Supplementary material for the paper "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

This technical report constitutes the supplementary material for the WACV2022 submission ID 775. The report is structured as follows: Section A describes the models used for this work; Section B presents the additional loss functions used for physical realizability; Section C details the projective transformation used in the scene-specific attack; Section D discusses a few issues that arose with the finetuning of the models on the CARLA dataset; and Section E presents additional results and details on the proposed attacks.

A. Semantic Segmentation Models inolved

To provide fair experiments we used pretrained models published by the authors. Each network structure was imported into our repository keeping the original image normalization parameters required by each the pretrained model. The first three models are designed for real-time semantic segmentation, while the last one is more expensive in terms of memory footprint and execution time.

DDRNet We used the DDRNet23Slim version provided by the author [6][1] that has shown to be one of fastest network in the state-of-the-art of run-time SS.

BiSeNet We used the original Pytorch implementation [11][2] with its pretrained weights. The applied version uses an Xception39 model [3] as backbone.

ICNet The original pretrained model[3] is provided by the authors [12]. It was trained using the Caffe framework [7]. We imported it as a PyTorch model.

PSPNet Also this model was imported from the original Caffe pretrained version provided by [13][4] that uses a RESNet101 [5] as backbone for feature extraction. Since PSPNet is not a real-time semantic segmentation model, we studied its robustness against our patch-based attacks only for the EOT-based attack on the Cityscapes dataset.

B. Loss functions for physical realizability

Classical adversarial examples (sometimes referred to "digital" adversarial examples) rely on image-sized additive perturbations r, that require the optimization of a single loss function $\mathcal{L}_{adv}$ (the adversarial loss). This loss function expresses, in

*equal contribution
[1]https://github.com/ydhongHIT/DDRNet
[2]https://github.com/ycszen/TorchSeg
[3]https://github.com/hszhao/ICNet
[4]https://github.com/hszhao/PSPNet

the untargeted attack formulation, the distance of the network prediction corresponding to the adversarial image $x + r$ from the true label corresponding to the input image x.

Although this loss function is sufficient to optimize digital adversarial examples, which rely on the assumption that the attacker has full control on the digital representation of the image. As discussed in the main paper, this is difficult to obtain for safety-critical systems. Hence, adversarial examples should lay in the physical world and still maintain their fooling power when they enter the field of view of the camera providing the input image for the perception system. Also, usually, a real-world perception system is composed of several different modules, including preprocessing functions, such as crop, resize, and other data augmentations techniques.

These factors explain the need for a more specialized adversarial loss function, which is based on universal perturbations [8] (to produce image-agnostic perturbations), and the EOT paradigm [1] (to render the perturbation robust to transformations typical of the real world), as explained in the main manuscript [2] and referred to as $\mathcal{L}_a dv$.

However, physical realizability is another important factor that has to be accounted for. In particular, it is likely that the printer used to print the adversarial perturbation is not able to print the entire continuous spectrum of colors. Hence, the non-printability score [10] is introduced to take into account this effect. Assuming that a pixel p of the patch δ is composed of an RGB triplet, the non-printability score $\mathcal{L}_N$ is defined as

$$\mathcal{L}_N(p) = \prod_{c \in C} \|p - c\| \tag{1}$$

where C is the set of RGB triplets that compose the printable color palette of the printer. The non-printability score is then averaged for the totality of pixels in the perturbation, and should be low when the totality of the patch pixel colors is close to the printable ones.

Also, typical digital perturbations are very noisy, since there is very little correlation between adjacent pixels. These patterns in the real world are smoothed out by both the printing process, and the camera acquisition process (which introduces noise and blurring). This effect can be taken into account by considering another additional loss, the "smoothness" loss, defined as

$$\mathcal{L}_S = \sum_i \sum_j [(\delta_{i+1,j} - \delta_{i,j})^2 + (\delta_{i,j+1} - \delta_{i,j})^2] \tag{2}$$

where (i, j) are the 2D index of the patch. This additional loss function introduces correlation between adjacent pixels, and should help craft a more "smooth" perturbation, without noisy patterns.

Summing these additional loss terms, the total loss function for physically-realizable real-world adversarial examples is

$$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_N\mathcal{L}_N + w_S\mathcal{L}_S \tag{3}$$

where w_{adv}, w_N, w_S are the weights corresponding to each loss component.

In practice, the adversarial loss (for the untargeted attack formulation) tends to increase in norm during the optimization. This increase masks the effect of the other losses during the advancement of the optimization. To make the weighting actually effective, the gradient of each loss is computed individually, it is normalized, and then averaged according to the weights. This total gradient is applied to advance the optimization, as in Equation 2 in the main paper [2].

The experiments in Section 4 of [2] are performed with $w_{adv} = 1, w_N = 0, w_S = 0.1$. Empirically, we noticed that the non-printability score is not strictly needed for this kind of evaluation, while the smoothness loss is crucial for transferring to the real world.

C. Projective transformation for scene-specific attack

The scene-specific attack, as explained in Section 3 of [2], requires a projective transformation to precisely apply the patch onto the 2D attackable surface.

With the CARLA simulator, it is possible to extract the pose (with respect to the World reference frame) of both the attackable surface and the camera.

Using 3D rototranslations in homogeneous formulation [4], it is possible to express the position of a 3D point on the attackable surface p^S (initially expressed in the Surface reference frame) in the Camera reference frame. Given that T_a^b is the roto-translation matrix from generic reference frame a to b, expressed as composition of a rotation matrix R_a^b and translation vector t_a^b,

$$T_a^b = \begin{bmatrix} R_a^b & t_a^b \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{4}$$

the expression for a point on the attackable surface in the camera reference frame is

$$p^C = T_W^C T_S^W p^S \tag{5}$$

where T_W^C is the World-to-Camera rototranslation, and T_S^W is the Surface-to-World rototranslation.

