applied sciences

MDPI

Article

Stylized Pairing for Robust Adversarial Defense

Dejian Guan, Wentao Zhao * and Xiao Liu

College of Computer, National University of Defense Technology, Changsha 410000, China
* Correspondence: wtzhao@nudt.edu.cn

Abstract: Recent studies show that deep neural networks (DNNs)-based object recognition algorithms overly rely on object textures rather than global object shapes, and DNNs are also vulnerable to human-less perceptible adversarial perturbations. Based on these two phenomenons, we conjecture that the preference of DNNs on exploiting object textures for decisions is one of the most important reasons for the existence of adversarial examples. At present, most adversarial defense methods are directly related to adversarial perturbations. In this paper, we propose an adversarial defense method independent of adversarial perturbations, which utilizes a stylized pairing technique to encourage logits for a pair of images and the corresponding stylized image to be similar. With stylized pairing training, DNNs can better learn shape-biased representation. We have empirically evaluated the performance of our method through extensive experiments on CIFAR10, CIFAR100, and ImageNet datasets. Results show that the models with stylized pairing training can significantly improve their performance against adversarial examples.

Keywords: stylized pairing; robust optimization; adversarial defense; deep learning

check for updates

Citation: Guan, D.; Zhao, W.; Liu, X. Stylized Pairing for Robust Adversarial Defense. *Appl. Sci.* **2022**, *12*, 9357. https://doi.org/10.3390/app12189357

Academic Editor: Flavio Cannavò

Received: 16 August 2022
Accepted: 16 September 2022
Published: 18 September 2022

1. Introduction

Deep neural networks (DNNs) have shown great success in many applications; however, DNNs are not always robust, and they are especially vulnerable to adversarial examples [1–3]. By adding small adversarial perturbations which are less perceptible to human beings on benign examples, attackers can fool DNNs with alarmingly high probabilities [1]. This phenomenon has quickly attracted a wide range of research interests, and there are many works trying to explain the underlying reasons for the existence of adversarial examples. A recent study [4] has shown that DNNs work very differently from human beings and pointed out that CNNs overly rely on object local features such as textures rather than global object shape. That is, since DNNs have different behavior patterns from human beings, DNNs tend to make decisions by local textures which are easily distorted by all kinds of noise. Therefore, although adversarial perturbations are weak and less perceptible to human beings, they still can lead to DNNs with erroneous decisions [5].

There are many defense mechanisms proposed to defend against adversarial examples. Robust optimization is an important research field. On the one hand, there are approaches aiming to generate robust models against adversarial examples by introducing robust architectures or robust parameters. This line of research mainly includes adversarial training [5–7], certified defense [8–10], and the regularization approach [11–13]. In particular, adversarial training is considered as the most effective approach for adversarial defense, but it has also a high computational complexity. Meanwhile, because of the transferability of adversarial examples [1], that is, the adversarial examples which can mislead one model can often mislead other models with the same task, and many seemingly effective methods can also be bypassed. On the other hand, some researchers solve this issue by adding better intuition to the models through explainability. These methods guide models to learn human preferences such as shape bias [14] or action pattern [15]. In general, these methods need to use some domain knowledge, so there is relatively little research in this area at present, but it is also a promising direction.

In this paper, we provide a novel method to increase models' explainability and further improve the model's robustness against adversarial examples. To our best knowledge, it is the first time that the DNN models' different classification strategies from human beings as an important factor in the existence of adversarial examples are proposed, and we propose stylized pairing to bridge the gap. Different from other robust optimization methods, the proposed method utilizes stylized pairing training to encourage logit outputs of a pair of the original image and the corresponding stylized image to be similar and enforce the model to learn more generalized representations and therefore generate more robust models. That is, it tries to increase the semantic similarity between original images and stylized images and utilizes such semantic similarity to increase the models' explainability.

As shown in Figure 1, a stylized network is pre-trained to generate stylized examples in which the VGG Encoder and Decoder are the main parts to construct the stylized images with content images and style images. The AdaIN operation performs the style change by aligning the mean and variance of content features (the latent features) with those of the style image's feature. We then minimize the outputs' difference between stylized examples and corresponding normal examples as pairing loss and combine it with original task loss to train DNN networks. The code is available at https://github.com/lingKok/robust-optimization-with-stylized-pairing, accessed on 15 August 2022.

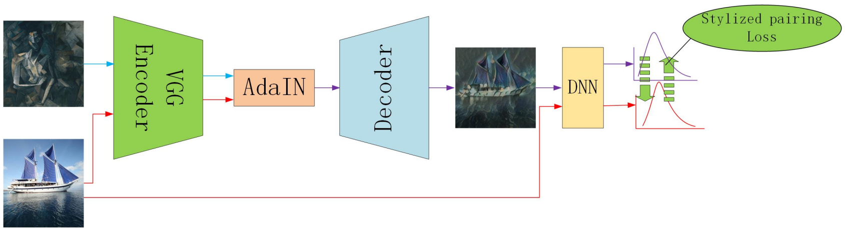

Figure 1. The architecture of our defense method based on stylized pairing: In the training process, stylized images and corresponding clear images are used to train the DNN model, and the stylized pairing loss is utilized to encourage the outputs to be as similar as possible. More details can refer to Section 3.3.

