# Shape Matters: Deformable Patch Attack

Zhaoyu Chen[1], Bo Li[2†], Shuang Wu[2], Jianghe Xu[2],
Shouhong Ding[2], and Wenqiang Zhang[1,3]

[1] Academy for Engineering and Technology, Fudan University
[2] Youtu Lab, Tencent
[3] Yiwu Research Institute of Fudan University

**Abstract.** Though deep neural networks (DNNs) have demonstrated excellent performance in computer vision, they are susceptible and vulnerable to carefully crafted adversarial examples which can mislead DNNs to incorrect outputs. Patch attack is one of the most threatening forms, which has the potential to threaten the security of real-world systems. Previous work always assumes patches to have fixed shapes, such as circles or rectangles, and it does not consider the shape of patches as a factor in patch attacks. To explore this issue, we propose a novel Deformable Patch Representation (DPR) that can harness the geometric structure of triangles to support the differentiable mapping between contour modeling and masks. Moreover, we introduce a joint optimization algorithm, named Deformable Adversarial Patch (DAPatch), which allows simultaneous and efficient optimization of shape and texture to enhance attack performance. We show that even with a small area, a particular shape can improve attack performance. Therefore, DAPatch achieves state-of-the-art attack performance by deforming shapes on GTSRB and ILSVRC2012 across various network architectures, and the generated patches can be threatening in the real world.

**Keywords:** Adversarial example, patch attack, shape representation

## 1 Introduction

Despite achieving considerably excellent performance on various computer vision tasks [52,53,66,67,68,16,13,14,17,15,26,25,63,46,59,61,4,58], deep neural networks (DNNs) have been shown to be susceptible and vulnerable to adversarial examples, where an adversary introduces an imperceptible perturbation to an image for inducing network misclassification [48]. Currently, adversarial examples have been found in most visual tasks, such as object detection [21,23] and visual tracking [8,36]. Previous attacks and defenses place emphasis on the classic setting of adversarial examples that have a global small $L_p$ distance on the benign example [48,12,38,22,60]. However, the classic $L_p$ setting requires global perturbation to an image, which is not always practical in the physical world.

In this paper, we focus on patch attacks. It is one of the most dangerous forms of adversarial examples that an adversary can arbitrarily modify the pixels of

† indicates the corresponding author (libraboli@tencent.com).

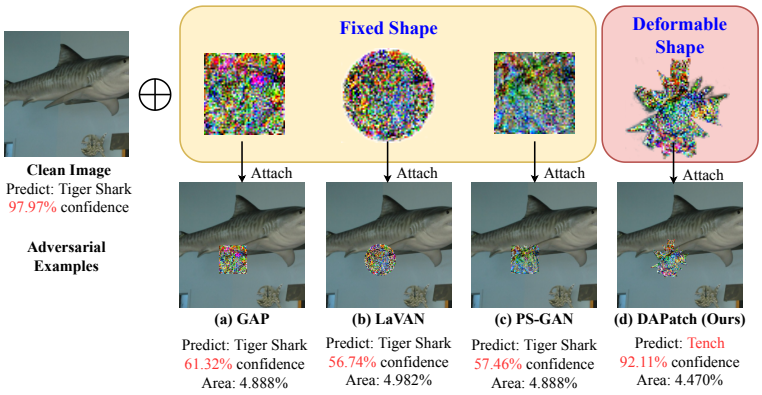



**Fig. 1.** Adversarial patches are generated by different methods under the untargeted attack setting. Previous work always assumes patch shapes to be circles or rectangles but our proposed DAPatch can deform the patches. As a result, DAPatch obtains a higher attack performance.

a continuous region, and the region has to be small enough to reach the victim object in the physical world. For example, GAP [3] creates physical adversarial image patches, which cause the classifiers to ignore the other items in the scene and report a chosen target class. Then Lavan [24] shows that networks can also be fooled by much smaller patches of visible noise that cover a substantially smaller area of the image when relaxing the requirement in the digital domain.

The current patch attacks mainly consider generating robust perturbations and the shape of the patch is usually fixed, such as circles or rectangles. However, both shape and texture are shown to be essential clues for the identification of objects. Geirhos et al. [11] show that ImageNet-trained CNNs are strongly biased towards recognizing textures rather than shapes. Then Li et al. [34] find shape or texture bias has a massive impact on performance and shape-texture debiased learning can improve the accuracy and robustness. However, existing work and other physical attacks [10,8] ignore the importance of shape and assume patches to have fixed shapes, such as circles or rectangles. Specifically, adversarial patches are typically generated using gradients iteratively, and adversarial perturbations within patches could be equivalently regarded as a kind of texture. As previously mentioned, existing studies tend to concentrate on obtaining a robust adversarial texture to fool DNNs, but in this paper, we focus on another perspective of patch attacks, that is shape.

To explicitly explore the effect of shape in patch attacks, the direct approach is to deform the patch in the adversarial attack. Hence, an iterative and differentiable shape representation is required. Existing deform-related work [7,57] needs additional data for training and cannot compute differentiably during the attack patch generation. Rethinking shape modeling, we first need a deformable contour which can be represented by a point and a series of rays in the Cartesian

coordinate system. Then we also need a differentiable calculation procedure to determine whether each position is outside or inside the contour.

To address this issue and explore the effect of shape on patch attacks, we propose a novel Deformable Patch Representation (DPR). The geometric structure of the triangle is used to construct a judgment point whether the point is inside or outside the contour, and the shape model can be mapped into a binary mask while ensuring the computation is differentiable. Then, to achieve a better attack performance, we propose a shape and texture joint optimization for adversarial patches. As illustrated in Figure 1, Deformable Adversarial Patch (DAPatch) improves attack performance by deforming the shape of patches. Extensive experiments on ILSVRC2012 and GTSRB show that, under the same constraint of area, DAPatch have higher attack performance and are effective for various network architectures, such as CNNs [47,19,20,44,49] and Vision Transformer (ViT) [9,37]. Our main contributions are summarized as below:

- We propose a novel Deformable Patch Representation (DPR) that can harness the geometric structure of triangles to support the differentiable mapping between contour modeling and masks, and the shape can be differentiably deformed during patch generations.
- Based on DPR, we propose a shape and texture joint optimization algorithm for adversarial patches, named DAPatch, which can effectively optimize the shape and texture to improve attack performance.
- We show that a particular shape can improve attack performance. Extensive experiments on GTSRB and ILSVRC2012 demonstrate the adversarial threats of shapes with different networks in both digital and physical world.
- DRP first explicitly investigates the significance of shape information on DNNs' robustness through an adversarial lens and contributes to understanding and exploring the very nature of DNNs' vulnerability.

## 2 Related Work

**Adversarial Patch.** The adversarial patch currently can be mainly divided into iterative-based and generative-based methods. For the iterative-based method, GAP [3] proposes adversarial physical image patches, which cause the classifiers to predict a target class. With relaxed requirements in the digital domain, LaVAN [24] shows that networks can also be fooled by much smaller patches of visible noise that cover a much smaller area of the image. For the generative method, PS-GAN [35] refers to the patch generation via a generator as a patch-to-patch translation and simultaneously enhances both the visual fidelity and the attacking ability of the adversarial patch. Other visual tasks are also threatened by patch attacks, such as object segmentation [56,31,28,29,32,30,51,50,65,64], object detection [21,23] and visual tracking [8,36]. The above work can only generate patches of fixed shapes, such as circles or rectangles, without considering the impact of shape on attack performance. The generative-based method requires additional data to train a generator, which requires additional time. Furthermore, the shape of the patch cannot be deformed according to the ad-

versarial attack. In this work, we propose Deformable Patch Representation, which can differentiably deform the patch during the adversarial attack without additional data.

**Defenses Against Patch Attacks.** Several empirical patch defenses are proposed such as Digital Watermark [18] and Local Gradient Smoothing [40]. However, Chiang et al. [6] demonstrate that that these empirical defenses can easily be breached by white-box attacks that take advantage of the pre-processing procedures during the optimization process. Wu et al. [54] and Sukrut et al. [42] adapt adversarial training to increase the robustness of a model against adversarial patches. Chiang et al. [6] propose the first certifiable defense against patch attacks, which gives a certificate when an output lies in the interval bound formed during the training process. Despite this, both robustness approaches require additional training and are inefficient at the ImageNet scale [6,62]. Derandomized Smoothing [27], DPGLC [33], Patchguard [55] and ECViT [5], recently proposed to improve certifiable robustness and to extend the defense to ImageNet, further improve the defense. We select patch defense that can be extended to ImageNet as the benchmark to test the effectiveness of patch attacks. Existing work has demonstrated favorable performance in defending against patch attacks. In this paper, by introducing adversarial shapes, we establish a new baseline to reflect the robustness of the defending methods against adversarial patch attacks from a novel perspective.

