IMPROVING AUGMENTATION AND EVALUATION SCHEMES FOR SEMANTIC IMAGE SYNTHESIS

Prateek Katiyar
Bosch Center for Artificial Intelligence
Renningen, Germany
prateek.katiyar@de.bosch.com

Anna Khoreva
Bosch Center for Artificial Intelligence
Renningen, Germany
Anna.Khoreva@de.bosch.com

ABSTRACT

Despite data augmentation being a *de facto* technique for boosting the performance of deep neural networks, little attention has been paid to developing augmentation strategies for generative adversarial networks (GANs). To this end, we introduce a novel augmentation scheme designed specifically for GAN-based semantic image synthesis models. We propose to randomly warp object shapes in the semantic label maps used as an input to the generator. The local shape discrepancies between the warped and non-warped label maps and images enable the GAN to learn better the structural and geometric details of the scene and thus to improve the quality of generated images. While benchmarking the augmented GAN models against their vanilla counterparts, we discover that the quantification metrics reported in the previous semantic image synthesis studies are strongly biased towards specific semantic classes as they are derived via an external pre-trained segmentation network. We therefore propose to improve the established semantic image synthesis evaluation scheme by analyzing separately the performance of generated images on the biased and unbiased classes for the given segmentation network. Finally, we show strong quantitative and qualitative improvements obtained with our augmentation scheme, on both class splits, using state-of-the-art semantic image synthesis models across three different datasets. On average across COCO-Stuff, ADE20K and Cityscapes datasets, the augmented models outperform their vanilla counterparts by $\sim$3 mIoU and $\sim$10 FID points.

1 INTRODUCTION

In spite of the recent successes of Semantic Image Synthesis (SIS) models (Park et al., 2019; Liu et al., 2019), one can still observe unsatisfactory artifacts in the synthesized images. Since the input label maps do not provide any supervision about the structural content within the semantic segments, the generated images often lack class-relevant structural information and additionally contain undesirable distortions (see Fig. 1). Inspired by the task-specific augmentation studies in other vision applications (Dwibedi et al., 2017; Tripathi et al., 2019), in this work, we propose an augmentation method specifically designed to overcome the above mentioned limitations of the SIS models.

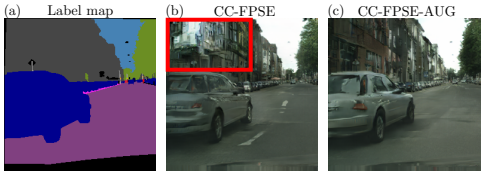

Figure 1: Label map input (a) and the synthetic images generated using the baseline (b) and augmented (c) CC-FPSE (Liu et al., 2019) models. The proposed augmentation scheme fixes the artifact (highlighted with the red box) introduced by the baseline and improves the overall perceptual and structural details of the synthesized image.

Our proposed augmentation method greatly improves the quality of the synthetic images by enabling the generator to focus more on the fine-grained details (see Fig. 1). We achieve this by randomly warping objects in the label map fed to the SIS model as an input. The local shape variations between the semantic input and non-warped real image enable the generator to learn geometric properties of the scene better, which may otherwise be ignored as the generator has access to the original layout that perfectly aligns with the real image. Besides, the discriminator also utilizes the misalignment between the real image and the warped label map to identify the real and fake images, forcing the generator to correct distortions introduced to its input by learning the object-level shape details.

We show the efficacy of our augmentation scheme by improving recent SIS models on three datasets, both quantitatively and qualitatively. Besides standard image synthesis metrics, we also use semantic segmentation metrics for the SIS evaluation that were adopted with a two-fold reasoning (Isola et al., 2017; Park et al., 2019). First, a good SIS model should generate images whose layout aligns well with the ground truth label map. Second, realistically-looking semantic classes in the generated image should be recognized well by an external segmentation network trained on the real images from an independent dataset. However, we discover that the biases learned by the segmentation network during training leak into the quantification of the synthetic images, resulting in an overestimation of the SIS model's performance. We therefore propose to mitigate this issue by identifying the biased and unbiased classes in all datasets for the given segmentation networks and show the advantages of our augmentation scheme using an extended evaluation on both class splits.

