PAD: Patch-Agnostic Defense against Adversarial Patch Attacks

Lihua Jing[1,2], Rui Wang[1,2*], Wenqi Ren[3], Xin Dong[1,2], Cong Zou[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University
{jinglihua, wangrui, dongxin, zoucong}@iie.ac.cn, renwq3@mail.sysu.edu.cn

Abstract

Adversarial patch attacks present a significant threat to real-world object detectors due to their practical feasibility. Existing defense methods, which rely on attack data or prior knowledge, struggle to effectively address a wide range of adversarial patches. In this paper, we show two inherent characteristics of adversarial patches, semantic independence and spatial heterogeneity, independent of their appearance, shape, size, quantity, and location. Semantic independence indicates that adversarial patches operate autonomously within their semantic context, while spatial heterogeneity manifests as distinct image quality of the patch area that differs from original clean image due to the independent generation process. Based on these observations, we propose PAD, a novel adversarial patch localization and removal method that does not require prior knowledge or additional training. PAD offers patch-agnostic defense against various adversarial patches, compatible with any pre-trained object detectors. Our comprehensive digital and physical experiments involving diverse patch types, such as localized noise, printable, and naturalistic patches, exhibit notable improvements over state-of-the-art works. Our code is available at https://github.com/Lihua-Jing/PAD.

1. Introduction

Adversarial attacks substantially challenge the security of object detectors, leading to potentially severe consequences in various fields (*e.g.*, autonomous driving). Traditional adversarial attacks typically involve adding perturbations to the entire image. However, modifying every pixel is unrealistic in real-world attack scenarios. Adversarial patch attacks, on the other hand, focus on introducing disturbances in a limited area. Their practical feasibility makes them one of the most threatening forms of adversarial attacks.

Defenses against adversarial patch attacks on object detectors can be broadly categorized into three main types: *i)* modifying or intervening within detection models [21, 23, 39], *ii)* locating and eliminating adversarial patch regions in images[7, 10, 28, 33, 42], and *iii)* certifiably robust defenses [46, 50]. Among these, methods falling under the second category, which act as preprocessing, offer the broadest range of applications.

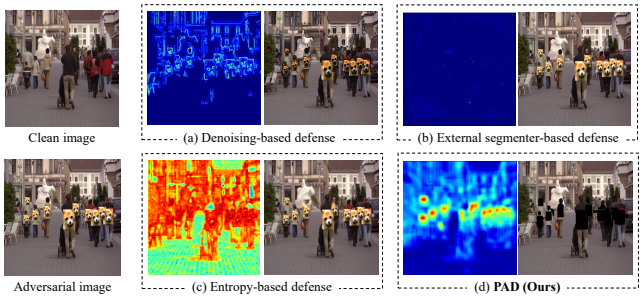


Figure 1. When attacked by natural-looking adversarial patches, (a) locates high-frequency areas, and eliminates edge lines instead of patches [33]; (b) fails to detect the existence of patches since no such patch data in the training set [28]; (c) produces a heat map where the patch area and background are difficult to distinguish, failing to locate the patches [42]. Our proposed PAD achieves accurate patch location and removal.

Researchers have explored various patch localization methods to effectively remove adversarial patches from images. Denoising-based defenses [33] smooth out noise-like regions in images, providing an effective defense against early localized noise patches. However, they fail to address natural-looking patches. External segmenter-based defenses [10, 28] train an adversarial patch segmentation model for patch localization, using adversarial images generated by existing attack techniques. However, the reliance on training data makes them ineffective against unseen patch types. Entropy-based defenses [7, 42] achieve patch localization by identifying high entropy kernels and patch shape reconstruction. Nevertheless, the entropy threshold setting requires prior knowledge of the distribution for clean data and patches, and shape reconstruction relies on training data, posing challenges in practical applications. Despite progress in certain aspects, these methods face a common challenge of locating various adversarial patches without relying on prior attack knowledge. As shown in Figure 1, these three categories of methods fail to effectively remove patch areas.

In this paper, we propose a new approach for various adversarial patch localization without relying on prior attack

*Corresponding author

knowledge (*e.g.,* appearance, shape, size, or quantity). The proposed approach is derived from two inherent characteristics of adversarial patches: semantic independence and spatial heterogeneity.

Semantic independence implies zero information gain from the surrounding semantic space, while spatial heterogeneity refers to inconsistent image quality in the same space introduced by adversarial patches. In both digital and physical attacks, the adversarial patch, added as a separate component to the image or environment, is semantically independent in the image. The surroundings provide no information about the content of adversarial patches, and vice versa. Additionally, different imaging devices, generation processes, and compression methods may lead to variations in image quality. With a source different from the original clean data, the quality of patch regions exhibits heterogeneity compared to other areas in space.

Based on these observations, we propose to identify patch areas by quantifying local semantic independence and spatial heterogeneity. We measure the information gain between adjacent regions based on mutual information, and evaluate residuals from recompression at different quality factors to address the unequal impact upon areas of varying quality. While complex backgrounds confuse entropy-based methods [7, 42] since the high information density caused by complicated textures, our method exhibits more robustness as semantic correlations remain between adjacent background areas. In addition, to eliminate the reliance on training data, we present a patch localization and removal pipeline that requires no prior knowledge or additional training. Different from current works, our defense accurately identifies the patch region mask without any reference to existing adversarial images and imposes no limitations on the quantity or proportion of patches in the image.

Our contributions can be summarized as follows:

- We reveal two inherent characteristics of adversarial patches, semantic independence and spatial heterogeneity, and propose patch locating based on mutual information and recompression, which is agnostic to patch appearance, shape, size, location, and quantity.
- We propose a patch-agnostic defense (PAD) method for adversarial patch localization and removal, which requires no prior attack knowledge or additional training and is compatible with any object detector.
- We conduct experiments on adversarial patches with different appearances, shapes, sizes, locations, and quantities, evaluating the defense effectiveness of PAD in both digital and physical scenes. Experimental results demonstrate our superior defense performance compared to the current state-of-the-art methods.

