AntiStyler: Defending Object Detection Models Against Adversarial Patch Attacks Using Style Removal

Idan Yankelev[1] Edita Grolman[1] Yarin Yerushalmi Levi[1] Amit Giloni[2]

Omer Hofman[2] Toshiya Shimizu[3] Yuval Elovici[1] Asaf Shabtai[1]

[1]Ben-Gurion University of the Negev [2]Fujitsu Research of Europe [3]Fujitsu Limited

Presented by: Idan Yankelev

AI Researcher, Cyber @ Ben Gurion University Labs

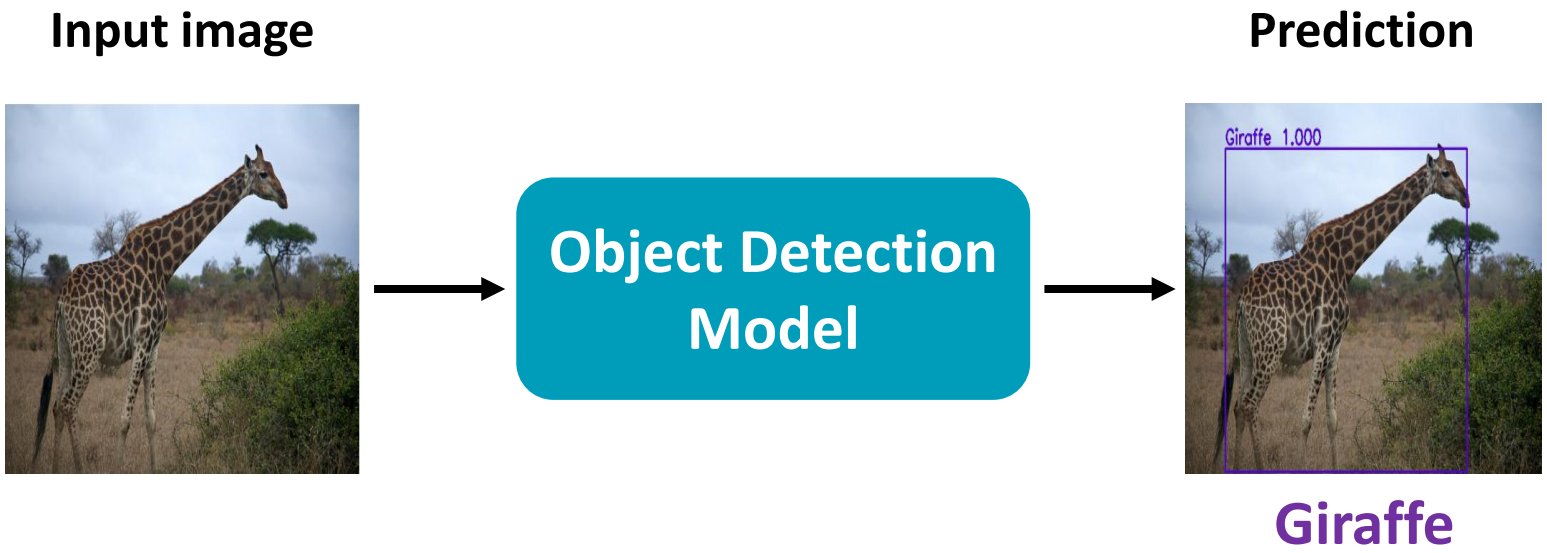
Curious?
Check out our paper and demo!

