도시 환경에서의 이미지 분할 모델 대상 적대적 물리 공격 기법

수랸토 나우팔[1], 라라사티 하라스타 타티마[1], 김용수[2], 김호원[1]
[1]부산대학교, [2]스마트엠투엠
{naufalso, harashta}@pusan.ac.kr, yongsu@smartm2m.co.kr, howonkim@pusan.ac.kr

Adversarial Wall: Physical Adversarial Attack on Cityscape Pretrained Segmentation Model

Naufal Suryanto[1], Harashta Tatimma Larasati[1], Yongsu Kim[2], Howon Kim[1]
[1]Pusan National University, [2]SmartM2M

Abstract

Recent research has shown that deep learning models are vulnerable to adversarial attacks not only in the digital but also in the physical domain. This becomes very critical for applications that have a very high safety concern, such as self-driving cars. In this study, we propose a physical adversarial attack technique for one of the common tasks in self-driving cars, namely segmentation of the urban scene. Our method can create a texture on a wall so that it can be misclassified as a road. The demonstration of the technique on a state-of-the-art cityscape pretrained model shows a fairly high success rate, which should raise awareness of more potential attacks in self-driving cars.

1. Introduction

Adversarial attack is a technique to fool deep learning models with deceptive data that look similar to human eyes. This is particularly harmful for critical applications like self-driving cars, which require high accuracy for maneuvering in various terrains, such as in urban scenes.

In this paper, we propose an adversarial attack technique that produces a texture on a wall so that it is misclassified as a road, i.e., adversarial wall. Subsequently, we demonstrate our method on a state-of-the-art pretrained segmentation model on the Cityscapes [1], the large-scale dataset for semantic urban street scenes. Our result shows the practical example of adversarial attack as a safety issue in self-driving cars.

2. Proposed Method

An adversarial wall is a targeted physical adversarial attack to find a texture of a wall such that the attacked segmentation model predicts it as a road. It can be achieved by optimizing the base texture η_{base} over the attack pipeline inspired by [2], as shown in Figure 1.

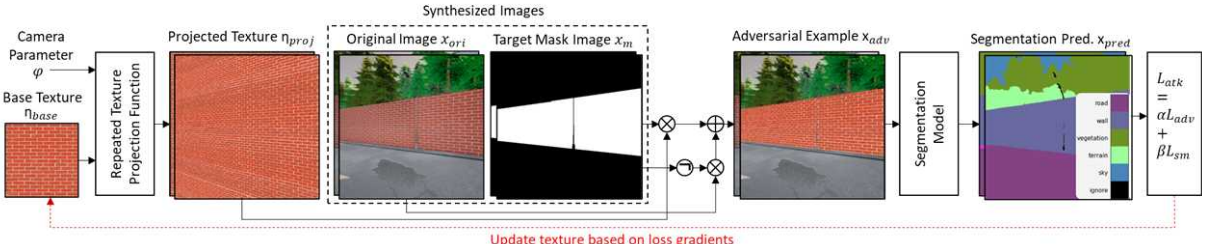
In this paper, we utilize a repeated texture projection function proposed by [2] to project the base texture η_{base} given the camera parameter Φ. The camera parameter consists of a transformation matrix M which covers shift, scale, and 3D rotation on 2D image operation. The function outputs the projected texture η_{proj} which can be formalized as follows:

$$\eta_{proj} = M \cdot \eta_{base} \tag{1}$$

The projected texture will then be masked with the target mask image x_m where the wall is located. Combined with the background image extracted from inverse mask $\neg x_m$ and original image x_{ori}, we can obtain an adversarial example x_{adv}. which can be written as:

$$x_{adv} = (\eta_{proj} \times x_m) + (x_{ori} \times \neg x_m) \tag{2}$$

The adversarial example x_{adv} is used as the input of target segmentation model $h(x)$. Since our target is to misclassify a wall into a road, we


