Robust Semantic Segmentation by Redundant Networks With a Layer-Specific Loss Contribution and Majority Vote

Andreas Bär[1] Marvin Klingner[1] Serin Varghese[1,2]
Fabian Hüger[2] Peter Schlicht[2] Tim Fingscheidt[1]
{andreas.baer, m.klingner, s.varghese, t.fingscheidt}@tu-bs.de
{john.serin.varghese, fabian.hueger, peter.schlicht}@volkswagen.de

[1]Technische Universität Braunschweig [2]Volkswagen Group Automation

Abstract

The lack of robustness shown by deep neural networks (DNNs) questions their deployment in safety-critical tasks, such as autonomous driving. We pick up the recently introduced redundant teacher-student frameworks (3 DNNs) and propose in this work a novel error detection and correction scheme with application to semantic segmentation. It obtains its robustnesss by an online-adapted and therefore hard-to-attack student DNN during vehicle operation, which builds upon a novel layer-dependent inverse feature matching (IFM) loss. We conduct experiments on the Cityscapes dataset showing that this loss renders the adaptive student to be more than 20% absolute mean intersection-over-union (mIoU) better than in previous works. Moreover, the entire error correction virtually always delivers the performance of the best non-attacked network, resulting in an mIoU of about 50% even under strongest attacks (instead of 1...2%), while keeping the performance on clean data at about original level (ca. 75.7%).

1. Introduction

Methods based on deep neural networks (DNNs) excel in benchmarks throughout several computer vision tasks, including image classification [29, 33, 63] and semantic segmentation [15, 41, 82, 83]. The latter can be seen as an extension of image classification to the task of pixel classification. Most semantic segmentation architectures are based upon *fully convolutional networks* introduced by Long *et al*. [46]. While past works primarily concentrated on exclusively increasing the performance [14, 68, 81, 83], today's research is dominated by approaches focusing on increasing the computational efficiency of semantic segmentation architectures [19, 38, 48, 57, 76, 80].

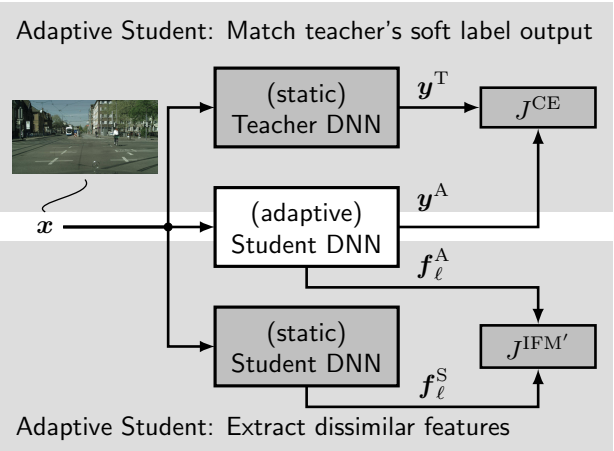


Figure 1: Overview of the proposed teacher-student framework. All three semantic segmentation networks are fed with the same input image $\boldsymbol{x}$. Both static networks in combination with their respective losses serve as learning objectives for the adaptive student. The static teacher forces the adaptive student's soft outputs $\boldsymbol{y}^{\mathrm{A}}$ to match the static teacher's soft outputs $\boldsymbol{y}^{\mathrm{T}}$ using the cross entropy loss J^{CE}. The static student on the other hand forces the adaptive student's feature representations $\boldsymbol{f}_\ell^{\mathrm{A}}$ at layer ℓ to be different from the respective static student's feature representation $\boldsymbol{f}_\ell^{\mathrm{S}}$ at layer ℓ, using a layer-dependent inverse feature matching loss $J^{\mathrm{IFM}'}$. For a practical implementation in the vehicle, the idea is that the adaptive student is in online learning mode, therefore hard-to-attack, and output error correction of any of the attacked two static networks is performed by a majority vote.

Closely related to increasing DNNs' computational efficiency are methods for teacher-student learning, also often referred to as model compression [4, 9]. The original idea is to compress the knowledge of a computationally complex teacher DNN into a computationally efficient small student DNN. This can be done by either matching the teacher's output exclusively [4, 32, 39, 49] or combining it with meth-

ods for feature guidance [40, 61, 67, 74, 75, 78]. Many works successfully applied teacher-student learning to semantic segmentation [6, 30, 43, 71, 73] showing its power and generalization across tasks.

DNNs have been tested thoroughly on clean data. However, for safety-critical approaches, *e.g*., environment perception in autonomous driving, not only the performance on clean data but also the robustness to input perturbations is an important criterion. Szegedy *et al*. [64] showed that DNNs can be easily fooled by perturbed data in the form of adversarial examples (AEs). This led to a rush in developing novel sophisticated adversarial attack algorithms [11, 23, 52, 55]. Looking at the task of semantic segmentation, Arnab *et al*. [1] showed that many state-of-the-art architectures for semantic segmentation are vulnerable to simple adversarial attacks [23, 36, 37] that have been initially introduced for the task of image classification. To increase the robustness against AEs, some works proposed defense strategies, including adversarial training [24, 47], feature denoising layers [31, 56, 70], or different kinds of (gradient masking) pre-processing strategies [25, 34, 44, 59].

Recently, Bär *et al*. [6] proposed an AE defense technique using redundant teacher-student frameworks for semantic segmentation. The authors assume that a possible intruder can very easily access the weights of a model which is already trained and fixed. Adaptive models on the other hand impede the intruder in fooling a system [22]. In general, their proposed redundant teacher-student framework incorporates a static teacher DNN and two architecturally identical student DNNs: A static one and an adaptive one. The adaptive student distills knowledge from the static teacher, while simultaneously being penalized by an inverse feature matching (IFM) loss computed across *all* layers to leverage dissimilarity to the static student's features. This way, the adaptive student obtains a certain level of robustness to AEs generated from static model knowledge, namely the static teacher and the static student. The idea is to create a model, which is robust to both, static teacher AEs *and* static student AEs and thus can serve as a watchdog allowing schemes for AE detection or even for correction by comparing the semantic segmentation outputs among the three networks. Nonetheless, using the IFM loss comes with the price of a significantly worse performance on clean data. Additionally, only a marginal increase in the robustness towards strong AEs can be observed.

In this work, our contributions are as follows: First, we outline that AEs have strong similarities to general error propagation. Thus, the IFM loss computation across *all* layers is identified as the main reason for the performance drop on clean data observed in [6]. This motivates the idea of a *layer-dependent* IFM loss to mitigate the error in form of adversarial examples. The proposed loss yields to significantly better results on clean and perturbed data compared to the results in [6]. Second, we introduce a simple yet effective AE detection function in a general form. Combined with an adaptive student trained with the layer-dependent IFM loss, it allows to detect the current static DNN under attack using the adaptive student as a watchdog. As a post-processing, one might perform a simple AE correction by choosing one of the non-attacked DNNs as the (corrected) output. Finally, we test both our proposed approaches on the Cityscapes dataset showing their effectiveness.