2 Augmentation Method

In SIS, training dataset consists of a pair of data samples (s, x), where s denotes the input semantic label map and x is the respective real image. In this task, the generator is trained to learn the distribution of real images conditioned on the semantic input. Thus, the loss functions for the generator G and the discriminator D take the following form:

$$\underset{min}{\mathcal{L}_G} = -\mathbb{E}_s[\log D(G(s), s)], \underset{max}{\mathcal{L}_D} = \mathbb{E}_{(x,s)}[\log D(x, s)] + \mathbb{E}_s[\log\left(1 - D(G(s), s)\right)]. \quad (1)$$

While modifications in the objective functions and architectures (Isola et al., 2017; Park et al., 2019; Liu et al., 2019) have led to steady improvements in the performance of SIS models, they do not explicitly guide the generator to learn the local shape details of objects in the real image. In fact, as the recent SIS generator architectures (Liu et al., 2019; Park et al., 2019) condition the features of all intermediate layers on the semantic input, the generator can simply copy the global structural layout of the scene directly from the conditioning input. This direct dependency further weakens the generator's ability to learn finer structural details, because the semantic input itself lacks the information about the composition of various classes, e.g. windows/doors for the building class. Thus, to prevent the generator from naively copying the scene layout and encourage it to learn local class-specific shape properties, we propose to warp the objects in the label map fed to the generator.

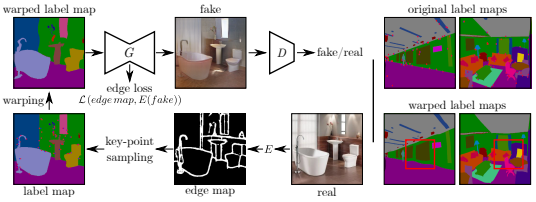

Figure 2: Augmented SIS pipeline. Left panel: in contrast to a vanilla model, the input to the generator G in the augmented model is a label map warped based on the edges, estimated by the edge detector E. Not shown is the real image and the warped label map fed into the discriminator D. Right panel: examples of original and the respective warped label maps. Zoom in for details.

Specifically, we obtain the warped label map $\tilde{s}$ using a transformer function: $\tilde{s} = t(s)$. Here, the thin-plate spline transform t is obtained by estimating the affine and non-affine warping coefficients, for a set of fixed $\{u, v\}$ and moving $\{\acute{u}, \acute{v}\}$ points (Bookstein, 1989). To selectively warp the objects in the input label map s, we sample the key-points $\{u, v\}$ uniformly from boundary pixels in the edge map. Afterwards, the moving points $\{\acute{u}, \acute{v}\}$ are obtained by adding random pixel shifts to the previously sampled key-points within a defined range. The amount of pixel shift controls the degree of warping. For each dataset, we determined this parameter experimentally by training multiple models with varying levels of distortions. Fig. 2 shows examples of such warped label maps. We train the augmented models by warping the input label maps from the entire training dataset and conditioning the generator and the discriminator on the warped semantic layouts. The real images fed to the discriminator remain unmodified. During inference, the non-warped semantic label maps are used to generate the synthetic images.

3 Evaluation Bias

Let $X \in \mathbb{R}^{H \times W}$ and $Y \in \{1, 2..., N_{cl}\}^{H \times W}$ denote an input image and its densely labelled semantic map with N_{cl} categories. Given an arbitrary image input X to a pre-trained segmentation network, let Y_{pred} be the predicted segmentation map. Using Y and Y_{pred}, we can calculate the following evaluation metrics for each class i: Pixel Accuracy $(PA_i) = n_{ii}/t_i$ and, Intersection over Union $(IoU_i) = n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$. Here, n_{ji} and t_i are the number of pixels of class j that are labelled as class i in Y_{pred} and the total number of pixels of class i in Y (Long et al., 2015).

Figure 3: ADE20K results of SPADE and CC-FPSE baselines and the augmented (-AUG) models.

To assess whether the segmentation model is biased towards specific semantic classes, we modify X by using different perturbation schemes and evaluate the metrics defined above by feeding the perturbed images into the segmentation network. For $(u, v) \in (\{1, 2, ..., H\}, \{1, 2, ..., W\})$, let $M_i(u, v) = \mathbb{1}\{Y(u, v) = i\}$ be the mask for class i, where $\mathbb{1}$ is the indicator function. We can define the perturbed image $\tilde{X}_i$ for class i as: $\tilde{X}_i(M_i, X) = X \circ (1 - M_i) + P \circ M_i$. Here, $\circ$ denotes the Hadamard product and $P \in \mathbb{R}^{H \times W}$ is the applied perturbation detailed below.

