sensors

MDPI

Article

Extended Spatially Localized Perturbation GAN (eSLP-GAN) for Robust Adversarial Camouflage Patches †

Yongsu Kim [1,2,‡], Hyoeun Kang [2,‡], Naufal Suryanto [2], Harashta Tatimma Larasati [2,3], Afifatul Mukaroh [2] and Howon Kim [1,2,*]

1 SmartM2M, Busan 46300, Korea; yongsu@smartm2m.co.kr
2 Department of Information Convergence Engineering, School of Computer Science and Engineering, Pusan National University, Busan 609735, Korea; hyoeun0915@pusan.ac.kr (H.K.); naufalso@pusan.ac.kr (N.S.); harashta@pusan.ac.kr (H.T.L.); afifatul.mukaroh@pusan.ac.kr (A.M.)
3 School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40116, Indonesia
* Correspondence: howonkim@pusan.ac.kr
† This paper is an extended version of our paper published in: Kim, Y.; Kang, H.; Mukaroh, A.; Suryanto, N.; Tatimma Larasati, H.; Kim, H. Spatially Localized Perturbation GAN (SLP-GAN) for Generating Invisible Adversarial Patches. In Proceedings of the International Conference on Information Security Applications, Jeju Island, Korea, 26–28 August 2020; pp. 3–15.
‡ These authors contributed equally to this work.

Abstract: Deep neural networks (DNNs), especially those used in computer vision, are highly vulnerable to adversarial attacks, such as adversarial perturbations and adversarial patches. Adversarial patches, often considered more appropriate for a real-world attack, are attached to the target object or its surroundings to deceive the target system. However, most previous research employed adversarial patches that are conspicuous to human vision, making them easy to identify and counter. Previously, the spatially localized perturbation GAN (SLP-GAN) was proposed, in which the perturbation was only added to the most representative area of the input images, creating a spatially localized adversarial camouflage patch that excels in terms of visual fidelity and is, therefore, difficult to detect by human vision. In this study, the use of the method called eSLP-GAN was extended to deceive classifiers and object detection systems. Specifically, the loss function was modified for greater compatibility with an object-detection model attack and to increase robustness in the real world. Furthermore, the applicability of the proposed method was tested on the CARLA simulator for a more authentic real-world attack scenario.

Keywords: adversarial patch; generative adversarial networks; camouflage

Citation: Kim, Y.; Kang, H.; Suryanto, N.; Larasati, H.T.; Mukaroh, A.; Kim, H. Extended Spatially Localized Perturbation GAN (eSLP-GAN) for Robust Adversarial Camouflage Patches. *Sensors* **2021**, *21*, 5323. https://doi.org/10.3390/s21165323

Academic Editor: Ilsun You

Received: 28 May 2021
Accepted: 3 August 2021
Published: 6 August 2021

1. Introduction

In addition to their use for computer vision tasks [1], such as facial recognition, object detection [2], and image segmentation, deep neural networks (DNNs) are known for their various potential applications. In particular, the use of DNNs have been found in lane line detection and traffic sign recognition in self-driving cars [3–5], Unmanned Aerial Vehicle (UAV)-based monitoring system [6], and even to the usage in the emerging technologies and framework such as big data [7] and blockchain [8] in the industrial network infrastructure. Despite their promising progress, recent research has shown examples of attacks on various intelligent systems [9,10]. In particular, it is known that DNNs included in those systems are vulnerable to adversarial attacks, which lead deep learning models to make incorrect predictions with high confidence.

Adversarial examples are typically created by adding a modicum of noise to the original image. The majority of early research efforts have concentrated on adversarial examples against image classification in digital environments [11–13]. Brown et al. [14] proposed a printable universal adversarial patch to deceive a classifier in the physical

world. The performance of the classifier was degraded by attaching adversarial patches to a physical object.

However, these adversarial patches could be easily identified by human vision because they prioritized attack performance rather than visual fidelity. Liu et al. [15] proposed a perceptual-sensitive GAN (PS-GAN) for generating adversarial patches through generative adversarial networks (GANs) that could learn and approximate the distribution of original instances. Utilizing a patch-to-patch translation and an attention mechanism, the PS-GAN simultaneously improved the visual fidelity and the capability of the adversarial patch to attack. Despite the considerable effort, adversarial patches generated by a PS-GAN retain an unnatural appearance and, thus, are conspicuous to human vision.

Object detection models have also emerged as critical components of computer vision systems for real-time recognition tasks. Recent studies have revealed that object detection models are vulnerable to adversarial attacks. Eykholt et al. [16] proposed a disappearance attack that caused physical objects to be ignored by a detector in the scene and a creation attack to detect nonexistent objects by using posters or stickers. Zhao et al. [17] implemented adversarial examples that can attack detectors within distances varying from 1 to 25 m and angles from -60° to 60°. However, these object detection model attack methods also have low visual fidelity, similar to the case of the classification model.

Attacks on object detection models are more challenging than those on classification models. While the classifiers focus only on a single label, object detectors would consider multiple labels and the relative positions of the detectors and objects. In addition, because the objects move dynamically, attackers must consider their distance, viewing angle, and illumination.

In our previous work, we proposed a spatially localized perturbation GAN (SLP-GAN) [18] that generated a spatially localized perturbation as an adversarial patch to attack classifiers. It had the advantage that it could generate visually natural patches while maintaining a high attack success rate. The patch region to attach to a target object was extracted from the most representative area of the target object using the Grad-CAM algorithm [19] to improve the ability to attack. However, there are two main limitations of SLP-GAN. First, SLP-GAN can attack only classification models, not object detection models. Second, SLP-GAN has a rather weak attack performance in the physical world because it does not consider the various physical world transformations.

In this paper, we propose an extended spatially localized perturbation GAN (eSLP-GAN) that can attack both classifiers and detectors. We modify the loss function of SLP-GAN to enable attack on the object detection models. We also improve the robustness of adversarial patches by applying terms that consider various transformations and printability in the real-world scenario.

2. Related Work

Attacks on deep neural networks (DNNs) have raised concerns for many researchers owing to their potentially fatal impact. Even a slight perturbation added to an image may cause the computer vision model to behave unexpectedly. When successful, this attack type has been proven to disrupt a wide range of computer vision tasks, ranging from simple image classification to object detection models. Recent works have identified the risks posed even in the real-world case, such as adversarial patches being attached to physical objects. In this section, we provide an overview of the state-of-the-art image classification and object detection models that are evaluated using the proposed method. Then, we provide the overview of related adversarial attacks in both the digital and physical domains, as well as generative adversarial networks (GANs), which have recently gained interest for use as methods to generate adversarial attacks.

2.1. Classification Model

Image classification or image recognition is the primary task in computer vision. Given image x, the goal is to predict the label $y \in Y$, where Y is the set of defined labels

corresponding to the image. The convolutional neural network (ConvNet) is the most common neural network architecture used in this field.

