IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 November 2023, accepted 15 November 2023, date of publication 20 November 2023,
date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335094

RESEARCH ARTICLE

Crafting Targeted Universal Adversarial Perturbations: Considering Images as Noise

HUIJIAO WANG, DING CAI, LI WANG, AND ZHUO XIONG
School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Corresponding author: Ding Cai (kwist_jay@outlook.com)

This work was supported in part by the Innovation Project of Guangxi Graduate Education under Grant 2023YCXS061, and in part by the Science and Technology Major Project of Guangxi under Grant Guike AA22068072.

ABSTRACT The vulnerability of Deep Neural Networks (DNNs) to adversarial perturbations has been demonstrated in a large body of research. Compared to image-dependent adversarial perturbations, universal adversarial perturbations(UAPs) is more challenging for indiscriminately attacking the model inputs. However, there are few studies on generating data-free targeted UAPs and the targeted attack success rate of the latest method remains unsatisfactory. Not only that, fewer studies have implemented their approach on Transformers and its efficacy remains uncertain. Therefore, a novel method denoted as Denoising Targeted UAP (DT-UAP) is proposed in this paper that considers the training input as the noise, and takes the input of the last layer into calculation. Specifically, the proposed method minimizes the distance between perturbations and adversarial examples, then incorporates a targeted loss function to generate targeted universal adversarial perturbations for different DNNs and Transformers based on different proxy datasets. DT-UAP has achieved an average improvement of 5% to 10% in terms of both fooling rate and targeted fooling rate comparing to the most recent method for generating targeted universal adversarial perturbation with proxy dataset for DNNs. Additionally, DT-UAP has also achieved a targeted attack success rate of over 80% on Transformers such as MaxVit and SwinTransformer.

INDEX TERMS Targeted universal adversarial perturbation, adversarial example, deep neural network, transformer, image as noise, proxy dataset.

I. INTRODUCTION

Deep Neural Networks have been extensively proven to be vulnerable to adversarial perturbations. A well-trained model could generate incorrect or attacker-specified outputs when these maliciously crafted and imperceptible perturbations are overlayed with natural images. The concept of adversarial perturbations was firstly introduced by Szegedy et al. [1]. They added the elaborately crafted perturbations to an image and made the model to misclassify it. These perturbations are known as image-dependent adversarial perturbations. However, universal adversarial perturbations (UAPs) are not specific to a single image, but can cause most of normal images input to DNNs to be misclassified, which was proposed by Moosavi-Dezfooli et al. [2]. Compared to image-dependent adversarial perturbations, the study of universal adversarial perturbations poses a greater challenge to the security of DNNs. Universal adversarial perturbations can indiscriminately attack any image input to DNNs. It allows attackers to launch real-time attacks on target models using a single well-trained universal adversarial perturbation, while it is not possible for image-dependent adversarial perturbations. There are numerous fascinating attributes for UAPs, such as dominant labels in non-targeted universal attacks, the original transferability of UAPs, and so on. However, few studies have explained these phenomena. Moreover, only a limited number of works are related to the crafting of data-free targeted UAPs, while recent works primarily focus on transferable non-targeted universal attacks. The success rate of current targeted universal attack works is unsatisfactory, warranting further research in this area. It should also be noted that only a few studies have applied their methods to transformers, and the performance of their methods on transformers remains unknown.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

Based on these facts above, there is an urgent need for additional research on targeted universal attacks, especially under data-free conditions, which motivates our current study.

Several studies have made significant contributions to the study of universal adversarial perturbations in recent years. Zhang et al. [33] discovered a strong similarity between universal adversarial perturbations and adversarial examples. They suggested that targeted universal adversarial perturbations themselves (independent of the images to attack) are features, while images behave like noise to them. In addition, Ye et al. [28] leveraged the phenomenon of Neural Collapse (NC) in their model and proposed FG-UAP. By calculating the loss based on the input to the last layer of the model, it effectively improved the attack effectiveness of the crafted UAP.

This paper is motivated by the aforementioned works and intriguing issues of universal perturbations and presents a novel method DT-UAP for generating targeted universal adversarial perturbations with proxy dataset. The generating process is illustrated in Figure 1. The performance of DT-UAP on different datasets is discussed along with the relationship between the targeted UAP, the model and the training dataset. The transferability of DT-UAP is also analyzed, which has explored the issue of weak transferability of targeted UAP. And DT-UAP was conducted on the latest transformer models to further evaluate attack performance. In summary, the main contributions of this paper are as follows

- It is suggested that training images can be considered as noise that affects the robustness of model output from target UAPs.
- A novel method, DT-UAP, is proposed for generating targeted UAPs with the proxy dataset by distinctively computing the distance between UAPs and adversarial examples, and then combining it with the Mean Square Error and Cross Entropy loss.
- To verify the effectiveness of the proposed method, the ImageNet and Place365 are used for the proxy datasets to attack the prevalent models of both DNNs and Transformers.
- The original transferability of the UAPs generated by DT-UAP is analyzed to provide an explanation for the transferable UAPs based on the experimental data.