**Shape versus Texture.** Object recognition relies on two prominent and complementary cues: shape and texture. The cue that dominates object recognition has been the subject of a long-running debate. Prior to deep learning, object recognition relied on a variety of handcrafted features, such as shape [2] and texture [39]. Recently, Geirhos et al. [11] suggest that CNNs pre-trained on ImageNet exhibit a strong texture bias. Shape-based representations improve object detection and provide previously unknown robustness in the face of a range of image distortions. Furthermore, Li et al. [34] shows the benefits of shape-texture debiased neural network training on boosting both accuracy and robustness. Generating adversarial perturbations within patches could be equivalently regarded as generating a kind of texture. Previously, patch attacks consider patches more for their texture rather than their shape, so there is no deformation method in adversarial attacks. In deformable-related work, Deformable Convolution [7] and contour-based instance segmentation [57] can explicitly model deformations, but they all rely on lots of training data, and can not be differentiably mapped into a mask to participate in the generation of texture. Motivated by this dilemma, we propose the Deformable Adversarial Patch for the shape and texture joint optimization. Note that some work [54,42] has demonstrated that the positions of patches can also affect the threat of an attack, but this paper focuses on investigating the significance of shape information on DNNs' robustness. Therefore, we randomly select and fix the positions of patches to control the effect of positions on attack performance in the experiments.

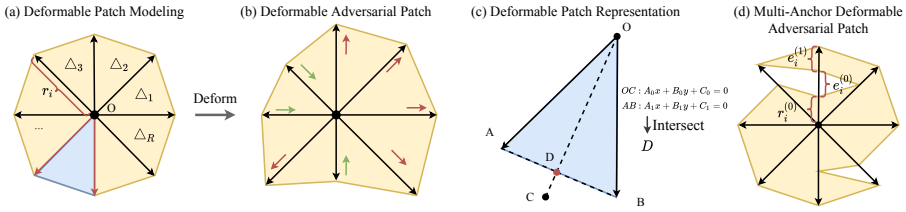



**Fig. 2.** Introduction of Deformable Adversarial Patch. (a) represents the contour modeling composed of one point and rays. (b) represents the deformation from (a) by updating $r$. (c) shows the mask obtained by differentiable calculation in a local triangle. (d) summarizes the multi-anchor mechanism on Deformable Patch Representation.

## 3 Method

In this section, we first introduce the problem of adversarial patch attacks on image classification and then propose our Deformable Patch Representation to deform patches during the patch generation. Then, we propose a joint optimization algorithm for improving the attack performance.

### 3.1 Problem Definition

For a image classifier $f : x \to y$, we denote the clean image as $x \in R^{c\times h\times w}$ and the corresponding label as $y$. In the traditional adversarial patch attack, adversaries attempt to find an adversarial patch $\delta$ to significantly degrade the performance of the classifier over per image. When the adversarial image at the $k$-th iteration is $x^k_{adv} \in R^{c\times h\times w}$, then the solving iteration will be:

$$x^k_{adv} = \delta^{k-1} \odot M + x \odot (I - M), \tag{1}$$

where $\odot$ represents the element-wise Hadmard product; $M \in \{0, 1\}^{c\times h\times w}$ denotes binary masks for $x^k_{adv}$; $I$ represents all-one matrices with the same dimension as $M$.

We denote the prediction result of $x$ by $f$ is $\hat{y}$. For untargeted attacks, the adversarial patch makes the model predict the wrong label, namely $\hat{y} \neq y$. For target attacks, the adversarial patch makes the model predict specified target class $y_t$, namely $\hat{y} = y_t$, and the target class is pre-specified.

### 3.2 Deformable Patch Representation

To model a deformable patch we first need a deformable contour. For simplicity, we use a polygon to represent the contour which consists of one center $O$ and $R$ rays in the Cartesian coordinate system, as shown in Figure 2 (a). Then the contour deforms through the updating of the length of rays $r = \{r_1, r_2, ..., r_R\}$ during attacking. The deformation is shown in Figure 2 (b).

Two rays and a center form a triangle and the whole patch mask can be divided into $R$ triangles, with the angle interval $\Delta\theta = 2\pi/R$. As shown in Figure 2 (c), for $\triangle AOB$, we define $|AO| = r_A$, $|BO| = r_B$. Therefore, for $\forall C \in x$, the mask $M$ is expressed as:

$$M(C) = \begin{cases} 1, & C \in \triangle AOB \\ 0, & C \notin \triangle AOB. \end{cases} \tag{2}$$

Therefore, we convert the contour representation into the question of whether the point is inside or outside the contour. Next, we use the geometric properties of triangles to differentiably calculate and obtain a deformable mask. For any $C$ falling in the area covered by $\angle AOB$, there will always be $CO$ or the extended line of $CO$ intersects $AB$ at $D$. Note that Equation 2 needs to be converted into computable, so we use $\frac{|CO|}{|DO|}$ to judge whether $C$ is inside the $\triangle AOB$. Obviously, if $\frac{|CO|}{|DO|} < 1$, then $C \in \triangle AOB$ and vice versa. Since $\frac{|CO|}{|DO|} \in R^+$, we want $M$ to be approximately binary and mapped to $\{0, 1\}$. To address the issue, we choose a special activation function $\Phi$, which is expressed as:

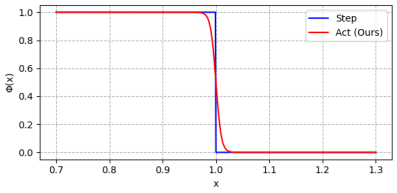



**Fig. 3.** The effectiveness of the activation function $\Phi(x)$. Step function represents the same effect as Equation 3 and our activation function $\Phi(x)$ well approximates the step function.

$$\Phi(x) = \frac{\tanh\left(\lambda(x-1)\right) + 1}{2}. \tag{3}$$

Here, $\lambda$ controls the sparsity of activation function and we take $\lambda = -100$. Figure 3 reflects the effectiveness of the activation function $\Phi(x)$. In the Cartesian coordinate system, the coordinates of $A$ and $B$ can be calculated from the ray length $r$ and angle intervals so $D$ can be solved by gaussian eliminations via $A, B$ and $O$. So Equation 2 can be rephrased as:

$$M(C, r) = \Phi(\frac{|CO|}{|DO|}) \in \{0, 1\}. \tag{4}$$

By focusing on the global mask $M$, we pre-calculate where $\triangle_i$ belongs based on the $\angle COx$. Using the ray length $r$ and angle interval $\Delta\theta$, we can directly calculate the coordinates of the ray endpoint $P = \{P^1, P^2, ..., P^R\}$ via triangular properties in the Cartesian coordinate system. For $\forall C \in \triangle_i$, we solve the linear equations of $AB$ and $CO$, calculate the coordinates of $D$, and determine the corresponding mask value $M(C, r)$ according to Equation 4. Based on parallel computing, the time complexity of calculating the global mask is $O(R)$ and the space complexity is $O(hw)$.

The proposed modeling strategy can be easily extended to more complex contours, as shown in Figure 2 (d). This situation mainly occurs when the ray

**Algorithm 1** Deformable Patch Representation (DPR)

**Input**: the center $O$, the number of rays $R$, ray length array $r = \{r_1, r_2, ..., r_n\}$
**Output**: the mask $M$

$\Delta\theta \leftarrow 2 * \pi / R$
Calculate each pixel $C$ belongs to $\triangle_i$
Calculate the coordinates of the ray endpoint $P$ by $\Delta\theta$
**for** $i \in [1,\ R]$ **do**
    Select the point set $C$ in the $\triangle_i$
    $A_0, B_0, C_0 \leftarrow \text{Gaussian\_Elimination}(P^i, P^{i+1})$
    $A_1, B_1, C_1 \leftarrow \text{Gaussian\_Elimination}(O, C)$
    $d \leftarrow A_0 * B_1 - A_1 * B_0$
    $D_x \leftarrow (B_0 * C_1 - B_1 * C_0)/d$
    $D_y \leftarrow (A_1 * C_0 - A_0 * C_1)/d$
    $M_{(x,y)\in p} = \Phi\left(\frac{|CO|}{|DO|}\right)$
**end for**
$M_O \leftarrow 1$
**return** $M$

passes through the contour multiple times. Specifically, in order to enhance the modeling ability and achieve more complex contour modeling, we introduce a multi-anchor mechanism:

$$r^{(0)} = \{r_1^{(0)}, r_2^{(0)}, ..., r_R^{(0)}\}, \tag{5}$$

$$e^{(i)} = \{e_1^{(i)}, e_2^{(i)}, ..., e_R^{(i)}\}, \quad i = 0, 1, ..., R-1, \tag{6}$$

$$r^{(i+1)} = r^{(i)} + e^{(i)}, \quad i = 0, 1, ..., R-1, \tag{7}$$

where $r^{(i)}$ represents the length of the ray in the $i$-th anchor and $e^{(i)}$ denotes the margin between $r^{(i)}$ and $r^{(i+1)}$. In practice, Deformable Patch Representation with a single anchor can obtain promising attack performance. Due to the space limitation of the paper, we mainly elaborate the single anchor strategy in this work, and the specific implementation is illustrated in Algorithm 1.

