Project report - COS 598B
Physical adversarial examples for semantic image segmentation

Vikash Sehwag
Princeton University
vvikash@princeton.edu

Abstract

In this project, we work on the generation of physical adversarial examples for semantic image segmentation. Adversarial examples generation has gained a wide interest due to its success for most of the state of deep learning models. However, much of this work focuses on image classification in limited settings. In this project, we extended these methods to semantic segmentation with further relaxing some constrained to resemble real word attack models.

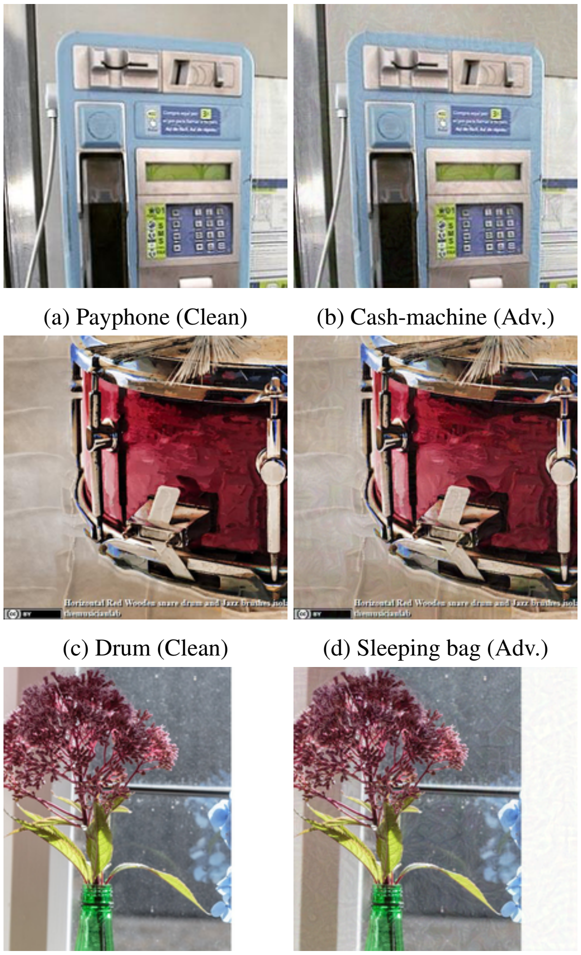
Initially, we provide a brief background on adversarial examples and further discuss the necessity of generating physical adversarial examples. In next section, we discuss how to generate robust adversarial examples and present the results with state of the art networks. Next, we discuss the limitation of the adversarial attacks for semantic segmentation models and present insight to this behavior by studying the effective receptive field of these networks. Finally, we present some examples of physical adversarial examples for these networks and discuss future direction given the limited adversarial success in this attack model.

1. Introduction

Recent advances in computer vision using deep learning methods have led to wide adoption of the neural network based approaches. However, the non-convex nature of neural network optimization poses a difficulty in understating the failure cases of these systems. One of the prominent examples from this class is adversarial examples. Given a images which will be classified correctly by a neural network, it is possible to add a very small amount of perturbation (ϵ) such that the resulted image will be misclassifed.

Given a neural network with network parameters (θ) and input(x), the output class for x can be written as $C(x)$. Therefore an adversarial example (x') for given input (x) can be define as,

$$x' = x + \epsilon \quad \text{s.t.} \quad C(x') \neq C(x), \quad ||\epsilon||_p \leq \epsilon_{max}$$



(a) Payphone (Clean) (b) Cash-machine (Adv.)

(c) Drum (Clean) (d) Sleeping bag (Adv.)

(e) Vase (Clean) (f) Mop (Adv.)

Figure 1: Clean and adversarial (Adv.) image for a resnet-50 network trained on Imagenet dataset. The adversarial image is obtained by adding a small perturbation to the clean image.

However, due to non-convexity of $C(x)$ a closed form solution of above optimization problem is highly difficult to obtain. Current state of the art attack algorithm for generating adversarial examples uses gradient based methods. Given a loss function $L(x, \theta, y)$ (x-input, y-target label, θ-model parameters), it simply add the perturbation such that this loss will decrease. As decrease in loss function (eg., cross-entropy loss in classification) depicts that the model is predicting target class instead of the true one.

1.1. Single step vs iterative attack algorithms

In single step attack algorithm it simply requires calculating a single gradient of loss function. The fast gradient sign methods (FGSM) which is a single step attack methods generates adversarial examples using following equation,

$$x' = x + \epsilon * sign(\nabla L(x, \theta, y))$$

. In iterative gradient attacks, we add a small perturbation ($\approx \frac{eps_{max}}{no.\ of\ iterations}$) to input in each iteration. As due to highly non-convex nature of loss function, iterative attacks are more likely to find the local maximum, thus perform much better. We refer the reader to [6] for a detailed discussion of these attack methods. In this work we will primarily use FGSM projected gradient descent [8] with iterative gradient attacks.

Fig. 1 demonstrate this effect where the adversarial examples are generated using Iterative gradient steps [8] with constraint $||\epsilon_{max}||_\infty \leq 16$ (while pixel values ranges 0. . .255). Though exact confidence not reported, it is generally greater than 95% for all adversarial examples. Note that adversarial examples can be easily generated with perturbation even much smaller perturbations. However, as the defense methods ([12, 8]) are also improving along with (work well mostly with small perturbations only), we consider ϵ =16.0 for these examples. A detailed discussion of these methods for image classification can be found in [7, 6].