II. RELATED WORK

A. IMAGE-DEPENDENT ATTACKS

The concept of adversarial perturbations was first introduced by Szegedy et al. [1]. They generated adversarial perturbations by using the L-BFGS method with optimization and specified constraints in 2013. Subsequently, numerous researchers attempted to explain the existence of adversarial perturbations. In 2014, Goodfellow et al. proposed the Fast Gradient Sign Method(FGSM) [3], which rapidly generated adversarial perturbations. They suggested that the existence of these perturbations was due to the local linearity of DNNs. This hypothesis has been supported and validated by subsequent works of the variants of FGSM, such as I-FGSM [6], MI-FGSM [5], DI-FGSM [4], and VMI-FGSM VMI-FGSM [7]. However, the linear hypothesis failed to explain the existence of non-linear adversarial examples [8] and the strong robustness of many linear models against adversarial perturbations [9], [10], [11]. Moosavi-Dezfooli et al. [12] introduced the concept of decision boundaries and developed the DeepFool method in 2016, which iteratively optimized the loss function to move the adversarial samples towards the decision boundary and eventually surpass it. Carlini and Wagner proposed the C&W method [13], which minimized the generated adversarial perturbations by optimizing the loss function using an optimizer. Another variant of FGSM, called Projected Gradient Descent (PGD) [8], argued that for a non-linear model, the direction obtained in a single iteration using FGSM was not necessarily the best. PGD incorporated an iterative process that repeatedly searched for the strongest adversarial examples within a specified range by controlling the perturbation size using the parameter ϵ. PGD is currently regarded as one of the most efficient attack methods to date. All of the aforementioned works are part of white-box attacks, where the target model is accessible. However, the information about the target model is unknowable in the scenario of black-box or transferable attacks, only the information of training data is required. The first black-box based attack was proposed by Papernoe et al. [14] in 2017, which generated substitute models to simulate the decision boundaries of the target model. The same year, Liu et al. [15] introduced the idea of ensemble model. In 2020, generative adversarial networks(GANs) were introduced to generate synthetic samples by Mingyi et al. [16]. Li et al. [17] combined differential evolution(DE) strategy with approximated gradient sign method to deploy black-box attack in 2022. And Giulivi et al. [18] utilized the form of the scratches in images to generate more deployable attacks in 2023.

B. IMAGE-AGNOSTIC ATTACKS

Universal adversarial attacks, also known as image-agnostic attacks, were first introduced by Moosavi-Dezfooli et al. [2] in 2017. They iteratively applied the DeepFool method [12] to generate UAPs that could cause most input images to yield incorrect outputs from DNNs. Poursaeed et al. proposed the GAP method [19], which utilized generative adversarial networks (GANs) to generate UAPs. As research progressed, efforts were made to construct data-free UAPs that do not rely on the original training dataset. Khrulkov et al. [20] constructed UAPs using the Jacobian matrix of hidden layers in the network. The Fast Feature Fool [21] approach utilized the mean activation outputs of intermediate layers in convolutional neural networks (CNNs) as a loss function to generate UAPs. Additionally, other works such as [22], [23], [24], and [25] have also attempted to construct data-free UAPs. The most recent approach for crafting data-free UAPs was intro-

duced by Li et al. [26] in 2022, where the instance-specific and universal attacks were integrated through a feature-based approach. In 2021, Zhang et al. proposed Cosine-UAP [27]. They treated the logit outputs of DNNs between the original image and the adversarial sample as high-dimensional vectors. Specifically, they generated UAPs by minimizing the cosine similarity between these vectors through iterative optimization. And an improved method FG-UAP [28] was introduced by Ye in 2023, which utilized the input of the last layer of the model as the model output. It further enhanced the effectiveness of UAP attacks. In 2023, Zhang et al. [29] introduced the spatial transformation technique for universal attacks, which eliminated the need for additive perturbations. Liu et al. [30] suggested that gradient aggregation presents an efficient means of augmenting the efficacy of universal attacks.

The methods mentioned above primarily focus on constructing non-targeted UAPs, yet there are fewer methods for constructing targeted UAPs. CD-UAP [31] specifically attacks images of a particular class, without affecting normal images of other classes. And DTA [32] overlays the targeted class image with the constructed UAP to redirect it to a pre-specified class. To the best of our knowledge, the latest method for constructing targeted UAPs with proxy datasets is the DF-UAP method proposed by Zhang et al. [33] in 2020. It explored the similarities between normal images, adversarial examples and the UAPs. Then they constructed targeted UAPs based on multiple loss functions, which has achieved a fabulous targeted attack performance. In 2023, Ma et al. [34] proposed an approach for class-balanced UAPs to enlarge the dispersion of the predicted labels for universal attacks. Weng et al. [35] leverage the Kullback–Leibler (KL) divergence loss to implement both targeted and non-targeted universal attacks. And universal adversarial attacks can be adopted to different tasks such as remote sensing [36], text recognition [37], watermarking [38], object detection [39] and so on.

III. CRAFT TARGETED UAPS BY DENOISING

The detailed reasoning and implementing process of DT-UAP are demonstrated in this section.

A. PROBLEM DEFINITION

In the image classification task, given a distribution of images $X \in \mathbb{R}^d$, where $X = \{x_1, x_2, \ldots, x_N\}$. By defining a classification function $f(\cdot)$, the classification process can be represented as $y = f(x)$, where $x \in X$. The objective of generating targeted universal adversarial perturbations is to find a perturbation vector $v \in \mathbb{R}^d$ that, when added to x, causes the predicted result of $f(\cdot)$ to direct most $x \in X$ to our predefined target label y_{target}. Furthermore, the perturbation vector v is subject to an l_p norm constraint, such that $|v|_p \leq \epsilon$, where ϵ is a given threshold that determines the magnitude of the influence of v on the image x. With all the requirements, the objective function for constructing targeted UAPs can be formulated as follows

$$C(x) = \begin{cases} 1, & f(x+v) = y_{target}. \\ 0, & otherwise. \end{cases}$$
$$\arg\max_{v} \sum_{i=0}^{N} C(x_i), \quad s.t. \quad \|v\|_p \leq \epsilon, \quad x \in \mathbb{R}^d \tag{1}$$

In the case of non-targeted universal adversarial perturbations, the objective is to maximize the change in the original output of $f(\cdot)$ by adding the perturbation vector v to the input image x. The specific objective function can be formulated as follows

$$\hat{C}(x) = \begin{cases} 1, & f(x+v) = f(x) \\ 0, & otherwise. \end{cases}$$
$$\arg\min_{v} \sum_{i=0}^{N} \hat{C}(x_i), \quad s.t. \quad \|v\|_p \leq \epsilon, \quad x \in \mathbb{R}^d \tag{2}$$

Indeed, it is evident that generating targeted universal adversarial perturbations is significantly more challenging than non-targeted ones. Because targeted universal adversarial perturbations need to not only change the original output result but also align the output towards a specific target class. In the case of targeted attacks, the objective is to find a perturbation vector v that, when added to the input image x, not only maximizes the change in the output of $f(\cdot)$, but also redirects the output towards a predefined target class y_{target}. It adds an additional constraint to the optimization problem, which makes it more difficult to achieve a successful targeted attack. A deeper understanding of decision boundaries of the target model and the relationship among different classes are required in generating targeted universal adversarial perturbations. It involves finding a delicate balance between perturbing the image to induce a misclassification and ensuring that the perturbation drives the output towards the desired target class. Due to this additional requirement of aligning the output towards the target class, the generation of targeted universal adversarial perturbations is more intricate and computationally demanding compared to non-targeted perturbations.