### 3.3 Deformable Adversarial Patch

Although Deformable Patch Representation provides a deformation modeling, generating adversarial patches with better attack performance is still a challenging issue. In this section, we propose our Deformable Adversarial Patch by the joint optimization of shape and texture.

Area denotes the percentage of pixels of the patch relative to the image and deformation affects the area of the patch. Obviously, the larger the area, the stronger its attack performance. In order to explicitly control the area of the patch and facilitate the joint optimization of shape and texture, the loss function $L$ can be written as:

$$L = \begin{cases} L_{adv}, & area \leq ps \\ L_{adv} + \beta \cdot L_{shape}, & area > ps \end{cases}, \tag{8}$$

where $area$ is the area of the deformable patch; $ps$ is defined as the upper limit of the patch area; $\beta$ is the hyper-parameter to limit the margin of $area$ and $ps$. $L_{adv}$ is the cross-entropy loss. In order to explicitly punish patches with too large areas, we average the mask $M$ and $L_{shape}$ is defined as:

$$L_{shape} = \text{mean}(M^k). \tag{9}$$

Suppose the deformable mask as $M^k \in R^{c \times h \times w}$ at the $k$-th iteration, Equation 1 can be re-expressed as:

$$x_{adv}^k = \delta^{k-1} \odot M^{k-1} + x_{adv}^{k-1} \odot (I - M^{k-1}). \tag{10}$$

In Equation 4, the generation of the global mask $M$ is controlled by $r$. Here, $\delta$ represents the update of texture and $r$ represents the update of shape. Based on gradient $\nabla L$, the updating process can be regarded as:

$$\delta^k \leftarrow \delta^{k-1} + \alpha \cdot \text{sign}(\nabla_{x_{adv}^k} L), \quad r^k \leftarrow r^{k-1} + \gamma \cdot \text{sign}(\nabla_{r^{k-1}} L). \tag{11}$$

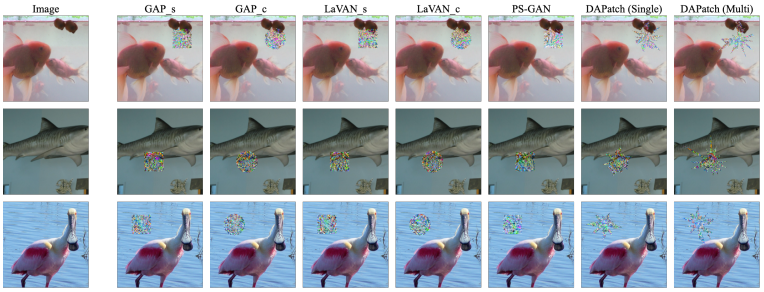
In Equation 3, we want $M$ to be approximately binary. Although differentiable computation can be realized in this way, $M$ is only close to binarization in numerical. To solve this problem, we introduce the shape ratio $s$ (%) for perturbation tuning. Specifically, when the joint optimization of shape and texture reaches the ratio $s$, we sharpen the mask. The fine-tuning texture is then adapted to the sharpened mask for improving attack performance. For simplicity, we choose the binarization for sharpening. Appendix 2 summarizes the algorithm of our proposed Deformable Adversarial Patch.

## 4 Experiments

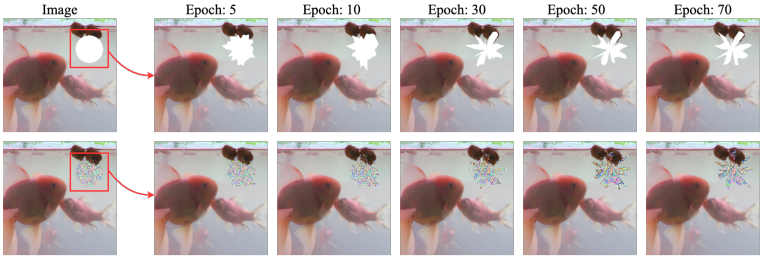
In this section, we evaluate our proposed DAPatch in the classification task. Firstly, we analyze the significance of shape information on DNNs' robustness through an adversarial lens. Second, we evaluate the effectiveness of the proposed DAPatch in the digital domain. Next, we verify the performance of patch attacks under patch defenses. Then, we generate patches and achieve physical attacks in the real world. Finally, we conduct ablation study for hyper-parameters in Appendix 6.

### 4.1 Experimental Setup

In our experiments, we use Pytorch [41] for the implementation and test on NVIDIA Tesla V100 GPUs. The proposed DAPatch compares with state-of-the-art methods, such as GAP [3], LaVAN [24] and PS-GAN [35]. Following the setting of previous work [3,24,35], we evaluate on two datasets, including German Traffic Sign Recognition Benchmark (GTSRB) and Imagenet Large Scale Visual Recognition Challenge (ILSVRC2012) [43]. We randomly select 1000 images from the ILSVRC2012 validation set and 500 images from the GTSRB test set. To




(a) Visualizations of different adversarial patches in untargeted attacks.




(b) Deformation process of DAPatch in the adversarial attack.

**Fig. 4.** Visualizations of patch attacks. Disabling Texture is the top of (b) and the deformation process is the bottom of (b). More details are shown in Appendix 7.

provide a fair platform for comparison, all the experimental results are reported under the white-box adversarial attack. Since attackers can design the white-box or adaptive attack according to models easily, the white-box setting can better reflect the robustness of models against patch attacks.

We explore the effectiveness of our proposed DAPatch on various model architectures. We divide the model architectures into three categories: **CNN** (VGG19 [47], Resnet-152 [19], DenseNet-161 [20] and MobileNet V2 [44]), **ViT** (ViT-B/16 [9] and Swin-B [37]), and **NAS** (EfficientNet-b7 [49]). To study the impact of different shapes on attack performance, we give patches different initial shapes. GAP_s and GAP_c represent the square and circular patch, and so does LaVAN. The initial shape of the DAPatch is the same as the circular patch. Here $s$ is 70 and $\beta$ is 200. For more details, please see the Appendix 3 and 6.

Attack Success Rate (ASR) is a quantitative metric in the attack performance. Here, we define ASR as the classification error rate. For untargeted attacks, if the predicted label $\hat{y}$ is inconsistent with the ground truth $y$, the attack is considered successful. For targeted attacks, we choose the most difficult setting to evaluate the attack performance. Specifically, we set the class with the smallest one in logits as the target class $y_t$. Only when $\hat{y} = y_t$, the attack is successful. Here, we do not consider the impact of locations on patches. All patches are randomly initialized at fixed locations for attacking 100 iterations under different areas. We select different patch areas and choose the size of squares and circles

**Table 1.** A specific shape can improve ASR even if the area is small.

| Network | Shape | ≈0.5% | | ≈1% | | ≈2% | | ≈3% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| MoblieNet v2 | Circle | 1.5 | 0.510 | 2.2 | 0.964 | 4.5 | 2.040 | 6.8 | 3.031 |
| | Square | 1.4 | 0.504 | 1.7 | 1.054 | 3.3 | 2.010 | 4.4 | 3.023 |
| | Ours | **8.9** | **0.377** | **13.4** | **0.790** | **21.0** | **1.648** | **25.8** | **2.496** |
| Vit-B/16-224 | Circle | 0.9 | 0.510 | 1.4 | 0.964 | 2.2 | 2.040 | 2.2 | 3.031 |
| | Square | 0.5 | 0.504 | 0.7 | 1.054 | 1.1 | 2.010 | 1.6 | 3.023 |
| | Ours | **8.6** | **0.355** | **12.0** | **0.789** | **16.3** | **1.563** | **20.7** | **2.507** |
| ResNet-152 | Circle | 0.9 | 0.510 | 1.2 | 0.964 | 2.6 | 2.040 | 3.3 | 3.031 |
| | Square | 0.5 | 0.504 | 0.6 | 1.054 | 0.8 | 2.010 | 1.3 | 3.023 |
| | Ours | **5.8** | **0.371** | **10.3** | **0.776** | **18.4** | **1.618** | **23.6** | **2.449** |

