Increasing the Robustness of Semantic Segmentation Models with Painting-by-Numbers

Christoph Kamann[1,2] and Carsten Rother[2]

[1] Corporate Research, Robert Bosch GmbH, Renningen, Germany
christoph.kamann@bosch.com
[2] Visual Learning Lab, Heidelberg University (HCI/IWR), Heidelberg, Germany
carsten.rother@iwr.uni-heidelberg.de
http://vislearn.de

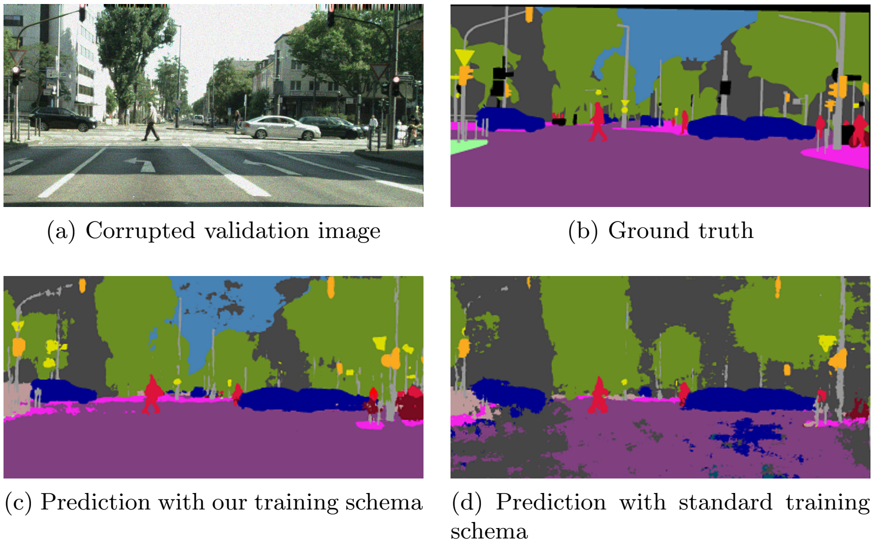
Abstract. For safety-critical applications such as autonomous driving, CNNs have to be robust with respect to unavoidable image corruptions, such as image noise. While previous works addressed the task of robust prediction in the context of full-image classification, we consider it for dense semantic segmentation. We build upon an insight from image classification that output robustness can be improved by increasing the network-bias towards object shapes. We present a new training schema that increases this shape bias. Our basic idea is to alpha-blend a portion of the RGB training images with faked images, where each class-label is given a fixed, randomly chosen color that is not likely to appear in real imagery. This forces the network to rely more strongly on shape cues. We call this data augmentation technique "Painting-by-Numbers". We demonstrate the effectiveness of our training schema for DeepLabv3+ with various network backbones, MobileNet-V2, ResNets, and Xception, and evaluate it on the Cityscapes dataset. With respect to our 16 different types of image corruptions and 5 different network backbones, we are in 74 % better than training with clean data. For cases where we are worse than a model trained without our training schema, it is mostly only marginally worse. However, for some image corruptions such as images with noise, we see a considerable performance gain of up to 25 %.

Keywords: Semantic segmentation, shape-bias, corruption robustness

1 Introduction

Convolutional Neural Networks (CNNs) have set the state-of-the-art for many computer vision tasks [37, 30, 56, 57, 41, 50, 5, 25, 29, 40, 49, 44]. The benchmark datasets which are used to measure performance often consist of clean and undistorted images [11]. When networks are trained on clean image data and tested on real-world image corruptions, such as image noise or blur, the performance can decrease drastically [23, 31, 17, 2, 35].

Common image corruptions cannot be avoided in safety-critical applications: Environmental influences, such as adverse weather conditions, may corrupt the image quality significantly. Foggy weather decreases the image contrast, and low-light scenarios may exhibit image noise. Fast-moving objects or camera motion cause image blur. Such influences cannot be fully suppressed by sensing technology, and it is hence essential that CNNs are robust against common image corruptions. Obviously a CNN should also be robust towards adversarial perturbations (e.g., [58, 33, 10, 27, 4, 47, 3]).


(a) Corrupted validation image

(b) Ground truth

(c) Prediction with our training schema

(d) Prediction with standard training schema

Fig. 1. Results of a semantic segmentation model that is trained with our data augmentation schema. (a) An image crop of the Cityscapes validation set is corrupted by severe image noise. (b) Corresponding ground-truth. (c) Prediction of a model that is trained with our schema. (d) Prediction of the same model with reference training schema, where training images are not augmented with noise. The prediction with our training schema (c) is clearly superior to the prediction of the reference training schema (d), though our model is not trained with image noise. In particular the classes *road*, *traffic signs*, *cars*, *persons* and *poles*, are more accurately predicted

Training CNNs directly on image corruptions is generally a possibility to increase the performance on the respective type of image corruption, however, this approach comes at the cost of increased training time. It is also possible that a CNN overfits to a specific type of image corruption trained on [23, 60].

Recent work deals with the robustness against common image corruptions for the task of full-image classification, and less effort has been dedicated to semantic segmentation. Whereas other work utilizes, e.g., a set of data augmentation operations [32] we propose a new, robustness increasing, data augmentation schema (see Fig. 1) that does: a) not require any additional image data, and b) is easy to implement and c) can be used within any supervised semantic segmentation network, and d) is robust against many common image corruptions.

For this, we build upon the work of Geirhos et al. [22], where it has been shown that increasing the network bias towards the shape of objects does make the task of full-image classification more robust with respect to common image corruptions. We applied the style-transfer technique of Geirhos et al. to Cityscapes, but found the resulting images to be quite noisy (see Fig. 2). Training on such data might, therefore, increase robustness not solely due to an increased shape bias, but rather due to increased image corruption. Our aim is to find a training schema that does not have any type of image corruption added.