The contributions of this paper are mainly threefold:

- We propose an adversarial defense method independent of adversarial perturbations, that is, stylized pairing training. By encouraging logit outputs for a pair of original image and corresponding stylized image to be similar, the proposed method increases the semantic similarity to improve models' explainability.
- We propose an evaluation method to measure the robustness of models against adversarial examples with linear interpolation and analyze the training strategies with stylized pairings.
- Extensive experiments have been conducted, and the experimental results show that our method can efficiently extract shape-biased representation and therefore improve model robustness against adversarial examples.

2. Related Works

In this work, we utilize the style transfer model to modify local textures and propose a stylized pairing training strategy to generate robust deep neural networks against adversarial examples. Since our work highly relates to style transfer and adversarial defense problems, we will briefly discuss them in this section.

2.1. Style Transfer

Style transfer has been a very important research field in the past decade. The early algorithms are designed for particular artistic styles and cannot be easily extended to other styles. With the appearance of convolutional neural networks (CNNs), style transfer based on CNN representation has gradually become a research hotspot. Gatys et al. [16] first studied how to use a CNN to reproduce famous painting styles on natural images. They proposed to model the content of an image as the feature responses from a pre-trained CNN and further model the style of the other image as the summary feature statistics. By minimizing the style loss and content loss, the DNN model iteratively optimizes the stylized image while fixing the model's parameters. Given style and content targets s and c and layers j and $\hat{j}$, the feature and style reconstruction are performed by optimizing the problem:

$$y^{n+1} = \underset{y}{\mathrm{argmin}} \lambda_c \mathcal{L}^{j}_{feat}(y^n, y^{n+1}) + \lambda^{J}_{style} \mathcal{L}(y^n, y^{n+1}) + \lambda_{TV} \mathcal{L}_{TV}(y^{n+1}), \tag{1}$$

where λ_c, λ_s, and λ_{TV} are regularization parameters, y is initialized with random noise, and the optimization is performed by using the L-BFGS method.

In order to generate stylized images in real time, Johnson et al. [17] proposed to add an autoencoder as a feedforward network which combines the benefits of feedforward image transformation tasks and optimization-based methods to fit the process of style transfer with the same loss functions used in [16]. In order to generate multiple styles with a single model, Chen et al. [18] introduce a style Bank Layer K to control the autoencoder to output different styles. Dumoulin et al. [19] proposed to use a conditional instance normalization so that different styles can be realized by different scale and shift operations. Furthermore, Huang et al. [20] argued that the variance and mean of the latent features decide the style; therefore, they proposed an adaptive instance normalization (AdaIN) layer which aligns the mean and variance of content features with those of the style features. Given a content input x and a style input y, the operation of AdaIN can be formulated as follows:

$$\text{Adathe IN}(x, y) = \delta(y)\left(\frac{x - \mu(x)}{\delta(x)}\right) + \mu(y), \tag{2}$$

where $\delta(\cdot)$ is the variance operation and $\mu(\cdot)$ is the average operation. The normalized content input x is scaled by $\delta(y)$ and shifted with $\mu(y)$.

2.2. Adversarial Defense

Since DNNs are vulnerable to adversarial examples [1], many researchers have been devoted to investigating adversarial defense. The adversarial defense methods can be divided into three categories: Robust Optimization, Input pre-processing, and Adversarial Detection.

- Robust Optimization-based methods aim to improve the robustness of models either by introducing regularization term [11], certification bounds [10], adversarial training [5] or explainability [14,15].
- Input pre-processing-based methods are based on the intuition to counteract the effect of adversarial perturbation, and they are usually achieved by data compressing [21], input encoding [22], input transforming [23], and so on.
- Adversarial Detection-based methods can be further divided into two categories: the methods utilizing the prediction inconsistency [24], and methods utilizing statistical characteristics to distinguish between adversarial and normal example [25].

We concentrate on Robust Optimization with explainability in this study. There are works trying to train robust models by adding better intuition to the models through explainability. Borji [14] utilized the edge map as an additional channel to guide models to learn the shape-bias representation. In this approach, the edge map is obtained by the Canny edge detector, and they perform adversarial training over the 2D (Gray+Edge) or 4D (RGB+Edge) input. Addepalli et al. [15] thought that unlike DNNs, people perceive images based on their predominant features; therefore, they attempted to train networks to form coarse impressions based on the information in higher bit planes and utilized the lower bit planes only to refine their prediction. Compared with other robust optimization methods, the robust optimization method based on explainability needs more domain knowledge, so the related research work is in its infancy, but it is obviously a field of great research value. Meanwhile, there are some works utilizing stylized images for other problems. Somavarapu et al. [26] and Brochu [27] augmented the dataset with stylized images to address the Domain Generalization problem, where the classifier must generalize to an unknown target domain. Different from our work, these works see stylized images more as a way of data augmentation.

