







Interpreting Universal Adversarial Example Attacks on Image Classification Models

Yi Ding , Member, IEEE, Fuyuan Tan , Ji Geng, Zhen Qin , Mingsheng Cao ,
Kim-Kwang Raymond Choo , Senior Member, IEEE, and Zhiguang Qin , Member, IEEE

Abstract—Mitigating adversarial deep learning attacks remains challenging, partly because of the ease and low cost in carrying out such attacks. Therefore, in this article, we focus on the understanding of universal adversarial example attack on image classification models. Specifically, we seek to understand the difference(s) between adversarial examples in two adversarial datasets (DAMageNet and PGD dataset) and clean examples in ImageNet learned by the classification model, and whether we can use such findings to resist adversarial example attacks. We also seek to determine if we can retrain a discriminator to discriminate whether the input image is an adversarial example, using adversarial training. We then design a number of experiments (e.g., class activation map (CAM) analysis, feature map analysis, feature maps/filters changing, adversarial training, and binary classification model) to help us determine whether the universal adversarial dataset can be successfully used to attack the classification model. This, in turn, contributes to a better understanding of adversarial defenses over pretrained classification model from an interpretation perspective. To the best of our knowledge, this work is one of the earliest works to systematically investigate the interpretation of universal adversarial example attack on image classification models, both visually and quantitatively.

Index Terms—Adversarial defense, adversarial example, deep learning, interpretability

1 INTRODUCTION

DEEP learning methods have been widely adopted in a broad range of domains, such as image classification [1], [2], image security [3], [4], medical aided diagnosis [5], [6] and facial recognition [7], [8]. In some of these application domains, the performance of deep learning models has reportedly exceed those of human. In practice, many classification models are carried out in a cloud-based environment, and thus are potentially subject to different attacks such as

- Yi Ding is with the Network and Data Security Key Laboratory of Sichuan Province, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China, and also with Ningbo WebKing Technology Joint Stock Co., Ltd, Ningbo, Zhejiang 315000, China. E-mail: yi.ding@uestc.edu.cn.
- Fuyuan Tan, Ji Geng, Zhen Qin, Mingsheng Cao, and Zhiguang Qin are with the Network and Data Security Key Laboratory of Sichuan Province, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China. E-mail: tanfuyuan@std.uestc.edu.cn, ljgeng, qinzhen, cms, qinzg@uestc.edu.cn.
- Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631 USA. E-mail: raymond.choo@fulbrightmail.org.

Manuscript received 27 June 2021; revised 30 May 2022; accepted 22 August 2022. Date of publication 29 August 2022; date of current version 11 July 2023.

This work was supported in part by the National Natural Science Foundation of China under Grants 62076054, 62072074, 62027827, 61902054, and 62002047, in part by the Frontier Science and Technology Innovation Projects of National Key R&D Program under Grant 2019QY1405, in part by the Sichuan Science and Technology Innovation Platform and Talent Plan under Grants 2020JDJQ0020 and 2022JDJQ0039, in part by the Sichuan Science and Technology Support Plan under Grants 2020YFSY0010, 2022YFQ0045, 2022YFS0220, 2019YJ0636, and 2021YFG0131, in part by the Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China under Grants ZYG-X2021YGLH212 and ZYGX2022YGRH012. The work of Kim-Kwang Raymond Choo was supported only by Cloud Technology Endowed Professorship. (Corresponding author: Zhen Qin.)

Digital Object Identifier no. 10.1109/TDSC.2022.3202544

adversarial example attacks on the classification model. The adversarial example attack is implemented by adding perturbations on the original images, which usually cannot be identified using our naked eyes. Then, these adversarial examples are used to misguide the classification model to make an error classification. As shown in Fig. 1, the classification model is placed on the server in the cloud-based environment, and the user can utilize the classification service from their client device by calling the application's interface. However, the end user can easily input an adversarial example to attack the classification model on the cloud server.

A number of researchers have explored how to generate adversarial examples. Examples include the L-BFGS Attack [7] (that uses L-BFGS method to solve the general targeted problem when generating adversarial examples), Fast Gradient Sign Method (FGSM) [8] (that performs one step gradient update along the direction of the sign of gradient at each pixel), Basic Iterative Method (BIM) and Iterative Least-Likely Class Method (ILLC) [9] (that extend FGSM by performing optimization for multiple iterations), Jacobian-based Saliency Map Attack (JSMA) [10], DeepFool [11] (that finds the closest distance from the original input to the decision boundary of adversarial examples), C&W's Attack [12], Zeroth Order Optimization (ZOO) [13] (that can be directly deployed in a black-box attack without model transferring), Universal Perturbation [14], and One Pixel Attack [15] (that generates adversarial examples by only modifying one pixel).

In addition to attacking the classification models, there have been efforts in designing approaches to defend against adversarial example attacks. Buckman et al. [16], for example, demonstrated that thermometer code discretization and one-hot code discretization of real-valued inputs to a model significantly improve its robustness to adversarial attacks.

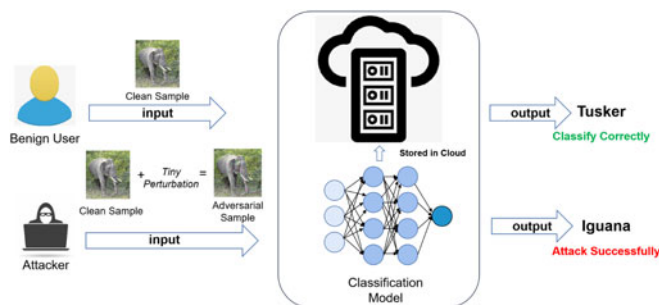


Fig. 1. Adversarial example attack on cloud-based classification model: An example.

Ma et al. [17] proposed a method of adopting local intrinsic dimensionality to characterize properties of adversarial examples, and Guo et al. [18] proposed five input transformations to resist adversarial examples. In a different work, Dhillon et al. [19] introduced randomness into the evaluation of a neural network to defend against adversarial examples. Xie et al. [20] proposed to defend against adversarial examples by adding a randomization layer before the input to the classifier. Song et al. [21] proposed using a PixelCNN generative model to project a potential adversarial example back onto the data manifold before feeding it into a classifier. Madry et al. [22] studied the adversarial robustness of neural networks through the lens of robust optimization. However, these approaches generally achieve good performance for specific situation(s) only. For example, most defensive approaches are mainly designed to statically resist one specific type of adversarial example attack, and its defensive ability is extremely limited. In other words, these approaches cannot solve general adversarial example attacks. This is due to a lack of understanding of why generated adversarial examples can successfully attack the classification model. In other words, the deep learning algorithm is a “black box”, and we lack the interpretability of how decision is made using the deep learning algorithm.

Several interpretable deep learning algorithms have also been proposed [23], [24], [25], [26], [27], [28], [29], [30], [31], [32]. Examples include CNNVis [23], Lucid [24], LIME (Local Interpretable Model-Agnostic Explanation) [25], CAM (Class Activation Mapping) [26], Grad-CAM [27], and Explanatory Graph Representation [28]. Also, DLIME (Deterministic LIME) [29] and Decision Tree [30] are two other examples, which employ interpretable models to deconstruct uninterpretable models. However, designing interpretable deep learning algorithms for adversarial example attacks is an understudied area. This is the gap we seek to address in this paper.

Specifically, we focus on the explainability of adversarial example attacks, by interpreting why a classification model can be successfully attacked by the adversarial examples. As a case study, the DAmAgeNet [33] is adopted as the adversarial example to attack the classification model evaluated on the ImageNet dataset [34], since this is the first universal adversarial dataset and has defeated many models trained in ImageNet. Besides, we introduce a popular model-specific adversarial attack method, projected gradient descent (PGD) [50], to generate 50000 adversarial examples based on ImageNet validation set for conducting ablation experiments with the DAmAgeNet dataset. Using the pretrained ResNet50 as

the classification model, we seek to answer the following two research questions (RQs).

RQ 1: What is/are the difference(s) between the adversarial example and the clean image learned by the classification model, and can the findings be used to resist the adversarial example attack?

To help us answer the above question, we systematically design various experiments to explore the difference(s) learned by the classification model. By evaluating the CAM analysis and feature maps analysis, we find that the salient region and the neuron has a different response to the adversarial example from the first layer are different. This finding partially explains why the adversarial example can be used to attack the classification model. Furthermore, based on the findings in the interpretable adversarial learning, we propose the following hypothesis: “If the weights and channels of the neurons in the classification network can be changed to avoid learning the adversarial feature, then the adversarial example attack can be resisted by improving the classification model.” We validate our hypothesis from two aspects. First, we mask the channels which are sensitive to the adversarial feature. Second, the weight of these channels is adjusted to avoid learning adversarial features in a fine-grained way.

RQ 2: Can we retrain a pretrained model to improve the precision and robustness by adopting the adversarial training, and can we train a discriminator to discriminate whether the input image is an adversarial example?

Based on the pretrained ResNet50, the adversarial examples from the DAmAgeNet and PGD dataset are adopted as the training dataset to implement the adversarial training, respectively. Moreover, an additional classification network is trained to classify the clean image and the adversarial example. Extensive experimental results show that it is hard to retrain a discriminator to discriminate the adversarial example when dealing with large datasets, such as ImageNet.