$$P(u, v) = \begin{cases} c_0 \\ \dfrac{1}{\sum_u \sum_v M(u, v)} \sum_u \sum_v M(u, v) * X(u, v) \\ G(\sigma_0) \circledast X \\ \sim lognormal(\mu, \sigma) \end{cases} \tag{2}$$

where c_0 is a fixed grayscale value, σ_0 is the standard deviation parameter of the Gaussian kernel, and the mean μ and standard deviation σ are determined from the masked image segment. By feeding the perturbed images into the segmentation model, we obtain the set $\mathbb{M}_{p_i}$ that contains the aforementioned metrics calculated for class i for all perturbation schemes. Given the score $m_i \in \{PA_i, IoU_i\}$ for the original image, the class i is considered as biased if the following criterion is met for any of the two respective perturbed metric sets:

$$i = \begin{cases} \textbf{biased} & \text{if } \exists\, m_{p_i} \ni m_{p_i} > \delta * m_i, \forall\, m_{p_i} \in \mathbb{M}_{p_i}, \\ \textbf{unbiased} & \text{otherwise.} \end{cases} \tag{3}$$

The factor $\delta * m_i$ defines the threshold for the perturbed metrics for class i to be regarded as biased.

4 Experiments

We conduct experiments on COCO-Stuff (Caesar et al., 2018), ADE20K (Zhou et al., 2017) and Cityscapes (Cordts et al., 2016) using Pix2PixHD (Wang et al., 2018), SPADE (Park et al., 2019) and CC-FPSE (Liu et al., 2019) as baselines. For consistent evaluations with the baselines, we use the following segmentation models: DeepLabV2 (COCO-Stuff) (Chen et al., 2017; Nakashima), UperNet101 (ADE20K) (Xiao et al., 2018; CSAILVision), and DRN-D-105 (Cityscapes) (Yu et al., 2017; Yu). Further experimental and implementation details are provided in the supp. data.

4.1 Evaluation Bias Results

Biased Classes. Following Sec. 3, we group the classes in the biased and unbiased categories. Note that we focus only on strongly biased classes. For a δ value of $2/3$ in Eq. 3, we find 29, 52 and 5 biased classes in COCO-Stuff, ADE20K and Cityscapes (provided in the supp. data). Smaller values of δ diluted the biased class split. Fig. 4 shows examples of original and perturbed image segmentations for two of the perturbation schemes defined in Eq. 2. In the Gaussian blur perturbation example, the network correctly identifies the mouse in front of the keyboard, while misses the one behind. This example shows the effect of context bias picked up by the segmentation model, as in majority of training images the mouse is placed either to the front or alongside the keyboard. The lognormal perturbation example shows bidirectional effects of the bias learned by the segmentation network during training. Here, the perturbed wall segment is classified accurately, whereas the unaltered shower-door segment is misclassified as mirror. These examples highlight that for the biased classes, even unrealistic SIS model images may lead to high evaluation metrics.

Analysis on Baselines. In Table 1 we report the results of evaluating the synthetic images of all baselines and the real images across all datasets. To our surprise–when considering all classes–the

Model	COCO-Stuff				ADE20K				Cityscapes			
	FID ↓	mIoU ↑	mIoU_{BC}	mIoU_{UC}	FID ↓	mIoU ↑	mIoU_{BC}	mIoU_{UC}	FID ↓	mIoU ↑	mIoU_{BC}	mIoU_{UC}
CC-FPSE	**18.9**	41.0	**47.3**	39.7	33.2	42.6	44.5	41.6	53.6	61.8	79.3	55.5
CC-FPSE-AUG	19.1	**42.1**	46.3	**41.2**	**32.6**	**44.0**	**45.8**	**43.1**	**52.1**	**63.1**	**80.1**	**57.0**
SPADE	**22.5**	37.8	43.5	36.7	**34.4**	39.6	41.7	38.6	64.7	59.2	**79.9**	51.9
SPADE-AUG	22.7	**38.2**	43.5	**37.1**	34.6	**41.2**	**43.2**	**40.2**	**62.3**	**60.4**	79.8	**53.5**
Pix2PixHD	128.7	12.0	21.2	10.1	59.1	24.4	27.9	22.6	73.6	56.7	78.8	48.8
Pix2PixHD-AUG	**54.2**	**21.9**	**31.8**	**19.9**	**41.5**	**32.5**	**36.0**	**30.7**	**72.7**	**58.0**	**79.1**	**50.5**
Average Δ	24.7	3.8	3.2	3.9	6	3.7	3.6	3.7	1.6	1.3	0.3	1.6
Real images	–	35.4	39.3	34.6	–	36.5	36.3	36.6	–	62.0	79.4	55.8