VGGNet [20] is a classic Deep ConvNet with its celebrated VGG-16 architecture, that was proposed to learn the effect of convolutional network depth on accuracy. Many researchers have used it as a baseline for large-scale image recognition.

Residual Network (ResNet) architecture [21] introduces a residual or skip connection function to the convolutional neural network that allows the deeper model to be trained. This novel architecture has been an inspiration for later state-of-the-art CNN-based models.

MobileNet [22] and MobileNetv2 [23] are convolution neural networks specifically designed for mobile or embedded vision applications that have limited computing resources. It introduces tunable hyperparameters that efficiently make trade-offs between latency and accuracy.

EfficientNet [24], the state-of-the-art ConvNet, was proposed as a method to scale up the model accuracy effectively by meticulously balancing the network width, depth, and resolutions. The best EfficientNet model can achieve 84.3% top-1 accuracy on a large-scale ImageNet dataset while being 8.4-times smaller and 6.1-times faster than the best-performing ConvNets [24].

2.2. Object Detection Model

Object detection is a computer vision task that detects instances of objects of a certain class within an image. Given image x, the goal is to predict each object's location, which is represented as bounding box B and label y that corresponds to the object. It is also known for combining the object localization and multi-label classification tasks. The common object detection framework consists of a feature extraction module from a pre-trained classification model added to a detection head module. The state-of-the-art object detection approaches can be categorized into two types: two-stage and one-stage object detection.

Two-stage object detection consists of a region of interest (RoI) proposal (where the candidate object is located) and an image classifier to process the candidate region. The R-CNN family [25–28] represents the most well-known methods using these concepts. The initial R-CNN [25] used a selective search method to propose RoIs and process each candidate region independently, which rendered the model computationally expensive and slow.

Fast R-CNN [26] aggregates the RoI into one unit and uses one CNN forward pass over the entire image, which shares the same extracted feature matrix as the RoI. The shared computation makes the model faster than the original R-CNN; however, it remains inefficient because the RoIs are produced by other models. Faster R-CNN [27] overcomes this issue by unifying the model and proposing a region proposal network (RPN) as the replacement selective-search method.

One-stage object detection is more efficient and has a higher inference speed. It directly predicts the output in a single step. You Only Look Once (YOLO) [29] was an early method for one-stage object detection. The latest version of YOLO, namely the YOLOv4 [30], is proposed as a major improvement to the YOLO family in terms of both speed and accuracy. EfficientDet [31] is another state-of-the-art one-stage object detection method. EfficientDet implements a weighted bi-directional feature pyramid network (BiFPN) for fast multi-scale feature fusion and uses the EfficientNet model as the backbone [31].

2.3. Adversarial Attack in Digital Environments

Adversarial attacks in digital environments mainly consist of adversarial perturbation methods that add pixel-level noise to the input images. Adversarial perturbation is tiny perturbation added to the image that causes the target model to misclassify with high confidence [32]. Suppose that a trained model M can correctly classify an original image x as $M(x) = y$. By adding to x a slight perturbation η, that could not be recognized by human vision, an adversary could generate an adversarial example $x' = x + \eta$ such that $M(x') \neq y$.

Based on the adversary's knowledge, adversarial attacks can be categorized as white-box and black-box attacks as follows:

- White-box attacks: The adversary knows the structure and all the parameters of the target model, including the training data, model architectures, hyper-parameters, and model weights. FGSM [11], Deepfool [33], C&W [12], and PGD [13], are examples of popular techniques for generating adversarial examples based on white-box attacks. Most white-box attack methods rely on model gradients to compute the perturbations.
- Black-box attacks: The adversary is ignorant of the target model and considers the target as a black-box system. The adversary analyzes the target model by observing only the output based on a given series of adversarial examples. White-box attack methods can be used in black-box attack scenarios by relying on the transferability of adversarial perturbations. The adversary uses a substitute model trained with the same output as the target model and generates adversarial examples with the white-box setting. The generated adversarial example is then used and evaluated in the target model. Using the same approach, we evaluate the transferability of our method to other models. In addition to relying on transferability, some black-box attack methods have been specifically designed for this setting. Gradient-estimation-based [34–36] and local-search-based [37–39] methods require no knowledge of the target model.

2.4. Adversarial Attack in Physical Environments

Adversarial perturbations are typically applied to digital environments in real-world attack scenarios. Conversely, an adversarial patch is feasible in physical environments because it can be attached to a specific location on a real object. Brown et al. [14] introduced a method to create a universal, robust, and targeted adversarial patch for real-world applications. The patch can successfully fool a classifier with a variety of scenes, transformations, and outputs to any target class.

Another proposal is a black-box adversarial-patch-based attack called DPatch. It can fool mainstream object detectors (e.g., Faster R-CNN and YOLO), which cannot be accomplished by the original adversarial patch [40]. It simultaneously attacks the bounding box regression and object classification.

Slightly different from adversarial patches that focus on being applied to physical objects, physical attack methods focus on robustness in the real world. Athalye et al. [41] presented the expectation over transformation (EOT), an algorithm for synthesizing robust adversarial examples over a chosen distribution of transformations. They synthesized adversarial examples that were robust to noise, distortion, and affine transformations and demonstrated the presence of 3D adversarial objects in the physical world.

Eykholt et al. [42] proposed robust physical perturbation (RP_2), an adversarial attack algorithm in the physical domain that generates robust adversarial perturbations under various physical conditions. The RP_2 algorithm is an enhanced version of the EOT algorithm that considers printability in the real world. They added a term that models printer color-reproduction errors based on the non-printability score (NPS) by Sharif et al. [43]. They can create adversarial patches that are close to the actual printable colors by minimizing the NPS.

The success of the EOT algorithm was also proven by Chen et al. [44], who proposed a robust physical adversarial attack on a Faster R-CNN object detector called ShapeShifter. The attack shows that EOT can also be applied in object detection tasks and significantly enhances the robustness of the resulting perturbation captured under different conditions. They tested the attack on real-world driving and showed that the perturbed stop signs can constantly fool the Faster R-CNN object detector. However, most adversarial patches or physical perturbation methods produce a visible pattern that can be easily recognized by human vision. Therefore, we focus on the invisibility of the patch while maintaining robustness in a real-world implementation by utilizing the EOT and RP_2 algorithms for our eSLP-GAN.

2.5. Generative Adversarial Networks (GANs)

Recent studies have shown that adversarial perturbation and patch generation mostly rely on optimization schemes. To generate a perceptually more realistic perturbation efficiently, researchers have proposed many variants of generative adversarial networks (GANs) [45]. Earlier, Goodfellow et al. [45] introduced a framework for estimating generative models via an adversarial process, which simultaneously trained two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. This discovery, coined as a generative adversarial network (GAN), has had a significant impact on data generation.