2. Related Work

2.1. Adversarial Patch Attacks

The concept of adversarial patch attacks is first introduced by [6]. They develop a generic patch capable of deceiving image classifiers and demonstrate the feasibility of physical attacks by attaching the patch in real-world scenarios. Early research on adversarial patch attacks primarily focuses on localized noise [22, 29]. DPatch [29] is the pioneering work on patch attacks specifically designed for object detection, targeting both bounding box regression and object classification components of the detection system. PatchAttack [53] proposes a reinforcement learning-based attack method to induce misclassification by superimposing small texture patches on the input image. Some work focus on physical attacks [8, 43, 45, 51, 56]. [56] and [8] attach patches to traffic signs, leading to the misidentification of those signs. [43] proposes a printable adversarial patch for pedestrian detection, introducing non-printable loss in the optimization process. [51] and [45] explore the integration of adversarial patches into wearable clothing. [17], [25] and [41] train generative adversarial networks (GANs) to generate natural-looking patches that match the visual properties of normal images.

2.2. Defenses against Patch Attacks

Adversarial training [13, 20, 32, 34, 40, 44, 55], which enhances model robustness by adding adversarial examples during training, is one of the most popular and effective defenses against digital attacks. However, such methods are not suitable for defending pre-trained models already in use and require significant resources for retraining when new attacks emerge, making them not so practical.

Some defense methods involve modification of specific models [21, 23, 39, 54]. [39] investigates the use of spatial context constraints in YOLOv2 [35] to enhance defense robustness against adversarial patches. [21] introduces a patch class into YOLOv2, enabling the detection of objects of interest as well as adversarial patches. [23] proposes adversarial patch feature energy(APE), and defense is achieved by incorporating an APE discovery and suppression module into the network. Although good defense effects can be achieved on specific detection models, they cannot directly provide defense for a wide range of object detectors.

To provide more general defense, researchers have explored locating patch areas in images and eliminating their effects [7, 9, 10, 14, 28, 33, 38, 42, 52]. Since early adversarial patches are usually in the form of localized noise, some defense methods focus on reducing the impact of noise-like areas in input images. LGS [33] observes that patch attacks introduce concentrated high-frequency noise and proposes gradient smoothing for regions with significant gradients. APM [10] and SAC [28] train external segmentation networks to locate noise-like regions. While these methods effectively defend against localized noise-based patch attacks, they struggle to counter the new types of natural-looking patches. DW [14] and Jujutsu [9] utilize saliency maps to identify patch areas and cover them to mitigate their impact on classification. In object detection tasks, which involve bounding box regression in addition to classification, accurately localizing patches becomes challenging us-

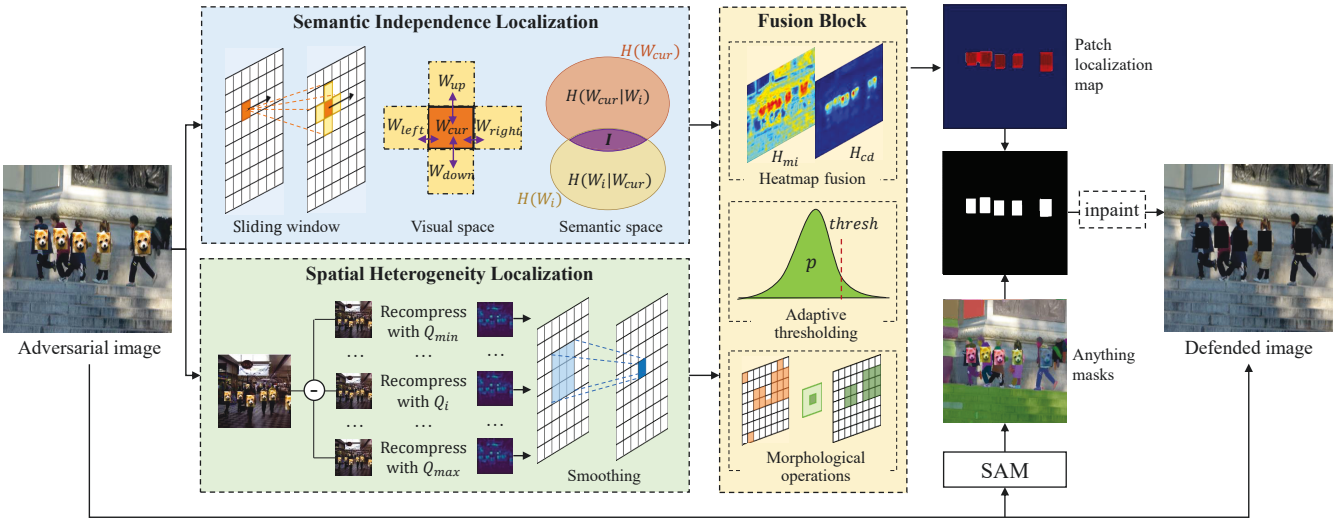


Figure 2. Overview of our proposed PAD. Semantic Independence Localization and Spatial Heterogeneity Localization find patch regions from two views, generating heat maps feeding into Fusion Block. The patch localization map output by Fusion Block is then matched with all masks from SAM, getting more accurate patch boundaries. Feeding the defended image into object detectors for robust prediction.

ing saliency maps. Jedi [42] and [7] use entropy to locate patch areas, but prior knowledge of entropy distribution of clean dataset and patch area is required.

In recent years, some researchers have proposed certifiably robust defenses against adversarial patches [46–50]. DetectorGuard [46] is an attack detection defense that raises an alert when an attack is detected without removing the adversarial patches, resulting in a loss of model functionality during an attack. ObjectSeeker [50] requires ensuring that, under at least one partition, the remaining images do not contain any adversarial patch pixels. As a result, there will be trouble handling attack scenarios with large patch proportions or multiple patches.

Different from these methods, PAD is derived from two general characteristics of patches that are independent of their appearance, shape, size, quantity, and location, allowing us to eliminate various patches without relying on prior attack knowledge.

3. Preliminaries

Differing from traditional adversarial attacks, adversarial patch attacks impose restrictions on the attacker, limiting the area where perturbations can be introduced. Within this constraint, the attacker has the flexibility to manipulate the pixels within the designated patch region.

