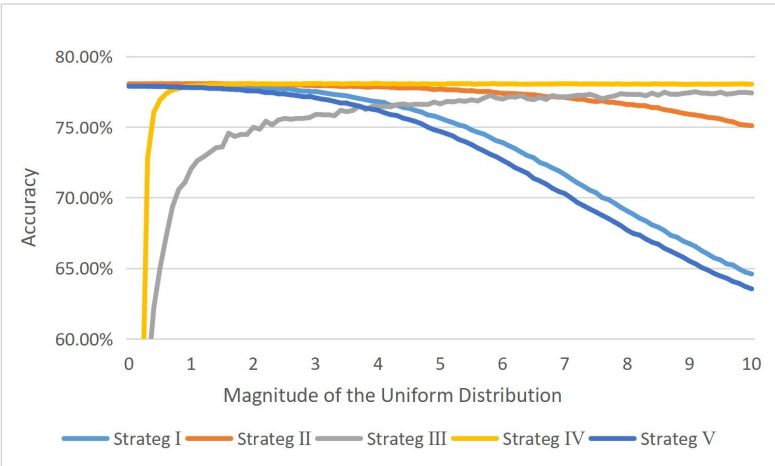


**Figure 2.** The average classification accuracy of Inc_v3, Inc_v4, IncRes_v2, and Xcep integrated with five different feature transformation strategies on ImageNet. The horizontal coordinate is the magnitude of uniform distribution and the vertical coordinate is the accuracy of the model.

#### 4.1.3. Feature Transformation Strategies

In this section, we show the experimental results of the proposed FTM with five feature transformation strategies. We set $m = 1$ and generate adversarial examples on the Inc_v3 by FT-FGSM, FT-MI-FGSM, and FT-SIM. The ASRs against the other three black-box models are presented in Table 1.

**Table 1.** The black-box ASRs (%) of FT-FGSM, FT-MI-FGSM, and FT-SIM with five strategies on ImageNet. The adversarial examples are generated on Inc_v3. The highest ASRs are shown in bold.

| Method | Strategy | Inc_v3 | Inc_v4 | IncRes_v2 | Xcep |
|---|---|---|---|---|---|
| FT-FGSM | I | - | 36.1 | 33.5 | 35.3 |
| | II | - | 37.3 | 33.7 | 35.1 |
| | III | - | 37.0 | **35.9** | **37.5** |
| | IV | - | 37.5 | 32.0 | 34.7 |
| | V | - | **37.7** | 33.4 | 34.4 |
| FT-MI-FGSM | I | - | 55.1 | 52.5 | **59.8** |
| | II | - | 53.0 | 50.4 | 54.4 |
| | III | - | 54.9 | 51.6 | 57.8 |
| | IV | - | 53.4 | 50.8 | 56.5 |
| | V | - | **57.0** | **53.3** | 59.2 |
| FT-SIM | I | - | **43.0** | 41.3 | 42.9 |
| | II | - | 38.5 | 34.9 | 39.3 |
| | III | - | 42.9 | **42.6** | **44.0** |
| | IV | - | 42.2 | 42.4 | 43.5 |
| | V | - | 41.1 | 41.9 | 42.6 |

When combined with FT-FGSM, Strategy III achieves the best overall attack performance, reaching 35.9% and 37.5% when attacking IncRes_v2 and Xcep, respectively. When attacking with FT-MI-FGSM, Strategy V attains the best overall attack performance, reaching 57% and 53.3% when attacking Inv_v4 and IncRes_v2, respectively. When FT-SIM is used to attack IncRes_v2 and Xcep, Strategy III achieves the ASRs of 35.9% and 37.5%, which outperforms the other strategies. It can be seen that the overall performance of Strategy III is better and it performs better in the experiments combined with SIM, which is an input transformation-based method. Thus, we adopt Strategy III in the following experiments.

#### 4.1.4. Attack with Input Transformations

We test the ASRs of MI-FGSM, SIM, DIM, and Admix, respectively. Then we combine these methods with FTM as FT-MI-FGSM, FT-SIM, FT-DIM, and FT-Admix. Some adversarial examples are shown in Figure 3. We adopt Strategy III, set $m = 1$, set the magnitude of uniform distribution $r = 4$, and then use the generated adversarial examples to attack the four models. We compare the black-box ASRs of FT-MI-FGSM, FT-SIM, FT-DIM, and FT-Admix with MI-FGSM, SIM, DIM, and Admix in Tables 2–5. In the tables, the first columns are the local models, and the first rows are the target models. The values in the tables are the attack success rates (ASRs) on the target models using the adversarial examples generated from the local models. The higher ASRs are bolded.

