


Article

Research and Application of the Median Filtering Method in Enhancing the Imperceptibility of Perturbations in Adversarial Examples

Yiming He, Yanhua Dong * and Hongyu Sun * 

College of Mathematics and Computer, Jilin Normal University, Siping 136000, China; yiming.he.work@mails.jlnu.edu.cn

* Correspondence: computerdyp@jlnu.edu.cn (Y.D.); hongyu@jlnu.edu.cn (H.S.)

Abstract: In the field of object detection, the adversarial attack method based on generative adversarial network efficiently generates adversarial examples, thereby significantly reducing time costs. However, this approach overlooks the imperceptibility of perturbations in adversarial examples, resulting in poor visual performance and insufficient invisibility of the generated adversarial examples. To further enhance the imperceptibility of perturbations in adversarial examples, a method utilizing median filtering is proposed to address these generated perturbations. Experimental evaluations were conducted on the Pascal VOC dataset. The results demonstrate that, compared to the original image, there is an increase of at least 17.2% in the structural similarity index (SSIM) for generated adversarial examples. Additionally, the peak signal-to-noise ratio (PSNR) increases by at least 27.5%, while learned perceptual image patch similarity (LPIPS) decreases by at least 84.6%. These findings indicate that the perturbations in generated adversarial examples are more difficult to detect, with significantly improved imperceptibility and closer resemblance to the original image without compromising their high aggressiveness.

Keywords: generative adversarial network; adversarial examples; imperceptibility; median filtering



Citation: He, Y.; Dong, Y.; Sun, H. Research and Application of the Median Filtering Method in Enhancing the Imperceptibility of Perturbations in Adversarial Examples. *Electronics* **2024**, *13*, 2458. <https://doi.org/10.3390/electronics13132458>

Academic Editor: Bahman Javadi

Received: 9 May 2024

Revised: 9 June 2024

Accepted: 17 June 2024

Published: 23 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since the emergence of deep learning, it has undergone rapid development and demonstrated significant potential for application in various fields, including large language models for natural language processing [1–3] and computer vision. In particular, remarkable achievements have been made in tasks such as image classification [4], object detection [5], and semantic segmentation [6]. Deep neural networks (DNNs) are the foundation of deep learning and have become an integral part of these fields. However, the inherent fragility of deep learning means that DNN may produce wrong recognition results with high confidence when facing adversarial examples [7]. Adversarial examples are images formed by adding small perturbations generated in a specific way to the original image [8]. Although such perturbations are small, they are enough to bias the judgment of DNN, thus posing a serious challenge to the security of DNN.

As one of the classic core tasks in the field of computer vision, object detection has a decisive impact on the performance of many computer vision tasks and their applications [9]. Compared with the early object detection technology based on hand-designed features, object detection technology based on deep learning has made more significant progress in efficiency and accuracy [10]. However, because it is based on DNN, it is also inevitable that the vulnerability of DNN to adversarial examples is inherited, which poses a direct threat to the security of the object detection field. In particular, object detection technology based on deep learning has been widely used in all fields of society, especially in key security fields such as automatic driving [11–13], security monitoring [14] and face recognition [15,16].

The safety of object detection technology is directly related to the reliability of the system and social stability, as well as the safety of people's lives.

The study of adversarial examples for object detection can not only reveal the potential weaknesses of object detection algorithms, but is also of great significance in promoting the development of safer and more robust object detection technology. In this research field, most existing adversarial attack research focuses on white-box attacks. According to the generation method of adversarial perturbation, these can be divided into attack methods based on optimization iteration and attack methods based on a generative adversarial network (GAN) [17]. The former method is characterized by its slow speed, inefficiency, requirement for multiple iterations, and high resource consumption in generating adversarial examples. On the other hand, the latter method significantly reduces the time required to generate adversarial examples by training a generator network to produce adversarial perturbations. However, due to limitations in the structure performance and optimization method of the generator, controlling the amplitude and area of generated adversarial perturbations becomes challenging, resulting in numerous redundant perturbations. Consequently, these generated adversarial examples exhibit low imperceptibility with poor visibility that can be easily detected by human observation [18]. Therefore, enhancing the imperceptibility of adversarial perturbations becomes crucial for making them more covert and difficult to perceive by humans. In this paper, we propose a median-filtering-based method to smooth out the generator-generated adversarial perturbations and improve their imperceptibility through noise filtering. This method enhances the visual effect of adversarial examples by making them resemble original images more closely while minimizing detectability. It not only explores a novel approach to enhance the generation process of adversarial examples through filtering techniques but also introduces fresh insights for adversarial defense, fostering the continuous advancement of defensive technologies against diverse forms of adversarial attacks.

2. Related Works

2.1. Object Detection

The object detection algorithms based on deep learning can be categorized into one-stage and two-stage algorithms, depending on whether there is a stage for generating region proposals. Two-stage algorithms are represented by R-CNN [19], Fast R-CNN [20], and Faster R-CNN [21]. Figure 1 presents the basic structure of Faster R-CNN. These algorithms include two steps of generating and filtering region proposals, and classifying the region proposals and bounding box regression, which has high accuracy but requires more computing resources. One-stage algorithms are represented by YOLO series [22–24] and SSD [25], which directly predict the category and location of targets through anchor boxes, enabling a fast detection speed but slightly lower accuracy. However, previous algorithms relying on anchor boxes have limitations. Consequently, anchor-free algorithms represented by CenterNet [26] and FCOS [27] have emerged to significantly improve detection efficiency.