We denote a clean image with dimensions $w \times h \times c$ as $X \in \mathbb{R}^{w \times h \times c}$. The adversarial patch, denoted as P_{adv}, can take any shape. The generation of P_{adv} is typically controlled by the loss function L_{patch}, which varies depending on the specific attack objective. Since our goal is to defend against various patch attacks, irrespective of their intention to conceal the target object, misclassify it, or generate false detections of non-existent objects, we do not make assumptions about L_{patch}.

The resulting adversarial image $X_{adv} \in \mathbb{R}^{w \times h \times c}$, produced by attack techniques, can be expressed as follows:

$$X_{adv} = M_{patch} \odot A(P_{adv}, X, l, t) + (1 - M_{patch}) \odot X, \quad (1)$$

where $M_{patch} \in \{0,1\}^{w \times h}$ represents the patch region in image X, with elements set to 1 within P_{adv} and 0 elsewhere. $A(P_{adv}, X, l, t)$ denotes the patch application function, incorporating patch transformations such as scaling and rotation denoted by t, and patch location denoted by l. $\odot$ refers to element-wise multiplication. In the case of attack methods that can be used physically, $A(P_{adv}, X, l, t)$ typically involves completely replacing the image area at position l with the transformed patch.

4. Method

4.1. Defense Pipeline

In this section, we introduce the pipeline of PAD, as shown in Figure 2. Firstly, we analyze the input image using the two inherent characteristics that all adversarial patches possess to obtain heat maps, H_{mi} and H_{cd}, which highlight the regions in the image that exhibit semantic independence and spatial heterogeneity, respectively. Next, we employ the Fusion block to merge H_{mi} and H_{cd}, generating the patch localization map H_p that accurately reflects the areas in the image that possess both of these characteristics.

To make the patch masks more accurate, we introduce the Segment Anything Model (SAM) [24]. Different from the segmentation models introduced in prior methods, since we do not need it to have recognition capabilities for adversarial patches, SAM can be replaced with any pre-trained segmentation models with similar capabilities, without additional training. In other words, we do not rely on known patch attack methods to generate adversarial images for training, preventing our defense from losing effectiveness when encountering new attack methods. With SAM's zero-shot segmentation capability, we segment the edges of all regions in

the image and obtain masks for each region. We then match each mask with H_p and consider all masks with Intersection over Area (IoA) greater than threshold t_m as the final patch masks. The calculation of IoA can be stated as follows:

$$IoA(mask, H_p) = \frac{area(mask \bigcap H_p)}{area(mask)}. \quad (2)$$

If the localization of adversarial patches is accurate enough, the removal process only needs to be able to eliminate the impact of the patches. Therefore, we employ a simple and fast inpainting method that is commonly used in previous works [10, 28]: filling the patch area with all black pixels. We also compare the coherence transport-based inpainting method [4, 42] with all-black, more details can be found in supplementary material.

4.2. Semantic Independence Localization

Semantic independence evaluation. The semantic independence of a region can be measured by its semantic correlation with surrounding regions. The smaller the semantic correlation, the stronger the independence. For two adjacent regions, A and B, their semantic correlation can be defined as the information gain they provide to each other. Given knowledge about region A, it quantifies how much information can be inferred about region B, i.e., the reduction in uncertainty about B given knowledge of A. This can be expressed using mutual information:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A) \quad (3)$$

where $H(*)$ represents information entropy, and $H(*|*)$ represents conditional entropy.

Semantic independence localization based on mutual information. Building on the analysis above, we perform adversarial patch discovery by computing the semantic independence of local regions across the entire image. We set up a sliding window, and calculate the mutual information between each window and its four neighboring windows (up, down, left, right). The average value of these mutual information scores is used as the heat value for the current window, generating a heat map for the entire image. We use W_{cur} to denote the current window, and W_{up}, W_{down}, W_{left}, W_{right} represent the four neighboring windows of the same size respectively. For each i in $\{up, down, left, right\}$, the mutual information between W_i and W_{cur} can be expressed as follows:

$$I(W_i; W_{cur}) = \sum_{w_i \in W_i} \sum_{w_c \in W_{cur}} p(w_i, w_c) \log \frac{p(w_i, w_c)}{p(w_i)p(w_c)}. \quad (4)$$

The heat value within W_{cur} can be expressed as follows:

$$H_{mi}\left[x_{cur} : x_{cur} + d, y_{cur} : y_{cur} + d\right] = \frac{1}{n}\sum_{i=1}^{n} I(W_i; W_{cur}), \quad (5)$$

where(x_{cur}, y_{cur}) denotes the coordinates of the upper left corner of the window W_{cur}, H_{mi} denotes the heat map generated by Semantic Independence Localization module, d denotes the size of the sliding window, and n denotes the number of neighboring windows. n equals 4 for most windows, 2 or 3 for windows located at the edge of the image.

4.3. Spatial Heterogeneity Localization

Impact of compression on image quality. Image compression leverages the insensitivity to certain components of human eyes to reduce storage space. Taking the most commonly used JPEG compression as an example, after color space transformation, it undergoes block-based Discrete Cosine Transform (DCT) to the image, converting the spatial domain into the frequency domain. The transformed low-frequency components have larger values, mainly concentrated in the upper-left corner, while high-frequency components have smaller values, distributed in lower-right regions. Subsequently, the DCT coefficients are quantized so that smaller coefficients close to 0 completely become 0, and non-zero values also generate a large number of repetitions, thereby reducing the coding length. Using $F(x, y, i)$ to represent the DCT coefficient of channel i at location (x, y), the quantization process can be expressed as:

$$F_q(x, y, i) = round(\frac{F(x, y, i)}{Q(x, y, i)}), \quad (6)$$

where $Q(x, y, i)$ represents the corresponding quantization step size. A larger Q leads to greater quantization loss and poorer image quality.