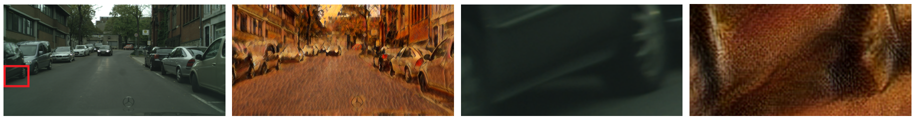

(a) Original data (b) Stylized data (c) Zoom of (a) (d) Zoom of (b)

Fig. 2. Illustration of the style transfer technique of [22]. An original training image (a) of the Cityscapes dataset is stylized by a painting (b). (c) and (d) show the image content of the red rectangle of (a), where (d) is clearly noisier compared to the original data (c)

Whereas the method of [22] delivers high-quality results, their approach requires a computationally intensive style transfer technique and additional image data. We propose a simple, yet effective, data augmentation scheme, that decreases the amount of texture in the training data, and does not need additional data. The basic idea is to alpha-blend some training images with a texture-free representation, as illustrated in Fig. 3 (b). By doing so, the texture-based appearance of a training image is less reliable, forcing the network to develop additional shape-based cues for the segmentation. In this way, our schema does not require additional training data, as we directly use the available semantic segmentation ground-truth. It can be easily utilized for training any supervised semantic segmentation model. We demonstrate our data augmentation scheme's effectiveness on a broad range of common image corruptions, evaluated on the Cityscapes dataset.

In summary, we give the following contributions:

- We propose a simple, yet effective, data augmentation scheme that increases the robustness of well-established semantic segmentation models for a broad range of image corruptions, through increasing the model's shape-bias. Our new training schema requires no additional data, can be utilized in any supervised semantic segmentation model, and is computationally efficient.
- We validate our training schema through a series of validation experiments. With respect to our 16 different types of image corruptions and five different network backbones, we are in 74 % better than training with clean data. We are able to increase the mean IoU by up to 25 %.

2 Related Work

Recent work has dealt with the robustness of CNNs for common image corruptions. We discuss the most recent work in the following.

Benchmarking robustness with respect to common corruptions. The work in [2, 18] demonstrates that shifting input pixels can change the outcome significantly. Dodge and Karam [17] show that CNNs are prone to common corruptions, such as blur, noise, and contrast variations, for the task of full-image classification. The authors further show in [16] that the CNN performance of classifying corrupted images is significantly lower than human performance. Zhou [68] et al. find similar results. Geirhos et al. [23] show that established models [57, 30, 56] for image classification trained on one type of image noise can struggle to generalize well to other noise types. Vasiljevic et al. [60] find a similar result w.r.t image blur, and further, a reduced performance for clean data.

Hendrycks and Dietterich [31] corrupt the ImageNet dataset [14] by many common image corruptions and image perturbations. In this work, we apply the proposed image corruptions to the Cityscapes dataset. Michaelis et al. [48] benchmark the robustness in object detection and find a significant performance drop for corrupted input data.

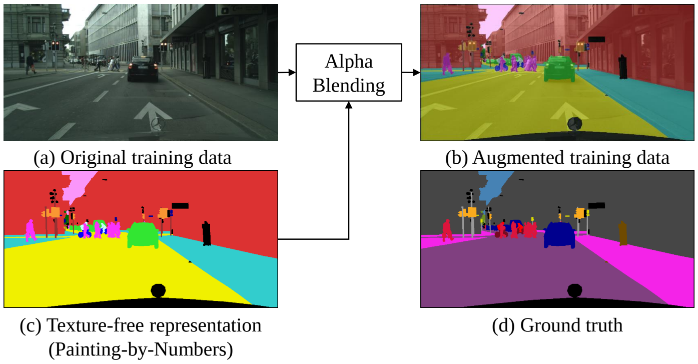
For the task of semantic segmentation, Vasiljevic et al. [60] find that model performance of a VGG-16 [56] is decreasing for an increasing amount of blur in the test data. Kamann and Rother [35] ablate the state-of-the-art semantic segmentation DeepLabv3+ architecture and show that established architectural design choices affect model robustness with respect to common image corruptions. Other work deals with robustness towards adverse weather conditions [54, 53, 61], night scenes [13], or geometric transformations [20, 51].

Increasing robustness with respect to common corruptions. The research interest in increasing the robustness of CNN models with respect to common image corruptions grows. Most methods have been proposed for the task of full-image classification. Mahajan et al. [46] and Xie et al. [62] show that using more training data increases the robustness. The same result is found when more complex network backbones are used, also for object detection [48] and semantic segmentation [35]. Hendrycks et al. [31] show that adversarial logit pairing [36] increases the robustness for adversarial and common perturbations. The authors of [66, 38] increase model robustness through stability training methods.

Several other works apply data augmentation techniques to increase generalization performance. Whereas some work occludes parts of images [67, 15], crops, replaces and mixes several images [63, 64, 59], or applies various (learned) sets of distortions [32, 12], other methods augment with artificial noise to increase robustness [24, 45, 52].

Geirhos et al. [22] demonstrate that classifiers trained on ImageNet tend to classify images based on an image's texture. They further show that increasing the shape-bias of a classifier (through style transfer [21]), also increases the robustness for common image corruptions. This work builds upon this finding to increase the shape-bias of semantic segmentation models and, thus, the robustness for common image corruptions.

3 Training Schema: Painting-by-Numbers



(a) Original training data

(b) Augmented training data

(c) Texture-free representation (Painting-by-Numbers)

(d) Ground truth

Fig. 3. Overview of our proposed training schema, which we refer to as Painting-by-Numbers. (a) An RGB image of the Cityscapes training set and the respective ground-truth in (d). We paint the *numbers*, i.e., ground-truth IDs of (d) randomly, leading to the texture-free representation shown in (c). Painting the numbers randomly is essential since these colors are not likely to appear in real imagery. The final training image (b) is then generated by alpha-blending (a) and (c). A fraction of training data is augmented as in b), which is used as training data that increases the robustness against common corruptions

Our goal is to generically increase the robustness of semantic segmentation models for common image corruptions. Here, *robustness* refers to training a model on clean data and subsequently validating it on corrupted data. Simply adding corrupted data to the training set does certainly increase the robustness against common corruptions. However, this approach comes along with drawbacks: Firstly, a significantly increased training time. Secondly, the possibility to overfit to specific image corruptions [23, 60] and reduced performance on clean data [60]. Thirdly, it further may be hard to actually identify all sources of corruption for new test scenarios. For our training schema, we build on the finding of [22]. We propose an augmentation schema (Painting-by-Numbers) that modifies the training process so that the model develops shape-based cues for the decision of how to segment a pixel, resulting in a generic increase of model robustness.

