RIA: A Reversible Network-based Imperceptible Adversarial Attack

Fanxiao Li, Renyang Liu
Engineering Research Center of Cyberspace
Yunnan University
Kunming, China
{lifanxiao, ryliu}@mail.ynu.edu.cn

Zhenli He*, Song Gao, Yunyun Dong, Wei Zhou
National Pilot School of Software
Yunnan University
Kunming, China
{hezl, gaos, dongyy92, zwei}@ynu.edu.cn

2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3503-9744-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTAI56018.2022.00152

***Abstract*—The robustness and security of deep neural network (DNN) models have received much attention in recent years. In-depth research on adversarial example generation methods that make DNN models make wrong judgments and decisions will facilitate further research on more comprehensive and practical adversarial defense methods. Most existing adversarial example generation methods focus too much on attack performance and design adversarial noise at the pixel level, resulting in the generated adversarial examples with redundant noise and evident perturbations. In this paper, we try to find the well-designed perturbations at the feature-level and propose a novel deep reversible network-based imperceptible adversarial examples generation method called RIA. Experimental results show that RIA can obtain more natural adversarial examples without losing attack performance and reducing redundant noise based on well-designed feature maps. To the best of our knowledge, in the white-box attack method research, this work is the first attempt to directly add perturbations to feature maps and use an reversible network to generate adversarial examples based on the perturbed feature maps.**

***Index Terms*—Adversarial Feature Map, Adversarial Attack, Imperceptible Adversarial Example, Reversible Network**

I. Introduction

Various deep neural network (DNN) models [1, 2] have been widely used in computational vision-related fields, such as object detection [3], autonomous driving [4], etc. One concern, however, is that these deep learning models can be fragile. Slight alterations to the original image could lead to erroneous predictions or decisions, which could be fatal for safety-related applications such as autonomous driving. For example, self-driving cars misjudged road signs with graffiti, causing severe traffic accidents. Issues like this would pose a significant threat to public safety.

In recent years, more and more scholars have paid attention to the security and robustness of deep learning models. One of the common ideas is to conduct in-depth research on adversarial example [5] generation methods (i.e., adding subtle and imperceptible perturbations to the original images) to improve attack performance, thereby motivating more comprehensive and effective adversarial defensive methods.

Among the existing research on how to design adversarial examples, the most famous is the white-box attack method [6, 7], that is, generating adversarial examples when knowing the model structure, parameters, and even the training datasets to assess the model's vulnerability. For example, some gradient-based attacks [6, 7] calculate adversarial noise based on the gradient information of the target model and then apply the well-designed perturbations to the original images at the pixel-level to synthesize adversarial examples. Some intermediate layer-based attacks [8] make adversarial examples more transferable by utilizing the models' intermediate layers to calculate the adversarial perturbations.

(a) Original (b) Ours (c) PGD (d) MIFGSM (e) ILA

Fig. 1. Comparison of the visual performance of the adversarial examples generated by four different attack methods: (a) The original image, (b) Our RIA method, (c) PGD, (d) MIFGSM, and (e) ILA(Intermediate level attack).

However, the above methods suffer from two flaws: 1) The perturbation of adversarial samples is often directly added to the pixel-level, which means that adversarial examples are unnatural to the naked eyes and are not concealed. 2) Adding noise to the whole image may be redundant, and in real attack scenarios, adversarial examples may only perturb local regions. Many works in existing research [9, 10] have demonstrated the possibility of making adversarial examples only by perturbing local regions.

In this paper, we attempt to construct more imperceptible adversarial examples and reduce the redundancy of noise by combining a deep reversible residual network [11] and guidance information to perturb the original image from a feature space representing essential information. More specifically, the deep reversible residual network, consisting of feature extraction and recovery module, enables the interconversion between image and feature map. The proposed method can benefit from the transforming capability of the deep reversible network and obtain the adversarial examples from the well-disturb feature map. The guidance information, such as classification loss, can point out the direction of the noise optimization. Compared with the previous approaches, the proposed method gener-

*Corresponding author

2375-0197/22/$31.00 ©2022 IEEE
DOI 10.1109/ICTAI56018.2022.00152

ates the adversarial examples by perturbing the feature map directly. Empirically, the optimal perturbations in pixel-level transformed from the feature-level are concentrated on the critical areas. It has significantly guaranteed the image quality and invisibility of the adversarial examples, as shown in Fig. I. Our experiments on benchmark datasets and models show that the proposed method can generate adversarial examples with higher imperceptibility and ensure attack performance.

