Indirect Local Attacks for Context-aware Semantic Segmentation Networks

Supplementary Material

Krishna Kanth Nakka[1] and Mathieu Salzmann[1,2]

[1]CVLab, EPFL, Switzerland [2]ClearSpace, Switzerland
{krishna.nakka, mathieu.salzmann}@epfl.ch

1 Implementation Details

In this section, we provide detailed explanations about the experiments described in Section 4 of the main paper.

1.1 Models

All models for the experiments were implemented in PyTorch [12]. To generate adversarial attacks, we use the advertorch [5] library. Since different networks may have different normalization values for the mean and standard deviation of the input, we model normalization as a first layer inside the network and pass it as input an RGB image scaled to the range [0,1].

FCN. We use the publicly released model[1] from the authors of [19], which is trained together with PSANet [19] with an additional auxiliary loss. We use the ResNet-50 version for our experiments.

PSPNet. We use the trained model[1] released by the authors of [19]. It uses the same ResNet-50 as backbone network. The pyramid pooling module is a 4-level pyramid, which is concatenated to the final convolutional spatial map and later fed to a classification layer.

PSANet. We experiment with the trained model[1] provided by authors of [19] with ResNet-50 as backbone network. The PSA layer contains two sub-branches, namely collect and distribute, that favor a bi-directional information flow from each position to all other positions in the spatial feature map.

DANet. We use the trained model[2] from the authors of DANet [7]. DANet uses ResNet-101 as backbone network followed by a spatial and channel wise attention module. We use DANet with a hierarchy of grids of different sizes (4,8,16) in the last layer of each

[1]https://github.com/hszhao/semseg
[2]https://github.com/junfu1115/DANet

ResNet block.

DRN. We use the trained model[3] released by authors of [17]. We choose ResNet-22 as backbone network with dilated version corresponding to type *D*.

U-Net. Along with the above-mentioned models, we evaluate the robustness of the U-Net architecture to local attacks. Due to the non-availability of a trained PyTorch [12] version of the U-Net model, we re-trained it ourselves, achieving 33.7% mIoU on Cityscapes.

Along with the six Cityscapes models discussed above, we experiment on PASCAL VOC [6] with trained FCN [11][1] and PSANet [19][1] models provided by the authors of [19]. We do not rely on ground truth masks and use predicted maps for our experiments. Note that predicted segmentation maps are very accurate, with state-of-the-art models reaching a pixel-wise accuracy $> 95\%$ on unattacked data. To be precise, we found the percentage of perturbed pixels lying within the targeted region to be $< 1\%$ in all cases. For example, in the adaptive attacks of Table 3 (a) of main paper, with S=75$\%$, this percentage is 0.2%, 0.16%, 0.12%, 0.14% for FCN, PSANet, PSPNet, DANet, respectively, which shows that our attacks truly are indirect. Given the dual objective of the loss functions, it may happen that the gradients to maximize the confidence of labels at non-targeted locations dominate those at targeted ones. Hence, as suggested in [8], we ignore the loss at locations where the label is predicted correctly as the target label with a confidence of at least 0.3.

1.2 Datasets

Cityscapes: We use the validation set of the Cityscapes [4] dataset consisting of 500 images from 19 classes. We divide the pixels at every position in the image into one of two sets, based on the category attribute provided by the authors. The first set consists of pixels belonging to static classes with category attribute *road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky*. The second set corresponds to regions of dynamic classes *person, rider, car, truck, bus, train, motorcycle, bicycle*.

The Cityscapes dataset has on average 8% of the pixels corresponding to dynamic classes in each image. Since our study was targeted to mis-classify the dynamic objects, images with dynamic instances that occupy small regions might not be meaningful as such regions lie in the immediate receptive field of their surroundings. Therefore, we take a subset of images consisting of 150 images whose combined region of instances corresponding to vehicle classes (*car, truck, bus, train, motorcycle, bicycle*) is greater than 8%. We provide the statistics of the resulting dataset in Table 1.

While the original Cityscapes dataset was captured at 2048×1024 resolution, we resize the images to the half resolution of 1024×512 as the original size is too large to fit into GPU memory. Furthermore, we crop the bottom region of the image corresponding to the ego-vehicle of height 62 pixels and resize the image back to 1024×512 pixels. For fair comparison, all models use the same 1024×512 resolution as input

[3] https://github.com/fyu/drn

Dynamic class	Images
Person	115
Rider	66
Car	150
Truck	33
Bus	23
Train	7
Motorcycle	24
Bicycle	88

Table 1: **Cityscapes-sampled dataset.** We provide the statistics of the 150 images whose combined instance area of *vehicle* categories is more than 8%.

to the network without any tiling.

PASCAL VOC: We use a subset of 250 images from the original validation set consisting of 1449 images. It contains 20 foreground classes and one background class. In all settings, we target the pixels corresponding to all 20 foreground classes by perturbing a subset of the background area.

1.3 Attack Algorithms

We solve the indirect attacks given in Sections 3.1 and 3.2 of the main paper using the efficient iterative projected gradient descent algorithm [1] with an ℓ_p-norm perturbation budget $\|\mathbf{M} \odot \delta\|_p < \epsilon$, where $p \in \{2, \infty\}$, using a step size α. In all our experiments, we set the maximum perturbation ϵ as 100 times α for ℓ_∞ attacks. For ℓ_2 attacks, we set the maximum ℓ_2 norm of the perturbation ϵ to 100.

Formally, given an input image $\mathbf{X}$, the adversarial attack minimizes the objective function, $J_t(\mathbf{X}, \mathbf{M}, \mathbf{F}, \delta, f, \mathbf{y}^{pred}, \mathbf{y}^t)$ to find the optimal δ. We solve for δ in an iterative manner as

$$\delta^{(0)} = \mathbf{0} \tag{1}$$

$$\delta^{(n+1)} = \text{Clip}_\varepsilon^p \left\{ \delta^{(n)} - \alpha \nabla_{\mathbf{X}} J_t(\mathbf{X}, \mathbf{M}, \mathbf{F}, \delta, f, \mathbf{y}^{pred}, \mathbf{y}^t) \right\}, \tag{2}$$

where $\text{Clip}_\varepsilon^p$ clips the perturbation within the ℓ_p ball of radius ϵ. For ℓ_∞-norm based attacks, the gradient update is given by

$$\nabla_{\mathbf{X}} J = \text{sgn}(\nabla_{\mathbf{X}}(J_t(\mathbf{X}, \mathbf{M}, \mathbf{F}, \delta, f, \mathbf{y}^{pred}, \mathbf{y}^t))), \tag{3}$$

where sgn is the sign function.

For ℓ_2-norm based attacks, the gradient update is given by

$$r = \nabla_{\mathbf{X}}(J_t(\mathbf{X}, \mathbf{M}, \mathbf{F}, \delta, f, \mathbf{y}^{pred}, \mathbf{y}^t)) \tag{4}$$

$$\nabla_{\mathbf{X}} J = \frac{r}{\|r\|_2}. \tag{5}$$

We observe that the DAG attack [16] is similar to the PGD-ℓ_2 attack. While DAG projects the gradient as $\frac{r}{\|r\|_\infty}$, PGD-ℓ_2 projects the gradient as $\frac{r}{\|r\|_2}$. We emphasize that our formalism for local indirect attacks is general and could be applied to other adversary generation techniques [16, 3].

1.4 Attack Detection Algorithms

State-of-the-art methods. In this paper, we compare our approach with the spatial consistency [15] method and image re-synthesis method [9] for adversarial attack detection at image level. For the former, following [15], given an input image of 1024×512 pixels, we crop 50 sufficiently overlapping pairs of patches of size 256×256 and compute the average mIoU of the overlapped patch regions as the confidence score for attack detection. For the latter, following [9], we use the pix2pix generator to re-synthesize the image from the label map and then compute the ℓ_2 distance of the input image and the re-synthesized one in HOG feature space.

Our method. Let us now provide the implementation details of our attack detection based on the Mahalanobis distance. During training, we compute the class-conditional mean μ_c^ℓ at every layer ℓ of the network within locations corresponding to class label c of the ground truth. Furthermore, we compute the group variance $\mathbf{\Sigma}^\ell$ for every layer ℓ of the network using the features extracted at layer ℓ. Since the number of features extracted from the training set can be high, we propose to compute the mean and variance of averaged features within locations corresponding to each label.

Formally, let $\mathbf{X}_j^\ell$ be the feature extracted at layer ℓ at position j for image $\mathbf{X}$. Let the size of the feature map $\mathbf{X}^\ell$ be $W_\ell \times H_\ell \times K_\ell$, where W_ℓ, H_ℓ, K_ℓ are the width, height and number of channels for layer ℓ. Let $L^c \in \mathbb{R}^{W_\ell \times H_\ell}$ be the label mask activated at positions where the label is c, i.e., L_j^c = 1 if the j-th pixel location belongs to label c and $L_j^c = 0$ otherwise.