Table 1: Evaluation comparison across datasets. BC and UC denote the biased and unbiased class splits. **Bold** indicates the best model between the baseline and its augmented variant (-AUG).

segmentation models perform better on the synthetic images than on the real images for COCO-Stuff and ADE-20K. The underlying cause for this observation becomes clear, as we bifurcate the metrics into the biased and unbiased classes. While overall the segmentation networks perform better on the biased classes than the unbiased ones, this performance gap is significantly higher for the synthetic images generated using the SIS baselines compared to the real images, especially for COCO-Stuff and ADE-20K. We also notice large differences between Pix2PixHD and the other baselines, indicating that newer architectures improve upon the synthesis of the unbiased classes.

4.2 Augmentation Results

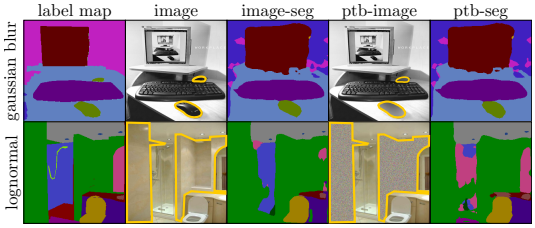

Figure 4: Image perturbation examples exhibiting the segmentation models' bias. The perturbed segments are shown with yellow contours. For each perturbed image (ptb-image), the segmentation model is able to correctly infer the semantic category of the affected segment (as in image-seg), despite the corruption of perceptual details.

We report quantitative results of the baselines and the respective augmented models in Table 1. Here we show only FID and mIoU metrics. The table with all metrics is given in the supp. data. It can be seen that for all datasets, the augmented models outperform the respective baselines. As the augmented generators focus more on the local shape and structural details, we notably see gains on the unbiased class metrics. In some cases, the overall gain is only contributed by the increments on the unbiased classes, e.g. CC-FPSE mIoU on COCO-Stuff. In fact, for each model pair and dataset combination in Table 1, the augmented models show consistent gains over the baselines on the overall and unbiased mIoU indicating that the fine-grained improvements of the augmentation extend across a variety of GAN-based SIS architectures as well as datasets containing diverse scenes and objects. The CC-FPSE and SPADE baselines trained on COCO-Stuff and ADE20K achieve lower FID scores than the respective augmented models. But, for both cases the augmented models perform better on almost all other metrics. We notice that the proposed class splits provide an additional level of granularity when benchmarking SIS models, as they allow to identify cases of pseudo-improvements where the overall boost in performance is caused mainly due to a gain in the biased class metric. For instance on Cityscapes, PA of SPADE (93.1) is higher than that of CC-FPSE (92.8). But, this improvement is only caused by the biased classes. On the unbiased classes, CC-FPSE (84.5) outperforms SPADE (82.3) by more than 2 points.

The qualitative examples comparing the two best baselines and the augmented models are shown in Fig. 3. Here, we show that the augmentation scheme reduces distortions, adds fine-grained structural details and enhances the perceptual realism of the synthetic images. For both examples, the augmentation approach greatly reduces the distortions introduced by the baselines (see red boxes). The first row shows how the augmented models gradually add local structural details to the synthetic images and produce a high fidelity image. Furthermore, both examples show that compared to the baselines, the augmented models also improve the overall perceptual realism of the translated images.

5 Conclusions

We propose a novel data augmentation method for GAN-based SIS models. Targeting the shortcomings of the recent SIS studies, the proposed method greatly improves the local shape and structural details inside the semantic classes. Moreover, to fairly analyze the improvements of the SIS mod-

els, we extend the semantic segmentation metrics into biased and unbiased class groups. Enabled by this new analysis, we observe that the recent SIS models strongly underperform on unbiased classes, while our proposed augmentation method improves their results on both class groups.

REFERENCES

Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 1989.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

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. *TPAMI*, 2017.

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 *CVPR*, 2016.

CSAILVision. Semantic Segmentation on MIT ADE20K dataset in PyTorch. `https://github.com/CSAILVision/semantic-segmentation-pytorch`.

Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, 2019.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Kazuto Nakashima. DeepLab with PyTorch. `https://github.com/kazuto1011/deeplab-pytorch`.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. In *CVPR*, 2019.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Fisher Yu. drn. `https://https://github.com/fyu/drn`.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.