The Unseen Adversaries: Robust and Generalized Defense Against Adversarial Patches

Vishesh Kumar, Akshay Agarwal
Trustworthy BiometraVision Lab,
Indian Institute of Science Education and Research Bhopal, India

Abstract

The vulnerabilities of deep neural networks against singularities have raised serious concerns regarding their deployment in the physical world. One of the most prominent and impactful physical-world adversarial perturbations is the attachment of patches to clean images, known as an adversarial patch attack. Similarly, natural noises such as Gaussian and Salt&Pepper are highly prevalent in the real world. The current research need arises from the above vulnerabilities and the lack of efforts to tackle these two singularities independently and, especially, in combination. In this research, we have, for the first time, combined these two prominent singularities and proposed a novel dataset. Using this dataset, we have conducted a benchmark study of singularity data-point detection using features from several convolutional neural networks. For classification, rather than the popular neural network-based parameter tuning, we have used traditional yet effective machine learning classifiers. The extensive experiments across various in- and out-of-distribution (OOD) singularities reveal several interesting findings about the effectiveness of classifiers and show that it is hard to defend against adversaries when they are treated independently, and inefficient classifiers are selected.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

The sensitivity of automated computer systems, including machine learning algorithms, against adversarial attacks is a serious concern [Li et al., 2023]. Among several known adversarial attacks, an adversarial patch attack is one of the stealthiest and most realistic. Physical adversarial patches are small and typically printed as posters or stickers, then placed on target objects in a scene. These patches are also observed to be agnostic to affine transformations, such as translation and rotation, of both target objects and patches. In 2017, [Brown et al., 2017] introduced the concept of adversarial patch attacks and demonstrated how they could be used to fool object detectors in the real world. Since then, several advances have been made in developing sophisticated adversarial patches to fool computer vision models. [Karmon et al., 2018] proposed LaVAN (Localized and visible adversarial noise), which focuses on exploiting the weaknesses of object detectors and creating stealthy patches. Adversarial QR patches [Chindaudom et al., 2020, Chindaudom et al., 2022] are introduced to make patches less suspicious. [Liu et al., 2019a] designed PS-GAN (Perceptual Sensitive Generative Adversarial Networks) to improve adversarial patches' visual quality and effectiveness. [Gittings et al., 2019] used deep image priors to create imperceptible perturbations that can still fool object detectors. [Zhou et al., 2021] proposed DiAP, a data-independent adversarial patch technique. [Lee and Kolter, 2019] improved DPatch for real-world applications and lighting conditions. [Wu et al., 2020] introduced DPAttack (Diffused Patch Attacks) to perturb small image areas effectively, and [Huang et al., 2021] extended this with RPAttack (Refined Patch Attack). [Thys et al., 2019] targeted surveillance cameras, and [Den Hollander et al., 2020] focused on military asset camouflage. [Lu et al., 2021] introduced PatchNoobj to scale patches adaptively. [Li et al., 2019] developed the Dynamic Adversarial Patch for dynamic scenes. [Zolfi et al., 2021] introduced the translucent patch for camera lenses, and [Wang et al., 2021] de-

Table 1: Review of recent papers on adversarial patch attack generation and detection algorithms along with defense against natural noises.

Adversary	Authors	Description
Patch Generation	[Sun et al., 2023]	Diversified Universal Adversarial Patch Generation Method (D-UAP).
	[He et al., 2023]	Imperceptible adversarial patch - Vulnerable target category + target attack.
	[Rasol et al., 2023]	Adaptive adversarial patch-generating network, AAPGNet.
	[Zhou et al., 2023]	Adversarial patch for a set of natural images using AdvEncoder.
	[Tang et al., 2023]	Generate adversarial patches against aerial imagery detectors.
Patch Detection	[Ojaswee et al., 2023]	Fintuned deep neural networks for patch detection.
	[Tarchoun et al., 2023]	A differential entropy analysis to detect adversarial patches.
	[Xu et al., 2023]	PatchZero defence, Zeros out patch region.
	[Kang et al., 2023]	Diffusion-based defense by localization of patch.
	[Liu et al., 2022]	Segment and complete defense to detect and remove the adversarial patch.
	[Xiang et al., 2022]	Twice pixel masking to neutralize patch using PatchCleanser.
	[Kim et al., 2022]	APE masking + APE refinement.
Defense Natural Noise	[Yao et al., 2023]	Interactive self-supervised denoising with user preferences.
	[Wang et al., 2022]	Self-supervised denoising from single noisy image.
	[Yang et al., 2022]	Improve recognition in low-quality images by using self-feature distillation.
	[Zhang et al., 2022b]	Jointly improve restoration using dual exposures.
	[Zhang et al., 2022a]	Unsupervised denoising without clean pairs.

signed the invisibility patch for specific class attacks. [Lennon et al., 2021] introduced mAST (mean Attack Success over Transformations), evaluating patch attack robustness for 3D transformations. [Lang et al., 2021] proposed AGAP using heat maps for patch generation, but faced challenges with feature density in real-world scenarios.

The above review shows that several adversarial patch generation algorithms exist in the literature, and they are effective for almost every computer vision task. However, a lack of patch datasets for benchmarking patch detection significantly limits the development of defense algorithms. Recently, [Pintor et al., 2023] and [Ojaswee et al., 2023] have proposed a benchmark adversarial patch dataset to push research towards defending against these effective physical-world singularities of deep networks. Apart from these artificial adversaries, the real world is also affected by several natural noises that predominantly occur in the real world [Agarwal et al., 2022, Hendrycks et al., 2021, Pei et al., 2019], which the above research ignores. And the combination of these adversarial patches and natural noise singularities further exacerbates neural network degradation, leading to significantly worse performance, as demonstrated in Section 3.1. However, the vulnerabilities of defense algorithms against out-of-distribution samples (e.g., unseen noise or adversarial patches) or the independent handling of singularities result in shallow security algorithms.

While a limited benchmark patch or noise dataset exists, several researchers have prepared in-house datasets to defend against adversarial patches and natural noise. For example, [Gittings et al., 2020] proposed a training-time defense against patch attacks, introducing VaN (Vaxa-Net), which used a DC-GAN (Deep Convolutional Generative Adversarial Network) [Radford et al., 2015] to generate effective adversarial patches and trained models to defend against them. [Radford et al., 2015] developed an adversarial training technique that improved model robustness against adversarial patches without compromising clean accuracy. Most empirical defenses against adversarial patches rely on adversarial training or saliency map inference, but [Cosgrove et al., 2020] introduced a distinct approach: CompNets, an interpretable compositional model that inherently resists occlusions and defends against adversarial patches. [Huang and Li, 2021] developed PatchVeto, a zero-shot certified defense based on Visual Transformers (ViT) [Dosovitskiy et al., 2020], while [Salman et al., 2022] leveraged ViTs for certified patch attacks. Table 1 provides a summary of the recent works on adversarial patch generation, detection, and defense against natural noises.