The basis of our schema is that we treat the segmentation ground-truth as a texture-free representation of the original training data (see Fig. 3). We then colorize (or *paint*) the ground-truth labels (or *numbers*) randomly (Painting-by-Numbers) to generate a representation as shown in Fig. 3 (c). We uniformly sample the color from the sRGB color gamut with range $[0, 255]$, similar to the images of the Cityscapes dataset. Painting the numbers randomly is essential since these colors are not likely to appear in real imagery. Finally, we alpha-blend this this representation with the original training image, according to eq. 1,

$$I_{blended} = \alpha \times I_{painting-by-numbers} + (1 - \alpha) \times I_{original} \tag{1}$$

where $I_{blended}$ is the resulting alpha-blended training image (Fig. 3 b), α is the blend parameter (where $\alpha = 1$ corresponds to a representation where original training input is entirely blended), $I_{painting-by-numbers}$ is the texture-free representation (Fig. 3 c) and $I_{original}$ is the original training image (Fig. 3 a). Training a network on such data forces the network to develop (or increase) its shape-bias since we actively corrupt the textural content of the image. The texture features of the image are, therefore, less reliable, and a model needs to develop additional cues to segment pixels correctly. Painting-by-Numbers is computationally efficient. For our setup, the training time increases by only 2.5 % (please see the supplementary material for more details).

Motivation blending with $\boldsymbol{0 < \alpha < 1}$. We conducted the following analysis. We trained a model solely on texture-free images, as shown in Fig. 3 c, meaning that the blend parameter α is fixed to 1. This network achieved a decent performance when tested on a texture-free variant of the Cityscapes validation set. This is a positive signal because it means that the model is able to learn from entirely texture-free training data.

When we augmented only half of a training batch, instead of every image, (α is still fixed to 1), the performance on both, the original validation set and texture-free validation set was, again, considerably high; However, the robustness of the new model with respect to common image corruptions was not increased. We hypothesize that such a model learns to predict well for two different domains, which are the original data and the texture-free data. This motivates us to choose $\alpha < 1$ for some training images. As we will see, with a varying degree of alpha-blending, the robustness of the model towards common image corruptions increases significantly and, at the same time, keeps a consistently good performance on clean data.

Training protocol. We use the state-of-the-art DeepLabv3+ [6, 8, 7, 5] semantic segmentation architecture as baseline model. We show the effectiveness of Painting-by-Numbers for many network backbones: MobileNet-V2 [55], ResNet-50 [30], ResNet-101, Xception-41, Xception-71 [9]. We augment exactly half of a batch by our Painting-by-Numbers approach and leave the remaining images unchanged. Doing so ensures that the performance on clean data is comparable to a network that is trained regularly on clean data only. We kindly refer to the next section for reasonable choices of the hyperparameters. We apply a similar training protocol as in [8]: crop size 513×513[3], initial learning rate 0.01, “poly” [43] learning rate schedule, using the Atrous Spatial Pyramid Pooling (ASPP) module [5, 26, 28, 39, 65], fine-tuning batch normalization [34] parameters, output stride 16, random scale data augmentation and random flipping during training. As suggested by [8], we apply no global average pooling [42]. We train every model using TensorFlow [1].

[3] Due to hardware limitations we are not able to train on the suggested crop size of 769.

Evaluation protocol. We use the image transformations provided by the ImageNet-C [31] dataset to generate Cityscapes-C, similar to [35]. The ImageNet-C corruptions give a huge selection of transformations. They consist of several types of blur (Gaussian, motion, defocus, frosted glass), image noise (Gaussian, impulse, shot, speckle), weather (snow, spatter, fog, frost), and digital transformations (JPEG, brightness, contrast). Please see the supplementary material for examples. Each corruption type (e.g., Gaussian noise) is parameterized in five severity levels. We evaluate the mean-IoU [19] of many variants of the Cityscapes validation set, which is corrupted by the ImageNet-C transformations.

4 Experimental Evaluation and Validation

In this section, we demonstrate the effectiveness of Painting-by-Numbers. In section 4.1 we discuss implementation details. We then show the results w.r.t the Cityscapes dataset in section 4.2. We conduct a series of experiments to validate the increased shape-bias of a model trained with Painting-by-Numbers in section 4.3.

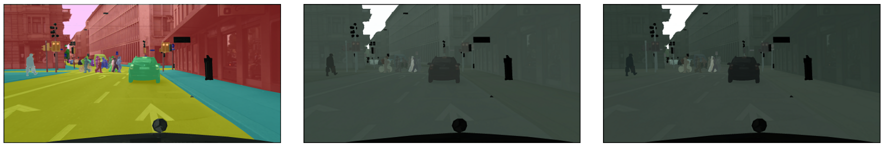
4.1 Implementation Details

We experiment with varying implementations and augmentation schemes, which we discuss next.

Parameters for alpha-blending. Our experiments show that a fixed value for α does not yield the best results. Instead, we use two parameters for alpha-blending, α_{min} and α_{max}. These values define an interval from which α is drawn. They are the essential hyperparameters needed to achieve the best results towards common image corruptions. If α_{min} is too low, i.e., the amount of texture in the image is high, the robustness increase for common corruptions is minor. If α_{min} is too high, i.e., the amount of texture in the image is further diminished, the robustness decreases with respect to common corruptions (as discussed previously). We observe that the models only connect learned features from the two domains (original data domain and alpha-blended data domain) if the latter's texture is present, i.e., $0 < \alpha < 1$.

Batch augmentation schemes. We always augment exactly the half of a batch by Painting-By-Numbers for each iteration of the forward path. To summarize, the only parameters to be optimized are α_{min} and α_{max}. We do not observe better results when for every image in the mini-batch is individually decided if it shall be augmented by Painting-by-Numbers.

Incorporating instance labels. Beside semantic segmentation ground-truth, the Cityscapes dataset also contains instance labels for several classes. We additionally utilize them in our augmentation scheme to paint each instance with a randomly chosen color (instead of painting each instance of a class with the same color), as illustrated in Fig. 4 (a). This produces promising results with respect to further increasing network robustness. Since Painting-by-Numbers is



(a) Instances are painted randomly (b) Instances are painted by RGB-mean (c) Instances are painted by RGB-median

Fig. 4. Examples of several coloring schemes used for Painting-by-Numbers

targeted for semantic segmentation task, we base our schema on the more general semantic labels, which are available for all reference datasets.

