FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 49 (2024) No. 1
ISSN 0867-6356
DOI: 10.2478/fcds-2024-0002 e-ISSN 2300-3405

sciendo

DefenseFea: An Input Transformation Feature Searching Algorithm Based Latent Space for Adversarial Defense

Zhang Pan[1] , Cao Yangjie[1] *, Zhu Chenxi[1], Zhuang Yan[1], Wang Haobo[1], Li Jie[2]

Abstract. Deep neural networks based image classification systems could suffer from adversarial attack algorithms, which generate input examples by adding deliberately crafted yet imperceptible noise to original input images. These crafted examples can fool systems and further threaten their security. In this paper, we propose to use latent space protect image classification. Specifically, we train a feature searching network to make up the difference between adversarial examples and clean examples with label guided loss function. We name it DefenseFea(input transformation based defense with label guided loss function), experimental result shows that DefenseFea can improve the rate of adversarial examples that achieved a success rate of about 99% on a specific set of 5000 images from ILSVRC 2012. This study plays a positive role in the further investigation of the relationship between adversarial examples and clean examples.

Keywords: Adversarial Attack; Adversarial Defense; Latent space; Adversarial Training

1. Introduction

As many deep neural networks (DNN) have achieved great performance in a variety of domains, especially in a lot of vision tasks. However, DNN models have exposed a problem that is vulnerable to adversarial examples [10]. Adversarial examples are deliberately crafted to fool DNN models by adding little perturbations to an image, these examples can mislead the classifier result which potentially reveals the vulnerabilities of the DNN models [25]. In many safety-first areas, we need to ensure safety first before considering the efficiency of the model. For example, face detectors can

*Corresponding Author:e-mail: caoyj@zzu.edu.cn,
[1] School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou, China,
[2] Department of Computer Science and Engineering, Shanghai Jiaotong University, China

be disabled by changing the face makeup style [13], and a cloth with special textures can hide human to fool detector [14]. Therefore, it is important to explore defense algorithms to improve the robustness of the classifiers to protect them from various attacks.

In general, training models with clean examples without any noise will lead to poor robustness of the model which only can recognize datasets that have been trained. Adversarial training [26] was proposed which mixed small perturbations in the training examples, the neural network adapts to this change to improve the robustness of the model. In one word, adversarial training is to train clean and adversarial examples together. However, adversarial training usually results in reduced generalization, that is, although the resistance to adversarial examples is improved, it may affect the classification ability of clean examples. As adversarial examples are created by adding crafted noises to clean images, it is effective to denoise adversarial examples before sending them to the model to improve its identification success rate. However, there is no method that can remove all added adversarial noises especially, in particular, the model has never trained clean examples corresponding to these adversarial examples. So, these residual small noises still can influence the result which enlarges many layers of the model.

To solve the above problems, we proposed a feature searching network for input transformation based defense with label guided loss function. Our approach uses the internal property which is the difference and similarity between adversarial examples and original examples in latent space. We investigate such differences through two mediums including feature map and channel-wise activation distribution. Motivated by results from the perspective of feature map and channel-wise activation, we then introduce a distribution searching network. The main advantage of our approach is the combination of adversarial training and the feature of images. Furthermore, in comparison to previous pre-processing based defenses, our defense demonstrates better robustness. In section 3.2, we demonstrate that all adversarial examples will make the channel-wise activation value increase, whereas our method is adaptable by adding prior to examples to make the activation value decrease so that can effectively defend against almost attacks, such as FGSM [10], PGD [17], CW [4], BIM [15]. Our contributions are summarized as follows:

- we prove the existence of potential information of latent space by feature map and channel-wise activation, which shows that most of the information about the pictures is also present in the adversarial examples;
- we design a feature searching network that can extract the difference and similarity information between adversarial and original examples in latent space. Once trained, our feature searching network could immediately extract the information of each input image and defend almost trained adversarial examples;
- our method not only defends against adversarial examples but also ensures the recognition rate of clean examples.

The rest of this paper is organized as follows. Section 2 briefly overviews the related works of adversarial attacks and defenses. In Section 3, we specifically state the

defense methodology of DefenseFea. We present the experimental results in Section 4, discuss the paper in Section 5, and conclude in Section 6.