B. MOTIVATION

This paper is initially inspired by the work of Zhang et al. [33]. In their study, the authors discussed the similarities between natural images, adversarial perturbations, and adversarial examples that were the images obtained by adding the perturbations to the original images.

The similarity mentioned above can be represented by a distance function denoted as $Dis(X, Y)$, where X or Y denotes the respective vectors. A smaller value of $Dis(\cdot)$ indicates higher similarity, while a larger value of $Dis(\cdot)$ indicates lower similarity. There are several metric options available to calculate $Dis(\cdot)$, including but not limited to Euclidean Distance, Manhattan Distance, Normalized Euclidean Distance, Negative Cosine Similarity, Negative Pearson Correlation Coefficient, and others.

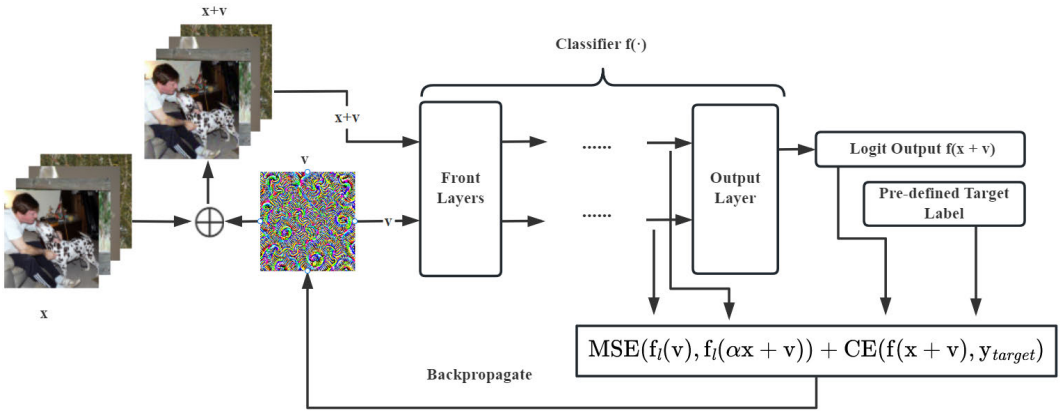


FIGURE 1. Procedure for generating targeted UAPs.

Zhang et al. found that for image-dependent adversarial perturbations, both targeted and non-targeted, the impact on natural images resembled Gaussian noise. In other words, when considering the similarity between natural images x and their corresponding adversarial example $x + v$, the value of $Dis(x, x + v)$ was found to be smaller than the distance between the perturbations v and the adversarial example $x + v$. It implied that image-based adversarial perturbations reduced the similarity between natural images and the corresponding adversarial examples by introducing noise.

However, the situation was different for universal adversarial perturbations, where the similarities between these three components exhibited completely different patterns. For non-targeted UAPs, $Dis(v, x + v)$ was smaller than $Dis(x, x + v)$. Especially for targeted UAPs, $Dis(v, x + v)$ was significantly smaller than $Dis(x, x + v)$. In this case, the behavior of the image resembled noise. Based on these observations, the proposed method in our paper considers the original input image as noise and continuously optimizes to reduce $Dis(v, x + v)$ by minimizing the influence of the input image on the logit outputs of targeted UAPs. The ultimate goal is to generate targeted UAPs that, when added to the input image and fed into the classification network, yield robust output results with minimal changes compared to the original outputs. Furthermore, inspired by the work of Ye [28], our method chooses to calculate the similarity using the input of the final layer of the neural network. Rather than calculating with the logit outputs directly, it enhances the effectiveness of targeted UAP attacks.

C. TARGETED ATTACK BY DENOISING

After stating the reasoning process of our viewpoint, now we will describe the specific implementation process of DT-UAP in detail below. A random perturbation vector v is initialized and added to the normal image x to create an adversarial sample $x + v$. In many existing methods for generating UAPs such as Cosine-UAP [27], FG-UAP [28], and DF-UAP [33], x and $x + v$ are input to the model, and the optimization of the perturbation v is performed by computing the relationship between these two inputs. However, what distinguishes our approach from the others is that we input v and $x + v$ into the model and compute the resemblance between their outputs, while other methods compute between x and $x + v$. It corresponds to the left part of Figure1. In our method, training images are considered as input noise that have a significant impact on the original output of the UAP. However, if the initial noise is too strong, it will be difficult to find the correct optimization direction during training. Therefore, a parameter α is introduced to vary during the training process to control the magnitude of the input noise, where $\alpha \in [0, 1]$. To continuously increase the similarity between v and $x + v$ during the training process, a distance function is required to measure the distance or similarity between them, denoted as $Dis(v, x + v)$. In the proposed approach, Mean Squared Error (MSE) is chosen to represent the distance between v and $x + v$, which is calculated as

$$Dis(X, \widehat{X}) = MSE(X, \widehat{X}) = \frac{1}{S}\sum_{i=1}^{S}(X_i - \widehat{X}_i)^2, \quad X, \widehat{X} \in \mathbb{R}^d \tag{3}$$

where S denotes the sum of all the points in X. Other distance metrics such as Cosine Similarity, Euclidean Distance, Mahalanobis Distance, and Pearson Correlation Coefficient, have also been conducted to measure the attack performance. However, the experimental results showed that they were inferior to Mean Squared Error (MSE), so MSE is ultimately chosen as the distance metric. Then following the insight from FG-UAP [28], to avoid the phenomenon of neural collapse [50] and achieve more efficient generation of universal adversarial perturbations, the logit outputs $f(\cdot)$ are not directly utilized for distance calculation. Instead, it is computed between the inputs of the neural network's last layer, denoted as $f_l(\cdot)$. The ultimate objective is to boost the logit output of the target class relative to the remaining classes. Therefore, the Cross-Entropy loss function that calculates the discrepancy between the target class y_{target} and