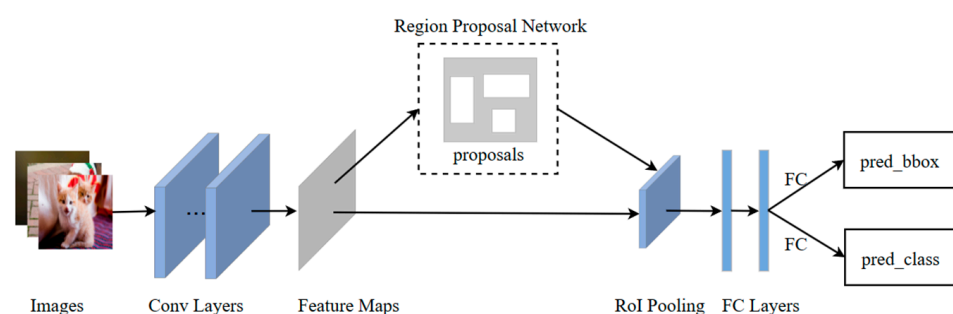


Figure 1. The basic structure of Faster R-CNN.

2.2. Adversarial Attack Methods for Object Detection

2.2.1. Attack Methods Based on Optimization Iteration

At present, most of the adversarial attacks for object detection are based on optimization iterations, which optimize the adversarial examples by designing an appropriate loss function and using gradient propagation backwards. This process does not require the additional training of other networks, and the method used to generate adversarial examples is relatively simple.

In 2017, the dense adversary generation (DAG), proposed by Xie et al. [28], proved for the first time that adversarial examples can also be applied to object detection tasks that seriously affect the security and robustness of the object detector. DAG mainly attacks the correctly predicted region proposals. For each region proposal, an error label different than the true class label is randomly assigned, and the iterative attack is carried out continuously, so that the predicted label of the object gradually deviates from the correct label, and the iteration moves toward the direction of low class confidence, until all region proposals are incorrectly predicted or when the set number of iterations is reached. However, DAG consumes a lot of resources and can only be attacked against two-stage object detectors. Robust adversarial perturbation (RAP) [29] is also an attack method for the two-stage object detector, but it reduces the confidence of the region proposal by attacking the RPN network so that the object detector classifies the object in the image as the background, thus failing to identify the object. In addition, the region proposals that can still be predicted correctly are interfered with, so that the positioning is wrong. In order to enhance the generalization of the generated adversarial examples, a set of iterative TOG [30] attack methods was proposed that can be applied to both one-stage and two-stage object detectors. These methods are not designed specifically for the unique structure of each object detector, but rather from the perspective of multi-task object detection, encompassing attacks on object classification errors, object disappearance, and more. Various attack techniques can generate adversarial examples for different object detectors. However, it should be noted that the iterative generation speed of adversarial examples is relatively slow, efficiency is comparatively low, and resource consumption is high.

2.2.2. Attack Methods Based on Generative Adversarial Network

The attack method based on optimization iteration is characterized by slow speed, relatively low efficiency, and significant resource consumption in generating adversarial examples. To address these limitations, a real-time adversarial example generation method based on the generative adversarial network (GAN) was proposed. The specific process is illustrated in Figure 2. The essence of the GAN-based attack method lies in learning the distribution of adversarial noise and utilizing the generator to directly generate such noise. Once the generator is trained stably, the distribution of the generated adversarial examples can be aligned more closely with that of the original image, thereby enabling the acquisition of higher-quality adversarial examples [31]. Additionally, due to solely requiring forward propagation for generating corresponding noise from an image, the generation speed is enhanced, and efficiency is improved.

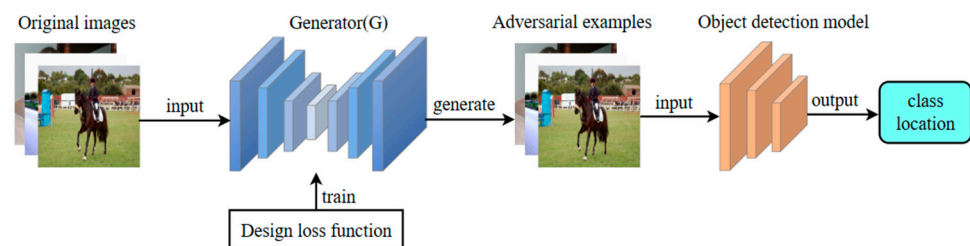


Figure 2. The process of generating adversarial examples by the generator.

The unified and efficient adversary (UEA) proposed by Wei et al. [32] introduces an attack method capable of targeting both one-stage and two-stage object detection algorithms. To train the generator in generating adversarial examples, a composite loss function is devised, combining GAN loss with high-level classification loss, low-level feature loss, and multi-scale attention feature loss. In terms of the time to generate a single adversarial example, the generation speed of UEA is 930 times that of DAG, and it has good transfer. Fast attack (FA), proposed by Li et al. [33], is also based on GAN, combining GAN loss with classification and position loss to rapidly generate adversarial examples. Furthermore, Deng et al. combined style transfer with GAN to design an attack algorithm that enhances the aggressiveness and transferability of adversarial examples through the application of style transfer techniques [34].

However, limited by the structure performance and optimization method of the generator network, compared with the original image, the adversarial examples generated by the GAN method have obvious noise, and the visual effect of the adversarial examples is still not ideal. The adversarial examples generated by the three methods are shown in Figure 3. It can be seen that the imperceptibility of the perturbation of adversarial examples is very low, which affects subjective visual perception, and they are easily distinguished by the naked eye.