Our main contributions in this paper can be summarized as follows:

- We propose a novel invertible network-based adversarial example generation method called RIA. This method use reversible networks to extract the feature map and add well-designed perturbations directly to the feature map. The adversarial examples are generated from the perturbed feature maps.
- We conduct multiple experiments based on a series of benchmark datasets and models. The experimental results demonstrate that the adversarial examples generated from the proposed method are more natural than those generated by adding perturbations directly to the original image using the additive transform.
- In real-world scenarios, malicious attacks are often elaborately designed. Our method can generate adversarial examples that are difficult to be perceived by the naked eye but still have superior attack performance, providing better inspiration for further exploring the robustness and defense measures of DNN models under such subtle malicious noise.

II. Related Work

In white-box attack settings, attackers can access the whole information about the target model, including parameters, model structure, and even the training datasets. Many methods can generate adversarial examples to attack the target model successfully. The most relevant to our work are pixel-level attacks, and feature-level attacks.

A. Pixel-level Attack

The Fast Gradient Sign Method (FGSM) [6] is a typical pixel-level attack method that generates adversarial examples in just one update step. BIM [12] extends FGSM to generate adversarial examples through multi-step updates. PGD [7] is similar to BIM [12] except that it randomly selects an initial point near a benign example as the starting point for an iterative attack.

B. Feature-level Attack

In order to enhance the attack capability of adversarial examples, many studies have focused on perturbing the intermediate layer by the guidance of pixel-level noise. Some approaches generate perturbations to interfering with the activation of the intermediate layer. These include [13], which generates more spurious activations by generating a universal perturbation to disturb the activation of the middle layer. FDA [14] perturbs each layer of the feature map by adding perturbations on the original image to change the value of the activations in the feature space. Through random mask and gradient aggregation, FIA [15] guides the generation of adversarial examples by disrupting aggregated gradients obtained from images processed differently.

In order to clearly position our investigation and highlight our unique features, we analyze the differences between our research and above existing research as follows:

- Unlike pixel-level attack methods that directly add perturbations at the pixel-level using additive transformations, RIA uses invertible transformations to obtain adversarial examples from perturbed feature maps.
- Unlike the related research on feature-level attacks, RIA does not need to add global perturbation at the pixel-level to destroy the feature map. However, it perturbs the feature map to reduce redundant noise while maintaining the attack performance.

III. Preliminary

A. Adversarial Attack

Given a clean image x with true label y and a targeted label y_{adv}, a well-trained classifier $f : f(x) \rightarrow y \in \{1, 2, ..., K\}$, can map x to it's corresponding label y correctly. The goal of adversarial attack is to find an adversarial example x' of clean image x by solving an optimization problem $L_{adv}(\cdot)$, which leads $f(x') \neq f(x)$ for untargeted attack or $f(x')$ = y_{adv} for target attack. For adversarial loss $L_{adv}(\cdot)$, the following cross-entropy loss is selected:

$$L_{adv} = \begin{cases} log(P_y(x')) & \text{for untargeted attack,} \\ -log(P_{yadv}(x')) & \text{for targeted attack.} \end{cases} \quad (1)$$

Where $P(\cdot)$ is the probability output (softmax on logits) of the target model f w.r.t class y or y_{adv}.

B. Deep Invertible Network

The deep reversible network(I-revnet) [11] proposed an architecture that enables the interconversion of original image and feature map. I-revnet realizes the interconversion between feature map and original image by utilizing feature extraction and restoration module. They all consist of two branches connected by residual modules G and F; during forward propagation, these two branches' results are alternated, as shown in Fig. 2.