Spatial heterogeneity localization through recompression. For an image containing regions of varying quality, compressing the entire image will affect each quality region differently, providing valuable clues for identifying abnormal regions in the image [3, 12, 15, 26, 30]. Inspired by this, we locate adversarial patches based on the quality differences during recompression. For a clean image with quality factor Q_c, the patch area with quality factor Q_p, we set different quality factors Q_r to re-compress the attacked image X_{adv}, and calculate the squared difference of pixel values before and after re-compression, as follows:

$$D(x, y, Q_r) = \frac{1}{c}\sum_{i=1}^{c} \left[f(x, y, i) - f_{Q_r}(x, y, i)\right]^2, \quad (7)$$

where c denotes the number of channels. When Q_r is close to Q_p, the D values of the patch region are minimized. When Q_r is close to Q_c, the D values of the uncovered region are minimized. To enhance the robustness to texture variations in the image, we apply convolutional smoothing and perform normalization.

4.4. Fusion Block

Section 4.2 and 4.3 aim to identify potential adversarial patch regions from the perspectives of semantic independence and spatial heterogeneity. In the fusion block,

we merge the heat maps obtained from these two methods. Additionally, to mitigate the influence of cluttered backgrounds, we apply adaptive thresholding and morphological operations to further process the fused results.

Heatmap fusion. Given the local mutual information heat map H_{mi} and the recompression difference heat map H_{cd} which have different value ranges, we first normalize them individually to scale the values into the range $[0, 255]$. The normalization process can be expressed as follows:

$$H^{'}_{mi}(x,y) = \frac{H_{mi}(x,y) - min(H_{mi})}{max(H_{mi}) - min(H_{mi})} \times 255, \quad (8)$$

$$H^{'}_{cd}(x,y) = \frac{H_{cd}(x,y) - min(H_{cd})}{max(H_{cd}) - min(H_{cd})} \times 255. \quad (9)$$

After normalization, we perform a weighted sum of $H^{'}_{mi}$ and $H^{'}_{cd}$ to obtain the fused heat map, $H^{'}_{fuse}$:

$$H_{fuse}(x,y) = r_{mi} \times H^{'}_{mi}(x,y) + (1 - r_{mi}) \times H^{'}_{cd}(x,y), \quad (10)$$

where $r_{mi} \in [0, 1]$ denotes the weight of mutual information heat map.

Adaptive thresholding. Since the heat values are significantly affected by the image content, using a static threshold may filter out adversarial patches or incorrectly treat other regions as adversarial patches, thus degrading the performance of the model. Therefore, we automatically set an adaptive threshold based on the distribution of the heat map for each image, and then set elements in H_{fuse} that are below the threshold to 0. The threshold here can be expressed as follows:

$$thresh = (1-j) \times Sort(H_{fuse})[i] + j \times Sort(H_{fuse})[i+1], \quad (11)$$

$$i = \lfloor (n-1) \times p \rfloor, j = (n-1) \times p - i, \quad (12)$$

where p is a fixed hyperparameter, n represents the number of elements in H_{fuse}, and $Sort(H_{fuse})$ represents sorted H_{fuse} in ascending order based on the heat values.

Morphological operations. To eliminate the interference of background with high heat values but unrelated to adversarial patches, we apply an OPEN-CLOSE-OPEN operation to the heat map after thresholding. The opening operation involves erosion followed by dilation and is mainly used to remove isolated small dots and bridges between different regions in the heat map. The closing operation involves dilation followed by erosion and is mainly used to fill in a few concave regions in the patch area that were filtered out by the threshold. The kernel size for the opening and closing operations is adaptively selected based on the image size, more details can be found in the supplementary material.

5. Defense Evaluation on Digital Attacks

5.1. Evaluation Settings

Target object detectors and dataset. In our experiments, we use Faster R-CNN [37] with a ResNet-50 [16] backbone, YOLOv2 [35], YOLOv3 [36], YOLOv5s [1] and YOLOv8n [2] as our target object detectors. All models are pre-trained on MS COCO [27]. Since most existing adversarial patch attacks that can be used physically are developed for pedestrian detectors [23], we mainly focus on the INRIA Person dataset [11] which consists of 614 person detection images for training and 288 for testing. Only test images are adopted since there is no training part in PAD. Experiments on other datasets can be found in supplementary material.

Adversarial patch attacks. To evaluate the defense effectiveness of PAD against different types of patches, we employ 11 distinct patches generated by DPatch [29], YOLO adversarial patch [43], and Naturalistic Patch [17], covering localized noise, printable, and natural-looking patches. DPatch generates a specific-sized patch (75×75 and 100×100 in our experiments) located in the upper left corner of each image, using 200 iterations with a learning rate of 0.01. The YOLO adversarial patch (P1-P6) and Naturalistic Patch (OBJ, OBJ-CLS, and Upper) generate multiple patches of varying sizes and positions based on the detectable pedestrians in the image. We also conduct defense experiments against two more attacks [18][19], the relevant results can be found in the supplementary material.

Implementation details. Throughout our experiments, we used fixed hyperparameter values for different patch types without any adjustments. We set r_{mi} to 0.5 in Heatmap fusion, which assigns equal weights to Semantic Independence Localization and Spatial Heterogeneity Localization. The value of p in Adaptive thresholding is set to 0.8. The IoA threshold t_m for mask matching is set to 0.5.

We compare PAD with four state-of-the-art adversarial patch defenses: LGS [33], SAC [28], Jedi [42], and ObjectSeeker [50], corresponding to denoising-based, external segmenter-based, entropy-based and certifiably robust defenses respectively. For LGS, we set the block size to 30, overlap to 5, threshold to 0.1, and smoothing factor to 2.3. For Jedi, due to the reliance on the prior entropy distribution values of the clean dataset and the patch region, using default parameters in code is less effective, we perform parameter tuning for some patches.

5.2. Overall Defense Performance

In object detection tasks, Average Precision (AP) is a widely used evaluation metric that assesses the area under the Precision-Recall Curve, representing the overall performance of a model. Therefore, we utilize mean Average Precision (mAP) at Intersection over Union (IoU) 0.5 to demonstrate the effectiveness of the attacks and defenses. We conduct experiments on different detectors, attacks, and defenses mentioned above and report the results in Table 1. Due to space limitations, it only shows results on Faster R-CNN, YOLOv3, and YOLOv5s. Results on YOLOv2 and YOLOv8n can be found in the supplementary material.

