ADVERSARIAL EXAMPLES FOR SEMANTIC IMAGE SEGMENTATION

Volker Fischer[1], **Mummadi Chaithanya Kumar**[2], **Jan Hendrik Metzen**[1] **& Thomas Brox**[2]
[1] Bosch Center for Artificial Intelligence, Robert Bosch GmbH
[2] University of Freiburg
{volker.fischer, janhendrik.metzen}@de.bosch.com
chaithu0536@gmail.com; brox@cs.uni-freiburg.de

ABSTRACT

Machine learning methods in general and Deep Neural Networks in particular have shown to be vulnerable to adversarial perturbations. So far this phenomenon has mainly been studied in the context of whole-image classification. In this contribution, we analyse how adversarial perturbations can affect the task of semantic segmentation. We show how existing adversarial attackers can be transferred to this task and that it is possible to create imperceptible adversarial perturbations that lead a deep network to misclassify almost all pixels of a chosen class while leaving network prediction nearly unchanged outside this class.

1 INTRODUCTION

While machine learning methods and in particular deep neural networks have led to significant performance increases for numerous tasks, several studies have found them to be vulnerable to adversarial attackers (Szegedy et al., 2014). These attackers compute perturbed versions of input data that fool the classifier to change its predictions on the new version while these perturbations stay almost imperceptible to the human eye. Most of the work on adversarial examples focusses on the task of image classification. In this paper, we investigate the effect of adversarial attacks on a localization task, in particular semantic segmentation.

Our work uses a common deep neural network for semantic segmentation and evaluates to which extent this task is vulnerable to adversarial examples. We introduce a way how adversarial examples could be defined for semantic segmentation and show how existing methods to create adversarial perturbations can be adapted for semantic segmentation. On the basis of the Cityscapes dataset, we find that it is easily possible to let a deep network misclassify entire regions of an image while replacing these misclassified areas with a convenient substitution. Finally we introduce a measure to analyse the effectiveness of a given attacker for a semantic segmentation network.

2 METHODS

Generating adversarial examples Let f_θ denote a function for semantic segmentation such as a deep neural network with parameters θ and x an input image of size $W \times H$ with ground-truth label y^{true}. As most semantic segmentation tasks, we assume that y^{true} has the same pixel dimensions as the image x times the number of classes. We also denote with $J_{\text{cls}}(f_\theta(x), y)$ the classification loss (i.e. cross-entropy) and assume it is differentiable with respect to θ and x.

Since adversarial examples were discovered by Szegedy et al. (2014), several methods have been introduced to generate adversarial perturbations (e.g. (Kurakin et al., 2016), (Liu et al., 2015), (Goodfellow et al., 2015)). In the scope of this work we will focus on the so-called *least likely* method presented by Kurakin et al. (2016). Despite its name this method is not restricted to target the least-likely class, but, in contrast to most other approaches, allows to explicitly specify a target class we want the classifier to predict. This method can easily be transferred from a single pixel to an entire image.

