Defensive Patches for Robust Recognition in the Physical World

Anonymous CVPR submission

Paper ID 1680

Abstract

To operate in real-world high-stakes environments, deep learning systems have to endure noises that have been continuously thwarting their robustness. Data-end defense, which improves robustness by operations on input data instead of modifying models, has attracted intensive attention due to its high feasibility in practice. However, previous data-end defenses show low generalization against diverse noises and weak transferability across multiple models. Motivated by the fact that robust recognition depends on both local and global features, we propose a defensive patch generation framework to address these problems by helping models better exploit these features. For the generalization against diverse noises, we inject class-specific identifiable patterns into a confined local patch prior, so that defensive patches could preserve more recognizable features towards specific classes, leading models for better recognition under noises. For the transferability across multiple models, we guide the defensive patches to capture more global feature correlations within a class, so that they could activate model-shared global perceptions and transfer better among models. Our defensive patches show great potentials to improve model robustness in practice by simply sticking them around target objects. Extensive experiments show that we outperform others by large margins (improve 20+% accuracy for both adversarial and corruption robustness on average in the digital and physical world).[1]

1. Introduction

Though deep neural networks (DNNs) have achieved significant successes in multiple areas, their robustness is challenged by noises, especially in physical world scenarios. Adversarial noise, an imperceptible perturbation designed to mislead the decision of DNNs, is now becoming a great threats [9,44]. Besides adversarial attacks, DNNs also show weak robustness against common corruptions in the daily environment (*e.g.*, snow, rain, brightness *etc*) [14, 15].

[1]Our codes are available in the Supplementary Material.

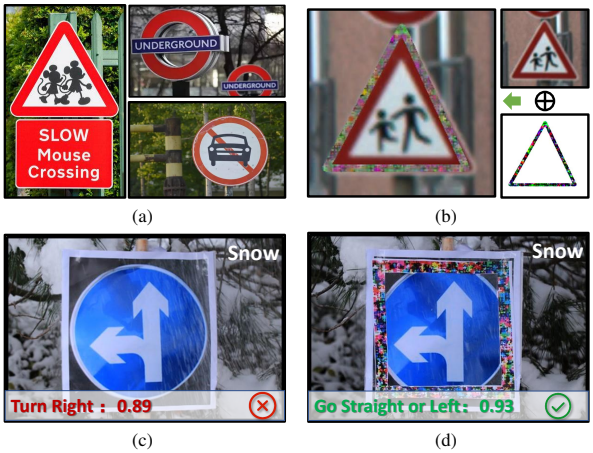


Figure 1. (a) Different guideboards in the physical world. (b) Samples with generated defensive patches in the digital world. (c) The model prediction is misled into `Turn Right` when it is snowy in the physical world. (d) Defensive patches can help models to conduct correct predictions during snow in the physical world.

For example, the guide boards will be incorrectly classified as `Turn Right` when it is snowy (Figure (1c)). What's worse, these inevitable noises have caused dozens of self-driving accidents with casualties and are casting a shadow over the deep learning applications in practice [22]. This urges us to investigate feasible defenses for building robust deep learning models in the physical world.

In the past years, a great number of efforts have been made to defend against the adversarial perturbations and further improve model robustness [12, 26, 31, 38, 54]. Most of the existing works focus on enhancing robustness from model-end (*e.g.*, data augmentation, adversarial training), which require an additional cost of the model architecture modification or model retraining. In contrast, another line of studies performs defenses from the data-end without imposing any model modification (*e.g.*, input transformation), which has shown great potential in practice [37, 51]. For example, by simply sticking a patch on the traffic sign, our proposed defensive patch can help DNNs to make robust recognition under noises (Figure 1d). Though showing great application potential, existing data-end defenses show

several limitations when applied in practice: (1) Weak generalization for diverse noises. Existing works show a significant drop when facing different unseen noises (*e.g.*, adversarial attacks, common corruptions). (2) Low transferability across multiple models. In other words, these works fail to perform defenses for black-box models and even being counteractive. We attribute this phenomenon to the underutilization of robust recognition characteristics.

To address the problems mentioned above, this paper proposes a data-end defensive patch generation framework, which could be effective against diverse noises and work among different models (Figure (1b) and Figure (1d)). Previous studies have revealed strong evidence that robust recognition highly depends on the exploitation of local and global features [34, 35, 53], we thereof improve the defense ability of our defensive patches by promoting better exploitation of both local and global features. Regarding the generalization against diverse noises, since deep learning models rely strongly on the local patterns for predictions [17, 24, 29], we optimize the locally confined patch priors to contain more class-specific identifiable patterns via reducing model uncertainty. Based on these class-level patch priors, the defensive patches can preserve more recognizable features for a specific class and help models to better resist the influence of different noises, *i.e.*, better generalization. As for the transferability across multiple models, recent studies found that different models share similar global perception during decision-making [2, 21, 46], we thus guide the defensive patches to capture more class-wise global feature correlations. In other words, the defensive patches could contain more global features correlated to the class. Thus, the generated defensive patches could better activate the model-shared global perception and enjoy stronger transferability among multiple models. In conclusion, our main contributions can be summarized as:

- To the best of our knowledge, we are the first to generate data-end defensive patches that could improve model robustness against diverse noises (adversarial attacks and corruptions) among different models.
- Our defensive patches improve robustness by injecting local identifiable patterns and enhancing global perceptual correlations, which can be easily deployed via sticking them around target objects.
- Extensive experiments show that our defensive patch outperforms others by large margins (**+20%** accuracy for both adversarial and corruption robustness on average in digital and physical world).