Email Contact:
idanyan@post.bgu.ac.il

Introduction

Object Detection

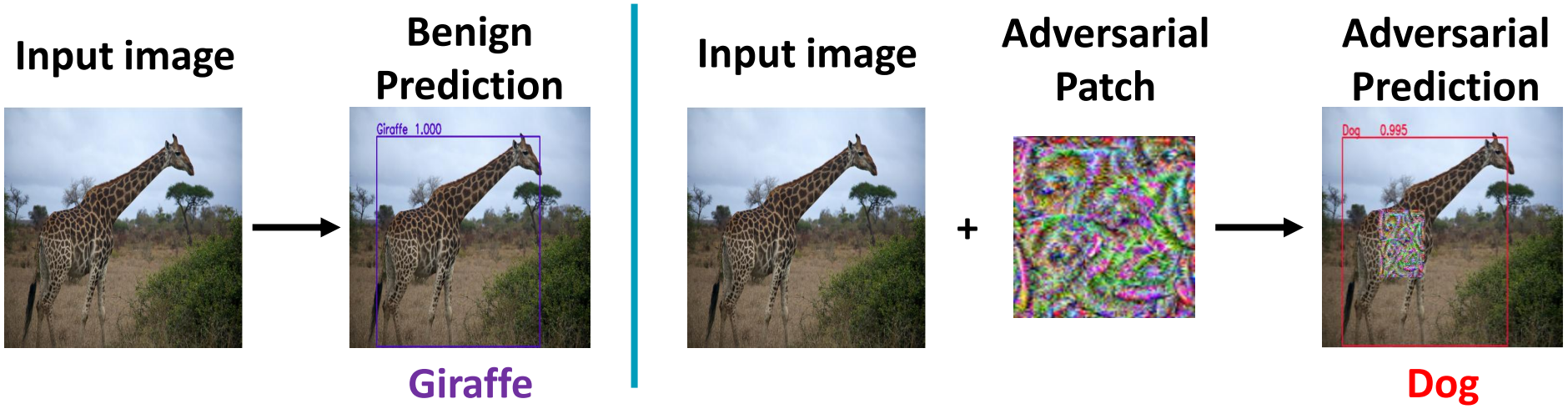
- In the field of computer vision, the object detection (OD) task consists of the localization and classification of objects within a scene [1].



Introduction

Adversarial Patch Attacks

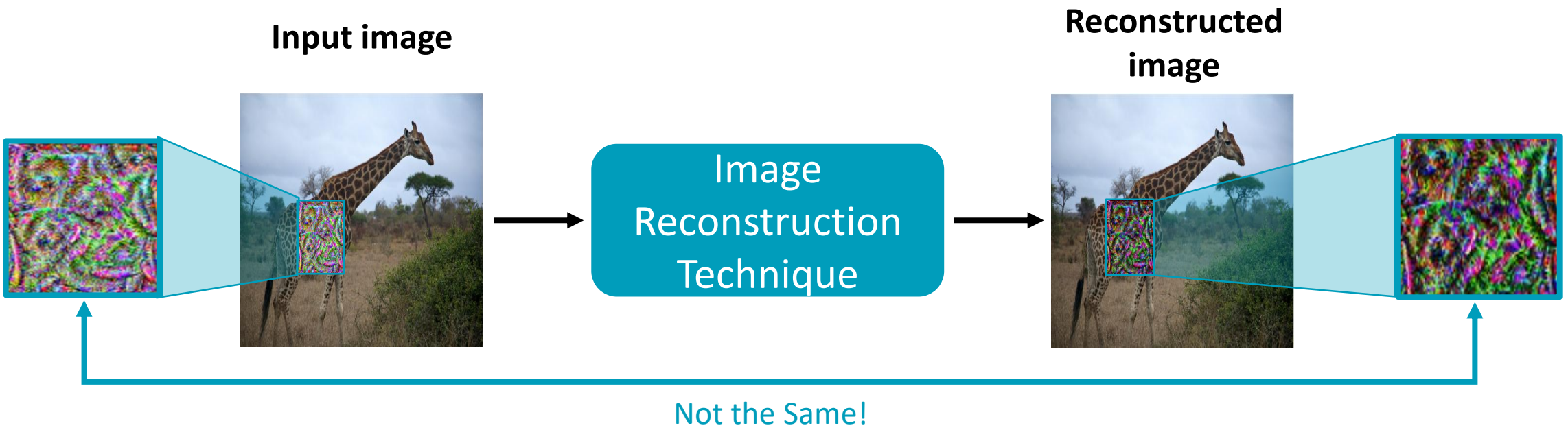
- Similar to other deep learning models, OD models are vulnerable to adversarial attacks, particularly adversarial patch attacks [2].
  - These attacks pose a significant threat to the reliability of an OD model by inserting a maliciously crafted patch onto an image, causing the model to make incorrect predictions.