The results demonstrate that PAD achieves the best defense performance against various adversarial patch attacks on different detectors. For natural-looking patches (P1-P6)

Table 1. mAP(%) under different adversarial patch attacks. The best performance is **bolded**, and the suboptimal performance is underlined.

Detector	Defense	Clean	Localized Noise [29]		Printable Patch [43]			Natural-looking Patch [17]					
			DPatch 75×75	DPatch 100×100	OBJ	OBJ-CLS	Upper	P1	P2	P3	P4	P5	P6
Faster R-CNN	Undefended	*96.13*	*52.52*	*3.84*	*50.37*	*67.40*	*49.99*	*60.70*	*74.30*	*62.80*	*73.52*	*72.23*	*47.66*
	LGS (WACV19) [33]	96.01	95.96	96.06	75.34	80.10	79.57	61.34	73.69	75.06	79.60	74.03	58.66
	SAC (CVPR22) [28]	**96.13**	96.23	96.16	80.70	86.20	81.05	62.60	74.00	62.80	78.74	72.41	48.00
	Jedi (CVPR23) [42]	95.97	94.15	94.20	61.40	73.10	54.45	60.70	75.80	64.30	70.15	68.22	66.12
	ObjectSeeker (SP23) [50]	95.96	52.03	4.61	49.67	64.32	48.47	57.04	66.94	53.05	71.32	66.32	38.53
	PAD (Ours)	96.11	**96.36**	**96.26**	**84.55**	**87.80**	**88.95**	**68.40**	**87.81**	**85.00**	**87.56**	**89.21**	**83.23**
YOLOv3	Undefended	*96.42*	*66.93*	*64.04*	*44.07*	*78.80*	*62.92*	*51.48*	*42.36*	*64.93*	*78.67*	*64.73*	*66.70*
	LGS (WACV19) [33]	96.03	96.23	95.35	60.18	84.63	83.67	69.27	74.18	68.42	78.76	63.10	73.58
	SAC (CVPR22) [28]	96.08	96.52	95.95	79.39	83.46	78.96	56.01	72.93	64.80	84.20	66.31	67.42
	Jedi (CVPR23) [42]	**96.60**	93.64	94.78	74.18	59.76	48.63	52.17	75.79	69.28	69.23	62.09	71.69
	ObjectSeeker (SP23) [50]	95.82	70.78	71.17	42.31	73.27	56.18	53.78	49.59	29.28	66.34	47.63	43.89
	PAD (Ours)	96.08	**96.63**	**96.51**	**85.84**	**91.06**	**88.56**	**78.00**	**87.38**	**87.46**	**89.13**	**87.76**	**86.13**
YOLOv5s	Undefended	*95.72*	*51.07*	*37.78*	*28.47*	*45.22*	*41.45*	*35.96*	*29.67*	*38.35*	*38.30*	*29.69*	*36.58*
	LGS (WACV19) [33]	96.04	91.56	91.37	18.19	60.86	67.61	37.87	30.32	41.40	61.49	39.61	48.49
	SAC (CVPR22) [28]	95.72	92.33	92.06	74.21	77.87	78.24	40.25	29.77	38.46	59.03	31.03	37.43
	Jedi (CVPR23) [42]	**96.69**	87.70	90.65	42.96	46.88	48.79	38.10	51.59	54.11	52.22	44.96	58.84
	ObjectSeeker (SP23) [50]	91.61	50.91	38.17	35.03	39.09	43.45	37.54	37.49	38.38	48.78	35.81	33.51
	PAD (Ours)	96.17	**93.97**	**93.03**	**84.01**	**83.62**	**84.54**	**42.01**	**58.38**	**69.87**	**78.97**	**67.31**	**61.08**

[17], which are more challenging to detect by both humans and machines, the mAP increases by more than 10% on average (absolute) compared to the suboptimal method.

From the experimental results, it can be observed that ObjectSeeker [50] performs poorly under these attacks, some even worse than undefended. This is because ObjectSeeker can only defend against hiding attacks, and the assumption does not hold when encountering multiple patches. SAC [28] is best at defending against localized noise patches since its segmenter is trained on noise-like patch data. However, its performance significantly drops when facing natural-looking patches, with almost no defense capabilities against some of the patches. The performance of Jedi [42] is unstable, due to the influence of non-patch high-entropy regions. In contrast, PAD demonstrates robustness against various patches, benefiting from the universality of semantic independence and spatial heterogeneity and the complete independence from prior knowledge of attacks.

We also report the mAP of clean samples after defense in Table 1. PAD achieves a similarly high clean performance as the vanilla object detectors (0.02% drop on Faster R-CNN, 0.34% drop on YOLOv3, and 0.45% rise on YOLOv5s). For clean images without patches, although areas with relatively high values may remain after heat map threshold processing, they are usually scattered and will be eroded during subsequent morphological operations, thus not significantly impacting the model's performance.

5.3. Patch Localization Performance

Patch localization is a crucial step in the defense process, as it forms the foundation for subsequent patch removal. Therefore, we conducted a further evaluation of the patch localization performance. For accurate quantification, we propose a new metric called Patch Localization Recall. For each ground truth mask M_{patch} corresponding to an introduced patch region, we calculate the IoA between M_{patch} and the generated masks $M_{defense}$ obtained by the defense method, and mark this patch using the following notation:

Table 2. Patch Localization Recall(%) on Faster R-CNN.