In addition, it is worth mentioning that Kannan et al. [28] proposed the concept of the logit pairing in which they introduced adversarial logit pairing and clear logit pairing. The adversarial logit pairing encourages logits for pairs of clear examples and their corresponding adversarial example to be similar, and the clean logit pairing selected two random clean training examples (typically not even from the same class) to perform pairing training in order to minimize the logit outputs between two classes. Nasser et al. [29] proposed stylized adversarial training which utilizes the content and the style of the target image as well as the classifier boundary information to generate adversarial perturbation and perform adversarial training. Unlike these methods, our proposed stylized pairing emphasizes a semantic similarity, while adversarial logit pairing and stylized adversarial training focus on a pixel-level similarity directly relate to adversarial perturbation, and clear logit pairing plays a role of regularity terms to minimize the difference between samples from different classes.

3. Material and Methods

In the following section, we first introduce and describe datasets and model architectures used for experiments. Then, we present the proposed stylized pairing defense method in detail.

3.1. Datasets

Three image datasets, including CIFAR-10, CIFAR-100, and ILSVRC2012, are selected for experiments. The images in all datasets are reshaped to a size of 224 × 224. The details of each dataset are presented as follows:

- **CIFAR-10:** It is a subset of the Tiny ImageNet dataset and is composed of 60,000 images. There are 10 classes including airplane, automobile, bird, cat, deer, horse, ship, and truck. All images were cropped to 32 × 32 pixels.
- **CIFAR-100:** Similar to CIFAR-10, CIFAR-100 is also a subset of the Tiny ImageNet dataset and consists of 60,000 32 × 32 color images. In total, 100 classes are grouped into 20 super-classes. For each class, there are 600 images including the 500 training images and 100 testing images.
- **ILSVRC2012:** ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) is a subset of large hand-labeled ImageNet datasets, containing 1000 categories and 1.2 million images. In our experiment, we select 100 categories to conduct our experiments due to hardware limitations. We use ImageNet to denote this subset in the rest of the paper.

3.2. Model Architectures

Three model architectures are selected for our experiments.

- **ResNet34:** [30] contains 34 layers in total. It consists of a 7 × 7 convolution layer and max pooling layer in its base layers, and 4 blocks 3 × 3 convolution layers, with residual links between two consecutive layers. The channel numbers vary from 64 to 512 with the increase in layer number. Finally, the classification layer is composed of an average pooling layer and a fully connected layer.
- **GoogLeNet:** [31] is designed to keep a low computational budget. It contains 22 layers and is composed of 7 × 7 convolution layers, max-pooling layers, and 1 × 1 and 3 × 3 convolution layers. After nine repeating inception modules, a fully connected layer is used for prediction.
- **MobileNet:** [32] has been extended to multiple versions. In this paper, we adopt the MobileNetV3 (small), which based on MobileNetV1's depth-wise separable convolu-

tions and MobileNetV2's linear bottleneck and invert residual structure, introduces lightweight attention modules and excitation into the bottleneck structure. For details, please refer to [32].

In experiments, the pre-trained models are implemented and executed in PyTorch [33]. In order to adapt to different class numbers, the fully connected layers are reconstructed and randomly initialized by the default method in PyTorch.

3.3. Stylized Pairing Training

As shown in Figure 1, the proposed method mainly consists of two unique parts: stylized pre-processing and stylized pairing loss. The stylized network generates the stylized images, and the pairing loss is utilized to force the content images and stylized images to be similar.

Stylized Pre-Processing: Stylized pre-processing takes a content image c and a style image s, and it creates an output image that combines the content of the c with the style of the s. In this paper, we follow the original settings of [20] (the code and pre-trained models are available at https://github.com/xunhuang1995/AdaIN-style (accessed on 1 September 2022)) and utilize a single encoder–decoder architecture to realize style transfer. The first several layers of encoder $f(\cdot)$ (up to $relu4\_1$) are fixed to that of a pre-trained VGG-19, and the decoder g is used to generate stylized images. The AdaIN layer aligns the mean and variance of the content feature map to those of the style feature map to produce the target feature map t. It can be expressed as follows:

$$t = AdaIN(f(c), f(s)). \tag{3}$$

In the training phase, the decoder g is learned to map the t back to the image space to generate the stylized image $T(c,s)$:

$$T(c,s) = g(t). \tag{4}$$

It is worth mentioning that the function $AdaIN(\cdot,\cdot)$ performs style transfer in the latent space by transferring the feature channel-wise mean and variance as shown in Equation (1). Because $AdaIN$ has no learnable affine parameters, it adaptively computes the affine parameters from the style input. Due to its low overhead to generate style transfer images, we utilize it as a part of our method to generate stylized images. In Figure 2, we show the stylized images on three datasets: CIFAR-10, CIFAR-100, and ImageNet, respectively.

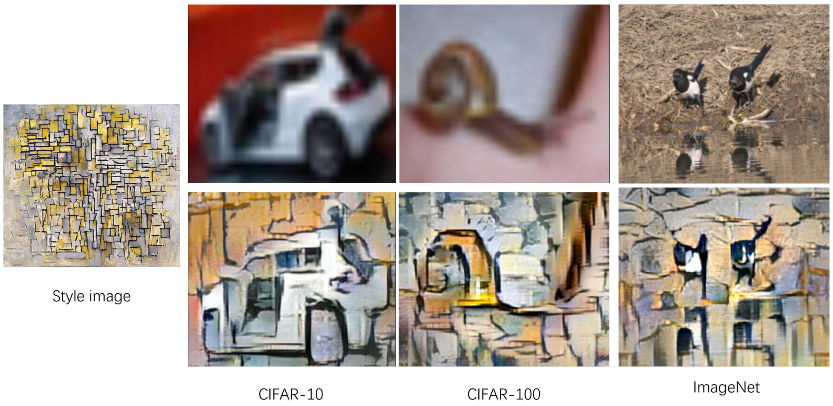

Figure 2. Stylized image examples on three dataset: CIFAR-10, CIFAR-100, and ImageNet.