Introduction

Distributional Gap Between Adversarial And Benign Images

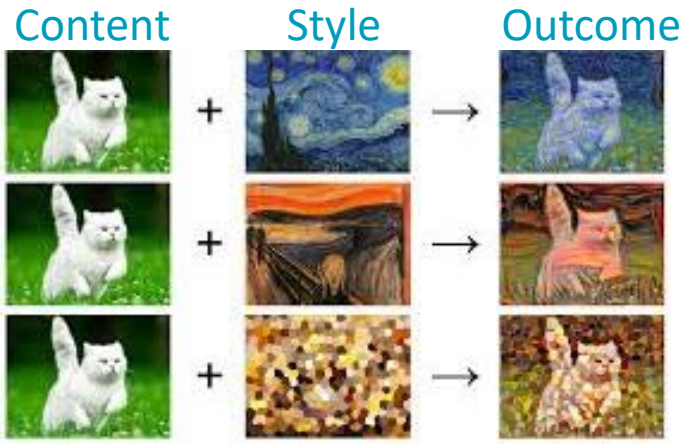
- Recent research [3-6] has observed that adversarial and benign images can be viewed as being drawn from different distributions.
  - Due to this distributional gap, image reconstruction techniques have been found to struggle with reconstructing adversarial regions.
    - This can be leveraged to identify and mask regions where reconstruction quality falls below a predefined threshold.

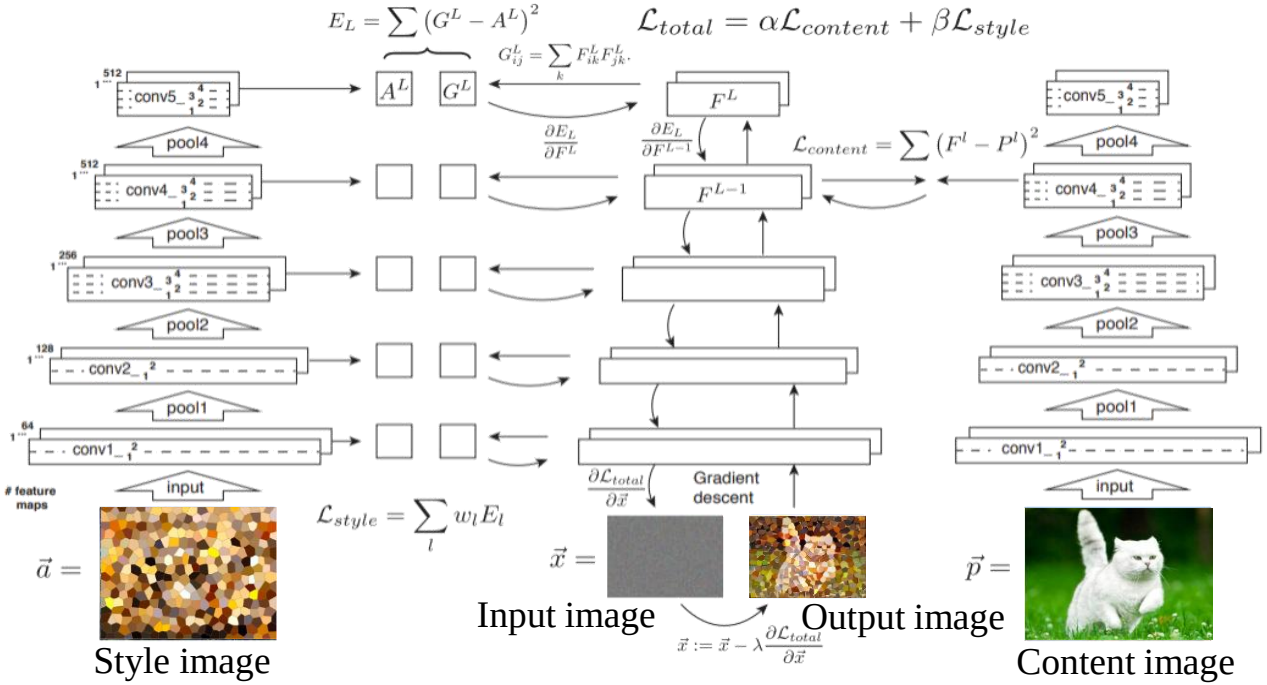


Background

Neural Style Transfer

- Neural style transfer [7] is a technique that optimizes the process of blending two images, a content image and a style reference image, into a single image.





Background

Neural Style Transfer

- Given a content image X_C, a style image X_S, and an initial blank/random noise image, the neural style transfer model generates a new image X_O that is based on the content image while incorporating the style properties of the style image.

$$L_{ST}(X_O, X_C, X_S) = \alpha L_C(X_O, X_C) + \beta L_S(X_O, X_S)$$

Measures the Mean Square Error (MSE) between the Feature Maps of the output image X_O and the content image X_C.

It ensures that the spatial structure and objects (e.g., cat) are preserved in the final result.

Measures the MSE between the Gram Matrices of the output image and the style image.