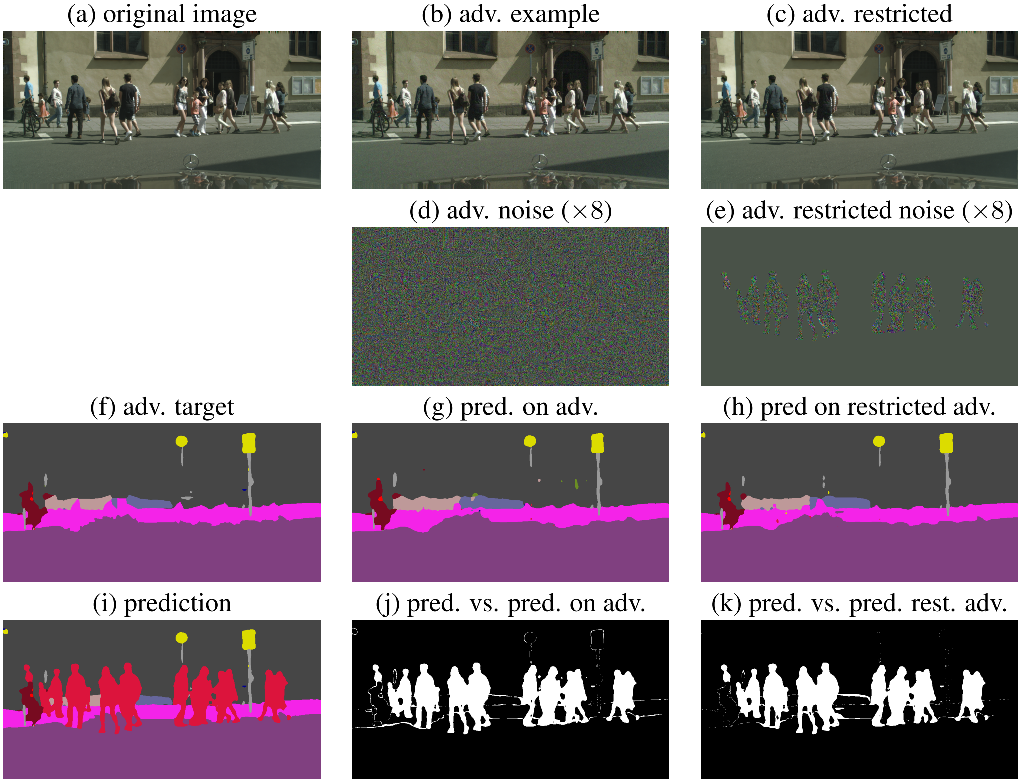

Figure 1: Adversarial example for the task of semantic segmentation: **(a)** The original Cityscapes image. **(b)** Adversarial example computed as described above for $\varepsilon = 10$. **(c)** Same as in (b) but noise is restricted to person pixels only. **(d)** Adversarial noise. **(e)** Adversarial noise restricted to person pixels. **(f)** The adversarial target. **(g)** Network prediction on adversarial example. **(h)** Network prediction on adv. example for restricted noise. **(i)** Network prediction on the original image. **(j)** Classification difference between (i) and (g). **(k)** Classification difference between (i) and (h).

For an arbitrary given target label y^{target} adapting (Kurakin et al., 2016), the least-likely method iteratively computes an adversarial perturbation $\xi^{(n)}$ as:

$$\xi^{(0)} = 0, \quad \xi^{(n+1)} = \text{Clip}_{\varepsilon}\left\{\xi^{(n)} - \alpha \text{sgn}\left(\nabla_x J_{\text{cls}}\left(f_\theta(x + \xi^{(n)}), y^{\text{target}}\right)\right)\right\}$$

Here ε denotes the maximum l^∞-norm of ξ and α the single step-size of one iteration. We set $\alpha = 1$, hence changing each pixel maximally by 1 for each iteration. As suggested in Kurakin et al. (2016), the number of iterations was set to $n = \min(\varepsilon + 4, 1.25\varepsilon)$ and adversarial perturbations were evaluated for different ε. The function $\text{Clip}_{\varepsilon}(\xi)$ clips all values of ξ into the intervall $[-\varepsilon, \varepsilon]$.

Definition of adversarial target for semantic segmentation In our approach the adversarial target covers the entire image and all pixels of a specific class c are changed towards other classes. To look more natural, we suggest to choose the target class y_i^{target} for all to be fooled pixels $i \in \{j : y_j^{\text{pred}} = c\}$ by replacing them with the class of their nearest neighbor that is different from c. For all pixels i not of class c in the original network prediction, we set $y_i^{\text{target}} = y_i^{\text{pred}}$. We measure the effect of adversarial examples by the percentage of pixels of the chosen class that were changed and the percentage of background pixels that were preserved. Because we want to fool the classifier relative to its actual prediction and not the ground-truth, we generate the target label y^{target} on the basis of the network's actual prediction on the unaltered image.

