PAD: Patch-Agnostic Defense against Adversarial Patch Attacks

Supplementary Material

1. Morphological Operations in Fusion Block

After applying the adaptive thresholding, we perform morphological operations on the heat map with the main purpose of removing isolated background points and purifying the patch area. For an input image with dimensions $w \times h \times c$, we set the adaptive base kernel size as follows:

$$k_{base} = \frac{min(w,h)}{\delta} \times \frac{min(w,h)}{\delta}, \quad (1)$$

where δ is a fixed hyperparameter that we set to 80 in our implementation. To minimize the impact on the integrity of the patch area, we adopt a progressive strategy. The initial opening operation uses a kernel of size $k_{base} \times 2$, followed by a closing operation with a kernel size of k_{base}, and finally a second opening operation with a kernel size of $k_{base} \times 3$. We visualize each step of the morphological operation process, as shown in Figure 1. As shown in the examples, small noise pixels are initially removed during the first opening operation. Subsequently, the closing operation fills in small holes within the patch region, preventing disconnected areas from being eroded in the subsequent opening operation. Finally, the larger kernel size in the opening operation effectively eliminates spikes or small bridges, resulting in a more accurate and complete patch region. The consecutive application of these morphological operations optimizes and refines the patch localization results.

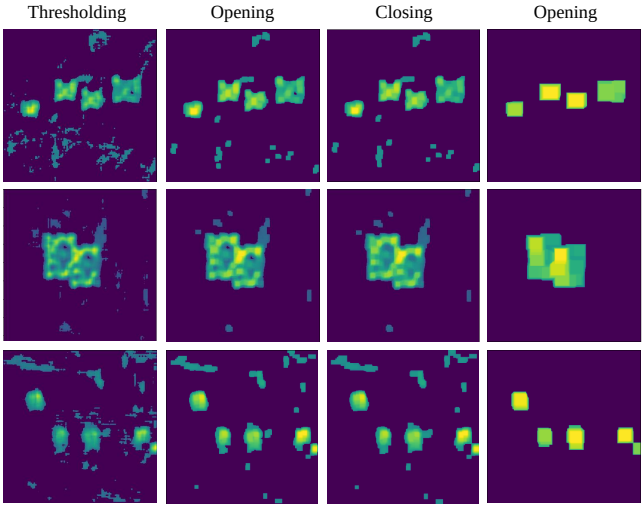


Figure 1. Visualization examples of morphological operations.

2. More About Evaluation on Digital Attacks

2.1. Defense performance against more attacks

We conduct additional defense experiments against more state-of-the-art physical patch attack methods [5][6] on different datasets. As shown in Table 1, PAD achieves the best performance against all of them. Among them, experiments against AdvTexture and T-SEA are conducted on YOLOv2, and the rest are conducted on Faster R-CNN. The experimental results demonstrate the generalizability of PAD in diverse detection scenarios.

Table 1. mAP (%) under more attacks on different datasets.

Attack	Dataset	Undefended	LGS	SAC	Jedi	PZ[11]	**PAD**
AdvTexture [5]	Inria	43.51	75.26	59.08	68.88	-	**79.83**
T-SEA [6]	Inria	17.05	35.15	17.60	40.22	-	**74.93**
	COCO-Person	10.56	13.33	10.59	15.32	-	**30.93**
Dpatch	COCO	29.1	40.7	44.1	38.5	-	**45.1**
Masked PGD	VOC	10.90	61.53	62.81	60.79	66.10	**67.12**

2.2. Defense performance on other detectors

In addition to the experimental results on Faster R-CNN, YOLOv3, and YOLOv5s listed in Table 1 of the main paper, we also conduct defense performance evaluations on YOLOv2 [8] and YOLOv8n [1] with the same setting. The detailed results are presented in Table 2.

From the experimental results, it can be observed that, consistent with the results on Faster R-CNN, YOLOv3, and YOLOv5s, our PAD achieves the best defense performance on YOLOv2 and YOLOv8n as well. PAD demonstrates a considerable improvement in mean average precision (mAP) compared to suboptimal state-of-the-art methods when defending against different types of adversarial patches while maintaining a similar clean performance.