2. Related Works

In the following section, the related works in three specific fields of research are introduced: *semantic segmentation*, *teacher-student learning*, and *robustness*.

Semantic segmentation. Today's deep neural networks (DNNs) for semantic segmentation are mostly based on the concept of fully convolutional networks [46]. Increasing the spatial context by incorporating dilated convolutions [13, 77], recurrent neural networks [83], forms of spatial pyramid pooling [14, 15, 28, 81], conditional random fields [13, 14, 35, 66], and different schemes for intermediate skip-connections [7, 15, 72] can further increase the performance. As pixelwise labeling is expensive, some approaches propose unsupervised domain adaptation [8] or label relaxation techniques [82] to further boost the performance with unlabeled images or videos.

Efficiency-oriented algorithms rely on factorization of the convolution operation [48, 60], depthwise seperable convolution [16] with inverted residual units [62], in-place batch normalization [10], or a more efficient architectural design in general [38, 57, 76, 80].

In this work, an efficient DNN [60] is combined with a non-efficient DNN [8, 45] for *teacher-student learning*.

Teacher-student learning. The most basic concept of teacher-student learning is to match the output of the student DNN with the output of the teacher DNN [4, 32, 39, 49]. Further work extended the idea of output matching by guiding and hint layers as a pre-step [61], direct matching of feature representations [40, 75, 78], flow of solution procedure as a form of internal feature transformation matching [74], integrating a GAN discriminator for feature alignment [67], and self-distillation [21, 79]. While being firstly introduced for the task of image classification or speech recognition, the field of teacher-student learning eventually expanded to more complex tasks, such as semantic segmentation [6, 30, 43, 71, 73].

In this work, a variation of the inverse feature matching (IFM) loss [6] is used and combined with a form of teacher output matching [39] for teacher-student learning.

Robustness. The robustness of algorithms relying on DNNs is heavily questioned since Szegedy *et al*. [64] discovered the existence of adversarial examples (AEs). Since then, the research in crafting AEs further evolved and

produced various kinds of adversarial attack algorithms. FGSM [24], DeepFool [52], LLCM [36, 37], C&W [11], MI-FGSM [18], and PGD [47] can be classified as image-dependent adversarial attack algorithms aiming at producing one perturbation per input image. UAP [51], FFF [55], and PD-UA [42] on the other hand aim at producing one exclusive perturbation to fool a respective DNN on a bunch of images. Moreover, the transferability of AEs to more complex tasks, such as semantic segmentation, is in general possible [1, 54], with some works even focusing on task-specific adversarial attack algorithms [2, 20, 50, 69].

As the research in adversarial attacks progressed, different techniques were proposed to mitigate the success rate of AEs, *e.g*., adversarial training [24, 47], robustness-oriented loss functions [12, 53], feature denoising layers [31, 56, 70], redundant teacher-student frameworks [6], and various kinds of (gradient masking) pre-processing strategies [5, 25, 26, 34, 44, 59, 65]. Nonetheless, *e.g*., Athalye *et al*. [3] showed that gradient masking is not a sufficient criterion for a reliable defense strategy.

In this work, the robustness strategy is built upon redundant teacher-student frameworks [6] and is expanded by an AE detection function as a form of majority vote.

3. Method

This section gives a short overview of the mathematical notation and our proposed approaches.

3.1. Mathematical Notation

Let $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{G}^{H \times W \times C}$ be an image with height H, width W, number of color channels $C = 3$, set of integer gray values $\mathbb{G}$, and dataset $\mathcal{X}$. The image $\boldsymbol{x}$ serves as an input for a neural network $\mathfrak{F}(\cdot)$ with network parameters $\boldsymbol{\theta}$ and output $\boldsymbol{y} = \mathfrak{F}(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{I}^{H \times W \times |\mathcal{S}|}$, with the set of classes $\mathcal{S}$, and $\mathbb{I} = [0, 1]$. Each element of $\boldsymbol{y}$ is considered to be a posterior probability $y_{i,s} = P(s|i, \boldsymbol{x})$ for the class $s \in \mathcal{S}$ at pixel index $i \in \mathcal{I} = \{1, 2, ..., H \cdot W\}$. The architecture of a neural network $\mathfrak{F}(\cdot)$ consists of several layers $\ell \in \mathcal{L}$, each having feature representations $\boldsymbol{f}_\ell \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$, with height H_ℓ, width W_ℓ, and number of feature maps C_ℓ.

As the setup in this paper unites three networks in total, namely a static teacher network (T), a static student network (S), and an adaptive student network (A), the superscript $h \in \mathcal{H} = \{\mathrm{T}, \mathrm{S}, \mathrm{A}\}$ is introduced to differentiate between those three networks. Note, whenever the superscript is omitted, the general case is referred.

3.2. Background

As this work is closely related to [6] and reuses certain aspects of it, a short overview of their approach is provided.

Bär *et al*. [6] propose the use of a teacher-student setup consisting of three networks in total, namely the static teacher network $\mathfrak{F}^{\mathrm{T}}(\cdot)$, the static student network $\mathfrak{F}^{\mathrm{S}}(\cdot)$, and the adaptive student network $\mathfrak{F}^{\mathrm{A}}(\cdot)$. Note that both student networks have the exact same network architecture, whereas the teacher network has a different one.

First, all three networks are pretrained on the same labeled dataset $\mathcal{X}_{\mathrm{labeled}}$. As a next step, the parameters of the teacher as well as the parameters of one of the students are frozen to obtain two static networks. Furthermore, sets of adversarial examples (AEs) are computed for both static networks using (iterative) LLCM [36, 37] (see Section 3.4). Now, the adaptive student is initialized using the trained static student parameters, so that $\boldsymbol{\theta}^{\mathrm{A}}_{t=0} = \boldsymbol{\theta}^{\mathrm{S}}_{t=\mathrm{trained}}$ holds for the initialization step $t = 0$. Next, the adaptive student is further finetuned for 10 epochs using two losses, namely the cross entropy (CE) loss and the inverse feature matching (IFM) loss, computed on an unlabeled dataset $\mathcal{X}_{\mathrm{unlabeled}}$. The cross entropy loss J^{CE} is defined as

$$J^{\mathrm{CE}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} y^{\mathrm{T}}_{i,s} \cdot \log\left(y^{\mathrm{A}}_{i,s}\right), \tag{1}$$

with $y^{\mathrm{T}}_{i,s}$ and $y^{\mathrm{A}}_{i,s}$ being the respective posterior probability of the static teacher and the adaptive student for the class s at pixel index i. The inverse feature matching loss J^{IFM} is defined as

$$J^{\mathrm{IFM}} = \left(\beta + \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{||\boldsymbol{f}^{\mathrm{A}}_\ell - \boldsymbol{f}^{\mathrm{S}}_\ell||^p_p}{H_\ell \cdot W_\ell \cdot C_\ell}\right)^{-\gamma}, \tag{2}$$