2. Related Work

2.1. Adversarial Attack

Adversarial attack was first proposed by Szegedy et al. in 2013[25], they found that imperceptible modification of an example was able to mislead deep learning model's classification result into a wrong category, which was then widely researched by experts from various fields. The examples with non-sensitive perturbations to human eyes but mispredicting classifiers are recognized as adversarial examples. The algorithm that generates adversarial examples is also known as adversarial attack.

In general, adversarial attack can be defined as white-box and black-box according to adversary's knowledge [1]. White box attacks know all the details, so they are easier to attack, while black boxes only can obtain information by querying, but with high transferability.

Many adversarial attack algorithms have been proposed to show the vulnerability of neural networks against imperceptible changes to inputs. A single-step attack, called Fast Gradient Sign Method (FGSM), was proposed in [10]. In follow-up work, Kurakin et al. [15] proposed a robust multi-step attack, called Iterative Fast Gradient Sign Method (I-FGSM, BIM). The Projected Gradient Descent (PGD) attack is considered as one powerful attack while referring to the seminal work of Madry et al. [17] as its origin. Moosavi-Dezfooli et al. [18] specifically aimed at minimizing the norm of the adversarial perturbations. Defensive distillation [19] was a prominent technique that promised an effective solution to the problem, by building on the insights of knowledge distillation in deep networks. However, Carlini & Wagner [4] developed a set of attacks that compute norm-restricted additive perturbations that completely break defensive distillation. There also are many specific adversarial methods for specific vision areas. Differentiable Transformation Attack (DTA) [23], is a framework for generating a robust physical adversarial pattern on a target object to camouflage it against object detection models with a wide range of transformations. The object-based diverse input (ODI) [3] method can draw an adversarial image on a 3D object and induces the rendered image to be classified as the target class. The Dual Attention Suppression attack(DAS) [27] generates transferable adversarial camouflages by suppressing both model and human attention. Shadows [32] also can be dangerous when applied to traffic signs and crafted dirty roads [22] on normal roads can fool advanced Automated Lane Centering (ALC) systems which are widely deployed.

2.2. Adversarial Defense

There are three main methods of defense against adversarial attacks. (1) modifying the target model, (2) input denoise for defense, and (3) adding external modules to the model. The research direction of adversarial defenses mainly follows these three paths.

Adversarial training is one of the most extensively investigated defenses that modify the targeted model itself. It makes the model exposed to adversarial examples to gain some immunity. To control the growth of the gradient, a new AT method, Subspace Adversarial Training (Sub-AT) [16] was proposed, which constrains AT in a carefully extracted subspace. However, adversarial training is more time consuming than training on clean images only and has poor generalization. Denoise based method aims at cleaning inputs. For instance, JPEG-based [7] compression of input for removing perturbations from images. An ensemble generative cleaning with a feedback loop is proposed to clean the image from adversarial patterns [30] but relies on external generative modules to denoise adversarial images.

Different from the above methods, some techniques are added to pre-trained models to defend against attacks. A more common approach is to detect adversarial examples before input into models. Qin et al. [20] showed that reconstructing images based on class-conditional can detect adversarial examples during test time. Deng et al. [8] proposed task agnostic detection of adversarial perturbations in the input using Bayes principle. However, according to recent surveys and literature, this line become less popular in computer vision and machine learning. There also are many specific defense methods. To move towards a practical certifiable patch defense, Chen et al. [5] introduce Vision Transformer (ViT) into the framework of Derandomized Smoothing (DS). In this paper, our method combined adversarial training and external modules to defend neural networks.

3. Defense Methodology

In this section, we first explore the internal property of the difference and similarity between original examples and adversarial examples in latent space. We investigate such differences through two mediums including feature map [28] (Section 3.1) and channel-wise activation distribution [2] (Section 3.2). Motivated by the visual results, we then introduce a feature searching network for input transformation based defense with label guided loss function (Section 3.3). We give a formal description and the whole algorithm procedure in Section 3.4.