Gram Matrices capture the correlations between different feature channels, representing textures and color patterns without spatial information.

Background

Neural Style Transfer

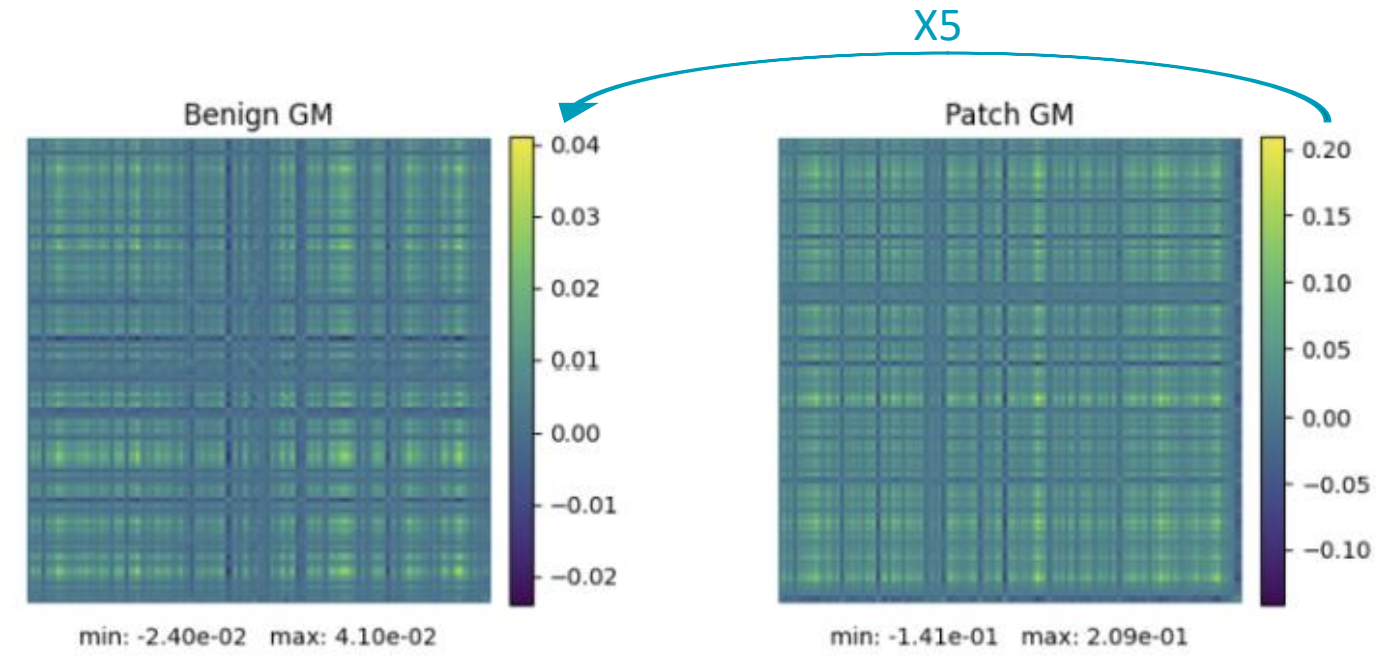
- Recently, style transfer was mathematically demonstrated [8] to be a form of domain adaptation (with a second-order polynomial kernel).
  - The goal of domain adaptation is to minimize the dissimilarity between the distributions of the source and target domains using the maximum mean discrepancy (MMD) metric.

$$\begin{aligned}\mathcal{L}^l_{style} =& \frac{1}{4N_l^2M_l^2}\sum_{k_1=1}^{M_l}\sum_{k_2=1}^{M_l}\Big(k(\mathbf{f}^l_{\cdot k_1},\mathbf{f}^l_{\cdot k_2})\\ &+k(\mathbf{s}^l_{\cdot k_1},\mathbf{s}^l_{\cdot k_2})-2k(\mathbf{f}^l_{\cdot k_1},\mathbf{s}^l_{\cdot k_2})\Big)\\ =& \frac{1}{4N_l^2}\mathrm{MMD}^2[\mathcal{F}^l,\mathcal{S}^l],\end{aligned}$$