When combined with MI-FGSM, the ASRs is increased by up to 9.4%, from 55% to 64.4% when attacking Xcep with Inc_v4. When FT-SIM is used to attack IncV3 with IncRes_v2, the ASR is improved from 62.6% to 75.1%, which outperforms the SIM by 12.5%. The adversarial examples generated by FT-DIM achieved about 55% ASR against all models. When FT-Admix is used to attack IncV3 with Xecp, the ASR reaches 72.2%.

According to the reported experimental results, it can be observed that FTM could improve the ASRs of adversarial examples generated by the SOTA black-box attack methods. It is confirmed that feature transformation can improve the transferability and robustness of adversarial examples.

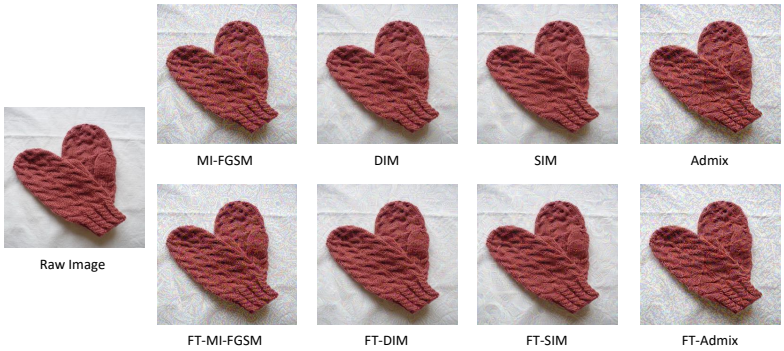


**Figure 3.** Adversarial examples generated by MI-FGSM, DIM, SIM, Admix, the proposed FT-MI-FGSM, FT-DIM, FT-SIM, and FT-Admix on the Inc_v3.

**Table 2.** The black-box ASRs of MI-FGSM and FT-MI-FGSM on ImageNet. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Inc_v3 | Inc_v4 | IncRes_v2 | Xcep |
|---|---|---|---|---|---|
| Inc_v3 | MI-FGSM | - | 51.3 | 49.6 | 53.0 |
| | FT-MI-FGSM | - | **54.9** | **51.6** | **57.8** |
| Inc_v4 | MI-FGSM | 56.0 | - | 48.5 | 55.0 |
| | FT-MI-FGSM | **58.9** | - | **53.1** | **64.4** |
| IncRes_v2 | MI-FGSM | 56.2 | 51.8 | - | 55.9 |
| | FT-MI-FGSM | **64.1** | **57.4** | - | **63.0** |
| Xcep | MI-FGSM | 51.4 | 50.8 | 45.3 | - |
| | FT-MI-FGSM | **54.4** | **55.0** | **48.7** | - |

**Table 3.** The black-box ASRs of SIM and FT-SIM on ImageNet. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Inc_v3 | Inc_v4 | IncRes_v2 | Xcep |
|---|---|---|---|---|---|
| Inc_v3 | SIM | - | 37.4 | 34.7 | 37.0 |
| | FT-SIM | - | **42.9** | **42.6** | **44.0** |
| Inc_v4 | SIM | 64.0 | - | 51.9 | 59.7 |
| | FT-SIM | **71.0** | - | **59.0** | **64.9** |
| IncRes_v2 | SIM | 62.6 | 52.8 | - | 55.2 |
| | FT-SIM | **75.1** | **63.4** | - | **65.2** |
| Xcep | SIM | 57.9 | 54.3 | 50.0 | - |
| | FT-SIM | **63.4** | **58.9** | **53.0** | - |

**Table 4.** The black-box ASRs of DIM and FT-DIM on ImageNet. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Inc_v3 | Inc_v4 | IncRes_v2 | Xcep |
|---|---|---|---|---|---|
| Inc_v3 | DIM | - | 59.5 | 55.3 | 56.3 |
| | FT-DIM | - | **61.8** | **58.3** | **60.4** |
| Inc_v4 | DIM | 59.0 | - | 52.0 | 61.7 |
| | FT-DIM | **63.4** | - | **56.5** | **66.6** |
| IncRes_v2 | DIM | 58.6 | 57.7 | - | 60.7 |
| | FT-DIM | **67.2** | **66.8** | - | **66.5** |
| Xcep | DIM | 57.3 | 64.3 | 55.6 | - |
| | FT-DIM | **61.8** | **69.1** | **58.2** | - |