The resulting point can be projected with a pinhole camera model, which is the one used by CARLA RGB camera sensor. The projection, which is expressed in pixels, and identify the position of the point on the image, can be obtained as

$$p_{pixels} = K p^C \tag{6}$$

where K is the intrinsic matrix of the CARLA RGB camera sensor. Its general form is

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where f is the focal, and (c_x, c_y) is the center of the image expressed in pixels (and still in homogeneous coordinate). For the CARLA RGB camera sensor, these parameters are easy to compute: the focal $f = \frac{w}{\tan(FOV/2)}$ (where FOV is the field of view in radians - $\pi/2$ by default - and w the image width), and the center of the image is simply $(w/2, h/2)$ (where h is the image height) expressed in pixels.

Then, to apply the patch, it is sufficient to compute a warping transformation, which maps the patch starting corners to the destination points on the image, computed as described above. The estimation of the warping transformation is done with the PyTorch-based library Kornia [9].

The transformation is then applied to the patch, which is correctly applied to the attackable surface.

D. CARLA environmental settings and finetuning

The fine-tuning of the models on the CARLA dataset was not an easy task: the images generated with CARLA sample a different distribution than Cityscapes. This justifies the limited performance on CARLA images of state-of-the-art models pre-trained on Cityscapes: despite being built on Unreal Engine, CARLA lacks the photo-realism that the modern graphic engines make available, often for free.

The particular distribution of CARLA-generated images poses a series of challenges, each of which was addressed:

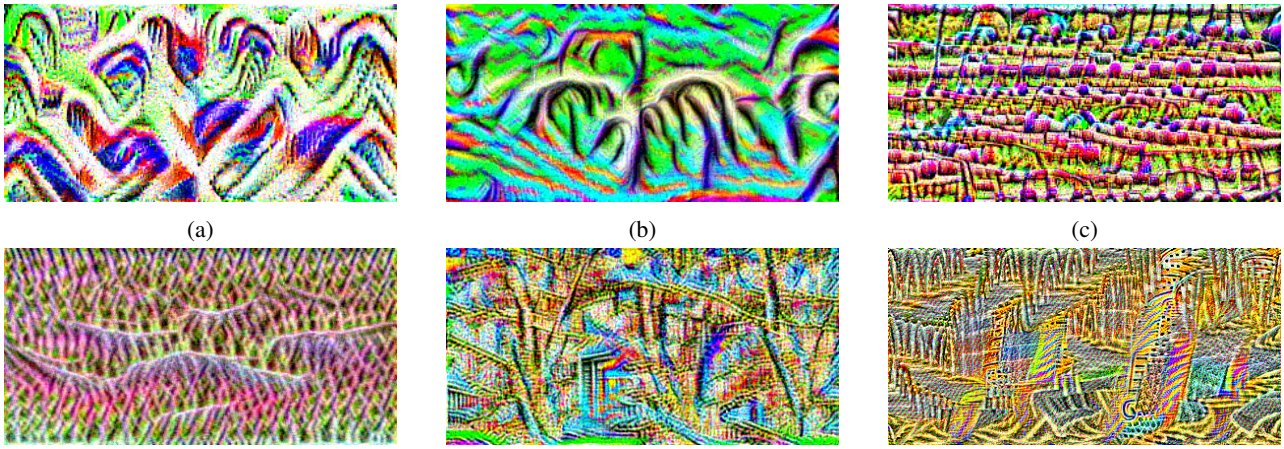
- Some classes are very prone to overfitting, since a large fraction of the total pixels of each image belongs to the classes road, sidewalk (that makes up a vast portion of the image), sky, and vegetation in some settings. Since the pre-trained versions of the networks already had good performance on those classes, during the fine-tuning a weight of 0.01 was assigned to the loss corresponding to these classes. This workaround solves the overfitting problem.
- The distribution is not "real-world", but digital, in the sense that it lacks the typical imperfections in the meshes composing the scene. Additionally, no camera noise is considered during acquisition. This was accounted for by adding noise during fine-tuning.
- The weather conditions heavily affect the appearance of the images, making the distribution change drastically. In order to avoid re-training from scratch, the choice fell on keeping the weather condition fixed as in the Cityscapes dataset.

E. Additional results

In this section we provide additional results and illustrative images extracted from the experiments performed on the Cityscapes (EOT-based attack) and CARLA datasets (scene-specific versus EOT-based attacks). Finally, we show additional examples of the real-world attack, obtained with real printed patches. Figure 1 reports some patches obtained by running EOT-based and scene-specific optimizations among CARLA, Cityscapes and our custom Patches-scapes datasets.

E.1. EOT-based patch attack on Cityscapes

Figure 2 and Figure 3 compare the predictions of different networks on images extracted from the Cityscapes validation set. In particular, Figure 2 shows how several patches (400×200) appear in a Cityscapes image, while Figure 3 shows



(a) (b) (c)

(d) (e) (f)

Figure 1: Patches obtained using specific attack on CARLA-scene 1 with BiSeNet (a), ICNet (b), DDRNet (c), while (d) and (e) are patches optimized using a digital EOT on the Patches-scapes dataset on ICNet and DDRNet, respectively. Finally, (f) is a patch optimized using again a digital EOT on Cityscapes with PSPNet.

the corresponding predicted semantic segmentation for each network (the top row of Figure 3 corresponds to the images in Figure 2, while next rows correspond to random images extracted from the Cityscapes validation set).