2. Related Work

2.1. Adversarial Attacks

Extensive studies have shown that deep learning models are highly vulnerable to adversarial attack [9, 44]. These imperceptible perturbations could easily make DNNs misclassify the input images. Besides adversarial perturbations, adversarial patches are designed to attack DNNs by attaching additional stickers for their feasibility in the physical world [1, 4, 25, 27, 28, 46], including patches [1], camouflages [46], and light [5]. [1] proposes the first adversarial patch generation strategy, revealing the possibility of generating physical adversarial examples. [46] generates patch-like adversarial camouflage in a 3D environment by suppressing model and human attention. Some researchers aim to perform adversarial attacks in the physical world with adversarial lights (*e.g.*, laser [5] and infrared light [55]).

Besides adversarial attacks, there exist another type of noise named common corruptions, which are commonly-witnessed natural noises, *e.g.*, blur, snow, and frost, *etc*. A line of works has been devoted to studying the influence of common corruptions for DNNs by various approaches [14–16]. [15] proposes a challenging datasets on ImageNets (*i.e.*, ImageNet-C), which contain 15 different types of common corruptions. [16] find that unmodified examples can mislead various unseen models reliably. In summary, the robustness of DNNs is highly challenged by the diverse noises in the physical world, which urges us to improve the model robustness and applicability.

2.2. Adversarial Defenses

Adversarial defenses aim to improve the model robustness against adversarial attacks, which play an important role in increasing the availability of DNNs in the real world. Recent studies indicate that there exists three mainstreams in adversarial defenses: (1) Gradient masking, which aims to hide the key information of the model (*i.e.*, gradients), including defensive distillation [33], shattered gradients [11], randomized gradient [3], *etc* [38, 42]. (2) Adversarial training, which improves the model robustness through adversarially training the classifier with adversarial examples [9, 26, 31, 40, 45, 47, 49, 52]. (3) Adversarial example detection, which aims to distinguish whether the input is clean or adversarial example [8, 10, 20]. The above-mentioned studies primarily focus on improving model robustness from the model-end, which requires the model architecture modification or model re-training. Besides, there also exists another type of defense, which is more feasible in the real world by modifying the input data (*i.e.*, data-end defenses), such as input transformation [50] or image compression [19].

In this paper, we focus on the data-end defense and design a defensive patch to improve model robustness against diverse noises in the physical world.

3. Approach

In this section, we first give the definition of the defensive patch and then elaborate on the proposed framework.

3.1. Problem Definition

A perturbed example x', which consists of a clean example and additional noises, can mislead a given deep neural network $\mathbb{F}$ into wrong prediction, *i.e.*, $\mathbb{F}(x) \neq \mathbb{F}(x')$. Given a k-class dataset $\mathcal{X} = \mathcal{X}^1 \cup \mathcal{X}^2 \cup \cdots \cup \mathcal{X}^k$, the clean example $x \in \mathcal{X}$ and its corresponding perturbed example x' are subject to a ϵ-constraint. Base on the above knowledge, we now provide the definition of defensive patch δ as

$$\mathbb{F}(x) = \mathbb{F}(x' \oplus \delta) \quad s.t. \quad \|\delta\| \leq \varepsilon, \tag{1}$$

where $\|\cdot\|$ is a distance metric which is often measured by ℓ_p-norm ($p \in \{1, 2, \infty\}$), and ε is a constraint value. The defensive patch δ also satisfies the $\mathbb{F}(x) = \mathbb{F}(x \oplus \delta)$ constraint. And the operation $\oplus$ obeys the following equation

$$x \oplus \delta = (\mathbf{1} - \mathbf{M}) \odot x + \mathbf{M} \odot \delta, \tag{2}$$

where $\odot$ is the element-wise multiplication and $\mathbf{M}$ is a shape mask to decide the masking position and appearance.

3.2. Framework Overview

Previous studies have indicated that robust recognition shows high dependence on the combination of local and global features [34, 35, 53], we thus propose to generate defensive patches with strong noise generalization and model transferability by helping models for the better exploitation of local and global features. Thus, our defensive patches can significantly improve the robustness of recognition. The overall framework is shown in Figure 2.

Regarding the generalization against diverse noises, inspired by the fact that deep learning models recognition depends heavily on local patterns [17, 24, 29], we inject more class-specific identifiable patterns into the confined local patch prior. Thus, the defensive patch optimized from the patch prior could preserve more class-specific recognizable features, which could lead the model to better recognition under diverse noises. As for the transferability across multiple models, since different models focus on the similar global perception when making decisions [2, 21, 46], we guide the defensive patches to capture more global feature correlations within a class using Gram matrix in an ensemble way. Thus, our defensive patches could better activate model-shared global perceptions and show stronger transferability among models.

3.3. Local Identifiable Patterns Guidance

Several previous studies have pointed out that deep learning models show a strong dependence on local patterns [17, 24, 29], *e.g.*, local patterns are exploited to improve the emotion recognition ability of the model [17]. Therefore, we aim to inject more class-specific identifiable patterns into the confined local patch prior. The defensive patch optimized from the prior can be treated as a typical class-specific representation [27], hence helping the model for better recognition under different noises.

In practice, we first consider the shape of the local patch prior. Since the defensive patch is designed to improve model robustness in the real-world scenario, it is necessary to evade the influence for human vision (*i.e.*, without covering the target object). Thus, we set the shape mask $\mathbf{M}$ in Equation 2 as a w-pixels square box surrounding the target object (*i.e.*, like guideboard border). Thus, the initial patch prior δ is reformulated as

$$\delta = \mathbf{0} + \mathbf{M} \odot \mathbf{1}, \tag{3}$$

where the $\mathbf{0}$ and $\mathbf{1}$ are respectively a tensor in which each element is 0 or 1, and their dimensions are the same with input size of $\mathbf{M}$. Note that the position mask can be replaced with any different shapes based on the scenario (see studies in Section 4.5.3).