Paint with mean and median RGB. We further paint the images with a more consistent color, such as the mean and median RGB value of the class or instance (instead of painting the semantic classes randomly), as illustrated in Fig. 4 (b) and (c). This approach does, as expected, not increase the model robustness. Instead of forcing a model to not rely on texture and color appearance, by corrupting these very properties, the network learns to assign a mean or median value to classes and instances, contrary to the effect of random painting. Hence, there is no need to increase the shape-bias for predicting the segmentation map when the colors are likely to appear in real imagery.

Best Setup. We train MobileNet-V2, ResNet-50, ResNet-101, Xception-41, and Xception-71 with Painting-by-Numbers. We evaluate the models on Gaussian noise to select the final values for α. For ResNet-50, and Xception-41, we observe the best results when we draw α uniformly from the interval $\alpha_{min} = 0.70$ and $\alpha_{max} = 0.99$. For the remaining networks, we observe the best results for $\alpha_{min} = 0.50$ and $\alpha_{max} = 0.99$.

4.2 Results on Cityscapes

In the following, we refer to a network that is trained with standard training schema as the reference model (i.e., trained on clean data only), and to a model that is trained with Painting-by-Numbers as our model. Fig. 5 shows qualitative and quantitative results on corrupted variants of the Cityscapes dataset, when a network (ResNet-50) is trained with both training schemes. Every image corruption is parameterized with five severity levels. Severity level 0 corresponds to the clean data.

The reference model (third row) struggles to predict well in the presence of image corruptions (Fig. 5 top). It segments large parts of *road* wrongly as *building* for *spatter* and *image noise*. When the same model is trained with Painting-by-Numbers, the predictions are clearly superior (fourth row). With respect to quantitative results (Fig. 5 bottom), our model performs significantly better for image corruptions of category *speckle noise*, *shot noise*, and *contrast*. Corruption *contrast* decreases the contrast of the full image, corrupting hence the textural image content strongly. A network that is able to rely also on shape-based cues for the image segmentation is hence a well-performing model for *contrast reduction*. The mean IoU on *spatter* is for both models comparable for the first severity level, but it is for our model higher by almost 15 % for the fourth severity level.

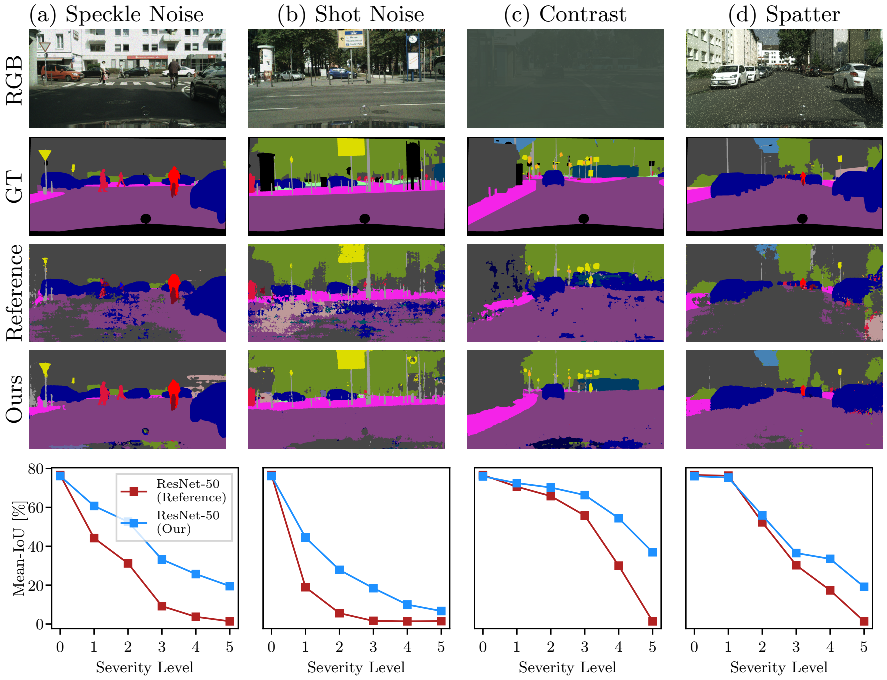


Fig. 5. (top) Qualitative results by the ResNet-50 backbone on four corrupted images of the Cityscapes validation dataset for both the reference model and our model (i.e., trained with Painting-by-Numbers). **(bottom)** Quantitative results on the corrupted variants of the Cityscapes dataset. Each image corruption is parameterized with five severity levels, where severity level 0 corresponds to clean (i.e., original) data. While for clean data, both models' performance is more or less the same, we see that our model is clearly superior for all types of noise added. For consistent performance on clean data, the performance on corrupted data increases when the model is trained with Painting-by-Numbers. For the first severity level of shot noise, the mIoU of our model is higher by 25 %

The results for the remaining image corruptions for the Cityscapes dataset are listed in Table 1. We show the effectiveness of Painting-by-Numbers besides ResNet-50 also for MobileNet-V2, ResNet-101, Xception-41, and Xception-71. In the first column, we report the performance on clean data, i.e., the original Cityscapes validation set. The mIoU evaluated on several types of image corruptions is listed accordingly. Each value is the average for up to five severity levels.

We report for both clean and corrupted data, the result of the reference model and our model. In the following, we discuss the main results of Table 1.