3.1. Exploration on Feature Map

Adversarial examples in image classification area are the clean image inputs added with specifically designed noise which could not only deceive the deep learning models based classifiers but also least noticeable to human visual sense. As the image is

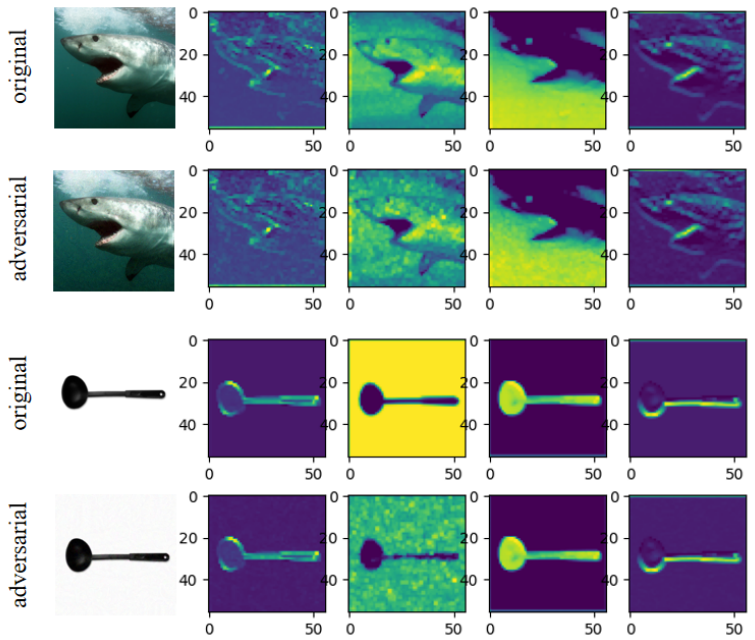


Figure 1. We show feature maps corresponding to original images and their adversarial perturbed versions. The feature maps for each pair of examples are from the same channels of the second layer in the same ResNet-50 trained on original images.

propagated through the network, disturbing values, which are constrained to be low in the input space (i.e. pixel values of the input image), update in an unlimited way in the latent space (i.e. values of the feature map) [28]. Unbounded growth of perturbations on feature maps mislead deep learning based classifiers to a wrong prediction.

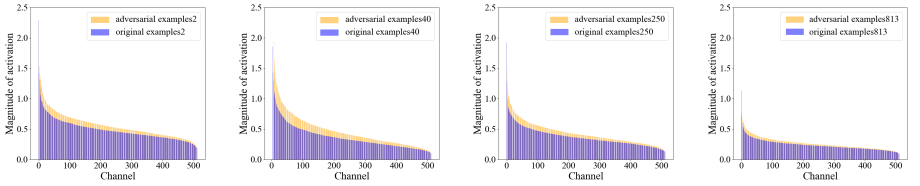
To further understand what happens in latent space, we print feature maps of the original image and the corresponding adversarial one. Given an original clean image with its adversarial version, we use the same network (i.e. ResNet-50 [11]) to capture its feature maps in the hidden layers.

Figure 1 shows that while adversarial perturbations successfully disturb maps in feature space, the distribution of original image is also remained, which could lead to correct prediction. Though adversarial perturbations are strong enough to make samples' distribution vary from the right position, the internal information reserved in image's latent space provides a probability for rectifying the adversarial examples [6]. We attempt to find out common areas between original examples and adversarial examples in hidden layers, which are crucial to transform malicious adversarial examples into benign ones before they are fed to classifiers. Based on the above finding, we propose a feature searching network to accurately obtain correct information lying

in latent space, thus such information could be used as a defensive prior for example transformation. The detailed description is demonstrated in Section 3.3.

3.2. Exploration on Channel-wise Activation Distribution

Channel-wise values contain abundant information due to the various sensitivities of each channel to different features, which demonstrates a strong connection between certain characteristics of intermediate activation and prediction results. As shown in Figure 2, we randomly choose original examples from 4 classes with their adversarial counterparts to compare channel activation caused by each pair. It is obvious that



(a) Class 2. (b) Class 40. (c) Class 250. (d) Class 813.

Figure 2. The magnitude (y-axis) of channel-wise activation at penultimate layer (512 channels evenly-skipping selected from 1024 channels at x-axis) for ResNet-50 model.

activation value distributions of adversarial examples are higher than that of original examples. This generally-occurring phenomenon indicates that adversarial perturbations have boosting effect on channels. There are advanced works [29] [2] aligning activation values of adversarial examples with those of original examples during adversarial training to improve model robustness against adversarial attacks.

However, a mass of computations is required for obtaining statistical parameters (i.e. global average pooling of thousands of images in each class), leading to time cost and storage costs. To simplify the computation procedure, we claim that the gradient trend of ultimate predicted label goes the same with activation value distribution and is, therefore, able to guide parameter iterations instead of direct operations on activation value distribution. Once obtain activation values in penultimate layer, a dense layer is supposed for reducing dimension to the same with classes' quantity for next Softmax operation. Since the parameters of target network are invariable during iterative process, gradient direction remains the same before and after dense layer. That is, altering direction of logits value before Softmax operation is consistent with activation values of penultimate layer. As a result, the relationship between predicted value and logits value could also reflect the relationship between predicted value and activation value.

