Adversarial Pixel Masking: A Defense against Physical Attacks for Pre-trained Object Detectors

Ping-Han Chiang
bhchiang@datalab.cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

Chi-Shen Chan
csch@datalab.cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

Shan-Hung Wu
shwu@cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

ABSTRACT

Object detection based on pre-trained deep neural networks (DNNs) has achieved impressive performance and enabled many applications. However, DNN-based object detectors are shown to be vulnerable to physical adversarial attacks. Despite that recent efforts have been made to defend against these attacks, they either use strong assumptions or become less effective with pre-trained object detectors. In this paper, we propose adversarial pixel masking (APM), a defense against physical attacks, which is designed specifically for pre-trained object detectors. APM does not require any assumptions beyond the "patch-like" nature of a physical attack and can work with different pre-trained object detectors of different architectures and weights, making it a practical solution in many applications. We conduct extensive experiments, and the empirical results show that APM can significantly improve model robustness without significantly degrading clean performance.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; • **Security and privacy** → **Software and application security**.

KEYWORDS

object detection; attack; defense; adversarial examples; adversarial patches; adversarial training; distribution shift; MaskNet

ACM Reference Format:
Ping-Han Chiang, Chi-Shen Chan, and Shan-Hung Wu. 2021. Adversarial Pixel Masking: A Defense against Physical Attacks for Pre-trained Object Detectors. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475338

1 INTRODUCTION

Object detection based on deep neural networks (DNNs) [3, 12, 23, 25, 32–34, 42] has achieved impressive performance in recent years. Due to the large amount of time and computing resources required to train an object detector, a lot of pre-trained object detectors have been released on the Internet, which simplifies application development and casts a huge impact in the industry. However, DNNs have shown to be vulnerable to *adversarial attacks* [14, 41], which aim to mislead model prediction by perturbing the input (an image) while keeping the input nearly indistinguishable from regular examples in human eyes or some distance measures in the input space. Depending on the types of perturbations, existing adversarial attacks for images can be roughly divided into two categories: *digital attacks* [6, 14, 26, 30] and *physical attacks* [2, 5, 8, 13, 19, 21, 22, 39, 43, 45, 50, 55, 57]. A digital attack slightly perturbs all pixels of a clean image to make it adversarial. On the other hand, a physical attack perturbs the pixels only within one or few small regions in an image, but the pixels can be completely changed, as shown in Figure 1.

(a) (b) (c)

Figure 1: Example physical attack and corresponding mask. (a) An input image with adversarial patches generated by a physical attack that makes a pre-trained YOLOv3 output nothing. (b) The mask generated by APM. (c) Detection result of the pre-trained YOLOv3 given the masked image.

The perturbations made by a physical attack are "patch-like" and thus can be printed and attached to real-world objects, such as traffic signs or human clothes [2]. This poses threats to object detectors [8, 21, 39, 43, 45, 50, 55, 57] and their security-sensitive applications such as self-driving cars [13] and face recognition systems [38]. Studies have shown that, by using physical attacks, an attacker can evade object detectors [43, 45, 50], pretend to be someone else in front of a face recognition system [38], or even change a traffic sign seen by a self-driving car [8, 13, 39, 57]. Since a pre-trained object detector may not have considered these threats, it is crucial to develop a technique that can improve the adversarial robustness of pre-trained object detectors.

However, existing defenses against the physical attacks either rely on strong assumptions that may not hold in practice [10, 15, 27, 28, 36, 46, 51, 56, 59] or do not work well with pre-trained models [31, 44, 54]. As we will show in Section 4, an attacker can easily work around the assumptions to break the former defenses. The latter studies [31, 44, 54] adapt *adversarial training*, a technique originally proposed to defend against digital attacks [26], to physical attacks.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475338

Nevertheless, adversarial training does not work well with a *well-trained* model due to a trade-off between adversarial robustness and clean performance [20, 47].[1] Our experimental results also show that it does not generalize to defending against stronger adversarial patches at test time. There is a crucial need for a new defense designed specifically for pre-trained object detectors to protect downstream applications.

In this paper, we take practicability into account and propose a simple, yet effective defense, called *adversarial pixel masking* (APM), against physical attacks for pre-trained object detectors. The APM alters adversarial training by prepending a data-preprocessing network, called MaskNet, to a given object detector. During adversarial training, we fix the weights of the object detector and let the MaskNet learn to mask adversarial patches, if existing, in input images. Our method does not require any assumption beyond the "patch-like" nature of a physical attack (i.e., a small region whose pixels can be arbitrarily changed by an attacker). Furthermore, it is agnostic to the architecture and weights of the object detector and thus can be applied to a wide range of downstream applications. We conduct extensive experiments to verify the effectiveness of APM, and the results show that APM can significantly improve the adversarial robustness of a pre-trained object detector without degrading clean performance in different learning tasks. Furthermore, APM takes little inference time and can support video frame rate above 30 fps on the INRIA dataset. Following summarizes our contributions:

- We propose a new defense called adversarial pixel masking (APM) that helps a pre-trained object detector defend against physical attacks.
- The APM does not use the expansive assumptions made by some existing heuristic-based defenses and thus is much more widely applicable to different downstream applications.
- Unlike adversarial training, the APM avoids the trade-off between adversarial robustness and clean performance. This make it applicable to mission-critical applications requiring high clean performance.
- We conduct extensive experiments to verify the effectiveness of APM. The experimental results show that APM can significantly improve the robustness of a pre-trained YOLOv3 in either normal or transfer learning tasks.