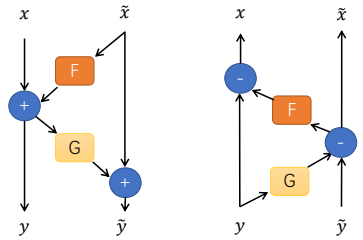


Fig. 2. Reverse block structure: G and F represent convolutional operation, each block has two branches.

An initial input is split into two sublayers of equal size, thanks to the following step according to the channel dimension:

$$(x_0, \widetilde{x}_0) = S(X), \tag{2}$$

where X represents the original image and $(x_0, \widetilde{x}_0)$ indicates the input of two branches of the network. $S(\cdot)$ represents the split operation, which firstly performs downsampling to reduce the resolution and increase the number of channels to ensure the integrity of information and then facilitates upsampling and dimension reorganization in the reverse process.

After sufficient steps, it recombines the two branches' output through the inverse operation $S^{-1}(\cdot)$ to obtain the feature map. The operations can be defined as:

$$\begin{cases} y_j = x_j + F(\widetilde{x}_j) \\ \widetilde{y}_j = \widetilde{x}_j + G(y_j), \end{cases} \tag{3}$$

$$X^{feature} = S^{-1}(y_n, \widetilde{y}_n), \tag{4}$$

where y_j and $\widetilde{y}_j$ is the output of two branches, j represents the layer j of the network. The $X^{feature}$ is the feature map of the raw image, and n is the last layer of the feature extraction module.

Compared with the feature extraction process introduced before, feature restoration is a corresponding reverse process. The whole process of restoration is defined as:

$$(y_n, \widetilde{y}_n) = S(X^{feature}), \tag{5}$$

$$\begin{cases} \widetilde{x}_j = \widetilde{y}_j - G(y_j) \\ x_j = y_j - F(\widetilde{x}_j), \end{cases} \tag{6}$$

$$x = S^{-1}(x_0, \widetilde{x}_0), \tag{7}$$

where $(y_0, \widetilde{y}_0)$ is the two branches obtained by split operation through the feature map.

IV. METHOD

A. Overview

We aim to find optimal perturbations at the feature-level and restore the perturbed feature map to corresponding spatial one. The framework of our proposed method is illustrated in Fig. 3. The whole workflow can be divided into the following three parts:

- **Feature Extraction** The clean feature map is obtained from the original image through the feature extraction module, and the step is denoted by $E(\cdot)$ in the following.
- **Adversarial Feature Map Optimization** RIA optimize perturbations by constantly increasing the adversarial loss to obtain the adversarial feature map, denoted as $op(\cdot)$.
- **Feature Restoration** RIA use this step to reverse the perturbed feature map to the corresponding image. This step is described as $R(\cdot)$ in the following.

Formally, we denote our final objective as:

$$\arg\max_{N} L_{adv}(x', y), where\ x' = R(op(E(x), N)),$$

$$s.t.\ ||x' - x||_\infty < \epsilon \tag{8}$$

where N is the initial noise draw from the normal distribution, RIA increased the adversarial loss between the restored image and the raw image by continuously optimizing the noise.

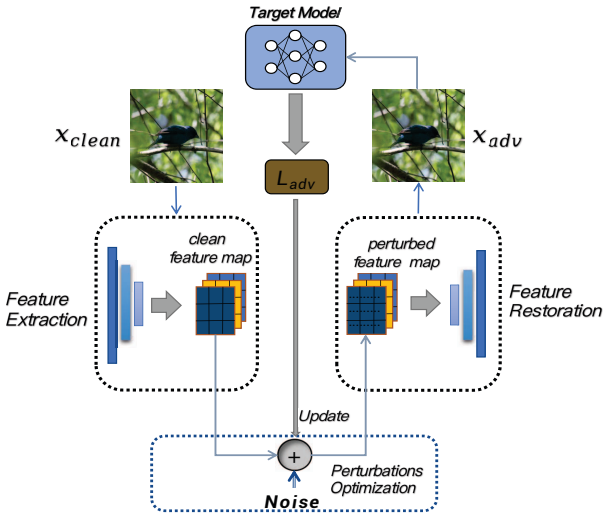


Fig. 3. An overview of the proposed method. Feature Extraction is a CNN module that extracts the feature map from the image. Feature Restoration is a reversible module that restores the image from the corresponding feature map. Noise is randomly initialized perturbations.