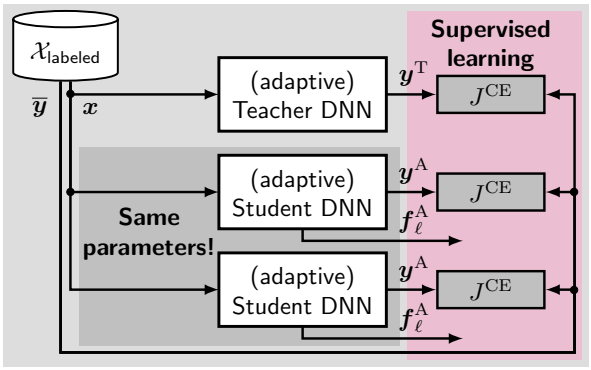
where $\boldsymbol{f}^{\mathrm{A}}_\ell$ and $\boldsymbol{f}^{\mathrm{S}}_\ell$ are the feature representations of the adaptive student and the static student at layer ℓ, respectively, when being fed by $\boldsymbol{x}$, as well as $p, \beta, \gamma \in \mathbb{R}^+$ being hyperparameters. The authors of [6] achieved the best results by setting $p = 1$, $\beta = 0.5$, and $\gamma = 1$. Combining (1) and (2) results in the teacher-student (TS) loss

$$J^{\mathrm{TS}} = (1 - \alpha) \cdot J^{\mathrm{CE}} + \alpha \cdot J^{\mathrm{IFM}}, \tag{3}$$

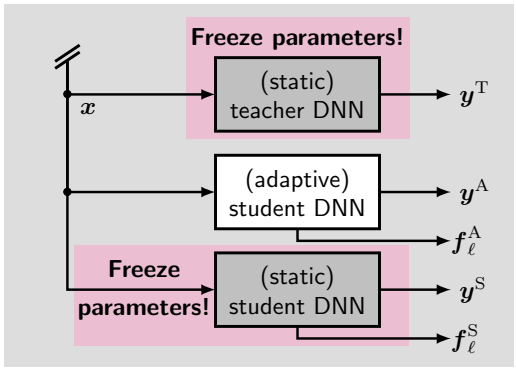
with the loss weighting factor α. Here, Bär *et al*. [6] observed that $\alpha = 0.6$ shows the best trade-off between peformance drop on clean data and robustness to the beforehand computed static teacher and static student AEs. The IFM loss increases the distance between the layer-wise feature representations of both students, which is bounded by the CE loss between adaptive student outputs and static teacher outputs. The described training procedure is also illustrated in Figs. 2a, 2b, and 2c. For further information about the exact training procedure and setup, please refer to [6].

3.3. Detection and Correction

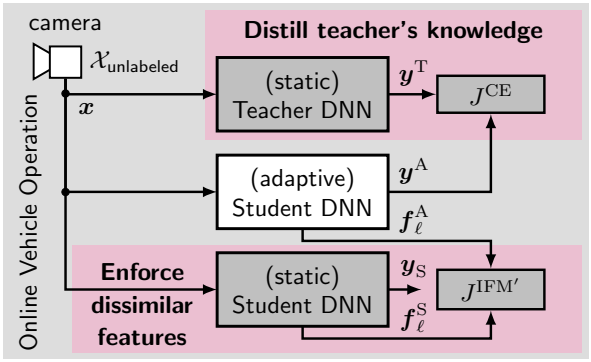
After following the training procedure in Section 3.2, one can use this system to detect and correct AEs during inference as illustrated in Fig. 2d. Note that in practice, the finetuning step (cf. Fig. 2c) is continuously ongoing in an online learning protocol, while the inference step



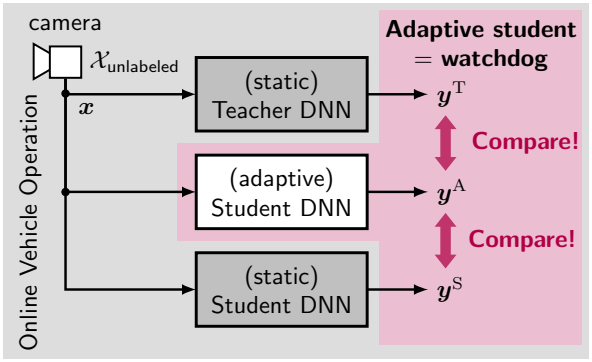
(a) Training step 1: All three networks



(b) Training step 2: Parameter freezing



(c) Test step 1: Adaptive student finetuning



(d) Test step 2: Inference and majority vote

Figure 2: General concept of the **proposed teacher-student framework**. **Training step 1**: Pretrain all deep neural networks (DNNs) on labeled data $\boldsymbol{x} \in \mathcal{X}_{\text{labeled}}$, with $\overline{\boldsymbol{y}}$ being the respective ground truth label of $\boldsymbol{x}$, using the cross-entropy (CE) loss J^{CE} between the ground truth $\overline{\boldsymbol{y}}$ and the respective DNNs' outputs $\boldsymbol{y}^{\text{T}}, \boldsymbol{y}^{\text{A}}$. Note that both student DNNs have the exact same architecture and parameters. **Training step 2**: Freeze the parameters of the teacher DNN and one student DNN. Both are now considered being static. **Test step 1**: Finetune the adaptive student DNN on unlabeled data $\boldsymbol{x} \in \mathcal{X}_{\text{unlabeled}}$ using the CE loss J^{CE} between the static teacher's soft outputs $\boldsymbol{y}^{\text{T}}$ and the adaptive student's soft outputs $\boldsymbol{y}^{\text{A}}$, and the layer-dependent inverse feature matching (IFM$'$) loss $J^{\text{IFM}'}$ between the static student's feature representations $\boldsymbol{f}_\ell^{\text{S}}$ and the adaptive student's feature representations $\boldsymbol{f}_\ell^{\text{A}}$ at layer ℓ. **Test step 2**: During inference, use the adaptive student DNN as a watchdog by comparing its output $\boldsymbol{y}^{\text{A}}$ with the static DNNs' outputs $\boldsymbol{y}^{\text{T}}$, $\boldsymbol{y}^{\text{S}}$. Whenever there is more similarity, the respective static network output is taken as the corrected semantic segmentation output. Test steps 1 and 2 are alternatingly executed in the online watchdog application of the adaptive student DNN, *e.g.*, in a vehicle (no labels required!). For the experimental validation in this paper, test step 1 is cast to the training (training step 3) and the alternating call schedule is omitted.

(cf. Fig. 2d) is executed in an alternating fashion. However, for evaluation in this paper, we do not perform the alternating schedule and execute both steps just once.

