

# We Can Always Catch You: Detecting Adversarial Patched Objects WITH or WITHOUT Signature

Bin Liang, Jiachun Li, and Jianjun Huang  
*Renmin University of China*  
`{liangb, jclee, hjj}@ruc.edu.cn`

## Abstract

Recently, the object detection based on deep learning has proven to be vulnerable to adversarial patch attacks. The attackers holding a specially crafted patch can hide themselves from the state-of-the-art person detectors, e.g., YOLO, even in the physical world. This kind of attack can bring serious security threats, such as escaping from surveillance cameras. In this paper, we deeply explore the detection problems about the adversarial patch attacks to the object detection. First, we identify a leverageable signature of existing adversarial patches from the point of the visualization explanation. A fast signature-based defense method is proposed and demonstrated to be effective. Second, we design an improved patch generation algorithm to reveal the risk that the signature-based way may be bypassed by the techniques emerging in the future. The newly generated adversarial patches can successfully evade the proposed signature-based defense. Finally, we present a novel signature-independent detection method based on the internal content semantics consistency rather than any attack-specific prior knowledge. The fundamental intuition is that the adversarial object can *appear locally but disappear globally* in an input image. The experiments demonstrate that the signature-independent method can effectively detect the existing and improved attacks. It has also proven to be a general method by detecting unforeseen and even other types of attacks without any attack-specific prior knowledge. The two proposed detection methods can be adopted in different scenarios, and we believe that combining them can offer a comprehensive protection.

## 1 Introduction

One of the most important applications of deep learning is the object detection [7, 17, 36, 37, 38], which is designed to identify and locate the instances of specific target classes, e.g., humans and cars, in images or videos. The detected object is marked with a bounding box and a target class label. Object detection has been applied to many tasks, including video surveillance, autonomous driving, information retrieval, etc. Obviously, the robustness of the object detectors is critical to the tasks. In the past years, various techniques [5, 10, 15, 19, 26, 33, 48] are proposed to attack the artificial intelligence models, e.g., to fool a classifier to misclassify a panda as a gibbon by introducing some imperceptible perturbations [19].

Unsurprisingly, object detection models have also become the attack target. Some attack methods [49, 52, 56] have been proposed to fool object detectors, making the target object evade the detection. Some of them can even successfully launch evasion attacks in the physical world rather than only in the digital world. Thys et al. [49] propose an adversarial patch generation algorithm to fool YOLO [37], which is the most popular object detection model. As demonstrated by them, a person with the patch can hide from the surveillance camera equipped with a YOLO detector. The adversarial patch can also be directly printed on the T-shirt to fool YOLO and Faster R-CNN object detection models [52, 56]. In a similar way, Komkov et

al. [23] design an algorithm to generate the adversarial sticker, which can be stuck on the hat to fool some state-of-the-art facial recognition systems. We believe that the more new attack methods will emerge in the future.

An important question arises naturally as *how we can effectively defend the known and unknown adversarial patch attacks*.

In this study, our first observation is that the adversarial patches generated by the existing technique [49] possess a distinct characteristics, which can be leveraged to effectively identify the patch pixels from the target video frames (can be regarded as a sequence of images). In fact, we are surprised to find that many pixels of the adversarial patch generated to hide person objects have a significant positive (not negative as expected) influence on classifying the candidate bounding box as the *Person* class.

Based on the observation, a straightforward but effective signature-based adversarial patch defense method is proposed. Grad-CAM [39], a gradient-based deep learning visualization method, is employed to measure the influence by getting a score with respect to every pixels. The pixels with the highest Grad-CAM scores are identified and filtered out from the current frame image. The object detection is performed to the filtered image instead of the original one, to suppress the effect of the adversarial patch as far as possible. The experiment demonstrates that the defense technique can effectively detect the object hidden by the adversarial patch.

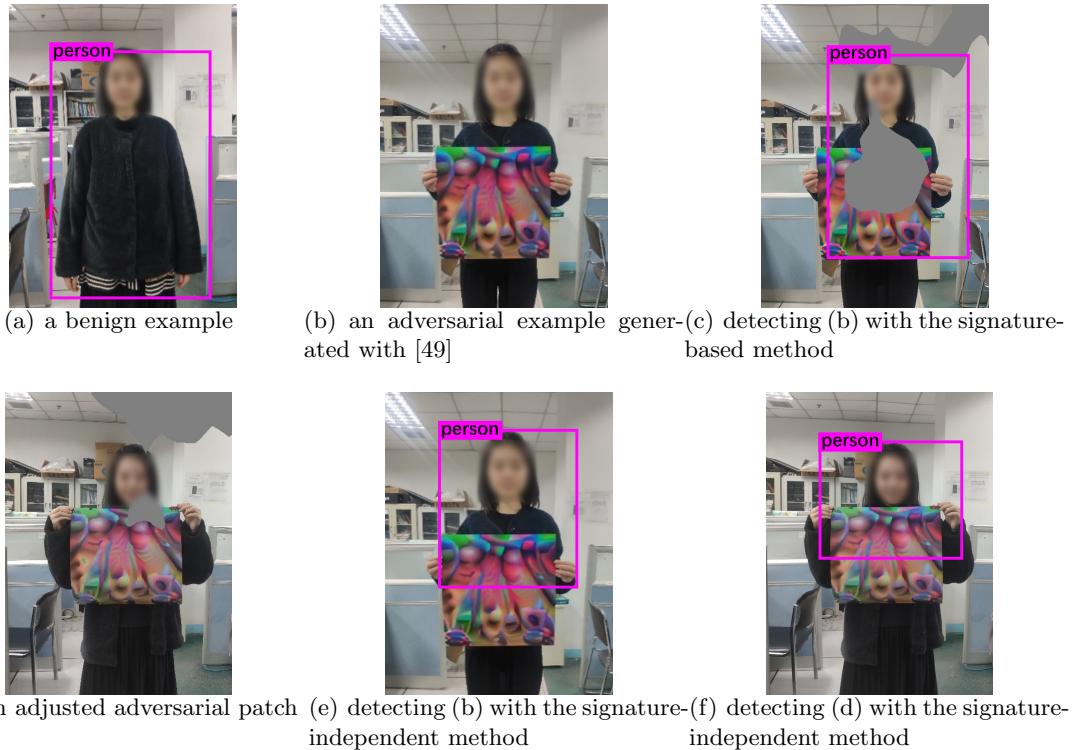


Figure 1: Object detection results. (a) the benign example is detected by YOLO; (b) the adversarial example generated with [49] evades YOLO; (c) the adversarial example in (b) can be detected with the signature-based method; (d) the adjusted adversarial patch can bypass the signature-based method; (e) the adversarial example in (b) can be detected by the signature-independent method; (f) the adjusted adversarial example in (d) can be detected by the signature-independent method too.

From Figure 1(a), we can see that YOLO can successfully detect a person and mark her with a pink rectangular box. However, as illustrated in Figure 1(b), she can hide herself from the detector by holding an adversarial patch generated with the algorithm presented in [49]. By integrating the proposed signature-based defense technique, the YOLO model can detect the person from the filtered input. As shown in Figure 1(c), the person is marked out again after

filtering the top 15% pixels with the highest Grad-CAM scores (colored in gray). The details of the signature-based defense will be presented in Section 3.

The overhead introduced by the above defense is very limited (about 85 milliseconds). It can be easily integrated with a trained object detection model to provide an efficient protection. However, as a sort of signature-based technique, the method has an essential limitation, i.e., the adopted signature is attack-specific, and it may lose its capability when the attackers upgrade their techniques purposefully. To demonstrate this risk, we propose an iteration algorithm to generate the adversarial patch without introducing detectable signatures.

In the iteration, the pixels of the adversarial patch are elaborately adjusted according to their Grad-CAM scores, to avoid introducing pixels with high scores and preserve the attack effectiveness at the same time. Consequently, as exhibited in Figure 1(d), the person with the adjusted patch can evade the YOLO detector. The details of the algorithm will be in Section 4.

Although there may still be some potential leverageable signatures in the adjusted patch, we believe that attackers can design a targeted adversarial patch generation algorithm to erase the corresponding signatures. To this end, we propose a signature-independent adversarial patched objects detection method, which is robust enough against the unknown attack techniques. The method is based on an insightful observation that *the local and global content semantics is inconsistent in adversarial patched examples*. In a benign example, if an object can be detected from a part of it, the object can also be detected from a larger part. However, in an adversarial example, the object may be invisible for the detector once the larger part includes enough adversarial pixels. In other words, the adversarial object can *appear locally but disappear globally*.