Our study has implications for other downstream media processing tasks. In particular, the masks produced by the MaskNet, as shown in Figure 1, can help localize/segment adversarial patches or analyze the factors in the input space that affect model predictions. They can also be used to explain the robustness or clean performance of an object detector.

2 RELATED WORK

An object detector, denoted by $f(\cdot\,;\boldsymbol{\theta})$ and parametrized by $\boldsymbol{\theta}$, normally detects objects only when the patterns in an input image are of high *objectiveness* and *classification* scores [3, 23, 25, 32–34]. In this paper, we consider physical adversarial attacks that manipulate the objectiveness scores [21, 43, 45, 55, 57], classification scores [8, 39], or both [43]. Next, we review existing defenses against physical attacks. Based on the underlying techniques, we roughly divide them into three categories.

Pixel-level Detection and Removal. Xu et al. [51] and Hayes et al. [15] employ the saliency map to detect the adversarial patches in an input image and then restore the pixels within the detected patches using an image inpainting technique. These approaches were proposed to enhance the robustness of *classification* models, as the saliency map of a classification model can be easily derived from the gradients of an output logit with regard to the input image or by the help of advanced interpretability methods such as guided backpropagation [40] or CAM [58]. However, it is non-trivial to apply such techniques to object detection tasks because an attacker can choose to lower the objectiveness scores to evade detection.

Guangzhi et al. [59] and Naseer et al. [28] observed that adversarial patches often consist of high-frequency noises and are relatively non-smooth compared to the rest of an image. They propose to make use of image analysis tools, such as the discrete entropy or image gradient, to locate and remove abnormally high-frequency areas in input images. However, smooth, low-frequency adversarial patches are still possible (see Section 4), which limit the applicability of these approaches.

Limited Receptive Field. The receptive field of a neuron at a deep layer of a convolutional neural network (CNN) is usually designed to be large in order to help the network make predictions by leveraging more information in the image. However, Saha et al. [36], Zhang et al. [56], and Chong et al. [46] believe that a large receptive field is a reason why deep CNNs are susceptible to physical adversarial attacks. Assuming that the adversarial patches do not overlap with the objects being detected, Zhang et al. [56] and Chong et al. [46] propose to employ a deep CNN with small receptive fields, such as BagNet [4], to prevent the network from using adversarial features. Saha et al. [36] also design a regularization term that encourages an object detector to only make use of the features inside a proposed bounding box. Nevertheless, the assumption does not hold in many real-world situations, such when an adversarial patch is attached to a traffic sign being detected by a self-driving car [8, 39, 55, 57], or when a patch is attached to the clothes of people being detected by a surveillance system [43, 45, 52].

Adversarial Training. Adversarial training [16–18, 26, 37], which enhances model robustness by adding adversarial examples during training, is one of the most popular defenses against digital attacks due to its effectiveness and openness to different attacks. It has the following objective:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\left[\max_{\boldsymbol{p}\in\mathcal{T}} L(f(\boldsymbol{x}';\boldsymbol{\theta}),\boldsymbol{y})\right], \tag{1}$$

where $\mathcal{D}$ is the underlying data distribution, $(\boldsymbol{x},\boldsymbol{y})$ is an example, $L(\cdot,\cdot)$ is a loss function, and $\boldsymbol{x}' = \boldsymbol{x}+\boldsymbol{p}$ is an adversarially perturbed image with the perturbation $\boldsymbol{p}$ whose allowable values are governed by a threat model $\mathcal{T}$. Recently, Wu et al. [44], Sukrut et al. [31], and Zhang et al. [54] adapt adversarial training to either physical attacks or object detection tasks. However, adversarial training has shown to be less effective when the model is well-trained [20] and could result in a trade-off between adversarial robustness and clean performance [47].

Today, many real-world applications use well-trained object detectors for high clean performance. The above limitations in

[1] To work around this, existing defenses usually start adversarial training from half-trained model weights [31, 44, 54].