To inject more class-specific identifiable patterns into the confined local prior, we borrow a pattern extraction model to optimize the patch prior by an entropy-based loss function. Since the entropy is widely used to depict the class uncertainty, *i.e.*, higher entropy indicates higher uncertainty to recognize the object. We thus force the patch priors to reduce the entropy of a certain class, *i.e.*, making it more recognizable for a specific class. In this way, the defensive patches can be optimized to contain more class-specific identifiable patterns and resist the influence of different noises. In particular, given a pattern extraction model $\mathbb{M}$ and specific class index k, we optimize the δ_0^k (initialized as δ) by calculating the identifiable pattern loss $\mathcal{L}_p$ as

$$\mathcal{L}_p = -\log \mathcal{P}_{\mathbb{M}}(\delta_0^k), \tag{4}$$

where $\mathcal{P}_{\mathbb{M}}$ is the prediction value of $\mathbb{M}$ with class index k. It is important to note that no input data is needed in the patch prior generation process and $\mathbb{M}$ could be any pre-trained model for this task.

Due to the fact that the typical patch priors contain more identifiable patterns, the defensive patches optimized from these priors could preserve more recognizable features towards a specific class and show better generalization against different noises. After the patch prior generation, we exploit the typical patch prior δ_0^k in the following defensive patch optimization procedure.

3.4. Global Perceptual Correlation Enhancement

Motivated by the fact that models often share similar global perception when making correct predictions towards a specific class [2, 21, 46], we aim to improve the transferability among different models by better activating the model global perception.

Since the context of the target objects is essential as well for making a correct perception [7], we make the defensive

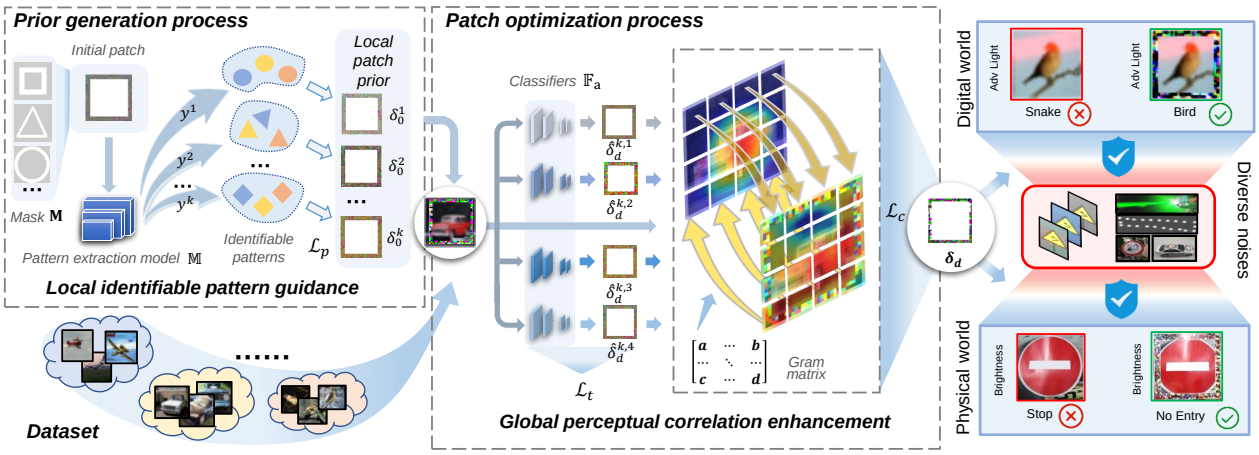


Figure 2. Defensive patch generation framework. We first generate a typical patch prior for each category and inject more class-specific identifiable patterns into the confined local patch from the local viewpoint. Further, we help the defensive patches to contain more global features correlated to each category in an ensemble way from the global viewpoint. Therefore, the generated defensive patches enjoy both strong generalization and transferability.

patches capture more globally contextual features within a certain class. Considering the fact that Gram matrix can be exploited to represent the correlations of features within an image [6], we design a global correlation loss based on Gram matrix by introducing stronger global feature correlations with respect to the class in an ensemble approach. Therefore, the generated defensive patches can better activate the model global perception and achieve stronger transferability among different models.

Specifically, given classifiers $\mathbb{F}_i$ $(i = 1, 2, ..., \mathrm{N})$, we first initialize the defensive patch δ_d^k of the i-th class as δ_0^k; we then optimize the intermediate defensive patch of the i-th classifier $\hat{\delta}_d^{k,i}$ from the δ_d^k using clean examples $x \in \mathcal{X}^k$ based on $\mathcal{L}_t$ as

$$\mathcal{L}_t = y^k \cdot \log \mathcal{P}_{\mathbb{F}_i}(x \oplus \hat{\delta}_d^{k,i}), \tag{5}$$

where y^k denotes the ground-truth label of x, $\mathcal{P}_{\mathbb{F}_i}$ denotes the prediction value of $\mathbb{F}_i$ with the input $x \oplus \hat{\delta}_d^{k,i}$.

We then optimize our defensive patches by exploiting the most similar global perception shared among these intermediate patches from different models. In detail, we introduce the Gram matrix to optimize δ_d^k based on the combination of multiple $\hat{\delta}_d^{k,i}$ by perceptual correlation loss $\mathcal{L}_c$ as

$$\begin{aligned} \mathcal{L}_c &= \frac{1}{\mathrm{N}} \sum_{i}^{N} \|\mathbf{G}(x \oplus \delta_d^k) - \mathbf{G}(x \oplus \hat{\delta}_d^{k,i})\|_2^2, \\ \mathbf{G}_{p,q}(\mathbf{I}) &= \sum_{c} \mathbf{I}_{pc} \cdot \mathbf{I}_{qc}, \end{aligned} \tag{6}$$

where $\mathbf{G}_{p,q}(\mathbf{I})$ means the Gram matrix value of input $\mathbf{I}$ at position (p, q), and $\mathbf{I}_{.c}$ indicates the pixel value of the input $\mathbf{I}$ at channel c. We conduct the above defensive patch generation process in a progressive manner, and the optimized defensive patch during each iteration will serve as the prior for the next iteration.