Algorithm 1 Denoising Targeted UAPs

Input: Train Dataset X, classifier f, output layer's input f_l, batch size m, max epoch N, UAP magnitude ϵ, target label y_{target}, image noise magnitude α

Output: Targeted UAP vector v

Randomly Initialize the vector v
for epoch=1,2,...,N **do**
  $B \sim X, |B| = m$ Randomly Sample B
  $g_v \leftarrow \mathop{\mathbb{E}}_{x \sim B} [\bigtriangledown MSE(f_l(v), f_l(\alpha x + v)) + \bigtriangledown CE(f(\alpha x + v), y_{target}]$ $\bigtriangledown$ *Gradient*
  $v \leftarrow optim(g_v)$
  $v \leftarrow clamp(v, -\epsilon, \epsilon)$
end for

TABLE 1. Targeted fooling rate(%) of the proposed mothod compared to other white-box targeted universal attack. Results are separated by accessing to the original ImageNet training data (upper) and data-free methods (lower). 8 classes are randomly selected and the average value is reported. Difference values are shown in below.

Method	Alexnet	GoogleNet	VGG16	VGG19	Resnet50	ResNet152
DF-UAP [34]	48.6	59.9	75.0	71.6	-	66.3
NKL-UAP [36]	-	-	65.99	-	65.44	-
Ours	**65.6**	**69.98**	**82.88**	**83.62**	**78.85**	**79.51**
DF-UAP [34]	42.6	60.0	73.4	64.5	-	62.5
ENK-UAP [36]	-	-	-	-	-	-
Ours(Place365)	**54.67**	**65.09**	**75.6**	**70.4**	**74.07**	**63.34**

the logit output of $x + v$, is incorporated into the current loss function. It can be formulated as follows

$$\begin{aligned} &Cross\_Entropy(f(x), y_{target}) \\ &= -\log(\frac{e^{f(x)[y_{target}]}}{\sum_{i=1}^{K} e^{f(x)[i]}}) \\ &= -f(x)[y_{target}] + \log(\sum_{i=1}^{K} e^{f(x)[i]}), \quad x \in \mathbb{R}^d \end{aligned} \tag{4}$$

where K represents the number of the classes. The overall objective function can be expressed as a combination of the original loss function and the additional cross-entropy loss. Based on the above, the loss function for generating targeted universal adversarial perturbations can be constructed as follows

$$L_T = MSE(f_l(v), f_l(\alpha x + v)) + CE(f(\alpha x + v), y_{target}) \tag{5}$$

The construction process of targeted universal adversarial perturbations is illustrated in the diagram in Figure 1. The detailed algorithm can be found in Algorithm 1. In this process, the training images are considered as noise that affects the robustness of model outputs of the perturbation. Therefore, the entire training process can be described as a denoising process, referring to this method as Denoising Targeted UAP (DT-UAP).

TABLE 2. Comparison of the proposed method to other data-free methods (Without access to the original Imagenet training data). The reported non-targeted fooling ratio(%) represents the mean value over 8 different target classes in targeted attack.

Method	Alexnet	GoogleNet	VGG16	VGG19	ResNet152
FFF [22]	80.92	56.44	47.1	43.62	29.78
GD-UAP [23]	87.02	71.44	63.08	64.67	37.3
AAA [24]	89.04	75.28	71.59	72.84	60.72
DF-UAP(COCO) [34]	89.9	76.8	92.2	91.6	**79.9**
Jigsaw image [28]	91.07	**87.57**	89.48	86.81	65.35
AT-UAP [27]	**96.66**	82.60	**94.50**	**92.85**	73.15
Ours(Place365)	90.5	78.18	90.3	91.1	73.74

TABLE 3. Performance of the proposed method against Transformers. Results are divided into non-targeted fooling ratio(upper) and targeted fooling ratio(lower). Access to the Imagenet training data is also considered(with data/data-free). 8 classes are randomly selected in the scenario of targeted attack, and the average value is reported.

Method	Swin	Swin_v2	MaxVit	Deit_tiny	Deit_small	Deit_base
with data	91.7	80.61	85.92	78.9	84.89	75.95
data-free(Place365)	86.4	85.34	85.02	73.77	74.81	63.49
with data	87.86	79.2	85.29	76.9	83.51	75.09
data-free(Place365)	82.71	83.68	84.1	71.79	72.29	61.84

IV. EXPERIMENTS

In our experiment, six DNN models are employed: AlexNet [40], GoogleNet [43], VGG16 [44], VGG19 [44], ResNet50 [45], and ResNet152 [45]. Six Transformer models are also included: MaxVit [47], SwinTransformer [48], SwinTransformer_v2 [49], Deit_tiny [46], Deit_small [46], and Deit_base [46]. These models are all pre-trained on the full ImageNet and have great performance on image classification tasks. Two datasets were used as surrogate training datasets: the classic ImageNet dataset [41] and the Places365 dataset [42]. The hyperparameters in Algorithm 1 were set as follows: the number of epochs N=10, the batch size m=16, the perturbation magnitude $\epsilon = 10/255$, Adam optimizer with a learning rate of 0.01 and weight decay of 1e-10. The experiments were conducted on an NVIDIA GeForce RTX 3070 GPU, and the code was implemented on PyTorch.