As mentioned above, the absence of an adversarial patch benchmark dataset limits the development of a unified defense. Furthermore, the combination of adversarial patches and natural noise remains unexplored. Inspired by the above limitations and the impact of both singularities [Kumar et al., 2025, Kumar and Agarwal, 2025], in this research, for the first time we have proposed the singularity dataset, which inherits both adversarial patches and natural noises. We assert that developing such a dataset will advance adversarial patch defense and ensure that the resulting defenses are resilient enough to operate in the noisy, unconstrained physical world. By utiliz-

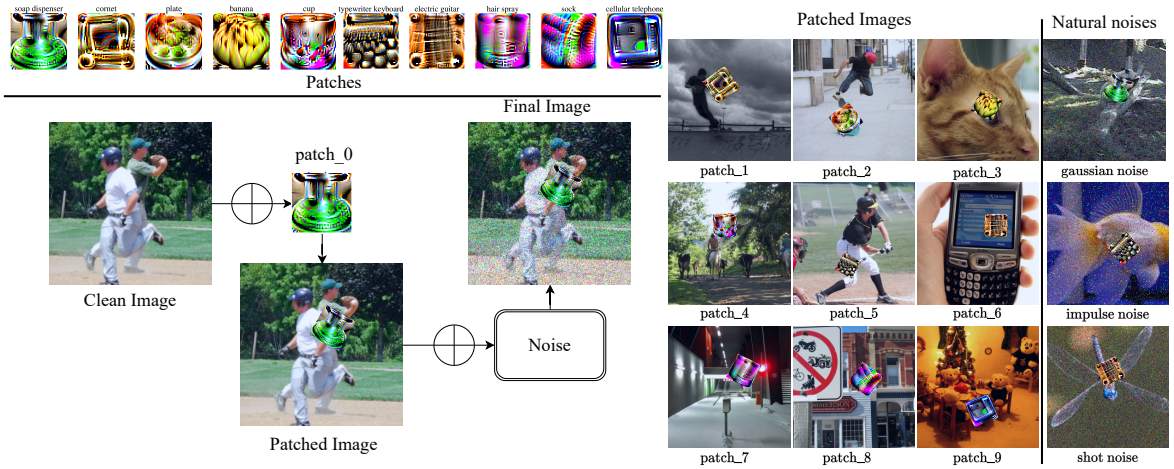

Figure 1: (Left) Overview of the steps involved in the generation of the proposed adversarial patch + natural noise dataset. (Right) Samples images of 10 patches and patch + 3 natural noises of the proposed dataset.

ing the proposed dataset, we have conducted a benchmark study to detect the adversarial patch-based singularity both in the presence and absence of natural noises. In contrast to existing neural network-based defenses, we have plugged in traditional yet effective machine learning classifiers, such as support vector machines, Bayes, and decision forests, for detection. The prime reason is that, firstly, these classifiers are underexplored for adversarial defense, and secondly, they are found to be less susceptible to noisy image alterations than neural networks [Agarwal et al., 2021, Liu et al., 2019b]. In brief, the contributions of this research are: (i) Proposed first-ever patch+noise singularity datasets containing images of multiple variations of physical patches and different natural noises together; and (ii) We comprehensively evaluated different deep neural networks, including the Vision Transformer (ViT) [Dosovitskiy et al., 2020] and machine learning classifiers to ensure the development of a unified and resilient defense algorithm.

We observed that these singularities are not limited to classification models (white or black-box) but are task-agnostic; they can even fool object detection models, even when explicitly developed for classification. Therefore, the proposed benchmark study is an important step towards a unified defense and responsible AI deployment.

2 PROPOSED ADVERSARIAL DATASET

The study introduces two novel datasets focusing on physical adversarial patches and natural noises to address the lack of research on developing a unified defense. For that, we have used two popular object recognition datasets, namely ImageNet [Deng et al., 2009] and COCO [Lin et al., 2014]. The images in these datasets are attacked with 10 different styles of physical adversarial patches and three types of natural noise (*Gaussian, Shot, and Impulse noise*). To construct adversarial patch images for ImageNet, we randomly selected $2,000$ images from the validation set (not cherry-picked). These images act as a clean subset in the proposed dataset. Subsequently, another set of $2,000$ images from ImageNet is selected, and on top of each image, 10 selected adversarial patches [Pintor et al., 2023] are applied. It generates $20,000$ adversarial patch images along with $2,000$ clean images from the ImageNet dataset. A similar procedure has been applied to the images of the COCO dataset, resulting in $2,000$ clean and $20,000$ adversarial patch images.

Table 2: Characteristics of the proposed dataset along with the statistics of images used in training and testing.

Type	Dataset	Adversary	Train	Test	Total
Clean (Real)	ImageNet	–	1200	800	2000
	COCO		1200	800	2000
Attack	ImageNet	10-Patches	12000	8000	20000
	COCO		12000	8000	20000
	ImageNet	3 - Noise	0	2400	2400
	COCO		0	2400	2400
	ImageNet	10-Patch + 3-Noises	0	24000	24000
	COCO		0	24000	24000

To evaluate the robustness of adversarial patch detectors, we have applied natural noise (*Gaussian, shot, and impulse noise*) to the test subset of each dataset. For that, images from clean, individual patches are divided into training and test sets in the ratio $3:2$. For example, there are 2000 clean images of the ImageNet dataset and 2000 images of a single patch. When

we divide the images into a 3 : 2 set, we obtain 800 test images, and applying three natural noises yields 2,400 noisy test, clean, and patch images. Since natural noises are inherently present in the physical world [Agarwal et al., 2020], they are applied only on the test set to ensure adversarial patch detectors are agnostic to unseen (out-of-distribution (OOD)) singularities. Fig. 1 shows the schematic diagram of the proposed patch and patch+noise image generation, and sample images reflecting various patches and noise used in the proposed research. It reflects the challenges in detecting patches due to significant style variations and their tendency to blend into complex image regions.

In total, our dataset contains a diverse set of images, including 4,000 clean images, and 40,000 adversarial patched images (20,000 from ImageNet and 20,000 from COCO). Further, 4,800 test images affected by natural noise and 48,000 images containing both adversarial patches and natural noise are also present to evaluate the robustness. The characteristics of the proposed dataset are summarized in Table 2.

