Single-step Adversarial Training for Semantic Segmentation

Daniel Wiens[1,2] and Barbara Hammer[2]
[1] *Mercedes-Benz AG, Stuttgart, Germany*
[2] *Bielefeld University, Bielefeld, Germany*

Keywords: Adversarial Training, Adversarial Attack, Step Size Control, Semantic Segmentation, Deep Neural Network.

Abstract: Even though deep neural networks succeed on many different tasks including semantic segmentation, they lack on robustness against adversarial examples. To counteract this exploit, often adversarial training is used. However, it is known that adversarial training with weak adversarial attacks (e.g. using the Fast Gradient Sign Method) does not improve the robustness against stronger attacks. Recent research shows that it is possible to increase the robustness of such single-step methods by choosing an appropriate step size during the training. Finding such a step size, without increasing the computational effort of single-step adversarial training, is still an open challenge. In this work we address the computationally particularly demanding task of semantic segmentation and propose a new step size control algorithm that increases the robustness of single-step adversarial training. The proposed algorithm does not increase the computational effort of single-step adversarial training considerably and also simplifies training, because it is free of meta-parameter. We show that the robustness of our approach can compete with multi-step adversarial training on two popular benchmarks for semantic segmentation.

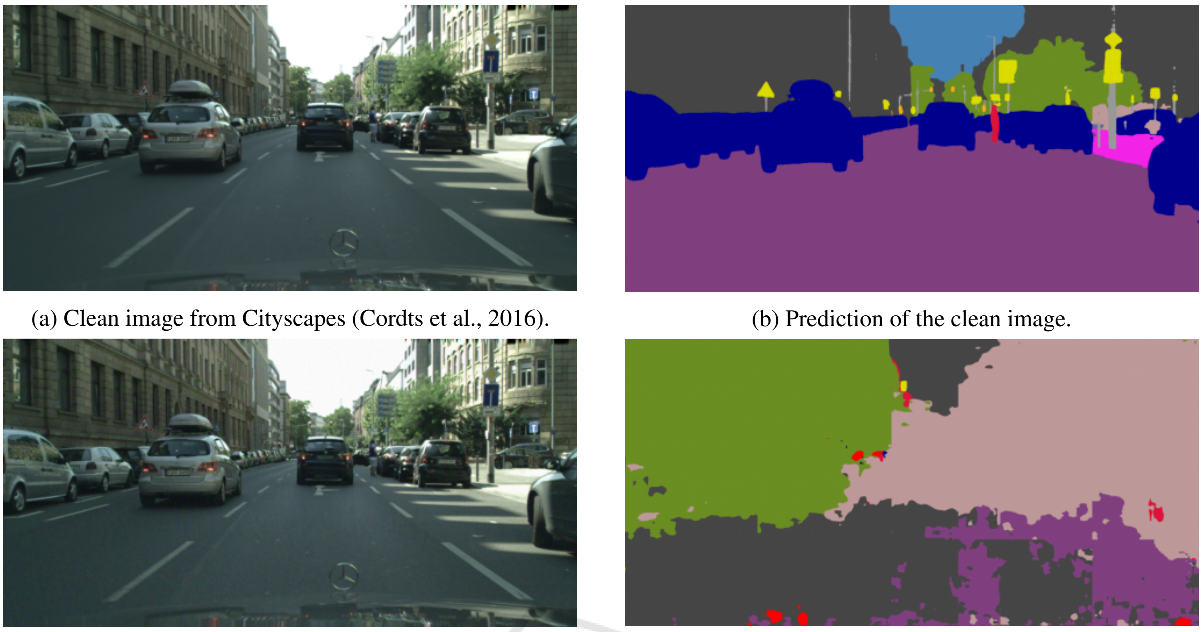
1 INTRODUCTION

Due to their great performance, deep neural networks are increasingly used on many classification tasks. Especially in vision tasks, like image classification or semantic segmentation, deep neural networks have become the standard method. However, it is known that deep neural networks are easily fooled by adversarial examples (Szegedy et al., 2014), i.e. very small perturbations added to an image such that neural networks classify the resulting image incorrectly. Interestingly, adversarial examples can be generated for multiple machine learning tasks, including image classification and semantic segmentation, and the perturbations are most of the time so small that humans do not even notice the changes (e.g. see fig. 1). This phenomenon highlights a significant discrepancy of the human vision system and deep neural networks, and highlights a possibly crucial vulnerability of the latter. This fact should be taken into consideration, especially in safety critical applications like autonomous driving cars (Willers et al., 2020).