B. Feature Extraction

RIA use $E(\cdot)$ to get the feature map to be optimized. Given a clean image x, the corresponding feature map $X^{feature}_{clean}$ can be obtained by the following step concretely:

$$(x_0^{clean}, \widetilde{x}_0^{clean}) = S(X^{clean}), \tag{9}$$

$$\begin{cases} y_j^{clean} = x_j^{clean} + F(\widetilde{x}_j^{clean}) \\ \widetilde{y}_j^{clean} = \widetilde{x}_j^{clean} + G(y_j^{clean}), \end{cases} \tag{10}$$

$$X^{feature}_{clean} = S^{-1}(y_n^{clean}, \widetilde{y}_n^{clean}). \tag{11}$$

C. Adversarial Feature Map Optimization

In order to obtain the adversarial feature map, RIA first initialize the noise δ_{init} by drawing from normal distribution $\mathcal{N} \sim (\mu, \sigma^2)$ with the same size as the feature map. Then it can get the feature map with the noise:

$$X^{feature}_{adv} = X^{feature}_{clean} + \delta_{init}. \tag{12}$$

To mislead the target model, RIA use Adam optimizer to optimize the δ_{init} by increasing $L_{adv}(x^{adv}, x)$ continuously. Moreover, RIA uses cross-entropy loss as the adversarial loss. The malicious noise δ_{adv} is calculated as the following optimization process:

$$\delta_{adv} = op(\delta_{init}), \tag{13}$$

where $op(\cdot)$ is an iterative optimization process by adversarial loss. After getting the final adversarial noise, the adversarial feature map can be obtained:

$$X_{adv}^{feature} = X_{clean}^{feature} + \delta_{adv}. \tag{14}$$

D. Feature Restoration

To reverse the feature map to an image, RIA use $R(\cdot)$ corresponding to $E(\cdot)$. Once the optimal feature map $X_{adv}^{feature}$ is gained, RIA can build the adversarial example x^{adv} by following steps:

$$(y_n^{adv}, \widetilde{y}_n^{adv}) = S(X_{adv}^{feature}), \tag{15}$$

$$\begin{cases} \widetilde{x}_j^{adv} = \widetilde{y}_j^{adv} - G(y_j^{adv}) \\ x_j^{adv} = y_j^{adv} - F(\widetilde{x}_j^{adv}), \end{cases} \tag{16}$$

$$x^{adv} = S^{-1}(x_0^{adv}, \widetilde{x}_0^{adv}). \tag{17}$$

For clarity, we present the whole algorithm of RIA is listed in Alg. 1, which could help readers to re-implement our method step-by-step.

Algorithm 1 RIA Attack

Requires: A clean image x, a pre-trained feature extraction module $E(\cdot)$ and feature restoration module $R(\cdot)$, a target model $f(\cdot)$, a initial perturbation δ_{init}, and a maximum number of iterations N.

Initialization:x^{adv} = x, $\delta = \delta_{init}$;
for $i = 1$ to N **do**
  Extract the feature map $X_{adv}^{feature}$ from x^{adv} by $E(\cdot)$ in Eq. 11;
  Obtain the perturbed feature map $X_{adv}^{feature}$ based on the current perturbation δ in Eq. 12;
  Reverse the perturbed feature map $X_{adv}^{feature}$ to a perturbed image x^{adv} in Eq. 17;
  Optimize the δ by maximizing the $L_{adv}(x^{adv}, x)$ in Eq. 2;
  if $f(x^{adv} \neq f(x))$ **then**
    break
  end if
end for