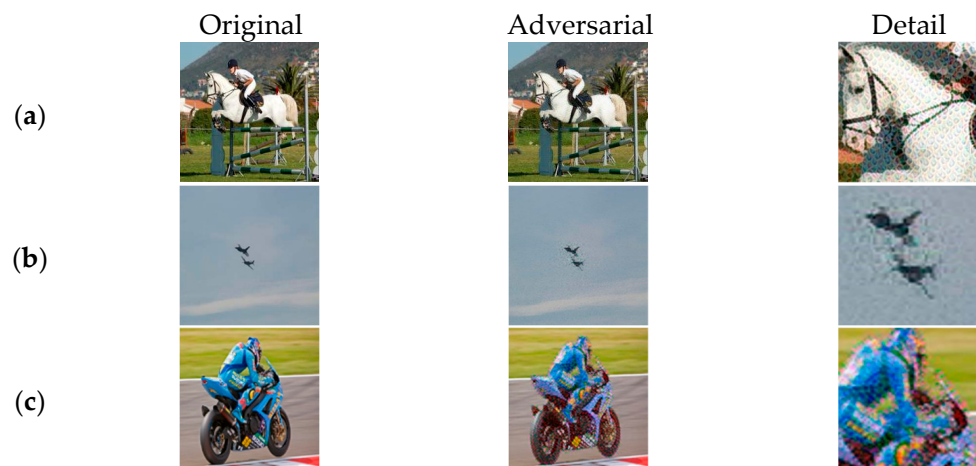


Figure 3. (a) Adversarial example generated by UEA; (b) adversarial example generated by FA; and (c) adversarial example generated using GAN and style transfer. The evident perturbation is clearly observable in the adversarial example.

3. Method

3.1. Improvements Based on Median Filtering

3.1.1. Median Filtering

The adversarial examples generated based on the generative adversarial network do not limit the amplitude and area of the perturbation well; therefore, the perturbation is too large and there is obvious red noise. Red noise is the pixel in the image with dramatic intensity changes, which belongs to high-frequency noise. This makes adversarial examples easily recognizable by human eyes. In order to further improve the imperceptibility of adversarial example perturbations, this paper introduced median filtering to process the adversarial perturbations generated by the generator. The attack effect of low-frequency perturbations on the model has been demonstrated to surpass that of high-frequency perturbations in certain studies [35]. By filtering out high-frequency information from adversarial examples, their attack capability can still be maintained while enhancing their robustness [36].

The median filter is a widely used image-processing technique that belongs to the category of nonlinear filters. It achieves image smoothing by calculating the median value within a sliding window centered at each pixel position. By effectively eliminating high-

frequency noise, such as isolated bright and dark points, this method preserves edge information while minimizing significant blurring. For a given position (x, y) in the image, the intensity of the pixel after median filtering can be expressed as follows:

$$I(x, y) = \text{median}(I(x', y')), \quad (1)$$

The coordinates (x', y') represent the pixel locations in the domain, and *median* refers to selecting the output pixel value as the median number after sorting all pixels in the filter by size. By applying median filtering, high-frequency components of the image can be eliminated while concentrating perturbations on low-frequency components. This reduces sensitivity of human visual perception to image changes, thereby enhancing subjective visual effects of adversarial examples and improving imperceptibility of adversarial perturbations.

3.1.2. Network Framework

The specific network framework depicted in Figure 4 enhances the imperceptibility of GAN-based generated adversarial perturbations through the utilization of median filtering. The overall structure can be categorized into three components: generating adversarial perturbations and subjecting them to median filtering, producing adversarial examples by combining a weight mask and optimizing the generated adversarial examples.

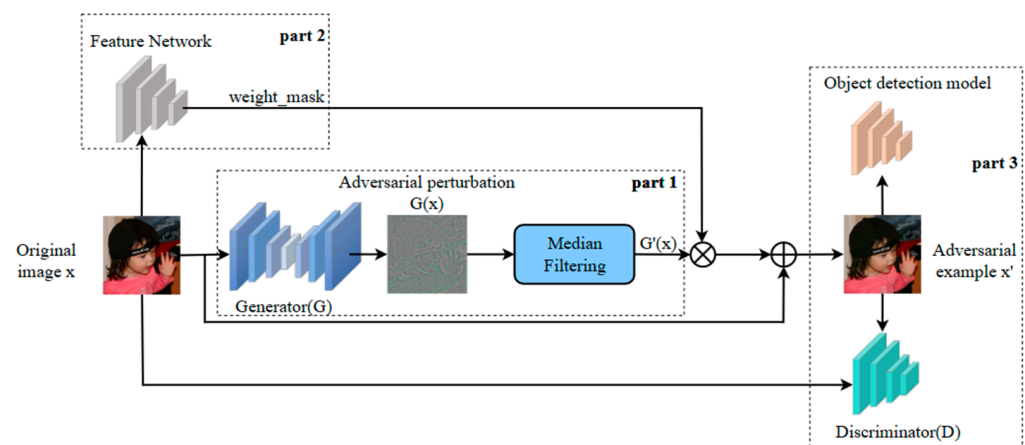


Figure 4. Improved framework for adversarial example perturbation imperceptibility based on median filtering. The various components are demarcated by dashed lines.

The first part involves generating the adversarial perturbation. GAN primarily consists of two components: the generator (G) and discriminator (D). Initially, GAN learns the mapping from the original image x to generate the adversarial perturbation, which is then produced by the generator G as $G(x)$. The high-frequency noise is filtered out, resulting in a new adversarial perturbation denoted as $G'(x)$. In the second part, an adversarial example is generated, and a weight mask is obtained through feature extraction network. This mask is multiplied with $G'(x)$ to constrain both the range and amplitude of the adversarial perturbation. Subsequently, it is added to the original image to form an adversarial example x' . In the third part, these generated adversarial examples are input into both discriminator D and the object detection model for optimization. The role of discriminator D lies in ensuring that these generated adversarial examples resemble their corresponding original images as closely as possible by calculating the GAN loss. Additionally, perturbation loss is computed using the L_2 distance between each adversarial example and its respective original image to further control the size of the generated perturbations. Simultaneously, the quality of these generated adversarial examples improves while they are fed into the object detector to obtain the object detection results. Misclassification loss occurs when comparing these results with real labels associated with the original images; this aids in the improved training of adversarial examples for deceiving object detection models.

Feature loss measures the discrepancy between the generated adversarial examples and their corresponding original images so that the detection results deviate from the true values, thereby enhancing aggressiveness of the adversarial example.