2.1 Experimental Setup

In this research, we first demonstrate the effectiveness of adversarial patches, such as robustness and attack success rate, when combined with natural corruptions, highlighting their impact beyond patch-only scenarios. Then, we have conducted an extensive experimental evaluation across several general settings for the detection of adversarial patches. The first setting is '**seen patch detection**': in this setting, the patch images from the test set are also used for training. For example, the ImageNet dataset is divided into a training and testing subset where 60% of 2000 images are used for training of clean and patch_0 classes and the remaining 40% images of clean and patch_0 are used for testing. In the '**unseen patch detection**' setting, patch images used for testing are not seen at the time of training. For example, 60% of 2000 images are used for training of clean and patch_0 classes, and the remaining 40% images of clean and $\text{patch}_{\neq 0}$ are used for testing. *In the last setting, we evaluated the robustness of detectors trained on clean and individual patch images (without noises) on the test images (clean and patch classes) perturbed with natural noises.* The training and test splits for each protocol are given in Table 2. Under these settings, the detection performance is analyzed along two main factors: (i) the robustness of deep image encoders (e.g., VGG and ViT) when coupled with traditional classifiers, and (ii) the effectiveness of the training patches in identifying adversarial patches under seen and unseen evaluation scenarios.

Inspired by the preliminary results of [Ojaswee et al., 2023], we have used two state-of-the-art convolutional neural networks (CNNs), namely VGG16 [Simonyan and Zisserman, 2014] and Vision Transformer [Dosovitskiy et al., 2020], as a feature extractor. While the authors [Ojaswee et al., 2023] have fine-tuned several CNNs, most are found to be ineffective in generalization settings. Moreover, based on the ineffectiveness of fine-tuning and robustness of traditional classifiers [Agarwal et al., 2023a, Agarwal et al., 2023b, Liu et al., 2019b], we have used several classifiers, including AdaBoost (AB), SGD (Stochastic Gradient Descent), Random Forest (RF), Logistic Regression (LR), and Support Vector Classifier (SVC). These classifiers are trained with the default parameters provided by the Scikit-learn library [Buitinck et al., 2013].

Table 3: Mean robust accuracy (%) of different ImageNet classifiers under adversarial patch attacks and their combination with additive noise corruptions (severity = 2, fixed across all experiments). The lower the value, the better the attack.

Model	Clean Acc.	Patch Only	Patch + Gaussian	Patch + Shot	Patch + Impulse
AlexNet	64.62	8.38	2.04	2.28	**1.39**
ResNet-18	76.08	29.78	3.40	2.86	**1.64**
SqueezeNet-1.0	66.62	10.45	0.41	0.48	**0.30**
ResNet-50	82.70	60.85	19.65	16.68	**12.49**
MobileNet-V2	78.06	51.62	6.86	6.10	**5.30**
VGG-16	78.30	46.45	6.89	5.40	**2.72**
ViT-B/16	85.10	82.68	68.23	64.88	**62.84**
GoogLeNet	76.46	47.39	15.06	14.00	**8.97**
Swin-Tiny	85.74	83.71	66.99	**64.16**	66.09
XceptionNet	74.94	38.88	14.27	12.95	**9.25**

Table 4: Mean attack success rate (%) of adversarial patches from [Pintor et al., 2023] when applied alone and in combination with additive noise corruptions. The higher the value, the better the attack.

Model	Patch Only	Patch + Gaussian	Patch + Shot	Patch + Impulse
AlexNet	26.26	**38.03**	36.89	37.91
ResNet-18	46.11	**78.31**	77.50	77.20
SqueezeNet-1.0	58.00	**77.94**	77.05	75.87
ResNet-50	7.04	27.72	28.48	**30.34**
MobileNet-V2	4.49	30.73	**31.21**	23.82
VGG-16	11.74	39.43	**39.60**	38.88
ViT-B/16	0.71	4.14	4.95	**5.70**
GoogLeNet	10.93	**31.69**	30.74	30.65
Swin-Tiny	0.67	8.93	**10.75**	8.68
XceptionNet	8.33	16.14	**16.79**	15.37

3 RESULT AND ANALYSIS

As discussed above, we first demonstrate the effectiveness of adversarial patches when combined with natural corruptions, analyzing robust accuracy and attack success rate. These results show that patch+noise combinations lead to a stronger degradation in model robustness than patch-only attacks. Building on these findings, we conduct a comprehensive set of experiments on adversarial patch detection in both seen and unseen settings, evaluated with and without natural noise corruptions.

3.1 Attack Effectiveness of Combined Singularities (Patch+Noise)

Table 3 shows the mean robust accuracy of 10 ImageNet classifiers under adversarial patches and their combination with natural noise on the ImageNet100 validation set containing 5000 images. The adversarial patches are taken directly from [Pintor et al., 2023] and originally optimized against the first three architectures in Table 3 (AlexNet, ResNet-18, and SqueezeNet). Despite this, they transfer well across all models: we observe a clear drop in accuracy under patch-only attacks; for example, ResNet-18 drops from 76.08% (clean) to 29.78%, and VGG-16 decreases from 78.30% to 46.45%. When adversarial patches are combined with noise (noise + patch), performance degraded further profoundly, with ResNet-18 values reaching 3.40%, 2.86%, and 1.64% accuracy under patch + Gaussian, shot, and impulse noise, respectively, and SqueezeNet falling below 0.5% across all noise types. Similar trends are observed for other CNNs, such as ResNet-50, which declines from 60.85% under patch-only attacks to 12.49% under patch + impulse noise. In contrast, transformer-based architectures such as ViT-B/16 and Swin-Tiny exhibit markedly higher resilience, retaining a large fraction of their patch-only robustness even under combined patch+noise settings. Among the noise types, impulse noise is consistently the most destructive, followed by shot and Gaussian noise, indicating that sparse, high-magnitude perturbations interact more severely with adversarial patches. These trends suggest that adversarial patches act as strong physical-world singularities, and their interaction with natural noise exposes fundamental architectural biases in convolutional models, while global-attention-based models demonstrate comparatively stronger robustness under such compound distribution shifts, although not ready for the real world.

Table 4 reports the mean attack success rate of adversarial patches from [Pintor et al., 2023] when applied alone and in combination with additive noise corruptions. Consistent with the sharp drop in robust accuracy observed under combined patch+noise settings, the attack success rate increases substantially once applied on noisy images. For example, ResNet-18 exhibits an attack success rate of 46.11% under patch-only attacks, which rises to 78.31%, 77.50%, and 77.20% when combined with Gaussian, shot, and impulse noise, respectively. A similar pattern is observed for SqueezeNet, where attack success increases from 58.00% to over 75% across all patch–noise combinations. Even architectures that appear relatively resistant to patch-only attacks, such as ResNet-50 (7.04%) and MobileNet-V2 (4.49%), show a marked increase

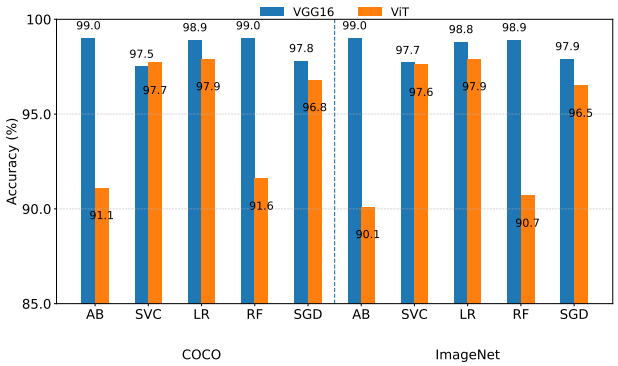

Figure 2: Average adversarial patch attack detection performance on COCO (left) and ImageNet (right) subset under, seen patches evaluation setting. The results showcase the effectiveness of VGG16 **when the patches are seen during training and testing**.