To increase the robustness of deep neural networks, progress along two different lines of research could be observed in the last years: provable robustness and adversarial training. Provable robustness has the goal to certify that the prediction does not change in a local surrounding for most inputs. This approach has the advantage of yielding robustness guarantees, but it is not that scalable to complex deep neural networks yet (Wong and Kolter, 2018; Raghunathan et al., 2018), or it severely affects the inference time (Cohen et al., 2019) which is problematic for many applications. In contrast to this theoretical viewpoint, the idea of adversarial training is more empirically driven: create adversarial examples during training and use them as training data (Goodfellow et al., 2015; Madry et al., 2018), this procedure can be interpreted as an efficient realization of a robustified loss function which minimizes the error simultaneously for potentially disturbed data (Shaham et al., 2016). Adversarial training has the advantage that it is universally applicable, and often results in high robustness albeit this holds empirically and w.r.t. a specific norm. Only reject options have the potential to improve the robustness w.r.t. different norms (Stutz et al., 2020).

An optimization of the inner loop of the adversarial loss function is often addressed by numeric methods which rely on an iterative perturbation of the input in the direction of its respecting gradient. But this leads to multiple forward and backward passes

Wiens, D. and Hammer, B.
Single-step Adversarial Training for Semantic Segmentation.
DOI: 10.5220/0010788400003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 179-187
ISBN: 978-989-758-549-4; ISSN: 2184-4313



(a) Clean image from Cityscapes (Cordts et al., 2016).

(b) Prediction of the clean image.

(c) Adversarial example created with the Basic Iterative Method ($\varepsilon = 0.03$).

(d) Prediction of the adversarial example.

Figure 1: Predictions for a clean input and an adversarial example produced by a deep neural network for semantic segmentation. The two inputs in (a) and (c) look the same for a human observer, but the predictions of the clean image and the adversarial example shown in (b) and in (d), respectively, are completely different.

through the deep neural network and therefore highly increases the training time. To reduce the computational effort of adversarial training Goodfellow et al. (2015) used the information of just a single gradient for computing the adversarial examples. But this kind of single-step adversarial training can be too weak and it has been observed that overfitting can take place, i.e. the specific adversarial is classified correctly, but not its immediate environment. In particular, it is not robust against multi-step attacks (Madry et al., 2018). Recent research discovered that this overfitting is caused by using a static step size while creating the adversarial examples for adversarial training (Kim et al., 2020). To overcome this, Kim et al. (2020) propose a method to find an ideal step size by evaluating equidistant points in the direction of the gradient. But this algorithm is in worst case as expensive as multi-step adversarial training.

Differently to most of the previous research on adversarial robustness, we will focus on adversarial training for semantic segmentation in this work. The task of semantic segmentation is to assign every pixel of the input image a corresponding class. Because semantic segmentation needs to address localization and semantic simultaneously, it is a more complicated task than image classification (Long et al., 2015). Consequently, the models for semantic segmentation are generally complex, and therefore the computational effort to train such models is in principal very high. Since adversarial training also negatively affects the training time, our goal is to find a computationally efficient method which at the same time increases the robustness of semantic segmentation models effectively.

For this purpose, we extend the idea of robust single-step adversarial training. On the one hand, we investigate single-step adversarial training for semantic segmentation as a relevant and challenging application problem for deep learning, and we demonstrate that it shares the problem of overfitting and limited robustness towards multi-step attacks with standard classification problems. On the other hand, we demonstrate that a careful selection of the step size can mitigate this problem insofar as even random step sizes improve robustness. We propose an efficient parameterless method to choose an optimum step size, which yields robustness results which are comparable to multi-step adversarial training for two popular benchmarks from the domain of image segmentation, while sharing the efficiency of single-step approaches.

2 BACKGROUND AND RELATED WORK

2.1 Adversarial Training

Adversarial training for a classical image classification task tries to solve the following robust optimization problem (Shaham et al., 2016): given a paired sample (x,y) consisting of an input sample $x \in X \subset [0,1]^{H\times W\times C}$ and an associated label $y \in Y \subset \{1,\dots,M\}$, where H, W, C and M are the height, the width, the number of color channels and the number of classes, respectively, let ℓ denote the loss function of a deep neural network $f_\theta : X \to [0,1]^M$, robust learning aims for the weights θ with

$$\min_\theta \mathbb{E}_{(x,y)\in(X,Y)} \left(\max_{\delta\in\mathcal{B}(0,\varepsilon)} \ell(f_\theta(x+\delta),y) \right), \qquad (1)$$

where $\mathbb{E}$ and $\mathcal{B}(0,\varepsilon)$ denote the expected value and a sphere at origin with radius ε, respectively. The sphere is dependent on a chosen distance metric. In the context of adversarial examples most often $\mathcal{L}_0$, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ are considered. In this paper we focus only on $\mathcal{L}_\infty$.

