Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks

Federico Nesti*, Giulio Rossolini*, Saasha Nair, Alessandro Biondi, and Giorgio Buttazzo
Department of Excellence in Robotics & AI, Scuola Superiore Sant'Anna
<name>.<surname>@santannapisa.it

Abstract

Deep learning and convolutional neural networks allow achieving impressive performance in computer vision tasks, such as object detection and semantic segmentation (SS). However, recent studies have shown evident weaknesses of such models against adversarial perturbations. In a real-world scenario instead, like autonomous driving, more attention should be devoted to real-world adversarial examples (RWAEs), which are physical objects (e.g., billboards and printable patches) optimized to be adversarial to the entire perception pipeline. This paper presents an in-depth evaluation of the robustness of popular SS models by testing the effects of both digital and real-world adversarial patches. These patches are crafted with powerful attacks enriched with a novel loss function. Firstly, an investigation on the Cityscapes dataset is conducted by extending the Expectation Over Transformation (EOT) paradigm to cope with SS. Then, a novel attack optimization, called scene-specific attack, is proposed. Such an attack leverages the CARLA driving simulator to improve the transferability of the proposed EOT-based attack to a real 3D environment. Finally, a printed physical billboard containing an adversarial patch was tested in an outdoor driving scenario to assess the feasibility of the studied attacks in the real world. Exhaustive experiments revealed that the proposed attack formulations outperform previous work to craft both digital and real-world adversarial patches for SS. At the same time, the experimental results showed how these attacks are notably less effective in the real world, hence questioning the practical relevance of adversarial attacks to SS models for autonomous/assisted driving.

1. Introduction

The rise of deep learning unlocked unprecedented performance in several scientific areas [25]. Convolutional neural networks [17] (CNNs) yielded super-human performance for many different computer vision tasks, such as image recognition [10], object detection [28] [27], and image segmentation [21]. Image segmentation, and semantic segmentation (SS) in particular, is used in autonomous driving perception pipelines [31], mainly for object detection [21].

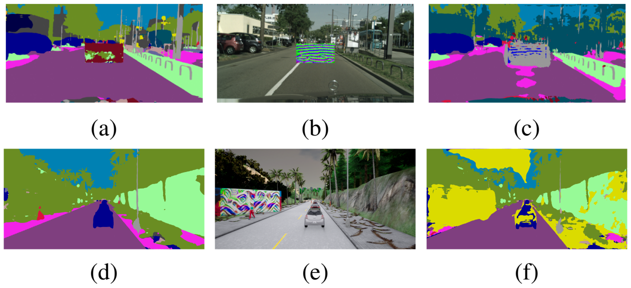


Figure 1: Proposed adversarial patches on Cityscapes [6] (b) and CARLA Simulator [7] (e); (c/f) show the corresponding SS predicted by BiSeNet [41]; (a/d) show the corresponding predictions obtained using random patches instead of adversarial ones.

Despite their high performance, CNNs are prone to adversarial attacks [32]. Most of the literature on adversarial attacks focuses on directly manipulating the pixels of the whole image, hence making the assumption that the attacker has control over the digital representation of the environment obtained by the on-board cameras. This kind of unsafe inputs are called *digital* adversarial examples.

Although such digital attacks do not transfer well into the real world, they continue to be used to evaluate the robustness of models in safety-critical systems [13, 4, 20]. *Real-world* adversarial examples (RWAEs), on the other hand, are physical objects that can be placed in the field of view of a camera, such that the resulting image acts as an adversarial example for the neural network under attack [18]. Thus, RWAEs can induce errors in neural networks without requiring the attacker to access the digital representation of the image, thereby making them a more realistic and dangerous threat to safety-critical systems.

This work focuses on RWAEs, as they represent a potential threat to tasks in autonomous driving today. Although the effects of RWAEs have been studied extensively in the literature for classification and object detection, those on SS

[*] Equal contribution

remain relatively unexplored. However, SS is an integral part of autonomous driving pipelines [31]. Thus, this paper examines various state-of-the-art models for real-time SS aiming at benchmarking their robustness to RWAEs in autonomous driving scenarios.

Of the several types of RWAEs proposed in the literature [33], the form of attack used in this paper is adversarial patches [5]. This is because attacks that perturb the whole image are not practically feasible in the real world. Conversely, such patches can be easily printed and attached to any visible 2D surface in the driving environment, such as billboards and road signs, thus making them a simple, yet effective attack strategy.

The paper starts by recognizing the shortcomings of the standard cross-entropy loss for optimizing adversarial patches for SS. Thus, an extension to the cross-entropy loss is proposed and integrated in all the performed attacks. This extension forces the optimization to focus on pixels that are not yet misclassified, thus obtaining patches that are more powerful compared to those generated with the standard cross-entropy-based setting [22].

Following this rationale, the robustness of real-time SS models to RWAEs attacks is benchmarked. The paper starts by first examining the **case of driving images**, crafting adversarial patches on the Cityscapes dataset [6], a popular benchmark of high-resolution images of urban driving. Robust real-world patches are crafted by following the Expectation Over Transformation (EOT) [2] paradigm, which has been extended in this work to attack SS models. Furthermore, a comparison against non-robust patches (without EOT) is presented to question their effectiveness on driving scenes.

Another set of experiments targeted a **virtual 3D scenario**, for which a stronger adversarial attack is presented and tested. The proposed *scene-specific attack*, defined in Section 3.4, is a more practical tool for crafting adversarial patches in a realistic autonomous driving scenario. It assumes that the attacker is interested in targeting an autonomous driving scene at a particular corner of a specific town, where information about the position of the attackable 2D surface (in our case, a billboard) is available. To satisfy such requirements we developed and tested this attack using the CARLA simulator, which provides all the needed geometric information. These experiments include a comparison with the EOT-based and non-robust patches, performed by importing them into the CARLA world and placing them on billboards to simulate a realistic study.