Stylized Pairing Loss: Kannan et al. [28] proposed the concept of adversarial logit pairing, which encourages the logit of adversarial example and corresponding normal examples to be similar. Inspired by this idea, we propose a novel method that encourages the stylized images and their corresponding clear images to be similar. For a deep neural network $F(\cdot)$, it takes an input x to compute an output $z = F(x)$, and the stylized loss can be formulated as follows:

$$\mathcal{L}_s = \lambda L(F(x), F(x')), \tag{5}$$

where x' is the stylized image generated by the stylized network, λ is a regularization parameter to determine the strength of the stylized pairing penalty, and $L(\cdot, \cdot)$ is the loss function to measure the similarity. In the proposed method, we use mean squared loss for L, which measures the mean squared error (squared L_2 norm) between two elements.

Stylized pairing Training: Stylized loss pairing matches the DNN's output from a normal image x and its corresponding stylized image x'. In the traditional training scheme, they treat stylized images as data augmentation. The model is trained to assign both x and x' to the same output class label, but the model has no access to receiving any information to indicate that x' is more similar to x compared to another example with the same class with x. Therefore, we propose to add a stylized pairing loss on the base of the normal training with stylized images. Let us denote with $J(M, \theta)$ the cost function of normal training with stylized images where a model with parameters θ is trained on a mini-batch M including normal images $\{x_1, x_2, \ldots, x_m\}$ and corresponding stylized images $\{x'_1, x'_2, \ldots, x'_m\}$. The proposed stylized pairing training aims to minimize the loss:

$$\mathcal{L}_t = J(\theta, M) + \lambda \frac{1}{m} \sum_{i=1}^{m} L(F(x_i; \theta), F(x'_i; \theta)). \tag{6}$$

where $J(\theta, M)$ is the cross-entropy loss to train the classifier on each example in the batch. Due to time and computational constraints, the models presented are initialized with parameters provided by PyTorch [33].

Evaluation: Adversarial attack is normally measured by two indexes: misclassification rate and perturbation amplitude. In this paper, we also adopt these two metrics for comparative evaluation. We used the Fast Gradient Sign Method (FGSM) to find the adversary, with a learning rate of 1×10^{-4} (perturbation amplitude per step), and ran it for 100 steps on 100 test set images from CIFAR-10, CIFAR-100, and ImageNet, respectively. For each image, we sample its L_2 distance from the unperturbed image and its L_∞ distance from the unperturbed image (out of 1) based on their individual linear interpolations, and we compute its interpolated accuracy rate as adversarial accuracy. Therefore, the adversarial accuracy and its corresponding perturbation amplitude (including L_2 and L_∞ norms) are used to measure the robustness of models.

4. Results

In this section, we compare the proposed stylized pairing training method with normal training and adversarial training. The performances are measured based on adversarial accuracy rate and perturbation amplitude including L_2 norm and L_∞ norm.

4.1. Comparisons with Normal Training

We compare the robustness of models with stylized pairing training and normal training against adversarial attacks. The ResNet34, GoogLeNet, and MobileNet models are trained with the proposed stylized pairing training scheme and the normal training scheme without stylized images. Figure 3 shows the average adversarial accuracy of stylized pairing and normal training with respect to different strengths of perturbation measured in the L_2 norm. It is worthy to note that the strength of perturbation measured in the L_∞ norm shows a similar performance. The red lines refer to the model with the stylized pairing training, and the cyan lines refer to the normal training. We can see that the stylized pairing training is significantly more robust than the normal training in terms of adversarial accuracy, and the robust performance shows a similar tendency on different models and different datasets, which means the stylized pairing training has a good generalization ability. This is in line with our hypothesis that stylized pairing training would be able to improve the model robustness against adversarial examples. We conjecture that in the stylized training process, the stylized pairing loss plays a role of regulation to increase the similarity of the outputs of models between normal input and stylized input, and it can actually improve the model's uncertainty further to improve the model's ability to defend against adversarial examples.