To the best of our knowledge, this work is one of the earliest works to systematically investigate how one can interpret the universal adversarial example attack on image classification models, both visually and quantitatively. In other words, our findings shed some light on why the universal adversarial dataset can be successfully used to attack the classification model. In doing so, the findings will also inform future design of mitigation strategies.

The rest of the paper is organized as follows. In the next section, we will introduce the related literature, prior to presenting preliminary knowledge in the third section. We then describe our experiments and analysis in the fourth section. Finally, section five concludes this article.

2 RELATED WORK

2.1 Adversarial Defenses

One of the current approaches to mitigate adversarial example attacks is to discriminate the adversarial example. Meng et al. [35] proposed MagNet, which is a defending framework to protect the neural network from adversarial example attack. It includes one or more separate detector networks and a reformer network. The detector networks learn to differentiate between clean and adversarial examples by approximating the manifold of clean examples. Inspired by the randomness in cryptography, MagNet was further improved

due to the diversity. The reformer network moves adversarial examples towards the clean examples, which is an effective way to correctly classify adversarial examples with small perturbation. MagNet can reportedly defend against most state-of-the-art attacks in both black-box and white-box scenarios, without impacting on false positive rate on clean examples. Feinman et al. [36] asked whether a DNN can distinguish adversarial examples from the clean examples. They investigated the confidence of network on adversarial examples by calculating the Bayesian uncertainty estimates, and realizing density estimation in the subspace of features. The experimental results demonstrated that it achieve a good performance for different architectures and attacks. Their findings reported that 85-93% ROC-AUC can be achieved on a number of standard classification tasks. Kimin Lee et al. [37] proposed a simple yet effective method for detecting any adversarial examples, which can be used on most pre-trained softmax neural classifiers. They obtained the Gaussian distributions for class condition, which result in a confidence score based on the Mahalanobis distance. Experiments showed that the proposed method achieves a good performance on detecting adversarial examples, and the network becomes more robust. Weilin Xu et al. [38] proposed a new strategy, feature squeezing, which can be used to improve DNN models by detecting adversarial examples. Feature squeezing reduced the search space available to the adversarial attack by aggregating examples belonging to many different feature vectors into a single example. By comparing a DNN model's prediction on the original input, feature squeezing detected adversarial examples with high accuracy and few false positives. These simple strategies were inexpensive and complementary to other defenses, and can be combined in a detection framework to achieve high detection rates against adversarial attacks. Pang et al. [32] proposed the AdvMind, a new class of estimation models that infer the adversary intent of black-box adversarial attacks in a robust and prompt manner. Specifically, to achieve robust detection, AdvMind accounted for the adversary adaptiveness such that her attempt to conceal the target will significantly increase the attack cost. And AdvMind proactively synthesized plausible query results to solicit subsequent queries from the adversary that maximally expose her intent to achieve prompt detection.

However, these existing adversarial defenses methods don't place their focus on analyzing the difference between the clean example and the adversarial example, and the internal feedback of the learning network. They only provide the solution for detecting the specific type of adversarial example attack under certain situations. When facing with the large datasets or different task environment, these adversarial defenses methods usually lost their effective.

2.2 Interpretability in Deep Learning

Fong et al. [39] proposed a general framework for learning different kinds of explanations for most black-box algorithms. Moreover, the framework tried to find the part of an image where is most responsible for a classifier decision. Different with previous works, the proposed method was model-agnostic and testable because it is grounded in explicit and interpretable image perturbations. Zhang et al. [40] proposed a method to modify a traditional convolutional neural network (CNN) to an interpretable CNN, to clarify

knowledge representations in high conv-layers of the CNN. In an interpretable CNN, each filter in a high conv-layer represented a specific object part. The interpretable CNNs used the same training data as normal CNNs without requesting any additional annotations of object parts or textures for supervision. Experiments have shown that filters in an interpretable CNN are more semantically meaningful than those in a traditional CNN. Zhou et al. [41] described the Network Dissection, a method to interpret networks by providing meaningful labels to their individual units. The proposed method quantified the interpretability of CNN representations by evaluating the alignment between individual hidden units and visual semantic concepts. Their results highlighted that interpretability is an important property of deep neural networks, which can provide new insights into what hierarchical structures can learn. Chen et al. [42] introduced a deep network architecture, Prototypical Part Network (ProtoPNet), to interpret in a similar way: the network learned the image by finding prototypical parts, and combined evidence from the prototypes to make a final classification. The experiments showed that ProtoPNet can achieve comparable accuracy with its non-interpretable counterpart. Moreover, the ProtoPNet provided an additional ability of interpretability where other interpretable deep models were limited. Zhang et al. [31] proposed i-Algebra, a first-of-its-kind interactive framework for interpreting the DNNs. At its core is a library of atomic, composable operators, which explain model behaviors at varying input granularity, during different inference stages, and from distinct interpretation perspectives.

Although many researches focus on the interpretable for deep learning models and implement lots of applications based on interpretability, only a few researches attempt to discuss the interpretability of adversarial example attack.

2.3 Interpretable Adversarial Defenses

Ross et al. [43] evaluated the effectiveness of defenses by differentially penalizing the degree to which small changes in inputs can change model predictions. They found that neural networks, which are trained with gradient regularization, show robustness to transferred adversarial examples. Moreover, the experiments demonstrated that regularizing input gradients make them more naturally interpretable as rationales for model predictions. And they concluded this work by discussing this relationship between interpretability and robustness in deep neural networks. Tao et al. [44] proposed a novel adversarial example detection technique for face recognition models from interpretability perspective. This work identified a novel bi-directional correspondence inference between attributes and internal neurons to locate neurons which are sensitive for individual attributes. Results showed that proposed method can achieve state-of-the-art performance, 94% detection accuracy for 7 different kinds of attacks with 9.91% false positives on original inputs. Ma et al. [45] analyzed the internal structure of DNN models under various attacks and proposed two common exploitation channels: the provenance channel and the activation value distribution channel. Then They further proposed a novel technique to extract DNN invariant and to adopt them to perform adversarial example detection in run-time environment. Their experimental results showed that the proposed method can effectively detect all common attacks with limited false

positives. They also compared the proposed method with three state-of-the-art methods, which includes the Local Intrinsic Dimensionality based method, denoiser based methods, and the prediction inconsistency-based approach. Their experiments showed the promising results. Based on the interpretability of adversarial example generation, Shahfahi et al. [46] presented an algorithm that eliminated the overhead cost of generating adversarial examples. This method was implemented by recycling the gradient information computed when updating model parameters. Their “free” adversarial training algorithm achieved comparable robustness to PGD adversarial training on the CIFAR-10 and CIFAR-100 datasets with negligible additional cost compared to natural training, and can be up to 7-30 times faster than other famous adversarial training methods.

3 PRELIMINARY

3.1 Adversarial Training

Adversarial training is an effective way to enhance the robustness of neural networks. In the process of adversarial training, the clean examples and the adversarial examples (the change in the adversarial example is small, but it can cause misclassification for the classification model) will be combined as the training dataset, and then the neural network is retrained to adapted to this change. The new trained network holds the potential to become more robust to the adversarial example. The adversarial training can be formulated as solving a robust optimization problem

$$\min_{\theta} \mathbb{E}_{(x,y) \sim D} \left[\max_{\delta \in S} L(f(x + \delta; \theta), y) \right] \quad (1)$$

where $f(x; \theta)$ represents the parameterized neural network with weights θ ; the input-output pair (x, y) indicates the example from the training set D ; δ denotes the adversarial perturbation and L is the chosen loss function, e.g., cross-entropy loss. S denotes a norm constraints.

The symbol \max represents to find the perturbations which maximize the loss function. It means to confuse the neural network as much as possible by adopting the perturbation. The symbol \min indicates to minimize the optimization of the neural network. It means, when the perturbations are fixed, the neural network model is trained to minimize the loss function. In other words, the neural network is trained to become more robust to these perturbations.

3.2 Class Activation Map

The class activation map (CAM) is a powerful technique used in computer vision for classification tasks. It allows us to inspect the image to be categorized and understand which parts/pixels of that image have contributed more to the final output of the model. The CAM algorithm implements the global average pooling before the final output layer on the CNN architecture. It can enhance the visual explanation of the deep learning model. The CAM highlights the class-specific discriminative regions, and indicates the significance part in the image used for the classification. Through supplementary analyzing the region where the CNN model is concentrated, it provides the interpretability of how the black-model make a decision. The final

classification score S^c for class c can be expressed as a linear combination of its global average pooled feature maps A^k :

$$S^c = \sum_k \omega_k^c \cdot \sum_i \sum_j A_{ij}^k \quad (2)$$

The class-specific salient map M^c can be generated by:

$$M_{ij}^c = \sum_k \omega_k^c \cdot A_{ij}^k \quad (3)$$

where M_{ij}^c directly correlates with the importance of a particular spatial location (i, j) for class c . When up-sampling the activation map to the size of input image, it can show the interested region of the input image focused by the learning network to predict a label.