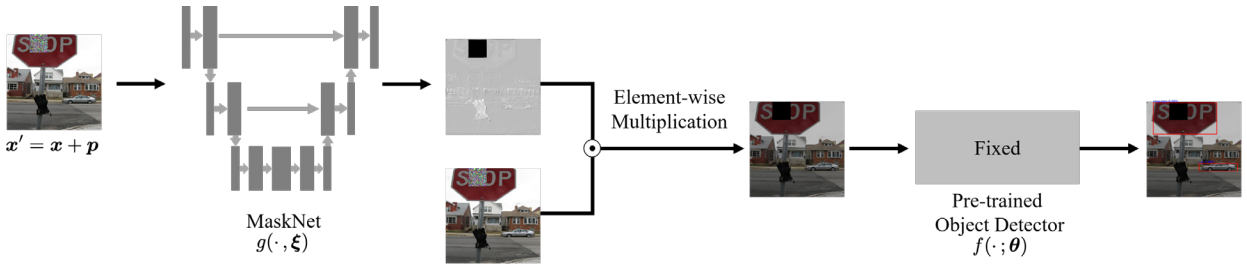


Figure 2: The feedforward flow of APM (left: an input image $x' = x + p$ containing adversarial patches p; middle: masked image $x' \odot g(x'; \xi)$; right: detection results $f(x' \odot g(x'; \xi); \theta)$). During adversarial training, the weights θ of the pre-trained object detector are fixed, and both θ and ξ are accessible to the threat model $\mathcal{T}$ that generates p.

existing defenses raise the concern about the integrity of these applications, which may seriously impact our lives. As such, it is crucial to develop a new defense for pre-trained object detectors.

3 ADVERSARIAL PIXEL MASKING

The reason why adversarial training becomes less effective with a well-trained model is because of the data distribution shift from clean examples to adversarial examples. Although looking similar to human eyes, adversarial and clean examples are very different in the feature space of a DNN [48]. This leads to a trade-off between robustness and clean performance. To mitigate the trade-off, existing adversarial training schemes usually add adversarial examples in the beginning [26] or middle [20] of training. Recently, Xie et al. [47] propose to use additional, dedicated batch norm layers for adversarial examples, trying to prevent the clean and adversarial examples from interfering with each other. However, the above approaches are not applicable to pre-trained models whose weights have been highly optimized and the architectures are fixed.

Here, we take another approach that leaves the pre-trained model as it is. We instead propose adversarial pixel masking (APM) that removes adversarial patches in images so that the distribution shift can be mitigated *in pixel space* (and therefore a feature space too).

The APM prepends a MaskNet parametrized by ξ, denoted by $g(\cdot, \xi)$, to a pre-trained object detector, as shown in Figure 2. The MaskNet takes an image $x' = x + p \in \mathbb{R}^{W \times H \times C}$ as the input and outputs a mask $g(x'; \xi) \in [0, 1]^{W \times H \times 1}$, where W, H, and C denote the width, height, and number of channels and p denotes adversarial patches that completely replace overlapping pixels in a begin image x. The MaskNet uses the Logistic function as the non-linearity in its output layer. Once having the mask, APM feeds the masked image $x' \odot g(x'; \xi) \in \mathbb{R}^{W \times H \times C}$ into the object detector, where $\odot$ denotes per-channel element-wise multiplications. Note that the design of hidden layers of MaskNet depends on applications. Empirically, we find that the U-Net [35] architecture works generally well due to its superior capabilities of handling pixel-level tasks.

To let the MaskNet learn to detect and remove adversarial patches, we alter adversarial training and solve the following objective during training:

$$\arg\min_{\xi} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{p \in \mathcal{T}} L(f(x' \odot g(x'; \xi); \theta), y) \right]. \quad (2)$$

Algorithm 1 Adversarial Pixel Masking

procedure GenAdvExample(x, λ, N, θ, $\mathcal{T}$)
  Initialize adversarial patches p based on a threat model $\mathcal{T}$;
  repeat
    $x' = x + p$; ▷ Apply patch p to x
    Update p by making a projected gradient descend step regarding the inner max problem in Eq. (2);
  until N times
  return $x + p$;
end procedure

procedure AdvTrainStep(x, y, λ, N, θ, $\mathcal{T}$, ξ)
  x' = GenAdvExpample(x, λ, N, θ, $\mathcal{T}$);
  Update MaskNet weights ξ by making a gradient descend step regarding Eq. (2);
end procedure

Since APM is designed specifically for well-trained object detectors, the weights θ of the object detector are *fixed* at training time in order to prevent a drastic change in clean performance. Algorithm 1 outlines the key steps of APM. Note that the APM can work with different threat models $\mathcal{T}$ and pre-trained object detectors $f(, ; \theta)$.

Defending Physical Attacks. Given the goal of minimizing L on either clean or adversarial examples, the MaskNet will learn to mask the adeversarial pixels (to minimize L on adversarial examples) while letting the clean pixels pass (to minimize L on clean examples). At first glance, APM is similar to gradient masking [49, 53],[2] a type of defenses that is proven useless if an attacker knows the existence of such prepending network [1, 7]. Despite the cosmetic similarity, APM works in a fundamentally different way than gradient masking. During adversarial training, the fixed weights θ expose the vulnerability of the object detector and the MaskNet is required to "fix" the vulnerability by solving Eq. (2). Since the threat model $\mathcal{T}$ can access *both* θ and ξ when generating p, the MaskNet has to defend against white-box attacks. This encourages the MaskNet to learn to *completely* remove the adversarial patches from x' so that a white-box attacker cannot bypass APM even when knowing ξ. Our experimental results confirm such learned behavior (as shown in Figure 4).

[2] In these works, a network/algorithm is prepended to a target model in order to hide the gradients of the target model from attackers.