Besides, it should be noted that this optimization process could also work under the single model setting, *i.e.*, N=1 (See experiments in Section 4.2). We hypothesize the reason might be that as the high-order interaction, the global perceptual correlation perceived by a model plays an important role in robust recognition against attacks [36].

To sum up, through enhancing the global perceptual correlations in an ensemble approach, the generated defensive patches can enjoy stronger transferability across multiple models by activating model-shared global perception.

3.5. Overall Training Process

We generate the defensive patch by serially conducting two optimization processes, *i.e.*, generating the typical patch prior by the identifiable pattern loss $\mathcal{L}_p$ and optimizing the defensive patch by the training loss $\mathcal{L}_t$ and the perceptual correlation loss $\mathcal{L}_c$.

Specifically, for each class, we first initialize all typical patch priors as δ. Then we optimize the patch prior of the $k-$th class by minimizing $\mathcal{L}_p$ with a pattern extraction model $\mathbb{M}$ and the local position constraint $\mathbf{M}$. Furthermore, we employ N different models to conduct an ensemble-based perceptual correlation enhancement optimization. In detail, for each epoch, we obtain N intermediate defensive patches $\hat{\delta}_d^{k,i}$ by maximizing the training loss $\mathcal{L}_t$. After that, we minimize $\mathcal{L}_c$ to generate the defensive patches δ_d^k and then perform defenses by simply using them as additional ornaments. Note that we set the model number N as 4 in this paper. The overall training algorithm can be described in Algorithm 1.

Algorithm 1 Defensive Patch Generation

Input: dataset $\mathcal{X} = \{\mathcal{X}^1 \cup \mathcal{X}^2 \cup \cdots \cup \mathcal{X}^k\}$, shape mask $\mathbf{M}$, pattern extraction module $\mathbb{M}$, classifiers $\mathbb{F}_{i=1,2,3,4}$.
Output: defensive patch $\boldsymbol{\delta}_d$.

initialize patch prior $\boldsymbol{\delta}_0$ as δ;
for k in number of classes **do**
  for the number of prior-training steps **do**
    optimize the δ_0^k by minimize $\mathcal{L}_p$;
  end for
end for
initialize defensive patch $\boldsymbol{\delta}_d$ as $\boldsymbol{\delta_0} = \{\delta_0^0, \delta_0^1, ..., \delta_0^k\}$;
for the number of patch-training epochs **do**
  for k in number of classes **do**
    for i in number of classifiers **do**
      for all $x \in \mathcal{X}^k$ **do**
        optimize the $\hat{\delta}_d^{k,i}$ by maximize $\mathcal{L}_t$;
      end for
    end for
    for all $x \in \mathcal{X}^k$ **do**
      optimize the δ_d^k by minimize $\mathcal{L}_c$;
    end for
  end for
end for

4. Experiments

In this section, we first illustrate our experimental settings, then evaluate the effectiveness of our defensive patch in both the digital and physical world.

4.1. Experimental Settings

Datasets and models. For the dataset, we choose the widely-used CIFAR-10 [23] and GTSRB (guideboard classification dataset) [43]. Regarding the models, we select the commonly used architectures including VGG-16 (denote "VGG") [41], ResNet-50 (denote "RNet") [13], ShuffleNet-V2 (denote "SNet") [30], and MobileNet-V3 (denote "MNet") [18].

Diverse noises. In this paper, we employ 3 types of noises which are realizable in the physical world, *e.g.*, corruptions [15], AdvP [1], and AdvL [5]. Specifically, for corruptions, we adopt the strategies from [15] and implement 16 kinds of corruptions such as fog, rainy, Gaussian, and light, *etc*. For each corruption, we select 5 different intensities.

Evaluation metrics and compared baselines. To evaluate the performance of our proposed method, we choose the widely used metric *accuracy* as the evaluation metric (the higher the better) following [37]. As for the compared baselines, we employ UnAdv [37] and Trans [50], which are the state-of-the-art data-end defenses. We use their released codes for implementation and select reasonable settings for fair comparisons.

Implementation details. For the hyper-parameter a, we set it as 4, which means 4 different models are employed. For the shape mask $\mathbf{M}$, we design a w-pixels bold box surrounding the object which constrains the patch size to 1/5 of image size following one of the implementations in [37]. The backbone of the pattern extraction model $\mathbb{M}$ is VGG-19 [41]. During the prior and patch generation process, we use Adam optimizer with the learning rate of 0.01, weight decay of 10^{-4}, and a maximum of 20 epochs. All codes are implemented in PyTorch. We conduct the training and testing processes on an NVIDIA GeForce RTX 2080Ti GPU cluster[2].

4.2. Digital World Evaluation

In this section, we first evaluate the performance of our generated defensive patches in the digital world. Note that we select the public dataset CIFAR-10 to conduct digital world experiments.

Since our defensive patch generation framework employs several different models, it is unfair to directly compare our method with other baselines. Therefore, we conduct 2 different experiments respectively: (1) we generate our defensive patch by only using the same single target model with UnAdv; (2) we perform similar ensemble training for both UnAdv. Since Trans performs defenses without requiring target models, we directly report its results[2].