Attack \ Defense		SAC [28]	Jedi [42]	**PAD**
Localized Noise [29]	DPatch-75	100.00	9.38	**100.00**
	DPatch-100	100.00	40.63	**100.00**
Printable [43]	OBJ	33.69	28.89	**85.90**
	OBJ-CLS	38.08	34.63	**86.24**
	Upper	35.68	33.95	**64.51**
Natural-looking [17]	P1	1.33	27.30	**31.16**
	P2	0.00	33.16	**66.97**
	P3	0.93	35.68	**70.43**
	P4	29.69	34.75	**81.49**
	P5	0.67	34.62	**74.57**
	P6	0.93	33.16	**70.84**

$$F_m = \begin{cases} 1 & IoA(M_{patch}, M_{defense}) \geq 0.5 \\ 0 & IoA(M_{patch}, M_{defense}) < 0.5 \end{cases} \tag{13}$$

$$Recall_{patch} = \frac{\sum_{m=1}^{M} F_m}{M}. \tag{14}$$

Since there is no patch localization process in certifiably robust defense, and the localization results of LGS [33] are not continuous regions, we primarily compare PAD with SAC [28] and Jedi [42]. The results are presented in Table 2, showing a significant improvement (30%-55% absolute and 2-3x relative) over existing state-of-the-art works. In the case of YOLO adversarial patch [43] and Naturalistic Patch [17], the ground truth masks include small regions that can be easily mistaken for the background. This is because the attack involves covering almost every visible pedestrian in the image with a patch, including small individuals in the

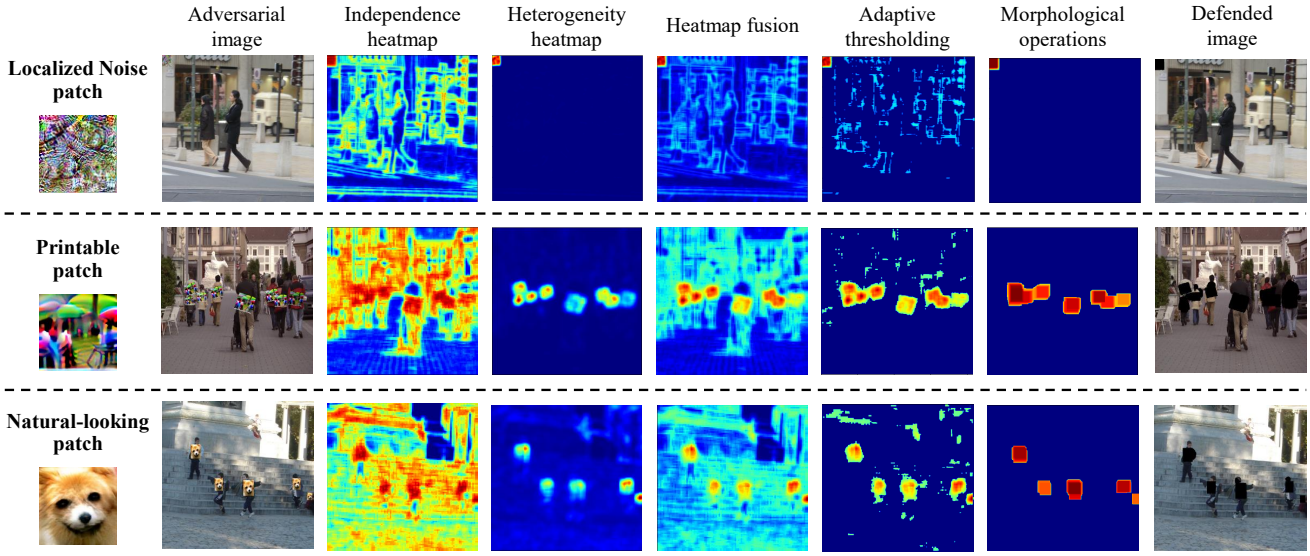


Figure 3. Visualization examples illustrating the patch localization process of PAD across different adversarial patch types.

distance. As a result, achieving high Patch Localization Recall values becomes more challenging. However, PAD still demonstrates effective performance against these attacks.

SAC [28] exhibits good performance for DPatch [29], as its segmenter is trained on adversarial images generated with PGD [31], which falls under the category of localized noise. However, it struggles when faced with natural-looking patches that it has not encountered before, leading to almost zero Patch Localization Recall. Jedi [42] performs poorly on DPatch-75, which may caused by the difficulty in the prior entropy distribution values adjustment. In contrast, PAD achieves high Patch Localization Recall for different types of patches, as it does not rely on prior knowledge or existing attack data.

According to the definition of Patch Localization Recall, It is natural to think that a defense method can achieve a high Patch Localization Recall by generating a mask with the widest possible coverage. However, removing a large number of non-patch areas from the image will inevitably result in a decrease in detection mAP. PAD achieves the highest values in both Patch Localization Recall and detection mAP, showcasing superior defense performance. We provide visualization of the patch localization process in Figure 3.

5.4. Ablation Study

To investigate the individual impacts of semantic independence and spatial heterogeneity in PAD, we conduct an ablation study that involves using only the H_{mi} from Semantic Independence Localization and only the H_{cd} from Spatial Heterogeneity Localization. Partial results on YOLOv8n are presented in Table 3. It can be observed that the full method, which combines semantic independence and spatial heterogeneity, achieves more stable overall performance.

Additionally, we have observed that spatial heterogeneity tends to outperform semantic independence in digital attack experiments. In digital attacks, patch generation is completely independent of the original clean image and less affected by interference, resulting in more pronounced heterogeneity, leading to better performance. However, in physical attacks where the patch is physically printed and imaged alongside other parts of the scene, the manifestation of heterogeneity may become weaker, and the role of semantic independence becomes more significant. In such cases, semantic independence outperforms spatial heterogeneity.

Table 3. mAP (%) of ablated defenses on YOLOv8n.

Defense \ Attack	OBJ	Upper	P3	P4	P5
LGS	47.5	82.0	53.1	79.4	62.4
SAC	81.9	58.1	51.8	78.2	53.5
Jedi	57.6	84.6	66.9	65.9	64.2
PAD-MI only	78.6	86.7	76.3	77.4	74.0
PAD-CD only	86.3	**89.8**	76.1	85.4	**82.2**
PAD-all	**87.5**	88.9	**78.7**	**85.4**	81.5

In this paper, to validate the defense performance of PAD without any parameter tuning, equal weights are assigned to Semantic Independence Localization and Spatial Heterogeneity Localization. By adjusting the weight allocation for digital attacks and physical attacks respectively, PAD can achieve even better results.