2.3. Ablation study for patch inpainting

After obtaining an accurate patch mask, PAD employs a simple inpainting method: replacing the patch region with black pixels, which is consistent with [3, 7]. To validate the effectiveness of this inpainting approach, we compare it with another inpainting method: the coherence transport-based inpainting method (inpaintBCT) [2], which is used in Jedi [9]. We evaluate the defense performance using these two inpainting methods on different detectors, as shown in Table 3.

According to the results, inpaintBCT works better in some cases, but in most cases, black works better. Comparing the results in Table 1 of the main paper, it can be observed that regardless of the specific inpainting method used, both achieve superior defense performance compared to other state-of-the-art methods. This highlights the absolute advantage of PAD in patch localization, making the choice of inpainting method less crucial. The simplest removal method is sufficient to achieve a satisfactory defense effect.

Table 2. mAP(%) on YOLOv2 and YOLOv8n. The best performance is **bolded**, and the suboptimal performance is underlined.

Detector	Defense	Clean	OBJ	OBJ-CLS	Upper	P1	P2	P3	P4	P5	P6
YOLOv2	Undefended	*90.13*	*3.71*	*10.48*	*20.99*	*12.07*	*60.50*	*24.50*	*49.23*	*33.93*	*21.10*
	LGS (WACV19)	89.28	5.27	46.63	55.75	20.43	61.59	27.62	60.88	38.41	27.30
	SAC (CVPR22)	**90.11**	60.48	67.90	68.20	17.70	24.75	24.75	64.58	35.05	22.69
	Jedi (CVPR23)	89.01	19.30	63.26	52.03	26.30	62.90	49.56	57.69	48.26	53.77
	PAD (Ours)	89.77	**76.79**	**82.92**	**81.33**	**35.79**	**79.71**	**75.00**	**80.88**	**78.23**	**71.88**
YOLOv8n	Undefended	*96.42*	*56.74*	*75.30*	*65.77*	*68.48*	*50.89*	*51.78*	*65.81*	*51.64*	*50.06*
	LGS (WACV19)	96.60	47.52	82.57	82.00	68.14	51.41	53.11	79.42	62.40	64.30
	SAC (CVPR22)	96.42	81.92	86.95	84.59	69.86	51.03	51.80	78.19	53.51	50.95
	Jedi (CVPR23)	**96.64**	57.63	64.35	58.13	69.85	66.20	66.94	65.90	64.20	69.51
	PAD (Ours)	96.40	**87.53**	**87.69**	**88.92**	**70.73**	**74.84**	**78.68**	**85.37**	**81.51**	**77.25**

Table 3. mAP (%) of inpainting-ablated PAD under different adversarial patch attacks. The best performance is **bolded**.

Detector	Defense	OBJ	OBJ-CLS	Upper	P1	P2	P3	P4	P5	P6
Faster R-CNN	PAD-inpaintBCT	82.60	**89.44**	85.84	**71.50**	83.05	83.14	84.75	84.79	79.64
	PAD-black	**86.80**	87.80	**88.95**	68.40	**87.81**	**85.00**	**87.56**	**89.21**	**83.23**
YOLOv2	PAD-inpaintBCT	74.51	82.07	79.45	23.73	76.17	74.25	78.12	76.67	64.71
	PAD-black	**76.79**	**82.92**	**81.33**	**35.79**	**79.71**	**75.00**	**80.88**	**78.23**	**71.88**
YOLOv3	PAD-inpaintBCT	84.49	89.98	87.19	**83.26**	82.90	83.73	86.28	84.22	80.53
	PAD-black	**85.84**	**91.06**	**88.56**	78.00	**87.38**	**87.46**	**89.13**	**87.76**	**86.13**
YOLOv5s	PAD-inpaintBCT	78.73	83.22	78.66	**45.21**	58.28	66.02	66.91	58.57	54.15
	PAD-black	**84.01**	**83.62**	**84.54**	42.01	**58.38**	**69.87**	**78.97**	**67.31**	**61.08**
YOLOv8n	PAD-inpaintBCT	83.72	**89.94**	86.08	**73.58**	74.61	**79.43**	84.67	75.70	67.57
	PAD-black	**87.53**	87.69	**88.92**	70.73	**74.84**	78.68	**85.37**	**81.51**	**77.25**

3. More About Evaluation on Physical Attacks

3.1. Ablation results on APRICOT

To investigate the individual contributions of semantic independence and spatial heterogeneity in defense against physical attacks, we conducted ablation experiments on the APRICOT dataset. Table 4 presents the adversarial success rate (ASR) for five detectors using semantic independence alone, spatial heterogeneity alone, and both semantic independence and spatial heterogeneity combined.