Accordingly, we design a region growing algorithm to check whether there is this kind of inconsistency in the current input and detect potential adversarial examples. As shown in Figure 1(e) and 1(f), the person can be successfully detected no matter which types of adversarial patches she holds. The experiment also shows that our detection technique can defend the upgraded adversarial patch techniques [56] without requiring any prior knowledge or parameters tuning. Furthermore, we demonstrate the concept of content semantics consistency is general and can be adopted to detect other types of local perturbation attacks, e.g., [8] and [21]. The details of the detection technique will be presented in Section 5.

Note that the two proposed defense methods can be used for different scenarios. The signature-based one can be adopted in efficiency-sensitive scenarios. It can provide effective defense via continuously identifying new signatures. Due to involving semantics analysis, the signature-independent method is inevitably relatively slow. However, it can definitely be used as a probe to catch the unknown attacks via sampling or be implemented in parallel. We believe that the two methods can be combined to provide a comprehensive protection against adversarial patch attacks.

In summary, the main contributions of this paper are as follows:

- A fast signature-based adversarial patch defense technique is proposed, which can effectively counter existing attack methods and easily be integrated with trained object detection models.
- An improved patch generation algorithm is designed to erase the detectable characteristic from adversarial patches. The risk that the signature-based defense may be ineffective to the future attacks is demonstrated.
- A novel general signature-independent adversarial patched objects detection method is presented based on the analysis of the local and global content semantics consistency. It is transparent to the attack technique and back-end object detection model, and can be applied to other types of local perturbation attacks.

## 2 Background

### 2.1 YOLO

YOLO [37] is one of the most popular object detection models. Different from the other two-step models [17, 18, 38], YOLO detects the object in only one step.

YOLO divides the input image into an  $S \times S$  grid. Each grid cell predicts  $B$  bounding boxes, and each bounding box is given 5 predictions:  $x, y, w, h$  and an *object confidence*. The former four determine the coordinates and size of the box, and the object confidence reflects the probability that the box contains an object. Besides, for each cell, YOLO also outputs  $P$  conditional *class probabilities*, the greatest one of which will be used to determine the class of the detected object. An object may cross multiple grid cells. But only one cell is responsible for detecting the object, which contains the center of the target object.

At test time, for each box, the product of the object confidence and the class probabilities is used to determine whether there is an object and which class the object is. Theoretically, the attacker can fool YOLO to miss a target object by either lowering the object confidence or class probabilities. The existing attack [49] chooses to reduce the object confidence of the target box by introducing an adversarial patch in the input.

### 2.2 Adversarial Patch Attack

Thys et al. [49] propose the concept of the adversarial patch to the object detection. As its name suggests, the adversarial patch is a small printable attack vector picture, which can be introduced into the input example like a “patch”. They design an optimization process to generate the adversarial patch targeting YOLO. The optimization goal consists of three components, the maximum object confidence  $L_{obj}$ , the non-printability score  $L_{nps}$ , and the total image variation  $L_{tv}$ . The total loss function of the optimization is the weighted sum of the three losses, as given in Equation (1).

$$L = L_{obj} + \alpha L_{tv} + \beta L_{nps} \quad (1)$$

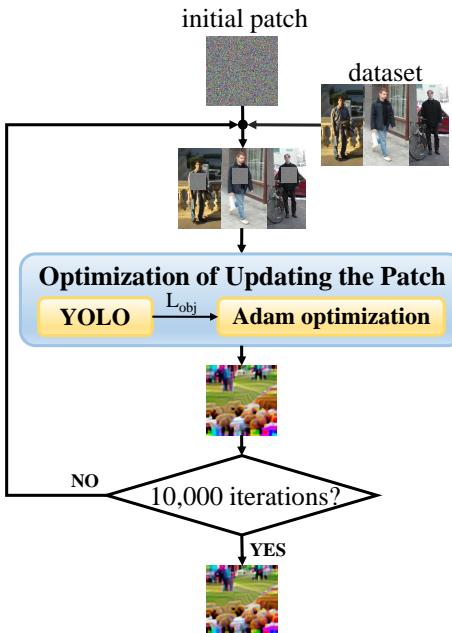


Figure 2: Workflow of the adversarial patch generation

Figure 2 illustrates the workflow of the optimization. The initial patch is a randomly generated image. In each optimization iteration, dozens of adversarial examples are generated by applying the current patch on the image samples in the MS COCO dataset [29]. YOLO is invoked to detect the examples and Adam algorithm [22] is employed to optimize the loss. The goal of the optimization is to minimize  $L$  with respect to the examples. During the process, all weights in the network are frozen and only the patch pixels are updated iteratively. After 10,000 iterations, the optimization is finished and outputs the final adversarial patch.

### 2.3 Fast Gradient Sign Method

Goodfellow et al. [19] propose an attack method named fast gradient sign method (FGSM) to generate adversarial examples against GoogLeNet [47]. FGSM is a milestone of adversarial attacks, which is easy to implement and very efficient. For an input  $x'$ , the corresponding adversarial example  $x'$  is straightforwardly crafted as Equation (2),

$$x' = x + \epsilon \cdot \text{sign}(\nabla x J(C, x, c)) \quad (2)$$

where  $c$  is the true class of  $x$ ,  $J(C, x, c)$  is the cost function used to train the DNN  $C$ . The hyperparameter  $\epsilon$  (ranging from 0.0 to 1.0) is used to set the perturbation intensity. The perturbation is computed as the sign of the model’s cost function gradient.

## 3 Signature-based Defense

### 3.1 Overview

To defend the adversarial patch, a natural way is to find an effective rule (e.g., a signature) to distinguish the adversarial pixels from the input example, and develop a mechanism to exclude them before feeding the example to the model. According to our observation, the adversarial patch actually possesses some distinct characteristics for identifying its pixels. In fact, the adversarial patch pixels often have measurable larger contribution to classifying the target bounding box as the attack target class, e.g., the *Person* class.

On the basis of the observation, we propose a defense method to detect the adversarial example by leveraging the signature. The basic idea behind our method is to regard the identified adversarial pixels as a kind of noise and employ image processing techniques to reduce its adversarial effect as far as possible. As shown in Figure 3, the key component of our method is a filter  $T$ . Before an example  $x$  is fed to the object detector  $D$  (e.g., YOLO), it will be denoised by  $T$  to generate a filtered sample  $T(x)$ . First, the potential patch pixels in  $x$  are identified with the Grad-CAM algorithm [39]. Second, the identified pixels are colored to a fixed value to erase their adversarial effect and generate  $T(x)$ . Finally, the object detector  $D$  will take as the input  $T(x)$  instead of the original  $x$ . With the help of the filter, the detector can effectively detect the object hidden by the adversarial patch.

### 3.2 Signature

In this study, the adversarial patch pixels are identified based on the intuition that *the patch actually works by introducing the significant negative contributions to object detection*.

In practice, the gradient can be leveraged to measure the contribution of the input dimension about a certain objective function, and to identify the dimensions of interest [19, 26]. To facilitate necessary observation, we borrow the idea of the gradient-based deep learning visual explanation [39, 57] to investigate the adversarial patch. Grad-CAM [39], a highly class-discriminative visualization method, is employed to highlight the potential important input pixels for certain model objectives. Each input pixel is assigned a quantitative importance estimation and we term it the *Grad-CAM score*.

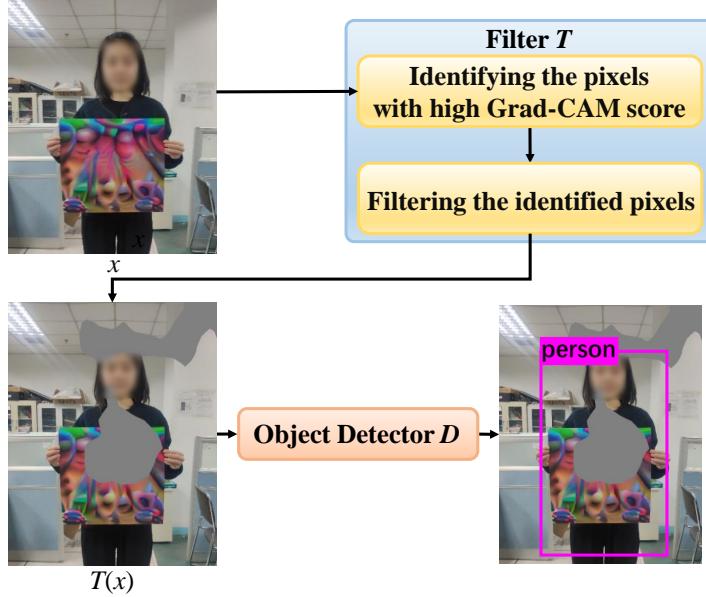


Figure 3: Workflow of the signature-based defense



Figure 4: The heat maps of adversarial examples. Blue represents low Grad-CAM score, and red represents high Grad-CAM score.