The detection is done by comparing the adaptive student's output on the one hand with the static teacher's output, and on the other hand with the static student's output by some distance metric $D(\cdot)$, implemented, *e.g.*, as a dissimilarity counter. The AE detection function in general form is defined as

$$d^{\text{AE}} = D(g(\boldsymbol{y}^{\text{T}}), g(\boldsymbol{y}^{\text{A}})) - D(g(\boldsymbol{y}^{\text{S}}), g(\boldsymbol{y}^{\text{A}})) \gtrless \delta, \quad (4)$$

with $g(\cdot)$ being a possible pre-processing step on the DNN outputs, and δ being a threshold. For $d^{\text{AE}} > \delta$, the adaptive student's output is more similar to the static student's output, pointing to a possible adversarial attack on the static teacher, whereas for $d^{\text{AE}} < \delta$ the adaptive student's output is considered to be more similar to the static teacher's output, pointing to a possible adversarial attack on the static student. In practice, it might be reasonable to consider DNN-specific thresholds $\delta^{\text{T}}, \delta^{\text{S}} \in \mathbb{R}$, *i.e.*, for $d^{\text{AE}} > \delta^{\text{T}}$ the static teacher is being attacked, while for $d^{\text{AE}} < \delta^{\text{S}}$ the static student is being attacked, otherwise no attack is assumed. After AE detection by the sketched majority vote, the AE correction is simply done by discarding the respective DNN being currently attacked and selecting the remaining network output with better clean input performance.

3.4. Adversarial Attack Design

The *iterative* least-likely class method (LLCM) described in [36, 37] is chosen to generate adversarial examples (AEs). An adversarial example is defined as

$$\boldsymbol{x}^{\mathrm{adv}} = \boldsymbol{x} + \boldsymbol{r}, \tag{5}$$

with the adversarial perturbation $\boldsymbol{r}$, which is bounded by

$$||\boldsymbol{r}||_\infty \leq \epsilon, \tag{6}$$

where ϵ is the upper bound of the adversarial perturbation's infinity norm $||\boldsymbol{r}||_\infty$. Following [36, 37], the adversarial examples are computed over $\lceil \min(\epsilon + 4, 1.25\epsilon) \rceil$ iterations, where in each step the value of each pixel is only changed by $\lambda = 1$. For our experiments, we generate sets of teacher adversarial examples (T-AEs) and student adversarial examples (S-AEs) to obtain both a weak and a strong attack, in accordance to [6]. For further detail, please refer to [6].

3.5. New Layer-Dependent IFM Loss

In this section, our novel approach is introduced by a motivation and is then explained more in detail.

Motivation. The IFM loss increases the robustness of the adaptive student to static student adversarial examples (S-AEs) by enforcing its feature representations at all layers to have a large distance to the respective static student ones. This in fact does increase the robustness to S-AEs, but at the same time also harms the performance on clean data.

We hypothesize that computing the IFM loss over the entire set of layers is not at all needed for better robustness towards adversarial attacks. We motivate the idea of layer-dependent computation by the fact that adversarial examples initiate an error propagation through a DNN. *With this assumption, it should be sufficient enough to only penalize a few layers using the IFM loss to mitigate the effect of an adversarial example.*

Loss. Following our motivation, layer set $\mathcal{L}$ in (2) is simply replaced by $\mathcal{L}'$ to obtain the *layer-dependent* inverse feature matching (IFM$'$) loss

$$J^{\mathrm{IFM}'} = \left(\beta + \frac{1}{|\mathcal{L}'|} \sum_{\ell \in \mathcal{L}'} \frac{||\boldsymbol{f}_\ell^{\mathrm{A}} - \boldsymbol{f}_\ell^{\mathrm{S}}||_p^p}{H_\ell \cdot W_\ell \cdot C_\ell} \right)^{-\gamma}, \tag{7}$$

with $\mathcal{L}' \subseteq \mathcal{L}$ being a subset of all layers. Throughout our experiments we chose $p = 1$, $\beta = 0.5$, and $\gamma = 1$, following [6]. Using the IFM$'$ loss, the overall TS loss (3) changes to

$$J^{\mathrm{TS}'} = (1 - \alpha) \cdot J^{\mathrm{CE}} + \alpha \cdot J^{\mathrm{IFM}'}. \tag{8}$$

4. Experiments

In this section, the dataset used in the experiments is introduced and the results are shown and discussed.

	mIoU	mIoU ratio Q			
	$\epsilon = 0$	$\epsilon = 1$		$\epsilon = 10$	
model	clean	T-AE	S-AE	T-AE	S-AE
T	**75.43**	23.35	**95.07**	2.21	**70.05**
S	66.06	**90.08**	14.94	**64.82**	1.63

Table 1: **mIoU** [in %] and **mIoU ratio** [in %] on variations of the Cityscapes *mini val set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). **Results are reported for the static teacher network (T) and static student network (S)** after being pretrained. Best numbers are printed in **bold**. The mIoU numbers in the first column are the same as in [6].

4.1. Datasets

The experiments are conducted on the widely-known Cityscapes dataset [17]. The 2,975 images of the official finely annotated Cityscapes training set are used as the labeled dataset $\mathcal{X}_{\mathrm{labeled}}$ for pretraining, and the 19,998 images of the official coarsely annotated Cityscapes training set are used as the unlabeled dataset $\mathcal{X}_{\mathrm{unlabeled}}$ for finetuning. Note that the latter is used to emulate the online application of the redundant teacher-student framework. First, the adaptive student DNN is finetuned on unlabeled data and then used as a watchdog for the static teacher DNN and the static student DNN (cf. Figs 2c, 2d).

The experimental evaluation is performed by using the (mean) intersection-over-union (mIoU)

$$\mathrm{mIoU} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{\mathrm{TP}_s}{\mathrm{TP}_s + \mathrm{FP}_s + \mathrm{FN}_s}, \tag{9}$$

with the class-specific true positives TP_s, false positives FP_s and false negatives FN_s. Additionally, the robustness is also reported using the mIoU ratio

$$Q = \frac{\mathrm{mIoU}_{\mathrm{adv}}}{\mathrm{mIoU}_{\mathrm{clean}}}, \tag{10}$$

with $\mathrm{mIoU}_{\mathrm{clean}}$ being the mIoU on clean data and $\mathrm{mIoU}_{\mathrm{adv}}$ being the mIoU on the respective LLCM-perturbed data.

The images captured in Lindau (59 images) are used as a mini validation set, and the images captured in Frankfurt and Münster (441 images) as the mini test set, following Bär *et al*. [6]. The adversarially perturbed validation sets were generated by applying LLCM from Section 3.4 on the static teacher network and static student network. The respective sets are referred to as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs).

4.2. Results

In this section, the experimental evaluation is presented and discussed.