Motivation

Empirical Motivation: Gram Matrix Analysis

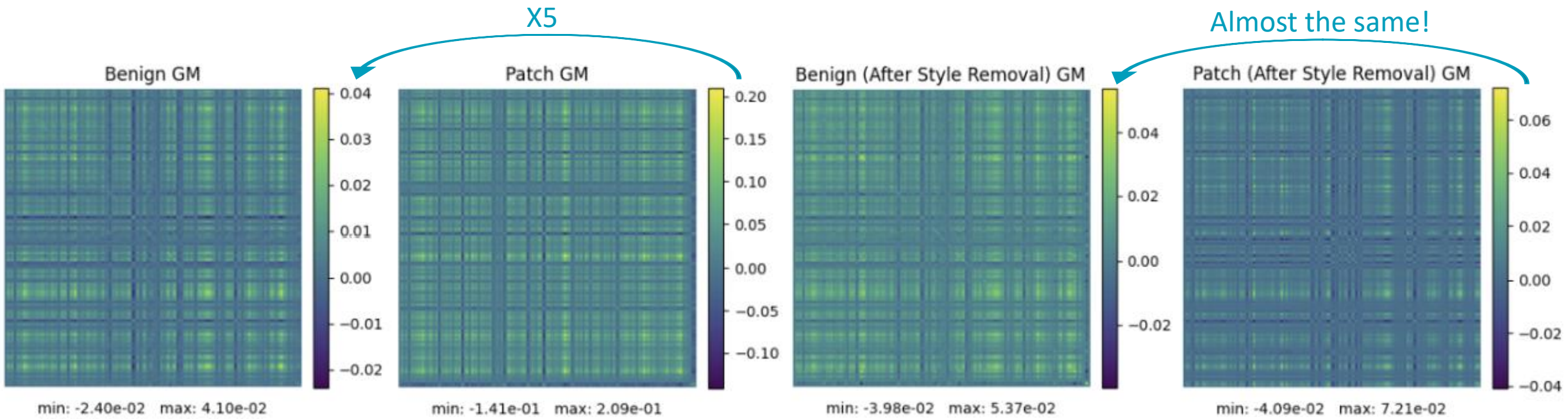
- When we analyzed the Gram matrices (GMs) of adversarial regions compared to benign ones, we observed that adversarial regions induce GMs with higher magnitudes than those induced by benign regions.



Motivation

Empirical Motivation: Gram Matrix Analysis

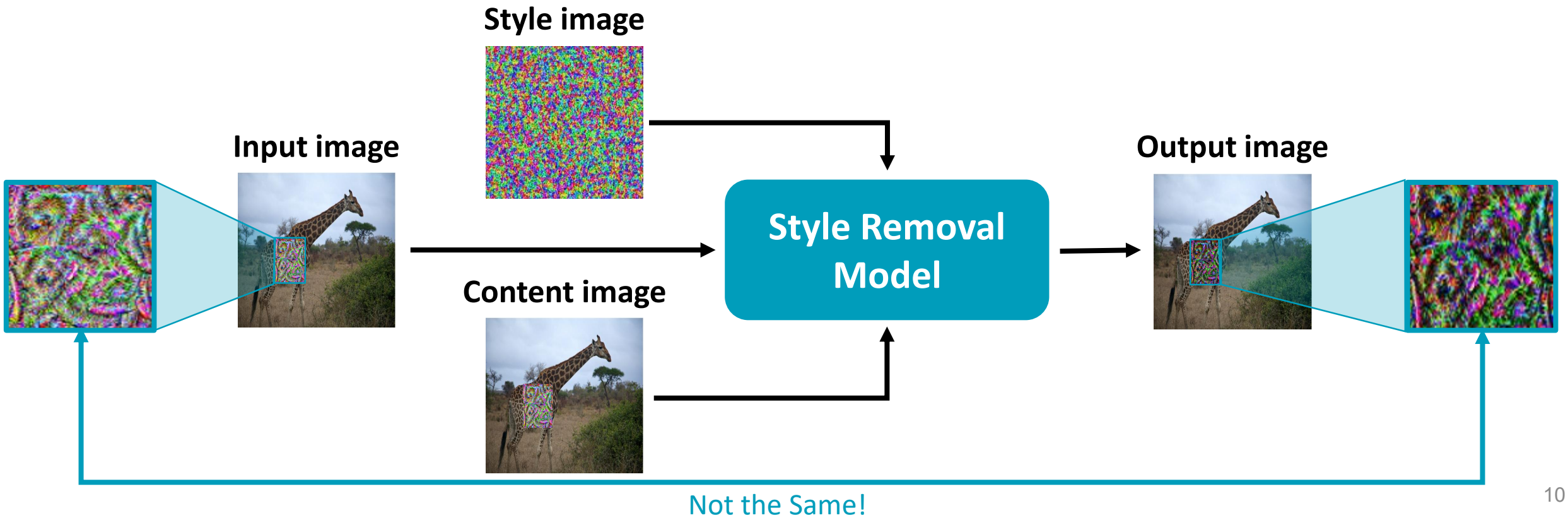
- To avoid constraining our method to a specific attack style, we sought a style whose GMs have similar higher magnitudes to those of adversarial regions.
  - We found that the GMs of a randomly sampled noise image yield similarly high magnitudes.
- Maximizing the style loss between an image and a randomly sampled noise had minimal effect on benign GMs, while flattening the magnitude of adversarial ones.