3.2. Loss Functions

In the improved method of perturbation imperceptibility of adversarial examples based on median filtering, the generation of adversarial examples is controlled by the following four loss functions, which are defined as follows:

$$L_{loss} = L_{GAN} + aL_{misclass} + bL_{perturb} + cL_{feature}, \quad (2)$$

where, a , b and c are the weight coefficients of each loss function.

In GAN, the training of the generator is guided by measuring the probability that the images generated by the generator are judged as real images by the discriminator, so that the generator can be continuously improved to generate more realistic images, and the discriminator is also prompted to improve the recognition ability of the generated images. GAN loss is defined as follows:

$$L_{GAN}(G, D) = E_x[\log D(x)] + E_x[\log(1 - D(G(x)))], \quad (3)$$

where x is the input image, G is the generator, and D is the discriminator.

In order to be able to attack both object detectors, the loss function proposed by the DAG method is added to achieve a better misclassification effect by assigning an error label to each generated region proposal, which is defined as follows:

$$L_{misclass}(G) = E_x \left[\sum_{n=1}^N [f_{l_n}(X, t_n) - f_{l'_n}(X, t_n)] \right], \quad (4)$$

where X is the extracted feature map, t_n is the n -th region proposal obtained from RPN, l_n is the true label corresponding to t_n , and l'_n is the random error label $f_{l_n}(X, t_n)$ represents the classification score vector on the n -th region proposal.

At the same time, L_2 distance is used to measure the difference between the original image and the adversarial example to limit the amplitude of the perturbation, which is defined as follows:

$$L_{perturb}(G) = E_x[\|x - G(x)\|_2], \quad (5)$$

In order to increase the transferability of adversarial examples, multi-scale attention feature loss is introduced [32], and adversarial examples that are offensive to the object detector can be generated by attacking the feature extraction network, which is defined as follows:

$$L_{feature}(G) = E_x \left[\sum_{m=1}^M \|W_m \circ (X_m - Y_m)\|_2 \right], \quad (6)$$

where X_m represents the feature map extracted by the m -th layer network of the object detector, Y_m is a random predefined feature map fixed during training, W_m is the attention weight calculated based on the region proposals of the RPN, and “ \circ ” represents the Hadamard product between the two matrices. This feature loss makes the perturbation more focused on the object, which can obtain better transferability.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

In this paper, two commonly used object detection datasets, Pascal VOC and MS COCO, were selected to verify the improvement effect of median filtering on the perturbation imperceptibility of the generated adversarial examples. Pascal VOC contains 4 broad classes called vehicles, household, animals, and person, for a total of 20 common object categories. The MS COCO dataset contains more than 300,000 images, providing 80 different classes of objects to be detected, covering a variety of common objects in real-world scenes.

It contains more objects per image and more small objects, and these are more complex to identify.

4.1.2. Object Model

The object model is the model that uses adversarial examples to attack, which can also be called the victim model. In white-box attacks, the model used to generate adversarial examples is usually the same as the object model. We chose the representative two-stage algorithm Faster R-CNN for the experiments and used VGG19 as the backbone network of Faster R-CNN to extract image features. This classic network architecture has achieved good performance in multiple image classification and object detection tasks. More specifically, the experiments mainly train Faster R-CNN on the Pascal VOC2007 training set and test it on the Pascal VOC2007 test set. The mAP (mean Average Precision) after training can reach 69.21%.

4.1.3. Experiment Details

The experiment employed the PyTorch1.2.0 deep learning framework and utilized the NVIDIA Tesla V100S 32G graphics card. However, only 16G of video memory was allocated for computation through the application of virtualization technology during usage. The generator and discriminator in the generative adversarial network employed in this experiment are based on the AdvGAN [37] network architecture. The optimizer utilized was Adam (Adaptive Moment Estimation). The initial learning rates for the generator and discriminator were set to 0.0001 and 0.0002, respectively, with each round consisting of 20 iterations. Additionally, considering variations in image sizes, a batch size of 1 was set.

4.2. Metrics

4.2.1. Visual Quality Evaluation Metrics

In order to evaluate the improvement effect of adversarial example perturbation imperceptibility, three metrics including structural similarity index (SSIM), peak signal-to-noise ratio (PSNR) and learned perceptual image patch similarity (LPIPS) were used as the standard to evaluate the visual quality of the image.

(1) SSIM

SSIM can determine the degree of visual similarity between two images, taking into account the brightness, contrast and structural information of the images. The closer the SSIM value of two images is to 1, the higher the similarity between them is. Conversely, the closer they are to 0, the less similar they are;

(2) PSNR

PSNR is a sensitivity evaluation based on the error between pixels, and the larger the value is, the smaller the distortion is. However, because it does not take into account the human visual sensitivity to different characteristics, the evaluation results may be inconsistent with the subjective perception of humans;

(3) LPIPS [38]

LPIPS considers that two images may be perceived as different by the human eye even if they are very close at the pixel level; therefore, it uses a deep learning model to extract image features and then calculates the distance between these features to evaluate the perceptual similarity between images. Compared with PSNR and SSIM, LPIPS not only focuses on the similarity of image content, but also pays more attention to the factors of human eye perception, which makes it better reflect the subjective feelings of humans when observing images. The smaller the value of LPIPS, the smaller the perceptual difference between the images and the greater the similarity between images.

4.2.2. Attack Evaluation Metrics

The mAP is an important measure of object detection performance and is the mean of the average precision of all classes. It is specifically expressed as follows:

$$mAP = \frac{\sum_{k=1}^n AP_k}{n}, \quad (7)$$

where n is the number of classes and AP_k is the average precision of the k -th class. Therefore, we use the degree of mAP decline after the attack to evaluate the aggressiveness of the generated adversarial examples, that is, the attack success rate (ASR), which is defined as follows:

$$ASR = 1 - \frac{mAP_{adv}}{mAP_{clean}} \quad (8)$$

where, mAP_{clean} is the detected mAP value when the original image is input, and mAP_{adv} is the detected mAP value after using the adversarial example attack. ASR values range between 0 and 1, with higher values indicating more powerful attacks.