As anticipated in the main manuscript [2] in terms of mIoU achieved, PSPNet predictions are completely ruined by the adversarial patches optimized with the EOT-based attack, meaning that such a network has a weak spatial robustness. DDRNet and ICNet instead are more robust: in fact, adversarial patches are only capable of distorting smaller areas of the prediction, which are located close to the patch. Finally, BiSeNet (for which the main paper shows the semantic segmentation predictions) is not as robust as DDRNet and ICNet, but more robust than PSPNet.

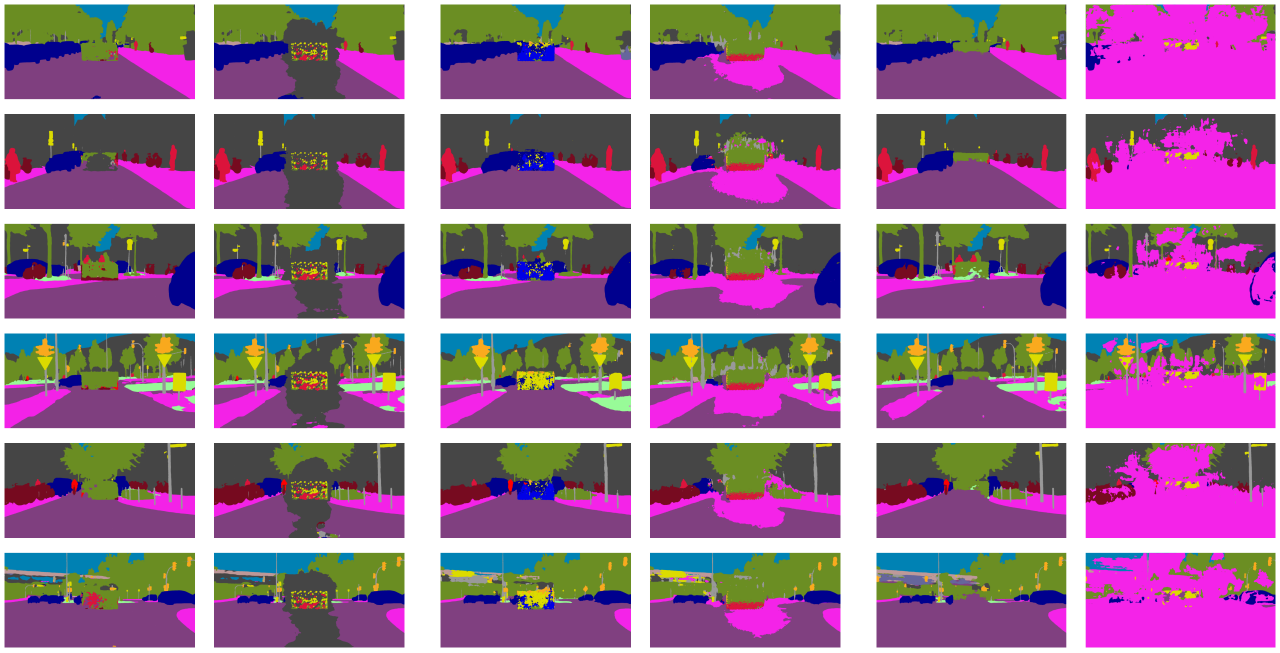
It is important to note that, although all the previous patches are robust to different positions and scaling factors, they were designed to attack models on Cityscapes following a digital approach (directly substituting pixels of a general test image). Without introducing any particular appearance constraints, their brightness and color are quite unrealistic within each image context. Therefore, they are more inclined to lose their adversarial strength in a real-world scenario, where also environmental lights and shadows are present.

To provide a fair interpretation of the achieved results with and without EOT on Cityscapes, we report the mIoU per class in Figure 4 for both PSPNet and BiSeNet. It shows that our adversarial patches were able to drastically reduce IoUs corresponding to each class of the Cityscapes dataset, while a simple random patch has a weaker effect. Clearly, patches optimized without EOT (i.e., fixing their appearance and position at the center of the image) achieve a more significant adversarial effect but lose their adversarial strength when applying simple transformations that are present in real-world scenarios.

Figure 2: First image represent a patched image with a random patch, while the others are adversarial patches (200×400) obtained with BiSeNet, DDRNet and ICNet, and PSPNet respectively.