6. Defense Evaluation on Physical Attacks

To validate the effectiveness of PAD against physical attacks, we conducted experiments using a publicly available dataset that consists of physical adversarial patches [5], as well as physical attack videos captured by our own. We kept all parameters unchanged, ensuring that the implementation details remained consistent with the digital experiments.

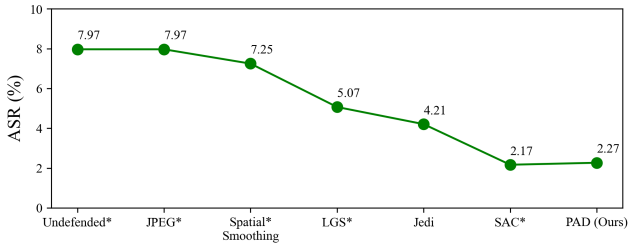


Figure 4. ASR (%) after defenses, lower values indicate better defense performance. Results with * are from [28].

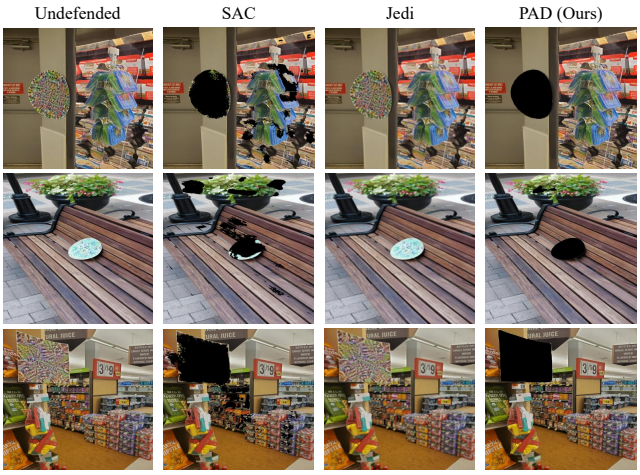


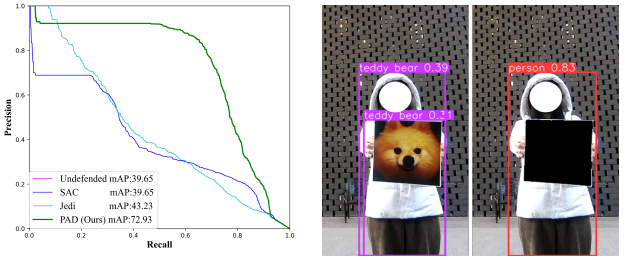
Figure 5. Comparison of defended images on APRICOT. Jedi [42] fails to locate patches, the drop in ASR is caused by resizing in Auto-Encoder. SAC [28] masks out most patch regions, but still affects many other regions despite the training with accurate mask annotations. PAD achieves the best removal performance.

6.1. Evaluation on APRICOT

APRICOT [5] consists of 1,011 photos with high resolution captured in real-world environments, encompassing both indoor and outdoor scenes. Each photo contains a printed physical adversarial patch, which varies in size, shape, location, viewing angle, and lighting conditions. These patches are generated to cause false detection of non-existent objects, targeting 10 specific classes.

We use Faster R-CNN [37] model pretrained on MS COCO [27] as our target object detector and evaluate the defense performance on the development set. Since this is a targeted attack, we use the Attack Success Rate (ASR) as our evaluation metric, setting the IoU threshold to 0.10 and the confidence threshold to 0.30. We present the results after applying different defense methods in Figure 4.

The results show that PAD, without any prior attack knowledge or training data of adversarial patches, significantly reduces the attack success rate to 2.27%. Moreover, SAC [28] utilizes APRICOT data with accurate masks to train its patch segmenter, getting an attack success rate of only 0.1% lower than PAD, highlighting the superiority of PAD. We provide examples of defended images produced by different defense methods in Figure 5, demonstrating our robustness against physical world patches of various sizes, shapes, lighting conditions, and angles.



(a) PR-curve of different defenses. SAC coincides with Undefended.

(b) Original and defended image. White circles added after prediction.

Figure 6. Defense results on our physical test set.

6.2. Evaluation on physical attack videos

To further evaluate the effectiveness of PAD against a wider range of patch types in the physical world, we print nine different patches, including P1-P6 [17], OBJ, OBJ-CLS, and Upper [43], and capture videos in five different indoor and outdoor scenes while holding these patches.

Due to the significant impact of lighting, distance, and angles on the success rate of physical attacks, we conduct extensive practical filming and testing to select a subset comprising images with relatively higher attack success rates. The final defense test set consists of 1100 photos, more details about the data distribution can be found in the supplementary material.

We use YOLOv8n as the target object detector and compare the defense performance of PAD with Jedi [42] and SAC [28] on this test set. The PR curves in Figure 6a demonstrate the superiority of PAD over the compared state-of-the-art methods. We show an example of the test image and defense result in Figure 6b.

7. Conclusion

In this paper, we identify two inherent characteristics of adversarial patches that are independent of their appearance, shape, size, location, and quantity. Leveraging these characteristics, we propose a patch-agnostic defense (PAD) method, which perform adversarial patch localization and removal without prior attack knowledge. PAD offers patch-agnostic defense against a wide range of adversarial patches, significantly enhancing the robustness of various pre-trained object detectors. Without training, PAD eliminates the reliance on existing attack data, making it more adaptable and capable of defending against novel patch attacks that have not been encountered yet. Our experimental results demonstrate the effectiveness in both digital space and the physical world, highlighting the practicality of PAD across different attack scenarios.

Acknowledgements. This work is supported in part by the National Natural Science Foundation of China Under Grants No.62176253 and No.U20B2066.