1.2. Targeted vs Non-targeted attacks

It can be observed that the adversarial examples generated in Fig.1 has output class a bit similar in some features to the original class. For examples, the adversarial example for payphone is a cash-machine which at least got some similar features to the pay-phone. This method is termed non-directed attacks, where we simply decrease the confidence of loss function in the correct class. This problem arises poor nature of the optimization problem for generating these non-targeted adversarial examples. As the attack algorithms use local gradients, which generally leads to switching the output class to a visually similar one. The solution is to either improve optimization algorithm to find the global maximum in the constrained space for perturbation or update the optimization problem. The second solution which is the easier to implement is termed as targeted attack algorithms. Fig. 2 shows the targeted adversarial examples in a similar adversarial model as previous (fig. 1). These can be simply obtained by minimizing the loss function for making the targeted class as the predicted class.

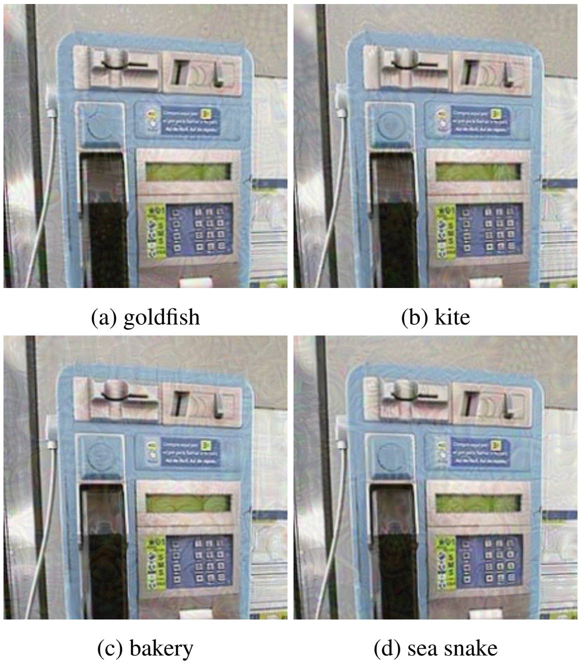

(a) goldfish (b) kite (c) bakery (d) sea snake

Figure 2: Targeted adversarial examples with corresponding predicted labels from a resnet-50 network. The target label is selected randomly from the imagnet data labels.

In this work we move forward from image classification and specifically focus on semantic image segmentation. Similar to classification, the adversary can corrupt the output segmentation by adding a small adversarial perturbation. Fig. 3 shows this effect where the adversary has control over the optimization problem, thus output segmentation of adversarial examples. Depending on the formulation of optimization problem i.e, the target label y in targetted attacks, the adversary can corrupt all output segmentation of all/selected pixels randomly or to a selected label.

1.3. Security model

However, for all these attacks we have assumed that the adversary has access to the neural network, which allows the adversary to calculate gradient and further generate adv. examples. This security model is called **open box** attack model. Given, the extent of security model it may not be possible for the adversary to access this information. An attack model where the adversary has access to neural network inputs and output prediction only is termed as **black box** attack model. It has been generally observed that adversarial examples generated for one model tend to be mis-

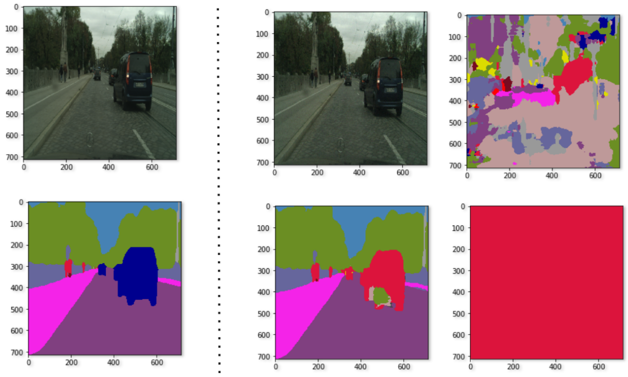


Figure 3: Output segmentation of clean and adversarial images. In left the original image and output segmentation is placed. In right the a adversarial images is showed with corresponding output segmentation. By adding different perturbation in the input image, adversary can easily influence output of selected pixels in a selected way.

classified by another model also. Adversarial examples reported in Fig. 2 are also adversarial for a vgg-16 network where they are mostly classified as cash-machine or pay-phone with a very low confidence. We evaluate the transferability of adversarial examples for segmentation networks in section 3.2.

2. Experimental setup

In this work we primarily work with Cityscape dataset [4] for semantic segmentation. We consider Pspnet [14] and DeeplabV3+ [3] which are state of the art approaches, for segmentation. We use the pre-trained model released in the official version, with tensorflow and keras as the learning framework (we obtain TF/keras compatible models/conversion-tools from 1,2).

To generate adversarial examples we use projected gradient descent with 100 iterations with Adam optimizer. For Cityscape, we consider 20 random images from the test set (fig. 15) and primarily test adversarial robustness with them. The primary reason to limit to 20 images is significant time overhead of generating adversarial example for each image. Further, if not specified, it should be assumed that the reported results are obtained with Pspnet.

3. Robust Adversarial examples

Though previous works [1, 13] have already demonstrated some of the basic attack (mostly non-targeted attacks) on semantic segmentation, all of them add noise to the images before predicting the segmentation mask. However, this attack model is unrealistic in the real world because it requires that the adversary add perturbation to the whole environment. For example, assume that we want to incorrectly predict the mask for all traffic signs for self-driving cars which used semantic segmentation in the first stage i.e, perception module. To fool such systems, an adversary needs to account for different position, angles, size of the signs in a given captured frame from the camera because they will vary depending on the distance of the car from them.