IFM′ loss (7) in ... down 1st	2nd	3rd	up 1st	2nd	rest	mIoU $\epsilon = 0$ clean	$\epsilon = 1$ T-AE	$\epsilon = 1$ S-AE	$\epsilon = 10$ T-AE	$\epsilon = 10$ S-AE
✓	✓	✓	✓	✓	✓	54.10	49.27	31.80	31.39	13.96
✓	✓	✓	✓	✓		60.77	59.16	43.60	48.05	26.40
					✓	50.37	42.09	24.65	23.22	8.43
✓	✓	✓				54.55	52.88	49.50	46.33	37.08
			✓	✓		**65.05**	55.94	14.42	35.80	2.63
✓						63.19	**59.71**	46.44	48.61	30.62
	✓					61.00	58.81	**51.87**	**49.61**	**37.19**
		✓				60.02	54.42	30.43	37.72	8.62

Table 2: **mIoU** [in %] on variations of the Cityscapes *mini val set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). **Results are reported for the adaptive student network (A)** trained with (8) on different subsets of layers ("down" = downsampling layers; "up" = upsampling layers; "rest" = all other layers), keeping $\alpha = 0.6$. For more details about the exact architecture, please refer to [60]. Best numbers are printed in **bold**. The mIoU value of 54.10% is the same as in [6].

IFM′ loss in ... down 1st	2nd	α	mIoU $\epsilon = 0$ clean	$\epsilon = 1$ T-AE	$\epsilon = 1$ S-AE	$\epsilon = 10$ T-AE	$\epsilon = 10$ S-AE	mean
✓		0.5	**63.46**	**59.81**	45.72	48.05	29.26	49.26
✓		0.6	63.19	59.71	46.44	**48.61**	30.62	49.71
✓		0.7	60.25	56.48	**49.94**	47.84	**36.35**	**50.17**
✓		0.8	54.93	52.87	46.98	44.79	35.10	46.93
✓		0.9	28.82	27.18	26.12	22.17	22.36	25.33
	✓	0.5	60.52	58.28	49.00	48.07	32.15	49.60
	✓	0.6	**61.00**	**58.81**	51.87	**49.61**	37.19	**51.70**
	✓	0.7	59.00	56.69	**52.45**	48.60	40.23	51.39
	✓	0.8	55.98	54.76	50.59	46.82	39.93	49.62
	✓	0.9	54.41	53.36	50.86	46.73	**40.45**	49.16

Table 3: **mIoU** [in %] on variations of the Cityscapes *mini val set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). **Results are reported for the adaptive student network (A)** trained with (8) on different subsets of layers ("down" = downsampling layers) and different values for α. For more details about the exact architecture, please refer to [60]. Best numbers are printed in **bold**.

Pretraining. First, all three networks are pretrained using the finely annotated Cityscapes training set, while evaluation takes place on variations of the Cityscapes mini validation set. The results are shown in Tab. 1. Note that just after pretraining, the static student and adaptive student have exactly the same parameters (cf. 2b), thus only the numbers for the static student (S) and static teacher (T) are reported.

As expected, the static teacher performs better on clean data than the static student. Looking at the robustness towards AEs, both networks are vulnerable to their respective AEs, but are more or less robust to the counter part's AEs. Compared to the static teacher, the static student performs worse when being exposed to its respective weak/strong AEs (cf. mIoU ratios of model S and S-AEs, 14.94% / 1.63%, compared to model T and T-AEs, 23.35% / 2.21%, in Tab. 1) and when being exposed to the counter part's weak/strong AEs (cf. mIoU ratios of model S and T-AEs, 90.08% / 64.82%, compared to model T and S-AEs, 95.07% / 70.05%, in Tab. 1).

Finetuning. After pretraining, the parameters of the static teacher and the static student are frozen. For the first experiments, the adaptive student is finetuned for 10 epochs setting $\alpha = 0.6, \beta = 0.5, \gamma = 1$ and $p = 1$ in (8), while varying the set of layers $\mathcal{L}'$ in (7). Here, the downsampling layers and upsampling layers of the ERFNet [60] are chosen to compute the IFM′ loss for a better transferability of our approach to other models. Note that the last upsampling layer is intentionally not used as it serves as the final classification layer. The results are shown in Tab. 2.

First, by exclusively using all the downsampling layers and upsampling layers for the IFM′ loss computation in (7) (cf. second row in Tab. 2), our approach already *beats the baseline from [6] in all categories* (cf. first row in Tab. 2). Second, using all layers, except the downsampling and the upsampling layers, seem to significantly harm the performance on clean and perturbed data in general (cf. third row in Tab. 2). Third, comparing the exclusive incorporation of all downsampling layers (cf. fourth row in Tab. 2) with the exclusive incorporation of all upsampling layers (cf. fifth row in Tab. 2) shows that the latter performs significantly better on clean data, while the former is significantly better across all forms of perturbed data, namely weak/strong T-AEs and S-AEs. Additionally, the configuration with IFM′ loss in all downsampling layers *still outperforms our baseline in all categories and has significantly higher robustness to both weak and strong S-AEs*. Fourth, looking at the downsampling layers separately, it can be observed that exclusively incorporating the second downsampling layer (cf. last but one row in Tab. 2) balances high performance on clean data and high robustness on perturbed data in a reasonable manner, as it shows the best numbers in 3 out of 5 cases, *i.e.*, weak/strong S-AEs, and strong T-AEs.

Next, as $\alpha = 0.6$ was optimized for the baseline from [6], we also experiment with different values for α. Here, the first two downsampling layers are chosen, as they both show bold numbers in Tab. 2 for one or more sets of AEs. The results for the α-experiments are shown in Tab. 3. Looking at experiments with the first downsampling layer (cf. upper half of Tab 3), increasing α increases the robustness towards weak and strong S-AEs, however, it also heav-

δ	model	$\epsilon = 0$ clean	$\epsilon = 1$ T-AE	$\epsilon = 1$ S-AE	$\epsilon = 10$ T-AE	$\epsilon = 10$ S-AE
0.0	T	**40**	0	**58**	0	**59**
	S	19	**59**	1	**59**	0
0.1	T	**59**	0	**59**	0	**59**
	S	0	**59**	0	**59**	0
0.2	T	**59**	1	**59**	0	**59**
	S	0	**58**	0	**59**	0
0.5	T	**59**	19	**59**	0	**59**
	S	0	**40**	0	**59**	0

Table 4: **AE detection and correction** (see Section 3.3) with different threshold values δ on variations of the Cityscapes *mini val set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). The reported numbers correspond to the **number of images, where either the static teacher's output (T) or the static student's output (S) have a smaller Hamming distance to the adaptive student's output (A), *i.e.*, $d^{\mathrm{AE}} > \delta$ or** $d^{\mathrm{AE}} < \delta$. An adaptive student with IFM$'$ loss (7) computed on the second downsampling layer setting $\alpha = 0.6$ in (8) was chosen. Desired behavior in **bold**.

ily harms the performance on clean data. Looking at the experiments with the second downsampling layer (cf. lower half of Tab 3), it can be observed that increasing α also harms the performance on clean data, but not as hard as it was the case for the first downsampling layer. Note that *all* configurations with the second downsampling layer outperform the baseline from [6] (cf. first row in Tab. 3). For a better comparison, we additionally compute the mean over all five mIoU values in one row (cf. last column in Tab. 3).

Finally, we conclude that incorporating the second downsampling layer with $\alpha = 0.6$ gives the best results, with 61.00% mIoU on clean data, 58.81% / 49.61% mIoU on weak/strong T-AEs, and 51.87% / 37.19% mIoU on weak/strong S-AEs, respectively, resulting in a mean of 51.70%.