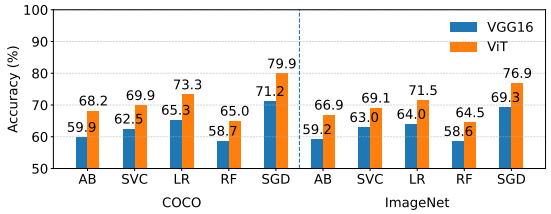

Figure 3: Average adversarial patch attack detection performance on COCO (left) and ImageNet (right) subset under, unseen patches evaluation setting. The results showcase the generalizability of ViT **when the patches are unseen during training and testing**.

in vulnerability when noise is added, reaching up to 30.34% and 31.21% attack success, respectively. In contrast, transformer-based models demonstrate lower absolute attack success rates; however, the same trend holds, with ViT-B/16 increasing from 0.71% (patch-only) to 5.70% under patch + impulse noise, and Swin-Tiny from 0.67% to 10.75% under patch + shot noise. These results indicate that additive noise consistently amplifies the effectiveness of adversarial patches across architectures, suggesting that evaluating patch attacks in isolation understates their practical impact under realistic noisy conditions.

3.2 Adversarial Patch Detection Analysis

This section analyzes the effectiveness of adversarial patch detection under both seen and unseen patch settings. We first study detection performance in noise-free conditions and then evaluate the resiliency of the detectors when patches are combined with natural noise perturbations.

Table 5: A detailed adversarial patch detection accuracy [0-1] of pure convolution (VGG) and attention (ViT) networks in conjunction with traditional classifiers in **seen patch detection** setting.

Dataset	Models	Classifier	Patch 0	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6	Patch 7	Patch 8	Patch 9
COCO	VGG16	AB	**0.99**	**1.00**	**1.00**	**1.00**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
		SGD	0.97	0.97	0.98	0.99	0.97	0.98	0.98	0.98	0.98	0.98
	ViT	AB	0.93	0.86	0.90	0.89	0.90	0.92	0.91	0.93	0.92	0.95
		SGD	**0.99**	0.95	0.98	0.96	0.97	0.95	0.95	0.98	0.98	0.97
ImageNet	VGG16	AB	**0.99**	**0.99**	**0.99**	**1.00**	**0.98**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
		SGD	0.97	0.98	0.98	0.99	0.97	0.97	0.98	0.98	0.98	**0.99**
	ViT	AB	0.90	0.86	0.90	0.86	0.89	0.92	0.90	0.93	0.91	0.94
		SGD	0.98	0.94	0.97	0.95	0.96	0.96	0.96	0.98	0.97	0.98

Table 6: A detailed adversarial patch detection accuracy [0-1] of pure convolution (VGG) and attention (ViT) networks in conjunction with traditional classifiers in **unseen patch detection** setting.

Dataset	Models	Classifier	Metric	Patch 0	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6	Patch 7	Patch 8	Patch 9
COCO	VGG16	LR	Accuracy	0.70	0.65	0.61	0.57	0.70	0.60	0.63	0.71	0.66	0.70
			STD	0.10	0.13	0.16	0.12	0.09	0.14	0.13	0.16	0.14	0.17
		SGD	Accuracy	**0.80**	0.78	0.64	0.59	0.79	0.67	0.71	0.75	0.67	0.72
			STD	0.06	0.11	0.17	0.14	0.07	0.16	0.13	0.18	0.15	0.17
	ViT	LR	Accuracy	0.67	0.81	0.74	0.73	0.83	0.72	0.71	0.72	0.72	0.68
			STD	0.09	0.07	0.10	0.09	0.09	0.10	0.13	0.14	0.15	0.10
		SGD	Accuracy	0.73	**0.86**	**0.78**	**0.81**	**0.85**	**0.80**	**0.79**	**0.78**	**0.76**	**0.81**
			STD	0.08	0.05	0.09	0.07	0.08	0.09	0.12	0.12	0.13	0.08
ImageNet	VGG16	LR	Accuracy	0.71	0.65	0.60	0.55	0.64	0.59	0.64	0.69	0.63	0.70
			STD	0.12	0.15	0.15	0.12	0.08	0.13	0.16	0.17	0.13	0.17
		SGD	Accuracy	**0.73**	0.74	0.63	0.57	0.73	0.65	0.74	0.73	0.68	0.73
			STD	0.11	0.15	0.16	0.13	0.12	0.16	0.15	0.17	0.14	0.17
	ViT	LR	Accuracy	0.64	0.78	0.71	0.77	0.83	0.69	0.71	0.68	0.69	0.65
			STD	0.10	0.09	0.11	0.07	0.10	0.09	0.14	0.14	0.15	0.09
		SGD	Accuracy	**0.73**	**0.82**	**0.74**	**0.81**	**0.84**	**0.75**	**0.75**	**0.76**	**0.76**	**0.73**
			STD	0.10	0.07	0.10	0.07	0.08	0.10	0.12	0.13	0.12	0.10

3.2.1 Without Noise

The results, shown in Fig. (s) 2 and 3, provide a broad analysis of the average adversarial patch detection performance of each network under distinct patch conditions (seen and unseen). It is interesting to note that, under seen conditions, adversarial patches are almost perfectly detected when a pure convolutional neural network (without any form of attention layer) architecture (VGG16) is used. Out of all the classifiers used, the AdaBoost classifier yields the highest accuracy. The observation has been noted in both datasets, indicating that there is no bias in the adversarial patch detection performance. For example, when the AdaBoost (AB) classifier is attached to the features of VGG16, it yields 99.30% and 99.00% average classification performance COCO and ImageNet datasets, respectively. Random Forest (RF) yields the second-highest detection performance in seen-patch detection scenarios, with an accuracy gap of at most 0.30% compared to AdaBoost.