Figure 1 provides some examples of the effect of our patches on Cityscapes and CARLA.

The last set of experiments were conducted on a **real-world driving scenario**, which required collecting a dataset within the city, optimizing a patch on it, physically printing said patch on a billboard, and finally evaluating SS models on images containing the printed patch.

To the best of our knowledge, this work represents the first exhaustive evaluation of the robustness of SS models against RWAEs for autonomous driving systems. The results of the experiments state important observations that should be taken into consideration while evaluating the trustworthiness of SS models in autonomous driving. First, they demonstrate that non-robust patches are not good candidates for assessing the practical robustness of an SS model to adversarial examples. Indeed, while they proved to be effective in attacking images related to driving scenes (from Cityscapes), they do not induce any real-world adversarial effect when crafted and tested in a virtual 3D world (based on CARLA). Conversely, robust patches, crafted with EOT or the proposed scene-specific approach, resulted to be less effective than non-robust ones on Cityscapes images, but were capable to accomplish the attack in both virtual 3D world and the real world. Nevertheless, their effectiveness in the latter two cases still resulted to be quite limited, hence questioning the practical relevance of RWAEs.

In summary, the paper makes the following contributions:

- It proposes an extension to the pixel-wise cross-entropy loss to enable crafting strong patches for the semantic segmentation setting.
- It proposes a novel technique for crafting adversarial patches for autonomous driving scenarios that utilize geometric information of the 3D world.
- It finally reports an extensive evaluation of RWAE-based attacks on a set of real-time semantic segmentation models using data from the Cityscapes dataset, CARLA, and the real world.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of related work existing in the literature, Section 3 formalizes the proposed loss function, pipeline, and attack strategy, Section 4 reports the experimental results, and Section 5 states the conclusions and proposes ideas for future work.

2. Related Work

Szegedy et al. [34] showed that small well-crafted perturbations when added to the input image were sufficient to fool strong classification networks. [4, 20, 38, 15, 14, 1, 22, 30] have studied such attacks for the specific use case of fooling SS models. However, these attacks directly manipulate the pixels of the image. Although such digital perturbations provide a convenient way to provide benchmarks in research, they do no extend well to real-world applications.

A more realistic threat model led to the introduction of RWAEs by Kurakin et. al. [18]. The attacker here is assumed to be able to craft adversarial pictures in the physical world, without the ability to manipulate the digital representation of inputs to the neural network. However, this work did not account for factors that affect images of objects in the real-world (e.g., varying viewpoints from which input images could be captured, changes in lighting conditions and so on). Athalye et. al. [2] address this issue by introducing the EOT algorithm. EOT accounts for such factors in the optimisation by modeling them as a distribution of transformation functions applied to the adversarial input. These transformations can be in the form of rotation, scaling, noise, brightening and so on. Then, the idea is to optimize the loss function in expectation across the range of selected transformation functions.

The EOT formulation led to the development of *adversarial patches*, introduced by Brown et al. [5] to fool image classifiers. They are robust, localized, image-agnostic perturbations, crafted with the EOT paradigm, capable of fooling neural networks when placed within the input scene or added digitally on images.

Although extensive prior work exists to construct such physical attacks for classification [5, 29, 9], object detection [37, 36, 19, 42], optical flow [26], LiDAR object detection [35], and depth estimation [40], only a few focus on autonomous driving tasks, since testing the adversarial robustness is more challenging, as it requires controlling the 3D outdoor environments. Other works [42, 36] have shown CARLA to be a viable solution in alleviating this issue by crafting and evaluating adversarial situations in virtual 3D environments. This paper also heavily relies on CARLA to evaluate how the optimized adversarial patches translate to a 3D world.

The work closest to ours is the one by Nakka et. al. [22], who attempted to fool a variety of SS models via local attacks (i.e., creating pixel perturbation in a specific area of the image). Despite the attacks being local, the objective of their study was not to evaluate the robustness to real-world attacks, which is instead the main focus of this paper. To the best of our knowledge, such a study is missing in the literature for the case of SS models, which represent essential components in an autonomous driving perception pipeline [31].

Additionally, this paper also improves the loss function used for generating patches. Section 3.5 presents a more general formulation of the cross-entropy loss for the SS setting, designed to optimize more powerful and effective adversarial examples, while all the previously mentioned papers use the standard pixel-wise cross-entropy loss.

3. Attack Formulation

This section presents the design of adversarial patches for semantic segmentation (SS), starting with a short recap of the basic notions behind SS. The patch optimization scheme for both the EOT-based and the scene-specific attacks is then presented. Finally, the proposed loss function is introduced.

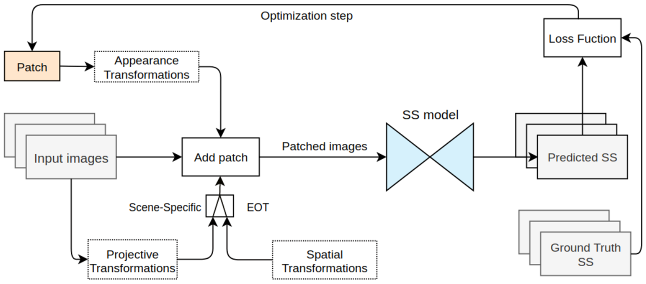


Figure 2: Outline of the proposed approach for crafting both the EOT-based and the scene-specific patches.