According to Table 1 and Table 2, we can conclude that our defensive patches show better performance for improving model robustness, *i.e.*, generalization against diverse noises and transferability among different models. We provide several conclusions as follows:

(1) For generalization against diverse noises, it can be observed that our defensive patches achieve higher accuracy under almost all noises. For example, for the single model setting on RNet, our method yields up to **10.81%** improvement compared with UnAdv under white-box settings; for ensemble setting on MNet, we outperform Unadv and Trans up to **44.11%** and **22.88%**, respectively.

(2) For transferability among different models, it can be clearly illustrated from Table 1 that our proposed defensive patch show higher accuracy values compared with Unadv under black-box settings. For example, our proposed method yields **10.51%** improvement on average against corruptions on RNet.

(3) Besides, we can witness that UnAdv with ensemble strategy shows lower defending ability compared to the single setting. More precisely, ensemble strategies decrease the performance of white-box and increase that of black-box on UnAdv. For example, UnAdv show 70.36% on VGG and 62.55% on RNet against corruptions in the ensemble setting, while the accuracy on VGG against corruptions is 98.16% under the single model setting. We attribute

Models	Methods	VGG				RNet				SNet				MNet			
		Raw	Cor	AdvL	AdvP	Raw	Cor	AdvL	AdvP	Raw	Cor	AdvL	AdvP	Raw	Cor	AdvL	AdvP
Clean		92.67	54.67	80.40	19.99	94.65	51.51	85.65	53.51	92.33	49.71	78.90	43.76	93.65	52.04	84.13	46.76
VGG	UnAdv	99.60	94.93	98.64	55.51	81.07	37.15	68.54	**27.47**	66.32	35.29	55.90	20.50	69.58	35.54	59.06	18.95
	Ours	**99.98**	**98.16**	**99.87**	**75.41**	**84.03**	**39.08**	**71.08**	27.12	**69.85**	**40.08**	**58.94**	**20.52**	**77.64**	**43.82**	**68.59**	**20.04**
RNet	UnAdv	64.21	37.05	57.50	15.31	99.23	78.83	96.94	74.41	77.35	42.28	65.64	25.61	67.92	32.69	57.60	17.36
	Ours	**72.44**	**45.66**	**66.00**	**18.39**	**99.93**	**90.90**	**99.50**	**92.22**	**83.37**	**52.33**	**72.93**	**29.92**	**78.45**	**44.00**	**68.52**	**22.62**
SNet	UnAdv	56.12	31.22	50.12	15.84	87.79	42.84	74.76	28.77	99.56	87.96	97.66	78.72	69.48	33.14	58.41	18.33
	Ours	**62.30**	**39.34**	**57.86**	**21.28**	**89.57**	**47.95**	**78.14**	**35.16**	**99.96**	**93.06**	**99.62**	**89.70**	**76.41**	**41.43**	**66.76**	**23.58**
MNet	UnAdv	68.57	39.80	62.02	17.02	84.45	38.95	71.19	25.94	71.76	38.46	59.37	**23.54**	99.93	90.26	99.50	80.25
	Ours	**71.16**	**45.84**	**65.01**	**19.48**	**86.13**	**43.32**	**73.64**	**25.95**	**74.51**	**43.75**	**63.48**	21.47	**99.99**	**93.94**	**99.87**	**93.04**

Table 1. The experimental results under single model setting. Note that we do not compare with Trans [50] in this situation. It can be observed that "Ours" shows better generalization and transferability. Higher accuracy values are in bold, *i.e.*, better performance.

this observation to the deficiencies of the average ensemble strategy, *i.e.*, ignoring the exploitation of shared high-level characteristics such as correlation among global features.

To sum up, our defensive patch generation framework achieves high generalization and transferability in practical performance, showing significant accuracy improvements under diverse noises among multiple models, *i.e.*, **20.18%** improvement on average for adversarial robustness and **31.10%** improvement on average for corruption robustness on the mentioned 4 models.

4.3. Physical World Evaluation

To evaluate the effectiveness in the physical world, we select the traffic sign classification task under the consideration of the popularity of autonomous driving and its huge potential for applications. Therefore, we generate our defensive patches based on a widely-used traffic sign classification dataset, *i.e.*, GTSRB, and then print them using an HP Color LaserJet Professional CP5225 printer.

We choose three different real-world traffic signs from the campus environment as shown in supplementary[2], *i.e.*, *speed-limited 20* (denote "SL"), *no entry* (denote "NE"), and *go straight or left* (denote "GSL"). We choose 4 different situations (*e.g.*, raw, snow, brightness, adversarial patch) to simulate the different noises in the real world. Further, for adversarial attacks, we employ the AdvPatch and stick them on the traffic sign. For each kind of traffic sign under each situation, we sample images from 3 distances (*i.e.*, 0.5m, 0.75m, and 1m) and 3 orientations (*i.e.*, front side, left side, and right side). Regarding the defenses, we use our defensive patches and UnAdv. Therefore, we obtain $12 * 9 * 3 = 324$ images as the physical world test set in total, including diverse noises (corruptions and adversarial examples). Furthermore, we evaluate these real-world sampled images by RNet models to validate the practical effectiveness of our proposed method.