V. EXPERIMENTS

A. Experimental Setup

- **DataSets.** We evaluate the performance of RIA on three benchmark datasets, namely CIFAR-10, SVHN , and ImageNet-1K. For CIFAR-10, we selected the entire test set; for SVHN, we randomly selected 10k images from test set; for ImageNet-1k, we randomly selected 2k images with correct classification to verify the proposed method.
- **Implementation details.** In the proposed method, Adam is selected as the optimizer; the perturbation budget is set to 8/255 under L_∞. The initial noise drawn from the normal distribution is obtained by Xavier, we set the value of $gain$ = 0.01. The maximum number of iterations is 100, and the iteration is stopped after the adversarial example is obtained. Among the comparison methods we chosen, ϵ is the same as RIA for PGD [7], BIM [12], MI-FGSM [16], and ILA [8]. These comparative experiments were done using the adversarial attack tool library torchattacks [1]. As for ILA [8], the last avgpool is the attacked layer. We selected ResNet50, DenseNet161, GoogLenet, and Inception-V3 as target models; they achieved 93.65%, 94.05%, 92.84%, 93.74% test accuracy on cifar-10, 94.44%, 95.12%, 95.52%, 93.22% on SVHN, and 76.13%, 77.13%, 69.77%, 77.29% on ImageNet. All the experiments are conducted on NVIDIA RTX 3080 GPU with 10GB memory.
- **Evaluation metrics.** We selected attack success rate (ASR), Structural Similarity(SSIM) [17], Peak Signal-to-Noise Ratio(PSNR), and Deep Image Structure and Texture Similarity(DISTS) [18] to assess RIA.

B. Attack Success Rate

In this part, we will evaluate the attack ability of different white-box attacks.

TABLE I
ATTACK SUCCESS RATE. ALL THE RESULTS ARE UNDER $L_\infty = 8/255$ OF UNTARGETED ATTACK.

DataSets	Attacks	ResNet50	DenseNet161	GoogLeNet	Inception_V3
Cifar-10	PGD	99.77	98.16	100	97.92
	BIM	99.77	98.17	100	97.94
	MI-FGSM	99.57	97.93	100	97.82
	ILA	99.90	94.85	99.90	91.67
	Ours	99.62	94.30	**100**	92.95
SVHN	PGD	98.56	98.17	97.04	95.07
	BIM	98.56	98.17	97.04	95.10
	MI-FGSM	98.49	97.85	96.66	94.38
	ILA	99.88	99.81	99.51	99.31
	Ours	99.69	**99.88**	**99.55**	99.15
ImageNet	PGD	100	100	100	99.75
	BIM	100	100	100	99.80
	MI-FGSM	100	100	100	99.80
	ILA	100	100	100	99.55
	Ours	99.95	99.90	**100**	96.90

Table. I shows the attack performance of five attack approaches. Our proposed method is effective on different benchmark datasets and models, and it proves that by querying the gradient information of the target model and adding perturbations directly to the feature map is effective for building adversarial examples.

C. Imperceptibility

In this subsection, we evaluated the performance of five different attack methods on three image quality metrics. Generally, PSNR (Peak Signal to Noise Ratio) is used to measure the degree of distortion of an image, with larger values indicating less distortion. SSIM (structural similarity

[1]https://github.com/Harry24k/adversarial-attacks-pytorch

TABLE II
IMPERCEPTIBLE ASSESSMENT ON CIFAR-10 ACROSS DIFFERENT DATASETS AND METHODS. ALL OF THE RESULTS ARE UNDER L_∞=8/255 OF UNTARGETED ATTACK.