We denote z_i as logits value of the i^{th} class, then predicted probability of the i^{th}

class a_i after Softmax operation could be formulated as:

$$a_i = \frac{\exp(z_i)}{\sum_{c=1}^{C} \exp(z_c)}, \tag{1}$$

where c represents class index and C represents total number of classes. Then we calculate the derivative of the predicted value versus logits value to reveal relevance between their iterative directions. As is in Eq. 2:

$$\begin{aligned}
\frac{\partial a_i}{\partial z_i} &= \frac{\partial}{\partial z_i}\left(\frac{\exp(z_i)}{\sum_{c=1}^{C} \exp(z_c)}\right) \\
&= \frac{(\exp(z_i))' \sum_{c=1}^{C} \exp(z_c) - \exp(z_i)(\sum_{c=1}^{C} \exp(z_c))'}{(\sum_{c=1}^{C} \exp(z_c))^2} \\
&= \frac{\exp(z_i) \sum_{c=1}^{C} \exp(z_c) - \exp(z_i)\exp(z_i)}{(\sum_{c=1}^{C} \exp(z_c))^2} \\
&= \frac{\exp(z_i)}{\sum_{c=1}^{C} \exp(z_c)} \cdot \frac{\sum_{c=1}^{C} \exp(z_c) - \exp(z_i)}{\sum_{c=1}^{C} \exp(z_c)} \\
&= a_i(1 - a_i).
\end{aligned} \tag{2}$$

Since $a_i \in [0, 1]$, $\frac{\partial a_i}{\partial z_i} = a_i(1 - a_i) > 0$, the predicted value monotonically increases as activation value increases, indicating that directions of predicted value and activation value are consistent during gradient descent process. Thus label of example is proved to be capable of guiding iterative procedure.

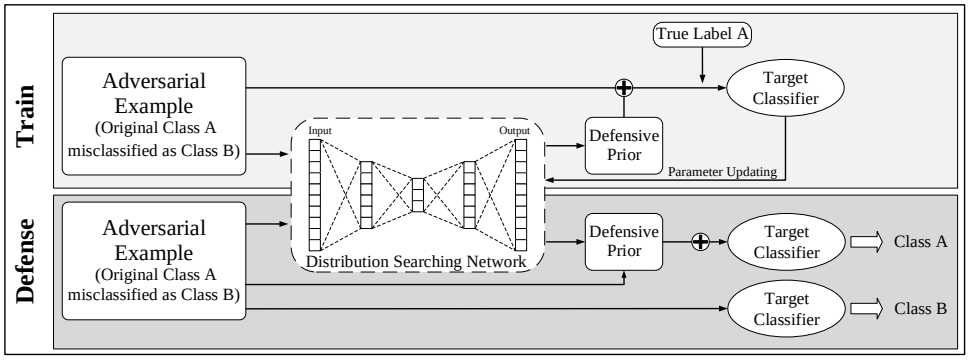


Figure 3. Overview of DefenseFea.

3.3. DefenseFea: Input Transformation Based Defense with Label Guided Loss Function

Explorations on feature maps indicate that adversarial examples contain plentiful information about their original version before attack. So the key work is to obtain

Algorithm 1: DefenseFea

Input : Adversarial example: x'; True label: y; Target classifier: $\mathcal{T}(\cdot)$; Distribution searching network: $\mathcal{S}(\cdot)$;

Output: Parameter of the extractor: θ; Defensive prior: δ;

Initialization θ_0;
for $i = 1$ **to** N **do**
  $\theta_i = \theta_{i-1}$;
  while $m = 1$ **to** M **do**
    repeat
      Extract defensive prior δ from adversarial example x'_i in latent space: $\delta = \mathcal{S}(x'_i)$;
      Calculate the cross entropy loss of input x'_i and y_i: $L(x'_i, y_i) = -\sum_{j=1}^{C} y_{i,j} \cdot \log(a_j(x'_i))$;
      Update the parameter $\theta_{i,m}$ of the extractor: $\theta_{i,m+1} = \arg\min_{\theta_{i,m}} L\{x'_i + \boldsymbol{\mathcal{S}}(x'_i), y_i\}$;
    until $\mathcal{T}(x'_i + \delta) == y_i$;
  end
end

the needed information for defensive transformation. We thus propose to train a distribution searching network to find out the true information hidden in latent space of adversarial examples and use such information as defensive prior to transforming poisonous input to innocent one.