Because the loss function is highly nonlinear and non-convex even solving the inner maximization in eq. (1) is considered intractable (Katz et al., 2017; Weng et al., 2018). Therefore, the maximization problem is often approximated by crafting adversarial examples, such that the optimization of adversarial training changes to

$$\min_\theta \mathbb{E}_{(x_{\text{adv}},y)\in(X_{\text{adv}},Y)} \ \ell(f_\theta(x_{\text{adv}}),y), \qquad (2)$$

where the adversarial examples $x_{\text{adv}} \in X_{\text{adv}} \subset [0,1]^{H\times W\times C}$ are generated by a chosen adversarial attack, such that $\|x - x_{\text{adv}}\|_\infty \leq \varepsilon$ holds. To guarantee the classification accuracy on the original clean samples usually x_{adv} is randomly set to be a clean sample or an adversarial example. For crafting adversarial examples while training, most often gradient based methods, like the Fast Gradient Sign Method or the Basic Iterative Method, are used.

2.1.1 Fast Gradient Sign Method

The Fast Gradient Sign Method (FGSM) is a single-step adversarial attack which is a particularly easy and computationally cheap gradient based method, because it uses just a single gradient for calculating the adversary noise (Goodfellow et al., 2015). If a step size ε is given, an adversarial example is determined by

$$x_{\text{adv}} = x + \varepsilon \cdot \operatorname{sign}(\nabla_x \ell(f_\theta(x),y)). \qquad (3)$$

Notice, while generating adversarial examples the color channels of the resulting images can leave their value range of $[0,1]$. To prevent the values of leaving the interval, they are throughout this work always clipped into the allowed range.

2.1.2 Basic Iterative Method

The Basic Iterative Method (BIM) is a multi-step generalization of the FGSM (Kurakin et al., 2016). Instead of evaluating just one gradient, the BIM uses n gradients iteratively, to reach a stronger adversarial example. Given a maximum perturbation ε, a number of iterations N and a step size α, an adversarial example created with BIM is given iteratively by

$$\begin{aligned} x_0 &= x, \\ x_{i+1} &= \Pi_{x,\varepsilon}\left(x_i + \alpha \cdot \operatorname{sign}\left(\nabla_x \ell\left(f_\theta\left(x_i\right),y\right)\right)\right), \qquad (4) \\ x_{\text{adv}} &= x_N. \end{aligned}$$

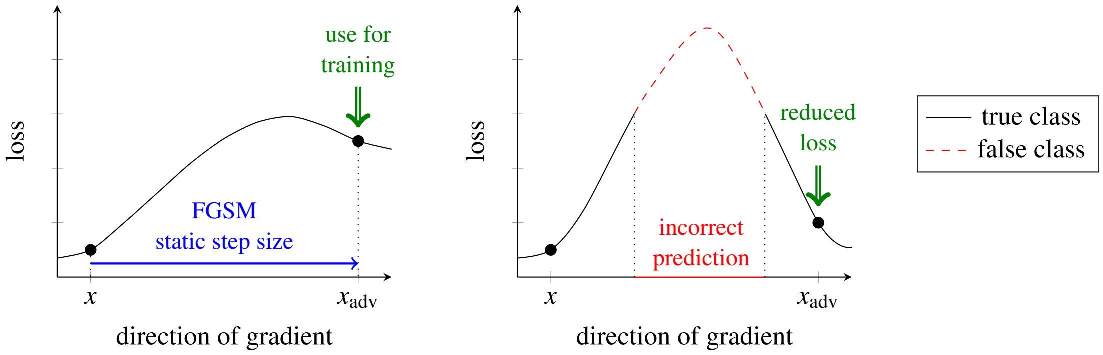
Where the function $\tilde{p} = \Pi_{x,\varepsilon}(p)$ projects p into the ε-neighbourhood of x, such that $\|\tilde{p} - x\|_\infty \leq \varepsilon$.

2.2 Restrictions of Single-step Adversarial Training

To generate adversarial examples while training, Goodfellow et al. (2015) chose the FGSM as a computationally cheap adversarial attack. But it was shown that models trained with such a single-step adversarial training are not robust against multi-step attacks (Madry et al., 2018). Therefore, to train robust models, multi-step adversarial training is commonly used. Since multi-step approaches require multiple forward and backward passes through the deep neural network, training a robust model becomes computationally very intensive.