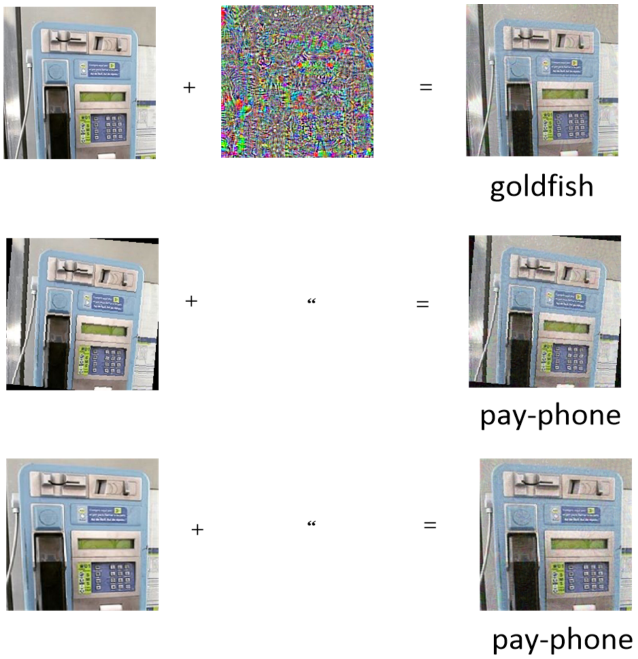
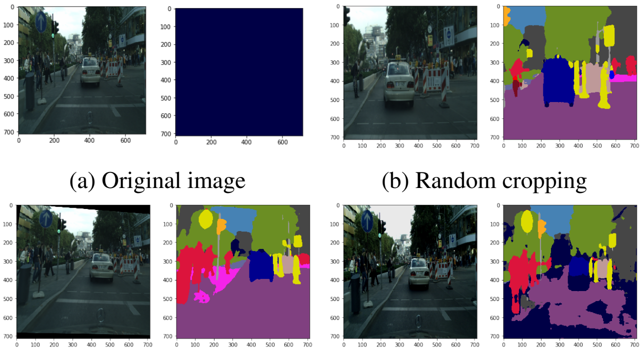


Figure 4: Poor transferability of adversarial perturbation between different images. Initially, an adversarial perturbation is added to the original image of a pay-phone to misclassify it as goldfish. However, as we rotate or crop the image, it is no longer classified as goldfish.

In a threat model, where the adversary has the access to vision pipeline of the concerned system, it is valid to assume that the adversary can add perturbation to the whole image. This threat model poses a significant risk given its relevance in the domain as medical image analysis, static vision APIs (such as Calrifai, Google cloud vision api). To investigate the complexity of robust adversarial examples for semantic segmentation, we first start with this threat model.

The primary constraint for the adversary in this threat model is to add only non-perceivable perturbation. The is realized by bounding the L_p norm of perturbation. It should be noted that bounds on L_p norm of perturbation is neither necessary nor sufficient condition for the non-perceivability of added perturbation. We refer the interested reader to [10] for detailed discussion. In this work we consider the bounds on L_1 norm for adversarial perturbations.



(a) Original image (b) Random cropping

(c) Affine transformation (d) Contrast and gamma variation

Figure 5: Limitation of non-robust adversarial examples in presence of image transformations. Similar to classification, the adversarial nature of added perturbations diminishes, as soon as we apply an input transformation.

3.1. Transformation Invariance

Previously, we assume that given an image the adversary can solve an optimization problem to generate the corresponding adversarial example. To be termed as a robust adversarial example the added perturbation should hold across different transformation. Fig. 5 demonstrate the limitation of adversarial examples generated for a given image to different transformations. Initially, we generate an adversarial example for a given image from cityscape dataset such that all pixels will be classified as Truck. Fig. 5a shows the success of the adversary where all pixels are classified as Truck. However, as we apply random cropping, perspective transformation or random contrast changes, the images no longer retain it's adversarial nature.

Note that this property of adversarial perturbation is not unique to segmentation. Fig. 4 shows similar results for image classification, where rotated or cropped images will no longer be adversarial. This behavior for classification has been studied in some of the previous works ([2, 11]), however, it's not studied for segmentation yet.

The lack of robustness in the adversarial examples can be evident from the formulated optimization problem. As the optimization problem doesn't account any information for such transformation, it is not surprising why the resulted solution breaks down when such transformations are applied.

To generated robust adversarial examples, adversary need to account the transformations to solve the fundamental optimization problem in first place [2]. This leads to the redefining the optimization problem as mentioned below.

$$\underset{x'}{arg\ min}\ \frac{1}{n(T)} \sum_{T} (c * L(x', y) + ||x' - x||_1)$$

This formulation allows adversary to minimize loss over a transformation set T, whre $n(T)$ is number of transformation applied.

In this work we consider the following transformations.

- **Scale Invariance** We randomly change the scale of the input image by $\pm 10\%$ to simulate this behavior. It allows the adversarial examples robust to change in scale of an object in the scene.

- **Affine transformation** To account for the different change in perspectives, we apply a random affine transformation for each input image.

- **Contrast invariance** we change the contrast and gamma for each image randomly to simulate the effect of different environmental condition.