The Grad-CAM score for the output *Person* class probability is calculated as follows. First, the gradient of the *Person* class probability with respect to the output layer feature maps are computed by Equation (3):

$$g_k^p = \frac{\partial P_{Person}}{\partial F^k}, \quad (3)$$

where  $P_{Person}$  is the probability of the *Person* class, and  $F^k$  is the  $k$ -th output layer feature map. Second, as shown in Equation (4), we use the global average pooling to get neuron importance weights,

$$\alpha_k^p = \frac{1}{H \cdot W} \sum_i^H \sum_j^W g_k^p(i, j), \quad (4)$$

where  $(H, W)$  are the height and width of  $F^k$ . Third, the coarse Grad-CAM score  $L^p$  are computed with Equation (5),

$$L^p = \text{ReLU} \left( \sum_k \alpha_k^p F^k \right). \quad (5)$$

Finally,  $L^p$  is upsampled to the input image resolution using bilinear interpolation to get fine-grained Grad-CAM scores for each input pixel.

We try to use the Grad-CAM score as a metric to analyze the adversarial example with the patch generated by the attack method [49]. Surprisingly, we observe that the Grad-CAM scores of adversarial patch pixels for the *Person* class are often much higher than the other input pixels. In other words, the pixels of an adversarial patch are likely to make more contributions rather than fewer as we expected to burdening the person classification, which is a sub-task of person object detection.

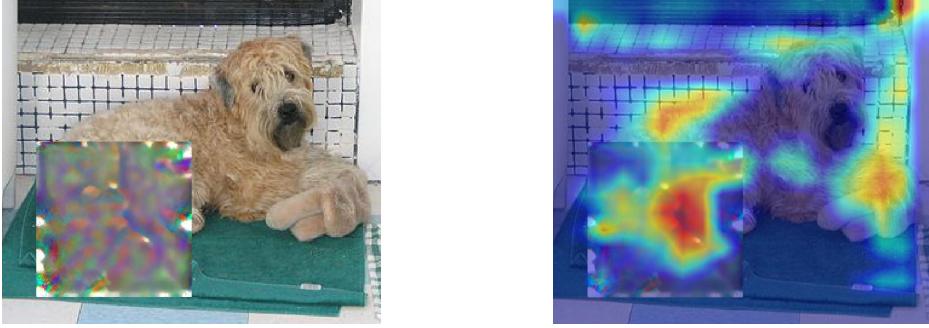


Figure 5: The heat map of the adversarial example targeting the *Dog* class.

The phenomenon is counter-intuitive. We visualize the magnitude of the Grad-CAM scores and generate corresponding heat maps. From Figure 4, it can be seen that the adversarial patch owns most of the hottest points no matter in the digital world or the physical world.

By investigating the optimization process of the adversarial patch generation, we find that the phenomenon results directly from the way of updating the patch pixels. The optimization objective is to decrease the object confidence of the example. This can make the detector believe there is not an object of interest in the current bounding box. To achieve this goal, during the optimization, the pixel value is iteratively updated according to the direction of its gradient about the output object confidence. However, according to our observation, the direction of the object confidence gradient is opposite to that of the Grad-CAM score gradient for many patch pixels. Our statistics show it is true for about a half (50.3%) of patch pixels but for only 7.93% of normal pixels. We also generate the adversarial patch targeting the other classes, e.g., the *Dog* class, to check whether the phenomenon is common. The answer is yes. As shown in Figure 5, the Grad-CAM score for the *Dog* class are much higher in an adversarial patch to hide a dog.

So far, the interpretability of deep learning is still an open problem. A possible explanation for the phenomenon is that applying the patch with high Grad-CAM scores can introduce some phantom person objects in the front of the target person object, and the effect of “*person in person*” actually violates the common knowledge learned from training examples. Consequently, the detector may be “confused” to miss the true target person. It is very difficult, if not impossible, to provide the perfect theoretical analysis for the phenomenon. However, the characteristics of the adversarial patch can be employed as an effective signature.

### 3.3 Filter

The first task of the filter is to identify the pixels with high Grad-CAM scores about the target class probability. Without loss of generality, we choose the *Person* class as the target class in this study. In YOLO, the target class probability  $P_{Person}$  can be collected in the last layer, and the Grad-CAM scores can be easily computed with Grad-CAM [39]. The top  $n\%$  pixels with high Grad-CAM scores are identified and filtered.

There is a trade-off on choosing a proper  $n$ . If  $n$  is too small, the adversarial effect cannot be effectively suppressed, and the object person can still be hidden by the patch. On the contrary, a large  $n$  may result in the benign pixels being filtered and missing the object person as well. As shown in Figure 6, the person object is missed in the filtered inputs, when  $n$  is under 10 or above 30. To determine a reasonable hyper-parameter  $n$ , an empirical study is performed on dozens of samples. We find that when  $n$  is set between 15 and 25, the proposed defense technique can achieve the best result.

We employ a straightforward way to filter the identified pixels by coloring them in gray, i.e., (128, 128, 128) in RGB. According to our experiments, coloring them in gray works better than other colors, e.g., black or white. Some of non-patch pixels may also possess high Grad-CAM

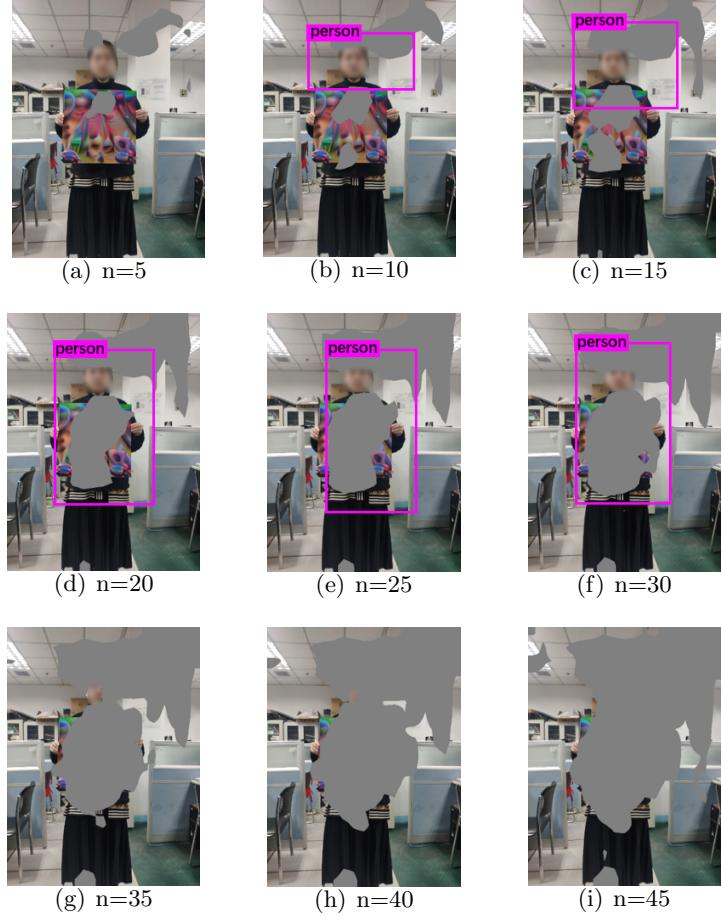


Figure 6: The defense performance when  $n$  ranges from 5 to 45.

scores. Filtering them may remove some details of the target person. Fortunately, the modern object detector, e.g., YOLO, can detect a person object just from a very small part of her or him. Filtering some benign pixels does not compromise the model detection performance.

### 3.4 Evaluation

On average, generating an adversarial patch takes six hours in our GPU server. To evaluate our defense technique, we spend hundreds of hours to construct a data set, which consists of 50 adversarial examples and 50 benign examples. The hyper-parameter  $n$  is set to 15.

Without our defense, the YOLO model can correctly detect all the 50 benign examples, but miss all adversarial ones. After applying our technique, 96 examples are successfully detected and only four adversarial examples evade the detection. Besides, the performance overhead introduced by the defense is totally acceptable. Filtering an input example only requires about 85 milliseconds.

The experiments demonstrate that our technique can be used to effectively defend existing adversarial patch attacks.