In the context of adversarial perturbation generation, Xiao et al. [46] proposed AdvGAN to generate adversarial perturbations using generative adversarial networks (GANs). It can learn and approximate the distribution of the original instances. Once the generator is trained, it can generate perturbations efficiently for any instance to potentially accelerate adversarial training for defense. This attack was placed first with an accuracy of 92.76% in a public MNIST black-box attack challenge. Liu et al. [15] proposed a perceptual-sensitive GAN (PS-GAN) that simultaneously enhances the visual fidelity and the attacking ability of an adversarial patch.

To improve the visual fidelity, we treat patch generation as a patch-to-patch translation via an adversarial process, feeding a seed patch, and outputting a similar adversarial patch that has a moderately high perceptual correlation with the attacked image. To further enhance the attacking ability, an attention mechanism coupled with adversarial generation was introduced to predict the critical attacking areas for patch placement. The limitation of PS-GAN is that it uses a seed patch that is quite different from the original image and, thus, may still not seem visually natural. The proposed method generates a patch from the original input image using an attention map that maximizes the attack rate while maintaining high visual fidelity.

3. Proposed Method

In this section, we describe the problem definition for attacking both the classification model and the object detection model and introduce the extended spatially localized perturbation GAN (eSLP-GAN) framework for generating robust adversarial camouflage patches. All mathematical notations used in eSLP-GAN can be referred to in Table 1.

Table 1. List of symbols in this paper.

Symbol	Meaning	Symbol	Meaning
h_θ	Hypothesis function of classification or object detection task	L	Total loss function of eSLP-GAN
x	Input data	L_{ADV}	Adversarial loss function of eSLP-GAN
x_A	Adversarial example	L_{ATK}	Attacking ability loss function of eSLP-GAN
y	Ground truth (label) of input data x	L_{PTB}	Perturbation loss function of eSLP-GAN
y_t	Attack target label	L_{NPS}	Printability loss function of eSLP-GAN
p	Spatially localized patch	l_{cls}	Classification loss function applied to the target model M
G	Generator of eSLP-GAN	l_{obj}	Objectness loss function applied to the target model M
D	Discriminator of eSLP-GAN	T	Chosen distribution of transformations
M	Attack target model	t	Transformation functions

3.1. Problem Definition

This section defines the problem of generating adversarial camouflage patches for classification and object detection models. Assume that h_θ denotes a hypothesis function capable of performing a classification or object detection task. x denotes input data to h_θ with a corresponding label of y, which satisfies the equation $h_\theta(x) = y$. The label is constructed differently depending on the task. There are two types of attack methods: untargeted and targeted. The purpose of the adversarial attack in the proposed approach is to generate the adversarial example x_A. The untargeted attack can be expressed as $h_\theta(x_A) \neq y$.

The targeted attack is described as $h_\theta(x_A) = y_t$, where y_t is the attack target label. In addition, x_A should be comparable to the original input data x in terms of visual fidelity. The adversarial example x_A is constructed by attaching a spatially localized patch p to the original input data x, as specified by the equation $x_A = x + p$. In the study's terminology, "spatially localized" refers to the property of perturbation that is applied to a subset of the input image rather than the entire image. The following section describes the proposed method for generating adversarial camouflage patches that meet the aforementioned requirements.

3.2. eSLP-GAN Framework

3.2.1. eSLP-GAN

The proposed eSLP-GAN architecture consists of three primary components: a generator G, a discriminator D, and a target model M. The target model M approximates the hypothesis function h_θ that performs the classification task or the object detection task mentioned above. Generator G takes the original data x as input and generates a perturbation over the entire region of the input image. In contrast to previous proposals [46], this method employs additional steps to generate spatially localized perturbations for adversarial camouflage patches that can be attached to a specific location.

We leverage the Grad-CAM algorithm [19] to extract patch regions and apply the generated perturbation to only the extracted region. The method is used for extracting the most representative area that affects the target model to determine the input image as a specific label. Our assumption is that attack performance will be improved in the case of attaching the adversarial patch to the representative area from an input image rather than attaching it to another area. As a result, spatially localized perturbations can be treated as adversarial patches p.

The discriminator D is responsible for differentiating the original data x and the adversarial example $x_A = x + p$. D encourages G to generate perturbation that visually conforms to the original data. Furthermore, G should be capable of deceiving the target model M. Thus, the entire structure of eSLP-GAN has four loss functions: adversarial loss L_{ADV}, attacking ability loss L_{ATK}, perturbation loss L_{PTB}, and printability loss L_{NPS}. The adversarial loss L_{ADV} is expressed by the following equation:

$$L_{ADV} = \mathbb{E}_x \log D(x) + \mathbb{E}_x \log(1 - D(x_A)). \tag{1}$$

As observed in the preceding equation, discriminator D aims to distinguish the adversarial example x_A from the original data x. Note that D promotes G to generate perturbation with visual fidelity in compliance with the above equation.

The attacking loss L_{ATK} is configured differently depending on the attack method and the type of target model M. We used the EOT algorithm to construct the attacking loss to improve robustness in real-world situations. In the case where M is a classification model, the attacking loss can be defined as follows:

$$L_{ATK} = \begin{cases} -\mathbb{E}_{x,t\sim T}\ell_{cls}(M(t(x_A)), y), & \text{if untargeted attack.} \\ \mathbb{E}_{x,t\sim T}\ell_{cls}(M(t(x_A)), y_t), & \text{otherwise.} \end{cases} \tag{2}$$

where y is the original label of the input data x and y_t is the target label. ℓ_{cls} is the classification loss function (e.g., cross-entropy loss) applied to the target model M. We use a chosen distribution T of transformation functions t, and fool the target model to misclassify $M(x_A)$ or classify $M(x_A)$ as the target label y_t by minimizing L_{ATK}. Differently expressed, we can generate robust adversarial examples that remain adversarial under image transformations that occur in the real world, such as angle and viewpoint changes.

In the case of the object detection model, it predicts bounding boxes that determine the location of objects in an input image. In general, each bounding box contains the objectness score, box position, and class probability vector [47]. We use two types of object detection losses: ℓ_{obj} and ℓ_{cls}. ℓ_{obj} represents the maximum objectness score over the predictions for the entire image. ℓ_{cls} is the classification loss, which is the concept also used in the classification model. We define the attacking loss L_{ATK} using ℓ_{obj} and ℓ_{cls} in the case where M is an object detection model as follows:

$$L_{ATK} = \begin{cases} \ell_{obj} - \mathbb{E}_{x,t\sim T}\ell_{cls}(M(t(x_A)), y), & \text{if untargeted attack.} \\ -\ell_{obj} + \mathbb{E}_{x,t\sim T}\ell_{cls}(M(t(x_A)), y_t), & \text{otherwise.} \end{cases} \tag{3}$$

This differs from the attacking loss of the classification model by the addition of the objectness score. In the case of an untargeted attack, the target model considers the bounding box empty of objects by minimizing ℓ_{obj}. In contrast, in the case of a targeted attack, the target model is induced to classify x_A as the target label y_t by minimizing ℓ_{cls}, and the objectness score for the target label is increased by maximizing ℓ_{obj}.