With the expectation over transformation, we hope to capture a large distribution of possible transformation in the real world. In turn, it will result in the adversarial image will likely be adversarial i.e, incorrect output segmentation, even if account to a real-world test. However, it should be noted that this is just a hypothetical case, as it's not possible to add perturbation in the whole environment (e.g, how to add perturbation in the sky). As discussed in Section 5, the adversary can select a few physical objects on which adversarial stickers can be placed.

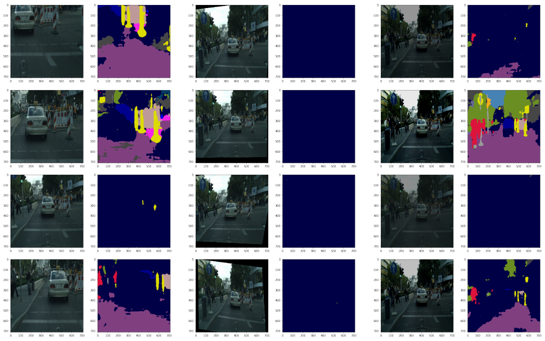
Fig.6, 7 shows the output results, after including input transformations in optimization problem, for Pspnet and DeeplabV3+ respectively. It can be noted that the generated adversarial examples are adversarial to included transformations. To refute any claim that the transformation itself may have degraded network output, we show output for original image with transformation in fig. 8 (output with Gaussian noise in fig 16). It shows that the degradation of output segmentation is only contributed by the added perturbation.

3.2. Black-box attacks

Previous work on adversarial attacks on image classification have shown transferability of adversarial examples between different networks. Here, we test this hypothesis by evaluating output segmentation of given network on adversarial examples generated from another. Table 2 shows the per-pixel accuracy for adversarial examples generated from the alternate model. It shows a poor transferability of adversarial examples between these two different models. Fig. 9 shows the output segmentation for some of these black box adversarial examples.

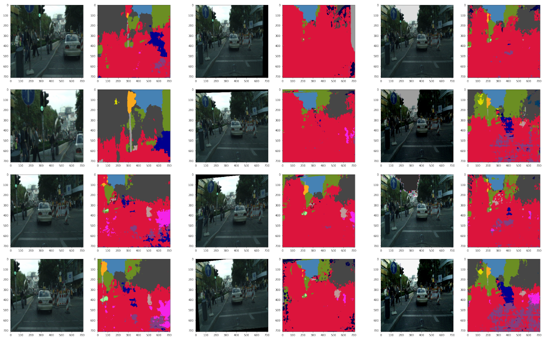
4. Receptive field

State of the art image segmentation networks such as DeeplabV3+, Pspnet uses resnet-50/101 as their backbone architecture. Theoretical receptive field for both resnet-50



(a) Random cropping (b) Affine transformation (c) Contrast and gamma variation

Figure 6: Robust adversarial examples for Pspnet. It can be noted that despite input transformation, the adversarial examples output are corrupted to a significant fraction.



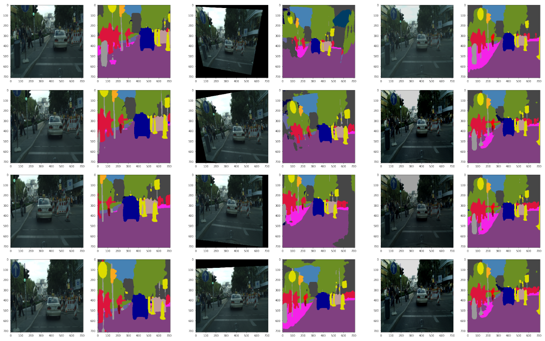
(a) Random cropping (b) Affine transformation (c) Contrast and gamma variation

Figure 7: Robust adversarial examples for DeeplabV3+. It can be noted that despite input transformation, the adversarial examples output are corrupted to a significant fraction.

and resnet-101 is even comparable to input size (which is 713×713 in this work). The receptive field size further increases with the addition of modules to incorporate global information in segmentation networks. This motivates us to believe that the changes in single part of the image should be able to have an effect output segmentation for all pixels. As explained later, it turns out that the effect is limited to only local output pixels, due to poor gradients for non-local pixels.

4.1. From Imagenet to Cityscape - a tale of effective receptive field

Though neglected thus far by the previous works on adversarial attacks and defenses, there exists an important relationship between the receptive field size and adversarial strength for the adversary. Fig 10 demonstrate the impact of adversarial perturbation when the adversary is limited to add it only to a pre-defined mask. Fig 10a demonstrate the impact of perturbation when it is added to pixels labeled as a car. Fig. 10b refer a more plausible model in the physical world (discussed in detail in section 5), where the adversarial perturbation is limited to space at the back of the car which can be thought of sticking a poster carrying adver-