First, we compute the averaged feature corresponding to label c given by $\hat{\mathbf{X}}_c^\ell = \sum_{j|L_j=1} \mathbf{X}_j^\ell$. We then learn μ_c^ℓ and $\mathbf{\Sigma}^\ell$ using $\{\hat{\mathbf{X}}_c^\ell | \mathbf{X} \in [\mathbf{X}_0, ..., \mathbf{X}_N]\}$ extracted from all N images in the training set. In the end, we obtain $\mu_c^\ell \in \mathbb{R}^{K_\ell}$ and $\mathbf{\Sigma}^\ell \in \mathbb{R}^{\mathbb{K}_\ell \times \mathbb{K}_\ell}$ for a layer ℓ in the network, and use these values to compute the confidence score of Eq.(6)

We extract features at the end of every block in the ResNet backbone followed by a context layer and a classification layer. By doing so, we obtain a feature vector for the logistic detector of size $L = 6$ for FCN; $L = 7$ for PSANet; $L = 7$ for PSPNet; $L = 5$ for DANet; $L = 5$ for DRN. We pass the confidence map at each level through 5×5 average kernel to smooth the scores. For evaluation purpose, we use 80% of the data for training and the remaining 20% for testing.

1.5 Performance Metrics

For evaluation, we use the following metrics to measure the effectiveness of our indirect local attacks.

Intersection over Union. We report the IoU used in the domain of segmentation to

evaluate the effectiveness of the attack. We report the mIoU at positions that we aim to fool (f) since at the remaining positions, the label is retained around 98% of the time. For untargeted attacks, we report $\text{mIoU}^f_{\mathbf{u}}$ as the mIoU calculated between the normal image prediction and its counterpart adversarial image prediction at fooling positions. In the case of targeted attacks, along with $\text{mIoU}^f_{\mathbf{u}}$, we report $\text{mIoU}^f_{\mathbf{t}}$ as the mIoU calculated between the normal image prediction and targeted label map at fooling positions.

Attack Success Rate. We report the attack success rate at the percentage of pixels mis-classified/preserved relative to the total number of pixels in the fooling/preserved positions, respectively. We report the mASR separately at two positions: 1) at positions that we aim to fool (f); 2) at the remaining positions where the label should be preserved (p). We report $\text{mASR}^f_{\mathbf{u}}$ and $\text{mASR}^p_{\mathbf{u}}$ as the success rates calculated between the normal prediction and its adversarial image prediction at the fooling and preserved positions, respectively, for untargeted attacks. Specifically to calculate $\text{mASR}^f_{\mathbf{u}}$, we assume the attack to be successful at a pixel if it misclassifies it to any label other than the normal predicted label. In the case of targeted attacks, we additionally report $\text{mASR}^f_{\mathbf{t}}$ as the success rates calculated between the normal prediction and targeted label map at fooling positions.

Perceptibility. We take the ℓ_∞-norm and ℓ_2-norm of the perturbation image as the two perceptibility scores.

We average the above metrics over the entire test set. Since in almost all experiments the labels are retained around 98% of the time at preserved positions, we omitted reporting mASR^p_u in the main paper. We reported only $\text{mASR}^f_{\mathbf{t}}$ and $\text{mIoU}^f_{\mathbf{u}}$ at the fooling positions in the main paper as these metrics values are the most diverse in our different attack settings.

AUROC. The area under the receiver operating characteristic curve (AUROC) is computed by plotting the true positive rate (TPR) against the false positive rate (FPR) by varying a threshold. We compute the AUROC both at image level and pixel level and report them in all perturbation settings.

1.6 Time Complexity

For an input image of size 512×1024, the PGD-based indirect attack of Eq. (2) in the main paper takes on average ~ 35 seconds for 100 iterations, whereas our group-sparsity-based attack in Eq. (4) of the main paper takes on average ~ 90 seconds when using a maximum of 400 gradient computations. For comparison, a dense adversary generation attack [16], consisting of projecting the gradient in each iteration, takes ~ 40 seconds for a maximum of 200 iterations. Importantly, these timings remain practical in the scenario of physical attacks where the perturbation can be computed offline.

2 Additional Results

2.1 Cityscapes Experiments

Tables 2 and 3 show the performance of different networks by varying the noise levels for ℓ_∞ and ℓ_2 attacks. Tables 4 and 5 show the impact of indirect attacks by perturbing static regions that are at least d pixels away from any dynamic object class with ℓ_∞ and ℓ_2 attacks. Furthermore, Table 6 shows the complete performance statistics of different networks by tuning the sparsity levels in our adaptive attack strategy. We then show the impact of universal, single fixed-size patch attacks in Table 7 by varying the size of the patch placed at the center of the image.

Finally, we show the attack detection results with four perturbation settings: Global image perturbations (Global) to fool the entire image; Universal patch perturbations (UP) at a fixed location to fool the entire image; Full static (FS) class perturbations to fool the dynamic classes; Adaptive patch (AP) perturbations in the static class regions to fool the dynamic objects. Tables 8, 9, 10 and 11 show the attack detection results of our method and of the two state-of-the-art detection techniques with FCN, PSP, PSANet, and DANet, respectively.

2.2 Transferability Analysis

Tables 12a and 12b show the performance of black-box attacks when the *entire image* is perturbed to misclassify the dynamic objects in an untargeted and targeted manner, respectively. We observe that transferring adversarial examples to FCN, PSPNet and PSANet is more successful than to DANet, which is more robust to black-box attacks.

Table 13a shows the transferability of perturbations when the complete regions belonging to the *static classes* are perturbed to perform a targeted attack on the dynamic objects. Note that the transfer rate $\text{mASR}_\mathbf{t}^f$ is low ($< 5\%$) in many cases across all networks for this setting.

Table 13b shows the transferability of perturbations when the regions belonging to the *static classes* that are $d = 50$ pixels away from any *dynamic object boundary* are perturbed to perform a targeted attack on the dynamic objects. This, however, results in an even lower transfer rate $\text{mASR}_\mathbf{t}^f < 3\%$ across architectures as the contextual dependency differs across different architectures.

Finally, Tables 14a and 14b provide the transferability analysis with adaptive patch attacks and universal local attacks. We observe similar results in this setting, where the success rates $\text{mASR}_\mathbf{t}^f$ of black-box attacks are very low ($< 5\%$) when attacked with patch attacks. We conjecture that attacking segmentation networks that differ in contextual reasoning can be difficult for patch based attacks that perturb a small region. Furthermore, note that the poor transferability of black-box attacks in semantic segmentation was also observed in [15], although only for global attacks.

2.3 Comparison to Other Patch Attacks

Similarly to [10, 14, 2], we perform universal patch attacks that are fixed in terms of size and location. Table 15 shows the success rate of class-specific, targeted universal patch attacks on dynamic object regions, preserving the background region. Furthermore, we also learn a patch that is targeted to all dynamic classes, which in most cases results in $\text{mASR}^f_{\mathbf{t}} < 5\%$ for all networks. We observe that such fixed-size and fixed-location patches are not suitable for semantic segmentation unlike for object detection [10, 14] and image recognition [2]. This can be attributed, for image recognition, to global average pooling, which extends the receptive field to most parts of the image, making the perturbation location invariant, and, for object detection, to the anchor proposal layer that typically encompasses different scales so as to cover the entire image for any location as anchor center. In both cases, this makes the networks more vulnerable to attacks, even when the perturbations are far from the object of interest. By contrast, for segmentation without contextual layer such as FCN, the receptive field is limited to that of the backbone network, and the impact of a patch perturbation is limited to its surrounding.

2.4 Qualitative Results on Cityscapes

Figure 1 visualizes adversarial images obtained by varying the step size α in both ℓ_∞ and ℓ_2 indirect local attacks with PSANet [19]. Figure 2 shows the outputs of indirect local attacks by perturbing static class pixels that are at least a distance d from a dynamic class pixel. Figure 3 shows the outputs of universal patch attacks on different networks by varying the patch area in $\{1\%, 2.3\%, 4\%, 9\%\}$ of the image area. Figure 4 shows the results of adaptive local attacks on different networks by varying the sparsity level of the perturbation.

To understand the effectiveness of the Mahalanobis distance for attack detection, we visualize the internal subspaces of normal and adversarial samples. Figures 5 and 6 show the visualizations of the nearest cluster assignment for each spatial location in the top-4 layers for PSPNet [18] and PSANet [19], respectively. Figure 7 depicts the output of pixel-level adversarial attack detection using the Mahalanobis distance on PSANet [19] with adaptive indirect local attacks at a sparsity level of 75%.

2.5 Quantitative Results on PASCAL VOC

Table 16 shows the better robustness to adaptive local indirect attacks of FCN [11] than of PSANet [19]. For example, at a sparsity level of 95%, FCN [11] has a success rate of 13%, compared to 68% for PSANet. Furthermore, Table 17 shows the higher vulnerability to fixed-size universal patch attacks of PSANet [19] and PSPNet [18] than of FCN [11].

2.6 Qualitative Results on PASCAL VOC

Figure 8 shows the results of adaptive local attacks on PSANet [19] at a sparsity level of 95%. Figure 9 depicts the outputs of universal local attacks on different networks for PASCAL VOC.