Attacks	ResNet50			DenseNet161			GoogLenet			Inception_V3		
	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS
PGD	0.9408	30.4991	0.0972	0.9429	30.6808	0.0973	0.9474	31.1049	0.0867	0.9522	31.6940	0.0830
BIM	0.9409	30.5023	0.0973	0.9429	30.6776	0.0974	0.9475	31.1088	0.0866	0.9522	31.6921	0.0831
MI-FGSM	0.9305	29.8219	0.1066	0.9279	29.6119	0.1117	0.9338	30.1978	0.0981	0.9253	29.6767	0.1088
ILA	0.9383	30.2525	0.1051	0.9260	29.5611	0.1248	0.9602	32.1383	0.0766	0.9401	30.5899	0.0981
Ours	**0.9718**	**34.2643**	**0.0639**	**0.9672**	**33.6060**	**0.0710**	**0.9804**	**35.8678**	**0.0492**	**0.9713**	**34.5781**	**0.0615**

TABLE III
IMPERCEPTIBLE ASSESSMENT ON SVHN ACROSS DIFFERENT DATASETS AND METHODS. ALL OF THE RESULTS ARE UNDER L_∞=8/255 OF UNTARGETED ATTACK.

Attacks	ResNet50			DenseNet161			GoogLenet			Inception_V3		
	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS
PGD	0.9215	32.6885	0.1385	0.9156	33.3183	0.1451	0.8857	30.8637	0.1831	0.8731	30.5213	0.1817
BIM	0.9215	32.6882	0.1385	0.9156	32.3176	0.1451	0.8857	30.8626	0.1831	0.8731	30.5222	0.1817
MI-FGSM	0.9114	32.1458	0.1479	0.9066	31.8551	0.1532	0.8777	30.6266	0.1877	0.8700	30.4146	0.1831
ILA	0.9235	32.6346	0.1478	0.9041	31.7292	0.1749	0.8857	30.5543	0.1903	0.8452	29.5350	0.2017
Ours	**0.9795**	**39.5456**	**0.0608**	**0.9802**	**39.6539**	**0.0592**	**0.9756**	**38.8568**	**0.0670**	**0.9744**	**38.9537**	**0.0686**

TABLE IV
IMPERCEPTIBLE ASSESSMENT ON IMAGENET ACROSS DIFFERENT DATASETS AND METHODS. ALL OF THE RESULTS ARE UNDER L_∞=8/255 OF UNTARGETED ATTACK.

Attacks	ResNet50			DenseNet161			GoogLenet			Inception_V3		
	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS	SSIM	PSNR	DISTS
PGD	0.8965	32.7266	0.1072	0.8939	32.5210	0.1160	0.8922	32.6143	0.1126	0.9020	33.0347	0.1075
BIM	0.8965	32.7268	0.1071	0.8939	32.5200	0.1161	0.8922	32.6118	0.1126	0.9020	33.0318	0.1074
MI-FGSM	0.8302	30.5998	0.1438	0.8282	30.4673	0.1521	0.8283	30.5487	0.1465	0.8320	30.6343	0.1446
ILA	0.9123	33.0199	0.1115	0.8830	31.8470	0.1669	0.9300	34.4472	0.0805	0.9323	34.6594	0.0808
Ours	**0.9721**	**35.8814**	**0.0419**	**0.9729**	**35.9983**	**0.0411**	**0.9709**	**35.7923**	**0.0417**	**0.9729**	**36.1336**	**0.0390**

index) is used to measure the similarity of two images, which is closer to the evaluation index of the human visual system. DISTS(Deep Image Structure and Texture Similarity) is used to evaluate the difference in human perception of two images by measuring texture similarity.

Table. II, Table. III, and Table. IV show the experimental results of several different methods on CIFAR-10, SVHN, and ImageNet, respectively. These results show that the adversarial examples generated by RIA significantly improved PSNR and SSIM, and effectively reduced DISTS. These adversarial examples have slight distortion and are more similar to the original images, which make them more imperceptible to the naked eye.

D. Analysis

We analyze why adversarial examples generated by RIA are more imperceptible than other methods that design pixel-level perturbations.

The feature layer aggregates the most important information of the original image, and a subtle interference may cause target model to produce wrong predictions. RIA initialize a noise drawn from a normal distribution with minimal means, and the optimized perturbations are also tiny when the attack is successful. Based on the tiny perturbation in the feature-level, the corresponding adversarial example reversed from the perturbed feature map also has a little noise. As shown in Fig. 4, the pixel-level perturbations are much smaller than other methods.