**Table 2.** Experiments on Multi-anchor DAPatch. Complex modeling can improve the attack performance in the same area.

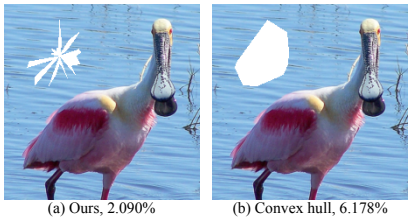
| Network | Method | ≈0.5% | | ≈1% | | ≈2% | | ≈3% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| MoblieNet v2 | Single | 65.8 | 0.423 | 88.9 | 0.847 | 97.6 | 1.735 | 99.4 | 2.684 |
| | Multi | 67.8 | 0.425 | 89.3 | 0.851 | 98.9 | 1.734 | 99.6 | 2.667 |
| Vit-B/16-224 | Single | 56.9 | 0.417 | 80.9 | 0.849 | 95.0 | 1.717 | 98.3 | 2.676 |
| | Multi | 57.0 | 0.434 | 80.2 | 0.855 | 97.2 | 1.723 | 99.2 | 2.682 |
| ResNet-152 | Single | 52.2 | 0.409 | 78.8 | 0.845 | 93.1 | 1.699 | 97.9 | 2.623 |
| | Multi | 53.2 | 0.421 | 82.3 | 0.832 | 94.5 | 1.711 | 99.4 | 2.636 |

approximately close to the area. Area (%) denotes the average area percentage of patches over the successfully attacked images. The experiments are under the condition of a constrained area.

### 4.2 Delving into shape and texture

Perturbations in patch attacks can be regarded as a special texture. To evaluate the significance of shape information on DNNs' robustness through an adversarial lens, we remove the texture and fix it to white. The patch is then deformed to study only the effect of shape, as illustrated in Figure 4 (b). As described in Table 1, placing a patch of the fixed shape on a white texture only slightly reduces accuracy. Further, we exploit the convex hull formed by deformable shapes to attack, as shown in Figure 5. The area of the convex hull is often larger than the deformable shape, and the attack performance is not as good as the deformable shape shape. For example, at 3% area, ResNet-152 produces the DAPatch with an area of 2.449% and an ASR of 23.6%, but its convex hull has a larger area (5.745%) but only a lower ASR (5.0%). In DAPatch, the deformation of shape can significantly improve the attack performance, which shows that *a particular shape can improve ASR*. Please refer to Appendix 4 for more details.

The textures of the patch play a significant role in magnifying the performance of patch attack. Even if the network has a bias against texture, the shape can improve attack performance. According to the relationship between shape and texture in object recognition, the shape-biased network can be more vulnerable to the DAPatch. Furthermore, models that make predictions largely based on the shape and texture of objects in images rather than only on shape or texture can be more adversarially robust. We consider the Shape-Network [11] as the most sensitive network to shape against DAPatch



(a) Ours, 2.090% (b) Convex hull, 6.178%

**Fig. 5.** The area of the convex hull is larger than DAPtach, but the attack performance is not as good as it, which shows that having a specific shape can improve the attack performance.

and Shape-Texture Debiased Network [34] is currently the best potential defense against DAPatch. The Shape-Network is supposed to be insensitive to texture, but more sensitive on shape for making predictions. ResNet50-SIN, ResNet50-SIN+IN, and ResNet50-SIN+IN-IN are proposed by Shape-Network and achieve ASR as 20.3%, 3.5% and 2.1% on randomly 1000 images before attacking. ResNet50-Debiased and ResNet152-Debiased are proposed by Shape-Texture Debiased Network and achieve ASR as 2.3% and 2.1% before attacking. The experimental results on ILSVRC2012 in Table 3 show that DAPatch easily confuses the Shape-Network and the Shape-Texture Debiased Network with basically no difference as against a normal deep network. In the untargeted attack, ResNet50-SIN increases 54.8% on ASR under 0.5% patch percentage and is much more than ResNet50-Debiased, which increases 44.3%. The same situation happens in the 1% patch area. In a larger area, the margin is not obvious because the ASR is relatively high.

We further use multi-anchor to study the effect of complex shape modeling. Here, the number of anchors is 3. The experiments show that *complex modeling can improve the attack performance in the same area, which also imlpies that the shape is important to the robustness of DNNs*, as shown in Table 2. In Figure 4, we show the visualizations of DAPatch and other patches.

DAPatch is not limited to one particular type of attack. DAPatch not only evaluates the robustness of existing classification models, but also shows that a particular shape can improve the attack performance. All in all, shape information has a great impact on the robustness of DNNs, which can be seminal in understanding and exploring the very nature of DNNs' vulnerability. More details are in Appendix 4 and 7.

### 4.3 Digital Attacks

In this section, we evaluate the effectiveness of DAPatch in the digital domain. Before performing the attacks, the ASR (%) of the models is 0%. The experimental results in untargeted setting on ILSVRC2012 and GTSRB are summarized in Table 5 and Table 6. *Note that when the patch area is small, DAPatch always obtains a higher ASR with a smaller area. Experiments show that under different patch areas, DAPatch can always obtain better attack effects with a smaller area.* For the more challenging targeted setting, the experimental results on ILSVRC2012 are reported in Table 7. We choose the most difficult setting and the target class is the class with the smallest one in logits. According to Table 7, DAPatch also achieves stronger attack performance under different areas.

Moreover, we test them using a traditional state-of-the-art defense method known as Adversarial Training. Fast-AT trains with the Fast Gradient Sign Method (FGSM) [12], when combined with the random initialization, is as effective as training based on Projected Gradient Descent (PGD) [38] but has significantly lower cost. Feature Denoising is the state-of-the-art defense against adversarial attacks in the white-box $L_p$ setting and PGD only decreases the accuracy to 55.7% and 45.5% after 10 and 100 iterations. Fast-AT, Adv-ResNet-152, ResNet-152-Denoise and Resnext-101-Deniose obtain ASR with 33.4%, 36.8%,