4 EXPERIMENT

4.1 Dataset Description

The ImageNet validation set is a universal dataset, which contains 50000 images with 1000 classes. It is usually used for image classification tasks. In this paper, the ImageNet validation set is mainly adopted as the clean examples while the DAmageNet is employed as the adversarial examples. The DAmageNet contains 96020 adversarial examples with 1000 classes, which attacks the ImageNet to create generalized adversarial examples by zero-query adversarial attack. These adversarial examples are with an average difference of 4.2 gray values per pixel by comparing with the corresponding clean example. For rigor, we keep all the images as they are without scaling them, though they look different ratios. However, facing with these adversarial examples, most neural network trained on the ImageNet can be easily attacked. The Top-1 recognition error rate of famous classification network, such as VGG, ResNet, Inception, Xception, DenseNet, and etc., can reach more than 90%. It means that these classification network can't resist the adversarial example attack.

Moreover, except for the adversarial examples from DAmageNet, we introduce a popular adversarial attack method, PGD, to generate 50000 adversarial examples based on ImageNet validation set for conducting ablation experiments with the DAmageNet dataset.

4.2 Experimental Setup

In this paper, one of the most widely used classification model ResNet50 is adopted to realize these sophisticated designed experiments. The detailed structure of ResNet50 is shown in Table 1. It is mainly composed of four residual blocks, which deepens the network, and meanwhile introduces the shortcut connection for avoiding the phenomenon of gradient disappearance.

Furthermore, both the DAmageNet and PGD dataset for adversarial examples, and ImageNet validation dataset for clean example are employed to evaluate the designed experiments. ImageNet validation set contains 50000 clean images, and DAmageNet includes 50000 adversarial examples corresponding to clean images, while the PGD also includes another 50000 adversarial examples corresponding to the same clean images. When implementing the experiments,

TABLE 1
The Detailed Structure of the ResNet50

Block	Output Size	Number	Layer Size
Conv1	112×112	1	7×7
Max Pool	56×56	1	3×3
Residual Block1	56×56	3	1×1
			3×3
			1×1
Residual Block2	28×28	4	1×1
			3×3
			1×1
Residual Block3	14×14	6	1×1
			3×3
			1×1
Residual Block4	7×7	3	1×1
			3×3
			1×1
Average Pool	1×1	1	7×7
Fully Connection	1000	1	1×1

the resolution of the input image is 224×224 . All experiments run on Nvidia GTX 2080Ti.

In Section 4.6, the ResNet50 pretrained on ImageNet is implemented as the initial model of retraining. The batch size is set to 32. The Adam optimizer is adopted to optimize the loss function. The initial value of the learning rate is 0.001. The exponential decay rate of the first-order moment estimation is 0.9, and the second-order moment estimation is 0.999.

In Section 4.7, the ResNet50 is adopted as the binary classification network which is trained from scratch. All the weight parameters of the network are randomly initialized, and the batch size is set to 32. The Adam optimizer is adopted to optimize the loss function. The initial value of the learning rate is 0.001. The exponential decay rate of the first-order moment estimation is 0.9, and the second-order moment estimation is 0.999.

4.3 CAM Analysis

In order to evaluate the difference learned by the classification network between the clean example and adversarial example, the CAM is adopted to visually present the difference. The CAM can obtain the attention map by projecting back the weights of the output layer in the classification network onto the convolutional feature maps. Therefore, when adopting the classification network to classify the image, the CAM holds the great potential to visually show the interested regions of the input image where the learning network focuses on. In this experiment, before adopting the CAM to analyze the difference between the clean and adversarial example, Fig. 2 first presents the histogram of the clean examples and corresponding adversarial images. It can be found the distribution of pixel value is with a tiny difference between clean and adversarial example. And this tiny difference can be regarded as the reason for classification network to make an error prediction.

Furthermore, the CAM is adopted to visually demonstrate the attention maps both for the clean example and

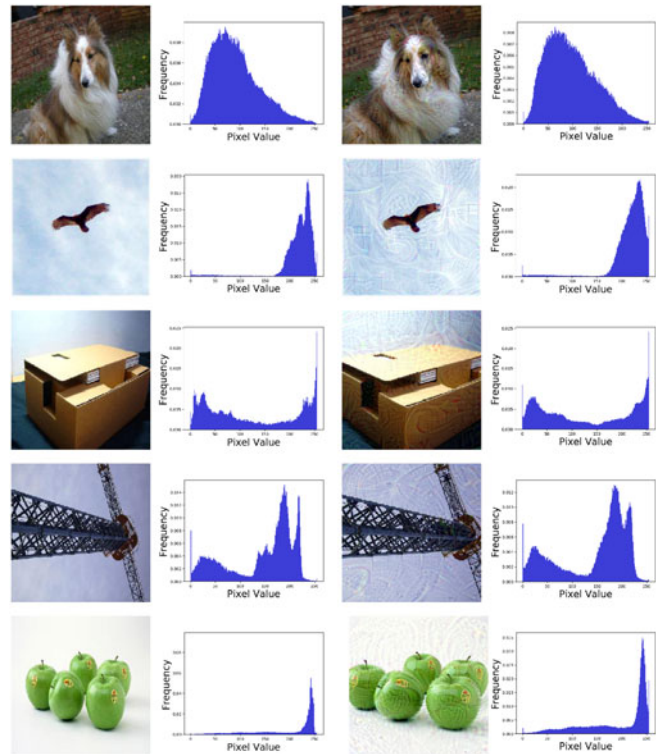


Fig. 2. Histogram Analysis. The first column is the clean images from ImageNet validation set. The second column is the histograms of images in the first column. The third column is the adversarial images corresponding to the first column from DAmageNet. The fourth column is the histograms of images in the third column.

adversarial example, to explicitly interpret why the classification network can be attacked by the adversarial example. Fig. 3 presents the experimental results and the red color represents the interested area focused by the network to support the decision-making process. It can be obviously found that in most situation, the attention map of the adversarial example is overlaid with attentions in different ways by comparing with the attention map of the clean example. To be more specific, because of the perturbations in the adversarial example, the classification network is misled with a wrong direction and focuses on the different regions of the adversarial example so as to make an error classification for the interested object. It also interprets the reason why the classification network can be attacked with the adversarial example. Moreover, this experiment is mainly used to analyze the interpretation of adversarial example attack in a visual way and to locate the interested region focused by the classification network.

More important, in order to further prove the finding of “the adversarial example obtains a different interested region from the classification network”, another experiment is designed and implemented. This experiment is originated from one phenomenon: “because the deep learning network is very sensitive to the change of input image, if some pixels in the interested region of input image is changed, the classification accuracy is disturbed with these changes.” By borrowing the idea from this, we can change some pixels in the interested region of clean example and the same position in the corresponding adversarial example. After processing these remodified images with the classification network, the

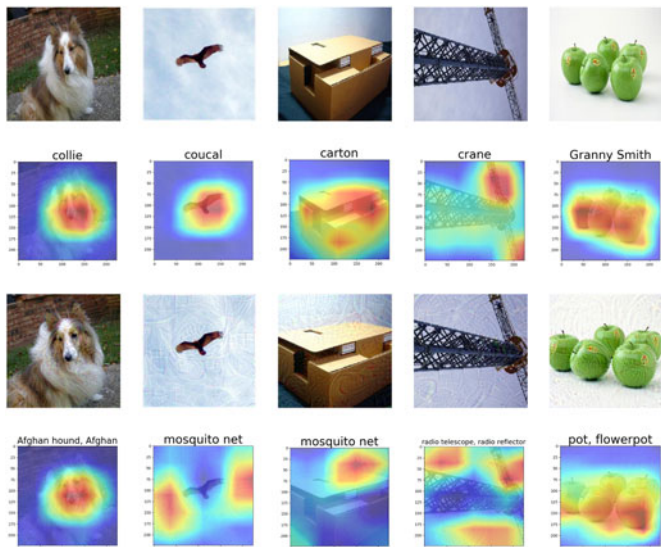


Fig. 3. CAM Analysis. The first row is the clean images from ImageNet validation set. The second row is the attention maps of images in the first row which highlighted by CAM. The third row is the adversarial images corresponding to the first row from DAMageNet. The fourth row is the attention maps of images in the third row which highlighted by CAM.

classification accuracy of the clean example should be disturbed while the accuracy of the adversarial example should almost declines slightly. The reason behind this is that only the change in the interested region can generate a great influence for the final classification accuracy. However, the changing area don't belong to the interested region of the adversarial example focused by the classification network, where the classification network adopts different interested region as the basis to classify the image. Therefore, if the experimental result can follow the assumption, it also can prove that the classification network focuses on different interested region when classifying the adversarial example and this is also the why the adversarial attack can be successfully implemented.

The wrong regions of interest in the image results due to misclassification is not a new finding. Li et al. [47], for example, noted that from the saliency maps for predicted labels, the surrounding background of the target object played an important role in misclassifying the image. Dong et al. [48] also used attention maps of trained models and defense models to indicate the discriminative regions by adopting CAM. In addition, Xiao et al. [49] proposed a new type of perturbation based on spatial transformation, and then utilized CAM to locate the discriminative regions implicitly identified by the DNN. However, these approaches merely adopted CAM to visual the saliency maps between the clean example and adversarial example to identify the differences in the region of interest. In addition to using CAM to locate the different regions of interest (similar to the existing approaches in the literature, such as those of [47], [48], [49], we also utilize attention maps visualized by CAM to conduct the following experiment.