To minimize the computational effort while training, one line of work tries to increase the robustness of single-step adversarial training. Wong et al. (2020) analyzed the robustness of single-step adversarial training epoch-wise and observed that the robustness against BIM increases in the beginning during training, but decreases after some epochs. This observation is called catastrophic overfitting. To overcome this phenomenon Wong et al. (2020) added random noise to the input before using the FGSM for adversarial training and also added early-stopping by tracking the multi-step robustness of small batches. A few further approaches follow this line of research (Li et al., 2020; Andriushchenko and Flammarion, 2020).

Because these methods need to calculate more than one gradient at some point, they are computationally inefficient. Kim et al. (2020) showed empirically that the static step size for generating the adver-



(a) Before catastrophic overfitting.

(b) Catastrophic overfitting.

Figure 2: Sketch: The static step size of single-step adversarial training as reason for catastrophic overfitting. (a) After initial single-step adversarial training the adversarial example is classified correctly. (b) Further training reduces the loss of the adversarial example in fixed distance to the original sample even more, but in between the loss increases resulting in a region of incorrect predictions.

sarial examples constitute a reason why single-step adversarial training is not robust. Single-step adversarial examples which are generated by a fixed step size have a fixed distance from their original data points. When training on such samples as well as on the original data points, the loss at these points becomes very low, but there is no incentive to reduce the loss between these extremes or to provide a monotonic output in this region. This can lead to catastrophic overfitting as displayed in fig. 2, i.e. the prediction of the neural network at the original data and the adversarial examples are correct, but in between there is a region where the prediction of the network becomes incorrect. Multi-step adversarial attacks iteratively use small step sizes, and therefore are able to find these areas of non-monotonicity, such that single-step adversarial training is not robust against multi-step attacks.

To overcome this problem, Kim et al. (2020) proposed to test a range of different step sizes by equidistant sampling in the direction of the gradient and to choose the smallest step size which leads to an incorrect prediction. Since this approach aims for a minimum step size to generate an adversarial sample, we can expect monotonicity of the network output in between, at least it is guaranteed that no closer adversarials can be found in the direction of the gradient. Hence overfitting seems less likely in such cases, as is confirmed by the experimental finding presented in (Kim et al., 2020).

Yet, the sampling strategy which is presented in (Kim et al., 2020) requires several forward passes through the network and is thus in worst case as demanding as multi-step approaches. Additionally, the strategy is not well suited for semantic segmentation, because on semantic segmentation it is not defined whether an adversarial example leads to a false prediction or not. In the following, we propose an approximate analytic solution how to compute a closest adversarial, which leads to an efficient implementation of this robust step size selection strategy and which can be utilized for the task of semantic segmentation.

3 METHODS

3.1 Choosing the Step Size for Image Classification

To explain the idea behind our step size control algorithm, we first start looking at image classification. We are interested in an efficient analytic approximation which yields in the direction of the gradient the closest adversarial example, i.e. a closest pattern where the classification changes.

The deep neural network is given as a function $f_\theta : X \to [0,1]^M$, where the output of the deep neural network $f_\theta(x)$ is computed using the softmax function. The inputs to the softmax $z(x)$ are called logits. We define the gain function $g : X \times Y \to \mathbb{R}$ as

$$g(x,y) = z_y(x) - \max_{i \neq y} z_i(x), \tag{5}$$

where $z_i(x)$ is the logit value of class i. The gain function in eq. (5) has the property that

$$g(x,y) \begin{cases} > 0 & \text{if } \arg\max f_\theta(x) = y \\ \leq 0 & \text{else} \end{cases} . \tag{6}$$

For given (x,y) a close adversarial example can be found at the decision boundary between a correct and an incorrect prediction, which holds if

$$g(x_{\text{adv}}, y) = 0. \tag{7}$$

Using the update rule of the FGSM in eq. (3) on the gain function, we create an adversarial example by

$$x_{\text{adv}} = x - \varepsilon \cdot \operatorname{sign}(\nabla_x g(x,y)). \tag{8}$$

For estimating the step size ε, we linearly approximate the gain function g with a Taylor approximation

$$g(x_{\text{adv}}, y) \approx g(x,y) + (x_{\text{adv}} - x)^T \cdot \nabla_x g(x,y). \tag{9}$$

Combining eq. (7), eq. (8) and eq. (9) results in

$$0 = g(x,y) - \varepsilon \cdot \operatorname{sign}(\nabla_x g(x,y))^T \cdot \nabla_x g(x,y), \tag{10}$$

which we can solve for the step size ε:

$$\begin{aligned} \varepsilon &= \frac{g(x,y)}{\nabla_x g(x,y)^T \cdot \operatorname{sign}(\nabla_x g(x,y))} \\ &= \frac{g(x,y)}{\|\nabla_x g(x,y)\|_1}. \end{aligned} \tag{11}$$

Notice, the idea of the above step size control is the same as doing one step of Newton's approximation method for finding the zero crossings of a function. We therefore call our approach Fast Newton Method.