## 4 Evading Signature-based Defense

### 4.1 Overview

As presented in Section 3, the signature-based defense method is proven to be effective against the existing attacks. However, the employed signature is attack-specific, and there is a risk

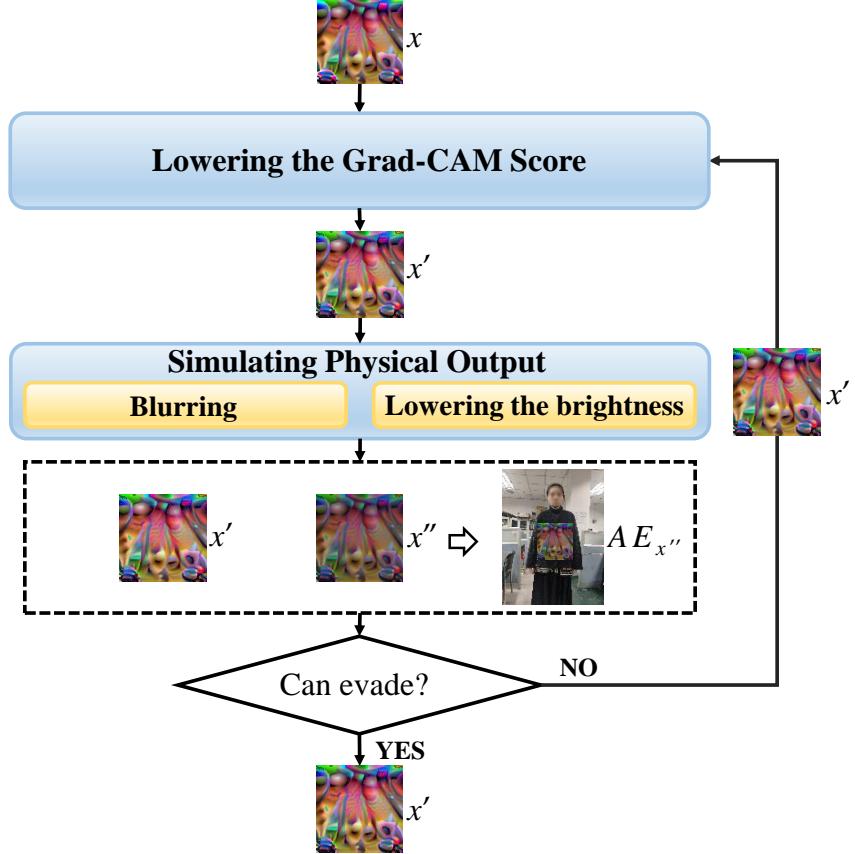


Figure 7: Workflow of adjusting patch.

that the defense may be ineffective when the attack techniques are upgraded purposefully. In this section, an improved attack technique is proposed to demonstrate this risk. We design an iteration algorithm to dress up the adversarial patch  $x$ . As illustrated in Figure 7, two procedures are introduced to manipulate the patch elaborately.

The patch pixels are first tuned to produce an adjusted version  $x'$  according to the gradients about their Grad-CAM scores and object confidences. The two gradients are calculated with the back propagation algorithm. The aim of the procedure is to lower the Grad-CAM score and preserve the attack effectiveness at the same time. Note that the adjustment may introduce some subtle details in  $x'$ , such as the superfine texture. It is very difficult to accurately reproduce the details with the printer in the physical world. Besides, the physical patch often has a lower brightness than its digital version. To this end, we develop another procedure to simulate the transformation from the digital patch to the physical version. The simulated physical patch  $x''$  is used to generate an adversarial example  $AE_{x''}$  to determine whether  $x''$  is likely to work well in the physical world. If the adjustment achieves the expected goal, i.e.,  $AE_{x''}$  can evade the signature-based defense in the digital world, the algorithm will end and output  $x'$ . Otherwise,  $x'$  will be iteratively polished to lower its Grad-CAM score further.

## 4.2 Lowering the Grad-CAM Score

To remove the detectable signature from adversarial patches, a straightforward way is to slightly adjust the patch pixel value in the opposite direction of its signature's gradient, i.e., the gradient of Grad-CAM score, as done in FGSM [19]. However, there is a side effect of the adjustment. For some patch pixels, the gradient directions of Grad-CAM score and the object confidence are just opposite. In other words, the object confidence of the input example will be increased and the attack may fail, if the patch pixel value is directly adjusted only according to the Grad-CAM



(a) digital example (success)  
(b) physical example (fail)  
(c) digital example after simulation (success)  
(d) physical example after simulation (success)

Figure 8: Examples in the digital and physical worlds.

score  $S_{GC}$ .

To address the issue, the adjustment is limited to the pixels whose object confidence and Grad-CAM score gradients have the same direction, or the absolute magnitude of the Grad-CAM score gradient is much larger (three times) than that of the object confidence gradient. In each iteration, the pixel value  $x_{ij}$  is updated as Equation (6):

$$x_{ij} = x_{ij} - \epsilon \cdot sign(\nabla x_{ij} S_{GC}) \quad (6)$$

where  $\epsilon$  is the amplitude of the adjustment, which is set to 0.1 in our experiment. The iteration ends when the sum of Grad-CAM scores of all patch pixels is reduced enough (about by 70%) to evade the defense.

### 4.3 Simulating Physical Output

The transformation is simulated in two ways, i.e., blurring and lowering the brightness.

**Blurring.** We observe that the digital adversarial examples have some superfine details, i.e. the texture edges, which cannot be totally captured by the camera or reproduced in the physical world. Gaussian Blur has proven to be effective to reduce the fine-grained features and make the digital spliced image more natural [43]. We employ it to reduce the superfine details in the patch. The key parameter, the blurring kernel size, is set to 2 in our experiment.

**Lowering the brightness.** We employ the Gaussian distribution to analyze the brightness difference between the digital patch and its physical version, as done in [32, 50]. The analysis result shows that the brightness of the printed patch is commonly lower than that of the digital patch, and the difference follows the Gaussian distribution with mean 30. Accordingly, the pixel value in the simulated patch is decreased by 30 for all three channels, R, G and B.

As illustrated in Figure 8, an adjusted patch can work in the digital world but fails after being printed. By introducing the simulation in the adjustment iteration, the obtained patch can attack successfully in the real world. The high-resolution patches are presented in Appendix A.

### 4.4 Evaluation

We directly employ the adversarial patches used in Section 3.4 to evaluate the improved attack. The patches are adjusted with the above algorithm, and 50 corresponding adversarial examples are generated. Our experiment shows that 28 (56%) of them can evade the signature-based defense. We believe that we can get more effective attack examples via fine tuning. Although not all adversarial examples can evade the defense, it is well demonstrated that the signature-based technique can be bypassed via concealing the signature purposefully.

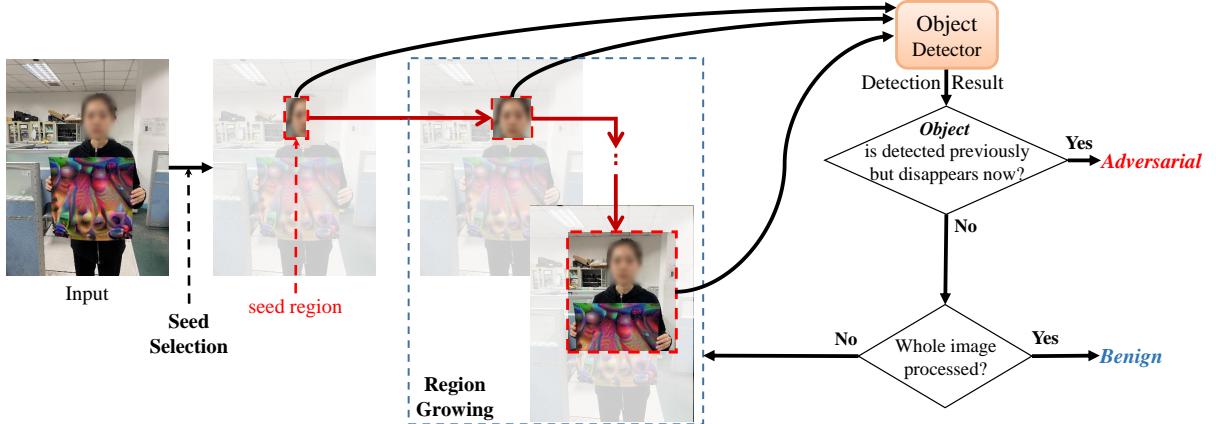


Figure 9: Detecting the adversarial example with the region growing algorithm.

## 5 Signature-Independent Defense

### 5.1 Overview

Section 4 demonstrates the signature-based approach is prone to be bypassed once the attack technique is pertinently improved. To defend the upgraded adversarial patch generation algorithm and other potential sophisticated attack techniques, we propose a general and signature-independent detection method in this section.

The proposed method is based on an internal semantics structure rather than some kind of mutable superficial characteristics. In fact, the local and global content semantics is consistent in benign examples but inconsistent in adversarial ones. If a benign object can be detected from a part of the input image, it will certainly be detected from a larger one. In other words, if an object can be detected locally, it should be detected globally. However, an adversarial object can become disappeared in the larger part containing enough adversarial patch pixels. This semantics structure can be employed as a robust detection rule to identify adversarial examples. Accordingly, we develop a region growing algorithm to detect adversarial examples by checking whether there is an inconsistency in the current input.