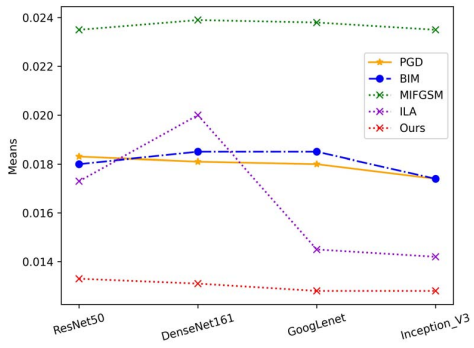
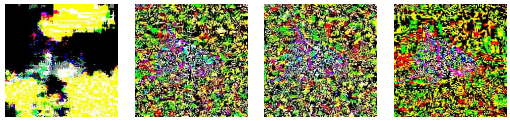


Fig. 4. Comparison of the noise's means generated by different methods in pixel-level.

There is some mapping relationship between feature map

and original image. Since the perturbation is focused on the essential features, the corresponding adversarial examples will reduce redundant noise. Most of these maliciously crafted examples generated by RIA have perturbations that concentrate on the critical regions. Fig. 5 shows the noise of different attack methods on ImageNet. In the proposed method, the prominent noise is only generated in a portion of the areas and channels, and the black area is noiseless.



(a) Ours (b) PGD (c) MIFGSM (d) ILA

Fig. 5. Niose of adversarial example generated by different attack method.

VI. Conclusion

Adversarial example generation methods are critical for simulating well-crafted malicious attacks, which can further enlighten more effective adversarial defense methods. However, well-designed malicious attacks are often highly stealthy and efficient, such that adversarial examples must be considered in terms of attack performance and stealth. Most existing adversarial example generation methods focus too much on attack performance and thus design adversarial noise at the pixel level, resulting in the generated adversarial examples with redundant noise and being accessible to perceptible. This work tries to directly disturb the feature map to generate adversarial examples with high concealment. Specifically, we proposed a reversible network-based attack method named RIA, which calculates perturbations in the feature-level straightforward and obtains the corresponding adversarial examples from the well-designed feature map with suitable noise. Extensive experiments across different datasets and pre-trained models have been conducted to validate the effectiveness of the proposed method. The results show that RIA significantly improves the image quality and invisibility of the adversarial examples while ensuring its attack ability.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62162067 and 62101480, in part by the Yunnan Province Science Foundation for Youths under Grant No.202005AC160007 and the fundamental research of Yunnan Province under Grant No.202001BB050076, in part by the Applied Basic Research Foundation of Yunnan Province under Grant 202201AT070156, in part by the Fund project of Yunnan Province Education Department No.2022j0008.

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[4] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR*, 2016.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[8] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *ICCV*, 2019, pp. 4733–4742.

[9] J. Chen, H. Zheng, H. Xiong, R. Chen, T. Du, Z. Hong, and S. Ji, "Finefool: A novel dnn object contour attack on image recognition based on the attention perturbation adversarial technique," *Computers & Security*, p. 102220, 2021.

[10] Q. Liao, X. Wang, B. Kong, S. Lyu, Y. Yin, Q. Song, and X. Wu, "Fast local attack: Generating local adversarial examples for object detectors," in *IJCNN*, 2020, pp. 1–8.

[11] J. Jacobsen, A. W. M. Smeulders, and E. Oyallon, "i-revnet: Deep invertible networks," in *ICLR*, 2018.

[12] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.

[13] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 2452–2465, 2019.

[14] A. Ganeshan, V. BS, and R. V. Babu, "Fda: Feature disruptive attack," in *ICCV*, 2019, pp. 8069–8079.

[15] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, "Feature importance-aware transferable adversarial attacks," in *ICCV*, 2021, pp. 7639–7648.

[16] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018, pp. 9185–9193.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, pp. 600–612, 2004.

[18] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE transactions on pattern analysis and machine intelligence*, 2020.