AE detection and correction. For the experiments validating the AE detection and correction mechanism from Section 3.3, the adaptive student trained with (8) on the second downsampling layer setting $\alpha = 0.6$ (cf. second row in lower half of Tab. 3) is chosen as a watchdog. Additionally, we use the argmax$(\cdot)$ operation on the posterior outputs $\boldsymbol{y}^{\mathrm{T}}$, $\boldsymbol{y}^{\mathrm{S}}$, $\boldsymbol{y}^{\mathrm{A}}$ as $g(\cdot)$ in (4) to obtain the most-likely class for each pixel, and combine it with the Hamming distance [27, 58] as distance metric $D(\cdot)$ in (4). The results are shown in Tab. 4.

By naively setting $\delta = 0.0$, *almost all* AEs are correctly detected, *i.e.*, whenever the static teacher is exposed to his respective AEs, the static student has a smaller Hamming distance to the adaptive student, and vice versa. However,

	mIoU				
	$\epsilon = 0$	$\epsilon = 1$		$\epsilon = 10$	
method	clean	T-AE	S-AE	T-AE	S-AE
[6]	53.01	45.81	28.55	27.74	11.14
ours	**61.12**	**58.60**	**48.39**	**48.67**	**33.38**

Table 5: **mIoU** [in %] on variations of the Cityscapes *mini test set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). **Results are reported for the adaptive student network (A)** being finetuned with the method from [6] and our here proposed approach, with IFM$'$ loss (7) computed on the second downsampling layer setting $\alpha = 0.6$ in (8). Best numbers are printed in **bold**. The mIoU value of 53.01% is the same as in [6].

when looking at the clean data case ($\epsilon = 0$), the adaptive student seems to be biased towards the static student due to the same architecture and pretraining. This can be mitigated by increasing the threshold to $\delta = 0.1$, where it can be observed that now *all* AEs are correctly detected and additionally *the static teacher is selected over the static student when being fed with clean data*. However, further increasing the threshold slowly leads to a bias towards the static teacher.

Final Results. For the final experiments, the adaptive student is finetuned for 10 epochs by computing the IFM$'$ loss in (7) exclusively on its second downsampling layer and setting $\alpha = 0.6$ in (8). We compare this model with the baseline model from [6] on the Cityscapes mini test set. The results are shown in Tab. 5. Similar to the previous observations, *our approach manages to significantly improve the baseline in all categories, namely performance on clean data and robustness towards static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs).* The biggest improvements are obtained in the robustness towards strong T-AEs as well as weak and strong S-AEs (each better by about 20% absolute mIoU).

Additionally, we also provide results for the AE detection and correction mechanism from Section 3.3 on the Cityscapes mini test set, with $g(\cdot)$ being the argmax taken individually over the posterior probabilities $\boldsymbol{y}^{\mathrm{T}}$, $\boldsymbol{y}^{\mathrm{S}}$, $\boldsymbol{y}^{\mathrm{A}}$, Hamming distance $D(\cdot)$, and threshold δ. The results are shown in Tab. 6. Similar to the observations in Tab. 4, depending on the threshold δ, *our proposed adversarial example detection and correction mechanism manages to correctly detect all adversarial examples (AEs), while only having very few false positives on clean data*. For the optimal $\delta = 0.1$ (as obtained on the mini val set, see Tab. 4) only a single error in total has been observed on the mini test set.

Using the proposed majority vote with $\delta = 0.1$, the mIoU numbers of the corrected network ensemble output are displayed in Tab. 7. Here, two configurations for major-

δ	model	$\epsilon = 0$ clean	$\epsilon = 1$ T-AE	$\epsilon = 1$ S-AE	$\epsilon = 10$ T-AE	$\epsilon = 10$ S-AE
0.0	T	**388**	0	**441**	0	**440**
	S	53	**441**	0	**441**	1
0.1	T	**440**	0	**441**	0	**441**
	S	1	**441**	0	**441**	0
0.2	T	**441**	0	**441**	0	**441**
	S	0	**441**	0	**441**	0
0.5	T	**441**	155	**441**	4	**441**
	S	0	**286**	0	**437**	0

Table 6: **AE detection and correction** (see Section 3.3) with different threshold values δ on variations of the Cityscapes *mini test set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). The reported numbers correspond to the **number of images, where either the static teacher's output (T) or the static student's output (S) have a smaller Hamming distance to the adaptive student's output (A), *i.e.*,** $d^{\mathrm{AE}} > \delta$ **or** $d^{\mathrm{AE}} < \delta$. An adaptive student with IFM$'$ loss (7) computed on the second downsampling layer setting $\alpha = 0.6$ in (8) was chosen. Desired behavior in **bold**.

method	mIoU $\epsilon = 0$ clean	$\epsilon = 1$ T-AE	$\epsilon = 1$ S-AE	$\epsilon = 10$ T-AE	$\epsilon = 10$ S-AE
T	**75.77**	16.89	**72.12**	1.84	**54.02**
S	64.55	58.39	9.14	41.21	0.87
A	61.12	**58.60**	48.39	**48.67**	33.38
our T/S maj. vote	75.74	58.39	**72.12**	41.21	**54.02**
our T/A maj. vote	75.74	**58.60**	**72.12**	**48.67**	**54.02**

Table 7: **mIoU** [in %] **of the proposed majority vote** based on the network triplet computed on variations of the Cityscapes *mini test set* (see Section 4.1), *i.e.*, clean data ($\epsilon = 0$), as well as static teacher adversarial examples (T-AEs) and static student adversarial examples (S-AEs), both created by using LLCM (cf. Section 3.4, $\epsilon = \{1, 10\}$). The same static student (S) and static teacher (T) reported in Tab. 1 are taken. An adaptive student network (A) trained with IFM$'$ loss (7) computed on the second downsampling layer setting $\alpha = 0.6$ in (8) was chosen. The threshold δ in (4) for our proposed majority vote is set to 0.1 according to Tab. 4. All numbers for the majority vote are based on the results in Tab. 6.

ity vote are compared to each other. In the first one, T/S, the adaptive student is only used as a watchdog, *i.e.*, depending on the Hamming distance, either the static teacher or the static student is chosen (cf. Tab. 6). The T/S majority vote leads to the (shared) best numbers in two categories, *i.e.*, weak/strong S-AEs. However, as the adaptive student clearly performs better on perturbed data than the static student, it is reasonable to use the adaptive student not only as a watchdog, but also as a replacement for the static student, whenever the decision of the AE detection and correction would favor the static student. Using the T/A majority vote leads to 75.74% mIoU on clean data, 58.60% / 48.67% mIoU on weak/strong T-AEs, and 72.12% / 54.02% mIoU on weak/strong S-AEs, which is signficiantly better than exclusively using the static teacher or exclusively using the static student. In a nutshell, our proposed T/A majority vote using an ensemble of three networks keeps an mIoU of about 50% under all circumstances, while providing a high performance (75.74%) in case of no attack.