Contrary to the classical solution of robust optimization, we use an approximation of the closest adversarial example for training. This has two effects: on the one hand, we expect monotonicity of the prediction in between, or a sufficient distance from 0. Hence a network which classifies x and the closest adversarial correctly likely classifies the enclosed interval in the same way, avoiding catastrophic overfitting. On the other hand, we do not consider any predefined radius; rather we use the smallest radius that includes an adversarial example. Using these adversarial examples for training, iteratively increases the radius of the sphere, and consequently the distance where adversarials can be found.

3.2 Robust Semantic Segmentation

The above mentioned results were presented in the context of image classification. In this work we want to concentrate on semantic segmentation. It is already known that deep neural networks for semantic segmentation are also vulnerable against adversarial examples (Xie et al., 2017; Metzen et al., 2017), but only few approaches address the question how to efficiently implement robust training for image segmentation tasks. As far as we know, there is only one work investigating the impact of multi-step adversarial training on semantic segmentation (Xu et al., 2020).

First, we formalize the learning objective for robust semantic segmentation. Because semantic segmentation determines the class of each single pixel of an image, semantic segmentation could be interpreted as multiple pixel-wise classifications. To predict all the pixels at the same time the deep neural network function $f_\theta : X \to [0,1]^{H\times W\times M}$ has increased output dimension. The set of labels are given by $Y \subset \{1,\ldots,M\}^{H\times W}$. Robust learning on semantic segmentation tries to optimize

$$\min_\theta \mathbb{E}_{(x,y)\in(X,Y)} \left(\max_{\delta\in\mathcal{B}(0,\varepsilon)} \frac{1}{HW} \sum_{j=0}^{HW} \ell(f_{\theta,j}(x+\delta), y_j) \right) \tag{12}$$

for the weights θ, where $f_{\theta,j}(x)$ and y_j are the prediction and the label of the j-th pixel, respectively. To find the weights for all the pixel-wise predictions simultaneously, the losses of the pixel-wise predictions are averaged.

Adversarial training on semantic image segmentation approximates the min-max loss by the following term:

$$\min_\theta \mathbb{E}_{(x_{\text{adv}},y)\in(X_{\text{adv}},Y)} \frac{1}{HW} \sum_{j=0}^{HW} \ell(f_{\theta,j}(x_{\text{adv}}), y_j), \tag{13}$$

where $x_{\text{adv}} \in X_{\text{adv}}$ constitutes a suitable image with $\|x - x_{\text{adv}}\|_\infty \leq \varepsilon$, which plays the role of an adversarial in the sense that it leads to an error of the segmented image for a large number of pixels. Yet, unlike for scalar outputs, it is not clear what exactly should be referred to by an adversarial: we can aim for an input x_{adv} such that all output pixels change, or, alternatively, approximate this computationally extensive extreme by an efficient surrogate, as we will introduce in the following.

3.3 Choosing the Step Size for Semantic Segmentation

Initially, we treat each pixel as a separate output. An adversarial corresponds to an input such that one specific output pixel changes. For this setting, the gain function from eq. (5) becomes:

$$g_j(x, y_j) = z_{j,y_j}(x) - \max_{i\neq y_j} z_{j,i}(x), \tag{14}$$

for $j \in \{1,\ldots,HW\}$, where $z_{j,i}(x)$ is the logit value of the j-th pixel of class i. To find a close adversarial example x_{adv} which changes all (or a large number of) pixels, each pixel-wise output should be at the boundary between a correct and an incorrect prediction. This holds if

$$g_j(x_{\text{adv}}, y_j) = 0, \quad \forall j \in \{1,\ldots,HW\}. \tag{15}$$

Using our results from sec. 3.1, we obtain a different step size ε_j for every pixel, hence a possibly different adversarial $x_{\text{adv}}^{(j)}$ for each pixel-wise prediction. A common adversarial example x_{adv} which approximately suits for all pixel-wise predictions in the same time, could be chosen as the average term

$$x_{\text{adv}} = \frac{1}{HW} \sum_{j=1}^{HW} x_{\text{adv}}^{(j)}. \tag{16}$$

But this procedure requires the computation of HW different gradients $\nabla_x g_j(x, y_j)$ for $j = 1, \ldots, HW$ which is computational too expensive to work with.