Synthesized Images
Camera Parameter
φ
Base Texture η_{base}
Repeated Texture Projection Function
Projected Texture η_{proj}
Original Image x_{ori}
Target Mask Image x_m
Adversarial Example x_{adv}
Segmentation Model
Segmentation Pred. x_{pred}
road
wall
vegetation
terrain
sky
ignore
$L_{atk} = \alpha L_{adv} + \beta L_{sm}$
Update texture based on loss gradients

(Figure 1) Adversarial Wall optimization pipeline.

calculate the adversarial loss L_{adv} as:

$$L_{adv} = -\frac{1}{w \times h}\sum_{i}^{h}\sum_{j}^{w} log(h(x_{adv})_{road_{i,j}}) \times x_{m_{i,j}} \quad (3)$$

where $h(x_{adv})_{road}$ is the prediction of road from the target object segmentation model. w and h are the width and height of adversarial examples.

Furthermore, to reduce the inconsistency between neighboring pixels of the base texture, we employ a smooth loss following [3]. However, instead of using the squared loss, we use the same log loss to equalize the loss scale.

$$L_{sm} = \frac{1}{w \times h}\sum_{i-1}^{h}\sum_{j-1}^{w}(-log(\eta_{base_{i,j}} - \eta_{base_{i,j+1}}) - log(\eta_{base_{i,j}} - \eta_{base_{i+1,j}}) \quad (4)$$

Additionally, we combine both losses as attack loss L_{atk} where α and β are the weight to tune the contribution of each loss.

$$L_{atk} = \alpha L_{adv} + \beta L_{sm} \quad (5)$$

Using a gradient-based algorithm, we then find the optimal base texture to minimize attack loss:

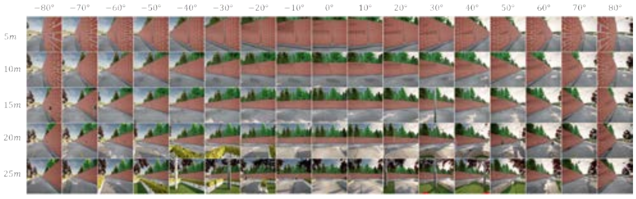
$$\eta_{atk} = argmin_{\eta_{base}} L_{atk} \quad (6)$$

3. Experiment and Result

3.1 Experiment Settings

This work uses pre-trained MaX-DeepLab [4] on Cityscape [1] dataset as the state-of-the-art target segmentation model. We synthesized wall images and masks using Unreal Engine 4 [5], a physical-based rendering engine, as our dataset. We capture the wall image for every 10-degree rotation and vary the distance for every 5 meters from as shown in Figure 2.

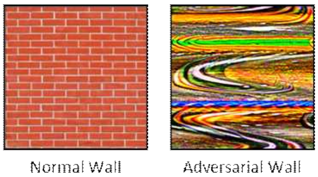
We randomly split the dataset into 3:1 as the training and validation dataset during the texture optimization. We optimize the texture using ADAM optimizer [6] and 10000 epochs. We use α = 1.0 and β = 2.0 for the attack loss.



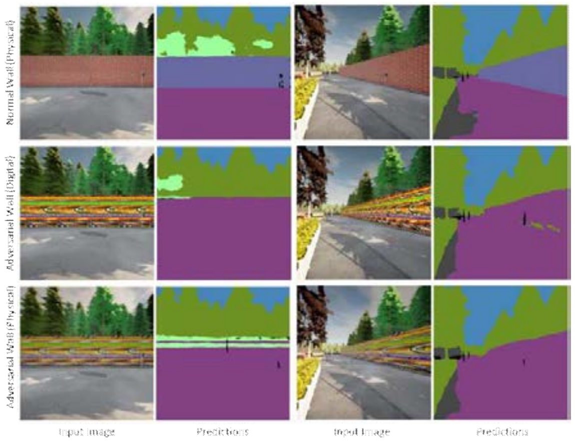
(Figure 2) Synthesized Datasets for Adversarial Wall

We evaluate the performance of adversarial wall in digital and physical simulation. In digital setting, we directly evaluate the performance of generated adversarial examples, while for the physical simulation, we evaluate the performance of optimized texture after it is rendered using the physical-based rendering engine. We calculate the attack success rate of each pixel where the wall should be predicted as road.