5. Conclusion

In this paper, we propose a novel error detection and correction scheme to mitigate the effect of adversarial examples on deep neural networks (DNNs) for semantic segmentation. To this end, we make use of the recently introduced redundant teacher-student frameworks, consisting of 3 DNNs, and emulate online adaptation using a novel layer-dependent inverse feature matching (IFM) loss to obtain an adaptive and thus hard-to-attack student DNN. Experiments on the Cityscapes dataset show that by incorporating this loss, the adaptive student is able to maintain an about 20% higher (absolute) mean intersection-over-union (mIoU) on perturbed data compared to previous works. In the end, our proposed error detection and correction scheme virtually always delivers the performance of the best non-attacked network, resulting in an mIoU of about 50% even under strong attacks (instead of 1...2%), while keeping the performance on clean data at about original level (ca. 75.7%).

Acknowledgement

The authors gratefully acknowledge support of this work by Volkswagen Group Automation, Wolfsburg, Germany. The research leading to the results presented above are funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for automated driving".

References

[1] Anurag Arnab, Ondrej Miksik, and Philip H. S. Torr. On the Robustness of Semantic Segmentation Models to Adversarial Attacks. In *Proc. of CVPR*, pages 888–897, Salt Lake City, UT, USA, June 2018.

[2] Felix Assion, Peter Schlicht, Florens Greßner, Wiebke Günther, Fabian Hüger, Nico M. Schmidt, and Umair Rasheed. The Attack Generator: A Systematic Approach Towards Constructing Adversarial Attacks. In *Proc. of CVPR - Workshops*, pages 1–12, Long Beach, CA, USA, June 2019.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumvent-

ing Defenses to Adversarial Examples. In *Proc. of ICML*, pages 274–283, Stockholm, Sweden, July 2018.

[4] Jimmy Ba and Rich Caruana. Do Deep Nets Really Need to Be Deep? In *Proc. of NIPS*, pages 2654–2662, Montréal, QC, Canada, Dec. 2014.

[5] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-Based Generative Defense for Adversarial Examples. In *Proc. of ICCV*, pages 4784–4793, Seoul, Korea, Oct. 2019.

[6] Andreas Bär, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On the Robustness of Teacher-Student Frameworks for Semantic Segmentation. In *Proc. of CVPR - Workshops*, pages 1–9, Long Beach, CA, USA, June 2019.

[7] Piotr Bilinski and Victor Prisacariu. Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation. In *Proc. of CVPR*, pages 6596–6605, Salt Lake City, UT, USA, June 2018.

[8] Jan-Aike Bolte, Andreas Bär, Daniel Lipinski, and Tim Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *Proc. of IV*, pages 366–373, Paris, France, June 2019.

[9] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model Compression. In *Proc. of KDD*, pages 535–541, Philadelphia, PA, USA, Aug. 2006.

[10] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. In *Proc. of CVPR*, pages 5639–5647, Salt Lake City, UT, USA, June 2018.

[11] Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *Proc. of SP*, pages 39–57, San Jose, CA, USA, May 2017.

[12] Hao-Yun Chen, Jhao-Hong Liang, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Improving Adversarial Robustness via Guided Complement Entropy. In *Proc. of ICCV*, pages 4881–4889, Seoul, Korea, Oct. 2019.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic Image Segmentation With Deep Convolutional Nets and Fully Connected CRFs. In *Proc. of ICLR*, pages 1–14, San Diego, CA, USA, May 2015.

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation With Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, Apr. 2018.

[15] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation. In *Proc. of ECCV*, pages 801–818, Munich, Germany, Sept. 2018.

[16] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proc. of CVPR*, pages 1063–6919, Honolulu, HI, USA, July 2017.

[17] 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 CVPR*, pages 3213–3223, Las Vegas, NV, USA, June 2016.

[18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. In *Proc. of CVPR*, pages 9185–9193, Salt Lake City, UT, USA, June 2018.

[19] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. BlitzNet: A Real-Time Deep Network for Scene Understanding. In *Proc. of ICCV*, pages 4174–4182, Venice, Italy, Oct. 2017.

[20] Volker Fischer, Mummadi Chaithanya Kumar, Jan Hendrik Metzen, and Thomas Brox. Adversarial Examples for Semantic Image Segmentation. In *Proc. of ICLR - Workshops*, pages 1–4, Toulon, France, Apr. 2017.

[21] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born Again Neural Networks. In *Proc. of ICML*, pages 1607–1616, Stockholm, Sweden, July 2018.

[22] Ian Goodfellow. A Research Agenda: Dynamic Models to Defend Against Correlated Attacks. In *Proc. of ICLR*, pages 1–9, New Orleans, LA, USA, May 2019.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Proc. of NIPS*, pages 2672–2680, Montréal, Canada, Dec. 2014.

[24] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proc. of ICLR*, pages 1–10, San Diego, CA, USA, May 2015.

[25] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering Adversarial Images using Input Transformations. In *Proc. of ICLR*, pages 1–12, Vancouver, BC, Canada, Apr. 2018.

[26] Puneet Gupta and Esa Rahtu. CIIDefence: Defeating Adversarial Attacks by Fusing Class-Specific Image Inpainting and Image Denoising. In *Proc. of ICCV*, pages 6708–6717, Seoul, Korea, Oct. 2019.

[27] Richard W. Hamming. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, 29(2):147–160, Apr. 1950.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1904–1916, Sept. 2015.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, pages 770–778, Las Vegas, NV, USA, June 2016.

[30] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge Adaptation for Efficient Semantic Segmentation. In *Proc. of CVPR*, pages 578–587, Long Beach, CA, USA, June 2019.

[31] Zhezhi He, Adnan S. Rakin, and Deliang Fan. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness Against Adversarial Attack. In *Proc. of CVPR*, pages 588–597, Long Beach, CA, USA, June 2019.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling Knowledge in a Neural Network. In *Proc. of NIPS - Workshops*, pages 1–9, Montréal, QC, Canada, Dec. 2014.

[33] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *Proc. of CVPR*, pages 4700–4708, Honulu, HI, USA, July 2017.

[34] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. In *Proc. of CVPR*, pages 6084–6092, Long Beach, CA, USA, June 2019.

[35] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Proc. of NIPS*, pages 109–117, Granada, Spain, Dec. 2011.

[36] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Examples in the Physical World. In *Proc. of ICLR - Workshops*, pages 1–14, Toulon, France, Apr. 2017.

[37] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. In *Proc. of ICLR*, pages 1–17, Toulon, France, Sept. 2017.

[38] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation. In *Proc. of CVPR*, pages 9522–9531, Long Beach, CA, USA, June 2019.

[39] Jinyu Li, Rui Zhao, Jui-Tang Huang, and Yifan Gong. Learning Small-Size DNN with Output-Distribution-Based Criteria. In *Proc. of INTERSPEECH*, pages 1910–1914, Singapore, Sept. 2014.