Noises	Methods	VGG	RNet	SNet	MNet
Raw	Vanilla	92.67	94.65	93.65	92.33
	UnAdv	88.57	95.51	82.00	73.14
	Trans	88.84	93.69	90.24	91.31
	Ours	**99.27**	**98.82**	**99.02**	**99.68**
Cor	Vanilla	54.67	51.51	52.04	49.71
	UnAdv	70.36	62.55	54.48	36.38
	Trans	51.71	51.53	49.15	50.41
	Ours	**91.02**	**76.04**	**83.26**	**87.37**
AdvL	Vanilla	80.40	85.65	78.90	84.13
	UnAdv	83.00	88.87	71.79	62.36
	Trans	72.71	81.61	73.53	78.65
	Ours	**97.27**	**96.04**	**96.07**	**98.67**
AdvP	Vanilla	19.99	53.51	43.76	46.76
	UnAdv	29.24	49.39	31.71	19.80
	Trans	33.18	59.52	52.91	53.17
	Ours	**40.83**	**70.81**	**67.67**	**64.82**

Table 2. The experimental results under four models ensemble setting on CIFAR-10 dataset. Unadv [37] and Trans [50] show weak defense ability.

According to Table 3, we can observe that under different situations in the real world, our proposed defensive patches achieve better performance and outperform others (*i.e.*, Vanilla and UnAdv) by large margins, *i.e.*, higher accuracy values. For all noises, our method achieves **26.86%** improvement on average[2].

Besides the above task, it should be noted that the proposed defensive patch generation framework owns the potential to perform defenses in other approaches, *e.g.*, product special clothes, camouflages, coating, *etc*. We generate some simple examples to demonstrate this viewpoint[2].

4.4. Discussion and Analysis

In this section, we first provide some discussions from the perspectives of model attention (*i.e.*, qualitative analysis) and decision boundary (*i.e.*, quantitative analysis) to better understand our defensive patches; then we show that our defensive patches can be employed with other model-end strategies to further improve robustness.

4.4.1 Model Attention Analysis

We first adopt Grad-CAM [39] to visualize the attention of models when making predictions towards the same images

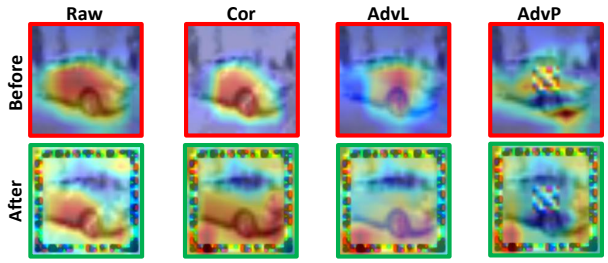


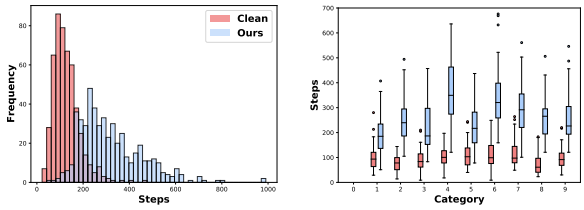
Figure 3. The model attention analysis. The red frames denote the examples without defensive patches (*i.e.*, "Before") and the green ones denote the examples with defensive patches (*i.e.*, "After"). The model global perception is better activated. Best in view.

with or without defensive patches.

Specifically, we select some samples from each category of CIFAR-10 and acquire their corresponding perturbed examples using noises (*i.e.*, Cor, AdvL, AdvP). These instances satisfy the conditions that vanilla models fail to classify correctly on these perturbed images whereas they could provide correct predictions with the help of our defensive patches. Then we calculate the attention map of each group of the sampled images to exhibit the model perception variation by the Grad-CAM [39]. As shown in Figure (3), after sticking the defensive patches, the model perception has been spread into a larger region globally over the image, which indicates that the model exploits more global features during the decision-making process[2]. Thus, by better activating the global perception, our defensive patches could improve model robustness and transfer among models.

Guide board	Class	Accuracy			
		Raw	Snowy	Brightness	AdvP
SL	Vanilla	44.44	44.44	33.33	11.11
	UnAdv	33.33	33.33	11.11	22.22
	Ours	**55.56**	**55.56**	**44.44**	**44.44**
NE	Vanilla	88.89	77.78	88.89	44.44
	UnAdv	77.78	88.89	88.89	33.33
	Ours	**100.00**	**100.00**	**100.00**	**66.67**
GSL	Vanilla	44.44	33.33	22.22	33.33
	UnAdv	44.44	11.11	33.33	33.33
	Ours	**55.56**	**66.67**	**66.67**	**77.78**

Table 3. Physical world experimental results. All images are tested on ResNet models. "Ours" accuracy value is much higher.



(a) (b)

Figure 4. Decision boundary analysis. (a) The frequency statistics for the number of average steps. (b) The distance statistics of a certain class, *i.e.*, distances of class 0 to another 9 classes.

4.4.2 Decision Boundary Study

To further understand our defensive patches, we follow [27] and provide a decision boundary analysis, which we aim to characterize the difficulty of fooling a classifier with or without our defensive patches.

Specifically, we perturb an instance x^i to specified classes and estimate the smallest optimization step numbers moved as the decision boundary distance. Given a learnt model $\mathbb{F}$ and a direction (*i.e.*, class y^j, $i \neq j$), we optimize the instance until satisfying $\mathbb{F}(x^i) \neq y^i$ by following [27]. In detail, we randomly sample 50 examples for each category (500 in total) and employ RNet as $\mathbb{F}$. By calculating the statistics (*e.g.*, average and median, *etc*) of distances, we can draw some meaningful conclusions from Figure (4). Firstly, it can be observed that the distribution of average step number shifts to bigger values after using our defensive patches (Figure 4a), which indicates that it is more difficult for adversaries to attack the models (*i.e.*, the distance for different instances to decision boundaries are larger). Moreover, for each specific class, the decision boundary distances are larger after adding defensive patches as shown in Figure 4b (*e.g.*, the blue boxes are higher than the red boxes).

Therefore, we demonstrate that our defensive patches could help models to better resist the influence of noises, *i.e.*, more difficult to be perturbed to other categories.