**Table 3.** Untargeted attacks on shape and texture bias.

| Network | Method | ≈0.5% | | ≈1% | | ≈2% | | ≈3% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| ResNet50 -SIN | GAP_s | 70.4 | 0.510 | 87.3 | 0.964 | 96.9 | 2.040 | 99.1 | 3.031 |
| | GAP_c | 70.1 | 0.504 | 88.5 | 1.054 | 96.9 | 2.010 | 99.6 | 3.023 |
| | LaVAN_s | 66.2 | 0.510 | 82.2 | 0.964 | 95.1 | 2.040 | 98.1 | 3.031 |
| | LaVAN_c | 65.6 | 0.504 | 84.2 | 1.054 | 96.0 | 2.010 | 98.7 | 3.023 |
| | PS-GAN | 70.0 | 0.510 | 85.3 | 0.964 | 96.8 | 2.040 | 99.5 | 3.031 |
| | Ours | **74.1** | **0.446** | **90.3** | **0.893** | **98.7** | **1.764** | **99.6** | **2.724** |
| ResNet50 -SIN+IN | GAP_s | 44.3 | 0.510 | 68.3 | 0.964 | 90.6 | 2.040 | 95.7 | 3.031 |
| | GAP_c | 44.7 | 0.504 | 72.4 | 1.054 | 90.9 | 2.010 | 96.3 | 3.023 |
| | LaVAN_s | 41.2 | 0.510 | 64.9 | 0.964 | 88.8 | 2.040 | 94.6 | 3.031 |
| | LaVAN_c | 42.4 | 0.504 | 69.4 | 1.054 | 89.4 | 2.010 | 96.1 | 3.023 |
| | PS-GAN | 44.5 | 0.510 | 68.9 | 0.964 | 90.6 | 2.040 | 95.2 | 3.031 |
| | Ours | **48.1** | **0.426** | **75.6** | **0.860** | **91.5** | **1.750** | **96.3** | **2.669** |
| ResNet50 -SIN+IN-IN | GAP_s | 41.6 | 0.510 | 62.2 | 0.964 | 85.6 | 2.040 | 93.0 | 3.031 |
| | GAP_c | 44.3 | 0.504 | 68.6 | 1.054 | 87.4 | 2.010 | 94.2 | 3.023 |
| | LaVAN_s | 38.7 | 0.510 | 58.2 | 0.964 | 83.1 | 2.040 | 92.8 | 3.031 |
| | LaVAN_c | 39.6 | 0.504 | 65.3 | 1.054 | 84.7 | 2.010 | 93.4 | 3.023 |
| | PS-GAN | 39.2 | 0.510 | 62.7 | 0.964 | 85.2 | 2.040 | 94.7 | 3.031 |
| | Ours | **44.3** | **0.420** | **70.1** | **0.845** | **88.9** | **1.729** | **95.1** | **2.651** |
| ResNet50 -Debiased | GAP_s | 42.5 | 0.510 | 64.2 | 0.964 | 85.8 | 2.040 | 93.1 | 3.031 |
| | GAP_c | 44.0 | 0.504 | 68.6 | 1.054 | 87.9 | 2.010 | 94.3 | 3.023 |
| | LaVAN_s | 38.2 | 0.510 | 59.7 | 0.964 | 83.3 | 2.040 | 90.9 | 3.031 |
| | LaVAN_c | 38.8 | 0.504 | 64.2 | 1.054 | 84.3 | 2.010 | 93.2 | 3.023 |
| | PS-GAN | 43.2 | 0.510 | 67.6 | 0.964 | 88.0 | 2.040 | 93.4 | 3.031 |
| | Ours | **46.6** | **0.438** | **68.6** | **0.852** | **88.1** | **1.744** | **94.7** | **2.665** |
| ResNet152 -Debiased | GAP_s | 33.3 | 0.510 | 53.0 | 0.964 | 81.4 | 2.040 | 91.0 | 3.031 |
| | GAP_c | 34.1 | 0.504 | 58.9 | 1.054 | 84.1 | 2.010 | 93.0 | 3.023 |
| | LaVAN_s | 30.6 | 0.510 | 49.8 | 0.964 | 77.0 | 2.040 | 88.8 | 3.031 |
| | LaVAN_c | 29.8 | 0.504 | 52.1 | 1.054 | 77.5 | 2.010 | 90.1 | 3.023 |
| | PS-GAN | 32.3 | 0.510 | 53.4 | 0.964 | 83.0 | 2.040 | 93.2 | 3.031 |
| | Ours | **37.4** | **0.422** | **61.8** | **0.850** | **85.3** | **1.735** | **94.5** | **2.626** |

**Table 4.** Untargeted attacks on networks with adversarial training. DAPatch can always obtain better attack effects on AT networks with a smaller area.

| Network | Method | ≈0.5% | | ≈1% | | ≈2% | | ≈3% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| Adv- ResNet-152 | GAP_s | 61.0 | 0.510 | 74.5 | 0.964 | 86.6 | 2.040 | 90.3 | 3.031 |
| | GAP_c | 60.6 | 0.504 | 77.4 | 1.054 | 87.2 | 2.010 | 91.7 | 3.023 |
| | LaVAN_s | 58.4 | 0.510 | 71.1 | 0.964 | 83.9 | 2.040 | 88.8 | 3.031 |
| | LaVAN_c | 57.2 | 0.504 | 72.6 | 1.054 | 83.9 | 2.010 | 89.3 | 3.023 |
| | PS-GAN | 59.2 | 0.510 | 77.7 | 0.964 | 84.3 | 2.040 | 89.7 | 3.031 |
| | Ours | **62.5** | **0.472** | **78.4** | **0.948** | **88.2** | **1.921** | **92.4** | **2.791** |
| ResNet-152 -Denoise | GAP_s | 59.3 | 0.510 | 74.5 | 0.964 | 86.5 | 2.040 | 92.6 | 3.031 |
| | GAP_c | 59.0 | 0.504 | 77.3 | 1.054 | 87.8 | 2.010 | 92.9 | 3.023 |
| | LaVAN_s | 59.6 | 0.510 | 72.6 | 0.964 | 84.7 | 2.040 | 91.8 | 3.031 |
| | LaVAN_c | 60.7 | 0.504 | 75.1 | 1.054 | 85.5 | 2.010 | 92.7 | 3.023 |
| | PS-GAN | 61.7 | 0.510 | 75.1 | 0.964 | 86.2 | 2.040 | 92.2 | 3.031 |
| | Ours | **62.3** | **0.464** | **77.4** | **0.959** | **88.0** | **1.835** | **92.9** | **2.853** |
| Resnext-101 -Denoise | GAP_s | 50.4 | 0.510 | 66.3 | 0.964 | 83.8 | 2.040 | 90.2 | 3.031 |
| | GAP_c | 51.1 | 0.504 | 70.5 | 1.054 | 84.2 | 2.010 | 89.9 | 3.023 |
| | LaVAN_s | 49.7 | 0.510 | 65.0 | 0.964 | 80.6 | 2.040 | 87.6 | 3.031 |
| | LaVAN_c | 49.5 | 0.504 | 67.9 | 1.054 | 81.2 | 2.010 | 88.4 | 3.023 |
| | PS-GAN | 51.2 | 0.510 | 68.1 | 0.964 | 80.9 | 2.040 | 89.9 | 3.031 |
| | Ours | **52.9** | **0.471** | **68.9** | **0.949** | **85.5** | **1.928** | **90.2** | **2.814** |
| Fast_AT | GAP_s | 50.4 | 0.510 | 62.3 | 0.964 | 80.4 | 2.040 | 88.5 | 3.031 |
| | GAP_c | 50.6 | 0.504 | 65.5 | 1.054 | 80.3 | 2.010 | 88.8 | 3.023 |
| | LaVAN_s | 48.7 | 0.510 | 60.2 | 0.964 | 77.5 | 2.040 | 84.7 | 3.031 |
| | LaVAN_c | 48.7 | 0.504 | 62.6 | 1.054 | 78.0 | 2.010 | 85.3 | 3.023 |
| | PS-GAN | 48.9 | 0.510 | 63.4 | 0.964 | 79.1 | 2.040 | 85.2 | 3.031 |
| | Ours | **51.3** | **0.473** | **65.6** | **0.944** | **82.0** | **1.890** | **90.0** | **2.963** |

**Table 5.** Untargeted attacks of various network architectures on ILSVRC2012.

| Model | Method | GAP_s | | GAP_c | | LaVan_s | | LaVan_c | | PS-GAN | | Ours | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| ResNet-152 | 0.5% | 44.3 | 0.510 | 44.8 | 0.504 | 43.7 | 0.510 | 43.5 | 0.504 | 44.5 | 0.510 | **52.2** | **0.409** |
| | 1% | 71.0 | 0.964 | 74.4 | 1.054 | 67.5 | 0.964 | 71.2 | 1.054 | 68.9 | 0.964 | **78.8** | **0.845** |
| | 2% | 89.5 | 2.040 | 91.2 | 2.010 | 88.3 | 2.040 | 90.4 | 2.010 | 91.3 | 2.040 | **93.1** | **1.699** |
| | 3% | 96.5 | 3.031 | 97.8 | 3.023 | 95.9 | 3.031 | 96.8 | 3.023 | 97.4 | 3.031 | **97.9** | **2.623** |
| Efficientnet-b7 | 0.5% | 43.3 | 0.510 | 42.5 | 0.504 | 43.3 | 0.510 | 41.2 | 0.504 | 40.9 | 0.510 | **45.7** | **0.442** |
| | 1% | 63.5 | 0.964 | 68.8 | 1.054 | 64.7 | 0.964 | 69.2 | 1.054 | 65.8 | 0.964 | **71.1** | **0.956** |
| | 2% | 85.5 | 2.040 | 88.0 | 2.010 | 89.5 | 2.040 | 89.2 | 2.010 | 89.3 | 2.040 | **89.6** | **2.003** |
| | 3% | 91.5 | 3.031 | 94.4 | 3.023 | 95.9 | 3.031 | 95.9 | 3.023 | 95.2 | 3.031 | **95.9** | **3.014** |
| Vit-B/16-224 | 0.5% | 47.0 | 0.510 | 45.6 | 0.504 | 47.0 | 0.510 | 46.9 | 0.504 | 45.9 | 0.510 | **56.9** | **0.417** |
| | 1% | 72.0 | 0.964 | 77.2 | 1.054 | 71.8 | 0.964 | 74.9 | 1.054 | 71.9 | 0.964 | **80.9** | **0.849** |
| | 2% | 92.4 | 2.040 | 93.0 | 2.010 | 93.5 | 2.040 | 93.5 | 2.010 | 90.2 | 2.040 | **95.0** | **1.717** |
| | 3% | 97.2 | 3.031 | 97.8 | 3.023 | 98.3 | 3.031 | 98.3 | 3.023 | 97.4 | 3.031 | **98.3** | **2.676** |