References

[1] Ultralytics yolov5. https://github.com/ultralytics/yolov5, . 5

[2] Ultralytics yolov8. https://github.com/ultralytics/ultralytics, . 5

[3] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE TIFS*, 7(3):1003–1017, 2012. 4

[4] Folkmar Bornemann and Tom März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 28:259–278, 2007. 4

[5] A Braunegg, Amartya Chakraborty, Michael Krumdick, Nicole Lape, Sara Leary, Keith Manville, Elizabeth Merkhofer, Laura Strickhart, and Matthew Walmer. Apricot: A dataset of physical adversarial attacks on object detection. In *ECCV*, pages 35–50, 2020. 7, 8

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2

[7] Niklas Bunzel, Ashim Siwakoti, and Gerrit Klause. Adversarial patch detection and mitigation by detecting high entropy regions. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, pages 124–128, 2023. 1, 2, 3

[8] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68, 2019. 2

[9] Zitao Chen, Pritam Dash, and Karthik Pattabiraman. Turning your strength against you: Detecting and mitigating robust and universal adversarial patch attacks. *arXiv e-prints*, pages arXiv–2108, 2021. 2

[10] Ping-Han Chiang, Chi-Shen Chan, and Shan-Hung Wu. Adversarial pixel masking: A defense against physical attacks for pre-trained object detectors. In *ACM MM*, pages 1856–1865, 2021. 1, 2, 4

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 5

[12] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE TIFS*, 4(1):154–160, 2009. 4

[13] Thomas Gittings, Steve Schneider, and John Collomosse. Vax-a-net: Training-time defence against adversarial patch attacks. In *ACCV*, 2020. 2

[14] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *CVPRW*, pages 1597–1604, 2018. 2

[15] Junfeng He, Zhouchen Lin, Lifeng Wang, and Xiaoou Tang. Detecting doctored jpeg images via dct coefficient analysis. In *ECCV*, pages 423–435, 2006. 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[17] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *ICCV*, pages 7848–7857, 2021. 2, 5, 6, 8

[18] Zhanhao Hu et al. Adversarial texture for fooling person detectors in the physical world. In *CVPR*, 2022. 5

[19] Hao Huang et al. T-sea: Transfer-based self-ensemble attack on object detection. In *CVPR*, 2023. 5

[20] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *ECCV*, pages 514–529, 2018. 2

[21] Nan Ji, YanFei Feng, Haidong Xie, Xueshuang Xiang, and Naijin Liu. Adversarial yolo: Defense human detection patch attacks via detecting adversarial patches. *arXiv preprint arXiv:2103.08860*, 2021. 1, 2

[22] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *ICML*, pages 2507–2515, 2018. 2

[23] Taeheon Kim, Youngjoon Yu, and Yong Man Ro. Defending physical adversarial attack on object detection via adversarial patch-feature energy. In *ACM MM*, pages 1905–1913, 2022. 1, 2, 5

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 3

[25] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *CVPR*, pages 14254–14263, 2020. 2

[26] Guo-Shiang Lin, Min-Kuan Chang, and You-Lin Chen. A passive-blind forgery detection scheme based on content-adaptive quantization table estimation. *IEEE TCSVT*, 21(4): 421–434, 2011. 4

[27] 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 *ECCV*, 2014. 5, 8

[28] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *CVPR*, pages 14973–14982, 2022. 1, 2, 4, 5, 6, 7, 8

[29] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018. 2, 5, 6, 7

[30] Jan Lukáš and Jessica Fridrich. Estimation of primary quantization matrix in double compressed jpeg images. In *Proc. Digital forensic research workshop*, pages 5–8, 2003. 4

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 7

[32] Jan Hendrik Metzen, Nicole Finnie, and Robin Hutmacher. Meta adversarial training against universal patches. *arXiv preprint arXiv:2101.11453*, 2021. 2

[33] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *WACV*, pages 1300–1307, 2019. 1, 2, 5, 6

[34] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. In *ECCV*, pages 429–448, 2020. 2

[35] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 2, 5

[36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 5

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 5, 8

[38] Giulio Rossolini, Federico Nesti, Fabio Brau, Alessandro Biondi, and Giorgio Buttazzo. Defending from physically-realizable adversarial attacks through internal over-activation analysis. In *AAAI*, pages 15064–15072, 2023. 2

[39] Aniruddha Saha, Akshayvarun Subramanya, Koninika Patil, and Hamed Pirsiavash. Role of spatial context in adversarial robustness for object detection. In *CVPRW*, pages 784–785, 2020. 1, 2

[40] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *NeurIPS*, 32, 2019. 2

[41] Jia Tan, Nan Ji, Haidong Xie, and Xueshuang Xiang. Legitimate adversarial patches: Evading human eyes and detection models in the physical world. In *ACM MM*, pages 5307–5315, 2021. 2

[42] Bilel Tarchoun et al. Jedi: Entropy-based localization and removal of adversarial patches. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[43] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPRW*, 2019. 2, 5, 6, 8

[44] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. *arXiv preprint arXiv:1909.09552*, 2019. 2

[45] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *ECCV*, pages 1–17, 2020. 2

[46] Chong Xiang and Prateek Mittal. Detectorguard: Provably securing object detectors against localized patch hiding attacks. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 3177–3196, 2021. 1, 3

[47] Chong Xiang and Prateek Mittal. Patchguard++: Efficient provable attack detection against adversarial patches. *arXiv preprint arXiv:2104.12609*, 2021.

[48] 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 *USENIX Security 21*, pages 2237–2254, 2021.

[49] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. {PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier. In *USENIX Security 22*, pages 2065–2082, 2022.

[50] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. Objectseeker: Certifiably robust object detection against patch hiding attacks via patch-agnostic masking. In *IEEE Symposium on Security and Privacy (SP)*, pages 1329–1347, 2023. 1, 3, 5, 6

[51] 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 *ECCV*, pages 665–681, 2020. 2

[52] Ke Xu et al. Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *WACV*, 2023. 2

[53] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *ECCV*, pages 681–698, 2020. 2

[54] Cheng Yu, Jiansheng Chen, Youze Xue, Yuyang Liu, Weitao Wan, Jiayu Bao, and Huimin Ma. Defending against universal adversarial patches by clipping feature norms. In *ICCV*, pages 16434–16442, 2021. 2

[55] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *ICCV*, pages 421–430, 2019. 2

[56] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *ACM SIGSAC conference on computer and communications security*, pages 1989–2004, 2019. 2