Method

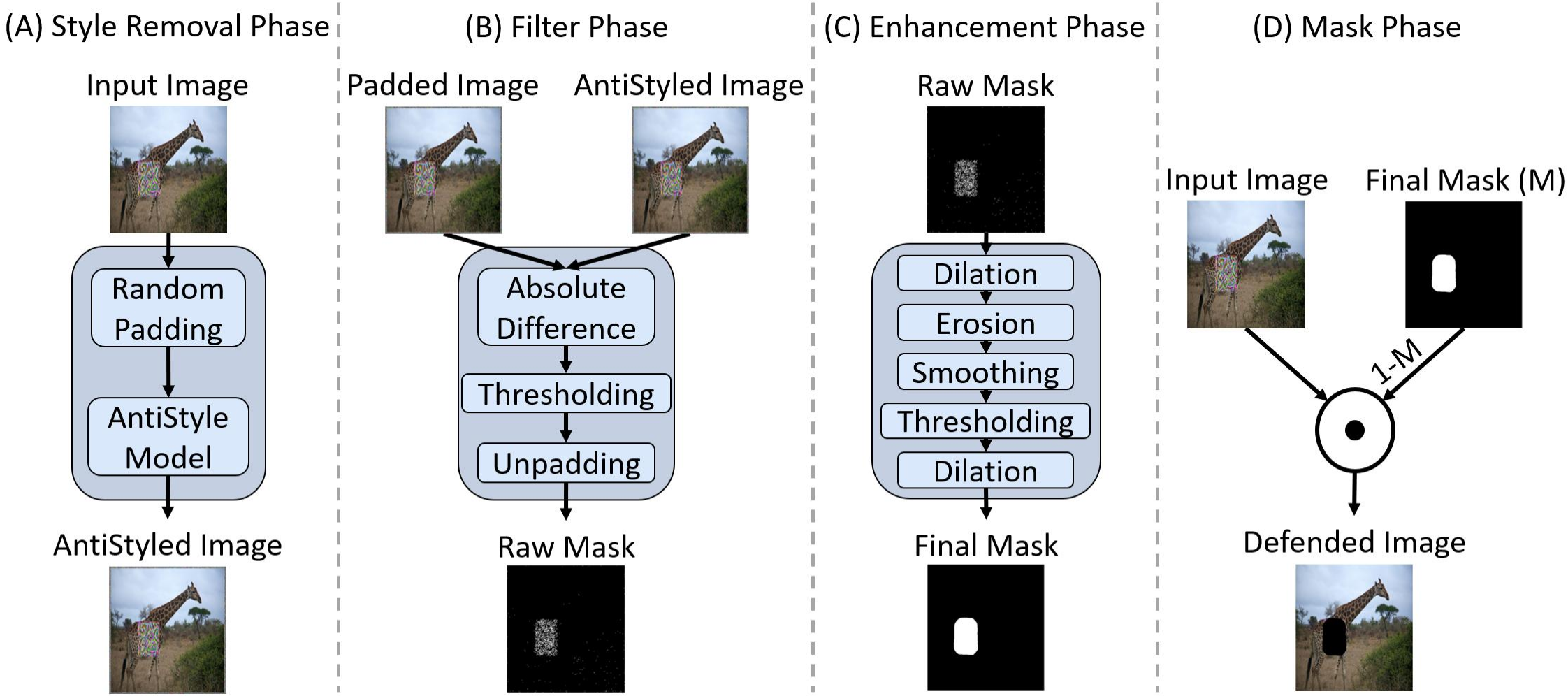
AntiStyle Model

- To create our desired effect, we created a customized style transfer model with random noise as a style reference; however, rather than generating an image closer to this style, our technique intentionally deviates from this style.