4.4.3 Combination with Other Model-end Defenses

While the data-end defenses are independent to model-end defenses, it is rational for us to explore the possibility of jointly exploiting them both to further improve model robustness against noises.

In particular, we select the typical and popular model-end defense strategy, *i.e.*, PAT [32], which adversarially train models with PGD adversarial attacks [32]. We use VGG, RNet, SNet, and MNet as backbone models and adversarially train them respectively following the PAT strategies. For evaluation, we adopt the same testing dataset as Section 4. As illustrated in Figure 5, we can clearly observe the positive effects of the defensive patches, *i.e.*, by adding our defensive patches with PAT, **+29.32%** on corruptions, **+23.03%** on adversarial noises. These results enable us to draw a meaningful conclusion that our defensive patches could serve as a strong method for real-world applications due to their flexible usage and significant promotion for robust recognition. Another model-end defense strategy, *i.e.*, PatG [48], is also compared[2].

4.5. Ablations Studies

In this section, we provide ablation studies to better understand the effectiveness of different parts of our defensive patch generation framework.

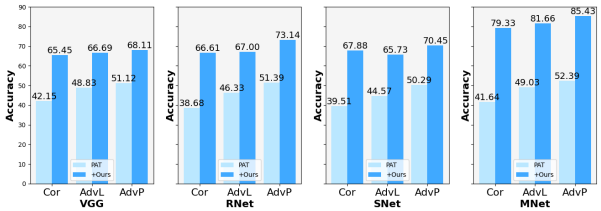


Figure 5. The experimental results of working with model-end defenses. "+Ours" indicates the joint performance and achieves better performance, *i.e.*, higher accuracy.

Method	Noises		
	Cor	AdvL	AdvP
$\mathcal{L}_t$	61.73	88.90	46.45
$+\mathcal{L}_p$	**71.21**	**94.35**	**59.02**
Ours	**76.04**	**96.04**	**68.81**

Table 4. Ablations on $\mathcal{L}_p$ for "Cor", "AdvL", "AvP" under ensemble settings.

Method	Models		
	VGG	SNet	MNet
$\mathcal{L}_t$	27.99	32.37	27.31
$+\mathcal{L}_c$	**34.51**	**43.03**	**30.19**
Ours	**46.85**	**50.35**	**45.86**

Table 5. Ablations on $\mathcal{L}_c$ under single model settings for "Cor".

Method	Models		
	VGG	SNet	MNet
$\mathcal{L}_t$	49.56	57.21	51.98
$+\mathcal{L}_c$	**53.63**	**65.41**	**54.13**
Ours	**66.80**	**71.84**	**71.72**

Table 6. Ablation on $\mathcal{L}_c$ under single model settings for "AdvL".

Method	Models		
	VGG	SNet	MNet
$\mathcal{L}_t$	13.60	**25.62**	16.48
$+\mathcal{L}_c$	**15.52**	25.52	**16.60**
Ours	**18.39**	**29.92**	**22.62**

Table 7. Ablation on $\mathcal{L}_c$ under single model settings for "AdvP".

4.5.1 Impact of Different Loss Terms

Here, we first investigate the impacts of the different loss terms, *i.e.*, $\mathcal{L}_p$ and $\mathcal{L}_c$.

Specifically, we first study the impact of the $\mathcal{L}_p$ to the generalization ability against diverse noises. To make it a fair comparison, we train two models with $\mathcal{L}_t$ and $\mathcal{L}_t$+$\mathcal{L}_p$ with the ensemble setting. According to Table 4, we can observe that the robustness under "$\mathcal{L}_t$+$\mathcal{L}_p$" setting is much higher than that under $\mathcal{L}_t$ setting. The results empirically prove that $\mathcal{L}_p$ could improve model robustness against diverse noises.

Then we study the impact of the $\mathcal{L}_c$ to the transferability across different models under the single model setting. In detail, we select the RNet as the source model and the other 3 models as target models (VGG, SNet, and MNet). As shown in Table 5, 6, and 7, the "$\mathcal{L}_t$+$\mathcal{L}_c$" always achieves higher accuracy on almost all target models under different noise settings, which strongly support that $\mathcal{L}_c$ could improve the transferability between models[2].

4.5.2 The Number of Ensemble Models

Since our defensive patch generation framework introduces the ensemble strategy, it is necessary to investigate the effects on the number of ensemble models.

Specifically, we adopt different ensemble settings (*i.e.*, optimize the defensive patch based on 1, 2, 3, and 4 models), and keep other settings the same for fair comparisons. We can summarize the following observations: (1) model robustness improves with the increasing of ensemble model numbers; (2) beyond the "white-box" models (*i.e.*, the ensemble models), our generated patches perform better on unseen models. Thus, we could conclude that the transferability between models is benefited from the ensemble perceptual correlation reinforcement[2].

4.5.3 The Shape of the Defensive Patches

Finally, we investigate the performance of defensive patches with different shapes (*i.e.*, different shape masks $\mathbf{M}$ in Equation (2). Note that, this experiment is designed to test the defense ability of our patches in more practical scenarios.

Specifically, we handcraft 3 different shape masks, including circle, triangle, and trapezoid as shown in *Section 3 in Supplementary Material*. Note that the sizes (pixel numbers) of these patches are set to be similar levels[2], which has no impact on the target object. We generate different defensive patches with different shape masks based on VGG, ResNet, ShuffleNet, and MobileNet, and then place them at the same positions and evaluate their performance (*i.e.*, Raw, Cor, AdvL, AdvP). According to *Table 7 in Supplementary Material*, we can conclude that the shape only has very limited impacts on the performance of the defensive patches, *i.e.*, 99.78%, 98.87%, 99.93% for circle, triangle, and trapezoid, respectively (Raw accuracy on VGG), which can be ignored in real applications[2]. Therefore, the proposed defensive patch generation framework can be more flexible in real-world applications for various scenarios.