In this experiment, based on the attention map presented by CAM, we randomly select 100 pixels from the red area in the clean example, and 100 pixels with the same position in the corresponding adversarial example, which are from the DAMageNet and PGD dataset. Then, the value of these

pixels is changed to 0, 127, 200 or 255, respectively. After changing the value of pixels in the interested region, the remodified clean and adversarial examples are input into the classification network to evaluate the classification accuracy. Tables 2 and 3 respectively present the classification accuracy (confidence value) before and after changing the pixel value on the area of interested both on the clean and adversarial examples. Eight categories are used to evaluate this experiment on DAMageNet, PGD and the corresponding clean images. Moreover, the bottom row in the Table presents the statistical results of the classification accuracy changes for different pixel values, which are calculated from the average differences between the confidence values of label-unchanged categories before and after the change, and the larger average difference represents a greater change. It can be found that after changing the pixel value on the regions of interest, the classification accuracy of clean examples becomes more unstable compared to the accuracy on the adversarial examples. Especially on the PGD dataset, the difference is even more pronounced. It can be said that the classification accuracy of most clean examples has a larger jitter while the accuracy on adversarial examples are more stable. The reason behind this is that the area of interest with the changed pixel is mainly focused by the classification network when classifying the clean example, while it is not the attention map to classify the adversarial example. This experimental result proves the earlier discussed assumption and it strengthens the findings from the CAM analysis.

Furthermore, there is an interesting finding in this experiment. In Table 2, it can be noticed that the adversarial example of granny smith obtains a great change after changing the pixel to 0, where the predicted label changes from pot to tennis ball. This bad case maybe caused by the great change on the input image, which the pixel value of interested region is changed to 0. While the classification network is just very sensitive to this change and makes totally wrong prediction on this specific case.

4.4 Feature Maps Analysis

In order to investigate the difference learned by the classification model when classifying the clean example and adversarial example respectively, this experiment is first designed to figure out the difference for each feature map in the learning network, where the feature map represents the neuron unit to make a respond to the classification task. It also means that we want to find out whether the classification model can obviously make a different respond when handling the adversarial example by comparing with the clean example. To be more specific, the clean example and its corresponding adversarial example are first input into the pre-trained ResNet50 to realize the classification process, respectively. After processing with the classification network, feature maps of the clean and adversarial example, which represent what have been learned by the network, can be obtained from each network layer. Then, each pair of feature maps for the clean and adversarial example respectively are compared with each other by calculating the pixel value of feature maps. This pixel values of the clean and adversarial example are further computed to obtain the difference value on the pair of feature maps with the same

TABLE 2
The Top-1 Labels and Their Confidence Values Before and After Changing the Pixels to 0, 127, 200 and 255 of Referred Regions on Clean Examples and Corresponding Adversarial Examples From DAmageNet Respectively

Category	Pixel Value	Change	Clean Example		Adversarial Example	
			Label	Confidence	Label	Confidence
collie	0	before	collie	95.10	afghan hound	90.42
		after	collie	81.73	afghan hound	90.34
	127	before	collie	95.10	afghan hound	90.42
		after	collie	90.13	afghan hound	90.08
	200	before	collie	95.10	afghan hound	90.42
		after	collie	84.09	afghan hound	91.53
	255	before	collie	95.10	afghan hound	90.42
		after	collie	85.40	afghan hound	91.00
kite	0	before	kite	47.62	mosquito net	50.38
		after	kite	62.45	mosquito net	33.92
	127	before	kite	47.62	mosquito net	50.38
		after	kite	67.70	mosquito net	58.22
	200	before	kite	47.62	mosquito net	50.38
		after	kite	54.68	mosquito net	50.76
	255	before	kite	47.62	mosquito net	50.38
		after	kite	53.85	mosquito net	56.64
carton	0	before	carton	78.84	mosquito net	77.28
		after	carton	83.60	mosquito net	80.41
	127	before	carton	78.84	mosquito net	77.28
		after	carton	80.53	mosquito net	98.08
	200	before	carton	78.84	mosquito net	77.28
		after	carton	83.48	mosquito net	77.37
	255	before	carton	78.84	mosquito net	77.28
		after	carton	86.80	mosquito net	82.69
crane	0	before	crane	99.12	radio telescope	84.53
		after	crane	99.17	radio telescope	85.98
	127	before	crane	99.12	radio telescope	84.53
		after	crane	99.08	radio telescope	84.48
	200	before	crane	99.12	radio telescope	84.53
		after	crane	98.05	radio telescope	84.04
	255	before	crane	99.12	radio telescope	84.53
		after	crane	99.07	radio telescope	84.55
granny smith	0	before	granny smith	61.83	pot	20.44
		after	granny smith	75.31	tennis ball	17.13
	127	before	granny smith	61.83	pot	20.44
		after	granny smith	79.76	pot	12.36
	200	before	granny smith	61.83	pot	20.44
		after	granny smith	69.61	pot	20.57
	255	before	granny smith	61.83	pot	20.44
		after	granny smith	69.19	pot	20.35
stingray	0	before	stingray	78.38	loggerhead	89.17
		after	stingray	70.19	loggerhead	88.73
	127	before	stingray	78.38	loggerhead	89.17
		after	stingray	71.67	loggerhead	86.92
	200	before	stingray	78.38	loggerhead	89.17
		after	stingray	69.29	loggerhead	84.23
	255	before	stingray	78.38	loggerhead	89.17
		after	stingray	74.21	loggerhead	84.16
cock	0	before	cock	80.22	jackfruit	99.86
		after	cock	61.01	jackfruit	99.10
	127	before	cock	80.22	jackfruit	99.86
		after	cock	66.50	jackfruit	95.37
	200	before	cock	80.22	jackfruit	99.86
		after	cock	49.36	jackfruit	91.77
	255	before	cock	80.22	jackfruit	99.86
		after	cock	49.65	jackfruit	90.71
jay	0	before	jay	92.53	bulbul	98.12
		after	jay	74.22	bulbul	91.81
	127	before	jay	92.53	bulbul	98.12
		after	jay	74.99	bulbul	85.58
	200	before	jay	92.53	bulbul	98.12
		after	jay	88.65	bulbul	97.99
	255	before	jay	92.53	bulbul	98.12
		after	jay	90.98	bulbul	98.10
change statistics	0			-3.245		-2.781
	127			-0.410		+0.111
	200			-4.553		-1.492
	255			-3.061		-0.250

TABLE 3
The Top-1 Labels and Their Confidence Values Before and After Changing the Pixels to 0, 127, 200 and 255 of Referred Regions on Clean Examples and Corresponding Adversarial Examples From PGD Dataset Respectively

Category	Pixel Value	Change	Clean Example		Adversarial Example	
			Label	Confidence	Label	Confidence
lorikeet	0	before	lorikeet	99.51	stinkhorn	55.10
		after	lorikeet	90.73	stinkhorn	56.70
	127	before	lorikeet	99.51	stinkhorn	55.10
		after	lorikeet	89.61	stinkhorn	55.92
	200	before	lorikeet	99.51	stinkhorn	55.10
		after	lorikeet	77.80	stinkhorn	50.85
255	before	lorikeet	99.51	stinkhorn	55.10	
	after	lorikeet	76.97	stinkhorn	49.89	
coucal	0	before	coucal	90.94	robin	72.13
		after	coucal	52.17	robin	69.61
	127	before	coucal	90.94	robin	72.13
		after	coucal	65.97	robin	67.35
	200	before	coucal	90.94	robin	72.13
		after	coucal	50.25	robin	69.64
255	before	coucal	90.94	robin	72.13	
	after	coucal	57.69	robin	71.96	
bee eater	0	before	bee eater	98.09	dragonfly	88.98
		after	bee eater	92.44	dragonfly	85.56
	127	before	bee eater	98.09	dragonfly	88.98
		after	bee eater	92.71	dragonfly	84.44
	200	before	bee eater	98.09	dragonfly	88.98
		after	bee eater	93.79	dragonfly	88.60
255	before	bee eater	98.09	dragonfly	88.98	
	after	bee eater	93.51	dragonfly	89.22	
hornbill	0	before	hornbill	85.19	bulbul	60.23
		after	hornbill	80.00	bulbul	55.12
	127	before	hornbill	85.19	bulbul	60.23
		after	hornbill	72.96	bulbul	53.53
	200	before	hornbill	85.19	bulbul	60.23
		after	hornbill	71.21	bulbul	53.82
255	before	hornbill	85.19	bulbul	60.23	
	after	hornbill	79.52	bulbul	60.33	
hummingbird	0	before	hummingbird	99.77	howlermonkey	23.30
		after	hummingbird	65.65	howlermonkey	14.27
	127	before	hummingbird	99.77	howlermonkey	23.30
		after	hummingbird	67.71	howlermonkey	18.75
	200	before	hummingbird	99.77	howlermonkey	23.30
		after	hummingbird	75.60	howlermonkey	16.47
255	before	hummingbird	99.77	howlermonkey	23.30	
	after	hummingbird	72.30	howlermonkey	22.71	
jacamar	0	before	jacamar	94.90	jay	10.29
		after	jacamar	53.90	jay	13.65
	127	before	jacamar	94.90	jay	10.29
		after	jacamar	70.25	jay	10.23
	200	before	jacamar	94.90	jay	10.29
		after	jacamar	80.09	jay	13.41
255	before	jacamar	94.90	jay	10.29	
	after	jacamar	89.37	jay	11.73	
toucan	0	before	toucan	89.64	hornbill	99.96
		after	toucan	79.30	hornbill	90.11
	127	before	toucan	89.64	hornbill	99.96
		after	toucan	80.45	hornbill	91.30
	200	before	toucan	89.64	hornbill	99.96
		after	toucan	80.83	hornbill	93.77
255	before	toucan	89.64	hornbill	99.96	
	after	toucan	78.89	hornbill	92.37	
drake	0	before	drake	99.92	bullfrog	13.99
		after	drake	66.63	bullfrog	14.63
	127	before	drake	99.92	bullfrog	13.99
		after	drake	99.01	bullfrog	10.95
	200	before	drake	99.92	bullfrog	13.99
		after	drake	91.25	bullfrog	11.90
255	before	drake	99.92	bullfrog	13.99	
	after	drake	90.61	bullfrog	10.54	
change statistics	0		-22.13		-3.041	
	127		-14.91		-3.938	
	200		-17.14		-3.190	
	255		-14.88		-1.903	