While it is expected that the detection performance of classifiers drops drastically in unseen (OOD) scenarios, surprisingly, the attention-based architecture (ViT) shows significant robustness against adversarial patches in OOD scenarios compared to the pure convolutional architecture (VGG). Further, the AB and RF classifiers, which are found highly effective in seen patch detection settings, exhibit the highest levels of vulnerability/reduction. Moreover, the SGD classifier is found to be highly robust at detecting adversarial patches in unseen patch-detection settings. For example, as shown in Fig. 3, the SGD classifier achieves 79.90% and 76.90% average detection performance when ViT features are used to encode the real and attacked images, respectively. The SGD classifier, whether attached to VGG or ViT, yields the highest detection performance, making it agnostic to the feature extractor. Surprisingly, another simple classifier, namely logistic regression, shows greater robustness than other classifiers such as AdaBoost (AB), support vector machines (SVM), and random forests (RF). *In brief, in terms of classifiers, AdaBoost (AB) and Random Forest (RF) are observed to be highly effective; whereas, in terms of the encoder, VGG16 outperforms ViT. However, in the setting of unseen patches, ViT outperforms VGG, SGD, and Logistic Regression (LR) and is found highly resilient as compared to other classifiers.* It is to note here that we have also evaluated the effectiveness of other deep encoders such as NASNetMobile [Zoph et al., 2018] and Xception [Chollet, 2017], but found VGG outperforms them by a significant margin. For example, the performance of VGG on COCO and ImageNet is at least 5.4% better than that of NASNetMobile. Similarly, other traditional classifiers, such as k-nearest neighbour (KNN), decision

trees, and gradient boosting, are evaluated but found less effective than AB and SGD; hence, results using only the best classifiers are reported in this paper.

Table 5 demonstrates the detailed performance of each encoder using two best-performing classifiers, namely AB and SGD. As mentioned above, in the seen-patch detection setting, perfect patch-detection accuracy is observed when the VGG encoder is used with the AB classifier. When the ViT encoder is used with the SGD classifier, at least 94% accuracy is observed across the dataset, demonstrating that it is easy to detect patches when they are seen at test time. However, *'we believe such high accuracy can provide a false sense of security'* because it is hard to predict the future set of adversarial patches, and hence detection algorithm must be effective under those **zero-shot patches**. Henceforth, to demonstrate how easy (challenging) it is to detect adversarial patches, *we have performed* 10 *fold unseen patch cross-validation experiments*. In this setting, in every fold, images of a single patch along with clean images are used for training, and images of the other nine patches along with clean images are used for testing. The detailed performance of these experiments in terms of average accuracy along with standard deviation (STD) is reported in Table 6.

First and foremost, it is observed that the classifiers that are showing perfect detection performance in the seen setting suffer a drastic reduction in performance in the zero-shot (unseen) patch setting. Secondly, the SGD classifier outperforms the AB classifier in the unseen-patch robustness scenario, and ViT surpasses the VGG encoder's performance. It is also worth noting that ViT, along with SGD, not only yields high average accuracy but also shows lower standard deviation. Among all the patches, patch_1 and patch_4 (Fig. 1) are found to be highly effective at detecting unseen patches and achieve at least 85% and 82% average accuracy on the COCO and ImageNet datasets, respectively.

3.2.2 Resiliency Under Noise Perturbation

In this setting, we have evaluated the resiliency of adversarial patch detectors trained on clean patches (which did not see any form of noise during training). The prime reason is that natural noises are inherent in the environment [Agarwal et al., 2020] and, like patches, every form of noise cannot be used for training; hence, trained detectors must be resilient to handle unseen natural noises. As observed above, VGG and ViT are found to be effective in seen and unseen patches, respectively; only their resiliency is evaluated in the respective settings. The results of the resiliency of VGG and ViT in seen and unseen patch detection settings are reported in Fig. 4. While it is expected

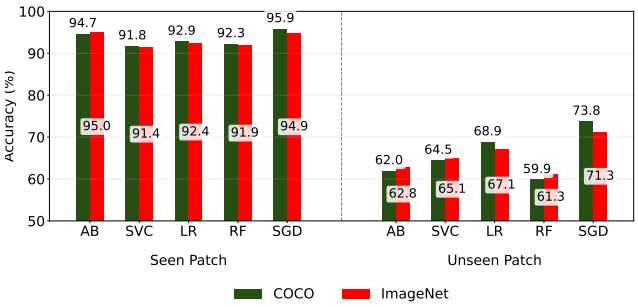

Figure 4: Average adversarial patch attack detection in the presence of noise corruptions. Performances are reported on VGG16 in the seen patch setting (left) and ViT in the unseen patch setting (right).

that the detectors will suffer drops in detection performance, surprisingly, only a marginal reduction is observed. For example, the performance of SGD drops from 97.8% to 95.9% and from 97.9% to 94.9% on the COCO and ImageNet datasets, respectively. Interestingly, in the seen patch evaluation setting under noise, the SGD classifier outperforms the best classifier, i.e., AB found in the seen patch without the noise setting. Similar to seen patch settings, a reduction of $5-6\%$ in the accuracy of SGD is observed in unseen noise patch settings from unseen without noise patch settings. The detailed performance in the seen path noise setting is reported in Table 7, which shows that on the COCO dataset, SGD yields the best performance most of the time; whereas on the ImageNet AB, SGD performs comparably, with an average accuracy difference of 0.1%.

Table 8 showcases the average 10 fold cross-validation performance in the unseen patch noise evaluation setting. Interestingly, when the images (clean and patched) are perturbed and the evaluation has been performed in a zero-shot setting, logistic regression (LR) outperforms the AdaBoost (AB) classifier. However, the SGD classifier plugged into the features of ViT outperforms each classifier and encoder by a significant margin. Overall, through an extensive experimental evaluation, it is observed that the *VGG encoder along with the AB classifier yields the highest effectiveness* in detecting adversarial patches but the constrained is these patches must be seen during detector training. Moreover, the *ViT encoder along with the SGD classifier is not only found generalized in unseen patch evaluation settings but also yields high resiliency* when the images are perturbed through noise corruptions. Therefore, **we assert that in real-world settings, a defender should pick the ViT encoder along with the SGD classifier to defend against adversarial patches**. It is to be noted since the eval-

Table 7: Adversarial patch detection accuracy [0-1] on COCO and ImageNet in seen patch but unseen noise evaluation setting. In other words, the resiliency of the detectors is evaluated when a seen patch perturbed with natural noises comes for classification.