Table 1. Results on the Cityscapes dataset. Each entry is the mean IoU of several corrupted variants of the Cityscapes dataset. Every image corruption is parameterized with five severity levels, and the resulting mean IoU are averaged. For image noise-based corruption, we exclude every severity level whose signal-to-noise ratio less than 10. The higher mIoU of either the reference model or the respective model trained with Painting-by-Numbers is bold. Overall, we see many (74 %) more bold numbers for our Painting-by-Numbers model

		Blur				**Noise**				**Digital**				**Weather**			
Network	Clean	Motion	Defocus	Frosted Glass	Gaussian	Gaussian	Impulse	Shot	Speckle	Brightness	Contrast	Saturate	JPEG	Snow	Spatter	Fog	Frost
Reference																	
MobileNet-V2	**73.0**	**52.4**	**47.0**	**44.7**	**48.1**	9.6	14.2	9.8	25.6	50.4	43.8	32.5	**20.3**	10.8	43.3	47.7	16.1
ResNet-50	**76.6**	57.1	**55.2**	45.3	**56.5**	10.7	13.4	12.1	37.7	59.8	52.7	41.7	**23.4**	**12.9**	39.8	56.2	19.0
ResNet-101	76.0	**58.9**	**55.3**	47.8	**56.3**	22.9	22.9	23.1	45.5	57.7	56.8	41.6	**32.5**	**11.9**	45.5	55.8	23.2
Xception-41	77.8	61.6	**54.9**	51.0	**54.7**	27.9	28.4	27.2	53.5	63.6	56.9	51.7	**38.5**	18.2	46.6	57.6	20.6
Xception-71	77.9	62.5	**58.5**	**52.6**	**57.7**	22.0	11.5	21.6	48.7	67.0	57.2	45.7	**36.1**	16.0	48.0	63.9	20.5
Painting-by-Numbers																	
MobileNet-V2	72.2	49.5	41.4	40.7	43.0	**17.4**	**18.4**	**16.8**	**35.7**	**62.5**	**50.8**	**51.0**	17.6	**12.1**	**46.9**	**56.5**	**22.4**
ResNet-50	76.1	**58.1**	53.5	**50.3**	55.1	**35.7**	**34.3**	**36.1**	**56.7**	**68.8**	**64.2**	**60.5**	21.3	10.6	**46.1**	**61.0**	**22.9**
ResNet-101	**76.3**	58.1	54.2	**48.7**	54.7	**41.6**	**44.3**	**40.6**	**57.4**	**70.5**	**64.4**	**65.0**	25.6	10.8	**50.1**	**56.9**	**28.0**
Xception-41	**78.5**	**65.5**	54.2	**51.1**	51.8	**46.9**	**44.9**	**46.9**	**64.3**	**73.4**	**60.2**	**68.8**	15.7	**19.3**	**55.8**	**65.7**	**28.2**
Xception-71	**78.6**	**63.0**	53.6	48.6	52.2	**35.5**	**38.4**	**34.2**	**57.6**	**74.9**	**63.9**	**69.1**	22.2	**18.2**	**57.4**	**65.4**	**25.5**

Performance w.r.t clean data. Even though we paint the exact half of the training data and train both models for the same amount of iterations, the performance on clean data is oftentimes barely affected.

Performance w.r.t image blur. The robustness of our model with respect to image blur does not notably increase. We assume that Painting-by-Numbers does not increase the performance for this category of image corruptions because blur corrupts the object shapes by smearing the object boundaries. Hence, our learned shape-bias does not work well.

Performance w.r.t image noise. Painting-by-Numbers increases the robustness with respect to image noise the most (see figures above). For example, the absolute mIoU of Xception-41 for Gaussian noise, impulse noise, shot noise, and speckle noise increases by 19.0 %, 16.5 %, 19.7 %, and 11.0 %, respectively.

Performance w.r.t digital corruptions. A network trained with Painting-by-Numbers increases significantly the robustness against the corruptions *brightness*, *contrast*, and *saturation*–but not JPEG artifacts. The reason is that *JPEG compression* corrupts the boundary of objects and incorporates new boundaries through posterization artifacts. Our network cannot hence profit from its increased shape-bias. We refer to the supplement for an illustration.

Performance w.r.t weather corruptions. Xception-71 and Xception-41 increases the performance with respect to *spatter* by 9.4 % and 9.2 %, respectively. Xception-41 further increases the mIoU against *frost* by 7.6 %. Every model increases the performance against *fog*. We cannot observe a significant performance increase for *snow*.

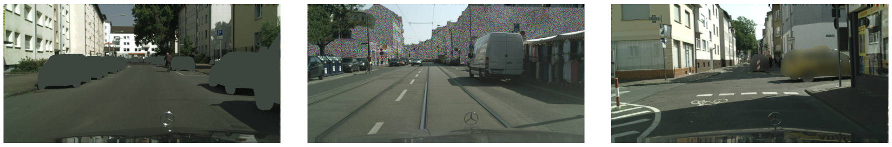
Though the performance increase on image corruptions of category *weather* is less than, e.g., for image noise, the predictions of a network trained with

Painting-by-Numbers are improved for key-classes such as *cars*, *persons*, and *traffic signs* than for a regularly trained network. Please see the supplementary material for more results.

4.3 Understanding Painting-by-Numbers

We explain the increased robustness towards common image corruptions, i.e., when a network is trained with Painting-by-Numbers, by an increased shape-bias. To validate this assumption, we conduct a series of experiments that are based on the following consideration: Classes that either have a) no texture at all or b) texture that is strongly corrupted should be more reliably segmented by a network trained with Painting-by-Numbers. In more detail, we generate numerous, on class-level corrupted, variants of the Cityscapes validation set, as illustrated in Fig. 6. In (a), we remove the texture of *cars* and replace it by the dataset-wide RGB-mean of the training set of the respective class. The respective class does, in this way, not contain any texture but homogeneous color information. In (b) and (c) we corrupt *building* and *car* by a high degree of additive Gaussian noise and Gaussian blur, respectively. Please note that Fig. 6 shows only a small set of examples. We apply these corruptions for every class.

We test the models on such images to evaluate if they are capable of segmenting the respective class when they cannot rely on the class texture. To achieve this, a network needs to utilize other cues, such as shape-based cues.



(a) Replaced *car* by RGB-mean (b) Corrupted *building* by severe noise (c) Corrupted *car* by severe blur

Fig. 6. Examples of image data to validate an increasing shape-bias when models are trained with Painting-by-Numbers. We remove, or strongly corrupt, the texture of each class in the Cityscapes dataset and evaluate the segmentation performance when a network cannot rely on the class texture. (a) Texture is fully replaced by the dataset-wide RGB-mean value of the respective class. (b) Class is corrupted by severe noise. (c) Class is corrupted by severe blur

Instead of IoU, we use the sensitivity s ($s = TP/(TP + FN)$, where TP are true-positives, and FN are false-negatives) as evaluation metric. The sensitivity is for these experiments more appropriate than IoU ($IoU = TP/(TP + FN + FP)$) since we are solely interested in the segmentation performance on the class-level. Because all classes but one is clean (i.e., not corrupted), false-positively (FP) segmented pixels are of less interest. Utilizing IoU could, especially for classes covering fewer image regions, result in misleading scores. The results of these experiments are listed in Table 2.