3.1. Background on SS

An image with height H and width W can be represented as $x \in [0,1]^{H\times W\times C}$, where C is the number of channels. An SS model returns $f(x) \in [0,1]^{H\times W\times N_c}$, where N_c is the number of classes. This output represents the predicted class-probability scores associated to each image pixel. In particular, $f_i^j(x) \in [0,1]$ denotes the predicted probability score for the i-th pixel of the image corresponding to the class j. Consequently, the predicted semantic segmentation $SS(x) \in \mathbb{N}^{H\times W}$ is computed by extracting those classes with the highest probability score in each pixel: $SS(x) = \text{argmax}_{j\in\{1,...,N_c\}} f_i^j(x)\ , \forall i \in \{1, ..., H \times W\}$.

The ground truth for the SS of x is defined as $y \in \mathbb{N}^{H\times W}$, and assigns the correct class (in $\{1, ..., N_c\}$) to each pixel. The performance of the SS models is evaluated by computing the cross-entropy loss $\mathcal{L}_{CE}(f_i(x), y_i) = -log(f_i^{y_i}(x))$. Thus, for each pixel i, the model's prediction f_i is compared against the ground truth class y_i.

3.2. Patch-based attack pipeline

Both the EOT-based and the scene-specific attacks share a similar pipeline, which is explained in the following paragraph and illustrated in Figure 2.

An adversarial patch of height $\tilde{H}$ and width $\tilde{W}$ is denoted as $\delta \in [0,1]^{\tilde{H}\times\tilde{W}\times C}$, where $\tilde{H} < H$ and $\tilde{W} < W$. This patch is then added to the original image x to obtain a patched image $\tilde{x}$. Thus, the output of the SS model on this patched image would now be $f(\tilde{x})$.

The attacks considered are both untargeted. This means that the objective is to maximize a certain loss function $\mathcal{L}$, without forcing the classification of pixels towards any specific class.

Inspired by the EOT method [2], the idea is to find an optimal patch δ^* (starting from a random patch) that maximizes the loss $\mathcal{L}$ for all the patched images in expectation according to the distribution of transformations used to apply the patch δ on the image set $\mathbf{X}$.

Formally, we need to define:

- A **set of appearance-changing transformations** Γ_a, for instance changes in illumination (brightness, contrast) and noise (uniform or gaussian). These transformations are directly applied to the patch, so obtaining a transformed patch $\zeta_a(\delta)$, where $\zeta_a \in \Gamma_a$. They are used to make the patch robust to illumination changes and acquisition noise.

- A **patch placement function** η that defines which portion of the original image x is occupied by the patch. This is the only part of the pipeline that differs between the two proposed attacks, and is discussed further in the following subsections.
- A **patch application function** $g(x, \zeta_a(\delta), \eta)$ that replaces a certain area of x with $\zeta_a(\delta)$ according to η and returns the patched image $\tilde{x}$.

These functions are sufficient to define both the *EOT-based* attack, which uses randomized spatial transformations to place the patch onto the image, and the *scene-specific* attack, which uses a precise projective transformation to enhance the accuracy of the patch placement.

3.3. EOT-based patch attack

The classic EOT-based attack, as in previous work [5] [39], uses a set of combinations of spatial transformations Γ, including translation and scaling, from which the patch placement function η is selected.

The parameters for each transformation are randomized within a pre-defined range. Section 4 provides a more detailed explanation of the set of transformations used. The optimal patch is then defined as

$$\delta^* = \underset{\delta}{\operatorname{argmax}}\ \mathbb{E}_{x\in\mathbf{X},\zeta_a\in\mathbf{\Gamma_a},\eta\in\mathbf{\Gamma}}\ \mathcal{L}(f(\tilde{x}), y) \tag{1}$$

In practice, the optimal patch is computed via an iterative optimization process. At each iteration t, the pixels values of the patch are modified in the direction of the gradient of the loss function computed with respect to the patch:

$$\delta_{t+1} = \text{clip}_{[0,1]}\left(\delta_t + \epsilon \cdot \sum_{x\in\mathbf{X}} \sum_{\substack{\zeta_a\in\mathbf{\Gamma_a}\\ \eta\in\mathbf{\Gamma}}} \nabla_{\delta_t}\mathcal{L}(f(\tilde{x}), y)\right), \tag{2}$$

where ϵ represents the step size.

$\mathcal{L}$ consists of a weighted sum of multiple loss functions. The adversarial patch effect is obtained through the optimization of the adversarial loss $\mathcal{L}_{adv}$ (discussed further in subsection 3.5). Additionally, to ensure that the patch transfers well to the real world, two losses are added to account for the *physical realizability* of the patch (see [3] for details): smoothness loss $\mathcal{L}_S$ and non-printability score $\mathcal{L}_N$.

3.4. Scene-specific patch attack

To provide a more realistic approach for autonomous driving environments, this work proposes an alternative attack methodology that exploits the geometrical information provided by the CARLA Simulator [7].

Here, the key assumption is the availability of an attackable 2D surface, e.g., a billboard, with a fixed location in close proximity to the road. The CARLA simulator features the possibility to extract camera extrinsic and intrinsic matrices (details in [3]), and the pose of the attackable surface. The billboard-to-image transformations can be computed using a 3D roto-translation composition, which allows the patch to be warped accordingly, thus obtaining higher precision in applying the patch to the attack surface.

This attack uses the same optimization pipeline as before, with one major difference: instead of placing the patch randomly, as in the previous attack, correct projective transformations are used to determine the placement of the patch on the attackable surface. Hence, η is no longer randomized, but is computed for each image in the dataset.

This method allows crafting precise attacks that are optimised for the region of the town that the attacker is interested in. The attacker would need to collect several images, from different viewpoints, of the desired attackable surface, along with the corresponding intrinsic and extrinsic matrices. This approach to image collection also implies that EOT is no longer needed for patch placement, thereby simplifying the optimization process.