Figure 3 demonstrates the training stage and defensive stage (i.e. application stage) of our proposed adversarial defenses algorithm – DefenseFea. The core idea of DefenseFea is obtaining defensive prior through a distribution searching network to modify adversarial examples before it is fed into target classifier. With true labels' guide at training stage, the searching network could locate the required defensive distribution accurately and thus accomplish input transformation at defensive stage. The distribution searching network will extract defensive prior from adversarial examples in latent sapce, then add the prior to adversarial examples and put them into the target classifier with true labels' guide to update the network. The procedure of DefenseFea is summarized in Algorithm 1.

3.4. Formal Description

Given an adversarial example, we use the distribution searching network as an extractor $\boldsymbol{\mathcal{S}}(\cdot)$ and extract defensive prior from latent space. Then adversarial example x' added with initial defensive prior $\delta = \boldsymbol{\mathcal{S}}(x')$ is fed into a protected classifier for transformation with the guidance of true label y. Instead of directly updating the

input, we update the parameter θ of the extractor as:

$$\theta = \arg\min_{\theta} L\{x' + \boldsymbol{\mathcal{S}}(x'), y\}, \tag{3}$$

in which $L(\cdot)$ is cross entropy loss function defined as:

$$L(x, y) = -\sum_{j=1}^{C} y_j \cdot \log(a_j(x)), \tag{4}$$

where x represents the input, j is the class index and C is the total number of classes. As mentioned before, a is the result of the Softmax function as well as the predicted label set. In Section 3.2, our experiments show that the activation values of the adversarial examples are higher than the original examples, and that the labels do guide the iteration procedures. With the use of labels to guide the updating of the distribution searching network, the prior containing true information of clean examples achieves a reduction in activation values achieves a reduction in activation values, which allows the example activation value distribution to converge to that of a normal example and thus be correctly identified. This can be reflected in Eq (2).

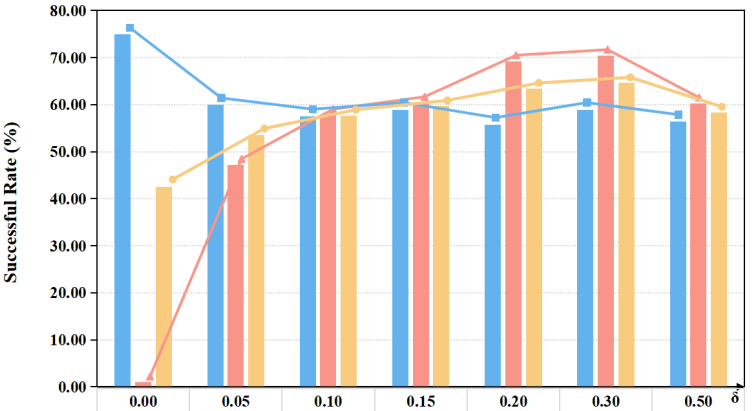
4. Results

We conduct extensive experiments only on ILSVRC 2012. As we consider ILSVRC 2012 is greater complexity and comprehensiveness. We test with four publicly available networks, including Inception-v3 [24], Resnet50, Resnet101 [11], and Resnet-152 [12]. These networks are pre-trained and we do not perform any re-training or fine-tuning on them for the whole experiment. The clean in Table stands for the success rate of clean image and adv stands for the success rate of adversarial image. The clean+prior in Table stands for the success rate of clean image with defensive prior and adv+prior stands for the success rate of adversarial image with defensive prior. The average in Table stands for the average success rate of clean+prior and adv+prior. The proposed algorithm is evaluated by the success rates – the success rate of adversarial examples and clean examples with defensive prior δ.