Additionally, we define the perturbation loss L_{PTB} using a soft hinge loss [48] on the L_2 norm to bound the magnitude of the generated perturbation as follows:

$$L_{PTB} = \mathbb{E}_x \max(0, \|G(x)\|_2 - c), \tag{4}$$

where c represents the user-specified bound and serves to stabilize the GAN training. To account for printability, we add a term L_{NPS} to our objective function that models the fabrication error. This term is based on the non-printability score (NPS) by Sharif et al. [43].

$$L_{NPS} = \prod_{p' \in P'} |p_i - p'|, \tag{5}$$

where $p' \in P'$ represents the printable RGB triplets and p_i is each pixel of the adversarial patch p. We can generate adversarial patches close to printable color by minimizing L_{NPS}. Finally, combined with the above visual fidelity and ability to attack the target model, the eSLP-GAN loss function can be expressed as follows:

$$L = L_{ADV} + \alpha L_{ATK} + \beta L_{PTB} + \gamma L_{NPS}, \tag{6}$$

where $\alpha > 0$, $\beta > 0$, and $\gamma > 0$ control the contribution of each loss function. In our eSLP-GAN, the generator G and discriminator D are trained by solving a minimax game denoted by the equation $\min_G \max_D L$. As a result, generator G can generate spatially localized perturbations suitable for adversarial patches while maintaining visual fidelity and attacking ability. The overall architecture of the eSLP-GAN is depicted in Figure 1, and Algorithm 1 describes the process for training the eSLP-GAN framework.

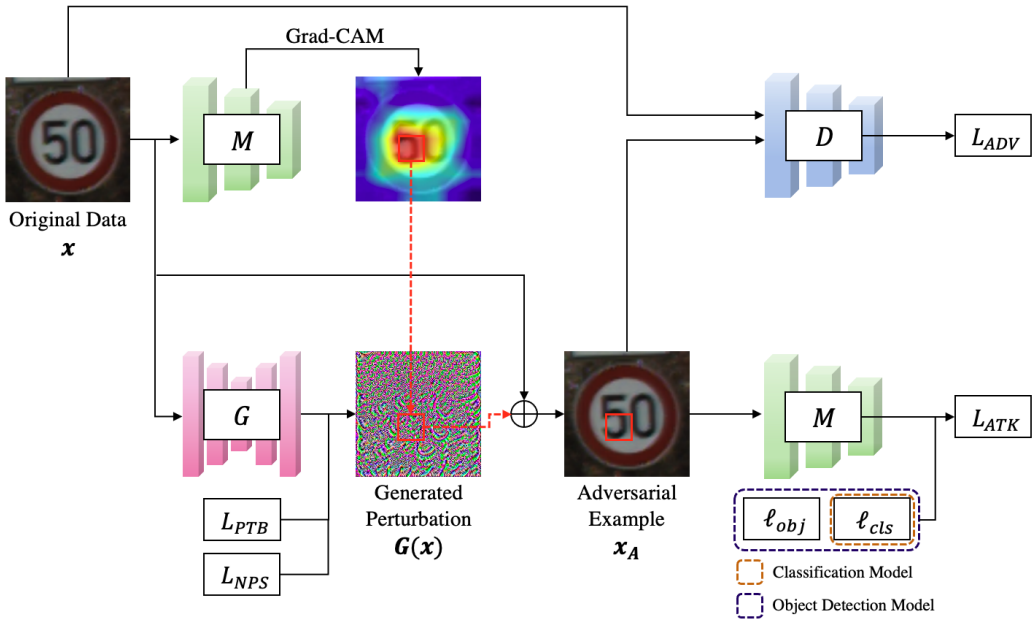

Figure 1. The eSLP-GAN framework consists of the generator *G*, the discriminator *D*, and the target model *M*.

Algorithm 1 Training process of the eSLP-GAN Framework.

Input: training image set $X_{image} = \{x_i \mid i = 1, \ldots, n\}$
Output: spatially localized patches $P = \{p_i \mid i = 1, \ldots, n\}$
for the number of training epochs **do**
  for k steps **do**
    sample minibatch of m images $\phi_x = \{x_1, \ldots, x_m\}$.
    generate minibatch of m adversarial perturbations $\phi_x^G = \{G(x_1), \ldots, G(x_m)\}$.
    obtain activation maps $M(\phi_x)$ by *Grad-CAM*.
    extract spatially localized patches $P = \{G(x_i)_{M(x_i)} \mid i = 1, \ldots, n\}$.
    create adversarial examples $x_A = \{x_i + p_i \mid i = 1, \ldots, n\}$.
    update D to $\max_D L$ with G fixed.
  end for
  sample minibatch of m images $\phi_x = \{x_1, \ldots, x_m\}$.
  create adversarial examples x_A (same as above).
  update G to $\min_G L$ with D fixed.
end for

3.2.2. Spatial Localization

Grad-CAM [19] is suitable for extracting a representative area from an input image. Grad-CAM produces "visual explanations" from convolutional neural network (CNN)-based models. It generates a localization map highlighting the important regions in the input image by using the gradients of any target label flowing into the final convolutional layer. To obtain a localization map according to the target label (class), defined as a class-discriminative localization map, we used a gradient of the score for class y^c with respect to the feature map activation A^k of the target layer.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{7}$$

Here, α_k^c indicates the neuron importance weight associated with feature map k for target class c. We take a weighted sum of the forward activation maps, A, with weights α_k^c, and combine them with a ReLU to obtain counterfactual explanations that have a positive effect on the target class.

$$L^{c}_{Grad-CAM} = ReLU(\sum_{k} \alpha^{c}_{k} A^{k}) \quad (8)$$

After obtaining a localization map with class activation mapping (CAM), we take the bounding boxes for the activated regions by filtering for values greater than 80% of the CAM intensity. The acquired bounding boxes are the most representative areas of the target model decision for the input image. Figure 2 shows examples of Grad-CAM visualizations with bounding boxes for traffic signs.