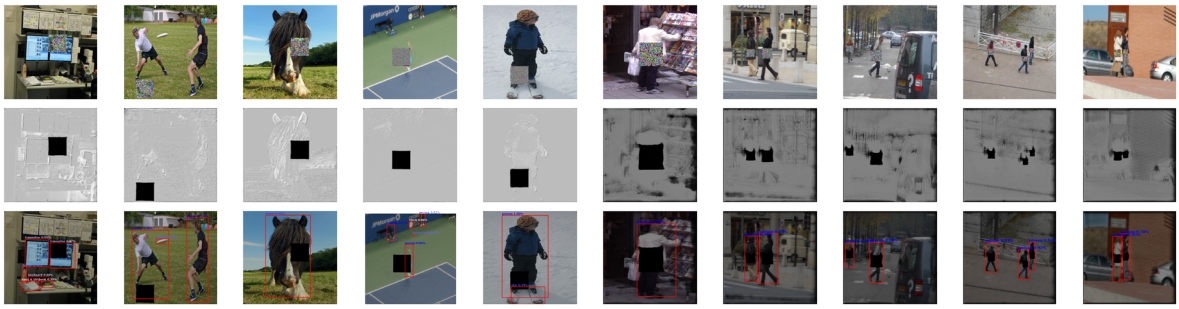


Figure 3: Randomly sampled adversarial examples (top), masks (middle), and detection results (bottom) of APM.

Threat Model. The APM can be used to defend against most existing physical attacks provided that the attacks have been considered by the threat model $\mathcal{T}$ in Algorithm 1. The threat model can control many factors, such as the number of adversarial patches to be added to an input image (line 2), the size and location of each patch (line 2), and its content pixels (lines 3-6). To the best of our knowledge, the GenAdvExample($\cdot$) routine can model the choices made by most existing physical attacks, including ignorance attacks [21, 43, 45, 55, 57] which aim to make some important objects disappear, false-positive attacks [8, 43] which aim to create non-existing objects, and classification attacks [8, 39] which aim to mislead object labels. For more details of supporting different attacks, please see Section 1 of the supplementary file [9].

4 EXPERIMENTS

In this section, we evaluate the adversarial robustness of APM and inspect the quality of its masks.

Pre-trained Models and Datasets. We use YOLOv3 [33] and RetinaNet [23] as the pre-trained object detectors. Both of these networks were pre-trained on the COCO dataset [24]. The COCO dataset comprises about 80K training images and 40K validation images where objects of totally 80 classes are identified and annotated in the images.[3] We also consider the INRIA-person [11] dataset. INRIA is a relatively small dataset which contains 614 training images and 288 testing images, where only objects of type "person" are identified and annotated. We use the INRIA dataset to evaluate the performance of APM in a transfer learning task, a common use case of a pre-trained model.

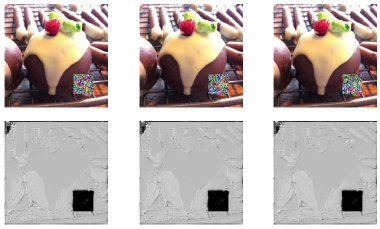
Baselines. We consider and implemented the following defenses against physical attacks. **ROC.** Assuming that an adversarial patch does not overlap with the objects being detected, Saha et al. [36] proposed to limit the receptive field of object detector by adding a regularization term during training. The regularization term encourages an object detector to only make use of the features inside a bounding box when predicting the corresponding objectiveness/classification scores. So, the object detector won't be mislead by the adversarial patches. We abbreviate this technique as ROC (Role of Spatial Concept) hereafter. **LGS.** Assuming that an adversarial patch contains high-frequency signals, Naseer et al. [28] divided an image into several grids, calculated the summation of image gradients inside each grid, and decided whether an adversarial patch exists in the image by checking if the summation passes a pre-defined threshold. We use a grid size of 40×40 pixels, and set the same threshold as in the original paper . We abbreviate this method as LGS (Local Gradient Smoothing) hereafter. Note that the LGS was originally proposed for classification models, but since it is model-agnostic, it can work with object detectors. **ADV.** Existing studies have adapted adversarial training, which was originally proposed for classification models against digital attacks, to either physical attacks [31, 44] or object detectors [54], but not both. Here, we consider the adversarial training used by APM except that 1) no MaskNet is used and 2) the weights of the pre-trained object detector are fine-tunable. We abbreviate this approach as ADV, and it shares the same threat model as APM.

Settings. During adversarial learning, we set N, the number of iterations for generating an adversarial example (see line 6 of Algorithm 1) to 10 and 30 on COCO and INRIA datasets, respectively. We configure the threat model $\mathcal{T}$ to generate the ignorance attacks [21, 43, 45, 55, 57], which are one of the most common and dangerous physical attacks because they aim to make an object detector ignore some important objects such as pedestrians or traffic signs. For the COCO dataset, the threat model places a single adversarial patch at a random position of an input image. The patch is of 80×80 pixels, which is 20% of the input. The top row of Figures 3(a)-(e) shows some example patches. For the INRIA dataset, the threat model creates multiple adversarial patches, one for each human object detected in an input image. For each adversarial patch, the threat model resizes it to fit into 80% of the shorter side of the corresponding human object and then places it on the middle of the longer side. Figures 3(f)-(j) show some example patches. We use the mean average precision (mAP) to evaluate both the clean performance and adversarial robustness of a model, supplied with clean and adversarial examples, respectively. Our implementation is built upon TensorFlow and we conduct the experiments on a cluster of machines with 80 NVIDIA Tesla V100 GPUs.