A. PERFORMANCE AND COMPARISON

Our attacks are firstly conducted on commonly used DNN models and trained the perturbations using popular datasets for image recognition tasks, the ImageNet and the Places365. Table 1 has shown the results comparing with the DF-UAP, which also constructs targeted universal adversarial perturbations. Compared to the DF-UAP method, DT-UAP achieves an average improvement of approximately 5% to 10% in terms of fooling rate and targeted fooling rate. The fooling rate indicates the percentage of model output that was fooled, and the targeted fooling rate indicates the percentage of model output that was redirected to the

TABLE 4. After selecting appropriate target classes(best performance class), the transfer attack capability of targeted UAPs generated by DT-UAP in terms of fooling rate(left) and targeted fooling rate(right) across both DNN models and Transformer models. The values in bold are the white-box attacks.

Model	AlexNet	GoogleNet	VGG16	VGG19	ResNet50	ResNet152	Swin_tiny	Swin_v2_tiny	MaxVit_tiny	Deit_tiny	Deit_small	Deit_base
AlexNet(109)	**97.08 83.43**	54.65 9.68	63.4 11.31	59.81 8.77	44.44 7.69	36.18 3.34	20.45 2.11	20.64 1.9	12.24 0.81	33.03 7.24	20.14 3.5	15.13 1.17
GoogleNet(109)	51.76 0.86	**92.47 86.39**	75.18 30.74	73.31 25.92	61.66 29.6	51.9 18.86	23.06 8.55	24.03 9.87	12.41 2.06	31.87 14.61	21.36 9.96	14.28 3.55
VGG16(109)	43.83 0.39	48.75 11.89	**96.47 89.73**	90.46 68.18	56.53 19.6	42.2 9.14	27.09 10.84	26.68 11.12	14.22 3.05	26.1 8.12	19.22 8.64	12.98 2.32
VGG19(109)	45.68 0.54	51.72 15.2	93.15 75.54	**96.71 90.08**	56.09 18.58	44.69 10.58	26.14 10.4	26.27 12.65	13.7 2.75	26.46 8.5	17.04 5.1	12.6 0.89
ResNet50(971)	48.44 0.05	41.99 0.73	58.42 1.71	57.27 2.95	**93.86 89.79**	55.32 35.05	18.71 1.65	19.36 1.81	11.18 0.8	22.07 0.21	14.39 0.17	11.4 0.65
ResNet152(854)	48.48 0.3	56.68 25.27	84.13 51.42	79.03 18.54	81.13 62.33	**94.36 88.97**	36.87 28.07	28.95 14.86	12.96 1.57	23.83 1.22	16.7 4.82	14.1 5.03
Swin_t(805)	31.61 0.04	19.78 0.02	27.15 0.02	26.06 0.03	19 0.03	15.69 0.03	**89.59 88.57**	24.72 14.91	8.77 0.29	15.36 0.02	9.73 0.02	9.1 0.04
Swin_v2_t(558)	32.04 0.06	18.45 0.0	26.88 0.03	25.88 0.06	18.47 0.02	14.55 0.02	16.57 3.66	**86.09 84.81**	8.57 0.03	15.82 0.05	10.1 0.02	9.1 0.05
MaxVit_t(892)	34.6 0.08	20.81 0.08	33.45 0.04	32.37 0.06	20.99 0.11	17.12 0.07	14.54 1.48	14.36 0.75	**87.72 87.19**	15.58 0.08	10.16 0.08	8.76 0.14
Deit_tiny(815)	56.15 2.55	48.28 7.69	58.06 3.39	57.37 4.07	39.88 6.18	33.77 3.64	22.41 8.39	22.95 10.14	11.11 0.98	**95.61 94.68**	31.97 22.76	20.7 12.77
Deit_small(828)	46.33 0.13	27.29 0.35	41 0.24	38.31 0.13	24.44 0.22	20.73 0.33	15.92 0.63	15.57 0.89	10.25 0.3	28.68 6.06	**88.91 87.64**	20.54 14.06
Deit_base(879)	48.4 0.1	31.5 0.14	43.53 0.07	41 0.11	27.12 0.11	21.96 0.02	16.32 0.11	16.39 0.28	9.96 0.0	27.04 0.69	28.97 19.37	**91.89 91.46**

pre-defined target label. The higher value of the fooling rate and targeted fooling rate indicates a preferable attacking performance. Regardless of whether it is a DNN model or a Transformer model, there are obvious differences in attack effectiveness when selecting different target attack classes, typically ranging from 10% to 20%. Moreover, when selecting the Dominant-Label [2] as the target class, both DNN models and Transformer models achieve the highest fooling rate and targeted fooling rate.

Compared to DNN models, when attacking Transformer models, the Dominant-Label is not the only effective target class. In other words, DT-UAP can achieve superior attack performance on multiple classes, not just the Dominant-Label. When using different datasets as training datasets, but with the same target class, the differences in attack results are not significant. However, these datasets describe totally different objects. Based on this result, randomly generated Gaussian noise is utilized as the training dataset. But it leads to a significantly reduced fooling rate and a near-zero success rate for targeted attacks.

The proposed method is also compared with other data-free UAP methods in the metric of non-targeted fooling rate to further evaluate attacking effectiveness in Table 2. It has been observed that our attack success rate is closely matched by the latest method, but it is undeniable that the non-targeted fooling rate of our proposed method is not the best. We will explain the specific reasons below. First of all, our non-targeted attack's success rate is not due to our altering the loss function, but rather to the non-targeted attack capability of the targeted universal perturbation generated by our proposed method. In other words, the success rate of non-targeted attacks we have calculated is the average of the attack effects after selecting 8 different target classes, not the highest value. This had also been illustrated in [35]. Secondly, it is easier to implement non-targeted universal attacks than targeted universal attacks. The former only involves modifying the original output results of the model, whereas the latter involves modifying the original output results while guiding the model's output towards a predefined target class. This also precisely demonstrates the capability of targeted attacks to conduct non-targeted attacks. The explanation for the difference between targeted fooling ratio and non-targeted fooling ratio from the same targeted universal perturbation is that targeted universal perturbations result in a change in the model's output, but they do not necessarily push it into the target class we have predefined. As for why targeted universal attacks have different attack effects on different classes, we believe that the model learns different levels of features for different classes. A general classification model's accuracy across different classes varies. Our attack depends on the specific model, which directly results in varying attack effects on different classes.