More specifically, as shown in Figure 9, the algorithm begins with a seed region, which is an image block selected from the input image. Iteratively, the region grows by including the neighbouring blocks following a proper growing direction. At each step, the grown region is fed to the object detector to check whether there is an object of interest (e.g., a person), and the detection result is recorded. When an object is detected in the previous regions but missed in the current region, we believe the input is an adversarial example and the object identified in the last region will be output. In this way, the adversarial examples can be effectively detected without requiring any attack-specific knowledge.

### 5.2 Content Semantics Consistency

The state-of-the-art object detection model, e.g., YOLO, is trained to be able to successfully detect an object of interest even the input only contains a part of the object. It is natural that when more parts are included in the input, the detection model is undoubtedly able to identify the object. Take the person detection scenario as an example. If a person can be detected in a local part, the person can also be detected in global. We demonstrate it in Figure 10, in which YOLO can consistently detect a person object from a small part, a larger part, and the whole of the input. We term the invariance between the local and global detection results *the content semantics consistency*.

For a benign example, the normal pattern of the detection result for the sequence of growing regions should be “*detected → detected → ... → detected*” or “*non-detected → non-detected →*



Figure 10: The content semantics consistency within the benign example.



Figure 11: The content semantics inconsistency within the adversarial example.

*non-detected  $\rightarrow$  detected  $\rightarrow$  detected  $\rightarrow \dots \rightarrow$  detected*”. The former pattern indicates that the object can always be detected, and the latter indicates the object may be missed at first but be always hit starting from a certain region. A detected object should not suddenly vanish when given a larger input region. In other words, the normal detection state transition should be “*non-detected  $\rightarrow$  non-detected*”, “*non-detected  $\rightarrow$  detected*”, or “*detected  $\rightarrow$  detected*”, but not “*detected  $\Rightarrow$  non-detected*” forever.

However, this kind of content semantics consistency often does not hold in an adversarial example. As shown in Figure 11, for a person with an adversarial patch, she is much likely to be detected by an object detector just from a local part of her, but disappears when the grown region includes enough adversarial patch pixels. Consequently, a counter-common-sense detection state transition “*detected  $\Rightarrow$  non-detected*” emerges, which should never happen for a benign input.

On the basis of the above discussion, we are confident that the content semantics consistency/inconsistency can be leveraged as a detection criterion to effectively discover adversarial examples. We can monitor the trace of object detection results to check whether there is an abnormal detection state transition, i.e., “*detected  $\Rightarrow$  non-detected*”.

It should be noted that the detection criterion is robust. In practice, the attacker would not want to create an oversize adversarial patch for preserving its utility, giving the modern object detection models a chance to identify the adversarial object from its uncovered part. For example, YOLO is able to effectively detect a person object just from a human face or the lower body. Meanwhile, in most scenarios, using a large adversarial patch to cover the face or most of the body is meaningless and weird. More importantly, the detection criterion is general and independent of specific attack techniques. We can apply it to detect any kinds of adversarial

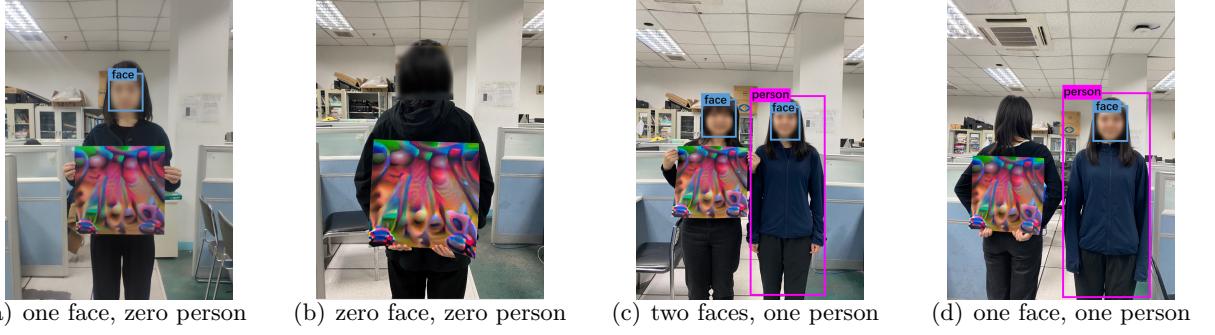


Figure 12: A weak content semantics structure.

patches without caring how they are generated.

In addition to the content semantics consistency, there are also other types of semantics structures. Some of them can contribute to develop a high-performance detection algorithm. For example, we can directly count the number of the human face in the input and compare it with the count of detected person objects in the person detection task. If the two numbers are different, we can conclude that there is an adversarial attack. The face counting can be accomplished in real time with the modern face recognition systems, e.g., ArcFace [14]. Accordingly, the adversarial example detection can also be exceptionally efficient. Unfortunately, the approach is not robust enough and cannot be adopted in practice. As illustrated in Figure 12, the attackers can stick the adversarial patch on the back and turn their back to the camera. In this way, they can easily fool the person detection and avoid introducing abnormal face-person counts. We suggest that the robustness of a content semantics structure should be carefully investigated when employing it as a detection criterion.

### 5.3 The Region Growing Algorithm

As described in Algorithm 1, the detection mainly involves seed selection, growth direction determination, and region update. The region growing algorithm will terminate when an inconsistency is monitored or the region has grown to include all the blocks.

**Seed Selection.** When an input image is fed to YOLO, it is divided into 361 ( $19 \times 19$ ) blocks for extracting features. YOLO will assign each block a classification probability in the output layer. One of the blocks will be used as the growing seed, i.e., the initial growing region. We employ a heuristic strategy to select the seed block to speed up the detection. The block with the highest *Person* class probability is chosen as the growing seed  $Seed_g$ . This may allow the model to identify the potential person object as soon as possible.

**Growth Direction Determination.** At each growing step, the region is extended to include more blocks following a chosen direction. The growth direction is determined mainly based on the blocks’ semantics.

If there is not a detected person object in the current region, it is desired that the region is enlarged to merge the blocks with the highest *Person* class probabilities. The probability of the blocks in the four candidate directions (i.e., up, down, left, and right) are summed respectively. The direction with the highest total probability is determined to be the growth direction.

If a person object has been detected in the current region, we hope the region can grow to encompass the potential adversarial patch as quickly as possible to trigger the abnormal detection transition. The image *entropy* is employed to rank candidate directions. The intuition here is that an adversarial patch can fool an object detector only if it hold enough “energy”. Consequently, the patch must be able to introduce a great deal of information in its limited space. In other words, the patch often has higher complexity and information density than the normal input. It is natural to use the entropy to seek the promising direction for touching the

---

**Algorithm 1** Region Growing for Adversarial Patched Examples Detection

---

**Input:**  $I$ : An input image;  $\theta$ : The gain threshold.  
**Output:**  $\varphi$ : Indicating if an adversarial object exists.

```

1:  $Blocks = \text{IMAGEPARTITION}(I)$ ;  $\varphi = False$ 
2:  $BProbabilities = \{\text{YOLO}_{\text{Person}}(b) \mid b \in Blocks\}$ 
3:  $BEntropies = \{H_{2d}(b) \mid b \in Blocks\}$ 
4:  $Region_g = Seed_g = \arg \max_b BProbabilities[b]$ 
5:  $Dir = [up, down, left, right]$ 
6:  $Blocks = Blocks - Seed_g$ 
7:  $prob = \text{YOLO}_{\text{Person}}(Region_g)$ 
8: while  $\varphi == False$  and  $Blocks \neq \emptyset$  do
9:    $S = [0, 0, 0, 0]$ ;  $O = 0$ ;  $Gain = 0$ 
10:   $\forall d \in DIR, DBlocks[d] = \text{BLKSINDIR}(Region_g, d)$ 
11:  if  $prob > Person_{\text{YOLO}}$  then
12:     $\forall d \in Dir, S[d] = \sum_{b \in DBlocks[d]} BEntropies[b]$ 
13:  else
14:     $\forall d \in Dir, S[d] = \sum_{b \in DBlocks[d]} BProbabilities[b]$ 
15:  end if
16:   $d_g = \arg \max_d S[d]$ 
17:   $Region_s = Region_g$ ;  $Region_n = \emptyset$ ;  $G = 0$ 
18:  while  $Gain \leq threshold$  do
19:     $Region_{adj} = \text{ADJBLKSINDIR}(Region_s, d_g)$ 
20:    if  $prob > Person_{\text{YOLO}}$  then
21:       $G = G + \sum_{b \in Region_{adj}} BEntropies[b]$ 
22:    else
23:       $G = G + \sum_{b \in Region_{adj}} BProbabilities[b]$ 
24:    end if
25:     $Region_n = Region_n \cup Region_{adj}$ ;  $Gain = \frac{G}{S[d_g]}$ 
26:    if  $Gain > \theta$  or  $Region_{adj} == \emptyset$  then
27:      break
28:    else
29:       $Region_s = Region_{adj}$ 
30:    end if
31:  end while
32:   $Region_g = Region_g \cup Region_n$ 
33:   $prob' = \text{YOLO}_{\text{Person}}(Region_g)$ 
34:   $\varphi = prob > Person_{\text{YOLO}} \wedge prob' \leq Person_{\text{YOLO}}$ 
35:   $Blocks = Blocks - Region_n$ 
36:   $prob = prob'$ 
37: end while
```

---

adversarial patch. We employ the two-dimensional entropy (2-D entropy) [1] to measure the complexity of an image block.