**Table 5.** The black-box ASRs of Admix and FT-Admix on ImageNet. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Inc_v3 | Inc_v4 | IncRes_v2 | Xcep |
|---|---|---|---|---|---|
| Inc_v3 | Admix | - | 52.8 | 49.1 | 56.2 |
| | FT-Admix | - | **57.3** | **54.4** | **60.0** |
| Inc_v4 | Admix | 70.8 | - | 61.1 | 67.2 |
| | FT-Admix | **72.2** | - | **64.0** | **68.3** |
| IncRes_v2 | Admix | 64.1 | 57.4 | - | 60.5 |
| | FT-Admix | **66.0** | **58.7** | - | **60.4** |
| Xcep | Admix | 70.4 | 64.3 | 60.0 | - |
| | FT-Admix | **72.2** | **65.2** | **61.6** | - |

#### 4.1.5. Attack against Defense Method

In this section, we quantify the effectiveness of FTM against several defense methods, including random resizing and padding (RandP) [40], JPEG compression (JPEG) [39], randomized smoothing (RS) [38], and the rank-3 submission in the NIPS-2017 (NIPS-r3). RandP is the top-1 submission in the NIPS competition, which mitigates the effect of adversarial perturbations by randomized resizing and padding. JPEG is a defensive compression framework, which could rectify adversarial examples without reducing classification accuracy on benign data. RS constructs a "smoothed" classifier from an arbitrary base classifier, which is more adversarially robust. NIPS-r3 fuses multiple adversarial trained models and performs several input transformation at inference time.

We choose SIM as the comparison method and generate adversarial examples with Inc_v3. The average ASRs on Inc_v4, IncRes_v2, and Xcep are shown in Table 6. The ASRs are improved by a large margin of 9.5% on average. It validates that the adversarial examples generated by FTM are more robust to fool models with defense mechanisms.

**Table 6.** The black-box ASRs of SIM and FT-SIM on ImageNet against four defense methods. The adversarial examples are generated with Inc_v3. The values in the table are the average ASRs (%) on the Inc_v4, IncRes_v2, and Xcep. The higher ASRs are shown in bold.

| Attack Method | RandP | JPEG | RS | Nips-r3 |
|---|---|---|---|---|
| SIM | 30.3 | 32.7 | 25.2 | 31.6 |
| FT-SIM | **38.5** | **41.0** | **37.8** | **39.5** |

#### 4.1.6. Parameter Analysis

In this section, we perform additional analysis for the difference among different numbers of iterations $m$. The adversarial examples are generated by FT-DIM on Inc_v3. The number of iterations of feature transformation ranges from 1 to 9.

As shown in Figure 4, the average black-box ASR increases from 59.2% for 1 iteration to 62.7% for 3 iterations. As the number of iterations increases to 9, the success rate of the attack increases to 65.3%. It validates that the ASR of FTM increases as the number of iterations of feature transformation increases. The sensitivity of the attack success rate gradually decreases as the number of iterations increases. Since a higher number of iterations results in a larger computational overhead, the trade-off between effectiveness and overhead needs to be made according to the specific scenario.

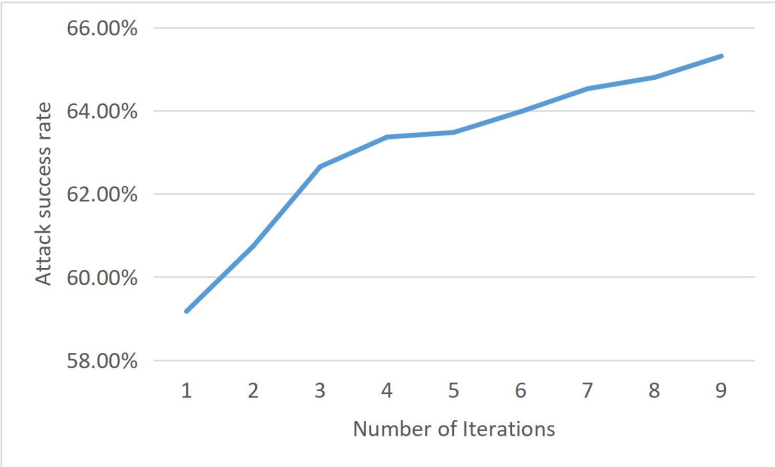


**Figure 4.** The black-box ASRs of FT-DIM attack with different number of iterations on ImageNet. The adversarial examples are generated on Inc_v3 and the ASRs are the average ASRs on Inc_v4, IncRes_v2, and Xcep.