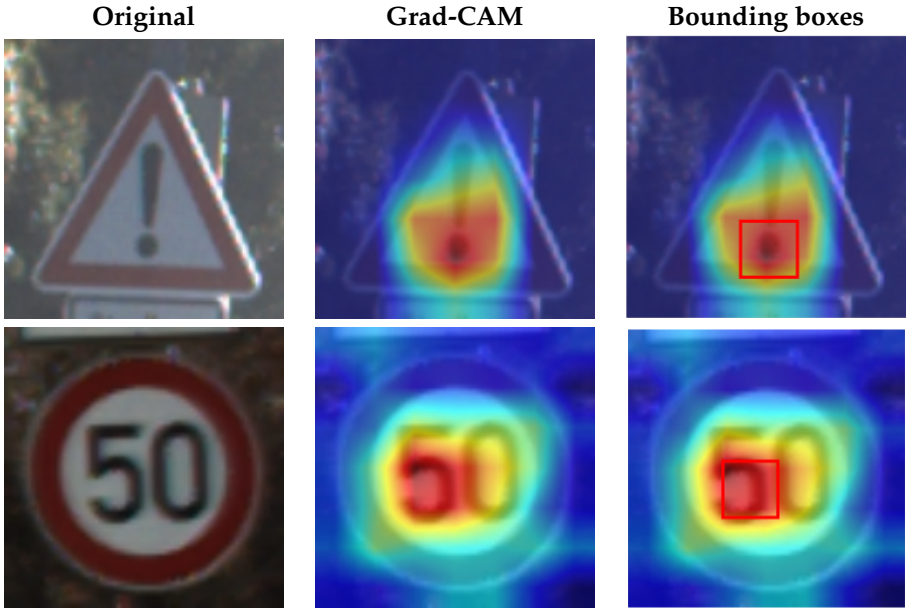

Figure 2. Samples of Grad-CAM visualizations and bounding boxes on traffic signs.

4. Experiments

4.1. Experimental Setup

4.1.1. Model Structure

To configure our eSLP-GAN framework, we utilize structures for generator G and discriminator D with image-to-image translation, as in [48,49]. Generator G consists of a U-Net [50] structure, which has an encoder–decoder network. U-Net has the advantage of generating high-resolution images by adding skip connections to construct a contracting path to capture the context and a symmetric expanding path that enables precise localization.

We adopted U-Net architecture as a generator because it is suitable for generating adversarial patches not only similar to input image but also with a relatively good attack performance. This is due to its ability to capture the context, enabling it to extract information from the low level to the high level of the input image. Furthermore, discriminator D uses a common CNN architecture in the form of Convolution–BatchNorm–ReLU to encourage generator G to increase the generation ability.

4.1.2. Target Models

In this experiment, we used two types of target models: a classification model and an object detection model. As the classification models, we used VGG16 [20], ResNet50 [21], MobileNetV2 [23], and EfficientNetB0 [24], which exhibit good performance in image classification problems. For object detection, we used Faster R-CNN [27], YOLOv4 [30], and EfficientDetD0 [31], which are state-of-the-art object detection models.

4.1.3. Implementation Details

For the implementation, we utilized PyTorch, with testing performed on a Linux workstation (Ubuntu 18.04) with four NVIDIA Titan XP GPUs, Intel Core i9-7980XE CPU,

and 64 GB of RAM. We trained the eSLP-GAN for 250 epochs with a batch size of 128, with an initial learning rate of 0.001, and dropped it by 10% every 50 epochs.

4.2. Experiment Results

4.2.1. Classification Models

In the case of classification models, we conducted a physical-world attack experiment to verify that the eSLP-GAN was enhanced to be more robust to the physical environment than the previous version. We used Korean traffic sign mock ups approximately 15 × 15 cm in size and from nine classes {Bike Path, No Entry, Danger, Motorway, No Bicycle, No Jaywalking, Roadworks, Railroad Crossing, and School Zone}, as shown in Table 2.

Table 2. Korean traffic sign mock ups used in the physical-world attack experiment against classification models.

		
Bike Path	No Entry	Danger
		
Motorway	No Bicycle	No Jaywalking
		
Roadworks	Railroad Crossing	School Zone

We first took approximately 150 pictures of each traffic sign mock up at varying distances {1, 1.25, 1.5, 1.75, and 2 m} and angles {0°, 15°, 30°, −15°, and −30°}. Figure 3 shows each point at which traffic sign images with different distances and angles were taken. We verified the robustness of the eSLP-GAN under various physical conditions by altering the environmental settings with different distances and angles.

Next, we trained the VGG16, ResNet50, MobileNetV2, and EfficientNetB0 as target models using these images that were divided into a training dataset (approximately 100 images) and a validation dataset (approximately 50 images). To configure a test dataset for both the original images and the adversarial examples equivalently, we randomly selected again approximately 100 pictures as the test dataset, and adversarial patches were generated using eSLP-GAN for each image and each target model.

In this case, we generated half of the patches using an untargeted attack and the remainder using a targeted attack by selecting a random label. After printing these patches, we attached them to the traffic sign mock ups and again took photos of each mock up with the patches attached. Table 3 presents the classification accuracy of the target models on normal validation images and the adversarial examples with attached patches using the test dataset.

Table 4 shows examples of traffic sign mock ups with and without adversarial patches and the classification results. The previous SLP-GAN version had an attack performance that reduced the classification accuracy of the target model from 97.8% to 38.0% [18]. We observe in Table 3 that the enhanced version, eSLP-GAN, had a higher attack performance than the previous version. In other words, eSLP-GAN is robust against real-world distortions, such as lighting conditions, various distances, and angles, by additionally applying the EOT algorithm and NPS loss.

Table 5 represents the physical-world attack result examples of the Roadworks' traffic sign mock ups with various transformations on the VGG16 target model. Table 6 presents the attack results of the traffic sign mock ups of a targeted attack and untargeted attack at different distances and angles. We can conduct both targeted and untargeted attacks by adjusting the attacking loss term of the eSLP-GAN, as shown in Equation (2). In the case of Table 6, we set the target label as "Motorway".

Table 3. Classification accuracy of target classification models on normal images and adversarial examples.

Target Model	Classification Accuracy	
	Original	**Adversarial**
VGG16	98.24%	**31.49%**
ResNet50	97.48%	**29.81%**
MobileNetV2	96.91%	**36.58%**
EfficientNetB0	96.35%	**34.24%**

Table 4. The physical-world attack result examples of traffic sign mock ups with and without adversarial patches generated by eSLP-GAN.

Original	**Adversarial**	**Original**	**Adversarial**
			
Roadworks	Danger	Bike Path	No Jaywalking
			
Motorway	School Zone	Railroad Crossing	No Entry

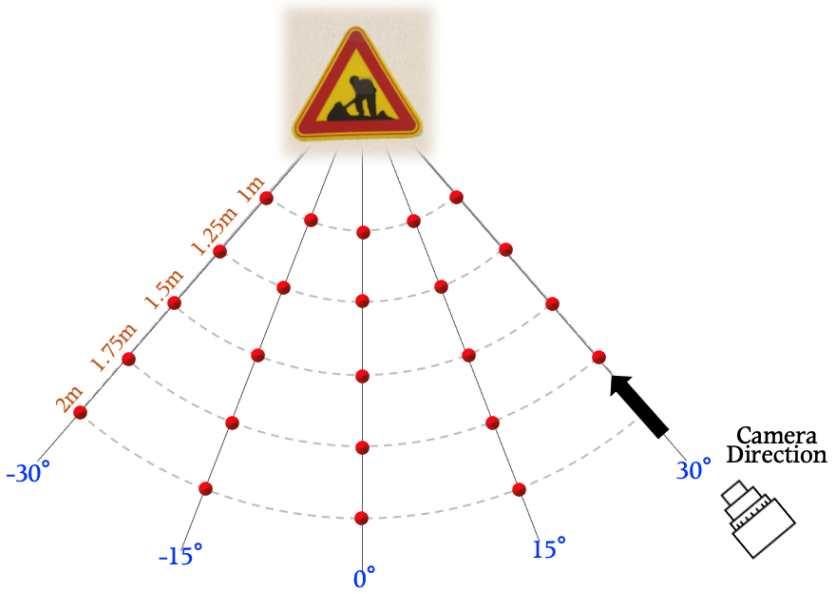

Figure 3. Physical-world attack experiment settings against classification models with different distances and angles.