Additionally, DT-UAP is also applicable to popular Transformer models, which have gained popularity in recent years. The attack effectiveness of DT-UAP on Transformer models is shown in Table 3, which demonstrates a great attack performance. In DNN models, the difference between fooling rate and targeted fooling rate can reach up to 30%, while in Transformer models, the difference is at most 3%. It indicates that DT-UAP demonstrates higher stability and is more suitable for Transformer models.

B. TRANSFERABILITY

Based on the understanding of previous related works, it has been observed that universal adversarial perturbations themselves possess certain transferability in terms of their attack capabilities. Therefore, the best-performing targeted universal perturbations for each model are selected to evaluate their transferability by conducting DT-UAP transfer attacks on all twelve models. The results are presented in Table 4.

Overall, the transferability of fooling rate is significantly higher than targeted fooling rate. And in many cases, the transferability of target rates is close to zero across different models. This discrepancy in transferability may be attributed to the significant differences between models,

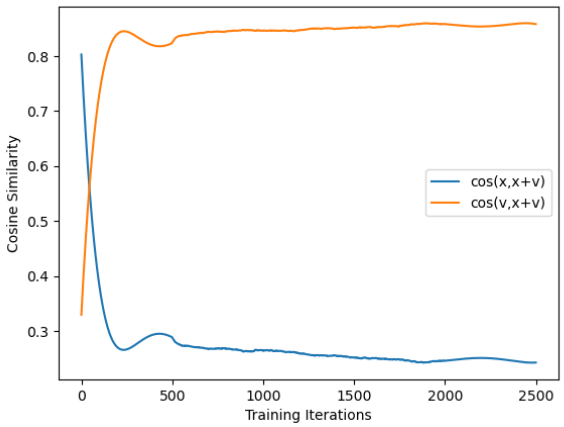
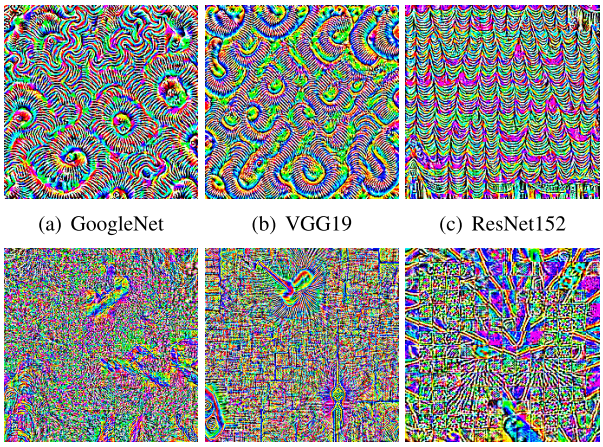


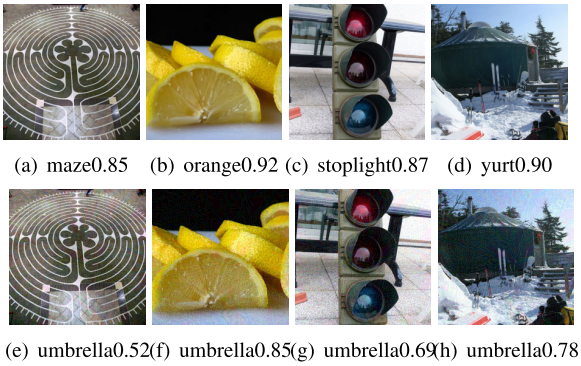
FIGURE 2. The variation of the cosine similarity between x, v, and $x + v$ during the training process.



(a) GoogleNet (b) VGG19 (c) ResNet152

(d) Swin_v2_tiny (e) MaxVit_tiny (f) Deit_base

FIGURE 3. The targeted UAPs generated with DT-UAP.



(a) maze0.85 (b) orange0.92 (c) stoplight0.87 (d) yurt0.90

(e) umbrella0.52 (f) umbrella0.85 (g) umbrella0.69 (h) umbrella0.78

FIGURE 4. For the Deit_base model, natural image(top) and the adversarial examples after applying the perturbations with target class 'umbrella'(bottom).

where fooling the model to produce incorrect outputs is easier than redirecting to specific classes.

The transferability of targeted universal perturbations is stronger when models are more similar. This is particularly evident in the case of VGG16 and VGG19, where the transferability is relatively strong. However, this phenomenon does not hold for Transformer models. The transferability among Deit_tiny, Deit_small, and Deit_base remains low. Similarly, the transferability between Swin_tiny and Swin_v2_tiny is also unsatisfactory. It suggests that the transferability of targeted perturbations is influenced by factors beyond the similarity between the models, and Transformer models have demonstrated different transferability patterns compared to DNN models.