30.1% and 20.5% respectively before attacking. Our results of untargeted attacks are summarized in Table 4. *Note that compared with the baselines, DAPatch can always obtain better attack effects on AT networks with a smaller area.*

### 4.4 Attack against Patch Defenses

We select patch defenses that can be extended to ILSVRC2012 as the benchmark to test the effectiveness of patch attacks, including Local Gradient Smoothing (LGS) [40], Digital Watermarking (DW) [18], PatchGuard [55] and Derandomized Smoothing (DS) [27]. For empirical defenses, LGS is regarded as a differen-

**Table 6.** Untargeted attacks of various network architectures on GTSRB.

| Network | Method | ≈0.5% | | ≈1% | | 2% | | ≈3% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| ResNet-152 | GAP_s | 13.6 | 0.510 | 21.4 | 0.964 | 37.8 | 2.040 | 56.0 | 3.031 |
| | GAP_c | 13.0 | 0.504 | 22.6 | 1.054 | 38.0 | 2.010 | 57.6 | 3.023 |
| | LaVAN_s | 14.6 | 0.510 | 22.8 | 0.964 | 41.6 | 2.040 | 60.4 | 3.031 |
| | LaVAN_c | 14.8 | 0.504 | 26.0 | 1.054 | 41.8 | 2.010 | 58.4 | 3.023 |
| | PS-GAN | 13.7 | 0.510 | 23.4 | 0.964 | 39.4 | 2.040 | 59.5 | 3.031 |
| | Ours | **15.0** | **0.477** | **27.1** | **0.831** | **42.3** | **1.932** | **61.5** | **2.873** |
| Efficientnet -b7 | GAP_s | 20.4 | 0.510 | 45.0 | 0.964 | 68.0 | 2.040 | 84.0 | 3.031 |
| | GAP_c | 20.6 | 0.504 | 39.4 | 1.054 | 66.6 | 2.010 | 82.6 | 3.023 |
| | LaVAN_s | 22.2 | 0.510 | 46.2 | 0.964 | 74.0 | 2.040 | 89.2 | 3.031 |
| | LaVAN_c | 22.8 | 0.504 | 51.4 | 1.054 | 74.0 | 2.010 | 89.0 | 3.023 |
| | PS-GAN | 21.5 | 0.510 | 46.2 | 0.964 | 71.2 | 2.040 | 85.6 | 3.031 |
| | Ours | **23.6** | **0.469** | **53.1** | **0.893** | **75.2** | **1.873** | **89.5** | **2.934** |
| Vit-B /16-224 | GAP_s | 28.6 | 0.510 | 61.2 | 0.964 | 90.0 | 2.040 | 97.6 | 3.031 |
| | GAP_c | 28.4 | 0.504 | 68.0 | 1.054 | 90.2 | 2.010 | 98.2 | 3.023 |
| | LaVAN_s | 28.2 | 0.510 | 61.6 | 0.964 | 91.8 | 2.040 | 98.2 | 3.031 |
| | LaVAN_c | 26.2 | 0.504 | 65.4 | 1.054 | 92.0 | 2.010 | 97.4 | 3.023 |
| | PS-GAN | 27.4 | 0.510 | 64.2 | 0.964 | 90.2 | 2.040 | 98.1 | 3.031 |
| | Ours | **30.1** | **0.483** | **68.1** | **0.896** | **93.5** | **1.783** | **98.9** | **2.892** |

**Table 7.** Targeted attacks of various network architectures on ILSVRC2012.

| Network | Method | ≈1% | | ≈3% | | ≈5% | | ≈7% | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| ResNet-152 | GAP_s | 5.30 | 0.964 | 41.1 | 3.031 | 68.8 | 4.982 | 87.3 | 6.938 |
| | GAP_c | 8.80 | 1.054 | 44.3 | 3.023 | 72.8 | 4.888 | 88.3 | 6.794 |
| | LaVAN_s | 3.50 | 0.964 | 22.6 | 3.031 | 45.1 | 4.982 | 61.8 | 6.938 |
| | LaVAN_c | 6.00 | 1.054 | 23.9 | 3.023 | 47.7 | 4.888 | 64.7 | 6.794 |
| | PS-GAN | 8.90 | 0.964 | 46.2 | 3.031 | 73.8 | 4.982 | 87.4 | 6.938 |
| | Ours | **9.10** | **0.849** | **48.7** | **2.668** | **78.1** | **4.553** | **90.2** | **6.439** |
| Efficientnet -b7 | GAP_s | 4.40 | 0.964 | 52.1 | 3.031 | 81.9 | 4.982 | 93.7 | 6.938 |
| | GAP_c | 6.20 | 1.054 | 53.6 | 3.023 | 81.4 | 4.888 | 93.4 | 6.794 |
| | LaVAN_s | 1.50 | 0.964 | 33.1 | 3.031 | 65.0 | 4.982 | 82.2 | 6.938 |
| | LaVAN_c | 2.30 | 1.054 | 34.0 | 3.023 | 62.1 | 4.888 | 83.6 | 6.794 |
| | PS-GAN | 4.90 | 0.964 | 53.6 | 3.031 | 81.5 | 4.982 | 93.2 | 6.938 |
| | Ours | **7.60** | **0.869** | **53.6** | **2.953** | **82.0** | **4.851** | **93.7** | **6.713** |
| Vit-B /16-224 | GAP_s | 6.20 | 0.964 | 48.8 | 3.031 | 85.4 | 4.982 | 97.3 | 6.938 |
| | GAP_c | 7.90 | 1.054 | 50.6 | 3.023 | 85.7 | 4.888 | 97.2 | 6.794 |
| | LaVAN_s | 3.10 | 0.964 | 25.4 | 3.031 | 52.8 | 4.982 | 78.3 | 6.938 |
| | LaVAN_c | 4.20 | 1.054 | 24.7 | 3.023 | 54.9 | 4.888 | 78.4 | 6.794 |
| | PS-GAN | 5.30 | 0.964 | 49.3 | 3.031 | 85.8 | 4.982 | 97.1 | 6.938 |
| | Ours | **9.60** | **0.850** | **50.7** | **2.697** | **86.2** | **4.688** | **97.4** | **6.727** |

**Table 8.** Patch attacks on patch defenses. A greater ASR means better.

| Type | Model | Method | Clean | GAP_s | | GAP_c | | LaVan_s | | LaVan_c | | PS-GAN | | Ours | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ASR | ASR | Area | ASR | Area | ASR | Area | ASR | Area | ASR | Area | ASR | Area |
| Empirical | ResNet-152 | Non-Defense | 0 | 89.5 | 2.040 | 91.2 | 2.010 | 88.3 | 2.040 | 90.4 | 2.010 | 91.3 | 2.040 | **93.1** | **1.699** |
| | | LGS | 3.7 | 51.0 | 2.040 | 51.4 | 2.010 | 47.3 | 2.040 | 47.3 | 2.010 | 44.8 | 2.040 | **53.9** | **2.003** |
| | | DW | 10.2 | 69.4 | 2.040 | 68.5 | 2.010 | 64.3 | 2.040 | 67.5 | 2.010 | 57.6 | 2.040 | **69.5** | **1.862** |
| | ViT-B | Non-Defense | 0 | 92.4 | 2.040 | 93.0 | 2.010 | 93.5 | 2.040 | 93.5 | 2.010 | 90.2 | 2.040 | **95.0** | **1.717** |
| | | LGS | 3.0 | 56.6 | 2.040 | 57.5 | 2.010 | 54.7 | 2.040 | 56.8 | 2.010 | 53.4 | 2.040 | **58.1** | **1.891** |
| | | DW | 10.3 | 68.0 | 2.040 | 69.5 | 2.010 | 66.4 | 2.040 | 69.3 | 2.010 | 60.1 | 2.040 | **69.8** | **2.001** |
| Certifiable | BagNet-17 | PatchGuard | 24.8 | 31.2 | 2.040 | 31.3 | 2.010 | 30.8 | 2.040 | 30.9 | 2.010 | 26.7 | 2.040 | **32.7** | **1.998** |
| | ResNet-50 | DS | 7.8 | 13.3 | 2.040 | 13.3 | 2.010 | 13.1 | 2.040 | 12.9 | 2.010 | 10.2 | 2.040 | **13.4** | **2.001** |

tiable pre-processing process to generate patches, and DW is added to Backward Pass Differential Approximation (BPDA) [1] to ignore the operator in backward propagation approximate gradient. For certifiable defenses, the patches are generated to attack the modified CNN model using PatchGuard and DS, which changes the forward propagation function of the CNN model.