### *4.2. Experiment on Cifar10*

Cifar10

To further demonstrate the effectiveness of our approach, we also conducted experiments on the Cifar10 [45] dataset. Cifar10 has 60,000 color images with 32 $\times$ 32 pixels and is divided into 10 categories. We select 1000 images belonging to the 10 categories from the test set, which are correctly classified by all the experimental models. We compare the effects of the FTM with the MI-FGSM, SIM and Admix using the ResNet [46] family of models. The maximum perturbation $\epsilon = 4$, number of attack iterations $T = 4$, and the step size $\alpha = 1$.

The experimental results for FT-MI-FGSM, FT-SIM, and FT-Admix are shown in Tables 7–9. The first columns are the local models and the first rows are the target models. It can be seen that our method improves the ASRs across all experiments. FT-MI-FGSM achieves 83.8% ASR, when attacking Res152 with Res50. FT-SIM improves the ASR of SIM from 66.6% to 73.9%, when attacking Res101 with Res152. FT-Admix boosts the ASR of Admix attack from 43.1% to 55.1%, when attacking Res101 with Res152.

The experimental results on Cifar10 validate that FTM is effective not only on large image dataset, but also on small image dataset. Moreover, FTM can significantly improve the transferability and robustness of the adversarial examples generated by the SOTA black-box attack methods.

**Table 7.** The black-box ASRs of MIM (MI-FGSM) and FT-MIM (FT-MI-FGSM) on Cifar10. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Res18 | Res34 | Res50 | Res101 | Res152 |
|---|---|---|---|---|---|---|
| Res18 | MIM | - | 78.3 | 68.7 | 67.3 | 71.1 |
| | FT-MIM | - | **78.8** | **69.2** | **70.5** | **73.4** |
| Res34 | MIM | 78.7 | - | 70.0 | 69.5 | 72.3 |
| | FT-MIM | **79.8** | - | **72.9** | **71.2** | **74.1** |
| Res50 | MIM | 76.5 | 76.8 | - | 80.2 | 82.5 |
| | FT-MIM | **77.8** | **78.1** | - | **82.4** | **83.8** |
| Res101 | MIM | 71.4 | 71.7 | 76.9 | - | 80.5 |
| | FT-MIM | **74.2** | **73.2** | **79.3** | - | **82.6** |
| Res152 | MIM | 75.2 | 73.4 | 76.8 | 81.0 | - |
| | FT-MIM | **76.8** | **74.9** | **78.7** | **82.0** | - |

**Table 8.** The black-box ASRs of SIM and FT-SIM on Cifar10. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Res18 | Res34 | Res50 | Res101 | Res152 |
|---|---|---|---|---|---|---|
| Res18 | SIM | - | 73.0 | 60.1 | 59.5 | 62.3 |
| | FT-SIM | - | **73.9** | **62.2** | **62.9** | **66.0** |
| Res34 | SIM | 74.9 | - | 60.2 | 60.9 | 63.3 |
| | FT-SIM | **76.2** | - | **61.5** | **62.8** | **63.4** |
| Res50 | SIM | 68.0 | 69.3 | - | 70.6 | 71.9 |
| | FT-SIM | **72.2** | **68.2** | - | **73.9** | **76.0** |
| Res101 | SIM | 69.2 | 67.7 | 71.0 | - | 73.9 |
| | FT-SIM | **71.5** | **69.9** | **71.9** | - | **75.9** |
| Res152 | SIM | 65.6 | 62.3 | 63.8 | 66.6 | - |
| | FT-SIM | **69.5** | **67.9** | **70.4** | **73.9** | - |

**Table 9.** The black-box ASRs of Admix and FT-Admix on Cifar10. The first column is the local model, and the first row is the target model. The values in the table are the ASRs (%) on the target models using the adversarial examples generated with the local models. The higher ASRs are shown in bold.