For each pixel in an image block with 256 pixel levels ( $0 \sim 255$ ), the average pixel value of the 1-order neighborhood is first calculated. This forms a pair  $(i, j)$ , the pixel value  $i$  and the average of the neighborhood  $j$ . The frequency of the pair is denoted as  $f_{ij}$ , and a joint probability mass function  $p_{ij}$  is calculated as Equation (7), where  $H$ ,  $W$  is the height and the width of the block. On the basis, the 2-D entropy of the block can be computed as Equation (8). Eventually, the 2-D entropy of a RGB color block is the average of its three color planes' entropies, which are computed individually. In a similar way, the 2-D entropies of

Table 1: The detection results.

Data Set	#Adv	#Benign	Accuracy	Avg. Time(s)
$ADV_1$	48	2	96.0%	1.52
$ADV_2$	47	3	94.0%	1.53
$BENIGN$	0	100	100%	1.52
Total	95	105	97.5%	1.52



Figure 13: Indoor adversarial T-shirt attack is detected.

the blocks in a candidate direction are summed. The one with the highest 2-D entropy will be chosen as the next growth direction.

$$p_{ij} = \frac{f_{ij}}{H \cdot W}, i \in [0, 255], j \in [0, 255] \quad (7)$$

$$H_{2d} = - \sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij} \log_2(p_{ij}) \quad (8)$$

**Region Update.** After determining the growth direction, we introduce the idea of *information gain* to compute how many blocks should be merged into the region. The *gain* resulting from merging neighbour candidate blocks into the current region is measured by computing the *Person* class probability or 2-D entropy. If the gain is greater than a predefined threshold, we believe that the current growth step has introduced enough content semantics. The candidate blocks will be merged and a new region is produced. Otherwise, more candidate blocks will be taken into consideration. In essence, the gain threshold determines the region growing speed. In our experiment, it is set to 5% with a small-scale empirical study.

Note that the proposed method can also be adopted for the multi-person scenario by introducing multiple growing seeds. Accordingly, the abnormal detection state transition can be specified as “( $n$  detected)  $\Rightarrow$  ( $< n$  detected)”, i.e., the number of the detected person objects becomes small.

## 5.4 Evaluation

We evaluate our detection method on a data set, which consists of 100 adversarial examples ( $ADV$ ) and 100 benign ones ( $BENIGN$ ). Among the adversarial examples, 50 ( $ADV_1$ ) are generated with the original attack technique [49], and the rest ( $ADV_2$ ) are produced with the improved algorithm mentioned in Section 4.

Table 1 shows the experiment results on the data set, in which #Adv and #Benign indicate the number of reported adversarial and benign examples, respectively. From the results, we can see that the proposed method can successfully detect 95% adversarial examples without



Figure 14: Outdoor adversarial T-shirt attack is detected.



Figure 15: The attack against Faster R-CNN is detected.

false positives. Only five evade the detection. The detection overhead is about 1.5 seconds per example.

To further evaluate the generalization of our method, we apply it to counter an emerging attack technique, *Adversarial T-Shirts* [56]. Using the technique can get robust physical adversarial examples for evading person detectors. The generated adversarial patch can be directly printed on the T-shirt, which can even withstand non-rigid deformation resulting from person moving.

The adversarial examples provided by the authors are directly used as the test cases, involving two representative scenarios (indoor and outdoor). We randomly choose ten adversarial examples for each scenario respectively. Without any fine tuning, our detection method can successfully catch all the 20 adversarial examples. As shown in Figure 13 and Figure 14, the abnormal detection state transition “*two detected*  $\Rightarrow$  *one detected*” can be effectively detected during the region growing. It should be emphasized that the above detection test is totally a blind experiment. We do not leverage any attack-specific knowledge and adjust any parameters in the detection. We confirm that our method can be applied to defend unknown adversarial patch attacks.

Besides YOLO, we have also evaluated our method on another popular object detector Faster R-CNN [38]. As presented in Figure 15, the semantics inconsistency can be detected when the region grows, showing that the proposed method is also transparent to the back-end object detection model, not only the attack technique.

## 6 Discussion

As mentioned above, the key of enforcing the signature-based defense is to find discriminating signatures. An important problem is how to identify the leverageable signature as soon as possible when facing an unknown attacks. Using the differential analysis to investigate the benign and adversarial example may be a feasible way, and the visualization technique can be

used as an observation tool to discover the interesting difference easily. Note that the unknown attacks can be caught by using the signature-independent detection.

In general, the signature-independent method is relatively slow compared with the signature-based one. However, it is also totally acceptable in some non-critical cases. We can limit the number of examples needed to be detected via sampling the input frames, especially when the signature-independent method is used to catch the unknown attack examples. Furthermore, the detection based on region growing can be implemented in parallel when we can equip more than one back-end object detectors. We can design an aggressive growing strategy to obtain multiple candidate regions simultaneously.

Huang et al. [21] propose a patch attack method named Universal Physical Camouflage Attack (UPC). It can fool object detectors to misidentify an object as a wrong category with multiple patches, such as detecting a person as a dog. Our method are not specially developed to defend the camouflage attack. However, our signature-independent method can be applied to detect the attack. In fact, as demonstrated in Figure 16, we can detect the patched camouflage example with the semantics inconsistency: the attack object is identified as a certain class locally but another one globally (*a person locally but a dog globally*).

It is illustrated that our idea can also be leveraged to counter other types of patch attacks. We believe this study actually provides a general methodology for detecting local perturbation attacks. For example, the adversarial patch targeting the image classification [8] can mislead the classifier to ignore the certain items (e.g., a banana) within the image, and report another class (e.g., a bowl). For detecting the attack, we can determine the seed region and growing directions according to the given class of interest (e.g., the *Banana* class), and check whether there is such a semantics inconsistency: *a banana locally but another globally*, as shown in Appendix B.

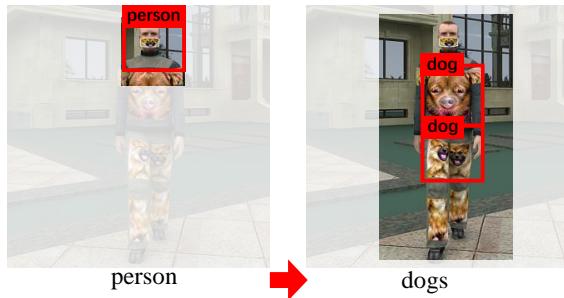


Figure 16: The UPC attack is detected.

## 7 Related Work

Many existing studies have paid much attention to the security of artificial intelligence systems. The arm race between attackers and defenders will never end.

### 7.1 Attacks

Existing attack studies mainly target at computer vision tasks, e.g., image classification. Szegedy et al. [48] propose the idea of generating adversarial examples with imperceptible perturbations to mislead the DNN-based image classifiers. Goodfellow et al. [19] propose FGSM to generate the adversarial examples. Some improved FGSM methods are also proposed, such as IFGSM [24] and PI-FGSM [16]. Carlini and Wagner [11] design an optimization algorithm to generate the adversarial examples with the least perturbations.

Some studies focus on launching attacks in the physical world. Mahmood et al. [41] evade the face recognition in the physical world by wearing a pair of printed adversarial eyeglasses. In their later work [42], they propose the Adversarial Generative Nets to train a generator

neural network for emitting adversarial examples that satisfy desired objectives. Papernot et al. [33] propose a black-box attack against the road sign classifier by introducing some precise perturbations. Eykholt et al. [15] propose the Robust Physical Perturbations algorithm to attack the road sign classifiers under different physical conditions. Brown et al. [8] create adversarial patches to fool image classifiers in the physical world.

The adversarial patch is mainly used to attack the object detection. Song et al. [44] implement an attack to cause the stop sign disappears with adversarial patches in front of YOLO. Thys et al. [49] generate a printable adversarial patch to fool YOLO. Xu et al. [56] and Wu et al. [52] print the adversarial patch on the T-shirt, which can successfully attack the person detectors in physical world. Komkov et al. [23] generate the adversarial hat sticker to hide the person from the face recognition. In this study, we focus on defending this kind of attack for the sensitiveness of person detection.