Table 5. The physical-world attack result examples of 'Roadworks' traffic sign mock ups with various transformations.

Original	Adversarial	
		
Roadworks	Danger	Danger
		
Roadworks	Danger	Roadworks

We also conducted transferability experiments between each target model. Table 7 shows the classification accuracy of each target model using transferability. First, we generate adversarial patches for the source models and, then, attached these patches to traffic sign mock ups. Note that adversarial examples against other source models significantly reduced the classification accuracy of the target models. This means that our eSLP-GAN encourages transferability among different target models in physical environments.

Table 6. The physical-world attack result of 'Roadworks' traffic sign mock ups of targeted attack and untargeted attacks with different distances and angles against VGG16.

Distance	Angle	Dangers	(Conf.)	Motorway	(Conf.)	Untargeted	(Conf.)
1 m	30°	Dangers	(0.71)	Motorway	(0.85)	Roadworks	(0.88)
	15°	Dangers	(0.93)	Motorway	(0.87)	Railroad Crossing	(0.78)
	0°	Dangers	(0.87)	Motorway	(0.93)	Railroad Crossing	(0.64)
	−15°	Dangers	(0.90)	Motorway	(0.61)	Railroad Crossing	(0.73)
	−30°	Roadworks	(0.91)	Motorway	(0.70)	Roadworks	(0.87)
1.25 m	30°	Dangers	(0.81)	Motorway	(0.65)	Danger	(0.91)
	15°	Dangers	(0.84)	Motorway	(0.70)	Danger	(0.83)
	0°	Dangers	(0.87)	Roadworks	(0.75)	Roadworks	(0.58)
	−15°	Dangers	(0.86)	Motorway	(0.71)	Danger	(0.64)
	−30°	Dangers	(0.77)	No Entry	(0.40)	Railroad Crossing	(0.70)
1.5 m	30°	Dangers	(0.78)	Motorway	(0.85)	Railroad Crossing	(0.91)
	15°	Dangers	(0.79)	Motorway	(0.91)	Railroad Crossing	(0.91)
	0°	Dangers	(0.80)	Motorway	(0.89)	Roadworks	(0.62)
	−15°	Dangers	(0.74)	Roadworks	(0.68)	Bike path	(0.80)
	−30°	Dangers	(0.68)	Motorway	(0.89)	No Jaywalking	(0.70)
1.75 m	30°	Dangers	(0.61)	Motorway	(0.68)	No Jaywalking	(0.63)
	15°	Dangers	(0.58)	Motorway	(0.62)	Dangers	(0.81)
	0°	Roadworks	(0.55)	Motorway	(0.61)	Railroad Crossing	(0.93)
	−15°	Dangers	(0.56)	Motorway	(0.61)	Railroad Crossing	(0.96)
	−30°	Dangers	(0.61)	No Entry	(0.55)	No Entry	(0.91)
2 m	30°	Motorway	(0.69)	Motorway	(0.60)	Railroad Crossing	(0.96)
	15°	Dangers	(0.63)	Motorway	(0.61)	Railroad Crossing	(0.94)
	0°	Roadworks	(0.51)	Motorway	(0.55)	Roadworks	(0.94)
	−15°	Dangers	(0.61)	Motorway	(0.58)	Danger	(0.91)
	−30°	Railroad Crossing	(0.55)	Motorway	(0.51)	Railroad Crossing	(0.98)
Success rates			73%		67%		78%

Table 7. Classification accuracy of each target classification model using the transferability.

		Target Models			
		VGG	**ResNet**	**MobileNet**	**EfficientNet**
Source Models	VGG	**31.49%**	57.32%	49.81%	54.38%
	ResNet	42.78%	**29.81%**	60.57%	53.13%
	MobileNet	56.04%	47.28%	**36.58%**	55.91%
	EfficientNet	44.38%	53.12%	50.45%	**34.24%**

4.2.2. Object Detection Models

For the classification models, we evaluated the attack performance of eSLP-GAN in a real-world environment using custom-trained models with traffic sign mock ups. In the case of the object detection models, we evaluated eSLP-GAN by expanding the scope of pre-trained models with large datasets, such as COCO [51]. In an attack scenario of object detection models, we applied an untargeted attack to classify a traffic sign as a different class or a targeted attack to classify a specific class for a label existing in the COCO dataset. Instead of various restrictions, such as financial and time constraints existing in the real world, we used a CARLA simulator [52] that was photo-realistic to increase the efficiency of the experiment. The CARLA simulator is built on the Unreal Engine and provides various maps, vehicle models, and traffic signs, in addition to an autonomous driving system.

We experimented using various maps and traffic signs from the CARLA simulator, as shown in Figure 4. First, we captured image datasets of various traffic signs using the CARLA simulator. Next, we added the "traffic sign" label to the object detection models pre-trained on the COCO dataset and fine-tuned the models using the collected traffic signs image dataset. Next, we generated adversarial camouflage patches using eSLP-GAN

extracted for the traffic sign labels. Here, to reproduce the experiment in the real world, the patches were not applied directly to the image, but we used a method of attaching the patch to the corresponding area in the texture map of the traffic sign object. That is, we use an approach similar to attaching a patch directly to a traffic sign in the real world.

(**a**) (**b**) (**c**)

Figure 4. Examples of photo-realistic environments in the CARLA simulator. (**a**–**c**) show the capture of some different panoramas and shades in the CARLA simulator.

We collected approximately 1000 images of traffic signs at various locations in the CARLA simulator, and we split the dataset into 700 training images, 200 validation images, and 100 test images. Next, we used the training datasets and validation datasets for fine-tuning the object detection models, and we used the test dataset to evaluate the object detection models as shown in Tables 8 and 9. To illustrate an object detection model attack used in autonomous driving, we captured images of traffic signs from the perspective of a vehicle running in a lane.