(a) Random cropping (b) Affine transformation (c) Contrast and gamma variation

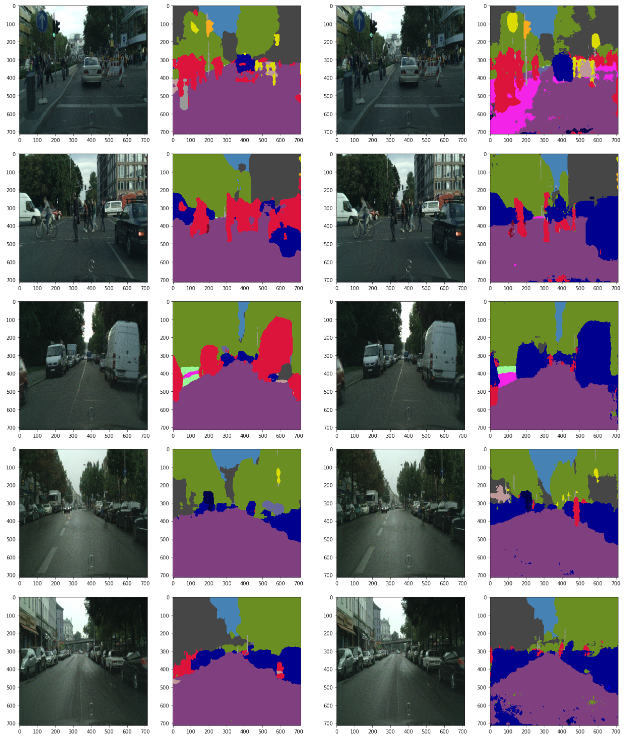
Figure 8: Output segmentation of original images with different transformation. It shows that the without the adversarial perturbation the networks output is not degraded by the transformation.

Table 1: Mean L-1 norm and per-pixel accuracy of the generated adversarial examples for the 20 images from the Cityscape test dataset. Lower L-1 norm refers to the imperceptibility of generated adversarial examples. Further, per-pixel accuracy measures the degradation of output segmentation for given images. As the ground truth segmentation for Cityscape test set is not released, we measure the accuracy w.r.t. model output for the original image. It should be noted the despite having a high L-1 norm the per-pixel accuracy of adversarial examples of DeeplabV3+ is high. We study this effect late in section 4.1.1.

	Pspnet	DeeplabV3+
L-1 norm (mean)	2.23	1.89
per-pixel accuracy($\%$)	3.43	22.3

Table 2: Per-pixel accuracy of adversarial examples in black-box attack model. We evaluate Pspnet on adversarial examples generate from DeeplabV3+ and vice versa. It can be noted that the transferability of adversarial examples between these two models is very poor.

	Pspnet	DeeplabV3+
per-pixel accuracy($\%$)	83.17	85.17

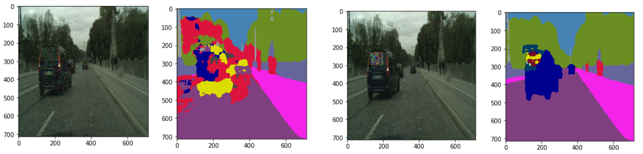


(a) Pspnet

(b) DeeplabV3+

Figure 9: Output segmentation of adversarial examples in black box attack model.

sarial perturbations. Both of these examples demonstrate how adversarial perturbation is only affecting the local pixels only. This strikes the question of why the effects are only local given that theoretical receptive field spans the whole image.



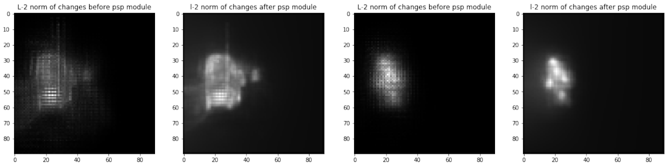
(a) Perturbation added to pixel having label as car.

(b) Perturbations added to a predefined mask.

Figure 10: Output segmentation of adversarial examples when perturbation are added to a selected mask. It can be noted that the effects of such perturbation are only local.

The answer can be traced back to to work in [15], which shows the effective receptive field for last convolution layer for Alexnet trained on Imagenet is approx. 70 pixels. Fig. 12a,12b demonstrate the propagation of changes of adding an adversarial mask to misclassify *Irish wolfhound* as *goldfish*. Due to the limited effective receptive field, both at last CNN layer and layer before that the changes are only local. In the end, the network wrongly classified the input because the global pooling is performed on all output features of final CNN layer before classification.

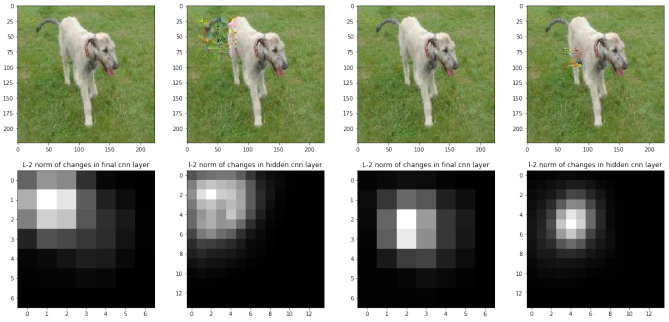
However semantic segmentation proposes a unique challenge because of the absence of any global pooling in the state of the art networks. Though, there exist some mechanism to incorporate more global information, such as PSP modules in Pspnet and dilate convolution in Deeplabv3+, the improvement in adversarial success is not significant. In spite of these steps if the effective receptive field of the network is not large enough, therefore, perturbation effects are only going to be local. Fig. 12 further demonstrate this effect, when local perturbation to whole car (fig. 12a,10a) and the mask on the back of car (fig. 12b,10b) are only propagating changes to local activations in the feature space. Though the addition of PSP module, which does a sparse global pooling, increase this impact, but not to a large extent.



(a) Perturbation added to pixel having label as car.

(b) Perturbation added to pixel having label as car.