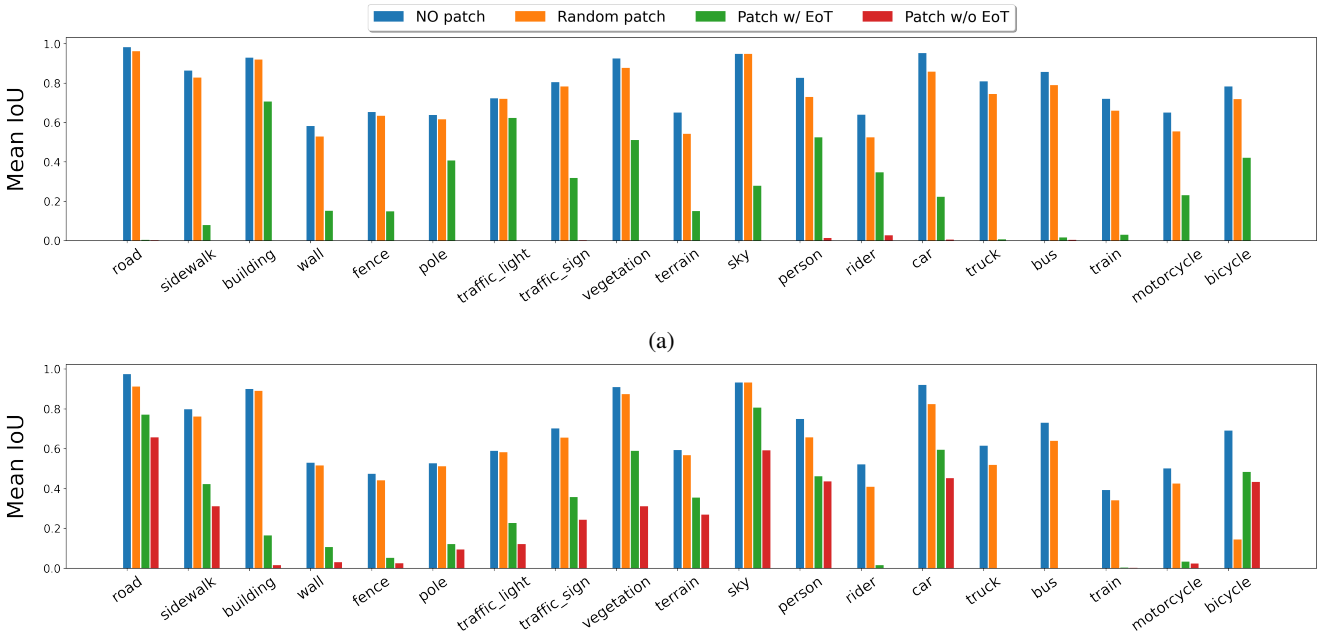
E.2. Scene-specific patch attack on CARLA dataset

Similarly to the previous results on Cityscapes, we present an attack analysis also on CARLA, comparing the EOT-based attack against the proposed scene-specific attack. Figure 5 shows the per-class mIoU on CARLA-scene1 with BiSeNet (Figure 5a) and CARLA-scene2 with DDRNet (Figure 5b). These histograms compare the effect of a patch for each class on the corresponding CARLA-scene test sets. We study the effect of applying a random patch, an EOT-based patch and



(a) (b) (c)

Figure 3: Semantic segmentations obtained from DDRNet (a), ICNet (b), PSPNet (c). For each block, the first column represents SS obtained from images patched with random patches (200×400), while the second column refers to images patched with patches crafted with EOT-based attack.



(a)

(b)

Figure 4: Per-class mIoU comparison on Cityscapes. Each bar shows the mIoU associated to a certain class, computed on the validation set. The results provide a comparison between 4 different cases: no patch applied into the images (original dataset), random patches, EOT-based and no-EOT-based patches. The reference networks were PSPNet (a) and BiSeNet (b).

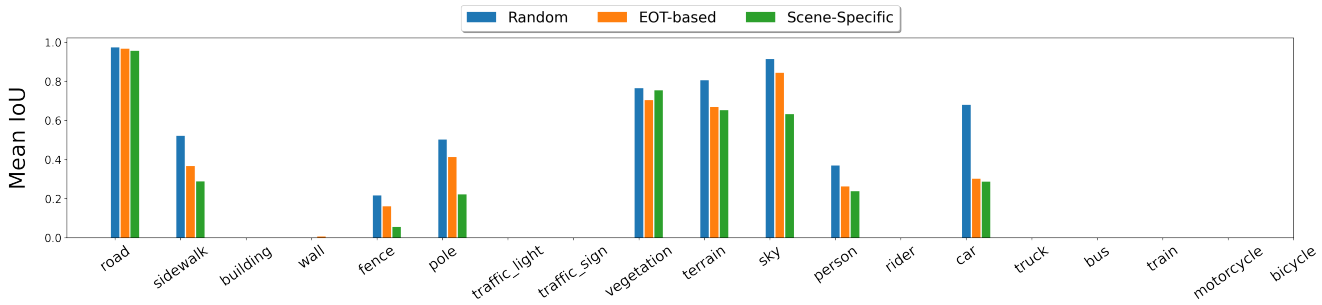
a scene-specific patch. In the following, we omitted the analysis with a non-robust patch (without-EOT), since it behaves

similar to a random patch, as shown by images and results reported in the main paper [2]. Clearly, the scene-specific attack outperforms the EOT-based attack, that achieves higher mIoU in almost all classes. Please note that, for sake of simplicity and to avoid issues during the fine-tuning process, we kept the same classes of Cityscapes for the CARLA datasets. However, some of them never appear in the scenarios simulated in CARLA, thus their corresponding mIoUs are usually equal to zero.