The downside of this approach is that a digital representation of the target scene is required to accurately capture the required matrices. Although CARLA provides the possibility to import cities via OpenStreetMaps (`https://www.openstreetmap.org/`), it requires some amount of manual effort to properly model 3D meshes to include objects in this simulated world. These objects need to be properly designed to ensure that the patches optimised in simulation transfer well to the real-world. Although this paper does not investigate CARLA-to-real-world transfer issues, future work will address this problem to improve the proposed methodology and adapt it for real-world attacks. Section 4 provides a comparison of this method against the EOT-based attack.

3.5. Proposed loss function

Cross-entropy (CE) is a popular choice as adversarial loss. Pixel-wise CE has been shown to work well when crafting an untargeted digital attack (i.e., by directly adding a perturbation r to the pixels of a digital image) [22] [4]. This is formulated as: $\mathcal{L}_{adv}(f(x+r), y) = \frac{1}{|\mathcal{N}|} \cdot \sum_{i\in\mathcal{N}} \mathcal{L}_{CE}(f_i(x+r), y_i)$, where $\mathcal{N} = \{1, ..., H \times W\}$ denotes the whole set of pixels in $\tilde{x}$. However, modifications can be introduced to this formulation to allow crafting stronger attacks for fooling SS models.

Following previous notation, let $\tilde{\mathcal{N}} = \{1, ..., \tilde{H} \times \tilde{W}\} \subseteq \mathcal{N}$ denote only the pixels that correspond to the patch δ. Then, $\mathbf{\Upsilon}$ defines a subset of image pixels that do not belong to the patch and are still predicted correctly by the model with respect to the trusted ground truth label y:

$$\boldsymbol{\Upsilon} = \{i \in \mathcal{N} \setminus \tilde{\mathcal{N}} \quad | \quad SS_i(\tilde{x}) = y_i\}. \tag{3}$$

Using $\boldsymbol{\Upsilon}$, the previous pixel-wise CE loss computed on $\mathcal{N} \setminus \tilde{\mathcal{N}}$ can be split into two distinct terms:

$$\mathcal{L}_M^{\tilde{x}} = \sum_{i \in \boldsymbol{\Upsilon}} \mathcal{L}_{CE}(f_i(\tilde{x}), y_i), \quad \mathcal{L}_{\overline{M}}^{\tilde{x}} = \sum_{i \notin \boldsymbol{\Upsilon}} \mathcal{L}_{CE}(f_i(\tilde{x}), y_i) \; . \tag{4}$$

$\mathcal{L}_M^{\tilde{x}}$ describes the cumulative CE for those pixels that have been misclassified with respect to the ground truth y, while $\mathcal{L}_{\overline{M}}^{\tilde{x}}$ refers to all the others.

Note that both $\mathcal{L}_M^{\tilde{x}}$ and $\mathcal{L}_{\overline{M}}^{\tilde{x}}$ do not consider pixels of the patch, which have been discarded to focus the optimization on attacking portions of the image away from the patch. By computing these separate contribution to the total loss, we avoid that the contribution of the non-misclassified pixels gets obscured by the other term, which is a problem we found during preliminary tests. Hence, the adversarial loss function gradient is redefined as follows:

$$\nabla_\delta \mathcal{L}(f(\tilde{x}), y) = \gamma \cdot \frac{\nabla_\delta \mathcal{L}_M^{\tilde{x}}}{||\nabla_\delta \mathcal{L}_M^{\tilde{x}}||_2} + (1 - \gamma) \cdot \frac{\nabla_\delta \mathcal{L}_{\overline{M}}^{\tilde{x}}}{||\nabla_\delta \mathcal{L}_{\overline{M}}^{\tilde{x}})||_2} \; , \tag{5}$$

where $\gamma \in [0, 1]$ is a parameter that determines whether the optimization should focus on decreasing the number of correctly classified pixels or improving the adversarial strength for the currently misclassified pixels. The rationale of γ is to provide an empirical balancing between the importance of $\mathcal{L}_M$ and $\mathcal{L}_{\overline{M}}$ at each iteration t depending on the number of pixels not yet misclassified.

Moreover, an adaptive value of $\gamma = \frac{|\Upsilon|}{|\mathcal{N} \setminus \tilde{\mathcal{N}}|}$ has been proposed to provide an automatic tuning of γ at each iteration. The idea is to initially focus on boosting the number of misclassified points. Over time, as this number increases, the focus of the loss function gradually shifts toward improving the adversarial strength of the patch on these wrongly classified pixels.

Section 4 provides an extensive analysis of the proposed loss function by comparing multiple values of γ with the standard pixel wise CE measured both on $\mathcal{N} \setminus \tilde{\mathcal{N}}$ and $\mathcal{N}$ (which is used by [22]), suggesting that our formulation is indeed more general and effective for this kind of attack.

4. Experimental results

This section describes the experimental setup and the results achieved with the proposed attacks. First, the proposed loss function is evaluated for different values of γ comparing its effectiveness against the standard pixel-wise CE, showing that it is a better alternative for this kind of problems. Following this, the results of patches crafted with and without EOT are presented on the Cityscapes dataset.

Subsequently, the results obtained with the scene-specific attack on three CARLA-generated datasets are presented. These results are compared against the EOT-based attack to show the improved effectiveness of this formulation. Finally, some preliminary results of real-world adversarial patches are presented. A more detailed analysis of all models tested against these attacks can be found in the supplementary material [3].

4.1. Experimental setup

The experiments were performed on a set of 8 NVIDIA-A100 GPUs, while the CARLA simulator was run on a system powered by an Intel Core i7 with 12GB RAM and a GeForce GTX 1080 Ti GPU.