Dataset	Models	Classifier	Patch 0	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6	Patch 7	Patch 8	Patch 9
COCO	**VGG16**	AB	0.92	**0.99**	**0.98**	**0.98**	0.89	0.94	0.97	**0.95**	0.89	0.95
		SGD	**0.94**	0.98	0.96	**0.98**	**0.91**	**0.97**	**0.98**	**0.95**	**0.96**	**0.97**
ImageNet	**VGG16**	AB	**0.95**	**0.98**	**0.98**	**0.98**	**0.88**	0.95	**0.98**	0.96	**0.92**	0.93
		SGD	**0.95**	**0.98**	0.97	0.97	0.83	**0.97**	**0.98**	**0.97**	0.91	**0.98**

Table 8: Adversarial patch detection accuracy [0-1] along with standard deviation (STD) on COCO and ImageNet in an unseen patch and unseen noise evaluation setting. In other words, the 'dual' resiliency of the detectors is evaluated when unseen patches perturbed with natural noises come for classification.

Dataset	Models	Classifier	Metric	Patch 0	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6	Patch 7	Patch 8	Patch 9
COCO	**ViT**	LR	Accuracy	0.58	0.74	0.66	0.67	**0.80**	0.72	0.66	0.67	0.74	0.65
			STD	0.08	0.08	0.10	0.09	0.08	0.10	0.12	0.11	0.12	0.09
		SGD	Accuracy	**0.60**	**0.77**	**0.71**	**0.75**	**0.80**	**0.75**	**0.73**	**0.73**	**0.74**	**0.79**
			STD	0.08	0.05	0.10	0.07	0.06	0.08	0.11	0.12	0.11	0.08
ImageNet	**ViT**	LR	Accuracy	0.59	0.71	0.64	0.70	**0.80**	0.66	0.65	0.68	0.68	0.61
			STD	0.08	0.09	0.10	0.07	0.08	0.09	0.12	0.12	0.14	0.07
		SGD	Accuracy	**0.66**	**0.73**	**0.66**	**0.76**	0.78	**0.70**	**0.68**	**0.73**	**0.75**	**0.68**
			STD	0.09	0.09	0.10	0.06	0.08	0.10	0.12	0.11	0.11	0.09

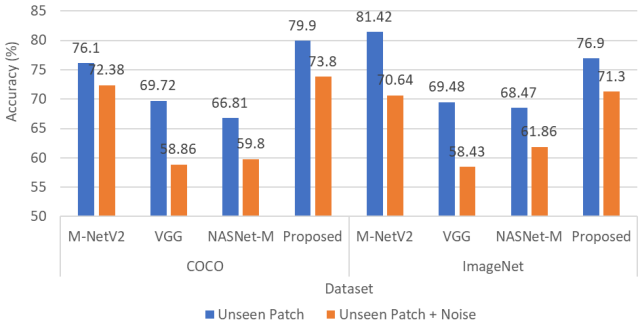

Figure 5: Comparison with the SOTA in terms of average adversarial patch detection accuracy for unseen patch and unseen patch + unseen noise detection.

uation of traditional classifiers and such detailed generalized settings are missing in the literature, the proposed research can pave a path for future research aiming to defend against adversarial patches and noises.

3.3 Comparison with Baseline

As mentioned above, in the literature, limited work has been done so far on adversarial patch detection containing a variety of patches [Ojaswee et al., 2023]. To demonstrate the strength of the proposed work, we have now performed the comparison with recent state-of-the-art (SOTA) adversarial patch detection work by Ojaswee et al. [Ojaswee et al., 2023]. For comparison, we have fine-tuned the models reported as best in their paper, such as MobileNet-V2 (M-NetV2) and NASNet-MobileNet (NASNet-M), and compared them with the proposed generalized and robust detection network (ViT + SGD). As showcased in Fig. 5, the proposed algorithm outperforms each SOTA except M-NetV2 on the ImageNet. However, it is observed that the standard deviation (STD) of the proposed algorithm is significantly lower (3.46%) than that of the best-performing existing model on ImageNet.

4 WHY ROBUST PATCH DETECTION MATTERS?

While, we have comprehensively demonstrated the transferability of adversarial patches across classification networks, one might argue that, since classification models are trained with similar loss functions, they might be inherently vulnerable even if they are not seen during patch generation. Therefore, we explicitly study the transferability of patch+noise attacks against unseen tasks and architectures, i.e., object detection. Furthermore, we demonstrate that robustness inheritance in patch detectors is difficult to achieve with standard data augmentation or noise injection strategies.

4.1 Transferability of Patch+Noise Attacks to Object Detection

To further examine the transferability of the proposed patch–noise attack, we extend our evaluation beyond image classification to the object detection task. All experiments in this setting are conducted on the COCO validation set. Specifically, we assess whether adversarial patches combined with natural noise corruptions remain effective when transferred to object detection architectures. As shown in Table 9, the combined patch–noise attack consistently degrades object detection performance across multiple state-of-the-art object detectors, even though the patches are not op-

Table 9: Vulnerability of state-of-the-art object detection models to adversarial patch attacks and their combination with natural noise. Results are reported as mAP (higher is better) on the COCO validation set.

Model	Clean	Patch	Patch + Noise
Faster R-CNN (ResNet-50)	36.88	33.62	20.89
YOLO-v8	36.69	33.41	15.87
YOLO-v9	51.87	47.49	34.70
YOLO-v10	38.22	35.20	19.35
YOLO-v11	45.66	41.51	25.54

timised for these models or the detection task. Compared to clean inputs, patch-only attacks already reduce mAP, and adding noise leads to a larger drop in all cases. For instance, Faster R-CNN with a ResNet-50 backbone drops from 36.88 mAP on clean images to 33.62 under patch-only attacks, and further to 20.89 when patch and noise are combined. Similar trends are observed for YOLO-based detectors, where YOLO-v8 decreases from 36.69 to 15.87 mAP and YOLO-v11 from 45.66 to 25.54 mAP under patch–noise attacks. These results indicate that the proposed patch–noise combination transfers across unseen architectures and from image classification to object detection.

4.2 Effect of Data Augmentation and Noise Injection on Patch Detection

We further analyze whether standard data augmentation and noise injection strategies can account for the observed robustness and generalization gains. We train our best-performing configuration, using a ViT backbone with an SGD-trained classifier, and evaluate it under generalized settings involving unseen adversarial patches and unseen patch–noise combinations. As reported in Table 10, commonly used augmentation techniques such as RandAugment, Cutout, and strong colour–geometry transformations achieve comparable performance across both evaluation settings. However, none of these approaches consistently outperforms our method across the two metrics. For instance, RandAugment achieves 0.86% accuracy under unseen patch evaluation and 0.78% under unseen patch + noise, while Cutout attains 0.83% and 0.81%, respectively. Our method achieves 0.85% and 0.80%, indicating competitive performance without relying on complex augmentations.

In addition, we study the effect of injecting random noise during training, a common strategy for improving robustness. Using the same ViT-based configuration trained with random noise on the best-performing patch (patch 4), we evaluate on unseen patches (patch IDs 0–9, excluding 4) and unseen noise types. Under this setting, performance decreases from 85.26% to 79.81%, indicating that random noise injection during training does not improve robustness to unseen adversarial patches and noise. These results suggest that the primary gains in our framework stem from the proposed singularity-aware training objective rather than from data augmentation or noise injection alone.