Next, we measured the mean average precision (mAP) with the IoU threshold set to 0.5 of the object detection model for traffic signs by attaching patches that were generated by the eSLP-GAN to target each object detection model. We used state-of-the-art object detection models, Faster R-CNN [27], YOLOv4 [30], and EfficientDet [31]. Figure 5 illustrates the collected original images and adversarial examples with patches, while Table 8 details the mAP of each object detection model. As shown, our eSLP-GAN performed well against both the classification and object detection models.

Table 8. mAP of target object detection models on normal images and adversarial examples.

Target Model	**mAP**	
	Original	**Adversarial**
Faster R-CNN	83.74%	**48.91%**
EfficientDet-D0	97.27%	**55.38%**
YOLOv4	98.13%	**61.55%**

Table 9. mAP of each target object detection model using the transferability.

		Target Models		
		Faster R-CNN	**EfficientDet-D0**	**YOLOv4**
Source Models	Faster R-CNN	**48.91%**	73.58%	75.39%
	EfficientDet-D0	62.14%	**55.38%**	72.15%
	YOLOv4	60.48%	71.38%	**61.55%**

As with the classification models, we experimented with the transferability in object detection models. We collected approximately 100 adversarial examples to be used against each source model and evaluated the mAP of each target model. Table 9 lists the mAP values of each target model. Our eSLP-GAN also encouraged transferability in object detection models.

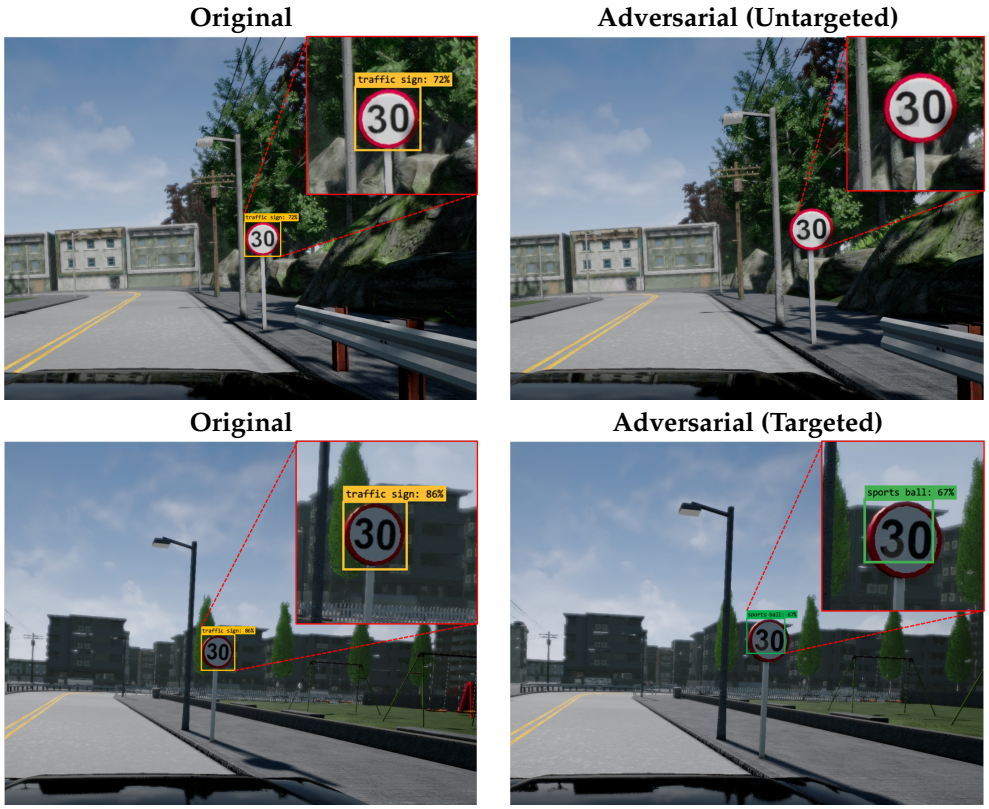

Figure 5. Samples of original images and adversarial examples for traffic signs.

5. Conclusions

In this paper, we proposed an extended spatially localized perturbation GAN (eSLP-GAN) for generating adversarial camouflage patches. We improved the robustness of the real-world attack over the previous version and extended it to attack an object detection model. We extracted the target region to place adversarial patches that could maximize the attack success rate by using the Grad-CAM algorithm. In addition, we generated robust adversarial patches in the real world by combining the EOT algorithm and NPS loss and attacked the object detection models through an appropriate loss function correction for the eSLP-GAN. The experimental results show that eSLP-GAN generated robust adversarial patches that were visually natural with a high attack performance that can be used for both classification and object detection models.

The proposed eSLP-GAN was validated on the custom traffic sign dataset against the given classification models and showed that the attack performance improved compared to the previous version. The proposed method also performed well in a black-box attack because, as can be observed from the experimental results, it encouraged transferability. We also verified the efficacy of attacking object detection models using a photo-realistic simulator.

The proposed methods emphasize the vulnerability of computer vision models that are widely used in the real world. These attacks, which are difficult to detect with human vision, are especially fatal to computer vision systems because they are concealed. Therefore, it is suggested that improvements to the security of computer vision models are essential to withstand attacks, such as by the proposed method.