Huang et al. [21] propose UPC to fool the object detector. We believe it can be detected with our signature-independent method, as discussed in Section 6.

Attacks to the 3-D object detection systems have also been explored. Cao et al. [9] propose a gradient-based approach to generating 3D-printed adversarial objects to evade LiDAR on autonomous vehicles. Xiao et al. [55] present meshAdv to generate adversarial meshes to manipulate the shape or texture of a 3-D object and mislead the classifiers and object detectors. Sun et al. [46] explore the vulnerability in the LiDAR-based object detection models and perform corresponding adversarial attacks to spoof the LiDAR sensors. In the future, we plan to research how to transfer our methods to detect the 3-D attacks. We believe there is also the same semantics inconsistency in 3-D adversarial examples, i.e., the adversarial object can appear locally but disappear globally.

Except computer vision, there are some attacks against other tasks. N. Srndic et al. [45] craft malformed PDF files to evade an online learning-based system for detecting PDF malware. Liang et al. [28] propose an attack way to crack and evade the Google phishing pages classification, which is a linear text classifier. They also develop an adversarial example generation method to fool the text classifiers based on deep learning [26].

## 7.2 Defense

A lot of techniques have been proposed to detect adversarial examples. Li et al. [25] utilize the statistical features to distinguish the adversarial examples from the benign. Amirian et al. [3] visualize the CNN features, track the adversarial perturbations and detect adversarial images based on statistical analysis of the feature responses. Liang et al. [27] denoise the adversarial examples and compare the classification results for adversary detection. Xiao et al. [54] detect adversarial examples by checking the spatial inconsistency after applying two randomly selected patches to the given image. Lu et al. [30] propose SafetyNet to detect unreal scenes in an RGBD image by checking whether the image is consistent with its depth map. While different kinds of inconsistency have been leveraged for adversary detection, in this paper, we focus on the semantics inconsistency that can be captured from the given input and robust enough to detect adversarial patched examples.

Besides detecting adversarial images, efforts have been done to detect adversaries in other areas. Sun et al. [46] detect spoofed fake vehicles by checking the violations of physics, i.e., anomalously sensed free space and laser penetration. Choi et al. [13] identify the physical attacks against robotic vehicles (RVs) by deriving and monitoring RVs' control invariants. Aghakhani et al. [2] propose FakeGAN to detect deceptive customer reviews.

Improving the robustness of the DNNs has also been studied. Biggio et al. [6] provide a complete understanding of the classifier's behavior about adversarial examples, helping to improve the DNN's robustness with better design choices. Lyu et al. [31] improve the robustness of the DNN against adversarial examples by considering the worst situation of perturbation and the gradient constraint in the model. Ba et al. [4] and Hinton et al. [20] extract the knowl-

edge from the DNN, with which they improve the resilience against the adversarial examples. Papernot et al. [34] smooth the DNN model to reduce the enlargement of perturbations during feature extraction and make the model robust to defend small adversarial perturbations. Sharif et al. [40] suggest that other metrics like the structural similarity [51] should be considered together with L<sub>p</sub>-forms to improve the model’s robustness. Quiring et al. [35] reconstruct the image pixels to reveal adversarial pixels and filtering them out for classification correctness. Chen et al. [12] build a robust PDF malware classifier with a new distance metric of operating on the PDF tree structure.

Recently, Xiang et al. [53] propose a novel method PatchGuard to defend the adversarial patch attacks to the image classification. A masking algorithm is introduced in feature aggregation of the model network to remove the potential corrupted features with abnormally large values, which can result in misclassification. PatchGuard targets to enhance the CNN-based image classification, and cannot be directly adopted for the object detection. In fact, the object detection attack method aims to lower the object confidence, instead of the class probability protected by PatchGuard. Besides, our two methods are transparent to the target models and do not require any modification of them.

## 8 Conclusion

To defend the adversarial patch attacks aiming the object detection, in this paper, we present two different defense methods, one is signature-based and the other is signature-independent. At first, we find that many adversarial patch pixels possess a discriminating characteristic. Accordingly, a defense filter is developed to exclude these patch pixels from the input such that the filtered adversarial patch loses its attack efficacy. This signature-based defense is demonstrated to be effective and efficient for the existing attack. However, it is also proven to be ineffective to an improved attack algorithm. In order to provide robust defense to potential attacks, we propose a general signature-independent detection method. The method is based on an insightful observation that the local semantics of an adversarial patched example is inconsistent with its local semantics. The signature-independent detection is proven to be effective to known and unknown attacks, although it is relatively slow. Essentially, the two defenses are complementary to each other and can be used for different scenarios, or be combined to get a comprehensive protection.

## References

- [1] Ahmed S Abutaleb. Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing*, 47(1):22–32, 1989.
- [2] Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Kruegel, and Giovanni Vigna. Detecting deceptive reviews using generative adversarial networks. In *2018 IEEE Security and Privacy Workshops*, pages 89–95. IEEE, 2018.
- [3] Mohammadreza Amirian, Friedhelm Schwenker, and Thilo Stadelmann. Trace and detect adversarial attacks on cnns using feature response maps. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 346–358. Springer, 2018.
- [4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.
- [5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013.

- [6] Battista Biggio, Giorgio Fumera, and Fabio Roli. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):984–996, 2014.
- [7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.
- [8] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.
- [9] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *CoRR*, abs/1907.05418, 2019.
- [10] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium*, pages 513–530. USENIX Association, 2016.
- [11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.
- [12] Yizheng Chen, Shiqi Wang, Dongdong She, and Suman Jana. On training robust PDF malware classifiers. In *29th USENIX Security Symposium*, pages 2343–2360. USENIX Association, 2020.
- [13] Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Deng. Detecting attacks against robotic vehicles: A control invariant approach. In *2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 801–816, 2018.
- [14] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4685–4694, 2019.
- [15] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.
- [16] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *European Conference on Computer Vision*, pages 307–322. Springer, 2020.
- [17] Ross Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.
- [18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- [20] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

- [21] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020.
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- [23] Stepan Komkov and Aleksandr Petushko. Advhat: Real-world adversarial attack on arcface face id system. In *25th International Conference on Pattern Recognition*, pages 819–826. IEEE, 2021.
- [24] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations*, 2017.
- [25] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *IEEE International Conference on Computer Vision*, pages 5764–5772, 2017.
- [26] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *27th International Joint Conference on Artificial Intelligence*, pages 4208–4215, 2018.
- [27] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Trans. Dependable Secur. Comput.*, 18(1):72–85, 2021.
- [28] Bin Liang, Miaoqiang Su, Wei You, Wenchang Shi, and Gang Yang. Cracking classifiers for evasion: a case study on the google’s phishing pages filter. In *25th International Conference on World Wide Web*, pages 345–356, 2016.
- [29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [30] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *IEEE International Conference on Computer Vision*, pages 446–454, 2017.
- [31] Chunhuan Lyu, Kaizhu Huang, and Hai-Ning Liang. A unified gradient regularization family for adversarial examples. In *IEEE international conference on data mining*, pages 301–309. IEEE, 2015.
- [32] Mounir Omari, Abdelkaher Ait Abdelouahad, Mohammed El Hassouni, and Hocine Cherifi. Color image quality assessment measure using multivariate generalized gaussian distribution. In *International Conference on Signal-Image Technology & Internet-Based Systems*, pages 195–200. IEEE, 2013.
- [33] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.
- [34] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

- [35] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *29th USENIX Security Symposium*, pages 1363–1380. USENIX Association, 2020.
- [36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
- [37] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.
- [38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(6):1137–1149, 2017.
- [39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2020.
- [40] Mahmood Sharif, Lujo Bauer, and Michael K Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1605–1613, 2018.
- [41] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, page 1528–1540. ACM, 2016.
- [42] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. A general framework for adversarial examples with objectives. *ACM Trans. Priv. Secur.*, 22(3), June 2019.
- [43] Chunhe Song and Xiaodong Lin. Natural image splicing detection based on defocus blur at edges. In *IEEE/CIC International Conference on Communications in China*, pages 225–230. IEEE, 2014.
- [44] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies*. USENIX Association, 2018.
- [45] Nedim Srndic and Pavel Laskov. Practical evasion of a learning-based classifier: A case study. In *IEEE Symposium on Security and Privacy*, pages 197–211, 2014.
- [46] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium*, pages 877–894, 2020.
- [47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [48] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

- [49] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2019.
- [50] Geert Verdoolaege and Paul Scheunders. Geodesics on the manifold of multivariate generalized gaussian distributions with an application to multicomponent texture discrimination. *International Journal of Computer Vision*, 95(3):265–286, 2011.
- [51] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [52] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020.
- [53] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium*, 2021.
- [54] Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, September 2018.
- [55] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019.
- [56] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020.
- [57] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929. IEEE Computer Society, 2016.