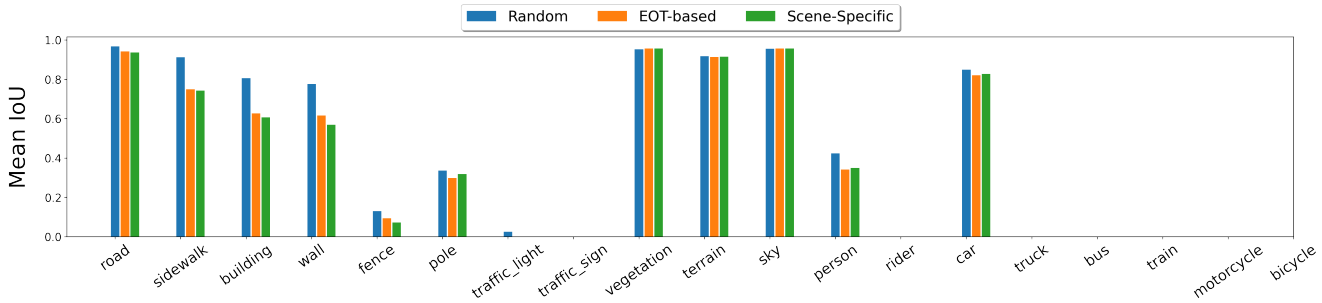
Note that to provide a fair evaluation, these results are extracted following the same approach introduced in the main manuscript [2]: the patch is imported in CARLA as a `.png` image, and stuck on the billboard as a *decal* object (`https://docs.unrealengine.com/4.26/en-US/Resources/ContentExamples/Decals/`). This simulates the real-world act of attaching a printed patch on a 2D surface.

To propose an additional comparison between all the implemented attacks on CARLA, we recorded a video that shows all the studied cases (i.e., no patch, random patch, EOT-based, Scene-Specific) on the same sequence of frames captured from CARLA on Scene-1. The video is available upon request.

Figure 7 reports some examples of predicted images that were extracted from CARLA-scene1 applying a scnee-specific patch (as before, top row is the prediction corresponding to the images showed in Figure 6). Clearly, BiSeNet (Figure 7a) is less robust than ICNet (Figure 7b), since the adversarial attack has effect on larger portions of the image, and also farther from the patch position. However also with ICNet, when the patched billboard is close to the camera, portions of the sidewalk disappear in the prediction, suggesting those attacks represent a possible safety threat against this model.



(a)



(b)

Figure 5: Per class mIoU comparison on CARLA. Each bar shows the mIoU corresponding to a certain class on the proposed CARLA dataset (we kept the same cityscapes classes) for Scene-1 (a) and Scene-2 (b), both meausered with their validation sets. The results show a comparison with random patch, EOT-based optimization and the proposed scene-specific on BiSeNet (a) and DDRNet (b). All the patches were imported directly in CARLA in order to provide a fair attack evaluation, as close as possible to a real world scenario.

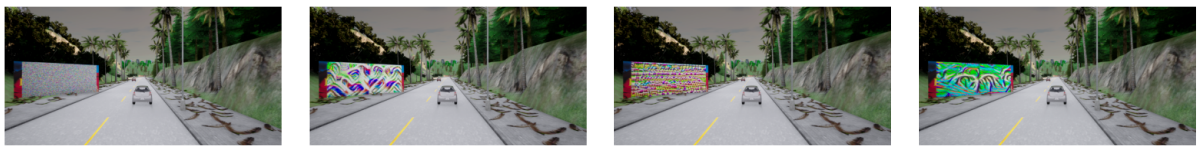
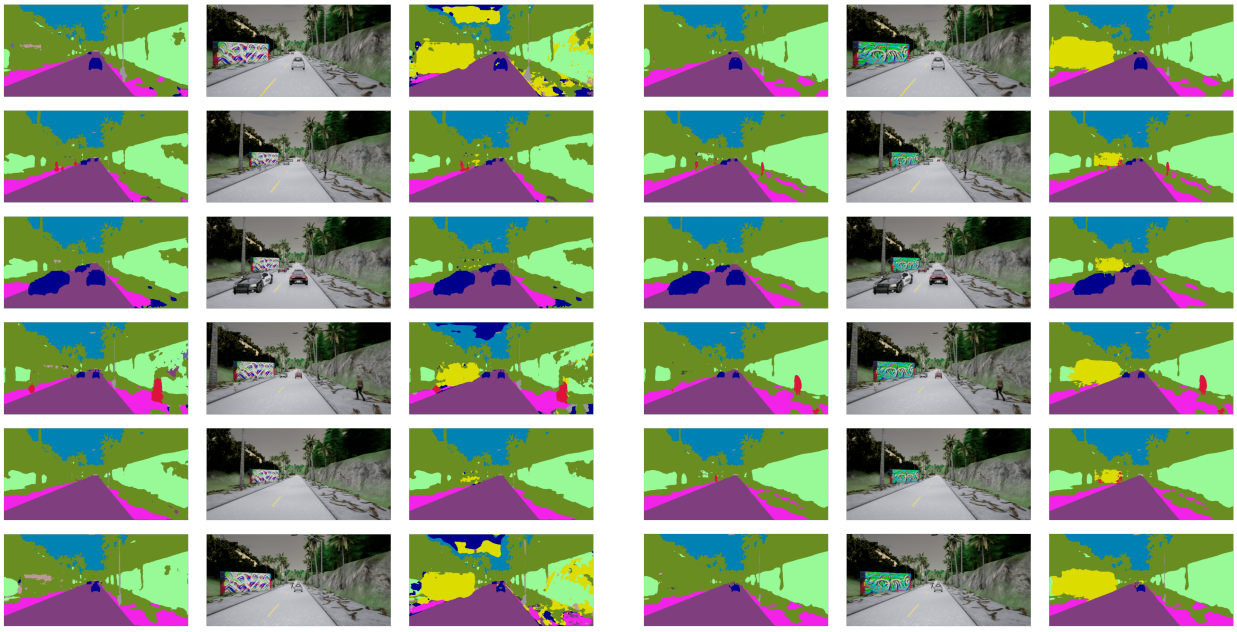


Figure 6: First image represent a patched image with a random patch, while the others are adversarial patches (200×400) obtained with BiSeNet, DDRNet and ICNet, respectively.