Table 2. Sensitivity score per class for several corrupted variants on the class-level of the Cityscapes datasets. **Clean:** The performance on clean (i.e. original, non-corrupted) data. **RGB-mean:** The texture of a class is replaced by the dataset-wide RGB mean of that class. **Noise:** The texture of a class is corrupted by severe additive Gaussian noise. **Blur:** The texture of a class is corrupted by severe Gaussian blur. The higher sensitivity score of a network backbone of either the reference (top) or our model (bottom) is bold. Overall, we see many more bold numbers for our Painting-by-Numbers model

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
Reference																			
Clean	98.8	**93.1**	**96.6**	53.3	**69.6**	**74.5**	**81.7**	**85.7**	**96.7**	**74.1**	**97.8**	**91.5**	**76.9**	**97.6**	85.1	**92.8**	**70.7**	**79.2**	**88.7**
RGB-mean	92.7	21.1	**88.9**	**40.7**	5.4	68.9	12.8	31.5	1.4	**3.3**	**97.7**	73.4	62.5	24.5	13.6	**16.3**	**9.0**	2.9	0.9
Noise (*scale* = 0.5)	5.8	0.8	**95.0**	0.2	1.7	4.7	6.4	**39.2**	0.1	1.4	2.8	8.1	2.9	4.5	0.0	0.0	**7.1**	0.2	3.4
Noise (*scale* = 1.0)	2.9	0.0	**94.0**	0.1	1.2	2.0	6.4	**40.2**	0.0	0.5	0.2	4.4	1.2	3.6	0.0	0.0	**3.0**	0.2	2.0
Blur ($\sigma = 20$)	**94.6**	42.8	**89.3**	**38.0**	1.8	63.6	18.4	19.1	0.6	**7.3**	94.0	55.4	55.5	56.7	32.5	**24.0**	**7.7**	9.0	0.9
Blur ($\sigma = 30$)	**94.1**	42.1	**89.4**	**33.2**	1.2	62.3	14.0	16.2	0.6	**2.3**	93.8	54.6	51.8	44.2	29.8	**18.6**	**8.1**	4.2	1.1
Painting-by-Numbers																			
Clean	**99.0**	90.3	96.3	**56.0**	67.1	68.9	76.5	81.1	96.2	66.3	97.1	89.5	74.0	96.8	**89.7**	86.0	59.6	72.2	87.8
RGB-mean	**97.9**	**53.8**	51.2	34.2	**14.9**	**79.7**	**38.4**	**40.5**	**1.8**	2.3	97.4	**78.4**	**66.3**	**78.6**	**37.6**	3.5	0.4	**9.1**	**4.6**
Noise (*scale* = 0.5)	**97.4**	**50.9**	92.1	**8.4**	**37.4**	**34.1**	**8.2**	11.1	**23.3**	**30.6**	**32.3**	**50.1**	**19.7**	**49.8**	**31.5**	**1.9**	0.0	**0.3**	**26.7**
Noise (*scale* = 1.0)	**95.9**	**51.7**	91.3	**9.6**	**29.4**	**32.3**	**7.1**	9.9	**12.2**	**27.2**	**33.6**	**52.7**	**21.3**	**40.6**	**25.8**	**1.1**	0.0	**0.4**	**23.3**
Blur ($\sigma = 20$)	49.3	**43.5**	86.5	18.7	**4.7**	**73.6**	**55.1**	**29.8**	**1.0**	0.8	**94.3**	**75.5**	**73.2**	**71.9**	**56.6**	7.9	0.5	**20.2**	**3.5**
Blur ($\sigma = 30$)	46.3	**48.0**	83.2	14.1	**4.5**	**73.6**	**49.7**	**25.4**	**1.0**	0.5	**94.7**	**74.6**	**71.1**	**73.7**	**47.9**	3.8	0.2	**18.2**	**3.8**

Quantitative results. The results in Table 2 are created by DeepLabv3+ with ResNet-50 as network backbone. As previously, we refer to a network that is trained with the standard training schema as the reference model (i.e., only clean data used), and to a model that is trained with Painting-by-Numbers as our model. The top (bottom) part of the Table contains the sensitivity score for each class of the reference model (our model). Each line shows the sensitivity for the corrupted data as described previously (the performance on clean data is also listed). The higher sensitivity of a network backbone of either the reference model (top) or our model (bottom) is bold. We separately discuss in the following the quantitative results for class categories "stuff" and "things".

Both networks perform well for classes "stuff" since the amount of texture is often poor, such as for *road*, *wall*, *sidewalk*, and *sky*. The sensitivity of both models differs for *road* by 5.2 %, for *wall* by 6.5 %, and for sky by 0.3 %. Whereas the absolute sensitivity for both models is above 90.0 % for *road* and *sky*, it is less than 41 % for *wall*. Our model performs for *sidewalk* better by 32.7 %.

Painting-by-Numbers performs worse than the reference for classes "stuff" with a large amount of textual information, such as *building*, *vegetation*, and *terrain*. For example, the sensitivity score of our model for *building* is 37.7 % less. Classes "stuff" have no distinct shape, hence, Painting-by-Numbers does not aid performance. When, additionally, the amount of texture of a class is large, the sensitivity of our model is less than of the reference model.

The reference model performs well when the texture of the category "things" is replaced by RGB-mean. Its sensitivity for *person* is 73.4 %, which is only 5.0 % less than for our model. The result for class *rider* is similar.