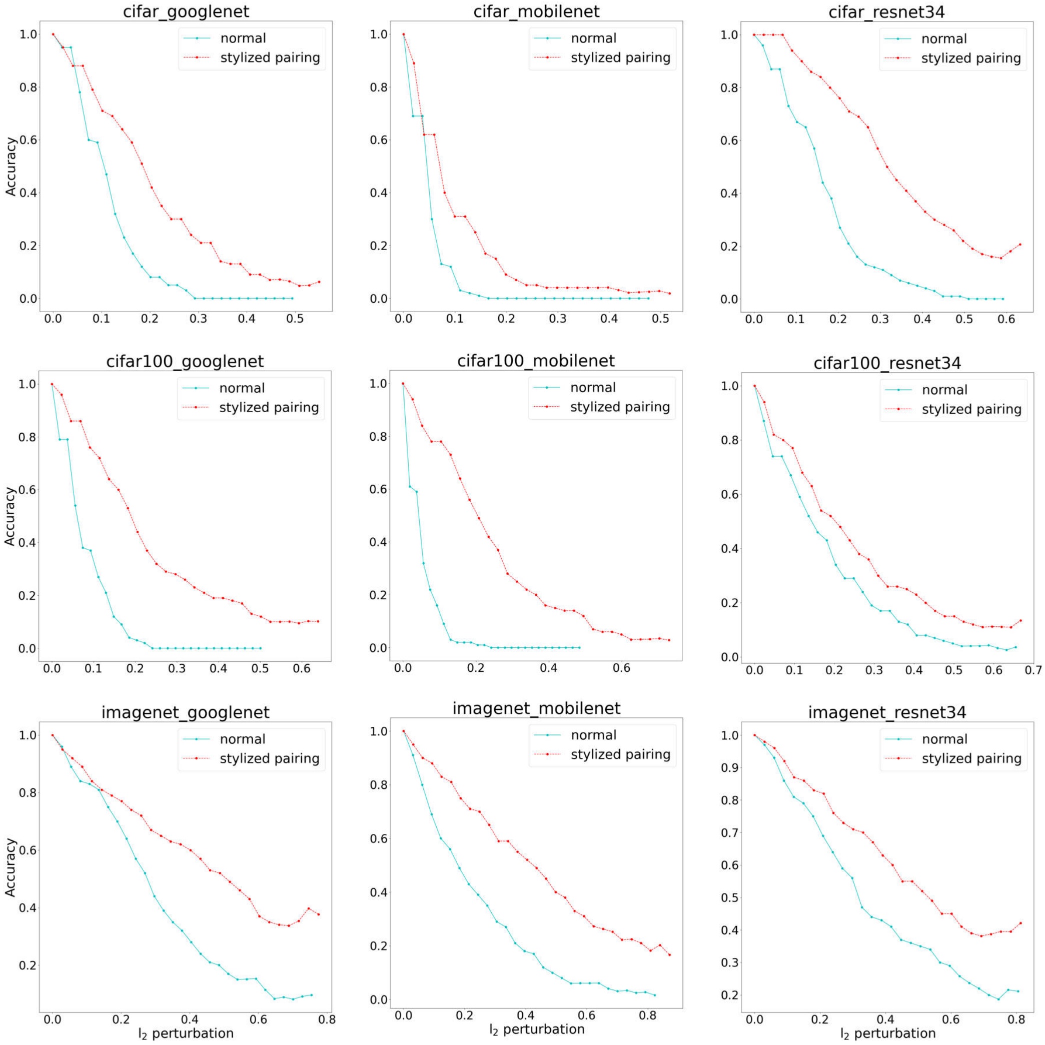

Figure 3. The average adversarial accuracy of a set of 100 test images with respect to different perturbation strengths imposed by the Fast Gradient Sign Method.

4.2. Comparisons with Adversarial Training

In the experiment, we introduce adversarial training, which is known as the most useful means to improve the model's ability to defend against adversarial examples. We compared the performance of models trained with stylized pairing training and those with adversarial training. For adversarial training, we reproduced the least-likely class method [5] in which the least-likely class loss is minimized and the perturbation amplitude is set to 0.1. In the training phase, adversarial examples are generated and added to the training set, and the training is ended when the accuracy of the validation set converges. Figure 4 shows the average adversarial accuracy of stylized pairing and adversarial training with respect to different strengths of perturbation measured in L_2 norm. We can see that the stylized pairing training and adversarial training achieve a similar performance against adversarial examples. This further proves that our method can effectively improve model robustness against adversarial attacks and may become another promising tool to improve the robustness of deep machine learning.

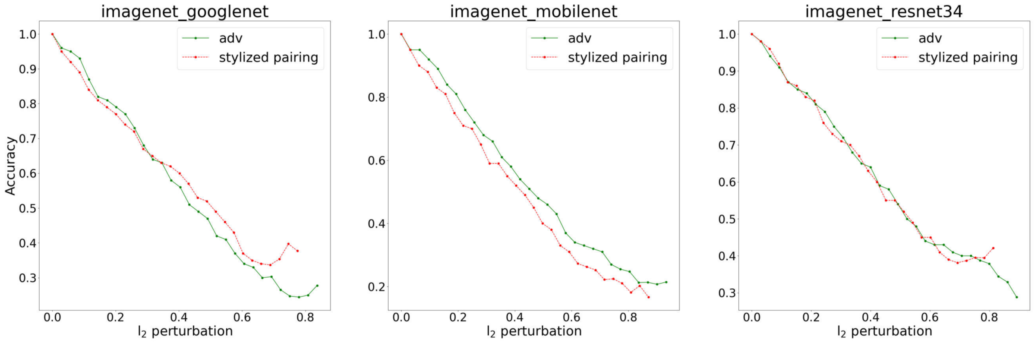

Figure 4. The performance comparisons between models with stylized pairing training and those with adversarial training on ImageNet. The red lines refer to stylized pairing training and the green lines refer to adversarial training.