Table 4. ASR (%) of ablated PAD on APRICOT. Lower values indicate better defense performance.

Detector	PAD-MI only	PAD-CD only	PAD-all
Faster R-CNN	3.03	4.54	**2.27**
YOLOv2	0.62	1.52	**0.43**
YOLOv3	**0**	0.64	0.60
YOLOv5s	**0**	**0**	**0**
YOLOv8n	**0.27**	0.45	0.42

The experimental results indicate that under the physical attack setting, semantic independence performs better than spatial heterogeneity, yielding lower ASR values across all five detectors. This finding aligns with our analysis in Section 5.4 of the main paper. Even without adjusting the weights of semantic independence and spatial heterogeneity, PAD still achieves significant defense effectiveness.

3.2. Visualization of patch localization on APRICOT

In Figure 2, we present visual examples showcasing the patch localization process of PAD on the APRICOT dataset. Compared with the visual examples under digital attacks in Figure 3 of the main paper, it can be observed that the influence of spatial heterogeneity in physical attack scenarios may be affected by lighting conditions, angles, and imaging processes. On the other hand, the significance of semantic independence becomes more prominent. These two characteristics play different roles in defending against digital and physical attacks, and their organic combination contributes to the exceptional performance of PAD.

Additionally, we compare the heat map when using semantic independence alone in PAD with the entropy-based defense approach [9] in Figure 3. It can be observed that complex textured backgrounds, such as shelves filled with goods, have a significant impact on the entropy-based defense, resulting in high entropy values in a large portion of the background area. However, the performance of semantic independence localization based on mutual information in PAD, remains relatively stable. This is because although the background area is complex, adjacent regions still exhibit semantic correlation, leading to lower values of semantic independence.