4.3. Experiment and Result Analysis

4.3.1. Visual Perception Experiment

(1) Subjective Visual Effect

As shown in Figure 5, the first row shows the subjective visual effects of adversarial examples processed by median filter kernels of different sizes (the filter kernel sizes are 3, 5, 7, 9, and 11, respectively). It can be seen that the imperceptibility of adversarial perturbations has been greatly improved. In order to observe and compare the details of the perturbation more closely, the second row shows a local magnification of the corresponding image. It can be clearly seen from the figure that, with the continuous increase of the filter kernel size, the subjective visual effect of the adversarial example becomes better and better, and the imperceptibility of the adversarial perturbation is gradually improved. Only with the naked eye, the difference between the adversarial example and the original image cannot be distinguished, and the image quality of the adversarial example is significantly improved.



Figure 5. The adversarial examples are obtained by applying various median filtering kernels to process the adversarial perturbations, where K represents the size of the filter kernel.

(2) Objective Evaluation Metrics

In order to further verify the improvement effect of adversarial perturbation imperceptibility, SSIM, PSNR and LPIPS values of the original image and the adversarial example processed by different median filtering kernels were calculated, respectively. The results are shown in Table 1. It can be seen from the table that with the continuous increase in the median filter kernel size, the SSIM can reach more than 0.99, which is basically the same as the structure of the original image. The PSNR is also higher and higher, indicating that the image quality is becoming better and better. The value of the LPIPS is closer and closer to 0, which means that the adversarial examples are more similar to the original image. All these metrics show that the adversarial examples processed by median filtering and the original image are more and more similar, and the imperceptibility of adversarial perturbations has been significantly improved.

Table 1. Results of visual quality evaluation metrics for adversarial examples processed with different median filter kernels, where K represents the filter kernel size.

	SSIM	PSNR	LPIPS
Adversarial example	0.827	29.002	0.363
K = 3	0.969	36.966	0.056
K = 5	0.982	39.748	0.024
K = 7	0.987	41.457	0.014
K = 9	0.990	42.601	0.010
K = 11	0.991	43.407	0.008

4.3.2. Comparison Experiments of Different Filters

The experiment also compared the effect of adversarial examples processed by mean filtering, Gaussian filtering and bilateral filtering. The mean filter is a linear filter whose output pixel value is the average of the pixels in its surrounding neighborhood. The Gaussian filter is also a linear smoothing filter, which mainly uses a Gaussian function to obtain a weighted average of the image to achieve the purpose of removing noise and retaining image details. The bilateral filter is a nonlinear filter based on spatial domain and gray domain. It takes into account the spatial distance and the similarity between pixel values and preserves the edge information of the image while removing the noise.

The results of the adversarial examples for two images are presented in Figure 6 after undergoing four different filtering processes with a filter kernel size of 3. To enhance visibility of the changes in the adversarial perturbation, local content of the adversarial examples is displayed in the image. Additionally, Table 2 lists the corresponding visual quality evaluation metrics for these adversarial examples. It is evident from the graph that median filtering renders the adversarial perturbation as more concealed and improves the visual quality of the adversarial example, whereas perceptible perturbations still persist in the other three filtering processes.

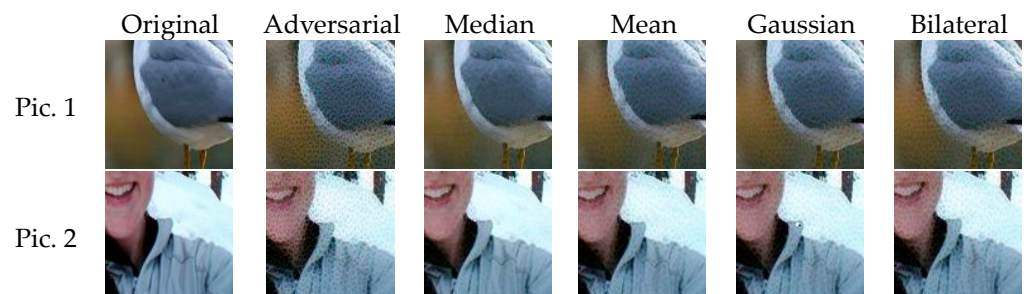


Figure 6. Adversarial examples obtained from two images after four different filtering processes.

Table 2. Results of visual quality evaluation metrics for adversarial examples obtained from two images after four different filtering processes.

		Median	Mean	Gaussian	Bilateral
Pic. 1	SSIM	0.972	0.941	0.925	0.912
	PSNR	38.117	36.053	35.043	34.370
	LPIPS	0.075	0.139	0.178	0.198
Pic. 2	SSIM	0.977	0.949	0.938	0.930
	PSNR	35.902	34.042	33.174	32.633
	LPIPS	0.047	0.117	0.156	0.173

The visual quality evaluation metric results of the adversarial examples processed by the four filters are presented in Figure 7 as the filter kernel size continues to increase. It is evident from the figure that median filtering yields the highest PSNR and SSIM, along with the lowest LPIPS, indicating that it renders the corresponding adversarial example as

more similar to the original image in terms of brightness, contrast, structure, and human visual effect. Furthermore, to further validate median filtering's superior effectiveness in enhancing the imperceptibility of adversarial perturbations, these four filters were combined pairwise for reprocessing generated adversarial perturbations at a filter kernel size of 3. The outcomes are displayed in Table 3. Notably, combining median filtering with other filters demonstrates better performance according to the data presented in this table. These findings substantiate that adversarial examples processed by median filtering retain more structural information from images and exhibit a closer resemblance to their original counterparts visually. Conversely, employing other filters may result in greater loss of image details. Consequently, considering subjective visual effects of adversarial examples alongside metrics such as SSIM, PSNR and LPIPS leads us to conclude that median filtering is more suitable for improving the imperceptibility of adversarial perturbations.