5. Discussions

In this section, we discuss the usefulness and influence on model performance and the selection of style images of the proposed stylized pairing method to guide how to use our method.

Necessity for pairing loss: The pairing loss is proposed based on the observation that standard stylized training without the pairing loss term sees stylized images as data augmentation and has no access to receiving any information to indicate clear images similar to the corresponding stylized images. In order to verify whether pairing loss is really useful, we conduct an experiment to compare stylized training and stylized pairing training. In Figure 5, we show the adversarial accuracy on the ImageNet dataset using the ResNet34 model architecture (other datasets and model architectures in terms of perturbation strength measured by its L_2 and L_∞ norms show similar performance). We can find that the model with stylized pairing training shows better performance compared to the model with stylized training without the pairing loss term. This indicates that pairing loss is useful to improve the performance against adversarial attacks. We conjecture that the semantic similarity of the stylized images and their corresponding normal images has been significantly enhanced, and for stylized training, semantic similarity is more important compared with pixel-level similarity. Unlike the work [28] where they utilize the adversarial logit pairing and clear logit pairing to train robust models, even though they use the pairing loss, for adversarial logit, the difference between adversarial examples and the corresponding original examples is still very small. In essence, it still pursues a pixel-level similarity. For clean logit, they aim to smooth the gradient of the cost function with regard to inputs by forcing the output of samples from different classes to be similar; therefore, in fact, it penalizes the sensitivity of the divergence between the predictions and uniform uncertainty.

Standard accuracy: We studied the performance of models on normal examples in order to test whether stylized pairing training would reduce the performance of models. In Tables 1 and 2, we compared the models with stylized pairing training and normal training and reported the accuracy rate of three model architectures on CIFAR10, CIFAR100, and ImageNet datasets, respectively. We can find that there exists a decline in accuracy rate in most cases, and stylized pairing training would lead to about a 1% accuracy reduction on clean examples. Similar situations also exist in adversarial training. Raghunathan et al. [34] pointed out that while adversarial training can improve robust accuracy in terms of adversarial examples, it hurts standard accuracy (when there is no adversary). There is a tradeoff between standard and robust accuracy. We speculate that similar to adversarial training, the stylized pairing loss acts as a regularization term leading to the reduction of performance, and the poor generating quality of stylized images is another main reason for affecting the performance of models. As a meaningful aspect, we will try to improve the performance of models with stylized pairing training in future work either by increasing the quality of stylized images or by losing the constraint.

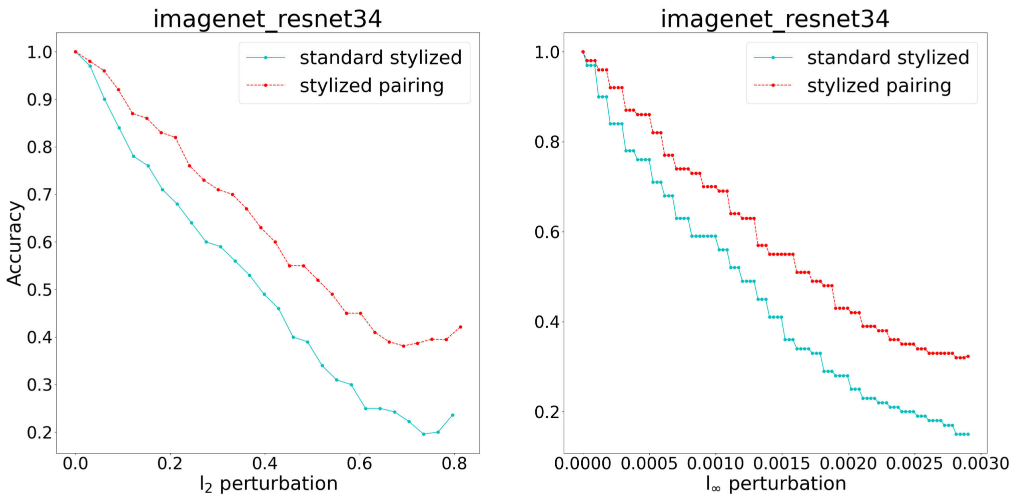

Figure 5. The adversarial accuracy rate of stylized pairing training style vs. standard stylized training without pairing loss where cyan lines denote models trained with standard stylized training and red lines denote models trained with stylized pairing. We show the perturbation strength measured by its L_2 norm (**left panel**) and L_∞ norm (**right panel**).

Table 1. The accuracy rate of the models with stylized pairing training and normal training on the CIFAR-10 and CIFAR-100 datasets. Stylized denotes stylized pairing training and Normal refers to normal training without stylized images.

	CIFAR-10			**CIFAR-100**		
	GoogLetNet	**MobileNet**	**Resnet34**	**GoogLetNet**	**MobileNet**	**Resnet34**
Stylized	94.05%	92.00%	94.92%	78.72%	75.44%	78.65%
Normal	95.01%	93.29%	95.59%	78.25%	74.80%	80.14%

Table 2. The accuracy rate of the models with stylized pairing training and normal training on ImageNet. Stylized denotes stylized pairing training and Normal refers to normal training without stylized images.