[3] Since the testing data of COCO is not publicly available, we use validation data for performance evaluation.

Table 1: Performance of different defenses with the pre-trained YOLO on COCO dataset.

	YOLO	ROC	LGS	ADV	APM
Clean	**52.3**	34.9	48.9	51.6	52.0
Noise	48.1	31.2	45.4	**49.4**	48.5
ATK(10)	34.7	20.2	44.4	45.4	**47.4**
ATK(20)	29.1	17.2	44.1	43.3	**46.8**
ATK(30)	26.1	15.7	44.0	41.9	**46.8**



(a) (b) (c)

Figure 4: Top: Adversarial patches of different strengths at (a) $M = 10$, (b) $M = 20$, and (c) $M = 30$. Bottom: Corresponding masks produced by APM.

4.1 Object Detection on COCO

We first evaluate the clean performance and adversarial robustness of different defenses on the COCO dataset, which was used to pre-train the YOLO and RetinaNet object detectors. At test time, we generate adversarial patches using the same threat model $\mathcal{T}$ of adversarial training except that the numbers of iterations N (see lines 6 of Algorithm 1) is replaced by M. The higher the M, the stronger the attack. We consider $M = 10, 20, 30$ and denote by "ATK(M)" the adversarial examples generated after M iterations. As a sanity check, we also consider the patches filled with random noises (denoted by "Noise").

Table 1 shows the results of different defenses given YOLO as the pre-trained object detector. The APM gives the highest adversarial robustness at the least cost of clean performance. It maintain a high clean performance because the weights of pre-trained YOLO is fixed and the MaskNet outputs all-pass masks for clean examples empirically.

The robustness of APM, unlike other baselines, is *not* significantly affected by the attack strength at test time. In particular, it remains roughly the same as M goes from 20 to 30. Investigating the masks, we find that APM tends to completely remove adversarial patches regardless of the attack strength, as shown in Figure 4. Since the APM was only trained by weaker attacks ($N = 10$), this verifies that APM successfully models some useful priors.

Due to space limitation, we omit the results of the pre-trained RenitaNet. Please see Section 3 of the supplementary file [9] for more details.

Table 2: Performance of different defenses with pre-trained YOLO on INRIA dataset.

	YOLO	ROC	LGS	ADV	APM
Clean	87.4	85.1	86.7	**88.9**	87.8
Noise	84.9	82.3	86.0	**95.4**	92.2
ATK(10)	6.0	2.6	78.1	**90.1**	**90.1**
ATK(30)	0.5	0.0	75.4	77.8	**89.4**
ATK(50)	0.0	0.0	75.0	67.8	**88.6**
ATK(100)	0.0	0.0	73.3	48.6	**88.3**

4.2 Object Detection on INRIA via Transfer Learning

In practice, it is common to use a pre-trained object detector for transfer learning tasks, where the weights of the detector are fine-tuned by domain-specific data to achieve better (clean) performance. Here, we evaluate the performance of different defenses on the INRIA dataset, which focuses on human objects and is smaller than the COCO dataset originally used to trained the YOLO and RetinaNet. Before running a defense, we fine-tune the weights of YOLO and RetinaNet using INRIA to simulate the cases where a pre-trained object detector has been fine-tuned for transfer learning. Note that, since INRIA is a small dataset, there is a high chance of overfitting. Hence, in APM, we do *not* fix the weights $\boldsymbol{\theta}$ of the object detector during adversarial training so that the adversarial examples can serve as augmented data and mitigate overfitting.[4]

Table 2 shows the results. Both ADV and APM improve the clean performance thanks to the data augmentation effect during adversarial training. However, ADV gives much worse robustness due to the distribution shift from clean to adversarial examples discussed in Section 3. We also notice that ADV gives abnormally high performance on the images with noisy patches. We suspect ADV leverages the existence of a patch to identify human objects, as each object has a corresponding patch in an adversarial example. On the other hand, APM does not use this information because a successfully masked patch contains zeros and does not activate a neuron in the object detector.

4.3 Overlapping Patches

As shown in Tables 1 and 2, the ROC consistently performs worse than other defenses. It assumes that physical adversarial patches does not overlap with the objects being detected, and tries to improve robustness by limiting the receptive fields of neurons of the detector. However, the limited receptive fields create a negative impact on clean performance. Furthermore, when the assumption does not hold (as in both of our experiments on COCO and INRIA), the ROC can actually *hurt* robustness because it wrongly encourages the object detector to focus on the adversarial features inside the bounding boxes of objects.

[4]We use two-phase training where $\boldsymbol{\theta}$ is fixed at first and becomes tunable after $\boldsymbol{\xi}$ catches up. We also implement the "bag of tricks" by [29].