Table 8 shows patch attacks under 2% area on patch defenses. With the help of DPR, DAPatch achieves better attack performance under all patch defense methods. We establish a new baseline to reflect the robustness of defending methods against patch attacks from the perspective of adversarial shapes.

### 4.5 Physical Attacks

Physical attacks are conducted to verify the effectiveness of DAPatch in real-world scenarios. We take 10 common classes from ILSVRC2012 [43] and 50 images in total are taken at five different placements (5 in each class). We conduct experiments with angles and lighting under 5% area in the untargeted setting with Total Variation (TV) loss [45]. After printing patches with CANON iR-ADV C5535, we place them next to the corresponding item and photograph via

(a) Envelope → Switch (b) Pen → Ladle (c) Ruler → Spotlight (d) White DAPatch

**Fig. 6.** Physical attacks of DAPatch in the untargeted setting. When the light and shadow remain unchanged, for example, (a) is originally predicted as envelope, we use the camera to photograph and generate a patch, and attach it near the object. It is predicted as the switch after photographing again.

an iPhone 12. The initial ASR (%) on 50 original images is 16. Then, the ASR in these angles ($-30°$, $0°$, and $30°$) are 40, 60, and 44 in the middle lightning, respectively. The ASR (%) in different lightning (low, middle, and high lighting) are 56, 60, and 60 in $0°$, respectively. Figure 6 shows some examples of physical attacks. We choose GAP and PS-GAN for comparison. Under the same experimental parameters (middle lighting, $0°$), the ASR (%) is 34 and 44, which is 26 and 16 less than DAPatch (60). In general, our DAPatch is less robust to angles (since it affects the shape of patches), but still outperforms other patch attacks. We also try to use the white DAPatch to attack. Although the white DAPatch only receives 30 of the ASR, it also proves that the deformation attack is effective. Please refer to Appendix 5 for more details.

## 5 Conclusions

As a special form of adversarial attack, patch attacks have been extensively studied and analyzed due to their threatening nature to the real world. However, due to the lack of an effective modeling strategy, previous work has to restrict the patches to fixed shapes, such as circles or rectangles, which neglects the shape of patches as a factor in patch attacks. In this paper, we present a new Deformable Patch Representation that exploits triangle geometry and adopts a differentiable mapping process between contour modeling and masking. To further improve attack performance, we propose a joint optimization algorithm named Deformable Adversarial Patch which supports simultaneous and efficient optimization of shape and texture. Extensive experiments show that a particular shape can improve attack performance. Finally, DAPatch poses a great threat in both the digital and real-world against various DNN architectures.

**Acknowledgements.** This work was done when Zhaoyu Chen was an intern at Youtu Lab, Tencent. This work was supported by National Natural Science Foundation of China (No.62072112), Scientific and Technological Innovation Action Plan of Shanghai Science and Technology Committee (No.20511103102), Fudan University-CIOMP Joint Fund (No. FC2019-005), and Double First-class Construction Fund (No. XM03211178).

## References


1. Athalye, A., Carlini, N., Wagner, D.A.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 274–283. PMLR (2018), http://proceedings.mlr.press/v80/athalye18a.html 13
2. Belongie, S.J., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002). https://doi.org/10.1109/34.993558, https://doi.org/10.1109/34.993558 4
3. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch (2017), http://arxiv.org/abs/1712.09665 2, 3, 8
4. Chen, C., Zhang, J., Lyu, L.: Gear: A margin-based federated adversarial training approach 1
5. Chen, Z., Li, B., Xu, J., Wu, S., Ding, S., Zhang, W.: Towards practical certifiable patch defense with vision transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15148–15158 (June 2022) 4
6. Chiang, P., Ni, R., Abdelkader, A., Zhu, C., Studer, C., Goldstein, T.: Certified defenses for adversarial patches. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=HyeaSkrYPH 4
7. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 764–773. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.89, https://doi.org/10.1109/ICCV.2017.89 2, 4
8. Ding, L., Wang, Y., Yuan, K., Jiang, M., Wang, P., Huang, H., Wang, Z.J.: Towards universal physical attacks on single object tracking. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. pp. 1236–1245. AAAI Press (2021), https://ojs.aaai.org/index.php/AAAI/article/view/16211 1, 2, 3
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=YicbFdNTTy 3, 9
10. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1625–1634. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00175, http://openaccess.thecvf.com/content_cvpr_2018/html/Eykholt_Robust_Physical-World_Attacks_CVPR_2018_paper.html 2
11. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves

accuracy and robustness. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=Bygh9j09KX 2, 4, 10
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6572 1, 11
13. Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., Huang, F., Ma, L.: Spatiotemporal inconsistency learning for deepfake video detection. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3473–3481 (2021) 1
14. Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., Ma, L.: Delving into the local: Dynamic inconsistency learning for deepfake video detection. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence (2022) 1
15. Gu, Z., Li, F., Fang, F., Zhang, G.: A novel retinex-based fractional-order variational model for images with severely low light. IEEE Trans. Image Process. **29**, 3239–3253 (2020) 1
16. Gu, Z., Li, F., Lv, X.G.: A detail preserving variational model for image retinex. Applied Mathematical Modelling **68**, 643–661 (2019) 1
17. Gu, Z., Yao, T., Yang, C., Yi, R., Ding, S., Ma, L.: Region-aware temporal inconsistency learning for deepfake video detection. In: Proceedings of the 31th International Joint Conference on Artificial Intelligence (2022) 1
18. Hayes, J.: On visible adversarial perturbations & digital watermarking. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1597–1604. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPRW.2018.00210, http://openaccess.thecvf.com/content_cvpr_2018_workshops/w32/html/Hayes_On_Visible_Adversarial_CVPR_2018_paper.html 4, 12
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.90, https://doi.org/10.1109/CVPR.2016.90 3, 9
20. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2261–2269. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.243, https://doi.org/10.1109/CVPR.2017.243 3, 9
21. Huang, H., Wang, Y., Chen, Z., Tang, Z., Zhang, W., Ma, K.: Rpattack: Refined patch attack on general object detectors. In: 2021 IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5-9, 2021. pp. 1–6. IEEE (2021). https://doi.org/10.1109/ICME51207.2021.9428443, https://doi.org/10.1109/ICME51207.2021.9428443 1, 3
22. Huang, H., Wang, Y., Chen, Z., Zhang, Y., Li, Y., Tang, Z., Chu, W., Chen, J., Lin, W., Ma, K.K.: Cmua-watermark: A cross-model universal adversarial watermark for combating deepfakes. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 989–997 (2022) 1
23. Huang, L., Gao, C., Zhou, Y., Xie, C., Yuille, A.L., Zou, C., Liu, N.: Universal physical camouflage attacks on object detectors. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020,

Seattle, WA, USA, June 13-19, 2020. pp. 717–726. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00080, https://openaccess.thecvf.com/content_CVPR_2020/html/Huang_Universal_Physical_Camouflage_Attacks_on_Object_Detectors_CVPR_2020_paper.html 1, 3

24. Karmon, D., Zoran, D., Goldberg, Y.: Lavan: Localized and visible adversarial noise. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 2512–2520. PMLR (2018), http://proceedings.mlr.press/v80/karmon18a.html 2, 3, 8
25. Kong, X., Liu, X., Gu, J., Qiao, Y., Dong, C.: Reflash dropout in image super-resolution. arXiv preprint arXiv:2112.12089 (2021) 1
26. Kong, X., Zhao, H., Qiao, Y., Dong, C.: Classsr: A general framework to accelerate super-resolution networks by data characteristic. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12016–12025 (June 2021) 1
27. Levine, A., Feizi, S.: (de)randomized smoothing for certifiable defense against patch attacks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/47ce0875420b2dbacfc5535f94e68433-Abstract.html 4, 12
28. Li, B., Sun, Z., Guo, Y.: Supervae: Superpixelwise variational autoencoder for salient object detection. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 8569–8576. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33018569, https://doi.org/10.1609/aaai.v33i01.33018569 3
29. Li, B., Sun, Z., Li, Q., Wu, Y., Hu, A.: Group-wise deep object co-segmentation with co-attention recurrent neural network. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 8518–8527. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00861, https://doi.org/10.1109/ICCV.2019.00861 3
30. Li, B., Sun, Z., Tang, L., Hu, A.: Two-b-real net: Two-branch network for real-time salient object detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019. pp. 1662–1666. IEEE (2019). https://doi.org/10.1109/ICASSP.2019.8683022, https://doi.org/10.1109/ICASSP.2019.8683022 3
31. Li, B., Sun, Z., Tang, L., Sun, Y., Shi, J.: Detecting robust co-saliency with recurrent co-attention neural network. In: Kraus, S. (ed.) Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. pp. 818–825. ijcai.org (2019). https://doi.org/10.24963/ijcai.2019/115, https://doi.org/10.24963/ijcai.2019/115 3
32. Li, B., Sun, Z., Wang, Q., Li, Q.: Co-saliency detection based on hierarchical consistency. In: Amsaleg, L., Huet, B., Larson, M.A., Gravier, G., Hung, H., Ngo, C., Ooi, W.T. (eds.) Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019. pp. 1392–