TABLE 4
Ten Most Different Feature Maps Obtained From DAmageNet

Feature Map	Differences
pool_1_pad_35	1148.25
pool_1_pad_23	1147.14
pool_1_pad_54	1141.76
pool_1_pad_29	1138.84
pool_1_pad_26	1127.24
activation_1_35	1108.30
pool_1_pad_3	1108.21
activation_1_23	1107.19
activation_1_37	1104.14
activation_1_54	1101.80

TABLE 6
Ten Most Different Feature Maps Obtained From PGD Dataset

Feature Map	Differences
pool_1_pad_31	1128.23
activation_1_54	1060.35
pool_1_pad_1	1056.91
activation_1_1	1016.91
pool_1_pad_41	1011.25
pool_1_pad_56	955.07
activation_1_56	955.07
pool_1_pad_8	952.31
pool_1_pad_6	935.65
pool_1_pad_48	927.72

TABLE 5
Ten Least Different Feature Maps Obtained From DAmageNet

Feature Map	Differences
activation_43_1098	0.0883
activation_42_226	0.0883
activation_42_144	0.0883
activation_41_451	0.0883
activation_35_52	0.0883
activation_35_244	0.0883
activation_35_183	0.0883
activation_35_161	0.0883
activation_32_94	0.0883
activation_32_219	0.0883

TABLE 7
Ten Least Different Feature Maps Obtained From PGD Dataset

Feature Map	Differences
activation_42_168	0.0883
activation_41_507	0.0883
activation_41_27	0.0883
activation_41_247	0.0883
activation_41_196	0.0883
activation_41_165	0.0883
activation_33_154	0.0883
activation_32_79	0.0883
activation_32_177	0.0883
activation_1_23	0.0883

spatial using the metric of RMSE (Root Mean Squard Error) as to discover why the adversarial example can mislead the classification model to make an error prediction. The calculation formula of RMSE is list as follows:

$$RMSE = \sqrt{\frac{1}{H \times W \times C} \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^C (a(i, j, k) - b(i, j, k))^2}, \quad (4)$$

where $H \times W \times C$ is the size of the feature map, a is the feature map of clean example and b is the feature map of adversarial example. The larger value of the RMSE represents the low similarity between two feature maps.

After calculating the difference value of each pair of feature maps obtained from the classification network on DAmageNet and PGD dataset, respectively. The result can be found in the Tables 4, 5, 6 and 7. Tables 4 and 6 list ten difference value with the biggest different; Tables 5 and 7 show the result with the smallest difference value. The “pool_1_pad_35” represents the 35th feature map produced by the first pooling layer with the patch padding operation. While the “activation_1_35” indicates the 35th feature map produced by the first activation layer. According to the experimental result both on the DAmageNet and PGD dataset, it can be found that classification model usually makes an error prediction (with large difference value) at the first pooling layer and the first activation layer when facing with the adversarial example. And the smallest difference value between clean and adversarial example usually can be obtained from the higher-level activation layers. It means the lower-level layer in the classification network is more sensitive to the adversarial example attack, which has a less

influence on the high-level layer in the network. This finding also follows the principle of generating the adversarial example, where the generating process usually adds some perturbations on the clean image to produce the adversarial example. Therefore, the low-level layer in the network, which is mainly used to handle the low-level features, is more vulnerable to adversarial example attack. This experiment also can explain the first question proposed in this paper: “What’s the difference between the adversarial example and clean image learned by the classification model implemented on the ImageNet?”

In Fig. 4, we visually show the four feature map pairs with the largest differences between the real and adversarial example referring Table 4. It can be seen that even if the feature map pairs are with the largest difference calculated by quantitative analysis, humans cannot visually distinguish the clean and adversarial example based on their feature maps.

Based on the experimental results, it can be found that the feature maps between the clean example and the adversarial example in the classification model are quite different. Next, we explore the changes in the prediction results of the classification model when masking the feature maps with large differences.

4.5 Changing Feature Maps / Filters to Resist Adversarial Example Attack

In Section 4.4, we discuss the difference of the feature map when the classification network processes the clean example and adversarial example. Based on the findings in last experiment, some sophisticated designed experiments are implemented to verify whether these differences can then be

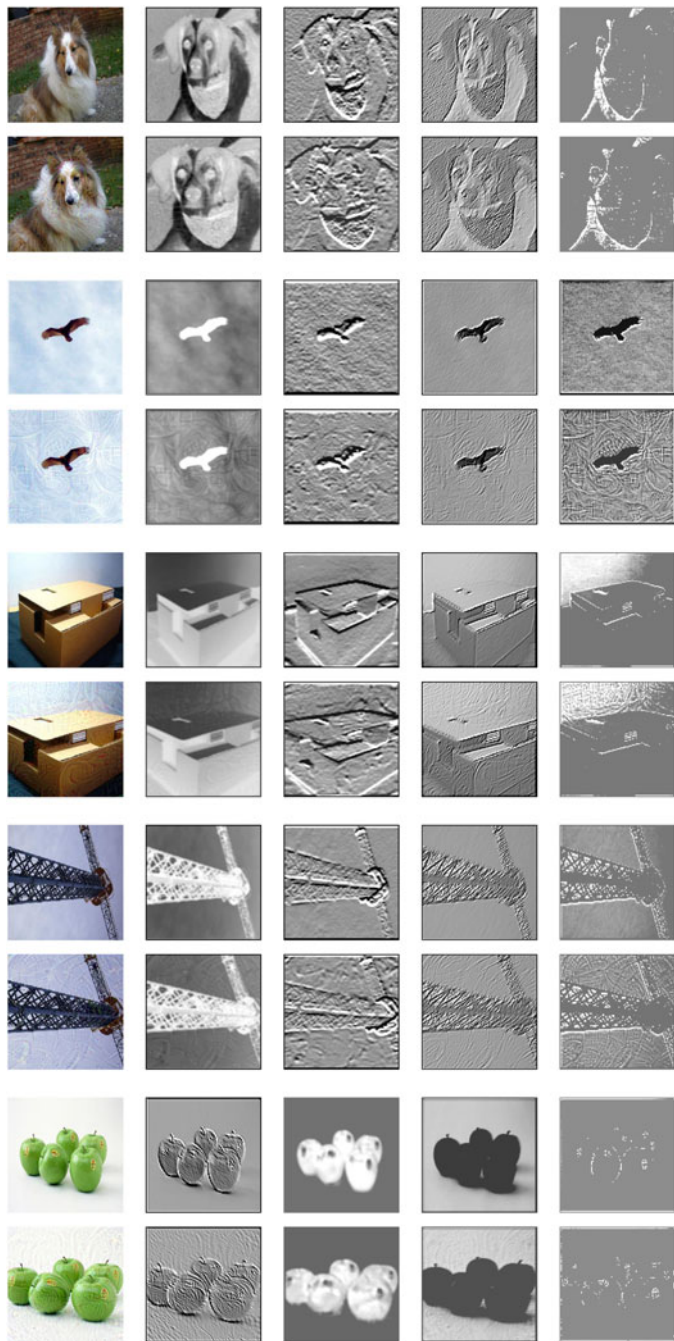


Fig. 4. Visualize the feature maps from clean (Top of each pair) and adversarial (Bottom of each pair) example.

used to resist the adversarial example attack. First, as shown in Tables 4 and 6, it can be observed that when the adversarial example is processed with the classification network, the network produces some totally different feature maps by comparing with the clean example. These feature maps can be regarded as the “sensitive” feature maps for the adversarial examples. It naturally raises a hypothesis, if these feature maps are masked during the network processing, the classification network may not to be misled by the adversarial example attack. In this experiment, total eight categories of images and corresponding adversarial examples are employed to evaluate the propose and sensitive feature maps are masked when processing these images.

As shown in Tables 8 and 9, based on the value of change statics, the confidence values of clean and adversarial examples become better in most categories after masking the sensitive feature maps. The reason behind this maybe that these sensitive feature maps are in the low-level and the low-level information usually include too much useless information. Masking these feature maps just reduces the interference of information and making the classification network to make a more accurate decision. Moreover, it can be proven that masking the sensitive feature map is no useful for defending the adversarial example attack. Furthermore, these sensitive feature maps can be used to interpret the adversarial example attack in a way but they are not the only reason for interpretable adversarial learning. Finally, this experiment also overturns the hypothesis of defending the adversarial example attack.