	ImageNet		
	GoogLetNet	**MobileNet**	**Resnet34**
Stylized	77.74%	73.24%	77.54%
Normal	79.12%	74.80%	79.00%

Shape-biased features: The motivation of stylized pairing training is to enhance the model's capability on learning shape-biased features and therefore to enhance the model's robustness against adversarial examples. In order to verify whether stylized pairing training is able to learn shape-biased features, we introduce LIME [35], which is a modular and extensible approach to explain the predictions of models that uses sparse linear explanations for image classifiers and highlights the super-pixels with positive weights toward a specific class. In Figure 6, we show the visualization results of LIME for a ResNet34 model with stylized pairing training and normal training on ImageNet, respectively. We randomly sampled two images correctly classified by both models from the test set and highlighted the super-pixels for the top 5 labels with the LIME method. We can find that the model with stylized pairing training shows better interpretability and that the outlines of images are correctly emphasized in the stylized training process, while normal training pays irregular attention. This result confirms that the model with stylized pairing training can improve the learning capability on shape-biased features to a certain extent. Different from the work [14] which utilized the edge map to guide models to learn the shape-bias representation, our approach is to learn by analogizing two semantically related examples, and at the same time, we achieve data augmentation in this process.

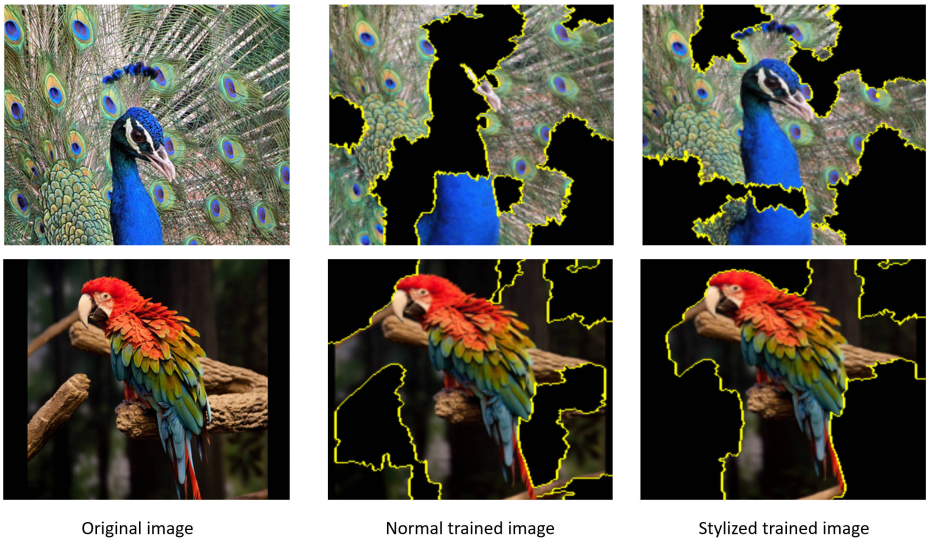

Figure 6. Explaining image classification prediction made by models with stylized pairing training and normal training.

Random style or single style: In this section, we investigate how to use style images in stylized pairing training, focusing on selecting randomly style images from multiple style images or fixing a style image to generate stylized images in the training phase. In Figure 7, we sampled the L_2 and L_∞ perturbation norms from models trained with random style images and single style images on the ImageNet dataset and ResNet34 model architectures. It is worth noting that results show similar performances on other datasets and model architectures. We can find that the benefit of the randomly selected style images is not significant; in contrast, a single style shows better performance. This result indicated that there is no need to use too many style images to train a model in stylized pairing training.

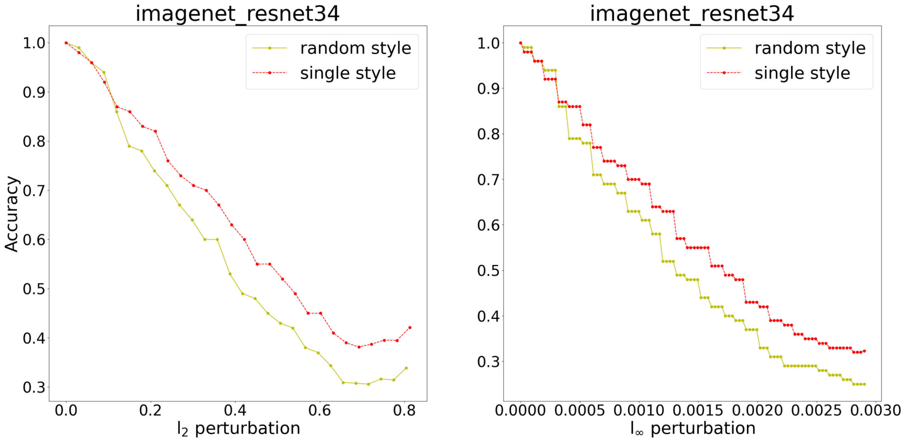

Figure 7. The adversarial accuracy rate of random style vs. single style where yellow lines denote models trained with randomly selected style images and red lines denote models trained with one single style of images. We show the perturbation strength measured by its L_2 norm (**left panel**) and L_∞ norm (**right panel**).