Author Contributions: Conceptualization, Y.K. and H.K. (Hyoeun Kang); methodology, Y.K.; software, Y.K. and N.S.; validation, H.K. (Hyoeun Kang) and A.M.; formal analysis, Y.K.; investigation, Y.K. and H.K. (Hyoeun Kang); resources, A.M. and N.S.; data curation, Y.K. and H.K. (Hyoeun Kang); writing—original draft preparation, Y.K. and H.K. (Hyoeun Kang); writing—review and editing, H.T.L. and H.K. (Howon Kim); visualization, N.S. and H.T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00706, Development of 3S (Security, Safety, Safeguard) Security Hub Platform for Security Enhancement of Major Facilities and Port Infrastructure)/(No.2019-0-01343, Regional strategic industry convergence security core talent training business).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, F.; Zhao, H.; Ying, W.; Liu, Q.; Raj, A.N.J.; Fu, B. Human face sketch to RGB image with edge optimization and generative adversarial networks. *Intell. Autom. Soft Comput.* **2020**, *26*, 1391–1401. [CrossRef]
2. Lee, Y.H.; Ahn, H.; Ahn, H.B.; Lee, S.Y. Visual object detection and tracking using analytical learning approach of validity level. *Intell. Autom. Soft Comput.* **2019**, *25*, 205–215. [CrossRef]
3. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
4. Tran, L.A.; Le, M.H. Robust U-Net-based Road Lane Markings Detection for Autonomous Driving. In Proceedings of the 2019 International Conference on System Science and Engineering (ICSSE), Dong Hoi, Vietnam, 20–21 July 2019; pp. 62–66. [CrossRef]
5. Shustanov, A.; Yakimov, P. CNN Design for Real-Time Traffic Sign Recognition. *Procedia Eng.* **2017**, *201*, 718–725. [CrossRef]
6. Qayyum, A.; Ahmad, I.; Iftikhar, M.; Mazher, M. Object detection and fuzzy-based classification using UAV data. *Intell. Autom. Soft Comput.* **2020**, *26*, 693–702. [CrossRef]
7. Ge, M.; Bangui, H.; Buhnova, B. Big Data for Internet of Things: A Survey. *Future Gener. Comput. Syst.* **2018**, *87*, 601–614. [CrossRef]
8. Zhang, J.; Zhong, S.; Wang, T.; Chao, H.C.; Wang, J. Blockchain-based systems and applications: A survey. *J. Internet Technol.* **2020**, *21*, 1–14.
9. Al-Wesabi, F.N.; Iskandar, H.G.; Alamgeer, M.; Ghilan, M.M. Proposing a High-Robust Approach for Detecting the Tampering Attacks on English Text Transmitted via Internet. *Intell. Autom. Soft Comput.* **2020**, *26*, 1267–1283. [CrossRef]
10. Gu, Z.; Su, Y.; Liu, C.; Lyu, Y.; Jian, Y.; Li, H.; Cao, Z.; Wang, L. Adversarial Attacks on License Plate Recognition Systems. *CMC-Comput. Mater. Contin.* **2020**, *65*, 1437–1452. [CrossRef]
11. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings.
12. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. 2016. Available online: http://xxx.lanl.gov/abs/1608.04644 (accessed on 18 June 2021).
13. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018; Conference Track Proceedings.
14. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial Patch. *CoRR*. 2017. Available online: http://xxx.lanl.gov/abs/1712.09665 (accessed on 20 June 2021).
15. Liu, A.; Liu, X.; Fan, J.; Ma, Y.; Zhang, A.; Xie, H.; Tao, D. Perceptual-Sensitive GAN for Generating Adversarial Patches. *Proc. Aaai Conf. Artif. Intell.* **2019**, *33*, 1028–1035. [CrossRef]
16. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramèr, F.; Prakash, A.; Kohno, T.; Song, D. Physical Adversarial Examples for Object Detectors. *CoRR*. 2018. Available online: http://xxx.lanl.gov/abs/1807.07769 (accessed on 27 June 2021).
17. Zhao, Y.; Zhu, H.; Shen, Q.; Liang, R.; Chen, K.; Zhang, S. Practical Adversarial Attack against Object Detector. *CoRR*. 2018. Available online: http://xxx.lanl.gov/abs/1812.10217 (accessed on 4 July 2021).
18. Kim, Y.; Kang, H.; Mukaroh, A.; Suryanto, N.; Larasati, H.T.; Kim, H. Spatially Localized Perturbation GAN (SLP-GAN) for Generating Invisible Adversarial Patches. In *Information Security Applications*; You, I., Ed.; Springer International Publishing: Cham, Switzerland, 2020; pp. 3–15.
19. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Why did You Say That? Visual Explanations from Deep Networks via Gradient-based Localization. *CoRR*. 2016. Available online: http://xxx.lanl.gov/abs/1610.02391 (accessed on 30 June 2021).
20. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *CoRR*. 2015. Available online: http://xxx.lanl.gov/abs/1512.03385 (accessed on 5 June 2021).

22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*. 2017. Available online: http://xxx.lanl.gov/abs/1704.04861 (accessed on 9 June 2021).
23. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *CoRR*. 2018. Available online: http://xxx.lanl.gov/abs/1801.04381 (accessed on 2 July 2021).
24. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2020. Available online: http://xxx.lanl.gov/abs/1905.11946 (accessed on 23 June 2021).
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
26. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, NIPS'15, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
28. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
30. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR*. 2020. Available online: http://xxx.lanl.gov/abs/2004.10934 (accessed on 26 June 2021).
31. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
32. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014; Conference Track Proceedings.
33. Moosavi-Dezfooli, S.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *CoRR*. 2015. Available online: http://xxx.lanl.gov/abs/1511.04599 (accessed on 16 June 2021).
34. Chen, P.Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.J. *ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models*; Association for Computing Machinery: New York, NY, USA, 2017.
35. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Black-box Adversarial Attacks with Limited Queries and Information. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholmsmassan, Stockholm, Sweden, 10–15 July 2018.
36. Tu, C.C.; Ting, P.; Chen, P.Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.J.; Cheng, S.M. AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 742–749. [CrossRef]
37. Alzantot, M.; Sharma, Y.; Chakraborty, S.; Zhang, H.; Hsieh, C.J.; Srivastava, M.B. GenAttack: Practical Black-Box Attacks with Gradient-Free Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'19, Prague, Czech Republic, 13–17 July 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1111–1119. [CrossRef]
38. Li, Y.; Li, L.; Wang, L.; Zhang, T.; Gong, B. NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 3866–3876.
39. Suryanto, N.; Kang, H.; Kim, Y.; Yun, Y.; Larasati, H.T.; Kim, H. A Distributed Black-Box Adversarial Attack Based on Multi-Group Particle Swarm Optimization. *Sensors* **2020**, *20*, 7158. [CrossRef] [PubMed]
40. Liu, X.; Yang, H.; Song, L.; Li, H.; Chen, Y. DPatch: Attacking Object Detectors with Adversarial Patches. *CoRR*. 2018. Available online: http://xxx.lanl.gov/abs/1806.02299 (accessed on 16 June 2021).
41. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing Robust Adversarial Examples. 2018. Available online: http://xxx.lanl.gov/abs/1707.07397 (accessed on 4 July 2021).
42. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Models. 2018. Available online: http://xxx.lanl.gov/abs/1707.08945 (accessed on 3 June 2021).
43. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M. *Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1528–1540. [CrossRef]
44. Chen, S.T.; Cornelius, C.; Martin, J.; Chau, D.H.P. ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In *Machine Learning and Knowledge Discovery in Databases*; Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 52–68.
45. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, NIPS'14, Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA, 2014; pp. 2672–2680.
46. Xiao, C.; Li, B.; Zhu, J.; He, W.; Liu, M.; Song, D. Generating Adversarial Examples with Adversarial Networks. *CoRR*. 2018. Available online: http://xxx.lanl.gov/abs/1801.02610 (accessed on 28 June 2021).

47. Zhao, Z.Q.; Zheng, P.; Tao, X.S.; Wu, X. Object Detection with Deep Learning: A Review. 2019. Available online: http://xxx.lanl.gov/abs/1807.05511 (accessed on 28 June 2021).
48. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR*. 2016. Available online: http://xxx.lanl.gov/abs/1611.07004 (accessed on 30 June 2021).
49. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251. [CrossRef]
50. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. Available online: http://xxx.lanl.gov/abs/1505.04597 (accessed on 13 June 2021).
51. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. 2015. Available online: http://xxx.lanl.gov/abs/1405.0312 (accessed on 17 June 2021).
52. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.