Networks	α	mIoU		mASR			Norm of δ	
		$\text{mIoU}^f_{\mathbf{u}}$	$\text{mIoU}^f_{\mathbf{t}}$	$\text{mASR}^p_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	1e-5	0.65	0.08	100%	6%	5%	0.001	0.83
	1e-4	0.29	0.27	100%	35%	29%	0.01	4.70
	1e-3	0.14	0.49	100%	63%	56%	0.10	15.12
	5e-3	0.11	0.55	100%	69%	62%	0.40	50.93
PSPNet [18]	1e-5	0.71	0.10	99%	15%	12%	0.001	0.77
	1e-4	0.06	0.53	100%	98%	86%	0.01	3.10
	1e-3	0.00	0.62	100%	100%	90%	0.05	8.30
	5e-3	0.00	0.63	99%	100%	90%	0.20	37.99
PSANet [19]	1e-5	0.60	0.10	98%	22%	14%	0.001	0.72
	1e-4	0.04	0.51	99%	99%	86%	0.01	2.68
	1e-3	0.01	0.60	99%	100%	90%	0.05	8.10
	5e-3	0.00	0.60	99%	100%	90%	0.18	35.71
DANet [7]	1e-5	0.80	0.06	100%	6%	5%	0.001	0.81
	1e-4	0.11	0.50	99%	91%	80%	0.01	3.90
	1e-3	0.01	0.65	99%	99%	90%	0.04	8.30
	5e-3	0.00	0.66	99%	100%	90%	0.15	31.71
DRNet [17]	1e-5	0.64	0.09	99%	9%	6%	0.001	0.87
	1e-4	0.15	0.44	99%	67%	56%	0.01	4.95
	1e-3	0.03	0.67	99%	92%	84%	0.08	12.78
	5e-3	0.02	0.67	99%	94%	87%	0.27	40.2
U-Net [13]	1e-5	0.35	0.15	99%	29%	20%	0.001	0.91
	1e-4	0.02	0.37	99%	95%	76%	0.01	5.74
	1e-3	0.00	0.48	99%	99%	87%	0.08	13.34
	5e-3	0.00	0.52	99%	100%	89%	0.28	38.89

Table 2: **Indirect attacks** on Cityscapes to fool dynamic classes while perturbing entire static ones with ℓ_∞ strategy. The success rate of the attacks increases with higher step size α although with higher perceptibility values. FCN is more robust to indirect attacks, while PSANet and PSPNet are more vulnerable to attacks even at small step sizes such as α = 1e-4.

Networks	α	mIoU		mASR			Norm of δ	
		$\text{mIoU}^f_{\mathbf{u}}$	$\text{mIoU}^f_{\mathbf{t}}$	$\text{mASR}^p_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	8e-3	0.60	0.10	100%	13%	10%	0.02	0.58
	4e-2	0.36	0.21	99%	33%	26%	0.05	1.75
	8e-2	0.27	0.28	99%	44%	36%	0.08	2.58
PSPNet [18]	8e-3	0.68	0.12	99%	24%	20%	0.01	0.51
	4e-2	0.23	0.37	99%	81%	67%	0.02	1.28
	8e-2	0.02	0.84	99%	96%	91%	0.03	1.17
PSANet [19]	8e-3	0.60	0.10	98%	25%	14%	0.01	0.39
	4e-2	0.21	0.32	99%	85%	63%	0.02	0.90
	8e-2	0.06	0.53	99%	96%	83%	0.03	1.44
DANet [7]	8e-3	0.79	0.08	99%	16%	12%	0.02	0.56
	4e-2	0.43	0.28	99%	62%	50%	0.03	1.32
	8e-2	0.13	0.54	99%	90%	79%	0.035	1.95
DRNet [17]	8e-3	0.63	0.10	99%	16%	10%	0.02	0.65
	4e-2	0.24	0.37	99%	60%	48%	0.06	2.14
	8e-2	0.13	0.45	99%	76%	65%	0.08	3.02
U-Net [13]	8e-3	0.32	0.17	99%	36%	25%	0.02	0.70
	4e-2	0.05	0.32	98%	85%	66%	0.08	2.76
	8e-2	0.02	0.43	98%	95%	79%	0.09	3.43

Table 3: **Indirect attacks** on Cityscapes to fool dynamic classes while perturbing entire static ones with ℓ_2 *strategy*. The perceptibility values of ℓ_2 attacks are much lower than those of ℓ_∞ attacks at a given success rate. As in the case of ℓ_∞ attacks, FCN is more robust to indirect attacks than PSANet and PSPNet.

Networks	d	mIoU		mASR			Norm of δ	
		$\text{mIoU}^f_{\mathbf{u}}$	$\text{mIoU}^f_{\mathbf{t}}$	$\text{mASR}^p_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{u}}$	$\text{mASR}^f_{\mathbf{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	50	0.77	0.05	100%	4%	3%	0.38	43.37
	100	0.98	0.00	100%	0%	0%	0.38	33.46
	150	1.00	0.00	100%	0%	0%	0.38	22.23
PSPNet [18]	50	0.14	0.37	99%	96%	74%	0.28	41.83
	100	0.24	0.26	98%	86%	60%	0.29	33.00
	150	0.55	0.12	97%	35%	23%	0.34	22.86
PSANet [19]	50	0.11	0.33	98%	98%	72%	0.25	42.11
	100	0.13	0.27	98%	97%	65%	0.25	33.00
	150	0.28	0.21	98%	75%	47%	0.30	22.47
DANet [7]	50	0.14	0.50	99%	92%	81%	0.29	41.17
	100	0.48	0.24	98%	53%	43%	0.33	34.50
	150	0.80	0.07	98%	14%	10%	0.35	23.45
DRNet [17]	50	0.37	0.20	99%	34%	22%	0.43	46.30
	100	0.73	0.05	99%	5%	3%	0.44	37.24
	150	0.94	0.00	100%	0%	0%	0.47	25.87
U-Net [13]	50	0.01	0.25	98%	97%	70%	0.43	44.62
	100	0.03	0.20	96%	90%	60%	0.47%	39.61
	150	0.10	0.17	95%	74%	47%	0.49%	33.27

Table 4: **Impact of local attacks** by perturbing pixels that are at least d pixels away from any dynamic class with ℓ_∞ *strategy*. We observe PSANet [19] and UNet [13] to be vulnerable to indirect attacks even when the perturbations are at large distances, such as $d = 150$, while FCN [11] is barely affected.

Networks	d	mIoU		mASR			Norm of δ	
		$\text{mIoU}^{f}_{\mathbf{u}}$	$\text{mIoU}^{f}_{\mathbf{t}}$	$\text{mASR}^{p}_{\mathbf{u}}$	$\text{mASR}^{f}_{\mathbf{u}}$	$\text{mASR}^{f}_{\mathbf{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	50	0.80	0.05	100%	3%	3%	0.31	10.71
	100	0.98	0.00	100%	0%	0%	0.32	9.95
	150	1.00	0.00	100%	0%	0%	0.40	9.43
PSPNet [18]	50	0.18	0.35	99%	94%	73%	0.13	9.58
	100	0.30	0.24	98%	78%	56%	0.16	9.70
	150	0.59	0.11	98%	29%	20%	0.24	9.65
PSANet [19]	50	0.10	0.37	99%	98%	76%	0.19	9.41
	100	0.14	0.29	98%	95%	67%	0.22	9.43
	150	0.31	0.21	98%	70%	45%	0.27	9.55
DANet [7]	50	0.27	0.40	99%	83%	72%	0.19	9.90
	100	0.67	0.15	98%	33%	26%	0.22	9.87
	150	0.85	0.05	98%	10%	7%	0.30	9.51
DRNet [17]	50	0.44	0.15	99%	30%	17%	0.31	12.55
	100	0.77	0.04	99%	5%	3%	0.32	12.23
	150	0.95	0.00	100%	0%	0%	0.37	11.50
U-Net [13]	50	0.02	0.23	98%	95%	67%	0.28	16.13
	100	0.12	0.16	95%	68%	42%	0.58	19.51
	150	0.12	0.16	95%	67%	42%	0.58	19.56

Table 5: **Impact of local attacks** by perturbing pixels that are at least d pixels away from any dynamic class with $\ell_\mathbf{2}$ *strategy*. We observe PSANet [19] and UNet [13] to be vulnerable to indirect attacks even when the perturbations are at large distances, such as $d = 150$, while FCN [11] is barely affected.