6. Conclusions

This study is inspired by the finding that convolutional neural networks (CNNs) overly rely on object textures; however, humans are more biased toward shape to recognize objects. The proposed method utilizes stylized pairing training to guide the models to lean shape bias representation. By increasing the model's semantic interpretability, we expected that the model trained with stylized pairing would have a stronger capability to learn shape-biased representation. With extensive experiments, we confirmed that stylized pairing training can effectively extract shape-biased representation and is robust against adversarial examples. It should be noted that pairing loss is efficient to guide models to learn semantic similarity.

In future work, we will address the limitations whereby our method leads to a decrease in standard accuracy and is relatively computationally complex; the main reason is that the generation process is time-consuming, and the quality of the generated stylized images is poor. Therefore, we will explore more efficient means to improve the quality of stylized images and decouple the process of generating stylized images and the process of training to accelerate the overall training speed. In addition, it is also a promising research direction to extend the current method beyond the classification task and explore more semantic similarities to increase model interpretability.

Author Contributions: Conceptualization, D.G. and X.L.; methodology, D.G.; software, D.G.; validation, D.G., W.Z. and X.L.; formal analysis, D.G.; investigation, D.G.; resources, D.G.; data curation, D.G.; writing—original draft preparation, D.G.; writing—review and editing, D.G.; supervision, W.Z.; project administration, W.Z. funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. U1811462).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available on request to the authors.

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

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
2. Yu, Z.; Zhou, Y.; Zhang, W. How Can We Deal With Adversarial Examples? In Proceedings of the 2020 12th International Conference on Advanced Computational Intelligence (ICACI), Dali, China, 14–16 March 2020; pp. 628–634.
3. Peng, Y.; Zhao, W.; Cai, W.; Su, J.; Han, B.; Liu, Q. Evaluating deep learning for image classification in adversarial environment. *IEICE Trans. Inf. Syst.* **2020**, *103*, 825–837. [CrossRef]
4. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv* **2018**, arXiv:1811.12231.
5. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
6. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
7. Sen, S.; Ravindran, B.; Raghunathan, A. Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks. *arXiv* **2020**, arXiv:2004.10162.
8. Katz, G.; Barrett, C.; Dill, D.L.; Julian, K.; Kochenderfer, M.J. Reluplex: An efficient SMT solver for verifying deep neural networks. In Proceedings of the International Conference on Computer Aided Verification, Heidelberg, Germany, 24–28 July 2017; pp. 97–117.
9. Gehr, T.; Mirman, M.; Drachsler-Cohen, D.; Tsankov, P.; Chaudhuri, S.; Vechev, M. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–23 May 2018; pp. 3–18.
10. Zhai, R.; Dan, C.; He, D.; Zhang, H.; Gong, B.; Ravikumar, P.; Hsieh, C.J.; Wang, L. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv* **2020**, arXiv:2001.02378.
11. Ross, A.; Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2–7 February 2018; Volume 32.
12. Gu, S.; Rigazio, L. Towards deep neural network architectures robust to adversarial examples. *arXiv* **2014**, arXiv:1412.5068.
13. Xie, C.; Wu, Y.; Maaten, L.v.d.; Yuille, A.L.; He, K. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 501–509.

14. Borji, A. Shape Defense Against Adversarial Attacks. *arXiv* **2020**, arXiv:2008.13336
15. Addepalli, S.; BS, V.; Baburaj, A.; Sriramanan, G.; Babu, R.V. Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1020–1029.
16. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576.
17. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.
18. Chen, D.; Yuan, L.; Liao, J.; Yu, N.; Hua, G. Stylebank: An explicit representation for neural image style transfer. In Proceedings of the IEEE Conference On computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1897–1906.
19. Dumoulin, V.; Shlens, J.; Kudlur, M. A learned representation for artistic style. *arXiv* **2016**, arXiv:1610.07629.
20. Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1501–1510.
21. Dziugaite, G.K.; Ghahramani, Z.; Roy, D.M. A study of the effect of jpg compression on adversarial images. *arXiv* **2016**, arXiv:1608.00853.
22. Buckman, J.; Roy, A.; Raffel, C.; Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
23. Guo, C.; Rana, M.; Cisse, M.; Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv* **2018**, arXiv:1711.00117.
24. Xu, W.; Evans, D.; Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv* **2017**, arXiv:1704.01155.
25. Ma, X.; Li, B.; Wang, Y.; Erfani, S.M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M.E.; Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv* **2018**, arXiv:1801.02613.
26. Somavarapu, N.; Ma, C.Y.; Kira, Z. Frustratingly simple domain generalization via image stylization. *arXiv* **2020**, arXiv:2006.11207.
27. Brochu, F. Increasing shape bias in ImageNet-trained networks using transfer learning and domain-adversarial methods. *arXiv* **2019**, arXiv:1907.12892.
28. Kannan, H.; Kurakin, A.; Goodfellow, I. Adversarial logit pairing. *arXiv* **2018**, arXiv:1803.06373.
29. Naseer, M.; Khan, S.; Hayat, M.; Khan, F.S.; Porikli, F. Stylized adversarial defense. *arXiv* **2020**, arXiv:2007.14672.
30. 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.
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
33. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv* **2019**, arXiv:1912.01703.
34. Raghunathan, A.; Xie, S.M.; Yang, F.; Duchi, J.C.; Liang, P. Adversarial training can hurt generalization. *arXiv* **2019**, arXiv:1906.06032.
35. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.