In this section, another hypothesis is that “if we can increase the influence of these sensitive feature maps (to further increase the difference of pair images), then the adversarial example can be detected.” In this experiment, the weights of filters in the first network layer, which are relative to sensitive feature maps, are enlarged by 1.8 times. Actually, it is implemented as an attention mechanism to increase the effects of filters so as to produce more influences when classifying the images. The experimental results are shown in Table 10 for the DAMageNet and Table 12 for the PGD dataset. Seen from the value of change statics, it can be found the confidence values of the most adversarial examples are decreased after enlarging the weights of filters. It can be said that enlarging the influence of sensitive filters can be an effective way to defend the adversarial example attack on one side. However, the confidence values of clean examples are also decreased. It means that enlarging the weights of sensitive filters actually changed the effectiveness of the classification network. Therefore, it can’t be a choice to defend the adversarial example attack.

Besides, there is an interesting finding that the predicted labels of adversarial examples in the category of kite, carton, crane and granny smith are changed after enlarging the weight of filters in the first layer in Table 10. The reason behind this is that adversarial examples are the clean image added with perturbations, and now enlarging the weight of filters in the first layer and the inherent perturbations make the prediction results of the model unpredictable.

Moreover, except for enlarging the weights of sensitive filters in the classification network, another direction is to enlarge the weights of filters, which are relative to the feature maps with smallest difference value. These filters can be regarded as the insensitive filters in the classification network for adversarial example attack. As shown in Tables 5 and 7, these insensitive filters are in the fifth residual block of classification network. In the experiment, the weights of insensitive filters are also enlarged by 1.8 times. The experimental results are shown in Tables 11 and 13, and they present the same trend shown in Tables 10 and 12. After enlarging the weights of insensitive filters, the confidence values of the most clean and adversarial examples are both decreased, even in a more largely way. In detailed, the value of change statics on the clean image and adversarial examples in DAMageNet is 25.79 and 4.656 for enlarging the weights of insensitive filters while this value is just 0.601

TABLE 8

The Top-1 Labels and Their Confidence Values Before and After Masking the Feature Maps With Large Differences on Clean Examples and Corresponding Adversarial Examples From DAmageNet Respectively

Category	Mask	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
collie	before	collie	95.10	afghan hound	90.42
	after	collie	98.37	afghan hound	95.23
kite	before	kite	47.62	mosquito net	50.38
	after	kite	62.43	mosquito net	61.63
carton	before	carton	78.84	mosquito net	77.28
	after	carton	88.14	mosquito net	83.62
crane	before	crane	99.12	radio telescope	84.53
	after	crane	99.10	radio telescope	82.68
granny smith	before	granny smith	61.83	pot	20.44
	after	granny smith	97.89	tennis ball	18.06
stingray	before	stingray	78.38	loggerhead	89.17
	after	stingray	79.32	loggerhead	90.63
cock	before	cock	80.22	jackfruit	99.86
	after	cock	77.62	jackfruit	99.93
jay	before	jay	92.53	bulbul	98.12
	after	jay	92.45	bulbul	96.83
change statistics		+7.710		+2.970	

TABLE 9

The Top-1 Labels and Their Confidence Values Before and After Masking the Feature Maps With Large Differences on Clean Examples and Corresponding Adversarial Examples From PGD Dataset Respectively

Category	Mask	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
lorikeet	before	lorikeet	99.51	stinkhorn	55.10
	after	lorikeet	99.58	stinkhorn	56.66
coucal	before	coucal	90.94	robin	72.13
	after	coucal	92.72	robin	69.21
bee eater	before	bee eater	98.09	dragonfly	88.98
	after	bee eater	98.89	dragonfly	86.59
hornbill	before	hornbill	85.19	bulbul	60.23
	after	hornbill	85.93	bulbul	61.10
hummingbird	before	hummingbird	99.77	howlermonkey	23.30
	after	hummingbird	99.84	howlermonkey	27.49
jacamar	before	jacamar	94.90	jay	10.29
	after	jacamar	95.62	jay	09.21
toucan	before	toucan	89.64	hornbill	99.96
	after	toucan	94.47	hornbill	99.96
drake	before	drake	99.92	bullfrog	13.99
	after	drake	99.93	bullfrog	24.57
change statistics		+1.127		+1.351	

TABLE 10

The Top-1 Labels and Their Confidence Values Before and After Enlarged the Weight of Filters in the First Layer on Clean Examples and Corresponding Adversarial Examples From DAmageNet Respectively

Category	Enlarge	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
collie	before	collie	95.10	afghan hound	90.42
	after	collie	97.59	afghan hound	77.46
kite	before	kite	47.62	mosquito net	50.38
	after	kite	73.50	red wine	16.70
carton	before	carton	78.84	mosquito net	77.28
	after	carton	74.58	four poster	47.02
crane	before	crane	99.12	radio telescope	84.53
	after	crane	99.85	dragonfly	44.73
granny smith	before	granny smith	61.83	pot	20.44
	after	granny smith	43.71	tennis ball	29.18
stingray	before	stingray	78.38	loggerhead	89.17
	after	stingray	77.49	loggerhead	90.00
cock	before	cock	80.22	jackfruit	99.86
	after	cock	79.18	jackfruit	95.81
jay	before	jay	92.53	bulbul	98.12
	after	jay	92.55	bulbul	95.73
change statistics		-0.601		-4.650	

TABLE 11

The Top-1 Labels and Their Confidence Values Before and After Enlarged the Weight of Filters at the Fifth Residual Block on Clean Examples and Corresponding Adversarial Examples From DAmageNet Respectively

Category	Enlarge	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
collie	before	collie	95.10	afghan hound	90.42
	after	collie	70.59	afghan hound	41.27
kite	before	kite	47.62	mosquito net	50.38
	after	kite	43.66	mosquito net	43.85
carton	before	carton	78.84	mosquito net	77.28
	after	carton	43.67	four poster	48.35
crane	before	crane	99.12	radio telescope	84.53
	after	crane	75.14	radio telescope	52.46
granny smith	before	granny smith	61.83	pot	20.44
	after	granny smith	43.87	pot	6.17
stingray	before	stingray	78.38	loggerhead	89.17
	after	stingray	22.40	loggerhead	90.63
cock	before	cock	80.22	jackfruit	99.86
	after	cock	60.92	hen	55.06
jay	before	jay	92.53	bulbul	98.12
	after	gardenspider	86.94	blackswan	73.76
change statistics		-25.79		-4.656	

TABLE 12

The Top-1 Labels and Their Confidence Values Before and After Enlarged the Weight of Filters in the First Layer on Clean Examples and Corresponding Adversarial Examples From PGD Dataset Respectively

Category	Mask	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
lorikeet	before	lorikeet	99.51	stinkhorn	55.10
	after	lorikeet	98.81	stinkhorn	49.49
coucal	before	coucal	90.94	robin	72.13
	after	coucal	66.17	robin	56.50
bee eater	before	bee eater	98.09	dragonfly	88.98
	after	bee eater	95.60	dragonfly	89.90
hornbill	before	hornbill	85.19	bulbul	60.23
	after	hornbill	80.12	bulbul	58.00
hummingbird	before	hummingbird	99.77	howlermonkey	23.30
	after	hummingbird	99.24	howlermonkey	10.15
jacamar	before	jacamar	94.90	jay	10.29
	after	jacamar	94.86	jay	02.05
toucan	before	toucan	89.64	hornbill	99.96
	after	toucan	88.67	hornbill	79.01
drake	before	drake	99.92	bullfrog	13.99
	after	drake	90.20	bullfrog	13.17
change statistics		-5.523		-8.221	

TABLE 13

The Top-1 Labels and Their Confidence Values Before and After Enlarged the Weight of Filters at the Fifth Residual Block on Clean Examples and Corresponding Adversarial Examples From PGD Dataset Respectively

Category	Mask	Clean Example		Adversarial Example	
		Label	Confidence	Label	Confidence
lorikeet	before	lorikeet	99.51	stinkhorn	55.10
	after	lorikeet	95.27	stinkhorn	42.49
coucal	before	coucal	90.94	robin	72.13
	after	coucal	56.51	robin	52.34
bee eater	before	bee eater	98.09	dragonfly	88.98
	after	bee eater	67.01	dragonfly	57.02
hornbill	before	hornbill	85.19	bulbul	60.23
	after	hornbill	61.03	bulbul	34.63
hummingbird	before	hummingbird	99.77	howlermonkey	23.30
	after	hummingbird	96.90	howlermonkey	13.65
jacamar	before	jacamar	94.90	jay	10.29
	after	jacamar	86.82	jay	04.14
toucan	before	toucan	89.64	hornbill	99.96
	after	toucan	62.50	hornbill	67.99
drake	before	drake	99.92	bullfrog	13.99
	after	drake	83.22	bullfrog	05.52
change statistics		-18.58		-18.27	

TABLE 14

The Classification Accuracy of the Retrained ResNet50 and the Pretrained ResNet50 on ImageNet Validation Set and DAmageNet Validation Set Respectively Under Different Training Set Sizes

Training Set Sizes	Acc.ImageNet	Acc.DAmageNet
10000	15.07 (41.77)	3.891 (4.430)
20000	19.21 (41.77)	5.684 (4.430)
30000	29.98 (41.77)	6.987 (4.430)
40000	54.60 (41.77)	19.739 (4.430)

and 4.650 for enlarging the weights of sensitive filters. For the clean image and adversarial examples in PGD dataset, the value of change statics is 18.58 and 18.27 for enlarging the weights of insensitive filters while the corresponding value is just 15.55 and 8.221 for enlarging the weights of sensitive filters.