(a) (b)

Figure 7: Semantic segmentations obtained from BiSeNet (a) and ICNet (b) on CARLA Scene-1. For each block, the first column is the SS obtained from images patched with random patches, the second represents the patched image with the patch optimized with the scene-specific attack, and finally, the third column is its predicted SS.

E.3. Loss function ablation for scene-specific attack

Figure 8 provides the same study on the loss function that was performed for the EOT-based attack in the main manuscript [2]. The performance of the attack is reported for CARLA-scene 1 on ICNet. Clearly, the proposed extension of the cross-entropy outperforms the classical cross-entropy for almost all values of γ. These experiments were performed with the largest real-world patch size (7.5m wide - almost equal to the billboard size), with 150×300 pixels, with a learning rate of 0.5.

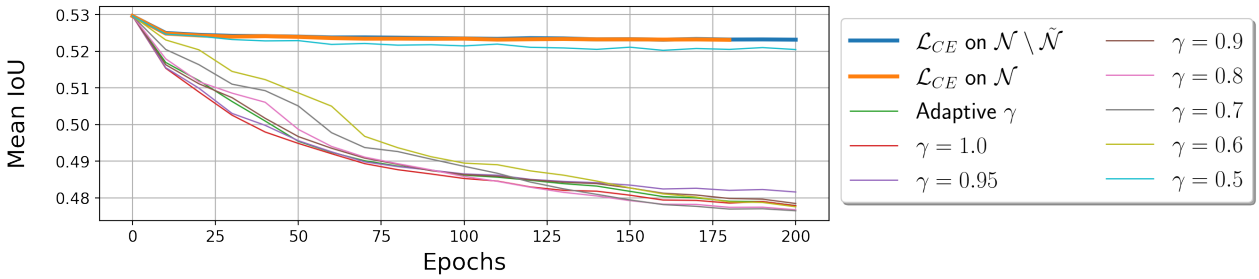


Figure 8: Comparison of adversarial patch optimizations on CARLA-scene 1 using different loss functions with the scene specific attack (two versions of the standard pixel-cross entropy and our formulation with multiple values of γ). The reference network here is ICNet [12].

E.4. Patch attributes for scene-specific attack

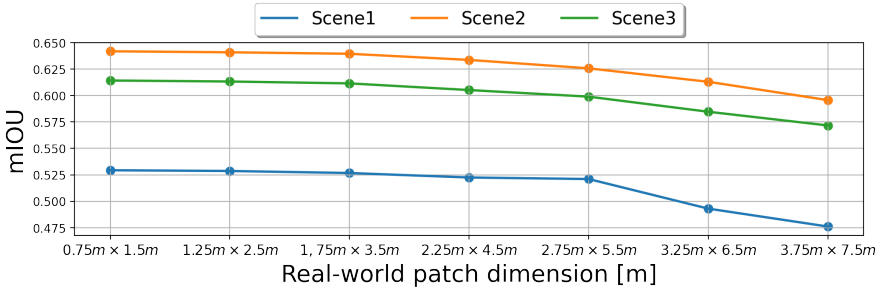
This section studies the effect of the real-world dimension of the patch and the effect of the number of pixels for a given dimension. As the above ablation study, the following tests are performed on CARLA-scene 1 using the ICNet model. The adversarial optimizations exploit the new proposed loss function with $\gamma = 1.0$.

Effect of real-world dimension of the patch For the EOT-based attack, the dimension of the patch within the image depends on the number of pixels (and, when used, on the scaling factor). For the scene-specific attack, the dimension of

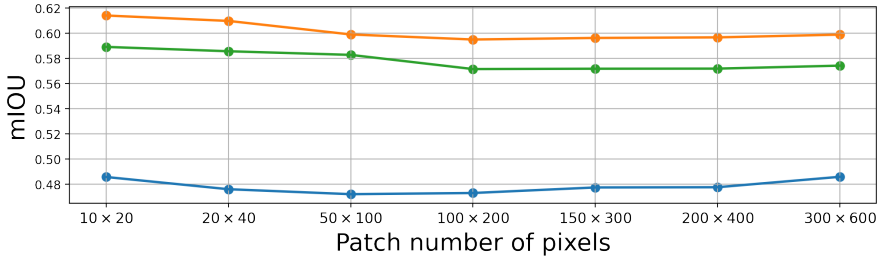
the patch within the image depends on both the real-world dimension of the patch (which, at most, is the dimension of the billboard), and the distance and orientation between the camera and the attackable surface. Figure 9a reports the results of the attack in terms of mIOU for different real-world patch dimensions and for the three different scenes. The number of pixels is fixed to 150x300.

It is clear that the larger the patch, the larger the effectiveness of the attack, independently of the scene considered. This is caused by the larger portion of the image that is occupied by the patch and it is an expected result.

Effect of number of pixels To understand the effect of the number of pixels on the attack performance, the real-world patch dimension is kept fixed to the maximum allowed (i.e., the dimension of the billboard), and the number of pixels is varied, keeping the same height/width ratio of 1/2. The results are summarized in Figure 9b and show that the attack becomes less effective when the number of pixels is too small or too large. This can be explained by observing that, when the number of pixels is high, the real-world pixels may become smaller than the image pixel (due to projection rescaling), making the optimization process not effective. On the other hand, having a few big real-world pixels reduces the degrees of freedom of the attack method.



(a)



(b)

Figure 9: Ablation study for the real-world dimension of the patch (a) and for the number of pixels used to design the patch (b).