However, our model performs often significantly better than the reference model for most of the remaining "things" such as *car*. The sensitivity score of our model for this class is $s_{ours} = 78.6\,\%$, which is 54.1 % higher than the sensitivity score of the reference model. We explain this high score with a large shape-bias due to both the distinct shape of *cars* and the comparatively large number of *cars* in the training set [11]. Our model performs for other classes of "things" also better than the reference model. For example, the sensitivity score for classes *traffic light*, *traffic sign* and *pole* is higher by 25.6 %, 9.0 %, and 9.8 %, respectively. Both models perform poorly on "vehicles" that are, compared to *cars*, less frequent present in the training set (e.g., *truck, motorcycle, train*).

In the presence of severe Gaussian noise, the reference model is struggling to segment classes. The sensitivity is poor for every class, except for *traffic signs* and *building*. In the presence of image noise, the reference model tends to segment pixels oftentimes as these very classes, as illustrated in Fig. 5 and Fig 7. The sensitivity scores of our model are often significantly higher. Similar to the previously discussed results, the sensitivity with respect to "stuff" with less texture is often high (e.g., $s_{ours} = 95.9\,\%$ for *road*). The sensitivity scores are also high for "things" such as *persons* and *cars* ($s_{ours} = 52.7\,\%$, and $s_{ours} = 40.6\,\%$, respectively). Our model segments many classes well that are corrupted by severe image noise, even though our model has not seen image noise during the training.

The reference model generally performs well when classes are low-pass filtered by severe Gaussian blur. This result is in accordance with [35], where the authors found semantic segmentation models to be relatively robust towards image blur. Again, for class category "things", our model outperforms the reference model in most cases. For example, the sensitivity score of our model for *person*, *rider*, and *car* is by approx. 20.0 % higher.

Qualitative results. See Fig. 7 for qualitative results of the previously discussed experiments. Please see the caption of Fig. 7 for discussion.

5 Conclusions

We proposed a simple, yet effective, data augmentation schema (Painting-by-Numbers) for semantic image segmentation in this work. This training schema increases the robustness for a wealth of common image corruptions in a generic way. Painting-by-Numbers corrupts training data so that the texture of image classes becomes less reliable, forcing the neural network to develop and increase its shape-bias to segment the image correctly. Painting-by-Numbers' benefits are that it does not require any additional data, is easy to implement in any supervised segmentation model, and is computationally efficient. It would be interesting to enforce other network biases, such as context bias or layout bias, and even to combine these with a shape bias, to further increase the robustness of semantic segmentation models with respect to common image corruptions.

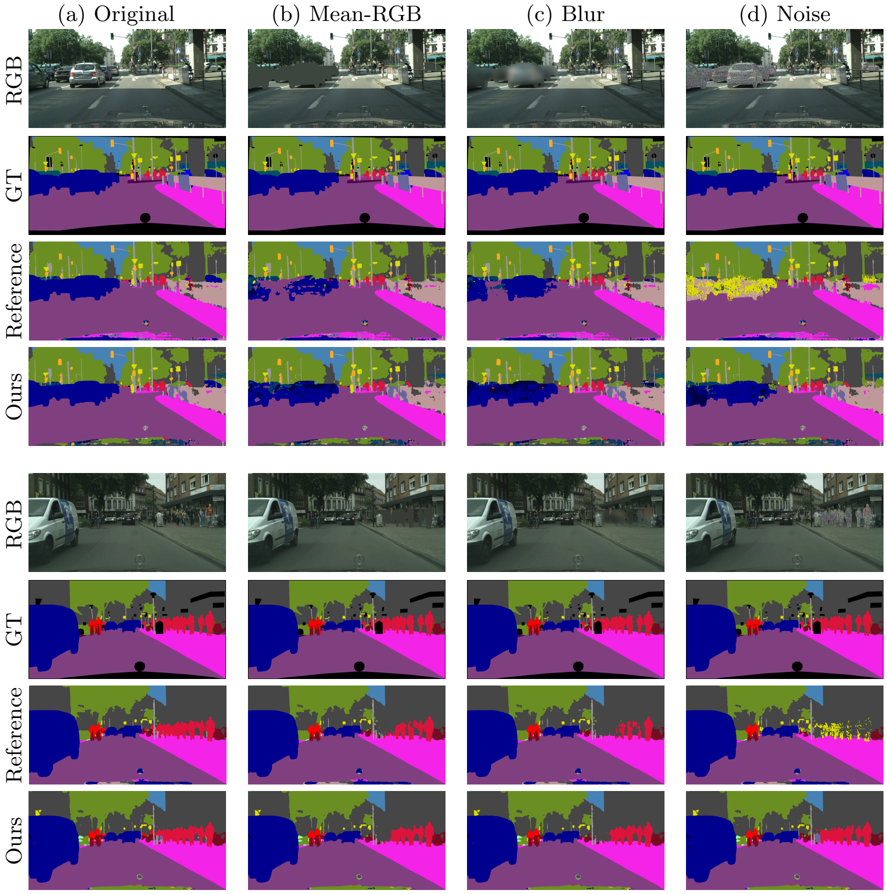


Fig. 7. Qualitative results of our experiments to understand the effect of Painting-by-Numbers. We train the ResNet-50 network backbone on Cityscapes with a standard training schema (i.e., with clean data only, reference model) and with Painting-by-Numbers (our model). **(top)** The first row shows the original validation image and the corrupted variants for class *car* and the respective ground truth in the second row. We replace either the class texture by the dataset-wide RGB-mean, strongly low-pass filtered the class, or added severe Gaussian image noise. The third row shows the predictions of the reference model. The fourth row shows the predictions of our model. The predictions in the fourth row (our model) are superior to the third row (reference model). Our model is able to withstand the image noise based corruption (last column) for which the reference model confuses *cars* with *traffic signs* mostly. **(bottom)** For *persons*, the reference model predicts well, when the RGB-mean replaces the texture of the class. Both models are relatively robust when the classes are low-pass filtered by severe Gaussian blur. Similar to the results with respect to class *car*, the reference model struggles to predict well for severe image noise and confuses *persons* also with *traffic signs* mostly