1400. ACM (2019). https://doi.org/10.1145/3343031.3351016, https://doi.org/10.1145/3343031.3351016 3
33. Li, B., Xu, J., Wu, S., Ding, S., Li, J., Huang, F.: Detecting adversarial patch attacks through global-local consistency. In: Song, D., Tao, D., Yuille, A.L., Anandkumar, A., Liu, A., Chen, X., Li, Y., Xiao, C., Yang, X., Liu, X. (eds.) ADVM '21: Proceedings of the 1st International Workshop on Adversarial Learning for Multimedia, Virtual Event, China, 20 October 2021. pp. 35–41. ACM (2021). https://doi.org/10.1145/3475724.3483606, https://doi.org/10.1145/3475724.3483606 4
34. Li, Y., Yu, Q., Tan, M., Mei, J., Tang, P., Shen, W., Yuille, A.L., Xie, C.: Shape-texture debiased neural network training. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=Db4yerZTYkz 2, 4, 11
35. Liu, A., Liu, X., Fan, J., Ma, Y., Zhang, A., Xie, H., Tao, D.: Perceptual-sensitive GAN for generating adversarial patches. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 1028–1035. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33011028, https://doi.org/10.1609/aaai.v33i01.33011028 3, 8
36. Liu, S., Chen, Z., Li, W., Zhu, J., Wang, J., Zhang, W., Gan, Z.: Efficient universal shuffle attack for visual object tracking. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2739–2743. IEEE (2022) 1, 3
37. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows (2021), https://arxiv.org/abs/2103.14030 3, 9
38. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=rJzIBfZAb 1, 11
39. Malik, J., Belongie, S.J., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. Int. J. Comput. Vis. **43**(1), 7–27 (2001). https://doi.org/10.1023/A:1011174803800, https://doi.org/10.1023/A:1011174803800 4
40. Naseer, M., Khan, S., Porikli, F.: Local gradients smoothing: Defense against localized adversarial attacks. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019. pp. 1300–1307. IEEE (2019). https://doi.org/10.1109/WACV.2019.00143, https://doi.org/10.1109/WACV.2019.00143 4, 12
41. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Pro-

cessing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 8024–8035 (2019), https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html 8

42. Rao, S., Stutz, D., Schiele, B.: Adversarial training against location-optimized adversarial patches. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part V. Lecture Notes in Computer Science, vol. 12539, pp. 429–448. Springer (2020). https://doi.org/10.1007/978-3-030-68238-5_32, https://doi.org/10.1007/978-3-030-68238-5_32 4
43. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y, https://doi.org/10.1007/s11263-015-0816-y 8, 13
44. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4510–4520. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00474, http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html 3, 9
45. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 1528–1540. ACM (2016). https://doi.org/10.1145/2976749.2978392, https://doi.org/10.1145/2976749.2978392 13
46. Shen, T., Zhang, J., Jia, X., Zhang, F., Huang, G., Zhou, P., Kuang, K., Wu, F., Wu, C.: Federated mutual learning. arXiv preprint arXiv:2006.16765 (2020) 1
47. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556 3, 9
48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014), http://arxiv.org/abs/1312.6199 1
49. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (2019), http://proceedings.mlr.press/v97/tan19a.html 3, 9
50. Tang, L., Li, B.: CLASS: cross-level attention and supervision for salient objects detection. In: Ishikawa, H., Liu, C., Pajdla, T., Shi, J. (eds.) Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020, Revised Selected Papers, Part III. Lecture Notes in Computer

Science, vol. 12624, pp. 420–436. Springer (2020). https://doi.org/10.1007/978-3-030-69535-4_26, https://doi.org/10.1007/978-3-030-69535-4_26 3
51. Tang, L., Li, B., Zhong, Y., Ding, S., Song, M.: Disentangled high quality salient object detection. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 3560–3570. IEEE (2021). https://doi.org/10.1109/ICCV48922.2021.00356, https://doi.org/10.1109/ICCV48922.2021.00356 3
52. Wang, Y., Song, W., Tao, W., Liotta, A., Yang, D., Li, X., Gao, S., Sun, Y., Ge, W., Zhang, W., Zhang, W.: A systematic review on affective computing: Emotion models, databases, and recent advances. Information Fusion **83–84**, 19–52 (2022). https://doi.org/10.1016/j.inffus.2022.03.009 1
53. Wang, Y., Sun, Y., Huang, Y., Liu, Z., Gao, S., Zhang, W., Ge, W., Zhang, W.: Ferv39k: A large-scale multi-scene dataset for facial expression recognition in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20922–20931 (June 2022) 1
54. Wu, T., Tong, L., Vorobeychik, Y.: Defending against physically realizable attacks on image classification. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=H1xscnEKDr 4
55. Xiang, C., Bhagoji, A.N., Sehwag, V., Mittal, P.: Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In: Bailey, M., Greenstadt, R. (eds.) 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021. pp. 2237–2254. USENIX Association (2021), https://www.usenix.org/conference/usenixsecurity21/presentation/xiang 4, 12
56. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.L.: Adversarial examples for semantic segmentation and object detection. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 1378–1387. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.153, https://doi.org/10.1109/ICCV.2017.153 3
57. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 12190–12199. IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.01221, https://doi.org/10.1109/CVPR42600.2020.01221 2, 4
58. Zhang, J., Chen, C., Dong, J., Jia, R., Lyu, L.: Qekd: Query-efficient and data-free knowledge distillation from black-box models. arXiv preprint arXiv:2205.11158 (2022) 1
59. Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Xu, J., Ding, S., Wu, C.: A practical data-free approach to one-shot federated learning with heterogeneity. arXiv preprint arXiv:2112.12371 (2021) 1
60. Zhang, J., Li, B., Xu, J., Wu, S., Ding, S., Zhang, L., Wu, C.: Towards efficient data free black-box adversarial attack. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15115–15125 (June 2022) 1
61. Zhang, J., Zhang, L., Li, G., Wu, C.: Adversarial examples for good: Adversarial examples guided imbalanced learning. arXiv preprint arXiv:2201.12356 (2022) 1
62. Zhang, Z., Yuan, B., McCoyd, M., Wagner, D.A.: Clipped bagnet: Defending against sticker attacks with clipped bag-of-features. In: 2020 IEEE Security and Privacy Workshops, SP Workshops, San Francisco, CA, USA, May 21, 2020.

pp. 55–61. IEEE (2020). https://doi.org/10.1109/SPW50608.2020.00026, https://doi.org/10.1109/SPW50608.2020.00026 4
63. Zhao, H., Kong, X., He, J., Qiao, Y., Dong, C.: Efficient image super-resolution using pixel attention. In: European Conference on Computer Vision. pp. 56–72. Springer (2020) 1
64. Zhong, Y., Li, B., Tang, L., Kuang, S., Wu, S., Ding, S.: Detecting camouflaged object in frequency domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4504–4513 (June 2022) 3
65. Zhong, Y., Li, B., Tang, L., Tang, H., Ding, S.: Highly efficient natural image matting. CoRR **abs/2110.12748** (2021), https://arxiv.org/abs/2110.12748 3
66. Zhou, Q., Feng, Z., Gu, Q., Cheng, G., Lu, X., Shi, J., Ma, L.: Uncertainty-aware consistency regularization for cross-domain semantic segmentation. Computer Vision and Image Understanding p. 103448 (2022) 1
67. Zhou, Q., Zhang, K.Y., Yao, T., Yi, R., Ding, S., Ma, L.: Adaptive mixture of experts learning for generalizable face anti-spoofing. In: Proceedings of the 30th ACM International Conference on Multimedia (2022) 1
68. Zhou, Q., Zhang, K.Y., Yao, T., Yi, R., Sheng, K., Ding, S., Ma, L.: Generative domain adaptation for face anti-spoofing. In: European Conference on Computer Vision. Springer (2022) 1