This result highlights another advantage of using the scene-specific attack: a smaller number of pixels is required during the optimization. This speeds the optimization up. For a given scene (i.e., a given distance between the camera and the patch) the optimal real-world pixel size is the one that makes it equal to the size of the pixel in the image.

Based on the above considerations, the final experiments on CARLA, described in the main paper [2], are performed using a resolution of 150×300, which is an average value that yields good performance for the entire range of the scaling applied.

E.5. Black-box transfer across models

This section summarizes the cross-model black-box evaluation in Tables 1 and 2 on CARLA and Cityscapes, respectively. For CARLA, we provided results for both patches obtained with the EOT-based and the scene-specific optimizations, while for Cityscapes we only studied EOT-based patches since patches optimized without EOT are, by construction, less robust to evaluation settings changes.

In all the tested cases, it appears that the adversarial patches lose adversarial effect when transferring between semantic segmentation models. The scene-specific attack shows slightly better transferability with respect to the EOT-based attack, but no remarkable difference can be appreciated. It is a promising result that points out the difficulties to attack a semantic segmentation CNNs in a black-box, real-world scenario.

Model	mIoU — mAcc (scene-specific /EOT)					
Evaluated on → Optimized on ↓	ICNet		BiSeNet		DDRNet	
ICNet	0.48 / 0.49	0.54 / 0.56	0.44 / 0.44	0.61 / 0.60	0.50 / 0.51	0.69 / 0.70
	0.61 / 0.61	0.73 / 0.73	0.60 / 0.60	0.75 / 0.76	0.62 / 0.61	0.75 / 0.75
	0.59 / 0.59	0.74 / 0.73	0.47 / 0.47	0.74 / 0.74	0.65 / 0.65	0.78 / 0.78
BiSeNet	0.50 / 0.51	0.60 / 0.60	0.31 / 0.36	0.49 / 0.55	0.50 / 0.50	0.69 / 0.70
	0.63 / 0.63	0.73 / 0.73	0.58 / 0.58	0.74 / 0.74	0.62 / 0.62	0.75 / 0.75
	0.63 / 0.63	0.75 / 0.75	0.45 / 0.46	0.73 / 0.73	0.65 / 0.65	0.78 / 0.78
DDRNet	0.51 / 0.51	0.60 / 0.60	0.44 / 0.44	0.60 / 0.60	0.46 / 0.46	0.69 / 0.69
	0.63 / 0.63	0.73 / 0.73	0.60 / 0.60	0.75 / 0.76	0.49 / 0.52	0.66 / 0.71
	0.63 / 0.63	0.75 / 0.75	0.47 / 0.47	0.74 / 0.74	0.59 / 0.58	0.76 / 0.76

Table 1: Black-box transferability across models in terms of mIoU and mAcc. The results are showed for both the EOT-based and the scene-specific, for the three scenes of the CARLA datasets.

Model	mIoU - mAcc							
Evaluated on → Optimized on ↓	ICNet		BiSeNet		DDRNet		PSPNet	
ICNet	0.50	0.61	0.54	0.69	0.62	0.79	0.68	0.76
BiSeNet	0.62	0.71	0.29	0.43	0.62	0.79	0.69	0.77
DDRNet	0.65	0.75	0.56	0.68	0.59	0.69	0.67	0.77
PSPNet	0.65	0.78	0.58	0.70	0.67	0.80	0.23	0.30

Table 2: Black-box transferability on the Cityscapes dataset across models in terms of mIoU and mAcc. The results are showed for EOT-based patches.

E.6. Additional examples of real-world attack

Figure 10 provides some images captured during the final real-world tests near our lab. The objective of these tests was to get an illustrative evaluation of the adversarial effects in real autonomous driving scenario by comparing a printed optimized patch (Figure 10b) and a random one (Figure 10a), both $1m \times 2m$. We optimized the patch on the Patches-scapes dataset, captured on the roads near our lab. Since no ground truth is available for Patches-scapes, we used the predicted clear semantic segmentation (i.e., prediction obtained by original images, without patches) as label during the optimization process.

The pretrained ICNet model on Cityscapes was the target for this real-world attack. Since Cityscapes images belong to a realistic distribution (very close to the distribution defined by Patches-scapes), the semantic segmentation predicted on Patches-scapes were satisfactory when evaluated without a formal metric (i.e., mIoU, mAcc) due to the absence of ground truth labels.

Clearly, the effect of the adversarial patch is more visible than in the random case, and might represent a dangerous threat for autonomous driving systems when using semantic segmentation.