5. Conclusion

This paper proposes a novel defensive patch generation framework to conduct data-end defense by better exploiting both local and global features. Our defensive patches could achieve strong generalization against diverse noises and transferability among different models. Extensive experiments demonstrate that our defensive patch outperforms others by large margins (*e.g.*, improve **20+%** accuracy for both adversarial and corruption robustness on average in the digital and physical world).

Our defensive patches could be easily deployed in practice to defend noises by simply sticking them around the target objects (*e.g.*, traffic signs in cities that often snow). In the future, we are interested in trying more convenient approaches such as employing these patches as a preprocessing procedure (automatic detecting and sticking the defensive patch onto the image and then feeding the image to the system). Moreover, applying this strategy in more visual tasks, *e.g.*, detection tasks, is also worth attempting.

[2]Please refer to the Supplementary Material for more details

References

[1] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2, 5

[2] Kim M Curby and Denise Moerel. Behind the face of holistic perception: Holistic processing of gestalt stimuli and faces recruit overlapping perceptual mechanisms. *Attention, Perception, & Psychophysics*, 81(8):2873–2880, 2019. 2, 3

[3] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018. 2

[4] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. K. Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[5] Ranjie Duan, Xiaofeng Mao, A Kai Qin, Yuefeng Chen, Shaokai Ye, Yuan He, and Yun Yang. Adversarial laser beam: Effective physical-world attack to dnns in a blink. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16062–16071, 2021. 2, 5

[6] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[7] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(10):1915–1926, 2011. 3

[8] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017. 2

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[10] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 2

[11] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 2

[12] Jinping Hao, Robert J Piechocki, Dritan Kaleshi, Woon Hau Chin, and Zhong Fan. Sparse malicious false data injection attacks and defense mechanisms in smart grids. *IEEE Transactions on Industrial Informatics*, 11(5):1–12, 2015. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016. 5

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 1, 2

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1, 2, 5

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 2

[17] Fernanda Hernández-Luquin and Hugo Jair Escalante. Multi-branch deep radial basis function networks for facial emotion recognition. *Neural Computing and Applications*, pages 1–15, 2021. 2, 3

[18] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 5

[19] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[20] Wei Jiang, Zhiyuan He, Jinyu Zhan, and Weijia Pan. Attack-aware detection and defense to resist adversarial examples. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020. 2

[21] Been Kim, Emily Reif, Martin Wattenberg, and Samy Bengio. Do neural networks show gestalt phenomena? an exploration of the law of closure. *arXiv preprint arXiv:1903.01069*, 2(8), 2019. 2, 3

[22] Alex Kopestinsky. 25 astonishing self-driving car statistics for 2021. https://policyadvice.net/insurance/insights/self-driving-car-statistics/, 2021. 1

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[24] Sanghak Lee, Cheng Yaw Low, Jaihie Kim, and Andrew Beng Jin Teoh. Robust sclera recognition based on a local spherical structure. *Expert Systems with Applications*, page 116081, 2021. 2, 3

[25] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1028–1035, 2019. 2

[26] Aishan Liu, Xianglong Liu, Hang Yu, Chongzhi Zhang, Qiang Liu, and Dacheng Tao. Training robust deep neural networks via adversarial noise propagation. *IEEE Transactions on Image Processing*, 2021. 1, 2

[27] Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, and Hang Yu. Bias-based universal adversarial patch attack for automatic check-out. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 395–410. Springer, 2020. 2, 3, 7

[28] 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

[29] Jia Lu and Wei Qi Yan. Comparative evaluations of human behavior recognition using deep learning. In *Handbook of Research on Multimedia Cyber Security*, pages 176–189. IGI Global, 2020. 2, 3

[30] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 5

[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. 1, 2

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. 2017. 7

[33] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016. 2

[34] Jamshid Pirgazi, Ali Ghanbari Sorkhi, and Mohammad Mehdi Pourhashem Kallehbasti. An efficient robust method for accurate and real-time vehicle plate recognition. *Journal of Real-Time Image Processing*, pages 1–14, 2021. 2, 3

[35] Yuankai Qi, Shengping Zhang, Feng Jiang, Huiyu Zhou, Dacheng Tao, and Xuelong Li. Siamese local and global networks for robust face tracking. *IEEE Transactions on Image Processing*, 29:9152–9164, 2020. 2, 3

[36] Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, et al. A unified game-theoretic interpretation of adversarial robustness. *arXiv preprint arXiv:2111.03536*, 2021. 4

[37] Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *arXiv preprint arXiv:2012.12235*, 2020. 1, 5, 6

[38] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 1, 2

[39] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, Oct 2017. 6, 7

[40] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019. 2

[41] Zisserman A Simonyan K. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, September 2014. 5

[42] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 2

[43] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011. 5

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[45] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 2

[46] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8565–8574, June 2021. 2, 3

[47] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 2

[48] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. 2020. 7

[49] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 2

[50] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 2, 5, 6

[51] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991, 2017. 1

[52] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *arXiv preprint arXiv:1906.03787*, 2019. 2

[53] Wei Xiong, Lefei Zhang, Bo Du, and Dacheng Tao. Combining local and global: Rich and robust feature pooling for visual recognition. *Pattern Recognition*, 62:225–235, 2017. 2, 3

[54] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 49–64, 2018. 1

[55] Xiaopei Zhu, Xiao Li, Jianmin Li, Zheyao Wang, and Xiaolin Hu. Fooling thermal infrared pedestrian detectors in real world using small bulbs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3616–3624, 2021. 2