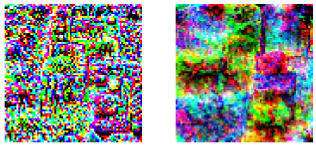


Figure 5: Example adversarial patches with (a) high-frequency and (b) low-frequency signals.

Table 3: Performance of LGS and APM under high- and low-frequency attacks on COCO dataset.

	YOLO		LGS		APM	
	High	Low	High	Low	High	Low
Clean	**52.3**		48.9		52.2	
Noise	48.1		45.4		**48.9**	
ATK(10)	34.7	40.2	44.4	42.3	**47.7**	**47.3**
ATK(20)	29.1	37.2	43.9	40.6	**46.8**	**46.9**
ATK(30)	26.1	35.6	44.2	39.5	**46.6**	**46.3**

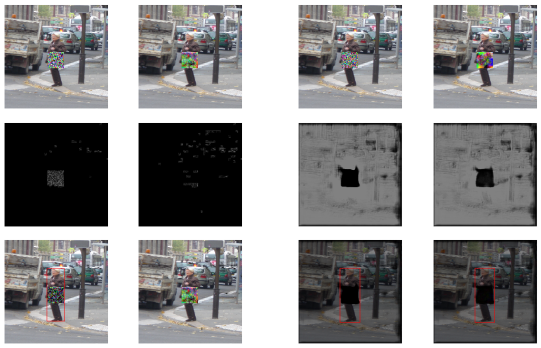
Table 4: Performance of LGS and APM under high- and low-frequency attacks on INRIA dataset.

	YOLO		LGS		APM	
	High	Low	High	Low	High	Low
Clean	**87.4**		86.7		82.9	
Noise	84.9		86.0		**91.5**	
ATK(10)	6.0	52.4	78.1	78.1	**90.3**	**89.8**
ATK(30)	0.5	7.3	75.4	69.4	**89.6**	**89.6**
ATK(50)	0.0	3.4	75.0	63.1	**89.5**	**89.3**
ATK(100)	0.0	1.7	73.3	58.5	**88.8**	**89.0**

4.4 Low-frequency Attacks

The LGS gives impressive performance in Tables 1 and 2. However, it assumes that an adversarial patch contains high-frequency signals, which may not hold in practice. To demonstrate this, we add a "low-frequency" attack into the threat model. We regularize $\boldsymbol{p}$ to have a small total variation loss, which encourages pixel smoothness. During adversarial training, we randomly sample λ from $\{0, 0.01\}$ in each training step, so the training data are augmented with both high-frequency and low-frequency patches. Figure 5 shows the visual difference between a high- and low-frequency patches.

As Tables 3 and 4 show, LGS gives worse robustness in the presence of low-frequency adversarial patches, yet the low-frequency patches seem to be easier to defend for the vanilla YOLO. Figure 6(a) shows how LGS fails to identify a low-frequency patch. On the other hand, APM is insensitive to the signal frequency because its



(a) LGS (high/low) (b) APM (high/low)

Figure 6: LGS and APM under high- and low-frequency attacks.

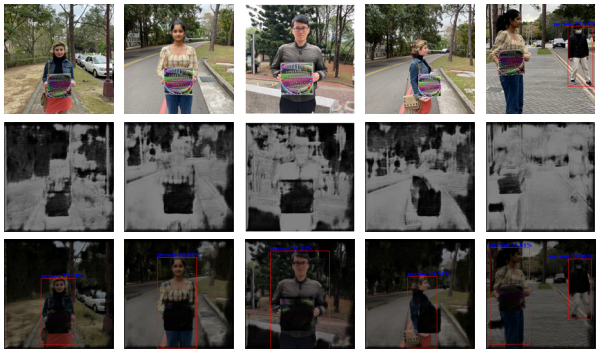


Figure 7: A physical attack in the real world, which aims to eliminate a pedestrian object when the pedestrian is attached to a printed universal adversarial patch. Top: Detection results of vanilla YOLO. Middle: Masks produced by APM. Bottom: Detection results of APM.

MaskNet can successfully identify and remove adversarial patches in either cases, as shown in Figure 6(b).[5]

4.5 APM in the Real World

Next we conduct experiments to see whether APM can defend against physical attacks on real-world streets. We create a printable adversarial patch [43, 50, 52] that 1) is universal so that it can be applied to any scene in the INRIA dataset and 2) takes into account post-print distortions such as white noise, rotation, brightness shift, etc. We print the universal patch out and verify that it can successfully attack the vanilla YOLO by muting a "person" object when the corresponding person carries the patch, as shown in the top row of Figure 7. We then use it to attack the APM trained on the INRIA dataset (following the settings in Section 4.2) and find that APM works well in most of the cases we have tested. The middle and

[5]The adversarial patches for LGS and APM looks different because they are computed based on the clean and masked images, respectively.

(a) (b) (c)

Figure 8: Feature maps of a YOLO filter given (a) clean image, (b) attacked image, and (c) masked image by APM.

Table 5: The average training and inference time of different defenses on (a) COCO and (b) INRIA datasets.