4.1. Implementation details

First, we use adversarial attack algorithms to generate adversarial examples from ILSVRC 2012 which is used to train our extractor $\boldsymbol{\mathcal{S}}(\cdot)$. Our extractor $\boldsymbol{\mathcal{S}}(\cdot)$ will learn the difference between adversarial and clean examples. It can extract a defensive prior δ and put it on the input image to defend the model. However, during training, we found that defensive prior δ with different coefficients k has a great influence on the accuracy of the input images. Therefore, we repeat the experiment with several different values of δ ranging from 0 to 0.5, and the results are shown in Figure 4. We use the FGSM algorithm attack model ResNet50. The recognition rates of clean and

adversarial examples are 76.16% and 0.07%. It can be found that, when δ is set to 0.3, our average success rate on ILSVRC 2012 is 64.66%. When $\delta = 0$, the defensive prior is not added to the input image, when $\delta = 0.05$ the success rate improves significantly when $\delta = 0.5$ the success rate is low because the defensive prior is too strong to influence clean images. So, in the following experiment, we set δ to 0.3.



	0.00	0.05	0.10	0.15	0.20	0.30	0.50
clean+prior	75.00	59.93	57.53	58.94	55.72	58.95	56.36
adv+prior	1.00	47.18	57.73	60.37	69.19	70.38	60.20
average	42.50	53.55	57.63	59.65	63.45	64.66	58.28

Figure 4. Different parameters δ of DefenseFea prior have different results. We can find from the table that 0.3 is the best.

4.2. Experimental Setup

In training, we generate adversarial examples with a certain percentage of clean examples of ILSVRC 2012. This allows the extractor $\boldsymbol{\mathcal{S}}(\cdot)$ to learn the difference between clean and adversarial examples, not only can the adversarial examples be correctly identified, but the misclassification of clean examples can be reduced. The proportion of adversarial examples in the whole data is η. However, we found that the average accuracy varies with η, and different η have an important impact on the results of the experiment. Therefore, we set η to 1 and 0.5 to test on ResNet50, and the result is shown in Table 1. When we set η as 1, we found that the δ extracted can be a good defense against adversarial examples but can not recognize the majority of clean images. While we set δ to 0.5, the defense success rate of adversarial examples has a sight drop and the success rate of clean examples has a great improvement, from 1% to 58.95%. According to our analysis, if we conduct a high percentage of adversarial examples will make the extractor $\boldsymbol{\mathcal{S}}(\cdot)$ learn over many knowledge of perturbations which will influence the knowledge of original images. Table 1. shows that the results are similar in the two models.

Table 1. Different η will influence the defense success rates (%) of adversarial examples and clean examples. We employ the FGSM attack method to generate adversarial examples.

models	η	adv+prior	clean+prior	average
ResNet50	0.50	70.38	58.95	64.66
	1.00	70.00	1.00	35.50
ResNet152	0.5	65.00	78.33	71.66
	1.00	70.00	1.00	35.50

Table 2. The success rates (%) of model's recognition after defending three different attacks in three models. All models have different improvements and drops.

Source	models	clean	adv	adv+prior	clean+prior	average
FGSM	ResNet50	76.16	0.07	70.38	58.95	64.66
	IncV3	77.22	0.32	59.93	67.47	63.70
	ResNet152	78.33	0.12	65.00	78.33	71.66
PGD	ResNet50	76.16	0.06	61.40	59.77	60.59
	IncV3	77.22	0.28	67.70	71.30	69.50
	ResNet152	78.33	0.11	66.80	69.44	68.12
CW	ResNet50	76.16	0.07	69.55	78.53	74.04
	IncV3	77.22	0.63	69.12	77.06	73.08
	ResNet152	78.33	0.04	70.00	78.16	74.08
	ResNet101	77.39	0.04	68.92	76.73	72.83

4.3. Comparison Experiments Among Algorithms

We conduct experiments on four models, we take four different adversarial attack algorithms, BIM [15], PGD [17], CW[4], and FGSM [10]. All the models are pre-trained and set δ to 0.3 and η to 0.5 to train. We take $\varepsilon = 8$ to attack models to generate adversarial images and attack iterations are 10. The train epoch is 100.

Table 2. shows that under three adversarial attack algorithms, our DefenseFea has a good performance. We can defend almost attack images and the success rate of adversarial images has a slight drop compared to clean images. We also found a phenomenon that adding defensive prior to clean images may improve the recognition success rate to some extent. This can also reflect that we set η to 0.5 convincingly. DefenseFea not only has a defensive effect against adversarial examples but also improves the accuracy of clean examples.

Table 3. The success rates (%) of model's recognition after defending four different attacks in three models.