Networks	Sparsity	mIoU		mASR			Norm of δ	
		$\text{mIoU}^{f}_{\mathbf{u}}$	$\text{mIoU}^{f}_{\mathbf{t}}$	$\text{mASR}^{p}_{\mathbf{u}}$	$\text{mASR}^{f}_{\mathbf{u}}$	$\text{mASR}^{f}_{\mathbf{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	75%	0.52	0.12	100%	18%	13%	0.15	4.04
	85%	0.67	0.07	100%	9%	6%	0.14	3.11
	90%	0.73	0.05	100%	6%	4%	0.12	2.54
	95%	0.84	0.03	100%	2%	2%	0.10	1.78
PSPNet [18]	75%	0.19	0.38	99%	89%	71%	0.09	4.87
	85%	0.32	0.28	98%	74%	55%	0.11	5.25
	90%	0.42	0.21	98%	60%	42%	0.13	5.30
	95%	0.60	0.11	98%	33%	22%	0.15	4.85
PSANet [19]	75%	0.10	0.44	99%	97%	79%	0.09	4.76
	85%	0.16	0.38	98%	94%	71%	0.10	5.20
	90%	0.20	0.32	98%	89%	64%	0.12	5.19
	95%	0.36	0.22	98%	70%	44%	0.14	5.07
DANet [7]	75%	0.30	0.37	99%	78%	65%	0.12	5.63
	85%	0.49	0.23	99%	57%	46%	0.14	5.79
	90%	0.64	0.16	99%	40%	30%	0.15	5.80
	95%	0.71	0.12	99%	29%	21%	0.13	3.95
DRNet [17]	75%	0.42	0.19	100%	35%	22%	0.18	5.40
	85%	0.55	0.11	100%	22%	13%	0.15	4.43
	90%	0.63	0.08	100%	15%	10%	0.14	3.84
	95%	0.77	0.05	100%	8%	5%	0.13	2.81
U-Net [13]	75%	0.12	0.20	96%	70%	44%	0.15	6.56
	85%	0.19	0.15	96%	52%	32%	0.19	6.81
	90%	0.25	0.13	96%	42%	25%	0.22	6.54
	95%	0.36	0.11	96%	27%	16%	0.23	5.73

Table 6: **Adaptive indirect local attacks** on Cityscapes. We compute the performance statistics for different sparsity levels of perturbation. By enforcing group sparsity, we can attack context-aware networks such as PSANet [19], PSPNet [18] and DANet [7] with higher success rates than for the baseline FCN [11].

Networks	Patch size $h \times w$ (area%)	mIoU	mASR	Norm of δ	
		$\text{mIoU}^{\ell}_{\mathbf{u}}$	$\text{mASR}^{\ell}_{\mathbf{u}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	51×102 ($\mathbf{1.0}\%$)	0.86	2%	0.30	25.36
	76×157 ($\mathbf{2.3}\%$)	0.78	4%	0.30	37.60
	102×204 ($\mathbf{4.0}\%$)	0.73	10%	0.30	51.80
	153×306 ($\mathbf{9.0}\%$)	0.58	18%	0.30	78.32
PSPNet [18]	51×102 ($\mathbf{1.0}\%$)	0.80	3%	0.30	25.52
	76×157 ($\mathbf{2.3}\%$)	0.63	10%	0.30	38.43
	102×204 ($\mathbf{4.0}\%$)	0.44	27%	0.30	50.32
	153×306 ($\mathbf{9.0}\%$)	0.09	84%	0.30	74.92
PSANet [19]	51×102 ($\mathbf{1.0}\%$)	0.41	38%	0.30	26.69
	76×157 ($\mathbf{2.3}\%$)	0.23	60%	0.30	38.60
	102×204 ($\mathbf{4.0}\%$)	0.14	71%	0.30	50.39
	153×306 ($\mathbf{9.0}\%$)	0.04	90%	0.30	78.02
DANet [7]	51×102 ($\mathbf{1.0}\%$)	0.79	4%	0.30	26.45
	76×157 ($\mathbf{2.3}\%$)	0.71	10%	0.30	37.24
	102×204 ($\mathbf{4.0}\%$)	0.65	15%	0.30	49.86
	153×306 ($\mathbf{9.0}\%$)	0.40	42%	0.30	74.60
DRNet [17]	51×102 ($\mathbf{1.0}\%$)	0.82	2%	0.30	26.28
	76×157 ($\mathbf{2.3}\%$)	0.77	7%	0.30	39.27
	102×204 ($\mathbf{4.0}\%$)	0.70	14%	0.30	52.23
	153×306 ($\mathbf{9.0}\%$)	0.55	28%	0.30	78.32
U-Net [13]	51×102 ($\mathbf{1.0}\%$)	0.32	26%	0.30	29.95
	76×157 ($\mathbf{2.3}\%$)	0.13	58%	0.30	44.42
	102×204 ($\mathbf{4.0}\%$)	0.06	76%	0.30	58.15
	153×306 ($\mathbf{9.0}\%$)	0.02	90%	0.30	86.06

Table 7: **Universal local attacks** on Cityscapes by tuning the patch size $h \times w$ (area%) on different networks. PSANet [19] and UNet [13] are highly sensitive to patch attacks even when the patch is 1% of image area. Note that the attack is untargeted and aimed to fool the entire scene by placing a fixed-size patch at the center of the image. We use ℓ_∞ based attacks with $\alpha = 0.001$ and $\epsilon = 0.3$.

Networks	Perturbation region	Fooling region	Norm of δ		Misclassified pixels %	Global AUROC	Local AUROC
			ℓ_∞	ℓ_2		SC [15] / Re-Syn [9] / **Ours**	**Ours**
FCN [11]	Global	Full	0.09	17.67	91%	**1.00 / 1.00** / 0.94	0.90
	FS	Dyn	0.001	0.83	1%	0.48 / 0.53 / **0.89**	0.80
			0.01	4.70	5%	0.54 / 0.67 / **1.00**	0.83
			0.10	15.12	9%	0.65 / 0.75 / **1.00**	0.83
			0.40	50.93	10%	0.93 / 0.76 / **1.00**	0.73
			0.02	0.58	2%	0.51 / 0.56 / **0.58**	0.83
			0.05	1.75	5%	0.55 / 0.67 / **0.82**	0.86
			0.08	2.58	6%	0.57/ 0.71 / **0.90**	0.87
	UP	Full	0.30	25.46	2%	0.70 / 0.55 / **0.88**	0.96
			0.30	37.60	4%	0.82 / 0.64 / **1.00**	0.94
			0.30	51.80	10%	0.90 / 0.75 / 1.00	0.94
			0.30	7.32	18%	0.99 / 0.94 / **1.00**	0.95
	AP	Dyn	0.15	4.04	3%	0.68 / 0.65 / **0.92**	0.88
			0.14	3.11	2%	0.61 / 0.57 / **0.87**	0.89
			0.12	2.54	1%	0.60 / 0.55 / **0.80**	0.90
			0.10	1.78	1%	0.60 / 0.52 / **0.73**	0.91

Table 8: **Attack detection** on Cityscapes with different perturbation settings on **FCN [11]**. We perform Mahalanobis-based attack detection in three settings, namely, Global: Global image perturbations; UP: Universal patch perturbations; FS: Full static class perturbations. We tune the noise level or patch size or sparsity level of the attack generation process to achieve different ranges of success rates. Note that SC [15] and Re-Syn [9] perform well only when large percentages of pixels are misclassified, while we outperform them by a large margin in all other settings.

Networks	Perturbation region	Fooling region	Norm of δ		Misclassified pixels %	Global AUROC	Local AUROC
			ℓ_∞	ℓ_2		SC [15] / Re-Syn [9] / **Ours**	**Ours**
PSPNet [18]	Global	Full	0.06	10.74	83%	0.90 / **1.00** / 0.99	0.81
	FS	Dyn	0.001	0.77	3%	0.49 / 0.56 / **1.00**	0.84
			0.01	3.10	14%	0.48 / 0.76 / **1.00**	0.90
			0.05	8.30	14%	0.52 / 0.77 / **1.00**	0.85
			0.20	37.99	14%	0.88 / 0.78 / **1.00**	0.88
			0.01	0.51	4%	0.50 / 0.59 / **1.00**	0.85
			0.02	1.28	12%	0.52 / 0.72 / **1.00**	0.87
			0.03	1.17	14%	0.52 / 0.73 / **1.00**	0.87
	UP	Full	0.30	25.52	3%	0.57 / 0.55 / **1.00**	0.93
			0.30	38.43	10%	0.62 / 0.70 / **1.00**	0.96
			0.30	50.32	27%	0.65 / 0.89 / **1.00**	0.96
			0.30	74.92	84%	0.87 / 1.00 / 1.00	0.97
	AP	Dyn	0.09	4.87	12%	0.65 / 0.82 / **0.99**	0.90
			0.11	5.25	10%	0.59 / 0.76 / **0.98**	0.82
			0.13	5.30	9%	0.56 / 0.72 / **0.99**	0.82
			0.15	4.85	5 %	0.55 / 0.69 / **1.00**	0.84

Table 9: **Attack detection** on Cityscapes with different perturbation settings on **PSPNet [18]**. We perform Mahalanobis-based attack detection in three settings, namely, Global: Global image perturbations; UP: Universal patch perturbations; FS: Full static class perturbations. We tune the noise level or patch size or sparsity level of the attack generation process to achieve different ranges of success rates. Note that SC [15] and Re-Syn [9] perform well only when large percentages of pixels are misclassified, while we outperform them by a large margin in all other settings.