V. DISCUSSIONS

A. SIMILARITIES

By measuring the cosine similarity, Figure 2 has shown the changes observed in the similarity between the original image x, the perturbation v, and the perturbed image $x+v$. The result shows that the generated targeted UAP indeed reduces the similarity between x and $x+v$ while increasing the similarity between v and $x+v$, further confirming the effectiveness of DT-UAP. Based on Figure 2, we discovered that the similarity between the perturbation v and the adversarial sample $x+v$ increases in the initial training stage, reaches a maximum, and then decreases. After decreasing to a certain degree, the similarity rises again and continues to rise steadily. However, the process of falling and then rising in the middle is an intriguing issue. As mentioned previously, our belief remains that a well-trained universal adversarial perturbation's model output is robust, with the superimposed image only affecting its robustness as noise. This view is the one that has consistently been held in our paper. However, there is a risk of encountering a local extremum problem. In the early stages of training, as the process continues, the perturbation v will be optimized in our predetermined direction. At this stage, the model's output against perturbation v lacks robustness due to inadequate training data and a limited number of iterations. Each training iteration's gradient superposition significantly alters the outcome of v's model output, leading to an optimization direction change, thus causing a decline in similarity. However, as the training data and training times increase, the ongoing comprehension of the model leads to more robust model output. Consequently, similarity gradually increases during subsequent training sessions.

B. VISUALIZATIONS

The impact of different datasets on the effectiveness of generated targeted UAP is not significant. It is believed that the generation of targeted UAP depends more on the model rather than the training dataset. In other words, it has limited correlationships with the dataset. Additionally, by observing the generated UAP images in Figure 3, it is noticed that there are noticeable differences between the UAP images generated for DNN models and Transformer models. UAPs generated for DNN models tend to be more continuous, while UAPs generated for Transformer models are more discrete and modular. The observation above is particularly evident in the UAPs generated for Deit models, where the 224×224 images

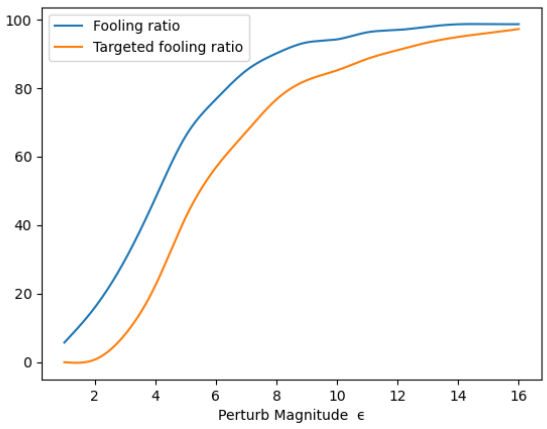


FIGURE 5. The variation of the attack performance with the varying perturb magnitude ϵ, which denotes the maximum value of the perturbation in the pixel value in [0,255].

are divided into 14 × 14 rectangles that are 16 × 16-sized, which aligns with the basic architecture of Deit models. It further validates the notion that targeted UAPs depend more on the model. And the natural images and the adversarial examples overlaid with UAPs are shown in Figure 4.

C. ABLATION STUDIES

The discussion about the variation of the attack performance was conducted for constant epsilon values ranging from 1 to 16 with the pixel value in [0,255] when applied to VGG16, as illustrated in Figure 5. In most white-box attacks, the epsilon was set as 10 (our setting) while it was 16 in the black-box setting. It was apparent that the targeted and non-targeted fooling ratio could reach 100% if the epsilon was set to 16. And higher value of epsilon indicated that we could make greater changes to the image, but it also led to more perceptible perturbations. There are typically two methods to limit the magnitude of the perturbations. One is to continuously optimize the perturbation through an optimizer to achieve better attack performance while minimizing the perturbation. However, a major issue with this approach is the lack of direct control over the impact of perturbations on the image, which may result in excessive changes to the image, making it difficult for human observers to detect these perturbations or causing incorrect identification due to excessive interference. Therefore, the advantages of this approach are quite limited. Another more commonly used method is to directly limit the maximum value of the perturbation, in which we can more intuitively limit and observe the impact of the perturbation on natural images. As shown in the figure, when limiting the maximum perturbation value to less than 8, the target and non-target attack performances are greatly improved as the ϵ increases. When the value of ϵ is between 8 and 10, the success rate of both attacks tends to be stable. Here we view it as a trade-off between attack effectiveness and magnitude of the perturbation, taking into account both attack effectiveness and limiting the magnitude of perturbation as much as possible. This is also the fundamental idea behind the first method for constraining the magnitude of perturbations, indicating that we can benefit from both approaches by selecting an appropriate value for epsilon. When the limit on ϵ is further relaxed, although the attack effectiveness may be limitedly improved, this improvement comes at a greater expense to the visual quality of the adversarial examples.

D. FURTHER DISCUSSIONS

Based on the analysis of the transferability result, it is believed that improving the transfer attack capability of targeted UAPs should be approached from the perspective of the model. The similarity between models significantly influences the transferability of UAPs, while improving the transferability from the perspective of the dataset is not practical or has limited effectiveness. The initial intention behind designing this method is based on the idea that targeted UAPs act as noise that affects the robustness of the model's output. The obtained attack results further validate this idea. However, there is still a need to explore why perturbations that are significantly smaller in magnitude compared to the normal image can have a greater impact on the model, or in other words, what are the distinct characteristics of targeted UAPs that cannot be concealed. This is a issue that requires further exploration.

VI. CONCLUSION

In this paper, a novel method called DT-UAP is proposed for generating targeted universal adversarial perturbations based on the idea that training images act as noise that affects the robustness model outputs of targeted UAPs. The ImageNet and Places365 are utilized as training datasets, and the proposed method aim to minimize the distance between the perturbation v and the adversarial examples $x + v$ after taking the input of the model's final layer. Then a directional loss function is introduced to create targeted universal adversarial perturbations, which steer almost all samples toward the predefined target class. The results have demonstrated that DT-UAP has achieved an improved targeted attacking effectiveness not only on DNN models but also on Transformer models. The transferability of targeted UAP is also discussed and analyzed, revealing that the generation of targeted UAP relies more on the model rather than the training data. It is prospective to integrate the idea of considering the training images as noise with existing black-box attack techniques to explore the methods for improving the transferability of universal adversarial perturbations. Additionally, the reasons behind the relatively low transferability of targeted universal adversarial perturbations and the theoretical explanations for the existence of universal adversarial perturbations should also be investigated in future works.