Overall, for the first question proposed at the beginning of this paper, the difference discovered when adopting the classification network to classify the clean example and adversarial example respectively, can't be further used to resist the adversarial example attack.

4.6 Adversarial Training to Resist the Adversarial Example Attack

For defending the adversarial example attack, one of the most common used method is the adversarial training. It means to retrain the classification network with the clean and adversarial example together to make the network more robust. Therefore, in this paper, the adversarial training is also adopted to verify whether the retrained classification network can defend the adversarial example attack. In this experiment, because the ResNet50 adopted in this paper is a pretrained classification network trained on the ImageNet dataset, we only need retrain the ResNet50 with the additional adversarial examples from DAmageNet to realize the adversarial training. To be more specific, in order to evaluate the performance of retrained ResNet50, we select 10000, 20000, 30000 and 40000 adversarial examples from the DAmageNet as the additional training dataset, respectively. And another 10000 adversarial examples and 10000 clean examples from ImageNet validation set are remained as the two validation dataset. It can be noted here that the label of adversarial examples in training dataset is set as the real one instead of the fake one (e.g., the adversarial example which looks like a cat is set with a real "cat" label instead of the "dog" label obtained from the classification network with the wrong prediction) so as to improve the robustness of the classification network to make a correct prediction, which is easily generalized in practical application. Moreover, the pretrained ResNet50 can achieve the 41.77% classification accuracy on the ImageNet validation set and 4.430% accuracy on the adversarial example from DAmageNet validation set, which is used to measure how many clean and adversarial examples can be correctly classified after the adversarial retraining, showed in parentheses in the Table 14.

After implementing the adversarial training with the 10000, 20000, 3000 and 40000 adversarial examples respectively, the experimental results can be found in Table 14. It

TABLE 15

The Classification Accuracy of the Retrained ResNet50 and the Pretrained ResNet50 on ImageNet Validation Set and PGD Validation Set Respectively Under Different Training Set Sizes

Training Set Sizes	Acc.ImageNet	Acc.PGD set
10000	25.71 (41.77)	11.77 (0.130)
20000	31.53 (41.77)	12.21 (0.130)
30000	44.98 (41.77)	15.68 (0.130)
40000	67.42 (41.77)	24.08 (0.130)

can be found that when the size of additional training set is less than 40000 adversarial examples, the classification accuracy on the ImageNet validation set achieves a worse performance by comparing with the pretrained ResNet50. It can be said that the classification network is messed up with the adversarial training datasets of these sizes on classifying clean examples from ImageNet validation set after the process of retraining. However, with the size of training dataset increase, the performance both on the ImageNet validation set and DAmageNet validation set becomes better and better. And when the training dataset is increased to 40000, the classification accuracy of retrained ResNet50 achieve the best one 54.60% on ImageNet validation set, even better than the original accuracy. Meanwhile, when the size of additional training set is larger than 10000 adversarial examples, the classification accuracy on the DAmageNet validation set are beyond 4.430% which obtained on the original pretrained ResNet50. Moreover, the classification accuracy on adversarial examples is up to 19.73%, with almost five times increase by comparing with the original one.

Furthermore, the adversarial training on the another attacking method PGD is also evaluated to resist the adversarial example attack. Specifically, for the adversarial examples generated by the PGD, 10000, 20000, 30000, and 40000 adversarial examples are adopted as the additional training dataset, respectively. And the left 10000 adversarial examples are remained as the validation dataset, which also includes another 10000 clean images selected from the ImageNet validation set. The experimental results can be found in Table 15. Before implementing the adversarial training, the pretrained ResNet50 can achieve the 41.77% classification accuracy on the ImageNet validation set and only 0.13% on the PGD validation set. As shown in Table 15, comparing with Table 14, it can be found that the classification accuracy after retraining on the PGD validation set is higher than the adversarial training on the DAmageNet under the same training size when evaluated both on the ImageNet validation set and PGD validation set. To be more specific, when the training dataset is increased to 30000, the classification accuracy of retrained ResNet50 achieves a better performance (44.98%) compared to the original one (41.77%), and the best accuracy is the 67.42% when the training dataset is increased to 40000. Moreover, on the PGD validation set, the classification accuracy of retrained ResNet50 also achieves a much better performance compared to the original one, from 11.77% to 24.08%. Furthermore, by comparing the adversarial training on the DAmageNet and PGD dataset, it can be found that the adversarial training achieves a better performance on the PGD dataset to resist the adversarial example attack. In

other words, it is more difficult for the classification network to resist the adversarial example attack when facing with the DAmageNet dataset.

It can be said the adversarial training can be used to improve the robust of the classification network to support the adversarial defense. Moreover, according to this experiment, another useful usage of adversarial training is discovered, which can be used to improve the classification accuracy of the original classification network. It can interpret that the adversarial training increases the diversity of the training dataset in practice so as to improve the precision and robustness of the classification network.

4.7 Training Binary Classification Network to Detect the Adversarial Example

In order to explore how to defend the adversarial example attack in an interpretable way, we design a two-class classifier which adopts the ResNet50 and is trained from scratch to directly classify the clean image and adversarial image and visually presents the attention map. The loss function of the binary classifier is as follows:

$$Loss(y, \hat{y}) = -\frac{1}{N} \sum_{n=1}^N [y_n \cdot \log(\hat{y}_n) + (1 - y_n) \cdot \log(1 - \hat{y}_n)], \quad (5)$$

where N is the number of examples, y is the true distribution, and \hat{y} is the softmax multinomial prediction distribution. Minimizing this softmax loss leads to maximizing the predicted probability of the ground truth class, which would be a good property for model optimization.

All the weight parameters of the network are randomly initialized, and the batch size is set to 32. The Adam optimizer is adopted to optimize the loss function. The initial value of the learning rate is 0.001. The exponential decay rate of the first-order moment estimation is 0.9, and the second-order moment estimation is 0.999. The training epoch is set to 47000 to achieve a better performance. The training dataset is composed of 25000 images from the ImageNet validation set and 25000 images from the DAmageNet. In order to evaluate the performance of this classification network, the validation dataset contains the rest of 25000 adversarial example from the DAmageNet and 25000 clean examples from ImageNet validation set.

To assess the predictive performance of the binary classification model, we calculated its performance in terms of accuracy and area under roc curve (AUC). AUC is used in the classification analysis to evaluate the quality of a classifier. In general, an AUC score of 0.5 means that there is no discrimination, a score between 0.6 and 0.8 is considered acceptable, a score between 0.8 and 0.9 is considered excellent, and more than 0.9 is considered outstanding [51].

After training the binary classification model with the training dataset, the validation dataset is evaluated on this model. It achieves the 98.42% classification accuracy on the clean examples and 99.07% accuracy on the adversarial examples. Besides, the binary classification model is with an AUC of 0.9875. According to the experimental result, it can be said that this binary classification model can be used to detect the adversarial example. Moreover, the Fig. 5 presents the CAM analysis by adopting the binary classification model to classify the clean and adversarial

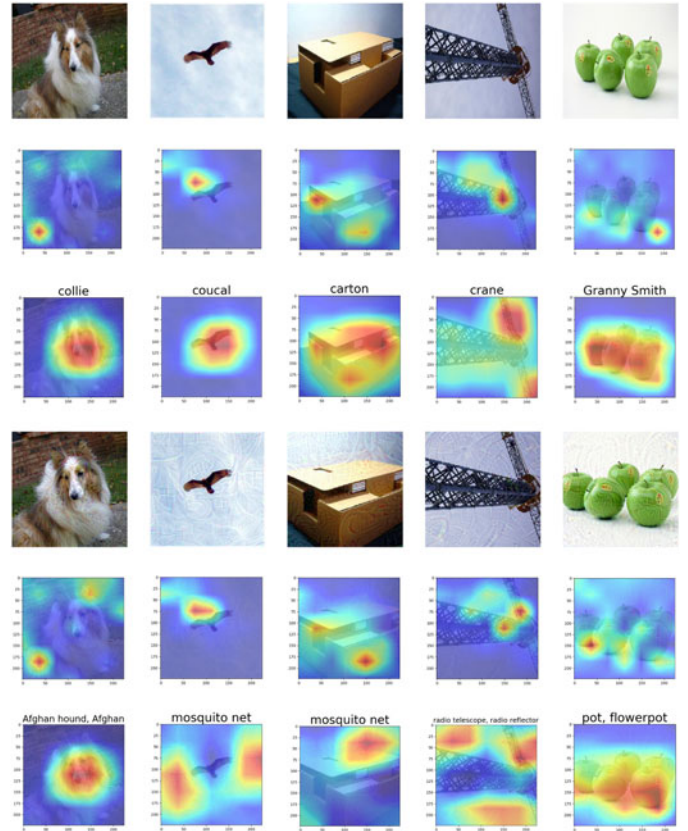


Fig. 5. CAM Analysis. The first row is the clean images from ImageNet validation set. The second and third row are the attention maps of images in the first row which obtained from 1000-class pretrained ResNet and 2-class ResNet by CAM respectively. The fourth row is the adversarial images corresponding to the first row from DAmageNet. The fifth and sixth row are the attention maps of images in the fourth row which obtained from 1000-class pretrained ResNet and 2-class ResNet by CAM respectively.

examples. Moreover, the attention map both on the clean and adversarial example obtained by adopting the pretrained ResNet50 is also visually presented in Fig. 5 as the comparison. It can be found that if we want to accurately identify the difference between the clean image and adversarial image, and further to detect the adversarial example, the classification network should be trained to focus on the different region on the adversarial example. To be more specific, instead of focusing on the object itself by adopting the pretrained ResNet50, the detection network should not place their interest on the more semantic human-perceivable features or attributes and merely focus on some local small region. This findings can provide a new research direction to design more robust classification network to resist the adversarial example attack.