All experiments were performed in PyTorch [24]. The optimizer of choice was Adam [16], with learning rate set to 0.5 empirically. The effect of the adversarial patches on the SS models was evaluated using the mean Intersection-over-Union (mIoU) and mean Accuracy (mAcc) [21].

The code is available at this link: `https://github.com/retis-ai/SemSegAdvPatch`.

Datasets The experiments in the case of driving images were carried out using the *Cityscapes* dataset [6], a popular benchmark for urban scene SS. The dataset consists of 2975 and 500 high resolution images (1024×2048) for training and validation, respectively. The experiments reported in this paper were conducted on 250 images randomly sampled from the training set. Conversely, the entire validation set was used to evaluate the effectiveness of the patches.

The CARLA simulator was used to provide a 3D virtual scenario. This set of experiments was performed in Town01, one of the built-in maps provided with the simulator, with 'CloudyNoon' as the preset weather. To mimic the settings of the Cityscapes dataset, RGB images of size 1024×2048, along with their corresponding SS tags, were collected by placing a camera on-board the ego vehicle.

The SS models trained on Cityscapes had to be fine-tuned to ensure good performance on CARLA images. 600 images for fine-tuning, 100 for validation, and 100 for testing were collected by spawning the ego vehicle at random positions in Town01. Additional details about the fine-tuning can be found in the supplementary material [3].

To study the effectiveness of the patches in CARLA, the map of Town01 was manually edited to include three billboards. Thus, without-EOT, EOT-based, and the scene-specific attacks were carried out at three different locations within Town01. The datasets for the attacks were collected by spawning the ego vehicle at random locations within the proximity of each billboard to emulate varying viewpoints from which the patch might be captured in the real-world. For each of the three billboards, 150 training images, 100 validation, and 100 test images were gathered. Details about the position and orientation of the billboard and the camera were stored to compute the roto-translations used in the scene-specific attack.

model	mIoU / mAcc	
	cityscapes	CARLA (val - scene1 - scene2 -scene3)
ICNet	0.78 / 0.85	0.70 / 0.84 - 0.53 / 0.70 - 0.64 / 0.74 - 0.62 / 0.74
BiSeNet	0.69 / 0.78	0.47 / 0.69 - 0.47 / 0.69 - 0.61 / 0.74 - 0.47 / 0.73
DDRNet	0.78 / 0.85	0.72 / 0.88 - 0.54 / 0.74 - 0.62 / 0.76 - 0.64 / 0.78
PSPNet	0.79 / 0.85	nd

Table 1: mIoU and mAcc of the tested models on Cityscapes (pre-trained) and our CARLA dataset (fine-tuned).

Lastly, to study the effects of adversarial patches in the real world, an additional dataset of 1000 images, hereafter referred to as *Patches-scapes*, was collected by mounting an action camera on the dashboard of a vehicle using a setup similar that of the Cityscapes dataset, and then driving the vehicle within the streets of our city.

Models The attacks studied in this paper were evaluated using DDRNet [12], BiSeNet [41], and ICNet [43], which represent the state-of-the-art in real-time SS, making them preferable for the use case of autonomous driving. Additionally, PSPNet [44] was included in the study for the EOT-based attack on the CityScapes dataset, but not for the scene-specific attack on CARLA, since we are interested in real-time performance.

All the models were loaded with the pre-trained weights provided by the authors (further specifications are provided in the supplementary material [3]). Table 1 summarizes the performance of these models on both the Cityscapes and CARLA validation sets.

4.2. EOT-based patches on Cityscapes

The Cityscapes dataset is used to optimize two types of patches on the same training images, one with EOT and the other without EOT (non-robust). Three different patch sizes are studied: 150×300, 200×400, and 300×600 pixels.

The non-robust patches (without EOT) were optimized by placing them at the center of the image at each training iteration (i.e., $\eta(\cdot)$ = fixed position) and applying no appearance transformations (i.e., $\Gamma_a = \emptyset$).

Conversely, the robust optimizations with EOT apply multiple digital transformations. Γ_a includes only Gaussian noise with standard deviation 5% of the image range. Γ includes random scaling ($80\% - 120\%$ of the initial patch size) and random translation defined as follows: if (c_x, c_y) is the center of the image, the position of the patch is randomized within the range $(c_x \pm \tilde{r} \cdot \tilde{W}/2\ ,\ c_y \pm \tilde{r} \cdot \tilde{H}/2)$, where $\tilde{r} \in [0, 1]$ is a uniform random value. The translation range was kept limited, rather than considering the full image space, to ensure a greater stability and faster convergence. The patches were optimized over 200 epochs.

Loss functions analysis Figure 3 reports the mIoU obtained by training patches with the pixel-wise CE computed on $\mathcal{N}$ (used by [22]) and $\mathcal{N} \setminus \tilde{\mathcal{N}}$ compared to the extended CE loss proposed in this paper, with $\gamma \in \{0.5, 0.6, 0.7, 0.8, 0.95, 1.0, \frac{|\Upsilon|}{|\mathcal{N} \setminus \tilde{\mathcal{N}}|}\}$. Among the models evaluated in the paper, ICNet [43] appears to be most robust on the Cityscapes dataset. Thus, the loss functions are studied by optimizing a 200×400 patch with and without EOT on ICNet.

For all the tested values of γ, our formulation converges to a higher attack effect (i.e., smaller mIoU) with lesser number of epochs than the one based on the pixel-wise CE. Experiments without EOT show that all the compared implementations converge after only 10 epochs. In the EOT case, the advantages are even more evident: our proposed formulation converges at almost 25 epochs, while the CE cases still reduce slowly at 200 epochs (nearly 6 hours of optimization time).