References

1. Abadi, M., Barham, P., et al.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016), https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
2. Azulay, A., Weiss, Y.: Why do deep convolutional networks generalize so poorly to small image transformations? Journal of Machine Learning Research **20**(184), 1–25 (2019), http://jmlr.org/papers/v20/19-519.html
3. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 3–14. AISec '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3128572.3140444, http://doi.acm.org/10.1145/3128572.3140444
4. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. 2017 IEEE Symposium on Security and Privacy (SP) (2017)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. vol. abs/1412.7062 (2015), http://arxiv.org/abs/1412.7062
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: TPAMI (2017), http://arxiv.org/abs/1606.00915
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017), http://arxiv.org/abs/1706.05587
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 833–851. Springer International Publishing, Cham (2018)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR (2017). https://doi.org/10.1109/CVPR.2017.195, http://ieeexplore.ieee.org/document/8099678/
10. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., Usunier, N.: Parseval networks: Improving robustness to adversarial examples. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, PMLR, International Convention Centre, Sydney, Australia (2017), http://proceedings.mlr.press/v70/cisse17a.html
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
12. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
13. Dai, D., Van Gool, L.: Dark model adaptation: Semantic image segmentation from daytime to nighttime. In: ITSC. pp. 3819–3824. IEEE (2018)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
15. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
16. Dodge, S., Karam, L.: A study and comparison of human and deep learning recognition performance under visual distortions. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN). pp. 1–7. IEEE (2017)

17. Dodge, S.F., Karam, L.J.: Understanding how image quality affects deep neural networks. In: Quomex (2016)
18. Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A.: Exploring the landscape of spatial robustness. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 1802–1811. PMLR, Long Beach, California, USA (Jun 2019), http://proceedings.mlr.press/v97/engstrom19a.html
19. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. In: IJCV (2010). https://doi.org/10.1007/s11263-009-0275-4, http://link.springer.com/10.1007/s11263-009-0275-4
20. Fawzi, A., Frossard, P.: Manitest: Are classifiers really invariant? BMVC (2015)
21. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv (Aug 2015), http://arxiv.org/abs/1508.06576
22. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: ICLR (May 2019), https://openreview.net/forum?id=Bygh9j09KX
23. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. In: Advances in Neural Information Processing Systems 31 (2018), https://arxiv.org/abs/1808.08750
24. Gilmer, J., Ford, N., Carlini, N., Cubuk, E.: Adversarial examples are a natural consequence of test error in noise. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2280–2289. PMLR, Long Beach, California, USA (Jun 2019), http://proceedings.mlr.press/v97/gilmer19a.html
25. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
26. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)
27. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. NIPS Workshop on Deep Learning and Representation Learning **abs/1412.5068** (2014)
28. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 346–361. Springer International Publishing, Cham (2014)
29. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. ICCV pp. 1026–1034 (2015)
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016). https://doi.org/10.1109/CVPR.2016.90, http://ieeexplore.ieee.org/document/7780459/
31. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. ICLR (2019)
32. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. ICLR (2020)
33. Huang, X., Kwiatkowska, M.Z., Wang, S., Wu, M.: Safety verification of deep neural networks. In: Computer Aided Verification (2017)
34. Ioffe, Sergey, Szegedy, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)

35. Kamann, C., Rother, C.: Benchmarking the robustness of semantic segmentation models. In: CVPR (June 2020)
36. Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)
37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
38. Laermann, J., Samek, W., Strodthoff, N.: Achieving generalizable robustness of deep neural networks by stability training. In: Fink, G.A., Frintrop, S., Jiang, X. (eds.) Pattern Recognition. pp. 360–373. Springer International Publishing, Cham (2019)
39. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. Washington, DC, USA (2006)
40. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. In: Nature (2015), https://doi.org/10.1038/nature14539
41. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998)
42. Lin, M., Chen, Q., Yan, S.: Network in network. In: ICLR (2014)
43. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv:1506.04579 [cs.CV] (2015)
44. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. vol. abs/1411.4038 (2015)
45. Lopes, R.G., Yin, D., Poole, B., Gilmer, J., Cubuk, E.D.: Improving robustness without sacrificing accuracy with patch gaussian augmentation. arXiv preprint arXiv:1906.02611 (2019)
46. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 185–201. Springer International Publishing, Cham (2018)
47. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. In: ICLR (2017), https://arxiv.org/abs/1702.04267
48. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. In: Machine Learning for Autonomous Driving Workshop, NeurIPS 2019. vol. 190707484 (Jul 2019), https://arxiv.org/abs/1907.07484
49. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV. pp. 1520–1528 (2015)
50. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
51. Ruderman, A., Rabinowitz, N.C., Morcos, A.S., Zoran, D.: Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. arXiv preprint arXiv:1804.04438 (2018)
52. Rusak, E., Schott, L., Zimmermann, R., Bitterwolf, J., Bringmann, O., Bethge, M., Brendel, W.: Increasing the robustness of DNNs against image corruptions by playing the Game of Noise. arXiv (Jan 2020), https://arxiv.org/abs/2001.06057
53. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. IJCV **126**(9), 973–992 (2018)

54. Sakaridis, C., Dai, D., Van Gool, L.: Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In: ICCV (2019)
55. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR (2018)
56. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015), http://arxiv.org/abs/1409.1556
57. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015). https://doi.org/10.1109/CVPR.2015.7298594, http://ieeexplore.ieee.org/document/7298594/
58. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
59. Takahashi, R., Matsubara, T., Uehara, K.: Data augmentation using random image cropping and patching for deep CNNs. IEEE Transactions on Circuits and Systems for Video Technology (2019)
60. Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G.: Examining the impact of blur on recognition by convolutional networks. arXiv:1611.05760 [cs.CV] **abs/1611.05760** (2016), http://arxiv.org/abs/1611.05760
61. Volk, G., Stefan, M., von Bernuth, A., Hospach, D., Bringmann, O.: Towards robust cnn-based object detection through augmentation with synthetic rain variations. In: ITSC (2019)
62. Xie, Q., Hovy, E., Luong, M.T., Le, Q.V.: Self-training with noisy student improves imagenet classification. arXiv preprint arXiv:1911.04252 (2019)
63. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV. pp. 6023–6032 (2019)
64. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. ICLR (2017)
65. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017), http://arxiv.org/abs/1612.01105
66. Zheng, S., Song, Y., Leung, T., Goodfellow, I.J.: Improving the robustness of deep neural networks via stability training. In: CVPR. pp. 4480–4488 (2016)
67. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI (2017)
68. Zhou, Y., Song, S., Cheung, N.M.: On classification of distorted images with deep convolutional neural networks. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2017)