Networks	Perturbation region	Fooling region	Norm of δ		Misclassified pixels %	Global AUROC	Local AUROC
			ℓ_∞	ℓ_2		SC [15] / Re-Syn [9] / **Ours**	**Ours**
PSANet [19]	Global	Full	0.04	8.26	93%	0.90 / **1.00** / 0.94	0.75
	FS	Dyn	0.001	0.72	4%	0.49 / 0.56 / **1.00**	0.88
			0.01	2.68	14%	0.48 / 0.77 / **1.00**	0.92
			0.05	8.10	14%	0.50 / 0.78 / **1.00**	0.89
			0.18	35.71	14%	0.87 / 0.78 / **1.00**	0.87
			0.01	0.39	4%	0.51 / 0.57 / **1.00**	0.88
			0.02	0.90	13%	0.49 / 0.73 / **1.00**	0.92
			0.03	1.44	14%	0.49 / 0.77 / **1.00**	0.92
	UP	Full	0.30	26.69	38%	0.60 / 1.00 / 1.00	0.99
			0.30	38.60	60%	0.62 / 1.00 / 1.00	0.98
			0.30	50.39	71%	0.69 / 1.00 / 1.00	0.97
			0.30	78.02	90%	0.85 / 1.00 / 1.00	0.98
	AP	Dyn	0.09	4.76	14%	0.54 / 0.85 / **1.00**	0.95
			0.10	5.20	14%	0.52 / 0.83 / **1.00**	0.94
			0.12	5.19	13%	0.54 / 0.81 / **1.00**	0.92
			0.14	5.07	10%	0.52 / 0.78 / **0.94**	0.91

Table 10: **Attack detection** on Cityscapes with different perturbation settings on **PSANet [19]**. We perform mahalanobis based attack detection in four settings namely Global:Global image perturbations, UP:Universal patch perturbations; FS : Full static class perturbations. . We tune the noise level or patch size or sparsity levels of attack generation process to achieve different range of success rate. As observed, SC [15] and Re-Syn [9] perform well only when large percentage of pixels are misclassified while we outperform them by a large margin in all other settings.

Networks	Perturbation region	Fooling region	Norm of δ ℓ_∞	Norm of δ ℓ_2	Misclassified pixels %	Global AUROC SC [15] / Re-Syn [9] / **Ours**	Local AUROC **Ours**
DANet [7]	Global	Full	0.06	12.55	82%	0.89 / **1.00** / **1.00**	0.68
	FS	Dyn	0.01	0.81	1%	0.50 / 0.51 / **0.64**	0.88
			0.01	3.90	14%	0.52 / 0.72 / **0.96**	0.86
			0.04	8.30	14%	0.56 / 0.74 / **0.99**	0.92
			0.15	31.71	14%	0.84 / 0.75 / **1.00**	0.94
			0.02	0.56	3%	0.50 / 0.54 / **0.67**	0.86
			0.03	1.32	9%	0.48 / 0.64 / **0.89**	0.86
			0.03	1.95	14%	0.50 / 0.70 / **0.87**	0.88
	UP	Full	0.30	26.45	4%	0.74 / 0.57 / **0.77**	0.89
			0.30	37.24	10%	0.80 / 0.64 / **0.92**	0.83
			0.30	49.86	15%	0.73 / 0.75 / **0.99**	0.87
			0.30	74.60	42%	0.88 / 0.92 / **1.00**	0.89
	AP	Dyn	0.12	5.63	12%	0.58 / 0.75 / **0.99**	0.82
			0.14	5.79	9%	0.54 / 0.68 / **0.99**	0.82
			0.15	5.80	6%	0.50 / 0.63 / **0.95**	0.81
			0.13	3.95	5%	0.51 / 0.58 / **0.85**	0.83

Table 11: **Attack detection** on Cityscapes with different perturbation settings on **DANet [7]**. We perform mahalanobis based attack detection in four settings namely Global:Global image perturbations, UP:Universal patch perturbations; FS : Full static class perturbations. . We tune the noise level or patch size or sparsity levels of attack generation process to achieve different range of success rate. As observed, SC [15] and Re-Syn [9] perform well only when large percentage of pixels are misclassified while we outperform them by a large margin in all other settings.

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.00 / 90%	0.02 / 81%	0.01 / 88%	0.57 / 10%
PSPNet [18]	0.25 / 43%	0.01 / 83%	0.09 / 73%	0.74 / 4%
PSANet [19]	0.22 / 47%	0.23 / 55%	0.00 / 92%	0.77 / 3%
DANet [7]	0.14 / 45%	0.50 / 12%	0.50 / 12%	0.02 / 81%

(a) ℓ_∞ untargeted direct attack

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.00 / 90%	0.09 / 67%	0.03 / 84%	0.72 7%
PSPNet [18]	0.20 / 12%	0.06 / 91%	0.16 / 28%	0.72 / 7%
PSANet [19]	0.11 / 31%	0.14 / 45%	0.01 / 91%	0.72 / 6%
DANet [7]	0.54 / 1%	0.59 / 6%	0.60 / 5%	0.06 / 91%

(b) ℓ_∞ targeted direct attack

Table 12: **Transferability of direct attacks.** On the left, we show the transfer rate of *untargeted* attacks perturbing the entire static region to misclassify the dynamic regions. On the right, we show the transfer rate of *targeted* attacks perturbing the entire static region to misclassify the dynamic regions to their nearest static class. We set $\epsilon = 0.1$.

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.29 / 29%	0.30 / 13%	0.19 / 24%	0.77 / 6%
PSPNet [18]	0.42 / 3%	0.05 / 85%	0.35 / 5%	0.83 / 5%
PSANet [19]	0.34 % 5%	0.38 / 5%	0.04 / 85%	0.84 / 5%
DANet [7]	0.68 / 4%	0.68 / 4%	0.71 / 4%	0.11 / 79%

(a) ℓ_∞ indirect local attack

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.98 / 0%	0.75 / 2%	0.73 / 2%	0.86 / 6%
PSPNet [18]	0.97 / 1%	0.24 / 60%	0.50 / 3%	0.85 / 5%
PSANet [19]	0.98 / 1%	0.63 / 2%	0.13 / 65%	0.86 / 5%
DANet [7]	0.99 / 4%	0.82 / 5%	0.84 / 4%	0.48 / 43%

(b) ℓ_∞ attack with boundary $d = 50$ pixels

Table 13: **Transferability of indirect local attacks.** On the left, we show the transfer rate of attacks perturbing the complete static region to misclassify the dynamic regions. On the right, we show the transfer rate of attacks perturbing pixels that are at least $d = 100$ pixels away from the object boundary.

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.52 / 12%	0.52 / 5%	0.37 / 8%	0.85 / 6%
PSPNet [18]	0.69 / 1%	0.25 / 63%	0.50 / 3%	0.85 / 5%
PSANet [19]	0.69 / 1%	0.62 / 2%	0.10 / 79%	0.87 / 5%
DANet [7]	0.85 / 4%	0.80 / 4%	0.82 / 4%	0.32 / 62%

(a) adaptive indirect local attack with $S = 75\%$

Train / Eval	FCN [11]	PSPNet [18]	PSANet [19]	DANet [7]
FCN [11]	0.58 / 18%	0.61 / 10%	0.60 / 13%	0.74 / 5%
PSPNet [18]	0.69 / 8%	0.08 / 86%	0.52 / 21%	0.73 / 7%
PSANet [19]	0.66 / 8%	0.64 / 11%	0.04 / 90%	0.75 / 6%
DANet [7]	0.72 / 8%	0.67 / 8%	0.68 / 8%	0.39 / 42%

(b) ℓ_∞ universal local attack with patch size at the center of image

Table 14: **Transferability of adaptive indirect local attacks**. On the left, we report the transfer rate of adaptive local attacks with $S = 75\%$. On the right, we report the transfer rate of universal local attacks with patch size $153 \times 153 (\mathbf{9.0}\%)$.

Network	*person*	*rider*	*car*	*truck*	*bus*	*train*	*motorcycle*	*bicycle*	**All**
FCN [11]	1.00 / 0%	1.00 / 0%	1.00 / 0%	1.00 / 0%	1.00 / 0%	1.00 / 0%	0.98 / 1%	0.90 / 3%	0.97 / 2%
PSPNet [18]	0.87 / 11%	0.98 / 6%	0.94 /2%	0.63 / 30%	0.91 / 6%	0.92 / 3%	0.66 / 28%	0.53 / 2%	0.78 / 2%
PSANet [19]	0.67 / 23%	0.60 / 24%	0.78 / 7%	0.70 / 22%	0.66 / 22%	0.10 / 21%	0.50 /39%	0.94 / 5%	0.39 / 4%
DANet [7]	0.43 / 39%	0.86 /12%	0.90 / 7%	0.50 / 25%	0.29 / 69%	0.75 / 14%	0.28 / 41%	0.67 / 22%	0.87 / 6%

Table 15: Success rate of universal, class-specific targeted, fixed-size patches for Cityscapes similar to those in [14, 10]. We place a $102 \times 102 (\mathbf{4.0}\%)$ patch at the top left of the image. We observe that such patches at fixed location perform poorly for targeted attacks aiming to misclassify pixels to their nearest static label.