Source	models	clean	adv	adv+prior	clean+prior	average
FGSM	ResNet50	100.00	0.12	98.20	99.90	99.05
	ResNet101	100.00	0.24	98.84	99.92	99.38
	ResNet152	100.00	0.20	98.56	99.90	99.23
PGD	ResNet50	100.00	0.12	98.12	99.91	99.01
	ResNet101	100.00	0.12	98.40	99.84	99.12
	ResNet152	100.00	0.12	98.75	99.90	99.32
CW	ResNet50	100.00	0.12	99.90	100.00	99.95
	ResNet101	100.00	0.08	99.80	99.90	99.85
	ResNet152	100.00	0.08	99.92	100.00	99.96
BIM	ResNet50	100.00	0.04	98.42	99.96	99.19
	ResNet101	100.00	0.08	99.78	100.00	99.89
	ResNet152	100.00	0.08	98.50	99.96	99.23

According to [9], the robust accuracy of examples with high loss suffers from a heavier drop than that of examples with low loss under larger attack strengths. Therefore, the low-confidence and even misclassified examples might have an unnecessary or even harmful effect on the robustness establishment. So, we randomly select 5000 correctly-classified images from the ILSVRC 2012 validation set [21] for comparison. Table 3 shows that we have a great performance on 5000 correctly-classified images. Our average accuracy for attack images is above 99%, and we have even reached 100% accuracy on some models and methods. Figure 5 demonstrates the

comparison of the model ResNet152. We have the best performance when defending adversarial examples generated by BIM algorithm.

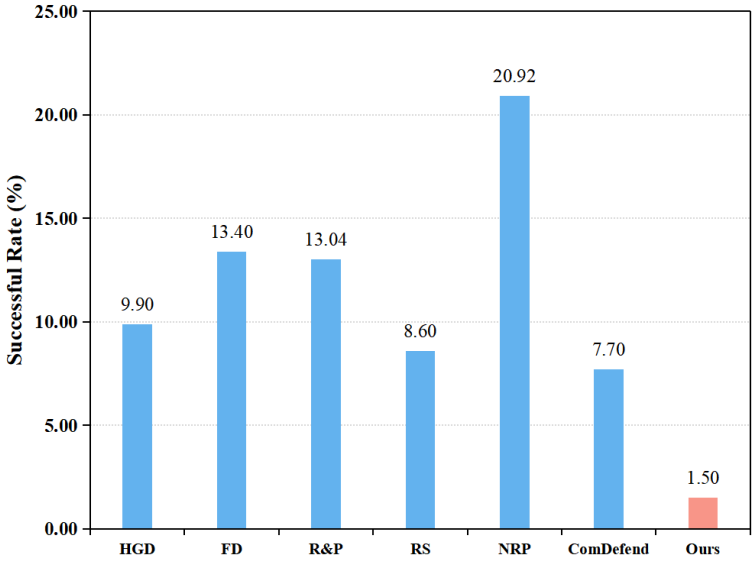


Figure 5. The success rates (%) of attacks crafted by the compared methods against 6 advanced defense mechanisms.

5. Discussion

Our method achieves good results on the ILSVRC 2012 and almost the best results to correctly-classified images from the ILSVRC 2012. Experiments once again prove that the use of misclassified natural examples will be negative and cause damage to the robustness of the model. The next step should be how to address the impact of erroneous natural examples on the model.

6. Conclusions

In this paper, we propose a novel feature searching network named DefenseFea to help models defend against adversarial attacks. Inspired by adversarial training, feature map, and channel-wise activation distribution, we introduce a feature searching network for input transformation based defense with label guided loss function. Experimental results demonstrate DefenseFea can effectively defend against four classical attack algorithms. Moreover, the proposed DefenseFea also can improve the success rates of some models when identifying correct examples. This study could be further

investigated the relationship between adversarial and clean examples.

Acknowledgment

This research was funded by the National Natural Science Foundation of China under Grant 61972092 and the Collaborative Innovation Major Project of Zhengzhou (20XTZX06013)

References

[1] Akhtar N, Mian A, Kardan N, et al., Advances in adversarial attacks and defenses in computer vision: A survey, *IEEE Access*, **9**, 2021, 155161-155196.

[2] Bai Y, Zeng Y, Jiang Y, et al., Improving adversarial robustness via channel-wise activation suppressing, *arXiv preprint arXiv*, 2021, 2103.08307.

[3] Byun J, Cho S, Kwon M J, et al., Improving the transferability of targeted adversarial examples through object-based diverse input, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 15244-15253.