The same study was performed for the scene-specific attack in the CARLA virtual world, and produced similar results, reported in the supplementary material [3].

Adversarial patch effects. Table 2 reports how varying the patch size affects each of the SS models. We used the adaptive γ (i.e., $\gamma = \frac{|\Upsilon|}{|\mathcal{N} \setminus \tilde{\mathcal{N}}|}$) that has shown the best overall effect among multiple experimental tests.

Figure 4 illustrates the effects of the optimized patches on the BiSeNet model. As expected, the non-robust patches (without EOT) obtain better attack performance with respect to the ones optimized with EOT. This is because the optimization process is simpler when not considering the randomized transformations. However, it is important to note that these patches would not be transferrable to the real world, and are not robust even to simple transformations [11, 23].

4.3. Scene-specific patches on CARLA

The scene-specific attack was performed on the same set of models as defined earlier. Each of these models were first fine-tuned on images generated via CARLA. The performance of these fine-tuned models on the CARLA datasets is summarized in Table 1. Please note that the mIoU score is computed as an average of the per-class IoU scores, which, for CARLA, might get to 0 for some classes due to the presence of a few pixels belonging to non-common objects.

As described in Section 3.4, the patch is optimized to be adversarial for a specific urban scene by reprojecting it on the attackable 2D surface, which, in this work, is a billboard placed in three different spots in the Town01 map of CARLA. This section reports the effect of the scene-specific attack compared against the non-robust (without-EOT) and the EOT-based attacks. The optimized patch is composed of 150×300 pixels, imposing a real-world dimension of $3.75m \times 7.5m$. Additional experiments on the effect of the real-world dimension of the patch and the number of pixels used are presented in the supplementary material [3]. For all

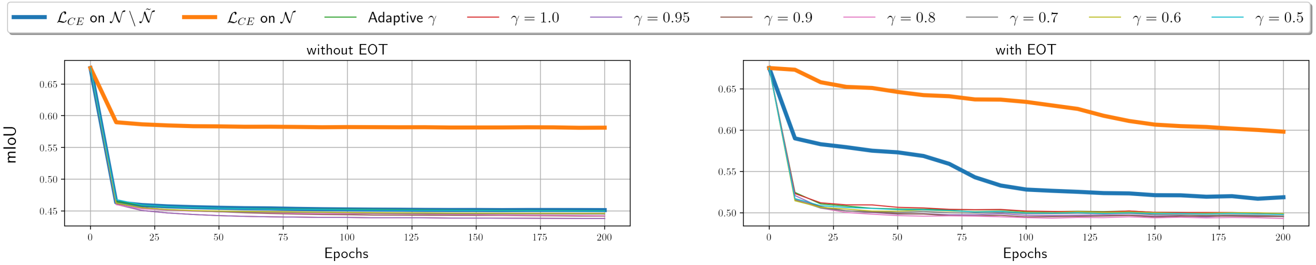
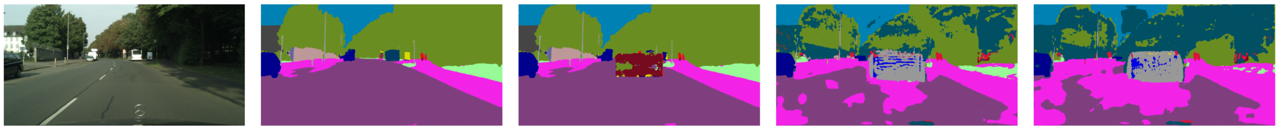


Figure 3: Comparison of adversarial patch optimizations (200 × 400) on ICNet and Cityscapes using different loss functions: two versions of the standard pixel-wise cross-entropy and our formulation with multiple values of γ. $\mathcal{L}_{CE}$ on $\mathcal{N}$ is the original version used by [22], while $\mathcal{L}_{CE}$ on $\mathcal{N} \setminus \tilde{\mathcal{N}}$ is an improved version based on the rationale presented in Section 3.

Model	mIoU — mAcc (rand / with EOT / without EOT)					
	150x300		200x400		300x600	
ICNet	0.70 / 0.58 / 0.54	0.81 / 0.69 / 0.65	0.67 / 0.50 / 0.45	0.79 / 0.61 / 0.55	0.60 / 0.38 / 0.28	0.72 / 0.42 / 0.34
BiSeNet	0.63 / 0.45 / 0.39	0.74 / 0.61 / 0.55	0.61 / 0.29 / 0.21	0.71 / 0.43 / 0.34	0.54 / 0.19 / 0.05	0.65 / 0.31 / 0.15
DDRNet	0.73 / 0.65 /0.55	0.82 / 0.76 / 0.64	0.71 / 0.59 / 0.42	0.80 / 0.69 / 0.50	0.65 / 0.45 / 0.09	0.73 / 0.53 / 0.19
PSPNet	0.76 / 0.42 / 0.33	0.82 / 0.57 / 0.45	0.73 / 0.23 / 0.00	0.79 / 0.30 / 0.05	0.67 / 0.01 / 0.00	0.73 / 0.06 / 0.05

Table 2: Adversarial patch results on the Cityscapes dataset. Each cell reports the final mIoU obtained with a random patch (no optimization), with EOT, and without EOT.



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

Figure 4: Semantic segmentations obtained from BiSeNet with no patch (b), a random patch (c), an EOT-based patch (d), and non-robust patch (without EOT) (e) added into a original image (a) of the Cityscapes validation set.

the following experiments, Γ_a includes contrast and brightness changes (both 10% of the image range), and Gaussian noise (standard deviation 10% of the image range).