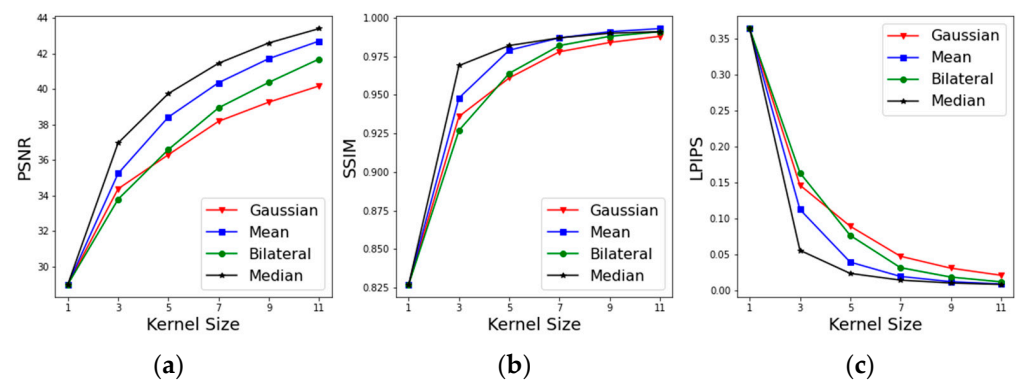


Figure 7. The visual quality evaluation metrics results of the adversarial examples processed by four filters with different size filter kernels are presented: (a) PSNR; (b) SSIM; and (c) LPIPS.

Table 3. Comparison of results for adversarial examples processed by different combinations of filters.

	SSIM	PSNR	LPIPS
Median + Mean	0.980	38.284	0.029
Median + Gaussian	0.979	38.068	0.034
Median + Bilateral	0.978	37.940	0.037
Mean + Gaussian	0.966	36.842	0.075
Mean + Bilateral	0.964	36.612	0.080
Gaussian + Bilateral	0.958	36.027	0.096

The ASR values of the adversarial examples processed by different median filtering kernels are listed in Table 4. With the progressive increase in filtering kernels, there is a decrease in ASR. However, it still maintains a high attack success rate. Furthermore, the experiment also reveals that, for similar objective quality evaluation metrics, a single median filtering process can sustain a higher attack success rate. Additionally, the experiment compared the AP values of the object detector for the original image, adversarial example, and adversarial example processed with a median filter using a kernel size of 3. The results are shown in Figure 8. It can be seen that, after applying median filter processing to the original adversarial example, significant improvements are achieved in AP values for the bird, cat, dog and horse classes. Consequently, the overall mAP value is improved, and the ASR value is decreased.

In fact, the adversarial examples processed by filter kernels of different sizes all have the problem that the AP values of these four categories change greatly before and after filtering, which is due to the different robustness and sensitivity of the object detector to different categories of objects. The object detector used in the experiment is more sensitive to changes in the details of these categories. After a filtering operation to remove some noise, the features of these categories became easier to recognize, resulting in a large

increase in AP value. It is less sensitive to other categories, so the effect of the filtering operation is relatively small. The AP values resulting in other categories did not change as much as these four classes. In addition, the median filtering operation not only effectively eliminates high-frequency noise, but also mitigates a certain amount of low-frequency noise. The reduction in low-frequency noise has an impact on the potency of adversarial examples, leading to a decrease in attack success rates. However, overall, adversarial examples processed by median filtering still maintain a high attack success rate and exhibit satisfactory visual clarity.

Table 4. Results for ASR with different median filtering kernels.

	ASR
Adversarial example	0.885
K = 3	0.780
K = 5	0.764
K = 7	0.761
K = 9	0.760
K = 11	0.753

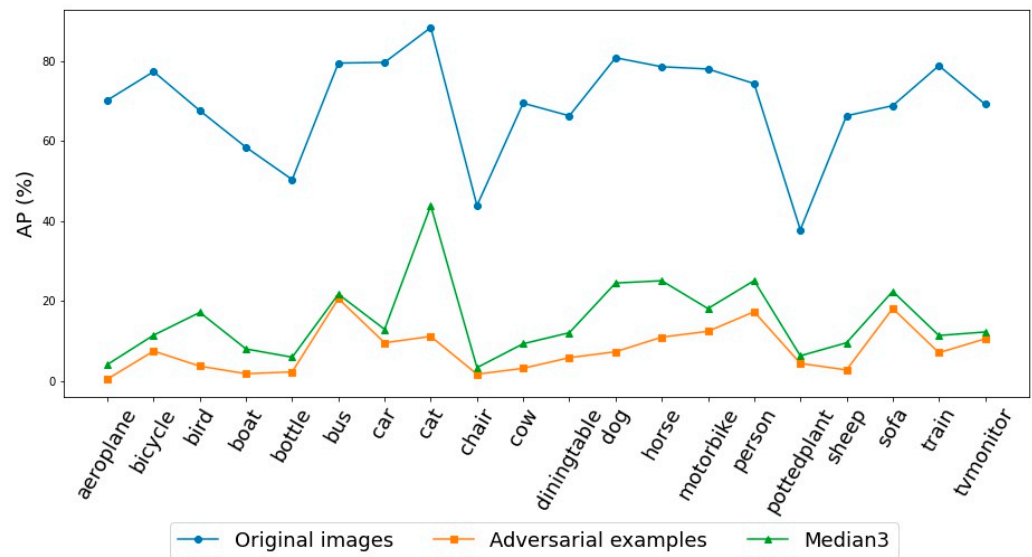


Figure 8. Comparison of the AP values of the original image, the adversarial example, and the adversarial example after filtering with the median kernel size of 3.