As an alternative, we can refer to adversarial examples as images for which a number of pixels change the segmentation assignment. In this case we consider the pixel-wise predictions at the same time from the beginning by using the average of the gain functions (rather than the average of the specific pixel-wise adversarials). A close adversarial example in this sense is then given if

$$\frac{1}{HW} \sum_{j=1}^{HW} g_j(x_{\text{adv}}, y_j) = 0. \tag{17}$$

Defining the averaged gain function as $\bar{g}(x,y) = \frac{1}{HW} \sum_{j=1}^{HW} g_j(x, y_j)$ and using our method from sec. 3.1 results in a single step size

$$\varepsilon = \frac{\bar{g}(x,y)}{\|\nabla_x \bar{g}(x,y)\|_1}, \tag{18}$$

such that the update rule for finding an adversarial example becomes

$$x_{\text{adv}} = x - \frac{\bar{g}(x,y)}{\|\nabla_x \bar{g}(x,y)\|_1} \cdot \operatorname{sign}(\nabla_x \bar{g}(x,y)). \tag{19}$$

Of course using the averaged gain function for calculating the step size ε in eq. (18) leads to a more loose approximation, but as a trade off this method does not increase the computational effort compared to adversarial training with the FGSM considerably, because we also calculate just one gradient.

4 EXPERIMENTS

4.1 Implementation

4.1.1 Datasets

To evaluate our approach, we use the datasets Cityscapes (Cordts et al., 2016) and PASCAL VOC (Everingham et al., 2015). Cityscapes contains 2975, 500 and 1525 colored images of size 1024×2048 for training, validation and testing, respectively. The labels assign most pixels one of 19 classes, where some areas are not labeled.

PASCAL VOC originally includes 1464, 1499 and 1456 differently sized images for training, validation and testing, respectively. Later the training set was increased to 10582 images (Hariharan et al., 2011). Including the background class, the pixels are labeled as one of 21 classes. Like in Cityscapes there are also some areas not labeled, such that we limit our loss and gain function for both datasets to the labeled pixels. We always normalise the color channels in the range of $[0, 1]$.

4.1.2 Model

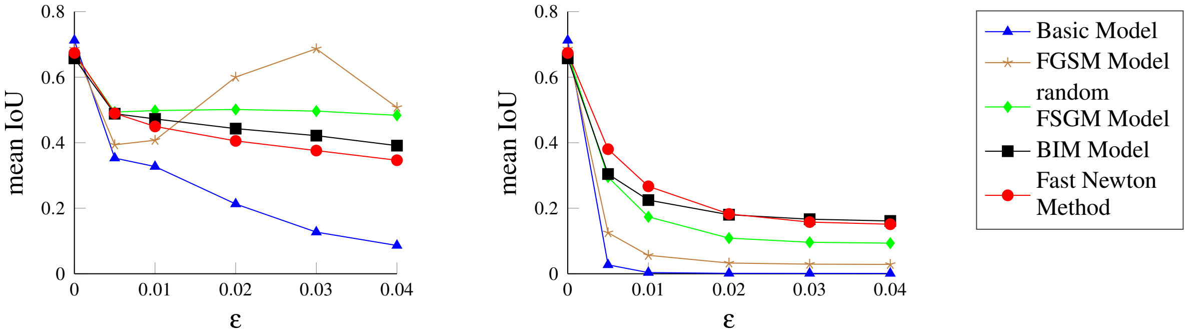
As base model we chose the popular PSPNet50 architecture from (Zhao et al., 2017) with a slight modification. Instead of using ResNet50 we work with the improved ResNet50v2 as fundamental pretrained network (He et al., 2016). For the training parameters we follow the paper (Zhao et al., 2017) using SGD with a momentum of 0.9. The learning rate decays polynomially with a base learn rate of 0.01 and power 0.9. We use a batchsize of 16 and also include $\mathcal{L}_2$ weight decay of 10^{-4}. However, for a better comparison to different models, we do not apply the auxiliary loss from (Zhao et al., 2017). To further increase the variation of the dataset, the data is randomly horizontally flipped, resized between 0.5 and 2, rotated between -10 and 10 degrees, and additionally randomly blurred. Afterwards, we randomly crop the images to a size of 712×712 and 472×472 for Cityscapes and PASCAL VOC, respectively. For evaluation, we will compare the robustness of the following models:

- The Basic Model without using any adversarial examples in training,
- The FGSM Model trained with FGSM at $\varepsilon = 0.03$,
- The random FGSM Model trained with FGSM at $\varepsilon \sim \mathcal{U}(0, 0.03)$ chosen uniformly,
- The BIM Model trained with BIM ($\alpha = 0.01$, $\varepsilon = 0.03$, $N = 3$),
- Fast Newton Method trained with our algorithm,

where the parameters of the BIM Model are taken from (Xu et al., 2020). For adversarial training the input is randomly chosen either a clean data sample or an adversarial example, each with probability of 0.5.