ACKNOWLEDGMENT

The authors would like to thank Huajiang Huigu Laboratory No. 306, Guilin University of Electronic Technology, for providing experimental equipment and space for this work.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[2] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," Le Centre pour la Commun., Scientifique Directe, HAL, memSIC, Tech. Rep., 2017.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[4] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2730–2739.

[5] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[6] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Boca Raton, FL, USA: CRC Press, 2018, pp. 99–112.

[7] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1924–1933.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[9] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[10] P. Tabacof and E. Valle, "Exploring the space of adversarial images," 2015, *arXiv:1510.05328*.

[11] T. Tanay and L. Griffin, "A boundary tilting persepective on the phenomenon of adversarial examples," 2016, *arXiv:1608.07690*.

[12] 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. (CVPR)*, Jun. 2016, pp. 2574–2582.

[13] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[14] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.

[15] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*.

[16] M. Zhou, J. Wu, Y. Liu, S. Liu, and C. Zhu, "DaST: Data-free substitute training for adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 234–243.

[17] C. Li, H. Wang, J. Zhang, W. Yao, and T. Jiang, "An approximated gradient sign method using differential evolution for black-box adversarial attack," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 976–990, Oct. 2022.

[18] L. Giulivi, M. Jere, L. Rossi, F. Koushanfar, G. Ciocarlie, B. Hitaj, and G. Boracchi, "Adversarial scratches: Deployable attacks to CNN classifiers," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 108985.

[19] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4422–4431.

[20] I. Oseledets and V. Khrulkov, "Art of singular vectors and universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8562–8570.

[21] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," 2017, *arXiv:1707.05572*.

[22] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2452–2465, Oct. 2019.

[23] K. R. Mopuri, P. K. Uppala, and R. V. Babu, "Ask, acquire, and attack: Data-free UAP generation using class impressions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.

[24] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu, and F. Huang, "Universal adversarial perturbation via prior driven uncertainty approximation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2941–2949.

[25] Y. Li, S. Bai, C. Xie, Z. Liao, X. Shen, and A. Yuille, "Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, Aug. 2020, pp. 795–813.

[26] M. Li, Y. Yang, K. Wei, X. Yang, and H. Huang, "Learning universal adversarial perturbation by adversarial example," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 2, pp. 1350–1358.

[27] C. Zhang, P. Benz, A. Karjauv, and I. S. Kweon, "Data-free universal adversarial perturbation and black-box attack," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7868–7877.

[28] Z. Ye, X. Cheng, and X. Huang, "FG-UAP: Feature-gathering universal adversarial perturbation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[29] Y. Zhang, W. Ruan, F. Wang, and X. Huang, "Generalizing universal adversarial perturbations for deep neural networks," *Mach. Learn.*, vol. 112, no. 5, pp. 1597–1626, May 2023.

[30] X. Liu, Y. Zhong, Y. Zhang, L. Qin, and W. Deng, "Enhancing generalization of universal adversarial perturbation through gradient aggregation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 4435–4444.

[31] T. Gupta, A. Sinha, N. Kumari, M. Singh, and B. Krishnamurthy, "A method for computing class-wise universal adversarial perturbations," 2019, *arXiv:1912.00466*.

[32] P. Benz, C. Zhang, T. Imtiaz, and I. S. Kweon, "Double targeted universal adversarial perturbations," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–17.

[33] C. Zhang, P. Benz, T. Imtiaz, and I. S. Kweon, "Understanding adversarial examples from the mutual influence of images and perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14521–14530.

[34] K. Ma, G. Cao, M. Xu, C. Wu, H. Wang, and W. Cao, "Class-balanced universal perturbations for adversarial training," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[35] J. Weng, Z. Luo, Z. Zhong, D. Lin, and S. Li, "Exploring non-target knowledge for improving ensemble universal adversarial attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 3, pp. 2768–2775.

[36] T. Bai, H. Wang, and B. Wen, "Targeted universal adversarial examples for remote sensing," *Remote Sens.*, vol. 14, no. 22, p. 5833, Nov. 2022.

[37] Y. Deng and L. J. Karam, "Frequency-tuned universal adversarial attacks on texture recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 5856–5868, 2022.

[38] H. Huang, Y. Wang, Z. Chen, Y. Zhang, Y. Li, Z. Tang, W. Chu, J. Chen, W. Lin, and K.-K. Ma, "CMUA-watermark: A cross-model universal adversarial watermark for combating deepfakes," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 989–997.

[39] A. Shapira, A. Zolfi, L. Demetrio, B. Biggio, and A. Shabtai, "Phantom sponges: Exploiting non-maximum suppression to attack deep object detectors," 2022, *arXiv:2205.13618*.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[41] 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.*, Jun. 2009, pp. 248–255.

[42] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[46] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[47] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-axis vision transformer," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, Oct. 2022, pp. 459–479.

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[49] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12009–12019.
[50] V. Papyan, X. Y. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 40, pp. 24652–24663, Oct. 2020.

HUIJIAO WANG received the Ph.D. degree from the Guilin University of Electronic Technology, in 2023. She is currently an Associate Professor and a Master Supervisor with the Guilin University of Electronic Technology. Her research interests include wireless sensor networks, network security, intelligent computing, and Internet of Things technology.

DING CAI received the bachelor's degree in network engineering from Yangtze University, in 2021. He is currently pursuing the master's degree with the Guilin University of Electronic Technology. His research interest includes adversarial examples in computer vision.

LI WANG received the bachelor's degree in engineering from Maanshan University, Maanshan, China, in 2021, and the master's degree in engineering from the Guilin University of Electronic Technology, Guilin, China, in 2021. Her research interests include privacy protection and network security.

ZHUO XIONG received the bachelor's degree in computer science and technology from the Guilin University of Electronic Technology, Guilin, China, in 2021, where he is currently pursuing the master's degree. His research interest includes image adversarial examples and their applications.