Method

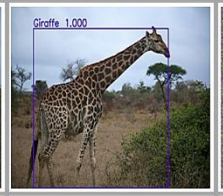
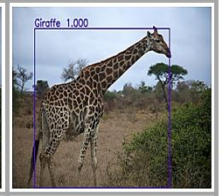
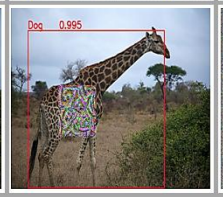
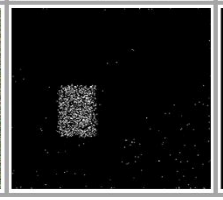
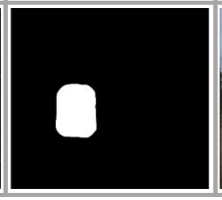
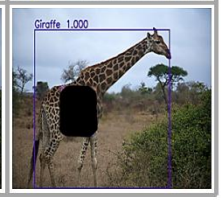
AntiStyler's Pipeline



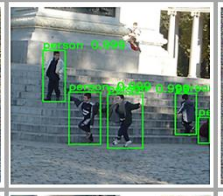
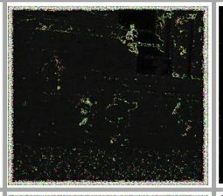
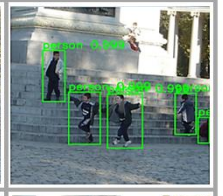
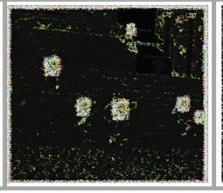
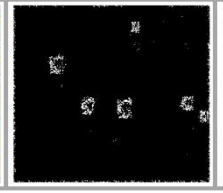
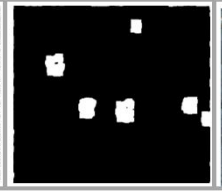
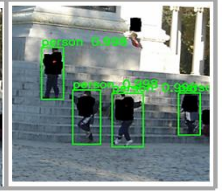
Method

AntiStyler's Effect

Effect Comparison on Single-Patch Attacked Images

	Original Image	Original Prediction	AntiStyled Image	Normalized Absolute Difference	Raw Mask	Enhanced Mask	Defended Image	Defended Prediction
Benign Image								
Adversarial Image								

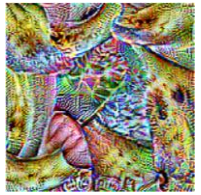
Effect Comparison on Multi-Patch Attacked Images

	Original Image	Original Prediction	AntiStyled Image	Normalized Absolute Difference	Raw Mask	Enhanced Mask	Defended Image	Defended Prediction
Benign Image								
Adversarial Image								

Evaluation

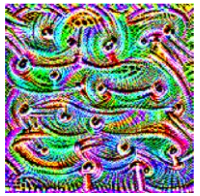
- Our evaluation was conducted on two digital datasets (COCO [9] and INRIA [10]) and two physical datasets (Superstore [11] and APRICOT [12]).
  - We used a wide range of adversarial patch attacks: DPatch [13], Google's patch [14], Masked-PGD (M-PGD)[15], T-SEA [16], natural patches [17], and printable patches [18].