[40] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking Very Efficient Network for Object Detection. In *Proc. of CVPR*, pages 6356–6364, Honolulu, HI, USA, July 2017.

[41] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global Aggregation then Local Distribution in Fully Convolutional Networks. In *Proc. of BMVC*, pages 1–13, Cardiff, Wales, Sept. 2019.

[42] Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation. In *Proc. of ICCV*, pages 2941–2949, Seoul, Korea, Oct. 2019.

[43] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured Knowledge Distillation for Semantic Segmentation. In *Proc. of CVPR*, pages 2604–2613, Long Beach, CA, USA, June 2019.

[44] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples. In *Proc. of CVPR*, pages 860–868, Long Beach, CA, USA, June 2019.

[45] Jonas Löhdefink, Andreas Bär, Nico M. Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On Low-Bitrate Image Compression for Distributed Automotive Perception: Higher Peak SNR Does Not Mean Better Semantic Segmentation. In *Proc. of IV*, pages 352–359, Paris, France, June 2019.

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. of CVPR*, pages 3431–3440, Boston, MA, USA, June 2015.

[47] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. of ICLR*, pages 1–28, Vancouver, BC, Canada, Apr. 2018.

[48] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In *Proc. of ECCV*, pages 552–568, Munich, Germany, Sept. 2018.

[49] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional Teacher-Student Learning. In *Proc. of ICASSP*, pages 6445–6449, Brighton, England, May 2019.

[50] Jan H. Metzen, Mummadi C. Kumar, Thomas Brox, and Volker Fischer. Universal Adversarial Perturbations Against Semantic Image Segmentation. In *Proc. of ICCV*, pages 2774–2783, Venice, Italy, Oct. 2017.

[51] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal Adversarial Perturbations. In *Proc. of CVPR*, pages 1765–1773, Honolulu, HI, USA, July 2017.

[52] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proc. of CVPR*, pages 2574–2582, Las Vegas, NV, USA, June 2016.

[53] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via Curvature Regularization, and Vice Versa. In *Proc. of CVPR*, pages 9078–9086, Long Beach, CA, USA, June 2019.

[54] Konda R. Mopuri, Aditya Ganeshan, and Venkatesh B. Radhakrishnan. Generalizable Data-free Objective for Crafting Universal Adversarial Perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(10):2452–2465, Oct. 2019.

[55] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast Feature Fool: A Data Independent Approach to Universal Adversarial Perturbations. In *Proc. of BMVC*, pages 1–12, London, UK, Sept. 2017.

[56] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial Defense by Restricting the Hidden Space of Deep Neural Networks. In *Proc. of ICCV*, pages 3385–3394, Seoul, Korea, Oct. 2019.

[57] Marin Oršić, Ivan Krešo, Petra Bevandić, and Siniša Šegvić. In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images. In *Proc. of CVPR*, pages 12607–12616, Long Beach, CA, USA, June 2019.

[58] John G. Proakis. *Digital Communications*. McGraw-Hill Book Company, 1989.

[59] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of Random Transforms for Adversarially Robust Defense. In *Proc. of CVPR*, pages 6528–6537, Long Beach, CA, USA, June 2019.

[60] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation.

IEEE Transactions on Intelligent Transportation Systems, 19(1):263–272, Jan. 2018.

[61] Adriana Romero, Nicolas Ballas, Samira E. Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *Proc. of ICLR*, pages 1–13, San Diego, CA, USA, May 2015.

[62] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proc. of CVPR*, pages 4510–4520, Salt Lake City, UT, USA, June 2018.

[63] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang. FishNet: A Versatile Backbone for Image, Region, and Pixel Level Prediction. In *Proc. of NIPS*, pages 754–764, Montréal, QC, Canada, Dec. 2018.

[64] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. In *Proc. of ICLR*, pages 1–10, Montréal, QC, Canada, Dec. 2014.

[65] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness. In *Proc. of CVPR*, pages 6988–6986, Long Beach, CA, USA, June 2019.

[66] Raviteja Vemulapalli, Oncel Tuzel, Ming-Yu Liu, and Rama Chellappa. Gaussian Conditional Random Field Network for Semantic Segmentation. In *Proc. of CVPR*, pages 3224–3233, Las Vegas, NV, USA, June 2016.

[67] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Adversarial Learning of Portable Student Networks. In *Proc. of AAAI*, pages 4260–4267, New Orleans, LA, USA, Feb. 2018.

[68] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *arXiv*, Nov. 2016.

[69] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial Examples for Semantic Segmentation and Object Detection. In *Proc. of ICCV*, pages 1369–1378, Venice, Italy, Oct. 2017.

[70] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature Denoising for Improving Adversarial Robustness. In *Proc. of CVPR*, pages 501–507, Long Beach, CA, USA, June 2019.

[71] Jiafeng Xie, Bing Shuai, Jian-Fang Hu, Jingyang Lin, and Wei-Shi Zheng. Improving Fast Segmentation With Teacher-student Learning. In *Proc. of BMVC*, pages 1–13, Newcastle, England, Sept. 2018.

[72] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. DenseASPP for Semantic Segmentation in Street Scenes. In *Proc. of CVPR*, pages 3684–3692, Salt Lake City, UT, USA, June 2018.

[73] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student Becoming the Master: Knowledge Amalgamation for Joint Scene Parsing, Depth Estimation, and More. In *Proc. of CVPR*, pages 2829–2838, Long Beach, CA, USA, June 2019.

[74] Junho Yim, Donggyo Joo, Jihoon Bae, and Junmo Kim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *Proc. of CVPR*, pages 4133–4141, Honolulu, HI, USA, July 2017.

[75] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from Multiple Teacher Networks. In *Proc. of KDD*, pages 1285–1294, Halifax, NS, Canada, Aug. 2017.

[76] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In *Proc. of ECCV*, pages 325–341, Munich, Germany, Sept. 2018.

[77] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *Proc. of ICLR*, pages 1–13, San Juan, Puerto Rico, May 2016.

[78] Sergey Zagoruyko and Nikos Komodakis. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *Proc. of ICLR*, pages 1–13, Toulon, France, Apr. 2017.

[79] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *Proc. of ICCV*, pages 3713–3722, Seoul, Korea, Oct. 2019.

[80] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. In *Proc. of ECCV*, pages 405–420, Munich, Germany, Sept. 2018.

[81] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Proc. of CVPR*, pages 2881–2890, Honulu, HI, USA, July 2017.

[82] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *Proc. of CVPR*, pages 8856–8865, Long Beach, CA, USA, June 2019.

[83] Yueqing Zhuang, Fan Yang, Li Tao, Cong Ma, Ziwei Zhang, Yuan Li, Huizhu Jia, Xiaodong Xie, and Wen Gao. Dense Relation Network: Learning Consistent and Context-Aware Representation for Semantic Image Segmentation. In *Proc. of ICIP*, pages 3698–3702, Athens, Greece, Oct. 2018.