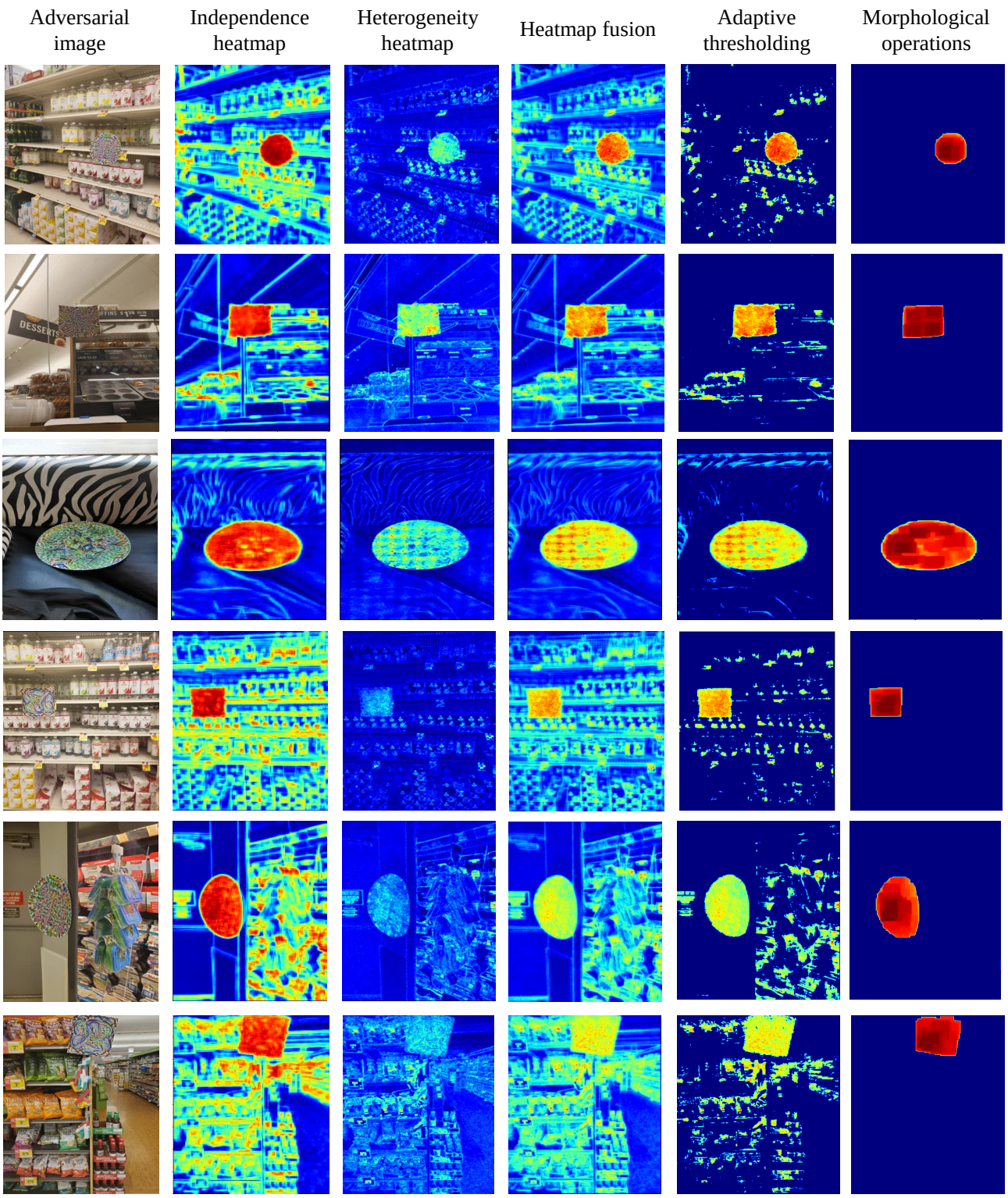


Figure 2. Visualization examples illustrating the patch localization process of PAD on APRICOT.

3.3. Data distribution of physical test set

We print 9 different patches including P1-P6 [4], OBJ, OBJ-CLS, and Upper [10], taking videos in both indoor and outdoor environments across five different scenes. We build the physical test set which consists of 1100 images. Figure 4a and Figure 4b display the distribution of the patches and indoor/outdoor scenes respectively.