M-PGD Patch



D-Patch Patch

Google Patch



T-SEA Patch

Natural Patch

Printable Patch

COCO Patched Image

INRIA Patched Image

Superstore Patched Image

Apricot Patched Image

Evaluation

Results on MSCOCO

OD Model	Defense Method	Processing Time [ms]	Google			M-PGD			DPatch		
			Benign	Adv	Mean	Benign	Adv	Mean	Benign	Adv	Mean
Faster RCNN	Undefended	43.2	51.6	16.6	34.1	45.8	22.3	34.1	49.4	23.0	36.2
	LGS (WACV19)	713.8	42.5	26.8	34.7	38.0	30.8	34.4	43.2	27.7	35.5
	Grad-Defense (CVPR20)	**43.2**	46.7	22.9	34.8	47.1	28.2	37.6	47.8	28.6	38.2
	SAC (CVPR22)	78.0	**51.6**	22.7	37.2	45.8	35.5	40.7	49.4	31.5	40.5
	ObjectSeeker (SP23)	4191.9	**51.6**	31.0	41.3	44.5	28.4	36.5	48.0	28.4	38.2
	PAD (CVPR24)	55098.8	39.8	27.2	33.5	39.0	36.0	37.5	43.0	35.9	39.5
	NAPGuard (CVPR24)	384.6	47.8	27.0	37.4	43.9	35.4	39.7	46.2	32.8	39.5
	DIFFender (ECCV24)	7606.0	34.1	19.4	26.8	32.1	26.3	29.2	35.8	25.4	30.6
	NutNet (CCS24)	45.2	42.4	21.4	31.9	36.7	31.5	34.1	41.5	31.1	36.3
	KDAT (AAAI25)	**43.2**	50.1	31.5	40.8	**47.6**	33.3	40.5	**49.9**	34.3	42.1
	AntiStyler (Ours)	92.7	**51.6**	**32.5**	**42.1**	45.8	**38.0**	**41.9**	48.9	**38.3**	**43.6**
DETR	Undefended	39.3	52.8	30.8	41.8	53.6	29.0	41.3	56.8	35.5	46.2
	LGS (WACV19)	705.8	45.3	36.0	40.7	48.2	36.3	42.3	47.3	43.8	45.6
	Grad-Defense (CVPR20)	**39.3**	49.9	37.4	43.7	49.8	41.2	45.5	53.0	40.5	46.8
	SAC (CVPR22)	71.7	51.9	34.6	43.3	53.5	32.9	43.2	56.7	43.1	49.9
	ObjectSeeker (SP23)	4072.4	52.8	35.0	43.9	50.5	36.2	43.4	56.8	40.8	48.8
	PAD (CVPR24)	54721.5	50.6	33.7	42.2	46.1	40.2	43.2	55.2	**46.3**	50.3
	NAPGuard (CVPR24)	378.9	50.8	39.9	45.4	50.6	**41.8**	46.2	56.5	46.0	**51.3**
	DIFFender (ECCV24)	7603.4	39.0	23.6	31.3	38.3	26.9	32.6	41.1	32.2	36.7
	NutNet (CCS24)	42.8	50.7	36.6	43.7	49.2	40.2	44.7	51.9	45.5	48.7
	KDAT (AAAI25)	**39.3**	**53.0**	37.8	45.4	52.7	37.9	45.3	55.9	45.5	50.7
	AntiStyler (Ours)	85.6	**53.0**	**41.7**	**47.4**	**53.6**	39.6	**46.6**	**57.8**	44.4	51.1

Evaluation

Results on INRIA

Defense Method	Benign	TSEA		Printable Patches			Natural Patches			
		B-Patch	C-Patch	OBJ	UPPER	CLS-DET	P1	P2	P3	P4
Undefended	94.9	17.6	17.3	45.6	40.2	59.7	54.6	66.2	53.1	67.4
ObjectSeeker (SP23)	95.1	16.2	17.6	44.9	41.1	61.2	55	60.8	49.6	69
PAD (CVPR24)	94.9	54.2	<u>86.2</u>	<u>78.5</u>	<u>78.7</u>	**80.2**	<u>74.3</u>	**82.3**	**79.2**	<u>81.4</u>
DIFFender (ECCV24)	92.7	57.8	55.1	71.2	64.3	68.8	65.5	57.1	<u>62.7</u>	60.8
NutNet (CCS24)	94.8	74.1	78.3	71.2	64.8	67.8	53.2	65.8	50.5	64.7
KDAT (AAAI25)	<u>95.6</u>	17.8	18.9	51.6	45.0	63.0	53.1	62.7	47.1	67.1
AntiStyler (Ours)	**95.8**	**94.9**	**87.8**	**82.0**	**83.0**	**80.2**	**74.4**	<u>81.9</u>	57.3	**81.7**

Conclusions

- ✓ What makes adversarial patches stand out?
  - We observe that **adversarial patch attacks introduce unnatural, random patterns** that are statistically different from benign image content.
- ✓ How does AntiStyler work?
  - **We reverse the effect of the style transfer technique to remove a "random" style from the image while preserving its content.**
    - Pixels that change the most during this process reveal the likely location of the adversarial patch and are subsequently masked.
- ✓ What do we achieve?
  - **AntiStyler is fully agnostic**: it requires no training, no knowledge of the attacked model, and no prior information about the attack or patch.
  - It **operates in real time** (10–12 FPS) and **improves robustness** by up to 15 mAP, **while maintaining clean-image accuracy**.

Thank you!

Curious?
Check out our paper and demo!

Email Contact:
idanyan@post.bgu.ac.il