Networks	Sparsity	mIoU		mASR			Norm of δ	
		$\text{mIoU}^{f}_{\text{u}}$	$\text{mIoU}^{f}_{\text{t}}$	$\text{mASR}^{p}_{\text{u}}$	$\text{mASR}^{f}_{\text{u}}$	$\text{mASR}^{f}_{\text{t}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	75%	0.50	0.32	100%	35%	32%	0.14	2.40
	85%	0.58	0.27	100%	30%	27%	0.13	2.15
	90%	0.66	0.22	100%	24%	22%	0.12	1.91
	95%	0.80	0.12	100%	13%	13%	0.11	1.37
PSPNet [18]	75%	0.22	0.79	99%	80%	79%	0.07	2.15
	85%	0.21	0.81	98%	83%	81%	0.08	2.40
	90%	0.22	0.80	98%	81%	79%	0.10	2.71
	95%	0.39	0.60	99%	63%	60%	0.15	3.07
PSANet [19]	75%	0.29	0.68	99%	70%	68%	0.07	1.77
	85%	0.22	0.78	98%	79%	78%	0.07	1.93
	90%	0.20	0.80	98%	82%	80%	0.08	2.21
	95%	0.30	0.69	99%	70%	68%	0.13	2.81

Table 16: **Adaptive indirect local attacks** on PASCAL VOC. We observe that PSANet [19] is more vulnerable to local adaptive attacks than FCN [11].

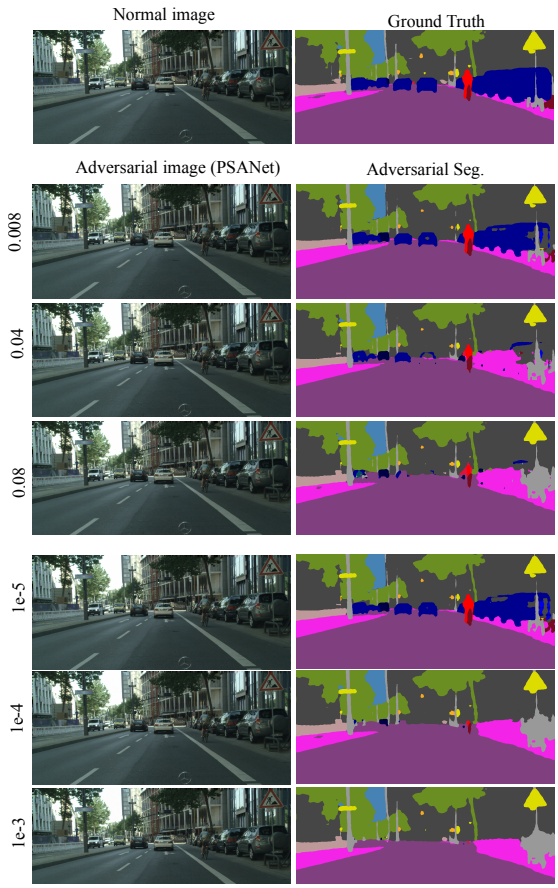

Fig. 1: **Indirect attacks** on Cityscapes to fool dynamic classes while perturbing complete static ones using ℓ_2 and ℓ_∞ attacks. We use $\alpha = \{8\text{e-}3, 4\text{e-}2, 8\text{e-}2\}$ for ℓ_2 attacks and α=$\{1\text{e-}5, 1\text{e-}4, 1\text{e-}3, 5\text{e-}3\}$ for ℓ_∞ attacks. We observe that PGD [1] is effective at computing an imperceptible perturbations for different ranges of step-size α.

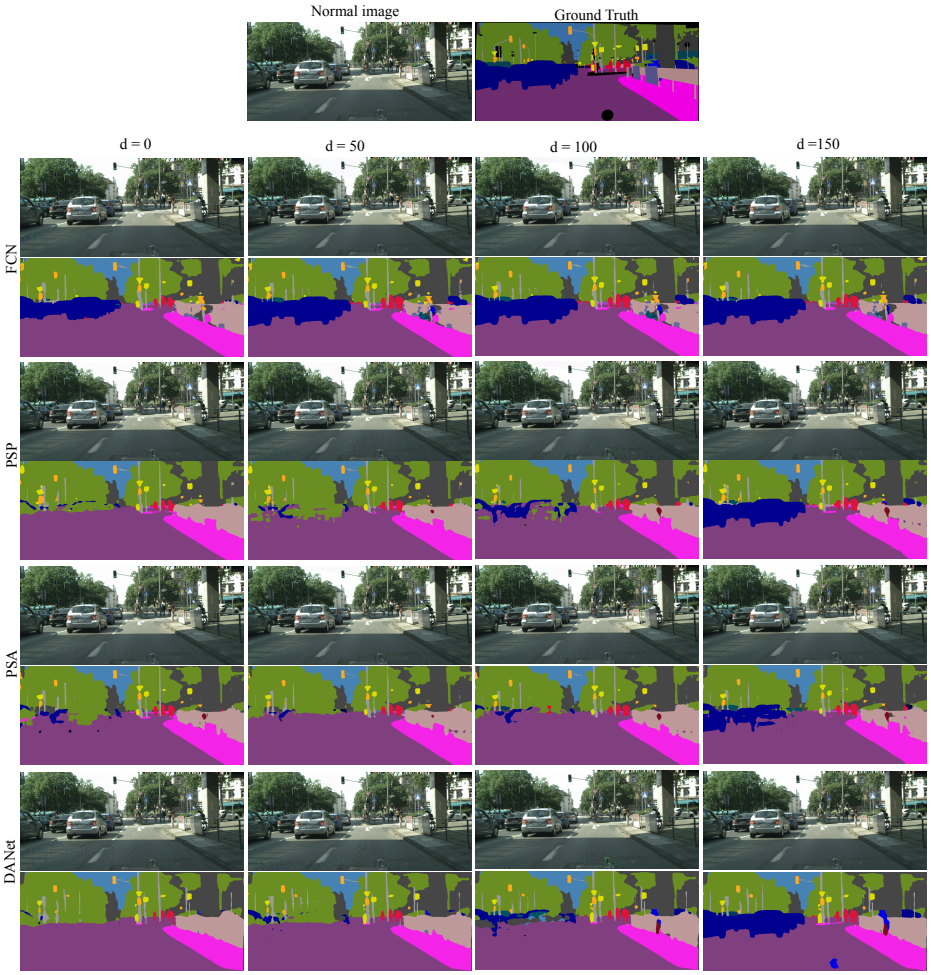

Fig. 2: **Indirect local attack** on different networks with perturbations at least d pixels away from any dynamic class. In most cases, FCN [11] is not affected by indirect attacks, while PSANet [19], PSPNet [18] and DANet [7] are affected due to their larger contextual dependencies for prediction.

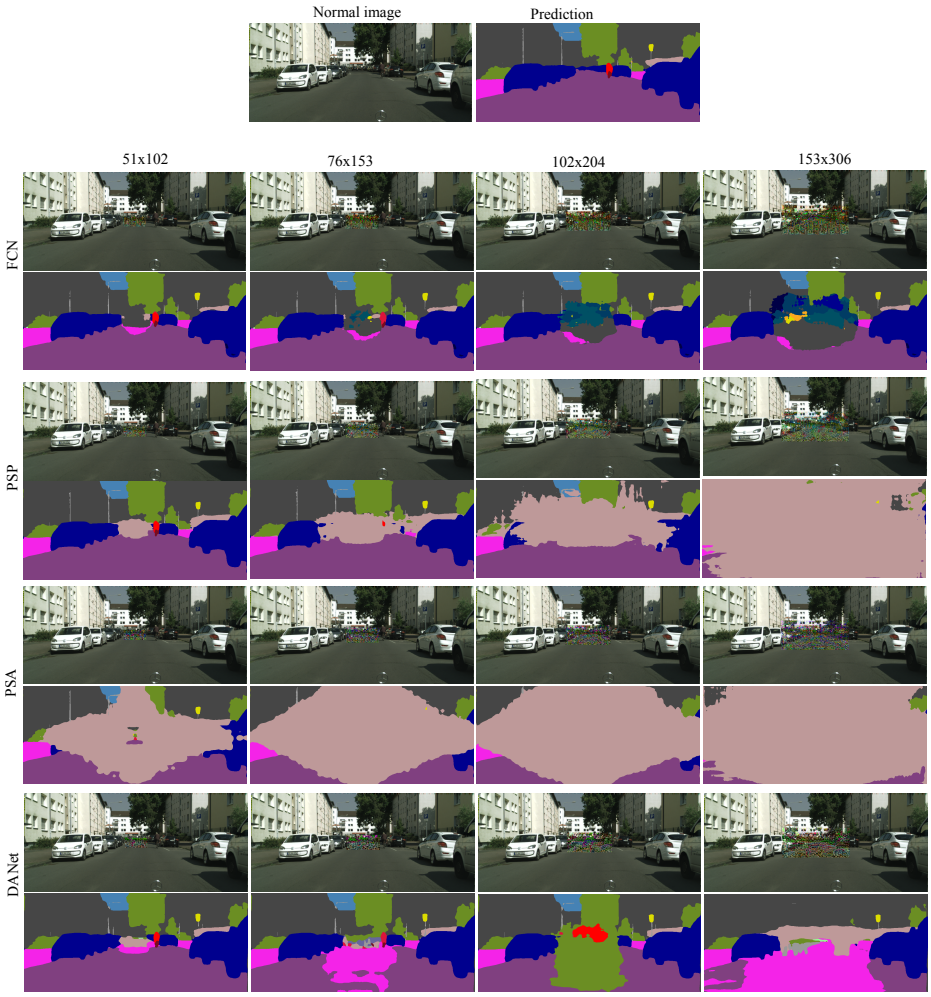

Fig. 3: **Universal local attacks** on segmentation networks. The degradation in FCN [11] is limited to the attacked area, whereas for context-aware networks, such as PSP-Net [18], PSANet [19], DANet [7], it extends to far-away regions.