(a)	ROC	LGS	ADV	APM
Inference	119 ms	4,724 ms	3,021 ms	204 ms
Training	1d14 h	–	5 h 40 m	3 d
(b)	ROC	LGS	ADV	APM
Inference	96 ms	371 ms	29 ms	24 ms
Training	10 m	–	33 m	15 m

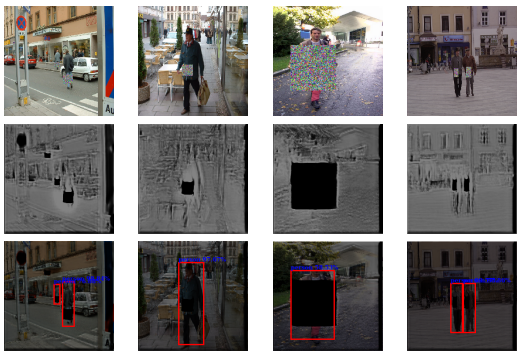
bottom rows of Figure 7 show some example masks and detection results of APM, respectively.

4.6 Feature Alignment

To understand how APM improves robustness, we visualize the pre-trained object detector in APM following the experiment in Section 4.1. Figure 8 shows the feature maps of a filter at the last layer of the feature extractor of YOLO given a clean image (Figure 8(a)), an attacked image (Figure 8(b)), and a masked image produced by MaskNet (Figure 8(c)). Although looking similar in the naked eye, the feature maps of the clean and attacked images are very different in the feature space. There exists a data distribution shift from clean to adversarial examples. On the other hand, the feature map of the masked image becomes similar to that of the clean image. The distribution shift is mitigated via pixel masking. We further verify this by comparing the average Euclidean distances between the feature tensors of raw adversarial and clean test images and that between the masked and clean test images. The former is 1.381, while the latter is only 0.286. By mitigating the distribution shift, APM helps the object detector function normally when under attack and meanwhile preserves clean performance.

4.7 Speed

Table 5 shows the time required to train and test different defenses on a commodity machine with a single NVIDIA Tesla V100 GPU. On the COCO dataset (Table 5(a)), APM takes longer time to train than ADV because 1) the MaskNet in APM is trained from scratch, and 2) ADV tends to stop early when paired up with a pretrained



(a) (b) (c) (d)

Figure 9: Threat models that generate adversarial patches (a) at different locations, (b) of smaller sizes, (c) of larger sizes, and (d) with different aspect ratios than the ones seen by APM trained on INRIA dataset.

Table 6: Performance of ADV and APM on INRIA dataset when the treat model $\mathcal{T}$ generates stronger adversarial patches during training ($N = 50$ or 80).

	$N = 50$		$N = 80$	
	ADV	APM	ADV	APM
Clean	82.3	**83.3**	**84.0**	83.0
ATK(50)	**97.4**	87.4	**97.6**	86.2
ATK(100)	73.7	**85.0**	**97.6**	86.1
ATK(200)	65.0	**84.4**	67.2	**84.2**
ATK(300)	56.8	**84.4**	68.3	**82.9**

object detector. At test time, ADV takes significantly longer time than APM to detect the objects in a test image. It turns out that 2,338 out of 3,021 ms was spend on the non-max suppression process (a deterministic algorithm run after the feedforward pass to decide the final bounding boxes) because there are significantly more overlapping candidate bounding boxes outputted by ADV. In the transfer learning task on the INRIA dataset where the weights of the object detector is not fixed (Table 5(b)), APM requires less time to train than ADV, showing that learning to mask patches is more efficient than learning to "fix" to object detector itself. APM takes only 24 ms to make inferences on the INRIA dataset, which supports video frame rate of 41 fps.

4.8 Ablation Study

Sensitivity to N. Sections 4.1 and 4.2 show that APM can better sustain stronger attacks at test time than ADV. We investigate whether this holds when the two defenses were trained with a larger N (i.e., stronger train-time attacks). Table 6 shows the results on the INRIA dataset. As we can see, APM still outperforms ADV in defending against stronger attacks. Note, however, that a larger N

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 10: Failed masks of APM.

Table 7: Performance of APM with tunable weights θ.

Dataset	θ	Clean	Noise	ATK(10)	ATK(30)
COCO	Fixed	**52.0**	48.5	47.4	46.8
	Tunable	51.1	**48.8**	**47.9**	**47.2**
INRIA	Fixed	83.0	84.7	83.7	82.0
	Tunable	**87.8**	**92.2**	**90.1**	**89.4**

Table 8: Generalizability of APM under unseen attacks (M = 50).

Seen	Diff. Loc	Sm. Size	Lg. Size	Diff. AR
88.6	81.7	82.5	82.3	81.0

significantly increases the training time of both methods, making them less applicable to large-scale applications.

Fixed or Tunable θ. In Section 4.2, we make the weights of pre-trained object detector tunable during adversarial training. This raises a question: is it better to always let the weights tunable? We answer this question by following the experiment in Section 4.1 but leave the YOLO weights tunable during adversarial training. Also, we extend the experiment in Section 4.2 by fixing the YOLO weights. The results are given in Table 7. For a transfer learning task on INRIA, the tunable YOLO weights indeed improve the overall performance of APM. We have discussed the reason in Section 4.2. However, for a normal task on COCO, we see a slight increase in robustness at the cost of degraded clean performance. It is up to the application to decide which comes in priority.