4.3.3. Comparative Experiments on COCO Dataset

The COCO dataset was retrained to generate the object detector, and the perturbation was also processed using the median filter. The same experiment was carried out as before, and the results are shown in Figure 9. It can be seen that, whether it is SSIM, PSNR, or LPIPS, the results are better on Pascal VOC dataset than on COCO dataset.

First, the image complexity differs between the two datasets. The COCO dataset encompasses a greater number of objects and more intricate scenes compared to the VOC dataset. Second, disparities exist in the trained models. The model trained on VOC exhibits a higher propensity for acquiring effective features due to its relatively simplistic nature, making it easier to handle perturbations. Conversely, the model trained on COCO may necessitate increased complexity or robustness to achieve an equivalent level of deception owing to the presence of more intricate data. Consequently, adversarial perturbations may not be as readily attenuated through median filtering as that observed in VOC. Lastly, the impact of median filtering varies while processing images with different complexities, whereas median filtering might prove more effective in reducing noise without a significant loss of detail in VOC’s simpler backgrounds and fewer targets, it may not yield comparable

results within COCO's complex scenes. This discrepancy can subsequently affect the final visual quality evaluation metrics. All these factors contribute to the limited discernible effect of median filtering on adversarial perturbations generated within the COCO dataset compared to that observed within VOC.

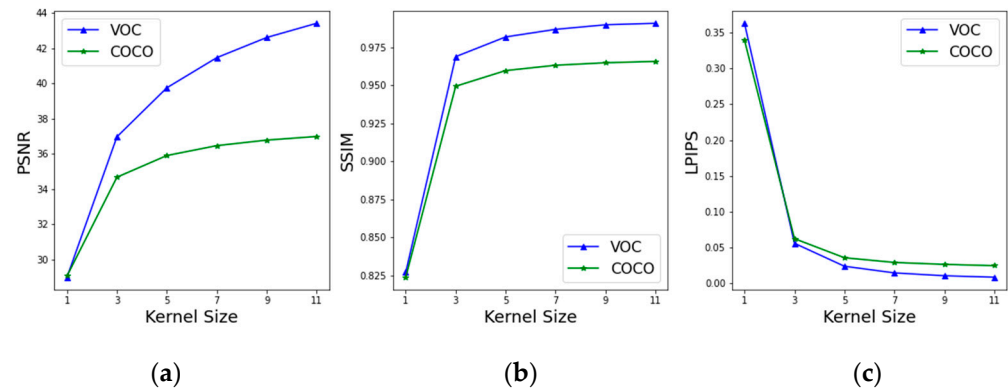


Figure 9. Results of visual quality evaluation metrics for different datasets: (a) PSNR; (b) SSIM; and (c) LPIPS.

In general, regardless of the Pascal VOC or COCO datasets, the perturbation imperceptibility of the adversarial examples processed by median filtering is still significantly improved, and the visual effect is also better.

5. Conclusions

Aiming to address the issue of excessive perturbation and poor visual visibility in adversarial examples generated using generative adversarial networks, this paper proposed an enhanced method based on median filtering. The effects of single median filtering, mean filtering, Gaussian filtering, bilateral filtering, and combined filtering were compared. Initially, the adversarial perturbations produced by the generator were smoothed through high-frequency noise filtration. Subsequently, these processed perturbations were overlain onto the original image to generate the adversarial examples. Experimental evaluations conducted on Pascal VOC and COCO datasets demonstrated that median filtering effectively enhances imperceptibility of the adversarial perturbations while improving the subjective visual visibility of the adversarial example. These adversarial examples closely resemble the original image and are challenging to distinguish with the naked eye.

The adversarial examples improved through median filtering align more closely with human visual perception, concealing the adversarial perturbations to a greater extent. This research explores advancements in generating adversarial examples using filtering technology, offering novel insights for adversarial defense, and fostering the continuous development of defensive techniques against various forms of attacks. However, due to the object detector's sensitivity towards category-specific details and the potential removal of low-frequency noise by the median filter, there is a reduction in the attack success rate of these adversarial examples while still maintaining a relatively high level of effectiveness.

Therefore, subsequent research will focus on devising an adversarial attack methodology that not only ensures the optimal visual visibility of the generated adversarial examples but also enhances their attack success rate. In future endeavors, we will contemplate incorporating an adaptive filtering module that dynamically adjusts filtering parameters based on image content and the characteristics of the adversarial perturbation to achieve more precise processing of such perturbations. For instance, in visually sensitive regions, filtering intensity can be reduced to preserve intricate details, while concentrated areas of adversarial perturbation may warrant increased filtering intensity for improved imperceptibility. Additionally, employing multi-objective optimization techniques can facilitate finding an optimal trade-off between concealment, attack success rate, and image quality in generating offensive yet imperceptible adversarial examples.