Since the objective of this work is to craft RWAEs, the performance of the attacks is evaluated by measuring the mIoU and mAcc scores on additional scene-specific datasets. These additional datasets are produced by collecting the same images of the validation set of each scene-specific dataset, but with a single major modification: the billboard object is modified in the *Unreal Editor* [8] by applying the optimized patch as a *decal* object, which is a way to stick an image on a surface in the virtual environment. This method should provide a simulated real-world application of the patches, since they are no more applied directly on the image, but the image itself includes the patched billboard.

Table 3 summarizes the results obtained on these three additional scene-specific datasets, with a random patch, a non-robust patch, an EOT-based patch, and a scene-specific patch. Figure 5 shows a comparison of all the discussed attacks on DDRNet.

For almost all the combinations of scene and network, the scene-specific attack outperforms the EOT-based attack, confirming that the scene-specific attack, for this kind of problems, is a better alternative to the EOT formulation for the placement of the patch within the image. The only case where the the two attacks show comparable performance is for scene 3, where the billboard is almost perpendicular to the camera plane, allowing the EOT method to cover realistic patch placing functions.

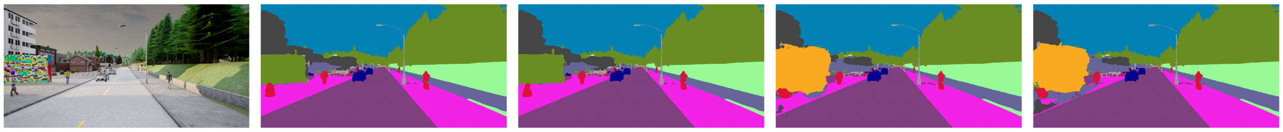
It is also worth noting that SS models are rather robust to adversarial patch attacks in general. Although it is possible to craft adversarial patches that cause a section of the image to be wrongly segmented, it tends to be more difficult than fooling models for tasks such as classification. Additional details are reported in the supplementary material [3].

4.4. Real-world patches

In order to prove that the proposed pipeline can be used for a real-world attack, we use the *Patches-scapes* dataset (described in Section 4.1) to craft an adversarial real-world patch using the EOT-based patch attack. This section presents the results of an attack in Figure 6. Although the presence of the optimized patch does alter a significant area of the predicted SS (while the random patch does not), portions of the image far from its position are not

Model	mIoU — mAcc (rand / without EOT / EOT / scene-specific)					
	Scene1		Scene2		Scene3	
ICNet	0.51 / 0.51 / 0.49 / 0.48	0.60 / 0.60 /0.56 / 0.54	0.64 / 0.64 / 0.61 / 0.61	0.74 / 0.74 / 0.73 / 0.73	0.63 / 0.63 / 0.59 / 0.59	0.76 / 0.76 / 0.73 / 0.74
BiSeNet	0.44 / 0.42 / 0.36 / 0.31	0.63 / 0.61 / 0.55 / 0.49	0.60 / 0.60 / 0.58 / 0.58	0.76 / 0.75 / 0.74 / 0.74	0.47 / 0.46 / 0.46 / 0.45	0.74 / 0.73 / 0.73 / 0.73
DDRNet	0.51 / 0.50 / 0.46 / 0.46	0.70 / 0.69 / 0.69 / 0.69	0.62 / 0.62 / 0.52 / 0.49	0.76 / 0.75 / 0.71 / 0.66	0.65 / 0.65 / 0.58 / 0.59	0.78 / 0.78 / 0.76 / 0.76

Table 3: Adversarial patch results on the three scene CARLA datasets. The Table reports the mIoU and mAcc obtained with random, non-robust (without EOT), EOT-based and scene-specific patches.



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

Figure 5: (a) is an image extracted from the scene-1 test dataset augmented with a scene-specific patch optimized on DDRNet, while (e) is its corresponding SS; (b), (c), and (d) are predictions obtained by augmenting the same test image with a random, non-robust and EOT-based patches, respectively.

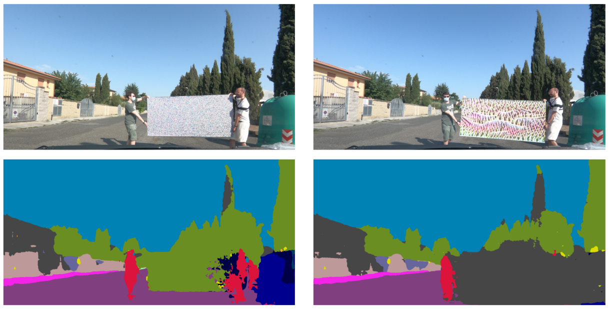


Figure 6: Real-world predictions on ICNet obtained with a printed random patch (left) and an optimized patch (right).

affected. Furthermore, the attack performance decreases as we move the patch away from the camera (details provided in the supplementary material [3]). The patch is optimized for 200 epochs on the pre-trained version of ICnet (since it showed good performance on the Patches-scapes dataset) and printed as a $1m \times 2m$ poster.

Testing adversarial patches for autonomous driving in the real world poses a series of difficulties that heavily limited the tests. First, it is not easy to find a urban corner with good prediction accuracy, and which is not crowded with moving vehicles (which might be dangerous). Second, the patch must be printed in the highest resolution possible on a large rigid surface, which might get expensive. Furthermore, since weather conditions are not controllable and change throughout the day, results can diverge from what is expected.

The scene-specific attack, which requires additional geometric information, could not be implemented at the time of writing, but will be considered in future work.

5. Conclusions and future work

This paper presented an extensive study of the adversarial robustness of semantic segmentation models. This was accomplished by extensively evaluating the effect introduced by adversarial patches, to investigate the limits of real-world attacks for segmentation neural networks in an autonomous driving scenario. Carrying out the investigations with increasingly "real-world" benchmarks, we studied the effect of non-robust and EOT-based patches on the Cityscapes dataset, on a virtual 3D scenario, and in a real-world setting. We also introduced a new method called *scene-specific attack*, which improves the EOT formulation for a more realistic and effective patch placement.