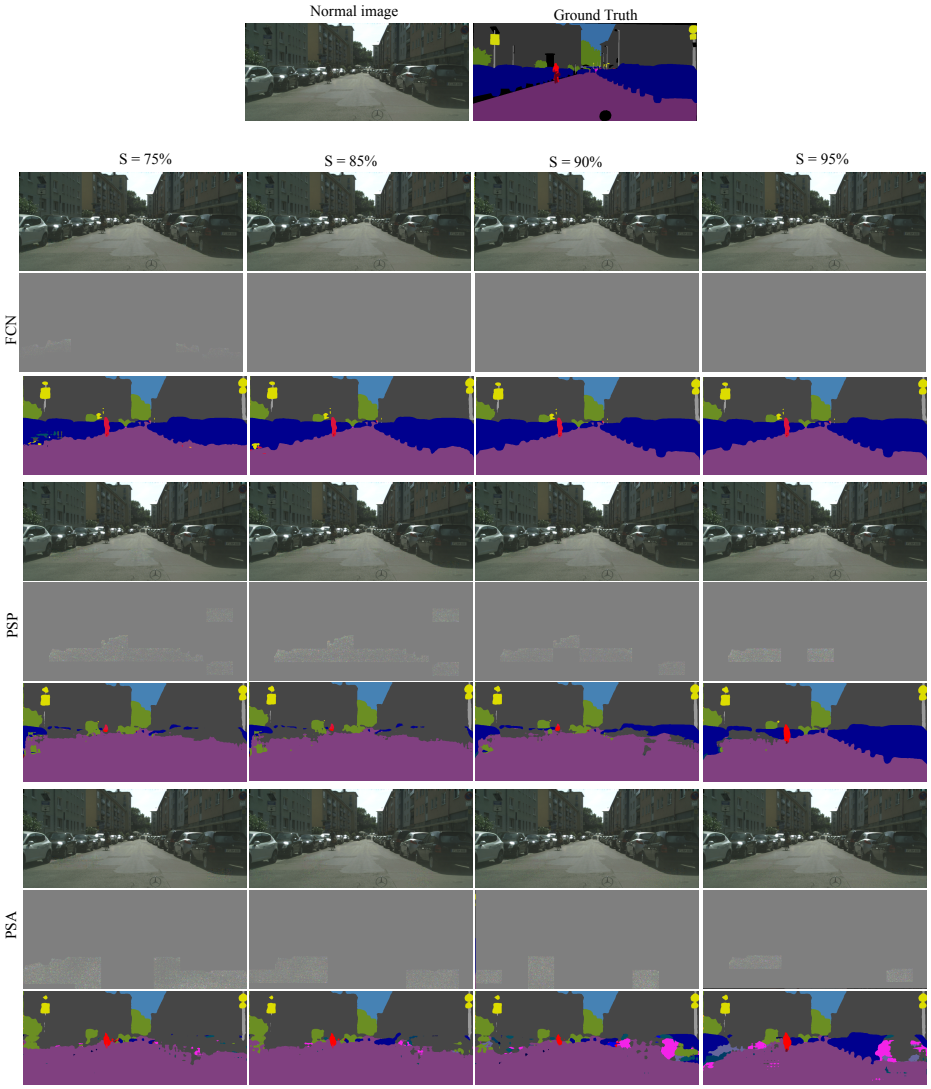

Fig. 4: **Adaptive indirect local attacks on Cityscapes with different networks by tuning the sparsity levels**. We observe that PSPNet [18] and PSANet [19] are vulnerable to adaptive indirect local attacks even with perturbations with high levels of sparsity, while FCN [11] is the least affected.

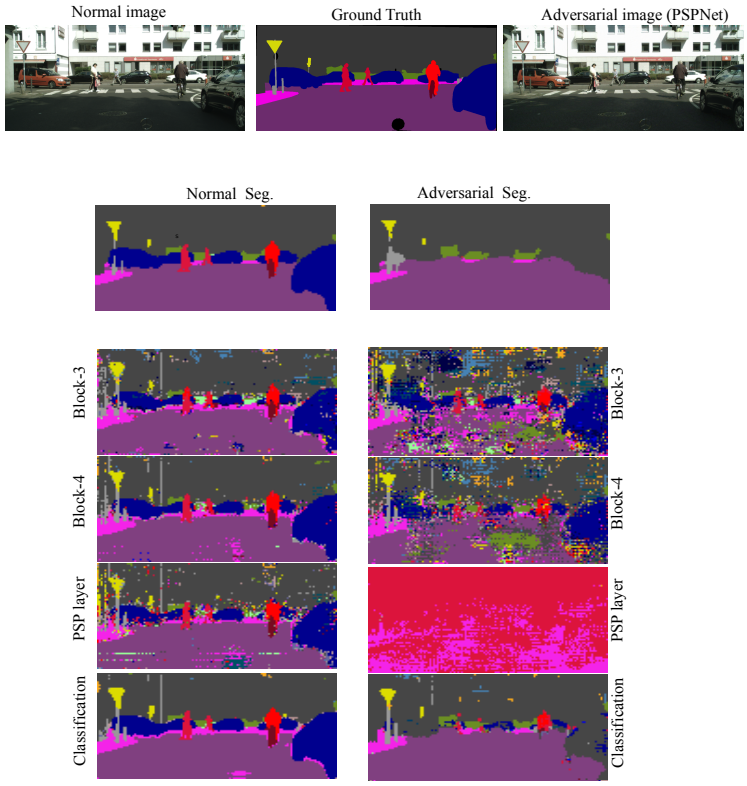

Fig. 5: **Visualizing internal subspaces of normal and adversarial samples of Cityscapes with PSPNet [18]**. For each spatial location of the extracted feature map at layer ℓ, we assign the label of the nearest pre-trained class-conditional distribution computed using the Mahalanobis distance. As shown in the figure, the nearest cluster label looks almost the same as the predicted label map for clean samples. By contrast, for adversarial samples, the nearest cluster moves towards the predicted adversarial label in the final layers. Furthermore, in the PSP context layer, for adversarial samples, the nearest conditional distribution values are completely erroneous and far away from the normal cluster assignments.

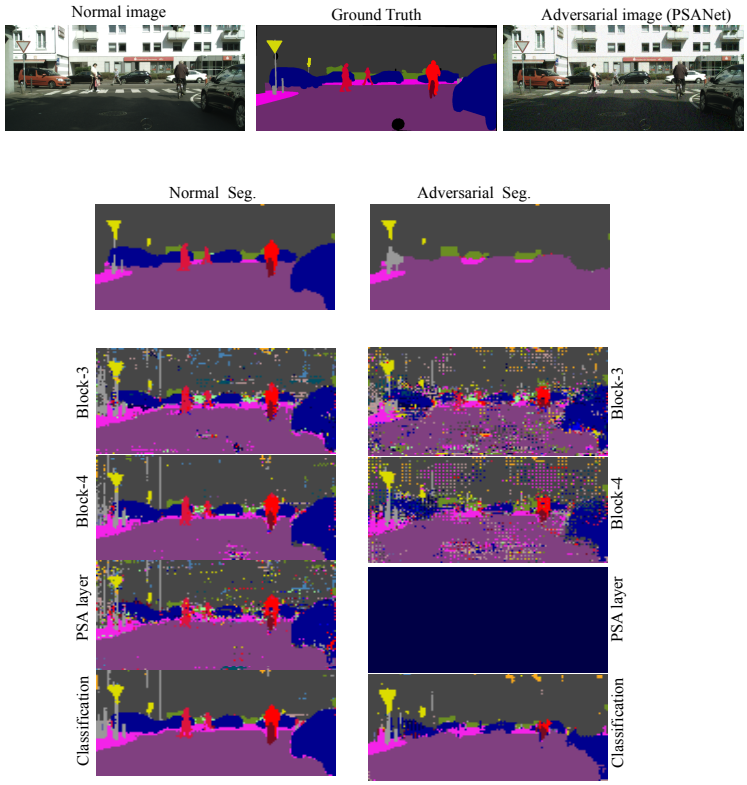

Fig. 6: **Visualizing internal subspaces of normal and adversarial samples of Cityscapes with PSANet [19]**. For each spatial location of the extracted feature map at layer ℓ, we assign the label of the nearest pre-trained class-conditional distribution computed using the Mahalanobis distance. As shown in the figure, the nearest cluster label looks almost the same as the predicted label map for clean samples. By contrast, for adversarial samples, the nearest cluster moves towards the predicted adversarial label in the final layers. Furthermore, in the PSA context layer, for adversarial samples, the nearest conditional distribution values are completely erroneous and far away from the normal cluster assignments.