## A High-resolution Adversarial Patches

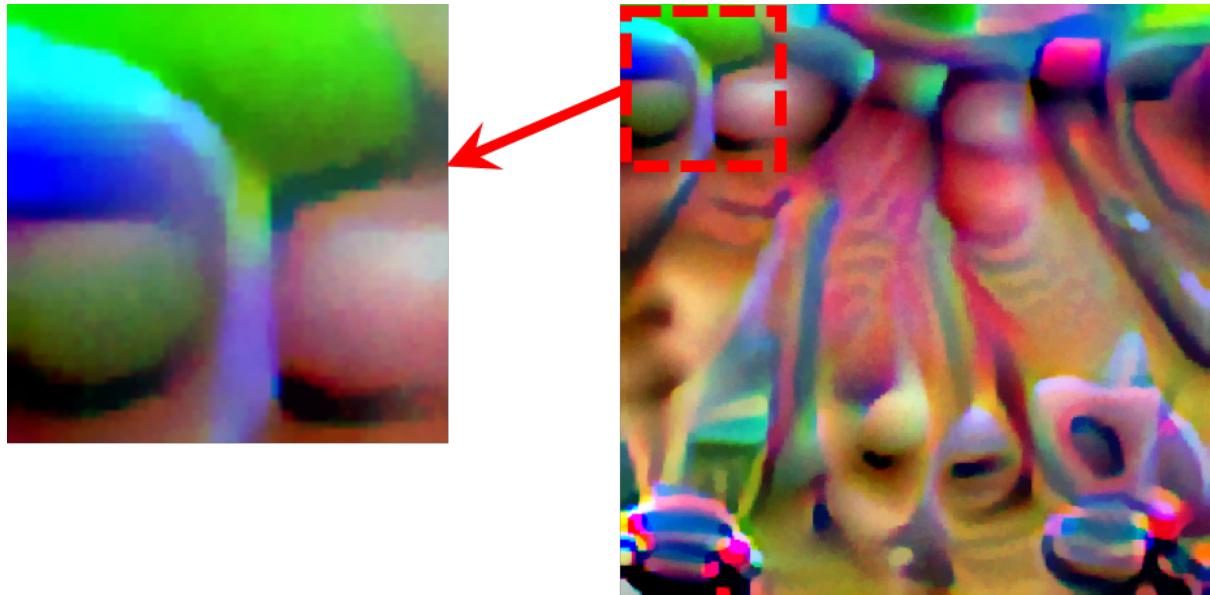


Figure 17: The original adversarial patch.

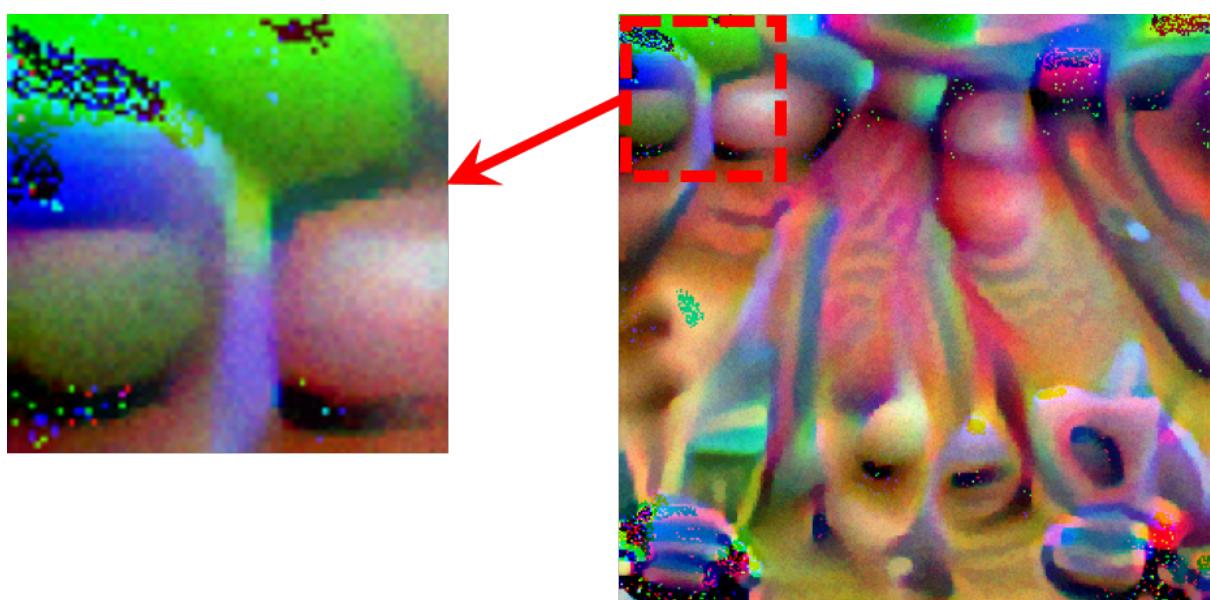


Figure 18: The adjusted adversarial patch.

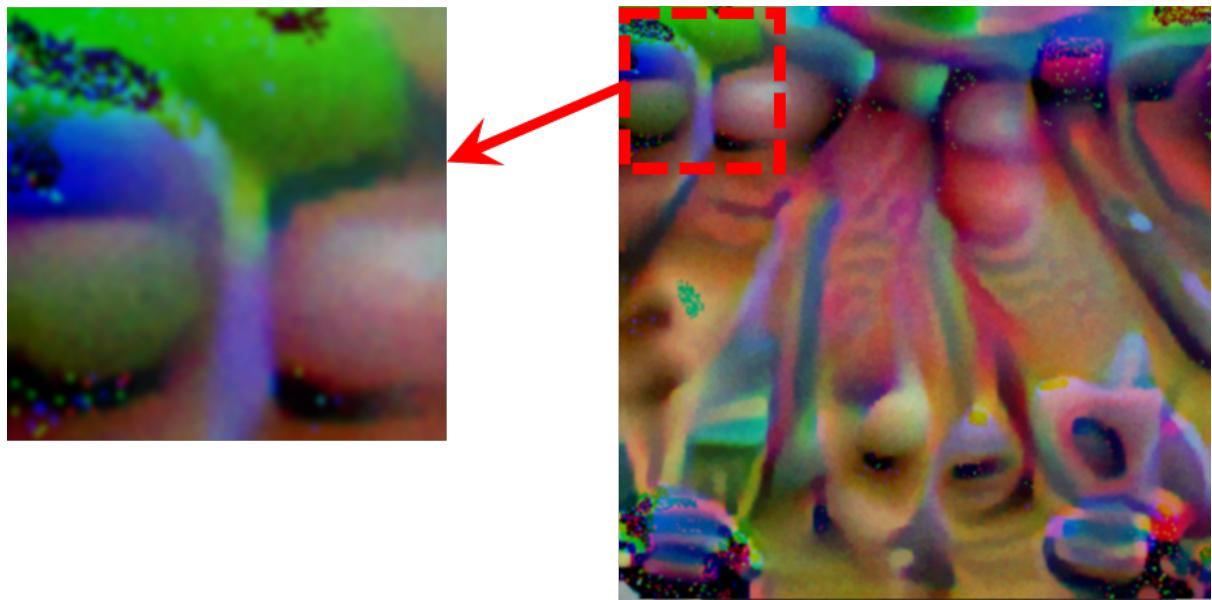


Figure 19: The simulated adjusted adversarial patch.

## B Adversarial Patch against Image Classification

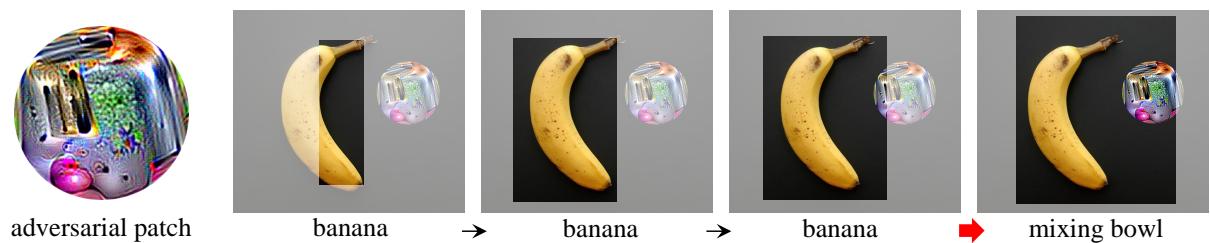


Figure 20: The content semantics inconsistency in the adversarial patched example targeting the image classification [8]: *a banana locally but a bowl globally.*