3.2 Experiment Results



(Figure 3) Base texture of normal and resulting adversarial wall after attack optimization is complete



(Figure 4) Sample evaluation prediction results

Figure 3 and 4 depicts base textures and sample prediction results of normal and adversarial walls in both digital and physical settings. It shows that normal walls (1st row) are segmented correctly (violet), while adversarial walls in digital (2nd row) and physical (3rd row) are predicted as road (purple).

Figure 5 and Figure 6 illustrate the performance of adversarial wall grouped by distances and angles, respectively. It can be inferred that variations in distance have a smaller effect on performance changes than variations in camera angle. Moreover, the distribution between evaluations in the digital and physical domains is similar. Overall, the physical evaluation performance is lower, which may be caused by physical factors such as lighting, materials, and interactions between objects not included in the optimization process.

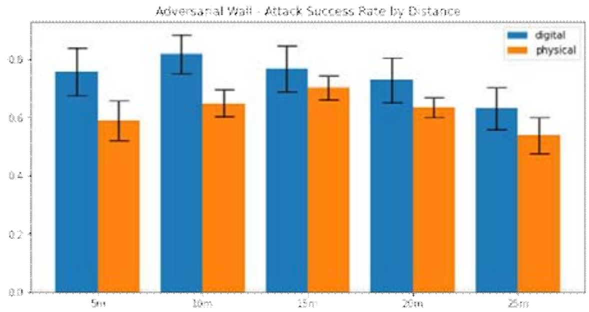


Figure 5. Performance of Adversarial Wall by distances

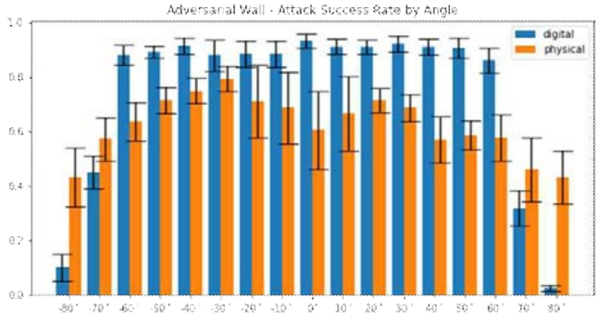


Figure 6. Performance of Adversarial Wall by angles

Table 1 shows the average adversarial wall performance for all scenes. Overall the success rate is quite high even though the object is viewed from various directions. The slight difference between digital and physical evaluation shows that the resulting texture is quite robust and can be implemented in the physical domain.

<Table 1> Average performance of Adversarial Wall for all scenes

Evaluation Setting	Digital	Physical
Avg attack success rate	74% ± 3%	62% ± 2%

4. Conclusions

This paper proposed a practical example of physical adversarial attack technique on urban segmentation model. We have shown that adversarial wall can potentially misguide a self-driving car by changing the wall prediction as road. To minimize the attack possibility, additional sensors should be equipped, such as lidar or proximity sensors, due to the limitation of a system relying solely on camera and AI.

Acknowledgment

This work is financially supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) as 「Innovative Talent Education Program for Smart City」.

References

[1] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding." In Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recognition, 2016.

[2] N. Suryanto et al., "DTA: Physical Camouflage Attacks using Differentiable Transformation Network." In Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recognition, 2022.

[3] M. Sharif et al., "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." In Proc. 2016 ACM SIGSAC Conf. on Comp. and Comm. Security, 2016.

[4] H. Wang et al., "Max-deeplab: End-to-end panoptic segmentation with mask transformers." In Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recognition, 2021.

[5] Epic Games. Unreal Engine, Available at: https://www.unrealengine.com.

[6] DP. Kingma, JL. Ba, "Adam: A Method for Stochastic Optimization." In Proc. the 3rd International Conference on Learning Representations, 2014.