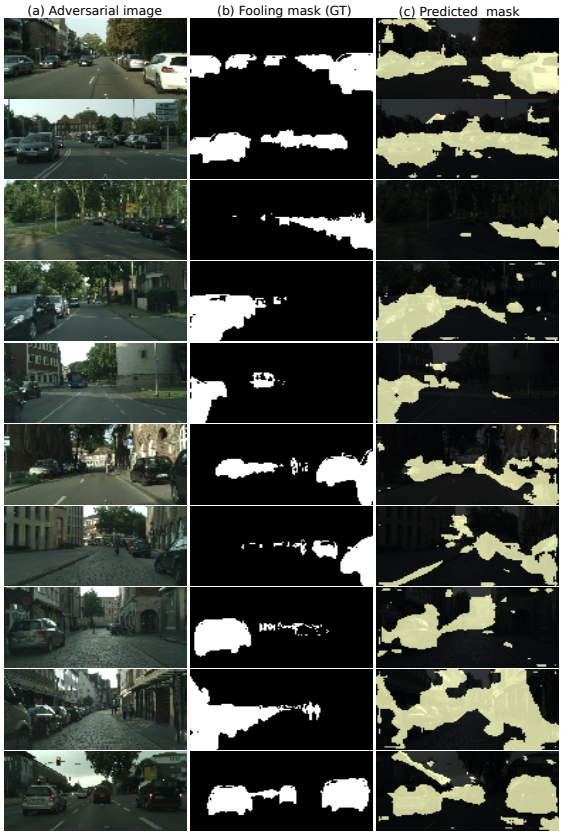

Fig. 7: **Visualization of attack detection** at pixel level by adaptive indirect local attacks on Cityscapes with PSANet [19]. The first column shows the adversarial image, the second column shows the ground-truth fooling locations and the third column the predicted fooling positions.

Networks	Patch size $h \times w$ (area%)	mIoU	mASR	Norm of δ	
		$\text{mIoU}^{l}_{\mathbf{u}}$	$\text{mASR}^{l}_{\mathbf{u}}$	ℓ_∞-norm	ℓ_2-norm
FCN [11]	51×51 ($\mathbf{1.0}\%$)	0.77	3%	0.30	19.84
	76×76 ($\mathbf{2.3}\%$)	0.69	6%	0.30	29.64
	102×102 ($\mathbf{4.0}\%$)	0.63	10%	0.30	38.81
	153×153 ($\mathbf{9.0}\%$)	0.51	19%	0.30	59.09
PSPNet [18]	51×51 ($\mathbf{1.0}\%$)	0.82	5%	0.30	20.20
	76×76 ($\mathbf{2.3}\%$)	0.75	9%	0.30	29.57
	102×102 ($\mathbf{4.0}\%$)	0.62	16%	0.30	38.62
	153×153 ($\mathbf{9.0}\%$)	0.39	41%	0.30	57.28
PSANet [19]	51×51 ($\mathbf{1.0}\%$)	0.83	4%	0.30	20.24
	76×76 ($\mathbf{2.3}\%$)	0.75	8%	0.30	29.40
	102×102 ($\mathbf{4.0}\%$)	0.56	28%	0.30	38.63
	153×153 ($\mathbf{9.0}\%$)	0.35	56%	0.30	57.6

Table 17: **Universal local attacks** on PASCAL VOC by tuning the patch size $h \times w$ (area%) on different networks. PSANet [19] and UNet [13] are highly sensitive to patch attacks even when the patch is 1% of the image area. Note that the attack is untargeted and aimed to fool the entire scene by placing a fixed size patch at the center of the image. We use ℓ_∞ based attacks with $\alpha = 0.001$ and $\epsilon = 0.3$.

Fig. 8: **Adaptive indirect local attacks** on PASCAL VOC with PSANet [19]. We show an adversarial image **(a)** perturbed with an imperceptible noise **(b)** at local background regions, which forces the foreground regions in the normal segmentation map **(c)** to be misclassified as background classes, as shown in **(d)**.

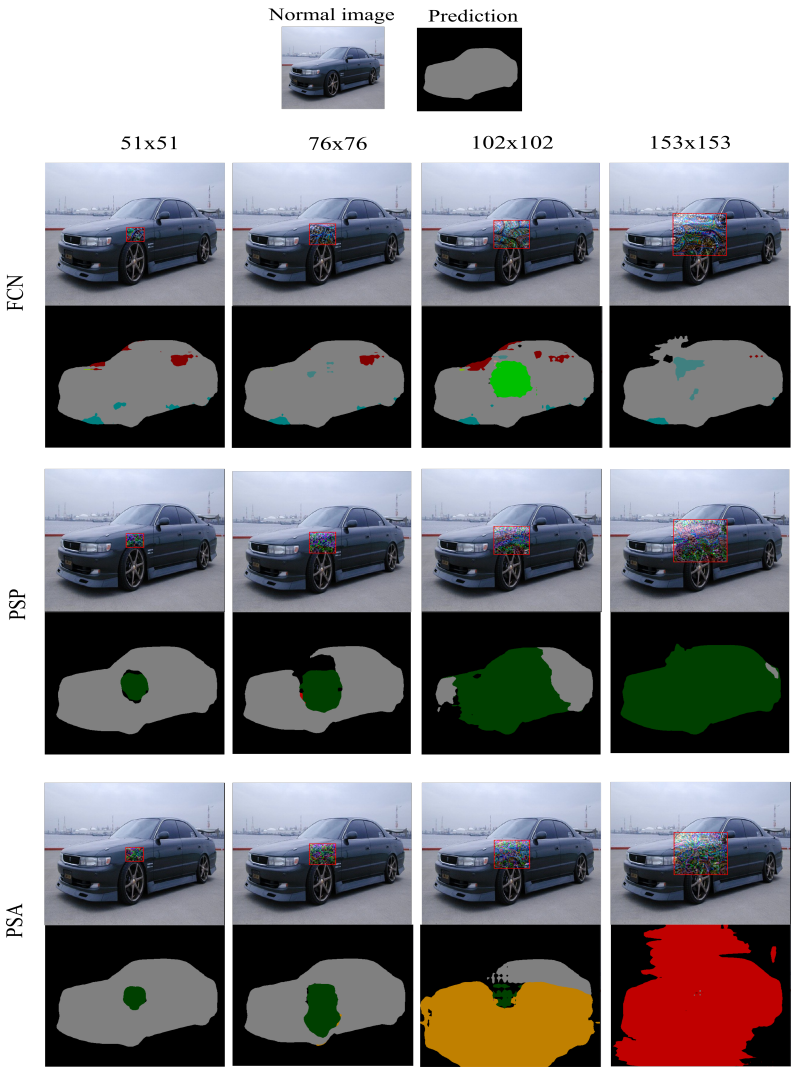

Fig. 9: **Universal local attacks** on PASCAL VOC segmentation networks. The degradation in FCN [11] is limited to the attacked area, whereas, for context-aware networks, such as PSPNet [18] and PSANet [19] it extends to far-away regions.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
2. Brown, T.B., Mané, D.: Aurko roy, martín abadi, and justin gilmer. Adversarial patch. CoRR, abs/1712.09665 (2017)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
4. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
5. Ding, G.W., Wang, L., Jin, X.: Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623 (2019)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge 2007 (voc2007) results (2007)
7. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)
8. Hendrik Metzen, J., Chaithanya Kumar, M., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2755–2764 (2017)
9. Lis, K., Nakka, K., Salzmann, M., Fua, P.: Detecting the unexpected via image resynthesis. arXiv preprint arXiv:1904.07595 (2019)
10. Liu, X., Yang, H., Liu, Z., Song, L., Li, H., Chen, Y.: Dpatch: An adversarial patch attack on object detectors. arXiv preprint arXiv:1806.02299 (2018)
11. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
12. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
14. Saha, A., Subramanya, A., Patil, K., Pirsiavash, H.: Adversarial patches exploiting contextual reasoning in object detection. arXiv preprint arXiv:1910.00068 (2019)
15. Xiao, C., Deng, R., Li, B., Yu, F., Liu, M., Song, D.: Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 217–234 (2018)
16. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1369–1378 (2017)
17. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 472–480 (2017)
18. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
19. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 267–283 (2018)