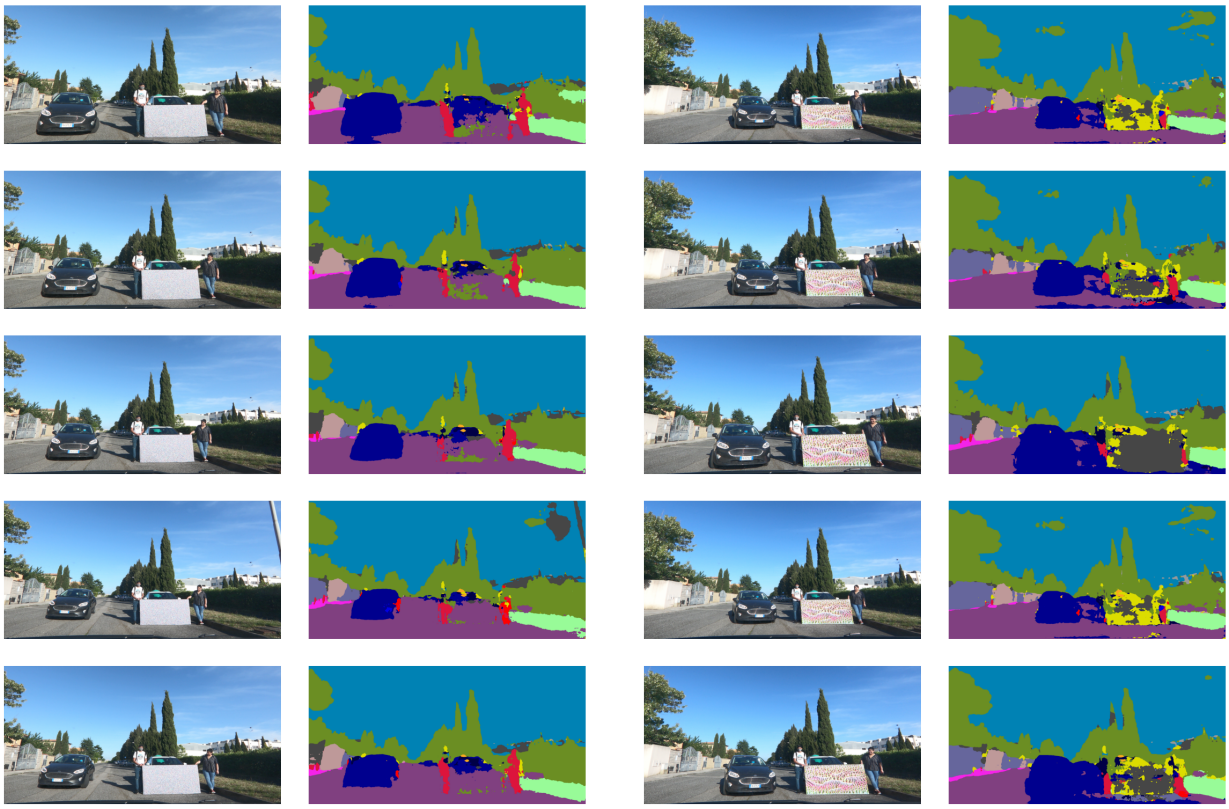
Moreover, it is important to remark that our printed patch is smaller, for cost reasons, than billboards that could be legally placed in a typical road scenario. Therefore, even if these experiments were performed as part of a preliminary investigation, they highlight how these attacks in the real world could represent an actual threat for a semantic segmentation model.

However, as discussed in the main paper, the effect of these patches is, in most of the cases, limited to the area surrounding the patch itself, due to a certain local spatial robustness of the SS models.

E.7. Effect of distance on adversarial power

As it can be expected, also from previous results, the distance from the camera and the adversarial patch plays a key role for the actual adversarial effect. Clearly, if the patch is far away from the camera, the patch will cover a small portion of the captured image, and will have small or no adversarial power.

However, if the patch is on one side of the road, the car will likely drive close to its location, and, consequently, the apparent size of the patch within the image will grow until it disappears from view. Figure 11 and 12 show several frames of a car approaching a patch in CARLA 3D virtual environment (scene 1) and real-world respectively. The adversarial effect is visibly growing as the car approaches the patch (as long as it is completely visible in the captured image).



(a) (b)

Figure 10: Effect of printed real-world patches ($1m \times 2m$). The images in (a) show the effect of a random patch, while in (b) an adversarial patch was used. The adversarial patch is a 200×400 patch optimized with the EOT-based attack on ICNet on Patches-scapes. For all the sampled images the corresponding semantic segmentation is placed on the right.

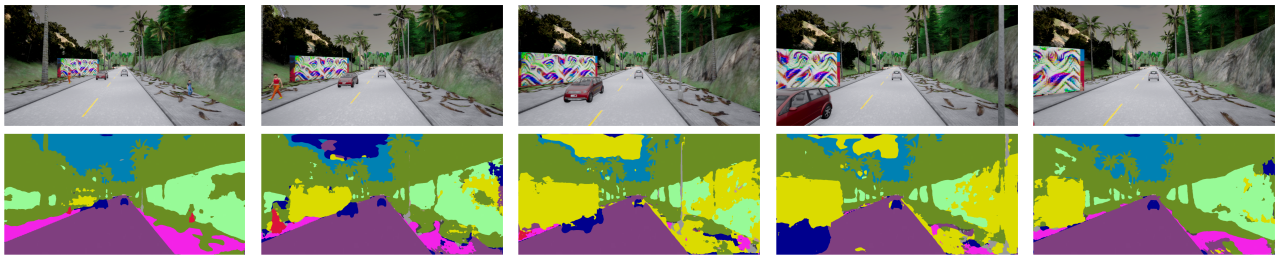


Figure 11: Illustration of the adversarial patch effect on CARLA simulator showing sequential frames of scene 1 processed by the BiSeNet model

Figure 12: Illustration of the real-world adversarial patch effect on sequential frames processed by the ICNet model.

References

[1] 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.

[2] Authors. Investigating real-world adversarial attacks against semantic segmentation for autonomous driving. *Submitted as WACV2022 ID161 main paper*, 2021.

[3] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv e-prints*, page arXiv:1610.02357, Oct. 2016.

[4] David A. Forsyth and Jean Ponce. *Computer Vision - A Modern Approach, Second Edition.* Pitman, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[6] 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.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 86–94. IEEE Computer Society, 2017.

[9] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary R. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 3663–3672. IEEE, 2020.

[10] 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.

[11] 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.

[12] 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.

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