Table 10: Impact of data augmentation on generalization to unseen adversarial patches and robustness under unseen patch + noise settings.

Augmentation	Unseen Patch	Unseen Patch + Noise
RandAugment	0.86	0.78
Cutout	0.83	0.81
Strong Color + Geometry	0.86	0.81
Ours	0.85	0.80

5 CONCLUSION

Adversarial patches are among the strongest forms of adversaries in the physical world, and are agnostic to multiple transformations such as rotation and translation. On a similar note, natural noises such as Gaussian and Impulse noise are inherently present in images due to the unconstrained environment. Due to these stealthy attacks, the development of deep neural networks in the physical world is risky and lacks trustworthiness. Interestingly, the defense against adversarial attacks failed to adequately address adversarial patches. Further, no research addresses patches and noise simultaneously, leaving the existence of a unified defense in jeopardy. Henceforth, in this research, we not only present a benchmark adversarial patch and noise-perturbed dataset but also present a detailed benchmark study. The study reveals several interesting observations and provides a classifier that is an effective defense against adversarial patches and natural noise. In the future, we aim to expand the dataset by incorporating additional manipulations, including adversarial perturbations, to facilitate the development of a universal defence mechanism. Along with the dataset, a novel attention-guided self-supervised patch-detection algorithm will be presented as a unified defense solution.

Acknowledgements

V. Kumar is partially supported through the Visvesvaraya PhD Fellowship.

Dataset Release

The dataset will be released at `https://tbvl22.github.io/website/resources.html`

References

[Agarwal et al., 2022] Agarwal, A., Ratha, N., Vatsa, M., and Singh, R. (2022). Benchmarking robustness beyond lp norm adversaries. In *ECCV*, pages 342–359. Springer.

[Agarwal et al., 2021] Agarwal, A., Singh, R., Vatsa, M., and Ratha, N. (2021). Image transformation-based defense against adversarial perturbation on deep learning models. *IEEE TDSC*, 18(5):2106–2121.

[Agarwal et al., 2023a] Agarwal, A., Vatsa, M., Singh, R., and Ratha, N. (2023a). Corruption depth: Analysis of dnn depth for misclassification. *Neural Networks*.

[Agarwal et al., 2023b] Agarwal, A., Vatsa, M., Singh, R., and Ratha, N. (2023b). Parameter agnostic stacked wavelet transformer for detecting singularities. *Information Fusion*, 95:415–425.

[Agarwal et al., 2020] Agarwal, A., Vatsa, M., Singh, R., and Ratha, N. K. (2020). Noise is inside me! generating adversarial perturbations with noise derived from natural filters. In *IEEE/CVF CVPRW*.

[Brown et al., 2017] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. *arXiv preprint arXiv:1712.09665*.

[Buitinck et al., 2013] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML-PKDDW*, pages 108–122.

[Chindaudom et al., 2020] Chindaudom, A., Siritanawan, P., Sumongkayothin, K., and Kotani, K. (2020). Adversarialqr: An adversarial patch in qr code format. In *IEEE ICIEV and icIVPR*, pages 1–6.

[Chindaudom et al., 2022] Chindaudom, A., Siritanawan, P., Sumongkayothin, K., and Kotani, K. (2022). Surreptitious adversarial examples through functioning qr code. *Journal of Imaging*, 8(5):122.

[Chollet, 2017] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *IEEE CVPR*, pages 1251–1258.

[Cosgrove et al., 2020] Cosgrove, C., Kortylewski, A., Yang, C., and Yuille, A. (2020). Robustness out of the box: Compositional representations naturally defend against black-box patch attacks. *arXiv preprint arXiv:2012.00558*.

[Den Hollander et al., 2020] Den Hollander, R., Adhikari, A., Tolios, I., van Bekkum, M., Bal, A., Hendriks, S., Kruithof, M., Gross, D., Jansen, N., Perez, G., et al. (2020). Adversarial patch camouflage against aerial detection. In *SPIE AIML in defense applications II*, volume 11543, pages 77–86.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Gittings et al., 2019] Gittings, T., Schneider, S., and Collomosse, J. (2019). Robust synthesis of adversarial visual examples using a deep image prior. *arXiv preprint arXiv:1907.01996*.

[Gittings et al., 2020] Gittings, T., Schneider, S., and Collomosse, J. (2020). Vax-a-net: Training-time defence against adversarial patch attacks. In *ACCV*.

[He et al., 2023] He, C., Zhao, M., Li, Y., and Jiang, W. (2023). Generating imperceptible adversarial patch based on vulnerable targeted attack. In *IEEE ICIPCA*, pages 910–914.

[Hendrycks et al., 2021] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. (2021). Natural adversarial examples. In *IEEE/CVF CVPR*, pages 15262–15271.

[Huang et al., 2021] Huang, H., Wang, Y., Chen, Z., Tang, Z., Zhang, W., and Ma, K.-K. (2021). Rpattack: Refined patch attack on general object detectors. In *IEEE ICME*, pages 1–6.

[Huang and Li, 2021] Huang, Y. and Li, Y. (2021). Zero-shot certified defense against adversarial patches with vision transformers. *arXiv preprint arXiv:2111.10481*.

[Kang et al., 2023] Kang, C., Dong, Y., Wang, Z., Ruan, S., Su, H., and Wei, X. (2023). Diffender: Diffusion-based adversarial defense against patch attacks in the physical world. *arXiv preprint arXiv:2306.09124*.

[Karmon et al., 2018] Karmon, D., Zoran, D., and Goldberg, Y. (2018). Lavan: Localized and visible adversarial noise. In *ICML*, pages 2507–2515.

[Kim et al., 2022] Kim, T., Yu, Y., and Ro, Y. M. (2022). Defending physical adversarial attack on object detection via adversarial patch-feature energy. In *ACM MM*, pages 1905–1913.

[Kumar and Agarwal, 2025] Kumar, V. and Agarwal, A. (2025). A unified, resilient, and explainable adversarial patch detector. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30387–30397.

[Kumar et al., 2025] Kumar, V., Shukla, S., and Agarwal, A. (2025). Robustness benchmarking of convolutional and transformer architectures for image classification. *IEEE Transactions on Big Data*, 11(6):3330–3341.

[Lang et al., 2021] Lang, D., Chen, D., Shi, R., and He, Y. (2021). Attention-guided digital adversarial patches on visual detection. *SCN*, 2021:1–11.

[Lee and Kolter, 2019] Lee, M. and Kolter, Z. (2019). On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*.