The novel loss function introduced in the paper enabled to advance the state-of-the-art for adversarial patches optimization methods, as it proved to be a more general and efficient alternative to the classic cross-entropy function for this kind of problems.

This exhaustive set of experiments practically opens to a new point of view for studying SS models in autonomous driving. Although the proposed attacks were able to reduce the baseline model accuracy, the SS models proved to be somehow robust to real-world patch-based attacks. This was especially noticeable when the tests were performed in more realistic settings using CARLA and the real world, where, in most cases, the patch only affected the proximity of the attacked surface.

Nevertheless, this is a promising result, since it shows how the prediction provided by these models is not easily corruptible, especially in real-world scenarios. This is in contrast with previous work on patch-based adversarial attacks against classification and object detection models.

Future work will further investigate the robustness properties of these models, introducing defense mechanisms and trying to enhance the robustness of SS model by adding a temporal dimension.

References

[1] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 888–897, 2018.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR, 10–15 Jul 2018.

[3] Authors. Additional material for wacv2022 submission id161. *Submitted as WACV2022 Additional Material*, 2021.

[4] Andreas Bär, Jonas Löhdefink, Nikhil Kapoor, Serin Varghese, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: Enhancing extensive environment sensing. *IEEE Signal Process. Mag.*, 38(1):42–52, 2021.

[5] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial Patch. *arXiv:1712.09665 [cs]*, May 2018. arXiv: 1712.09665.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016.

[7] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 2017.

[8] Epic Games. Unreal engine.

[9] 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 *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1625–1634. IEEE Computer Society, 2018.

[10] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognit.*, 77:354–377, 2018.

[11] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering Adversarial Images using Input Transformations. *arXiv e-prints*, page arXiv:1711.00117, Oct. 2017.

[12] Yuanduo Hong, Huihui Pan, Weichao Sun, Senior Member, IEEE, and Yisong Jia. Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes. *arXiv e-prints*, page arXiv:2101.06085, Jan. 2021.

[13] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

[14] Christoph Kamann and Carsten Rother. Benchmarking the Robustness of Semantic Segmentation Models. *arXiv e-prints*, page arXiv:1908.05005, Aug. 2019.

[15] Xu Kang, Bin Song, Xiaojiang Du, and Mohsen Guizani. Adversarial attacks for image segmentation on multiple lightweight models. *IEEE Access*, 8:31359–31370, 2020.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec. 2014.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[18] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[19] Mark Lee and Zico Kolter. On Physical Adversarial Patches for Object Detection. *arXiv:1906.11897 [cs, stat]*, June 2019. arXiv: 1906.11897.

[20] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2774–2783. IEEE Computer Society, 2017.

[21] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image Segmentation Using Deep Learning: A Survey. *arXiv:2001.05566 [cs]*, Nov. 2020. arXiv: 2001.05566.

[22] Krishna Kanth Nakka and Mathieu Salzmann. Indirect local attacks for context-aware semantic segmentation networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 611–628. Springer, 2020.

[23] Federico Nesti, Alessandro Biondi, and Giorgio Buttazzo. Detecting adversarial examples by input transformations, defense perturbations, and voting. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2021.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Infor-*

mation Processing Systems, volume 32. Curran Associates, Inc., 2019.

[25] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, 51(5), Sept. 2018.

[26] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. Attacking Optical Flow. *arXiv e-prints*, page arXiv:1910.10053, Oct. 2019.

[27] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016.

[28] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[29] 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, Vienna Austria, Oct. 2016. ACM.

[30] Guangyu Shen, Chengzhi Mao, Junfeng Yang, and Baishakhi Ray. Advspade: Realistic unrestricted attacks for semantic segmentation. *arXiv preprint arXiv:1910.02354*, 2019.

[31] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 587–597, 2018.

[32] Samuel Henrique Silva and Peyman Najafirad. Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey. *arXiv:2007.00753 [cs, stat]*, July 2020. arXiv: 2007.00753.

[33] Lu Sun, Mingtian Tan, and Zhe Zhou. A survey of practical adversarial example attacks. *Cybersecurity*, 1(1):9, Dec. 2018.

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[35] J. Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Binh Yang, Richard Du, Frank Cheng, and R. Urtasun. Physically realizable adversarial examples for lidar object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13713–13722, 2020.

[36] Tong Wu, Xuefei Ning, Wenshuo Li, Ranran Huang, Huazhong Yang, and Yu Wang. Physical adversarial attack on vehicle detector in the carla simulator. *CoRR*, abs/2007.16118, 2020.

[37] Zuxuan Wu, Ser-Nam Lim, Larry S. Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*, volume 12349 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2020.

[38] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan L. Yuille. Adversarial examples for semantic segmentation and object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1378–1387. IEEE Computer Society, 2017.

[39] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 665–681. Springer, 2020.

[40] Koichiro Yamanaka, Ryutaroh Matsumoto, Keita Takahashi, and Toshiaki Fujii. Adversarial Patch Attacks on Monocular Depth Estimation Networks. *arXiv e-prints*, page arXiv:2010.03072, Oct. 2020.

[41] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. *arXiv e-prints*, page arXiv:1808.00897, Aug. 2018.

[42] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. CAMOU: LEARNING A VEHICLE CAMOUFLAGE FOR PHYSICAL ADVERSARIAL ATTACK ON OBJECT DETECTORS IN THE WILD. page 20, 2019.

[43] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. *arXiv e-prints*, page arXiv:1704.08545, Apr. 2017.

[44] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. *arXiv e-prints*, page arXiv:1612.01105, Dec. 2016.