[4] Carlini N, Wagner D., Towards evaluating the robustness of neural networks, *2017 ieee symposium on security and privacy (sp)*, 2017, 39-57.

[5] Chen Z, Li B, Xu J, et al., Towards practical certifiable patch defense with vision transformer, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 15148–15158.

[6] Dai T, Feng Y, Wu D, et al., DIPDefend: Deep image prior driven defense against adversarial examples, *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, 1404-1412.

[7] Das N, Shanbhogue M, Chen S T, et al., Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression, *arXiv preprint arXiv*, 2017, 1705.02900.

[8] Deng Z, Yang X, Xu S, et al., Libre: A practical bayesian approach to adversarial detection, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 972-982.

[9] Dong J, Moosavi-Dezfooli S M, Lai J, et al., The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training, *arXiv preprint arXiv*, 2022, 2211.00525.

[10] Goodfellow I J, Shlens J, Szegedy C., Explaining and harnessing adversarial examples, *arXiv preprint arXiv*, 2014, 1412.6572.

[11] He K, Zhang X, Ren S, et al., Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770-778.

[12] He K, Zhang X, Ren S, et al., Identity mappings in deep residual networks, *European conference on computer vision*, 2016, 630-645.

[13] Hu S, Liu X, Zhang Y, et al., Protecting facial privacy: generating adversarial identity masks via style-robust makeup transfer, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 15014-15023.

[14] Hu Z, Huang S, Zhu X, et al., Adversarial texture for fooling person detectors in the physical world, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 13307-13316.

[15] Kurakin A, Goodfellow I, Bengio S., Adversarial machine learning at scale, *arXiv preprint arXiv*, 2016, 1611.01236.

[16] Li T, Wu Y, Chen S, et al., Subspace adversarial training, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 13409-13418.

[17] Madry A, Makelov A, Schmidt L, et al., Towards deep learning models resistant to adversarial attacks, *arXiv preprint arXiv*, 2017, 1706.06083.

[18] Moosavi-Dezfooli S M, Fawzi A, Frossard P., Deepfool: a simple and accurate method to fool deep neural networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 2574-2582.

[19] Papernot N, McDaniel P, Wu X, et al., Distillation as a defense to adversarial perturbations against deep neural networks, *2016 IEEE symposium on security and privacy (SP)*, 2016, 582-597.

[20] Qin Y, Frosst N, Sabour S, et al., Detecting and diagnosing adversarial images with class-conditional capsule reconstructions, *arXiv preprint arXiv*, 2019, 1907.02957.

[21] Russakovsky O, Deng J, Su H, et al., Imagenet large scale visual recognition challenge, *International journal of computer vision*, **115**, 3, 2015, 211-252.

[22] Sato T, Shen J, Wang N, et al., Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack, *30th USENIX Security Symposium (USENIX Security 21)*, 2021, 3309-3326.

[23] Suryanto N, Kim Y, Kang H, et al., Dta: Physical camouflage attacks using differentiable transformation network, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 15305-15314.

[24] Szegedy C, Vanhoucke V, Ioffe S, et al., Rethinking the inception architecture for computer vision, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 2818-2826.

[25] Szegedy C, Zaremba W, Sutskever I, et al., Intriguing properties of neural networks, *arXiv preprint arXiv*, 2013, 1312.6199.

[26] Tramèr F, Kurakin A, Papernot N, et al., Ensemble adversarial training: Attacks and defenses, *arXiv preprint arXiv*, 2017, 1705.07204.

[27] Wang J, Liu A, Yin Z, et al., Dual attention suppression attack: Generate adversarial camouflage in physical world, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 8565-8574.

[28] Xie C, Wu Y, Maaten L, et al., Feature denoising for improving adversarial robustness, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, 501-509.

[29] Yan H, Zhang J, Niu G, et al., CIFS: Improving adversarial robustness of cnns via channel-wise importance-based feature selection, *International Conference on Machine Learning*, PMLR, 2021, 11693-11703.

[30] Yuan J, He Z., Ensemble generative cleaning with feedback loops for defending adversarial attacks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 581-590.

[31] Zhang T, Zhu Z., Interpreting adversarially trained convolutional neural networks, *International Conference on Machine Learning*, PMLR, 2019, 7502-7511.

[32] Zhong Y, Liu X, Zhai D, et al., Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 15345-15354.

Received 13.01.2023, Accepted 16.05.2023