Figure 11: Changes in the of layer before and after psp modeuls in Pspnet due to added perturbation (fig. 10)



(a) mask-1

(b) mask-2

Figure 12: Perturbation added to the selected mask to an image from imagenet test dataset. In both cases, the adversarial image is classified as goldfish. It can be noted the changes in the CNN layer of classification networks are also local. But the image is misclassified due to global feature aggregation at the output CNN layers.

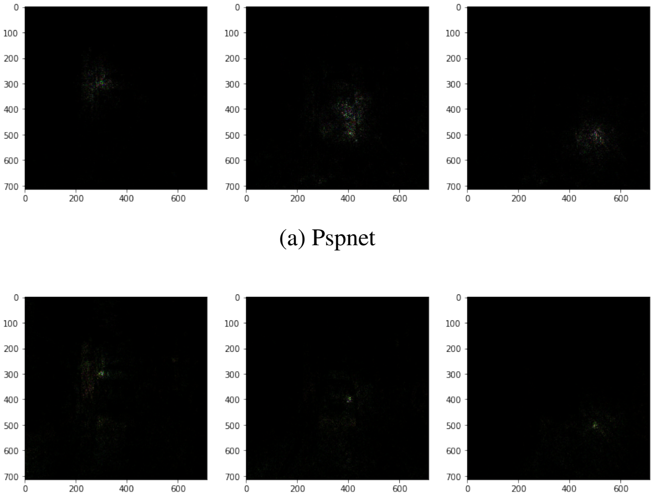
Table 3: Effective receptive field of the networks considered in this work. Resnet-50 is used for image classification study and rest for image segmentation. We report the mean of the receptive field across 1000 images from the respective test dataset.

	Resnet - 50	Pspnet	Deeplabv3+
Effective receptive field	222	353	344

4.1.1 Measuring effective receptive field

To get a better understanding of behavior of different neural networks under masked perturbations, we did a small scale empirical measurement of the receptive field. For each model, we select a few output neurons in the last CNN layer of the network. For each of these neurons, we calculate gradient at the input for a given image. As expected the gradient will be non-zero for all pixels in the input image. However, the value will be significant for only a few local pixels. To extract the effective receptive filed we select pixels with having gradient greater than 10% of the maximum value. Based on these pixels we first calculate the centroid and the then calculate distance of each of these pixels from the centroid. To avoid the impact of outliers, we select the top-50 farthest pixel and report their mean distance from the centroid (after multiplying by a factor of 2 because mean represent effective radius, while receptive field refers to the diameter of total pixels covered), as the effective receptive field. It should be noted that the number in table 3 are highly dependent on the input image and selected output neurons. To avoid this bias to some extent we report mean values across 1000 input images. However, a large-scaled, similar to [15], is necessary to calculate the correct effective receptive field size. Our motivation here is to only look at the relative values for different neural networks.

To our surprise, we observe that the receptive field for segmentation networks (Pspnet and DeeplabV3+) is also quite high. Both of these networks use Resnet-101 as the backbone network. However, we observe that the adversarial success for DeeplabV3+ is significantly lower than Pspnet. Given the approximately same effective receptive size, we wonder what may have triggered this behavior. One possible explanation can be the sparsity of input receptive field. We generally observed only a few pixels having significantly high gradients in DeeplabV3+, as compared to Pspnet (fig. 13). Fig. 13 shows the gradient at input node for three different neurons for a given image (fig. 5a). Note that for Deeplabv3+, the gradient is distributed across as many pixels as Pspnet, but more sparsely. As the magnitude of the gradient is input directly tied to the strength of adversarial attacks, we see less reduction in per-pixel accuracy for DeeplabV3+ (table 1).



(a) Pspnet

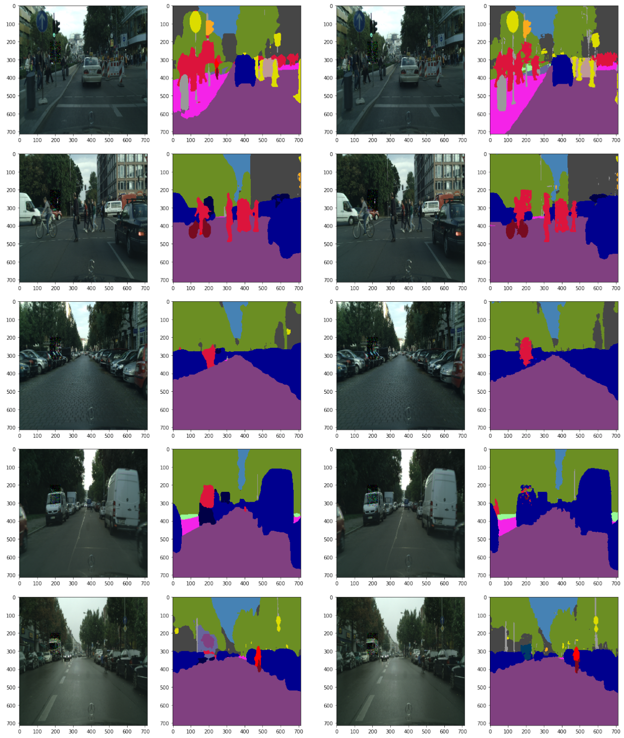
(b) DeeplabV3+

Figure 13: Input gradient of the image used in fig.10 for three output neurons of both networks. It can be noted that the gradients for the DeeplabV3+ model are much sparse than Pspnet, which results in a improved adversarial robustness of the former.