| Local Model | Attack Method | Res18 | Res34 | Res50 | Res101 | Res152 |
|---|---|---|---|---|---|---|
| Res18 | Admix | - | 49.0 | 41.9 | 42.5 | 45.0 |
| | FT-Admix | - | **56.4** | **47.3** | **50.4** | **51.9** |
| Res34 | Admix | 52.5 | - | 42.7 | 46.5 | 46.0 |
| | FT-Admix | **58.5** | - | **47.5** | **50.1** | **50.4** |
| Res50 | Admix | 48.6 | 43.9 | - | 44.9 | 47.4 |
| | FT-Admix | **56.1** | **50.7** | - | **53.9** | **54.4** |
| Res101 | Admix | 48.2 | 44.6 | 44.6 | - | 49.3 |
| | FT-Admix | **54.0** | **50.5** | **50.8** | - | **57.7** |
| Res152 | Admix | 45.3 | 40.6 | 39.5 | 43.1 | - |
| | FT-Admix | **55.0** | **51.6** | **50.2** | **55.1** | - |

## 5. Conclusions

We propose a novel feature transformation-based method (FTM), which effectively improves the transferability of adversarial examples. Five feature transformation strategies are proposed and the hyper-parameters of them are comprehensively analyzed. The experimental results on two benchmark datasets show that FTM can improve the transferability of the adversarial example significantly. It improves the ASRs of the SOTA methods by up to 12.5% on ImageNet. Our method can be combined with not only any gradient-based attack methods but also any neural networks that can extract features. However, the tuning of hyper-parameters is difficult, because different models and feature transformation strategies require a large number of experiments to choose the magnitude of uniform distribution. In the future, we will explore more feature transformation strategies to improve the transferability of adversarial examples while reducing the difficulty of tuning hyper-parameters.

**Author Contributions:** Conceptualization, H.-Q.X. and C.H.; methodology, H.-Q.X., C.H. and H.-F.Y.; software, H.-Q.X.; data curation, H.-Q.X.; resources, C.H.; writing—original draft preparation, H.-Q.X., C.H. and H.-F.Y.; project administration, C.H.; funding acquisition, C.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by the National Natural Science Foundation of China (Grant No. 62006097), in part by the Natural Science Foundation of Jiangsu Province (Grant No. BK20200593), in part by the China Postdoctoral Science Foundation (Grant No. 2021M701456), and in part by the Fundamental Research Funds for the Central Universities (Grant No. JUSRP121074).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The ImageNet and Cifar10 datasets were analyzed in this study. The ImageNet dataset can be found at https://image-net.org/ (accessed on 10 July 2022). Cifar10 dataset can be found at https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 10 July 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gou, J.; Yuan, X.; Du, L.; Xia, S.; Yi, Z. Hierarchical Graph Augmented Deep Collaborative Dictionary Learning for Classification. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]
2. Gou, J.; Sun, L.; Du, L.; Ma, H.; Xiong, T.; Ou, W.; Zhan, Y. A representation coefficient-based k-nearest centroid neighbor classifier. *Expert Syst. Appl.* **2022**, *194*, 116529. [CrossRef]
3. Gou, J.; He, X.; Lu, J.; Ma, H.; Ou, W.; Yuan, Y. A class-specific mean vector-based weighted competitive and collaborative representation method for classification. *Neural Netw.* **2022**, *150*, 12–27. [CrossRef] [PubMed]
4. Koo, J.H.; Cho, S.W.; Baek, N.R.; Lee, Y.W.; Park, K.R. A Survey on Face and Body Based Human Recognition Robust to Image Blurring and Low Illumination. *Mathematics* **2022**, *10*, 1522. [CrossRef]
5. Wang, T.; Ji, Z.; Yang, J.; Sun, Q.; Fu, P. Global Manifold Learning for Interactive Image Segmentation. *IEEE Trans. Multimed.* **2021**, *23*, 3239–3249. [CrossRef]
6. Cheng, C.; Wu, X.J.; Xu, T.; Chen, G. UNIFusion: A Lightweight Unified Image Fusion Network. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–14. [CrossRef]
7. Liu, Q.; Fan, J.; Song, H.; Chen, W.; Zhang, K. Visual Tracking via Nonlocal Similarity Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2826–2835. [CrossRef]
8. Zhu, X.F.; Wu, X.J.; Xu, T.; Feng, Z.H.; Kittler, J. Complementary Discriminative Correlation Filters Based on Collaborative Representation for Visual Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 557–568. [CrossRef]
9. Ma, C.; Rao, Y.; Lu, J.; Zhou, J. Structure-Preserving Image Super-Resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef]
10. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
11. Su, Y.; Zhang, Y.; Lu, T.; Yang, J.; Kong, H. Vanishing Point Constrained Lane Detection With a Stereo Camera. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2739–2744. [CrossRef]
12. Chen, Z.; Wu, X.J.; Yin, H.F.; Kittler, J. Robust Low-Rank Recovery with a Distance-Measure Structure for Face Recognition. In Proceedings of the PRICAI 2018: Trends in Artificial Intelligence, Nanjing, China, 28–31 August 2018; Geng, X., Kang, B.H., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 464–472.
13. Kortli, Y.; Jridi, M.; Al Falou, A.; Atri, M. Face Recognition Systems: A Survey. *Sensors* **2020**, *20*, 342. [CrossRef]

14. Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* **2020**, *9*, 1188. [CrossRef]
15. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
16. Li, J.; Ji, R.; Liu, H.; Hong, X.; Gao, Y.; Tian, Q. Universal Perturbation Attack Against Image Retrieval. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 4898–4907. [CrossRef]
17. Liu, H.; Ji, R.; Li, J.; Zhang, B.; Gao, Y.; Wu, Y.; Huang, F. Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 2941–2949. [CrossRef]
18. Li, H.; Zhou, S.; Yuan, W.; Li, J.; Leung, H. Adversarial-Example Attacks Toward Android Malware Detection System. *IEEE Syst. J.* **2020**, *14*, 653–656. [CrossRef]
19. Kwon, H.; Kim, Y.; Yoon, H.; Choi, D. Fooling a Neural Network in Military Environments: Random Untargeted Adversarial Example. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 456–461. [CrossRef]
20. Zhu, Z.A.; Lu, Y.Z.; Chiang, C.K. Generating Adversarial Examples By Makeup Attacks on Face Recognition. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 2516–2520. [CrossRef]
21. Wang, K.; Li, F.; Chen, C.M.; Hassan, M.M.; Long, J.; Kumar, N. Interpreting Adversarial Examples and Robustness for Deep Learning-Based Auto-Driving Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
22. Rana, K.; Madaan, R. Evaluating Effectiveness of Adversarial Examples on State of Art License Plate Recognition Models. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–3. [CrossRef]
23. Hu, C.; Wu, X.J.; Li, Z.Y. Generating adversarial examples with elastic-net regularized boundary equilibrium generative adversarial network. *Pattern Recognit. Lett.* **2020**, *140*, 281–287. [CrossRef]
24. Li, Z.; Feng, C.; Wu, M.; Yu, H.; Zheng, J.; Zhu, F. Adversarial robustness via attention transfer. *Pattern Recognit. Lett.* **2021**, *146*, 172–178. [CrossRef]
25. Agarwal, A.; Vatsa, M.; Singh, R.; Ratha, N. Cognitive data augmentation for adversarial defense via pixel masking. *Pattern Recognit. Lett.* **2021**, *146*, 244–251. [CrossRef]
26. Massoli, F.V.; Falchi, F.; Amato, G. Cross-resolution face recognition adversarial attacks. *Pattern Recognit. Lett.* **2020**, *140*, 222–229. [CrossRef]
27. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
28. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations Workshop, Toulon, France, 24–26 April 2017; pp. 1–14. [CrossRef]
29. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–24.
30. Lin, J.; Song, C.; He, K.; Wang, L.; Hopcroft, J.E. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–12.
31. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving Transferability of Adversarial Examples With Input Diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2725–2734. [CrossRef]
32. Dong, Y.; Pang, T.; Su, H.; Zhu, J. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4312–4321.
33. Wang, X.; He, X.; Wang, J.; He, K. Admix: Enhancing the Transferability of Adversarial Attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 16158–16167.
34. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting Adversarial Attacks with Momentum. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9185–9193. [CrossRef]
35. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Machine Learning at Scale. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–17.
36. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–28.
37. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; Mcdaniel, P. Ensemble Adversarial Training: Attacks and Defenses. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–22.
38. Cohen, J.M.; Rosenfeld, E.; Kolter, J.Z. Certified Adversarial Robustness via Randomized Smoothing. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 1–36.
39. Guo, C.; Rana, M.; Cisse, M.; van der Maaten, L. Countering Adversarial Images using Input Transformations. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–12.

40. Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; Yuille, A. Mitigating Adversarial Effects Through Randomization. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–16.
41. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
42. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
43. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
44. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–27 July 2017; pp. 1800–1807. [CrossRef]
45. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, USA, 2009; pp. 1–60.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.