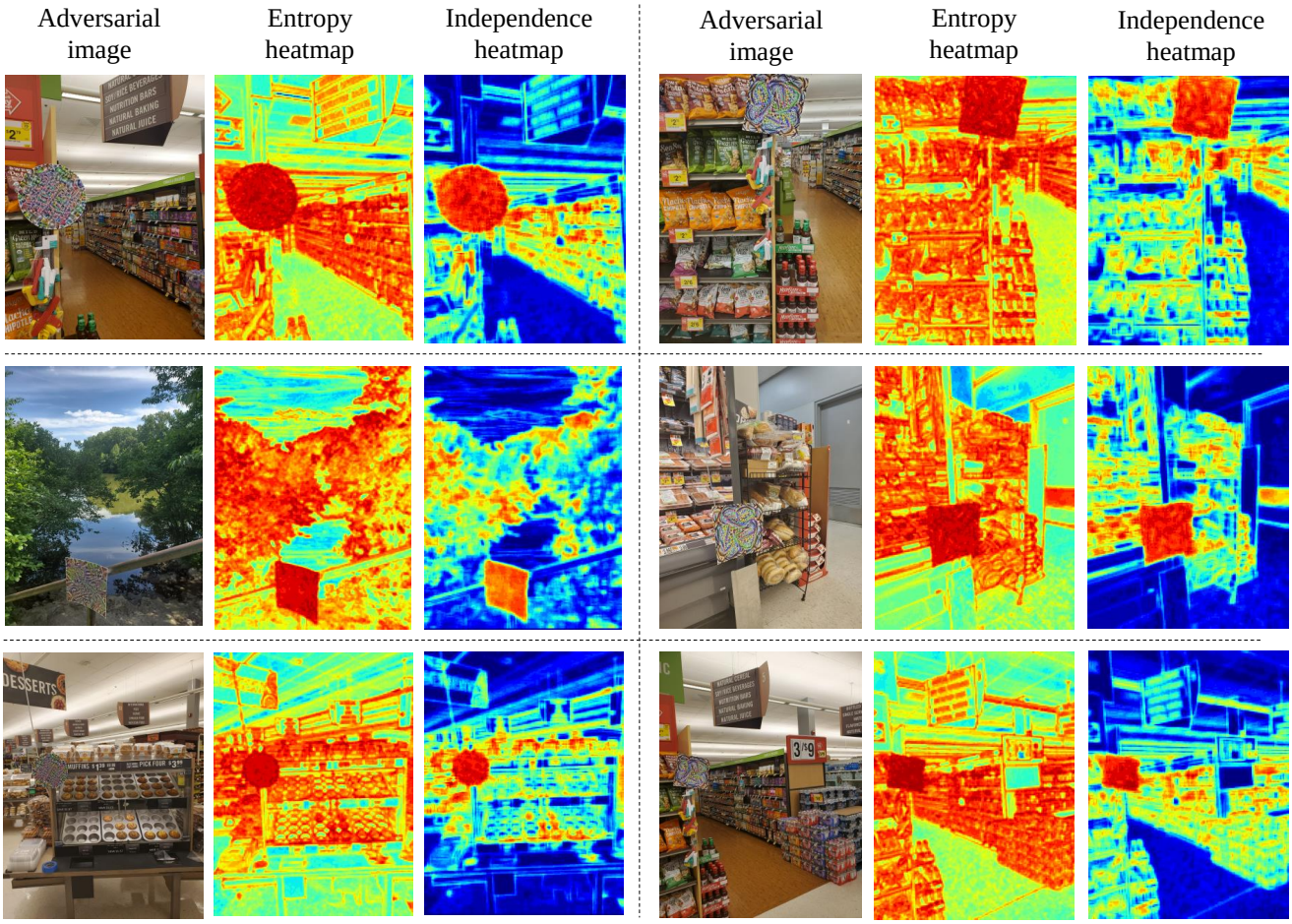
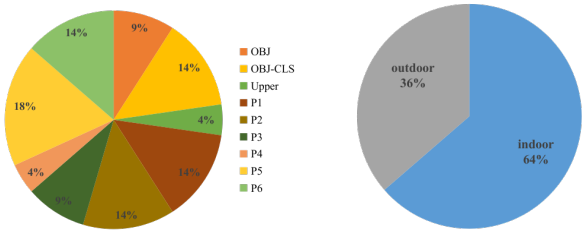


Figure 3. Visualization comparison of entropy heat map and semantic independence heat map.



(a) Adversarial patches (b) Scenes

Figure 4. Data distribution of our physical test set.

4. Discussion on Adaptive Attacks

In order to bypass the defense method proposed in this paper, attackers would need to ensure two key aspects during the patch generation process: 1) The image quality of the patch should be as consistent as possible with other parts of the image. 2) The patch should be semantically similar to its surrounding context. However, image quality can vary significantly depending on the capturing device and the image compression algorithms used. Additionally, the semantic content differs across each image and different positions within an image. Therefore, achieving both of these requirements would necessitate training unique patches for different positions within each image, rendering patch reusability impossible. This limitation proves fatal for physical attacks, meaning attackers cannot pre-train patches and simply print and place them in different scenes to execute attacks. Even if the goal is to target a fixed scene and angle, the presence of moving objects (such as pedestrians or vehicles) and the variability introduced by factors like camera angles and lighting condition would prevent the attacker from accurately predicting the final image composition. Additionally, if the patch content closely matches the surrounding semantic space, the detect model will likely confuse it with the surrounding context, resulting in a low probability of attack success. Therefore, the threat of adaptive attacks against PAD is weak.

References

[1] Ultralytics yolov8. https://github.com/ultralytics/ultralytics. 1

[2] Folkmar Bornemann and Tom März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 28:259–278, 2007. 1

[3] 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

[4] 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. 3

[5] Zhanhao Hu et al. Adversarial texture for fooling person detectors in the physical world. In *CVPR*, 2022. 1

[6] Hao Huang et al. T-sea: Transfer-based self-ensemble attack on object detection. In *CVPR*, 2023. 1

[7] 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

[8] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 1

[9] Bilel Tarchoun et al. Jedi: Entropy-based localization and removal of adversarial patches. In *CVPR*, 2023. 1, 2

[10] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPRW*, 2019. 3

[11] Ke Xu et al. Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *WACV*, 2023. 1