5. Physical adversarial examples

In previous experiments, we assumed that the adversary can add perturbation to the whole image. As argued previously this assumption is not realistic in real-world systems such as self-driving cars [5, 9], which uses segmentation in the vision pipeline of the systems. A more realistic system approach is where adversary add the perturbation to some real objects in the environment (similar to sticking a poster to an object). Fig.14b shows results for some of these images. Due to the limited effective receptive field, these perturbation have only local effect.

However, even with this limited capacity adversary can deliver significant damage to the learning algorithm used. A specific example can be wrong segmentation label for adversarial traffic signs. Some previous works have targeted this problem from the perspective of image classification by sticking adversarial patches on top of these signs. The work can be simply extended to image segmentation also. Another direction is out of distribution attacks [11]. Instead of adding perturbation in a given image from training distribution and making it adversarial, out-of-distribution attacks samples images from the random noise or other distributions. For examples, it can be first selecting a random noise or images of some random logo and add adversarial perturbation on top of it to classify as the target label. Now instead of classification, we can formulate this problem with respect to segmentation. However, due to limited time and space, we left these two problem for future work.



(a) Pspnet (b) DeeplabV3+

Figure 14: Output segmentation of adversarial examples when perturbation are limited to a predefined mask. It can be noted that for all images the effect of these perturbation are only local.

6. Conclusion and Future work

In this work, we study the success and limitation of generating adversarial examples for semantic image segmentation. We consider state of the art image segmentation networks (DeeplabV3+, Pspnet) for this study. First, we study the complexity of robust adversarial examples generation and shows adversarial success for both models. It turns out that adversary can generate examples robust to scale, perspective and contrast variations. However, we found that DeeplabV3+ is more robust than Pspnet to some extent.

Later we consider the threat model where adversary is only allowed to add perturbation to a predefined mask. It turns out that the segmentation networks are highly robust to non-local perturbation. We compare this with previous results on image classification and explain why the lack of global feature aggregation make these networks more robust.

To explain these results we further study the effective receptive field of these networks. As expected the effective receptive field for the output layer was significantly less than the theoretical receptive field. It also helps us to explain why DeeplabV3+ is more robust to adversarial perturbations than Pspnet.

For future work, we plan to extend these properties to image classification networks. We argue that simply a global pooling at the last CNN layer output feature is not a fruitful solution from the perspective of adversarial robustness of a network. More sophisticated classifiers should be designed at top of dense features. This can at least make the network more robust to adversarial patches. Another direction, which is totally unexplored, is the adversarial defenses (primarily adversarial training [6]). It will interesting to see how easy/difficult is to increase the robustness of image segmentation networks using the adversarial training.

References

[1] A. Arnab, O. Miksik, and P. H. Torr. On the robustness of semantic segmentation models to adversarial attacks. *arXiv preprint arXiv:1711.09856*, 2017.

[2] A. Athalye and I. Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[9] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer, 2018.

[10] M. Sharif, L. Bauer, and M. K. Reiter. On the suitability of l_p-norms for creating and preventing adversarial examples. *arXiv preprint arXiv:1802.09653*, 2018.

[11] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018.

[12] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[13] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object

detection. In *International Conference on Computer Vision. IEEE*, 2017.