4.1.3 Evaluating the Robustness

We measure the performance on the clean and the adversary data with the mean Intersection over



(a) Robustness against single-step adversarial examples created with FGSM.

(b) Robustness against adversarial examples created with BIM ($N = 10$, $\alpha = 0.004$).

Figure 3: Robustness curves on the Cityscapes dataset.

Table 1: Additional training time on Cityscapes shown as percentage compared to the training time of the Basic Model.

Method	Basic Model	FGSM Model	random FGSM Model	Fast Newton Method	BIM Model
Additional time	0.00%	12.52%	12.34%	11.98%	45.92%

Unit (IoU) (Everingham et al., 2015). The mean IoU is a standard measure for semantic segmentation. It is used over the standard accuracy per Pixel, because it better represents the desired accuracy in case of differently sized objects.

For the empirical robustness evaluation we attack each model with FGSM and BIM. The adversarial examples are generated by maximizing the cross-entropy loss. For both attacks we vary the attack radius ε from 0 to 0.04, such that we receive one robustness curve for each model and attack. As trade off between computational time and attack strength, we use BIM with $N = 10$ iterations. We chose the step size α as small as possible, such that the maximum perturbation rate of $\varepsilon = 0.04$ is still reachable. Hence, we use $\alpha = 0.004$ as step size for BIM.

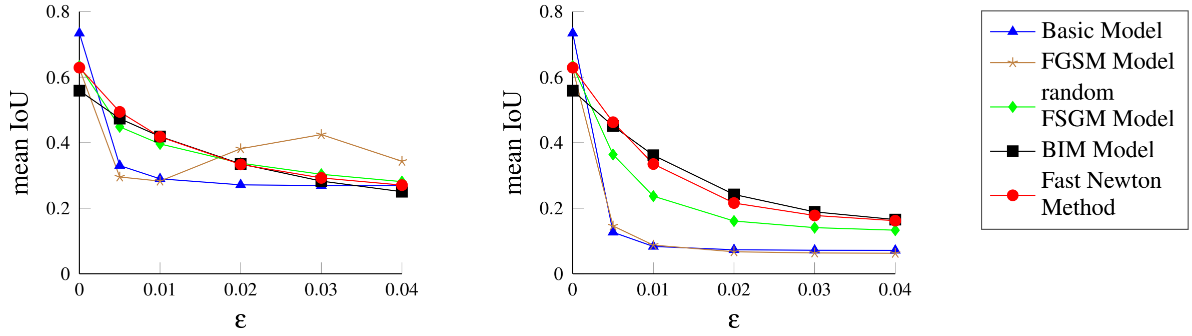
4.2 Results on Cityscapes

4.2.1 Single-step Robustness

In fig. 3(a) are the summarized results regarding the FGSM shown. We see that the mean IoU of all models except the FGSM Model and the random FGSM Model decrease with increased attack strength ε, whereas the performance of the BIM Model and our Fast Newton Method decrease slower. The BIM Model shows slightly better robustness than ours, but the FGSM Model and the random FGSM Model perform even better. Looking closer to the robustness curve of the FGSM Model, we observe that the model is most robust to adversarial examples with attack strength $\varepsilon = 0$ and $\varepsilon = 0.03$. For the other values of ε the robustness decreases. This observation matches with the phenomenon of catastrophic overfitting shown in fig. 2. The model is most robust exactly against the adversarial examples it was trained for. For the values between $\varepsilon = 0$ and $\varepsilon = 0.03$ the nonlinear character of deep neural networks leads to a drop in the robustness.

4.2.2 Multi-step Robustness

Fig. 3(b) shows the robustness of the trained models against attacks with BIM. As can be seen, all models perform significantly worse against this attack. Because the robustness is defined as the performance under the worst case attack, this evaluation represents the real robustness of the models far better. We can see that the Basic Model and FGSM Model drop very fast with increased attack strength ε. So we can confirm that the FGSM Model overfits. Even though the Fast Newton Method and the random FGSM Model are also single-step adversarial training models, they perform significantly better under the attack with BIM than the FGSM Model. That shows that the idea of controlling the step sizes of adversarial attacks makes single-step adversarial training more effective. The performance of our Fast Newton Method compared to the FGSM models shows the importance of choosing the correct step size for creating adversarial examples during training. Additionally, our Fast Newton Method shows for small ε even better robustness than the more sophisticated BIM model, and is for larger ε equally robust while being significantly less computational expensive (see table 1).



(a) Robustness against adversarial examples created with FGSM.

(b) Robustness against adversarial examples created with BIM ($N = 10$, $\alpha = 0.004$).

Figure 4: Robustness curves on the PASCAL VOC dataset.

Table 2: Additional training time on PASCAL VOC shown as percentage compared to the training time of the Basic Model.