Author Contributions: Conceptualization: Y.H.; software: Y.H.; formal analysis: Y.H.; investigation: Y.H.; data curation: Y.H.; writing—original draft preparation: Y.H.; writing—review and editing: Y.D. and H.S.; visualization: Y.H.; supervision: Y.D. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Jilin Province Science and Technology Development Plan Project—Youth Growth Science and Technology Plan Project (20220508038RC), New Generation Information Technology Innovation Project of China University Industry, University and Research Innovation Fund (2022IT096), Jilin Province Innovation and Entrepreneurship Talent Project (2023QN31), Natural Science Foundation of Jilin Province (No.YDZJ202301ZYTS157, 20240304097SF), and Innovation Project of Jilin Provincial Development and Reform Commission (2021C038-7).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yenduri, G.; Ramalingam, M.; Selvi, G.C.; Supriya, Y.; Srivastava, G.; Maddikunta, P.K.R.; Raj, G.D.; Jhaveri, R.H.; Prabadevi, B.; Wang, W.; et al. GPT (Generative Pre-Trained Transformer)—A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *IEEE Access* **2024**, *12*, 54608–54649. [[CrossRef](#)]
2. Khowaja, S.A.; Khuwaja, P.; Dev, K.; Wang, W.; Nkenyereye, L. ChatGPT Needs SPADE (Sustainability, PrivAcy, Digital divide, and Ethics) Evaluation: A Review. *Cogn. Comput.* **2024**, 1–23. [[CrossRef](#)]
3. Wu, X.; Duan, R.; Ni, J. Unveiling Security, Privacy, and Ethical Concerns of ChatGPT. *J. Inf. Intell.* **2023**, *2*, 102–115. [[CrossRef](#)]
4. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**. arXiv:1409.1556. [[CrossRef](#)]
5. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
6. Yuan, X.; Shi, J.; Gu, L. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Syst. Appl.* **2021**, *169*, 114417. [[CrossRef](#)]
7. Nguyen, A.; Yosinski, J.; Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [[CrossRef](#)]
8. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**. arXiv:1312.6199. [[CrossRef](#)]
9. Cao, J.; Li, Y.; Sun, H.; Xie, J.; Huang, K.; Pang, Y. A survey on deep learning based visual object detection. *J. Image Graph.* **2022**, *27*, 1697–1722. [[CrossRef](#)]
10. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [[CrossRef](#)]
11. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J. End to End Learning for Self-Driving Cars. *arXiv* **2016**. [[CrossRef](#)]
12. Lian, Z.; Zeng, Q.; Wang, W.; Xu, D.; Meng, W.; Su, C. Traffic Sign Recognition Using Optimized Federated Learning in Internet of Vehicles. *IEEE Internet Things J.* **2024**, *11*, 6722–6729. [[CrossRef](#)]
13. Guo, K.; Wu, Z.; Wang, W.; Ren, S.; Zhou, X.; Gadekallu, T.R.; Luo, E.; Liu, C. GRTR: Gradient Rebalanced Traffic Sign Recognition for Autonomous Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2024**, 1–13. [[CrossRef](#)]
14. He, Y.; Meng, G.; Chen, K.; Hu, X.; He, J. Towards Security Threats of Deep Learning Systems: A Survey. *IEEE Trans. Softw. Eng.* **2020**, *48*, 1743–1770. [[CrossRef](#)]
15. Liu, F.; Chen, D.; Wang, F.; Li, Z.; Xu, F. Deep Learning Based Single Sample Per Person Face Recognition: A Survey. *arXiv* **2020**. arXiv:2006.11395. [[CrossRef](#)]
16. Vakhshiteh, F.; Nickabadi, A.; Ramachandra, R. Threat of Adversarial Attacks on Face Recognition: A Comprehensive Survey. *arXiv* arXiv:2007.11709. **2020**. [[CrossRef](#)]
17. Wang, X.; Chen, J.; He, K.; Zhang, Z.; Du, R.; Li, Q.; She, J. Survey on adversarial attacks and defenses for object detection. *J. Commun.* **2023**, *44*, 260–277.
18. Wang, Y.; Cao, T.; Zheng, Y.; Fang, Z.; Wang, Y.; Liu, Y.; Chen, L.; Fu, B. Improving the Imperceptibility of Adversarial Examples Based on Weakly Perceptual Perturbation in Key Regions. *Secur. Commun. Netw.* **2022**, *2022*, 1–12. [[CrossRef](#)]
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014. [[CrossRef](#)]
20. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015. [[CrossRef](#)]

21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
22. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]
23. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [[CrossRef](#)]
24. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**. [[CrossRef](#)]
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37. [[CrossRef](#)]
26. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**. [[CrossRef](#)]
27. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27–28 October 2019. [[CrossRef](#)]
28. Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; Yuille, A. Adversarial Examples for Semantic Segmentation and Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1378–1387. [[CrossRef](#)]
29. Li, Y.; Tian, D.; Chang, M.C.; Bian, X.; Lyu, S. Robust Adversarial Perturbation on Deep Proposal-based Models. *arXiv* **2018**, arXiv:1809.05962. [[CrossRef](#)]
30. Chow, K.-H.; Liu, L.; Loper, M.; Bae, J.; Gursoy, M.E.; Truex, S.; Wei, W.; Wu, Y. TOG: Targeted Adversarial Objectness Gradient Attacks on Real-time Object Detection Systems. In Proceedings of the 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Atlanta, GA, USA, 28–31 October 2020. [[CrossRef](#)]
31. Wang, Z.; Wang, W.; Yang, Y.; Han, Z.; Xu, D.; Su, C. CNN- and GAN-based classification of malicious code families: A code visualization approach. *Int. J. Intell. Syst.* **2022**, *37*, 12472–12489. [[CrossRef](#)]
32. Wei, X.; Liang, S.; Chen, N.; Cao, X. Transferable Adversarial Attacks for Image and Video Object Detection. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019. [[CrossRef](#)]
33. Li, Y.; Xu, G.; Li, W. FA: A Fast Method to Attack Real-time Object Detection Systems. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China, 9–11 August 2020. [[CrossRef](#)]
34. Deng, X.; Fang, Z.; Zheng, Y.; Wang, Y.; Huang, J.; Wang, X.; Cao, T. Adversarial examples with transferred camouflage style for object detection. *J. Phys. Conf. Ser.* **2021**, *1738*, 012130. [[CrossRef](#)]
35. Sharma, Y.; Ding, G.W.; Brubaker, M.A. On the Effectiveness of Low Frequency Perturbations. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019. [[CrossRef](#)]
36. Song, S.; Chen, Y.; Cheung, N.M.; Kuo, C.C.J. Defense Against Adversarial Attacks with Saak Transform. *arXiv* **2018**, arXiv:1808.01785. [[CrossRef](#)]
37. Xiao, C.; Li, B.; Zhu, J.-Y.; He, W.; Liu, M.; Song, D. Generating Adversarial Examples with Adversarial Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018. [[CrossRef](#)]
38. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.