5 CONCLUSION

In this paper, we seek to understand universal adversarial example attacks on image classification models and interpret why the classification model can be successfully attacked by adversarial examples, in order to better design mitigation strategies to resist such adversarial example attacks. To do so, we designed and carried out a number of experiments (e.g., CAM analysis, feature map analysis, feature maps/filters changing, adversarial training, and

binary classification model). The findings from our evaluations showed that the distribution of pixel value has a tiny difference between clean and adversarial examples, but the main difference is the salient region. Moreover, changing the pixel value on the salient region, the confidence values of clean examples have larger jitter. We also noted that the neuron results in a large different response to the adversarial example from the shallow layer. Furthermore, we found that masking the specific feature maps or changing the weights of filters cannot be used to resist adversarial example attacks. Based on our findings, we also conclude that adversarial training can be used to improve the classification accuracy and robustness of the original classification network. While we found training a discriminator to discriminate the adversarial example is potentially useful in resisting adversarial example attack, this is challenging especially dealing with large datasets (e.g., ImageNet, social media or CCTV datasets).

The interpretation in our work is mainly analyzed from the input layer or from a certain layer in the network. In other words, we have yet to consider the combination of different layers from the learning network at the global level. Hence, this is a possible future extension. In addition, we also plan to focus on interpretation of the learning network on different application domains, such as medical image classification, as well as designing more effective adversarial defense methods to resist adversarial example attacks (by understanding why an adversarial example can successfully attack the classification network).

REFERENCES

- [1] F. Wang et al., "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.
- [2] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," *Med. Image Anal.*, vol. 54, pp. 10–19, 2019.
- [3] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "DeepKeyGen: A deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4915–4929, Sep. 2022.
- [4] Y. Ding et al., "MVFusFra: A multi-view dynamic fusion framework for multimodal brain tumor segmentation," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 4, pp. 1570–1581, Apr. 2022.
- [5] Y. Ding et al., "ToStaGAN: An end-to-end two-stage generative adversarial network for brain tumor segmentation," *Neurocomputing*, vol. 462, pp. 141–153, 2021.
- [6] D. Wu, Y. Ding, M. Zhang, Q. Yang, and Z. Qin, "Multi-features refinement and aggregation for medical brain segmentation," *IEEE Access*, vol. 8, pp. 57 483–57 496, 2020.
- [7] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
- [8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*.
- [10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 372–387.
- [11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.
- [12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
- [13] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.
- [14] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 86–94.
- [15] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
- [16] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [17] X. Ma et al., "Characterizing adversarial subspaces using local intrinsic dimensionality," 2018, *arXiv:1801.02613*.
- [18] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*.
- [19] G. S. Dhillon et al., "Stochastic activation pruning for robust adversarial defense," 2018, *arXiv:1803.01442*.
- [20] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," 2017, *arXiv:1711.01991*.
- [21] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "PixelDefend: Leveraging generative models to understand and defend against adversarial examples," 2017, *arXiv:1710.10766*.
- [22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
- [23] S. Liu, X. Wang, M. Liu, and J. Zhu, "Towards better analysis of machine learning models: A visual analytics perspective," *Vis. Inform.*, vol. 1, no. 1, pp. 48–56, 2017.
- [24] L. Schubert, "Tensorflow/lucid: A collection of infrastructure and tools for research in neural network interpretability," *GitHub*. Accessed: Jun. 27, 2021. [Online]. Available: <https://github.com/tensorflow/lucid>
- [25] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.
- [26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
- [27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
- [28] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu, "Interpreting CNN knowledge via an explanatory graph," 2017, *arXiv:1708.01785*.
- [29] M. R. Zafar and N. M. Khan, "DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems," 2019, *arXiv:1906.10263*.
- [30] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting CNNs via decision trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6254–6263.
- [31] X. Zhang, R. Pang, S. Ji, F. Ma, and T. Wang, "I-Algebra: Towards interactive interpretability of deep neural networks," 2021, *arXiv:2101.09301*.
- [32] R. Pang, X. Zhang, S. Ji, X. Luo, and T. Wang, "AdvMind: Inferring adversary intent of black-box attacks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1899–1907.
- [33] S. Chen, X. Huang, Z. He, and C. Sun, "DAMageNet: A universal adversarial dataset," 2019, *arXiv:1912.07160*.
- [34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
- [35] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 135–147.
- [36] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial examples from artifacts," 2017, *arXiv:1703.00410*.
- [37] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution examples and adversarial attacks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7167–7177.
- [38] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*.
- [39] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3449–3457.
- [40] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8827–8836.

- [41] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2131–2145, Sep. 2019.
- [42] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8930–8941.
- [43] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," 2017, *arXiv:1711.09404*.
- [44] G. Tao, S. Ma, Y. Liu, and X. Zhang, "Attacks meet interpretability: Attribute-steered detection of adversarial examples," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7717–7728.
- [45] S. Ma and Y. Liu, "NIC: Detecting adversarial examples with neural network invariant checking," in *Proc. 26th Netw. Distrib. Syst. Secur. Symp.*, 2019.
- [46] A. Shafahi et al., "Adversarial training for free!," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 3358–3369.
- [47] X. Li, J. Li, T. Dai, J. Shi, J. Zhu, and X. Hu, "Rethinking natural adversarial examples for classification models," 2021, *arXiv:2102.11731*.
- [48] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4307–4316.
- [49] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," 2018, *arXiv:1801.02612*.
- [50] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
- [51] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *J. Thoracic Oncol.*, vol. 5, no. 9, pp. 1315–1316, 2010.



Yi Ding (Member, IEEE) received the BS degree in software engineering from the University of Electronic Science and Technology of China (UESTC), in Chengdu, China, in 2008, and the PhD degree in computer science from the Dublin Institute of Technology, Dublin, Ireland, in 2012. He is currently a professor with the Network and Data Security Key Laboratory of Sichuan Province, and with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC). He is also a post-doctoral fellow

with Ningbo WebKing Technology Joint Stock Co., Ltd, China. His research interests include machine learning, deep learning, medical image processing, and computer-aided diagnosis.



Fuyuan Tan received the BE degree in software engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently working toward the MS degree in software engineering from the University of Electronic Science and Technology of China. His current research interests include deep learning, computer vision, digital image security, and Internet of Things.



Ji Geng received the MS degree from Southwest Jiaotong University, Chengdu, China, and the PhD degree in information security from the University of Electronic Science and Technology of China (UESTC) Chengdu, China, in 2019, where he is currently a professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. He has been involved in distributed computing and information security research.



Zhen Qin received the PhD degree in communication and information system from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012. He is currently an professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. He was a visiting scholar with the Department of Electrical Engineering and Computer Science, Northwestern University. His research interests include data fusion analysis, mobile social networks, wireless sensor networks, and image processing.



Mingsheng Cao received the PhD degree in computer science from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019. He is currently a lecturer with the Network and Data Security Key Laboratory of Sichuan Province. His research interests include machine learning, network security, and pervasive computing.



Kim-Kwang Raymond Choo (Senior Member, IEEE) currently holds the Cloud Technology Endowed Professorship with the University of Texas at San Antonio. He is the founding co-editor-in-chief of ACM Distributed Ledger Technologies: Research & Practice, the founding chair of IEEE Technology and Engineering Management Society Technical Committee on Blockchain and Distributed Ledger Technologies, and a Web of Science's Highly Cited Researcher (Computer Science - 2021, Cross-Field - 2020). He is the recipient of the IEEE Systems,

Man, and Cybernetics Technical Committee on Homeland Security Research and Innovation Award, in 2022, the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), and best paper awards from IEEE Systems Journal in 2021, IEEE Computer Society's Bio-Inspired Computing Special Technical Committee Outstanding Paper Award for 2021, IEEE DSC 2021, IEEE Consumer Electronics Magazine for 2020, Journal of Network and Computer Applications for 2020, EURASIP Journal on Wireless Communications and Networking in 2019, IEEE TrustCom 2018, and ESORICS 2015.



Zhiguang Qin (Member, IEEE) is the full professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), where he is also director of the Key Laboratory of New Computer Application Technology and director of UESTC-IBM Technology Center. His research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.