Method	Basic Model	FGSM Model	random FGSM Model	Fast Newton Method	BIM Model
Additional time	0.00%	31.76%	34.71%	31.18%	98.24%

4.3 Results on PASCAL VOC

4.3.1 Single-step Robustness

The robustness curves against the FGSM for models trained on PASCAL VOC are shown in fig. 4(a). We observe that the attack is not as effective on the Basic Model as on Cityscapes. We think that this is mainly reasoned in the characteristic of the dataset with its dominant background class. It seems difficult for the attack to switch the class of the whole background area. But as we already mentioned, the robustness is better represented by a strong multi-step attack. Our focus here lays again in the curve course of the FGSM Model. Like on Cityscapes the FGSM Model is most robust against the adversarial examples it was trained for ($\varepsilon = 0$ and $\varepsilon = 0.03$). So we see again an indicator for catastrophic overfitting.

4.3.2 Multi-step Robustness

Looking in Fig. 4(b) on the robustness of the trained models against the BIM, we can observe very similar results like on Cityscapes. Both, the Basic and the FGSM Model, perform much worse than the other models. So we can clearly say that the FGSM Model overfits to adversarial examples it was trained with. However, even if the random FGSM Model and the Fast Newton Method are trained with single-step adversarial examples, too, they perform significantly better than the FGSM Model. Thus, we can confirm that varying the step size increases the robustness of such models. Because the Fast Newton Method outperforms the random FGSM Model, we conclude that our proposed step size is more superior than determining the step size randomly. Additionally, even if our Fast Newton Method is less computational expensive than the BIM Model (see table 2), their robustness is very similar.

5 CONCLUSION

The research community focused so far on improving the robustness of deep neural networks for image classification. We on the other hand concentrate on semantic segmentation. We showed that single-step adversarial training for semantic segmentation underlies the same difficulties regarding the robustness against multi-step adversarial attacks. One reason for that non-robustness of single step adversarial training is the static step size for finding the adversarial examples while training. Therefore, we presented a step size control algorithm which approximates an appropriate step size for every input such that the robustness of single-step adversarial training increases significantly. As our method approximates the best step size based on the gradient, which needs to be calculated anyway for adversarial training, our method does not considerably increase the computational effort while training a significantly more robust model. In addition, our approach is easy to use, because it is free of any parameter. Finally, we showed on the datasets Cityscapes and PASCAL VOC, that our method equals in performance with the more sophisticated and the more computationally expensive multi-step adversarial training.

REFERENCES

Andriushchenko, M. and Flammarion, N. (2020). Understanding and improving fast adversarial training. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In *36th International Conference on Machine Learning*. PMLR.

Cordts, M., Omran, M., Ramos, S., Refeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schielde, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Everingham, M., Eslami, S., Gool, L. V., Williams, C., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. In *Int J Comput Vis*.

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015*.

Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*. Springer International Publishing.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. (2017). Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification*. Springer International Publishing.

Kim, H., Lee, W., and Lee, J. (2020). Understanding catastrophic overfitting in single-step adversarial training. arXiv. Last accessed 20 April 2021.

Kurakin, A., Goodfelow, I., and Bengio, S. (2016). Adversarial examples in the physical world. *CoRR*.

Li, B., Wang, S., Jana, S., and Carin, L. (2020). Towards understanding fast adversarial training. arXiv. Last accessed 20 April 2021.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018*.

Metzen, J., Kumar, M., Brox, T., and Fischer, V. (2017). Universal adversarial perturbations against semantic image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*.

Raghunathan, A., Steinhardt, J., and Liang, P. (2018). Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018*.

Shaham, U., Yamada, Y., and Negahban, S. (2016). Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *Neurocomputing*.

Stutz, D., Hein, M., and Schiele, B. (2020). Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *37th International Conference on Machine Learning*. PMLR.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

Weng, T., Zhang, H., Chen, H., Song, Z., Hsieh, C., Boning, D., Dhillon, I., and Daniel, L. (2018). Towards fast computation of certified robustness for ReLU networks. In *35th International Conference on Machine Learning*. PMLR.

Willers, O., Sudholt, S., Raafatnia, S., and Abrecht, S. (2020). Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*. Springer International Publishing.

Wong, E. and Kolter, Z. (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. In *35th International Conference on Machine Learning*. PMLR.

Wong, E., Rice, L., and Kolter, Z. (2020). Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020*.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *IEEE International Conference on Computer Vision (ICCV)*.

Xu, X., Zhao, H., and Jia, J. (2020). Dynamic divide-and-conquer adversarial training for robust semantic segmentation. arXiv. Last accessed 20 April 2021.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.