Generalization. We also investigate how APM performs under *unseen* attacks. APM was trained by the adversarial patches of 1) square shape, 2) size equal to 0.8 of the width of the corresponding objects, and 3) horizontal placement at the middle of the corresponding objects. At test time, we modify the threat model $\mathcal{T}$ such that it generates adversarial patches 1) at different locations (bottom of a bounding box), 2) of 50% smaller sizes, 3) of 12.5% larger sizes, or 4) with a different aspect ratio (2:1) than the ones seen by APM during training on the INRIA dataset. Table 8 shows the results. The robustness of APM drops, but APM still outperforms other baselines (cf. Table 2). Figure 9 shows some example unseen adversarial patches and their corresponding detection results. As we can see, APM successfully generalizes to these challenging cases.

4.9 Failed Cases

Although APM can identify and remove adversarial patches in many challenging situations (see Figure 3), there are some failed cases too. The masks proposed by APM occasionally make object detector output false-positive objects, as Figures 10(a)-(e) shows (where a person is detected at the top-left corner). APM may also fail when (i) the masks do not fully remove adversarial patches (Figures 10(a)(b)), (ii) the masks aggressively cover too many clean pixels (Figures 10(g)(j), where people at the periphery are not detected), (iii) the adversarial patches are dense and obscuring each other (Figures 10(f)(i)), or (iv) the objects being detected is too small (Figure 10(h)). We leave the further investigation of causes and solutions as our future work.

4.10 More Experiments

We have conducted more experiments, in particular the black-box attacks where the adversarial patches were generated using RetinaNet and then applied to YOLO. We found that the black-box physical attack does not seem to transfer well across object detectors as the vanilla YOLO already achieves more than 80% robustness against the attack. However, APM can still improve the robustness up to 10%. Please read the supplementary file [9] for more details.

5 CONCLUSION

We proposed APM that helps a pre-trained object detector defend against physical attacks. APM does not require strong assumptions and is agnostic to the internals of the object detector and threat model. We conducted extensive experiments to verify the effectiveness of APM. We also inspected the masks generated by the MaskNet to understand how APM works. As our future work, we plan to address the failed cases reported in Section 4.9. We will also evaluate the effectiveness of APM in real-world applications, such as the self-driving cars, where security is in high demand.

REFERENCES

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

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

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[8] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018.

[9] Ping-Han Chiang, Chi-Shen Chan, and Shan-Hung Wu. Adversarial pixel masking: Supplementary materials. http://www.cs.nthu.edu.tw/~shwu/pubs/shwu-mm-21-sup.pdf.

[10] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[12] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[13] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

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

[15] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1597–1604, 2018.

[16] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *arXiv preprint arXiv:1705.08475*, 2017.

[17] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[18] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[19] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2507–2515. PMLR, 2018.

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[21] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019.

[22] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International Conference on Machine Learning*, pages 3896–3904. PMLR, 2019.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[26] 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.

[27] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David Wagner. Minority reports defense: Defending against adversarial patches. In *International Conference on Applied Cryptography and Network Security*, pages 564–582. Springer, 2020.

[28] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307. IEEE, 2019.

[29] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[30] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[31] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. *arXiv preprint arXiv:2005.02313*, 2020.

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

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

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[36] Aniruddha Saha, Akshayvarun Subramanya, Koninika Patil, and Hamed Pirsiavash. Role of spatial context in adversarial robustness for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 784–785, 2020.

[37] 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.

[38] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[39] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.

[40] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[41] 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.

[42] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[43] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

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

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

[46] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. *arXiv preprint arXiv:2005.10884*, 2020.

[47] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[48] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[49] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020.

[50] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Evading real-time person detectors by adversarial t-shirt. *arXiv preprint arXiv:1910.11099*, 3, 2019.

[51] Zirui Xu, Fuxun Yu, and Xiang Chen. Lance: A comprehensive and lightweight cnn defense methodology against physical adversarial attacks on embedded multimedia applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 470–475. IEEE, 2020.

[52] Darren Yu Yang, Jay Xiong, Xincheng Li, Xu Yan, John Raiti, Yuntao Wang, HuaQiang Wu, and Zhenyu Zhong. Building towards" invisible cloak": Robust physical adversarial attack on yolo object detector. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 368–374. IEEE, 2018.

[53] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[54] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 421–430, 2019.

[55] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *International Conference on Learning Representations*, 2018.

[56] Zhanyuan Zhang, Benson Yuan, Michael McCoyd, and David Wagner. Clipped bagnet: defending against sticker attacks with clipped bag-of-features. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 55–61. IEEE, 2020.

[57] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019.

[58] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[59] Guangzhi Zhou, Hongchao Gao, Peng Chen, Jin Liu, Jiao Dai, Jizhong Han, and Ruixuan Li. Information distribution based defense against physical attacks on object detection. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020.