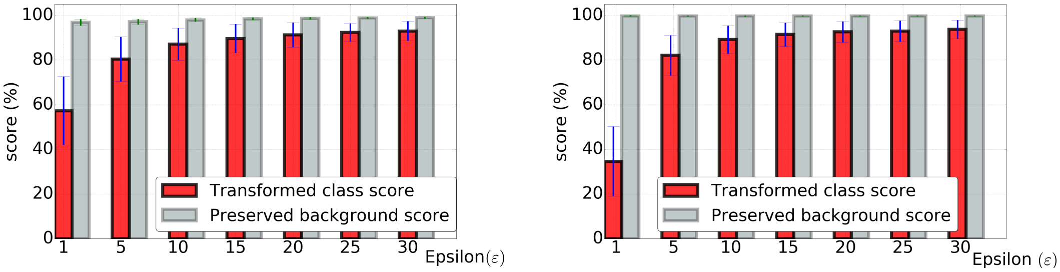

Figure 2: Mean and standard deviation over Cityscapes validation dataset for percentages of preserved (light grey) background and fooled (dark red) person pixels for different ε and for noise applied to the entire image (left) and restricted to person pixels (right).

3 Results & Discussion

Experiments were run with the fully convolutional network architecture introduced by Long et al. (2015) for the VGG16 model. We trained a version of the FCN8 achieving an intersection over union of 55.5%. Network training and evaluation as well as generation and evaluation of adversarial examples was done on the downscaled (1024×512) Cityscapes dataset (Cordts et al., 2016).

For the task of hiding all pixels of the same class, we chose to hide all person pixels of the Cityscapes validation dataset. As described above, adversarial target labels y^{target} were created by first fixating all non-person pixels and second by replacing person pixels with their nearest-neighbor non-person class (see example in Fig.1 original image (i) and target label (f)). These labels were created for all network predictions on Cityscapes validation images and then used to compute adversarial variations from the images as described above. We used different values for ε investigating the effect for different magnitudes of adversarial perturbations. In Fig. 1 a representative example is shown, we see that it is possible to fool the deep network into almost not recognising any person pixels while predicting the substituted background and preserving the background outside the original persons nearly perfectly. We also see that the adversarial perturbation is hard to detect.

Finally we restricted the adversarial perturbations computed above and applied them only on person class pixels of the network's prediction on the original image. The right column of Fig. 1 shows an example for this case. We see that a majority of person pixels can be cloaked while preserving the background. This indicates that especially the noise on the humans is important to hide them.

Fig. 2 shows the mean and standard deviation for percentages of person pixels changed and background pixels preserved for different ε. For the left plot in Fig. 2 adversarial perturbations were applied to the entire image, for the right only on person class pixels. In the first case, we see that for sufficiently low ε of about 10, over 85% of person pixels could be hidden while over 97% of background pixels were preserved. Considering the case adversarial noise was restricted to person pixels (right plot of Fig. 2), we see that the background is preserved even for smaller ε while the number of cloaked person pixels decreases for small ε but recovers for larger values of ε.

4 Conclusion

We adapted the concept of adversarial examples to the task of semantic segmentation and showed that existing approaches to generate adversarial examples for classification can be easily transferred to this task. We showed that there exist imperceptible adversarial perturbations that cloak almost all pixels of a target class while leaving the other classes across the image nearly unchanged. Many open topics remain, such as: usage of more performant networks, comparison of different network architectures, more sophisticated methods to measure the effectiveness of adversarial examples for semantic segmentation, can these adversarial examples be applied in the physical world?

Acknowledgments We thank the developers of Theano (The Theano Development Team, 2016) and keras (`https://keras.io`).

REFERENCES

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv:1607.02533*, July 2016.

Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

The Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*, May 2016.