[Lennon et al., 2021] Lennon, M., Drenkow, N., and Burlina, P. (2021). Patch attack invariance: How sensitive are patch attacks to 3d pose? In *IEEE/CVF ICCV*, pages 112–121.

[Li et al., 2019] Li, J., Schmidt, F., and Kolter, Z. (2019). Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *ICML*, pages 3896–3904.

[Li et al., 2023] Li, Y., Wu, D., and Wang, S. (2023). Future-generation attack and defense in neural networks.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*, pages 740–755.

[Liu et al., 2019a] Liu, A., Liu, X., Fan, J., Ma, Y., Zhang, A., Xie, H., and Tao, D. (2019a). Perceptual-sensitive gan for generating adversarial patches. In *AAAI*, volume 33, pages 1028–1035.

[Liu et al., 2022] Liu, J., Levine, A., Lau, C. P., Chellappa, R., and Feizi, S. (2022). Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *IEEE/CVF CVPR*, pages 14973–14982.

[Liu et al., 2019b] Liu, J., Zhang, W., Zhang, Y., Hou, D., Liu, Y., Zha, H., and Yu, N. (2019b). Detection based defense against adversarial examples from the steganalysis point of view. In *IEEE/CVF CVPR*, pages 4825–4834.

[Lu et al., 2021] Lu, M., Li, Q., Chen, L., and Li, H. (2021). Scale-adaptive adversarial patch attack for remote sensing image aircraft detection. *Remote Sensing*, 13(20):4078.

[Ojaswee et al., 2023] Ojaswee, O., Agarwal, A., and Ratha, N. (2023). Benchmarking image classifiers for physical out-of-distribution examples detection. In *IEEE/CVF ICCV*, pages 4427–4435.

[Pei et al., 2019] Pei, Y., Huang, Y., Zou, Q., Zhang, X., and Wang, S. (2019). Effects of image degradation and degradation removal to cnn-based image classification. *IEEE TPAMI*, 43(4):1239–1253.

[Pintor et al., 2023] Pintor, M., Angioni, D., Sotgiu, A., Demetrio, L., Demontis, A., Biggio, B., and Roli, F. (2023). Imagenet-patch: A dataset for benchmarking machine learning robustness against adversarial patches. *PR*, 134:109064.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[Rasol et al., 2023] Rasol, J., Xu, Y., Zhang, Z., Zhang, F., Feng, W., Dong, L., Hui, T., and Tao, C. (2023). An adaptive adversarial patch-generating algorithm for defending against the intelligent low, slow, and small target. *Remote Sensing*, 15(5):1439.

[Salman et al., 2022] Salman, H., Jain, S., Wong, E., and Madry, A. (2022). Certified patch robustness via smoothed vision transformers. In *IEEE/CVF CVPR*, pages 15137–15147.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Sun et al., 2023] Sun, L., Wang, X., Yang, Y., and Mao, X. (2023). D-uap: Initially diversified universal adversarial patch generation method. *Electronics*, 12(14):3080.

[Tang et al., 2023] Tang, G., Jiang, T., Zhou, W., Li, C., Yao, W., and Zhao, Y. (2023). Adversarial patch attacks against aerial imagery object detectors. *Neurocomputing*, 537:128–140.

[Tarchoun et al., 2023] Tarchoun, B., Ben Khalifa, A., Mahjoub, M. A., Abu-Ghazaleh, N., and Alouani, I. (2023). Jedi: Entropy-based localization and removal of adversarial patches. In *IEEE/CVF CVPR*, pages 4087–4095.

[Thys et al., 2019] Thys, S., Van Ranst, W., and Goedemé, T. (2019). Fooling automated surveillance cameras: adversarial patches to attack person detection. In *IEEE/CVF CVPRW*, pages 0–0.

[Wang et al., 2021] Wang, Y., Lv, H., Kuang, X., Zhao, G., Tan, Y.-a., Zhang, Q., and Hu, J. (2021). Towards a physical-world adversarial patch for blinding object detection models. *Information Sciences*, 556:459–471.

[Wang et al., 2022] Wang, Z., Liu, J., Li, G., and Han, H. (2022). Blind2unblind: Self-supervised image denoising with visible blind spots. In *IEEE/CVF CVPR*, pages 2027–2036.

[Wu et al., 2020] Wu, S., Dai, T., and Xia, S.-T. (2020). Dpattack: Diffused patch attacks against universal object detection. *arXiv preprint arXiv:2010.11679*.

[Xiang et al., 2022] Xiang, C., Mahloujifar, S., and Mittal, P. (2022). {PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier. In *USENIX Security*, pages 2065–2082.

[Xu et al., 2023] Xu, K., Xiao, Y., Zheng, Z., Cai, K., and Nevatia, R. (2023). Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *IEEE/CVF WACV*, pages 4632–4641.

[Yang et al., 2022] Yang, Z., Dong, W., Li, X., Wu, J., Li, L., and Shi, G. (2022). Self-feature distillation with uncertainty modeling for degraded image recognition. In *ECCV*, pages 552–569. Springer.

[Yao et al., 2023] Yao, M., He, D., Li, X., Li, F., and Xiong, Z. (2023). Towards interactive self-supervised denoising. *IEEE TCSVT*.

[Zhang et al., 2022a] Zhang, Y., Li, D., Law, K. L., Wang, X., Qin, H., and Li, H. (2022a). Idr: Self-supervised image denoising via iterative data refinement. In *IEEE/CVF CVPR*, pages 2098–2107.

[Zhang et al., 2022b] Zhang, Z., Xu, R., Liu, M., Yan, Z., and Zuo, W. (2022b). Self-supervised image restoration with blurry and noisy pairs. *NeurIPS*, 35:29179–29191.

[Zhou et al., 2021] Zhou, X., Pan, Z., Duan, Y., Zhang, J., and Wang, S. (2021). A data independent approach to generate adversarial patches. *MVA*, 32:1–9.

[Zhou et al., 2023] Zhou, Z., Hu, S., Zhao, R., Wang, Q., Zhang, L. Y., Hou, J., and Jin, H. (2023). Downstream-agnostic adversarial examples. In *IEEE/CVF ICCV*, pages 4345–4355.

[Zolfi et al., 2021] Zolfi, A., Kravchik, M., Elovici, Y., and Shabtai, A. (2021). The translucent patch: A physical and universal attack on object detectors. In *IEEE/CVF CVPR*, pages 15232–15241.

[Zoph et al., 2018] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *IEEE CVPR*, pages 8697–8710.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable] **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable] **Yes**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable] **Not Applicable**
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable] **Not Applicable**
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable] **Not Applicable**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable] **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable] **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable] **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable] **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] **Yes**
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable] **Not Applicable**
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable] **Not Applicable**
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable] **Not Applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable] **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable] **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable] **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable] **Not Applicable**