[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

A. Appendix

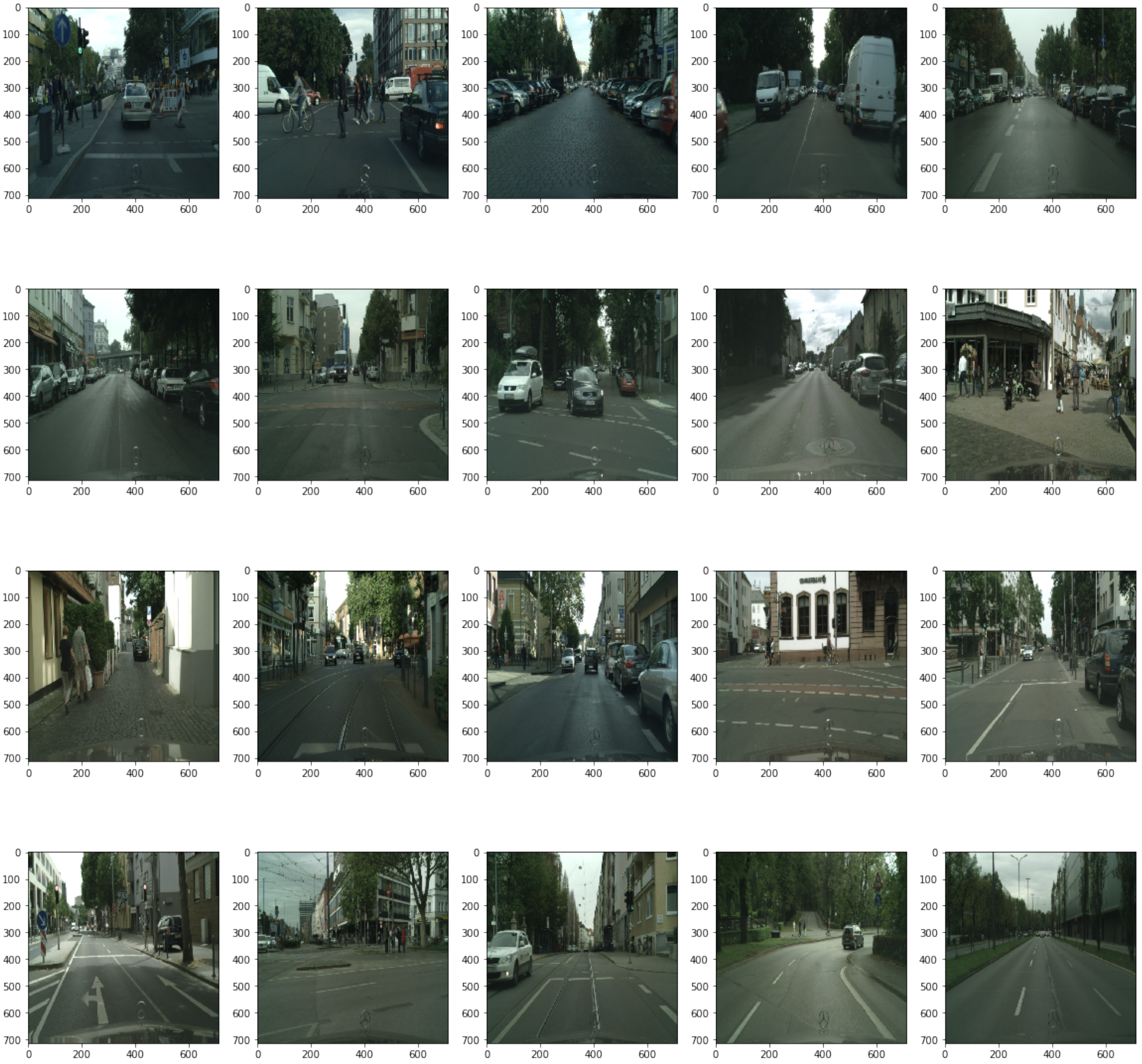
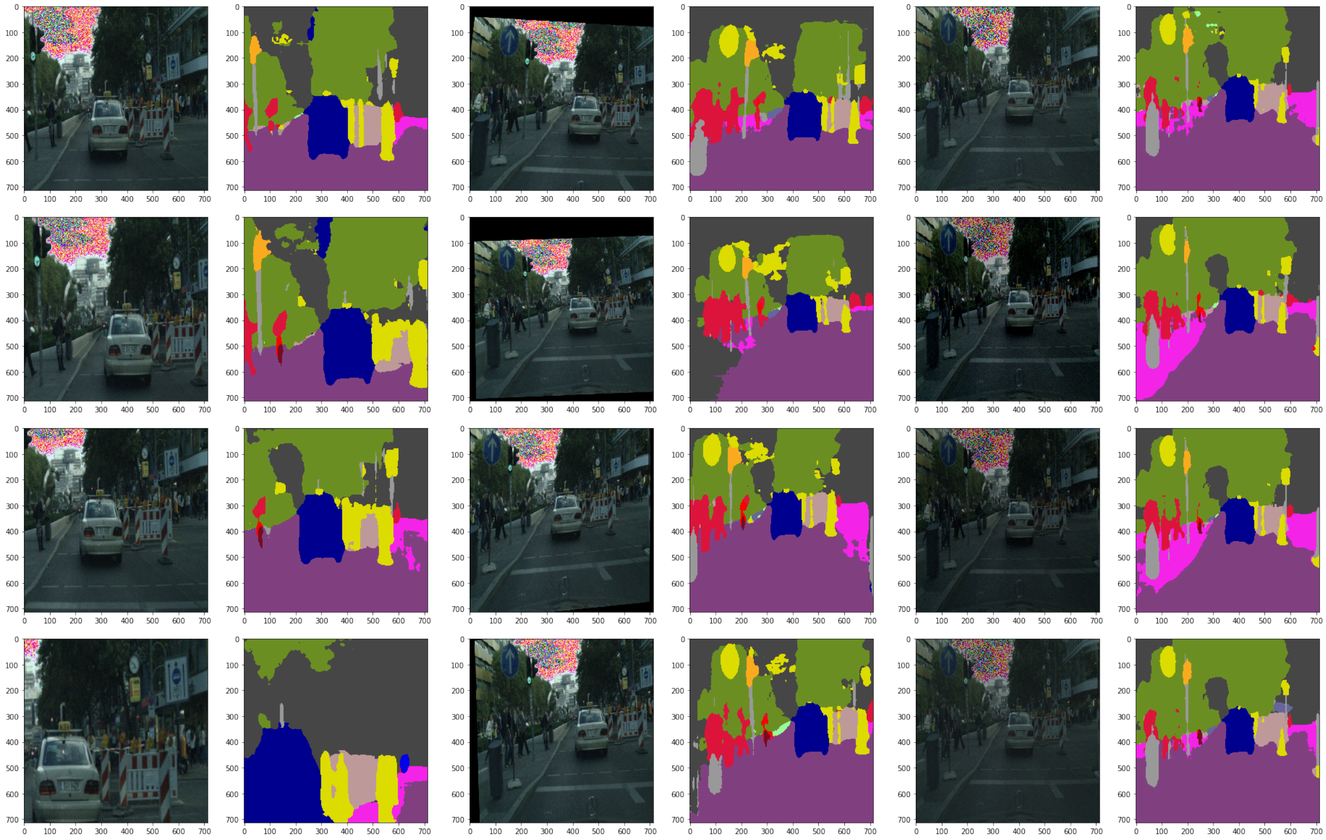


Figure 15: 20 random input images from Cityscape test dataset considered in this work.



(a) Random cropping

(b) Affine transformation

(c) Contrast and gamma variation

Figure 16: Output segmentation for different transformation when a 5% white guassian noise is added to the original image. It shows that then even output with guassian noise is also consistent across transformations.