ADVERSARIAL SEGMENTATION LOSS FOR SKETCH COLORIZATION

Samet Hicsonmez[*]   *Nermin Samet*[†]   *Emre Akbas*[†]   *Pinar Duygulu*[*]

[*] Computer Engineering, Hacettepe University, Ankara, Turkey
[†] Computer Engineering, Middle East Technical University, Ankara, Turkey

ABSTRACT

We introduce a new method for generating color images from sketches or edge maps. Current methods either require some form of additional user-guidance or are limited to the "paired" translation approach. We argue that segmentation information could provide valuable guidance for sketch colorization. To this end, we propose to leverage semantic image segmentation, as provided by a general purpose panoptic segmentation network, to create an additional adversarial loss function. Our loss function can be integrated to any baseline GAN model. Our method is not limited to datasets that contain segmentation labels, and it can be trained for "unpaired" translation tasks. We show the effectiveness of our method on four different datasets spanning scene level indoor, outdoor, and children book illustration images using qualitative, quantitative and user study analysis. Our model improves its baseline up to 35 points on the FID metric. Our code and pretrained models can be found at https://github.com/giddyyupp/AdvSegLoss.

Index Terms— sketch colorization, sketch to image translation, Generative Adversarial Networks (GAN), image segmentation, image to image translation

1. INTRODUCTION

Generating an image from an input sketch (i.e. edge map), a task known as "sketch to image translation", or "sketch colorization" for short, is an attractive task as sketches are easy to obtain and convey essential information about the content of an image. At the same time, it is a challenging task due to the large domain gap between single channel edge maps and color images. In addition, sketches usually lack details for background objects, and even sometimes for foreground objects.

Sketch colorization has been explored in a variety of domains including faces [1, 2, 3], objects [4, 5, 6, 7], animes [8, 9, 10, 11, 12], art [13] and scenes [14, 15, 16]. Most of the methods in these studies require some form of user-guidance, as additional input, in the form of, e.g., a reference color, patch or image. Without this guidance, these methods produce unrealistic colorizations. Another important observation is that, except Liu et al.'s work [7], all methods follow the "paired" approach, which limits the method to datasets that have a ground-truth image per sketch.

In this work, we propose to leverage general purpose semantic image segmentation to alleviate the two shortcomings mentioned above. Semantic segmentation methods have matured to a level that they produce useful results even for datasets on which they were not trained (Section 4). We hypothesize that a correctly colored sketch would yield a proper segmentation result and leverage this result in an extra adversarial loss in a GAN setting. By doing so, our method neither requires additional user guidance nor becomes limited to the "paired" domain.

We propose a new method which utilizes semantic segmentation for sketch based image colorization problem. We introduce three models considering different levels of segmentation feedback in the sketch to image translation pipeline. Our models could be integrated into any paired or unpaired GAN model. We demonstrate effectiveness of using segmentation cues through extensive evaluations. Our contributions in this paper can be summarized as follows. (i) We propose to use general purpose semantic segmentation as an additional adversarial loss in a GAN model, for the sketch colorization problem. Ground-truth segmentation labels are not a requirement for our approach. (ii) Our method is neither specific to a domain (e.g. face, art, anime, etc.) nor limited to the "paired" approach. (iii) We conduct extensive evaluations on four distinct datasets on both paired and unpaired settings, and show the effectiveness of our method through qualitative, quantitative and user study analysis.

2. RELATED WORK

Even though the edge map and sketch of an image are different concepts, in practice, XDoG [17] or HED [18] based edge maps are considered as sketches (e.g., [14, 10]). Moreover, some sketch based models [4] use edge maps for data augmentation. Hence, we refer to all these methods as sketch to image translation models. General purpose image-to-image translation methods [19, 20, 21, 22, 23] could be used to solve sketch to image translation tasks. However, the results of these generic methods are usually not very satisfactory.

One widely used solution to improve the colorization performance is to employ additional color [14, 10, 11, 9], patch [5], image [2, 12, 13, 8] or language guidance [16]. For instance, in color guidance, users specify their desired colors for the regions in the sketch image, and the model utilizes this information to generate exact or similar colors for these regions. Some automatic methods also utilize user guidance to improve their performance resulting in a hybrid approach. Most of the sketch to image translation methods are based on the "paired" training approach [14, 10, 4, 15], however, recently unpaired methods have also been presented [7, 13].

In Scribbler [14], one of the very first paired and user guided scene sketch colorization models, in addition to pixel, perceptual and GAN losses, they use total variation loss to encourage smoothness. They use XDoG to generate sketch images of 200k bedroom photos, and produce 128×128 colorized images. Zou et al. [16] use text inputs to progressively colorize an input sketch, in such a way that a novel text guided sketch segmenter segments and locates the objects in the scene. EdgeGAN [15] maps edge images to a latent space during training using an edge encoder. During inference, the edge encoder encodes the input sketch to the latent space to subsequently generate a color image. They experimented with 14 foreground and 3 background objects from COCO [24] dataset.

EdgeGAN [15] and Scribbler [14] use a supervised approach where input sketches and corresponding output images exist. How-
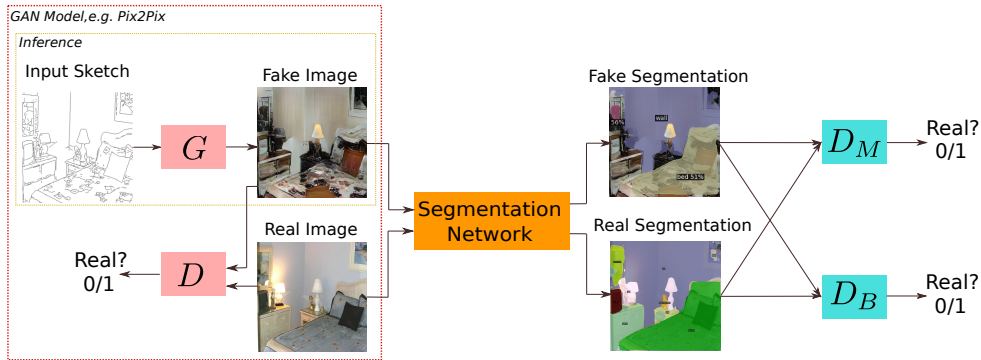
Fig. 1: Our proposed model with adversarial segmentation loss for sketch colorization.

ever, it is hard to collect sketch image pairs. Liu et al. [7] propose a two stage method to convert object sketches to color images in an unsupervised (unpaired) way. They first convert sketches to grayscale images, then color images. Self supervision is also used to complete the deliberately deleted sketch parts and clear the added noisy edges from sketch images. In Sketch-to-Art [13], an art image is generated using an input sketch and a target art style image. They encode content of the input sketch and style of the art image, then fuse both features to generate a stylized art image.

3. MODEL

Figure 1 shows the overall structure of our proposed model. The box with dashed yellow borders shows the inference stage of our model. Red border marks the GAN model used for sketch to image translation. In this work, we used *Pix2Pix* and *CycleGAN* as baselines for paired and unpaired training, respectively. This preference is made based on the effectiveness of both methods across a variety of tasks and datasets. Our model could be integrated into any other GAN model.

Our model consists of a baseline GAN, a panoptic segmentation network (*Seg*) and two discriminators (D_M and D_B). Panoptic segmentation network is trained offline on the COCO Stuff [25] dataset and its weights are frozen during the training of our model. Real and fake images are fed to the *Seg* network to get real and fake segmentation maps. Then, these two segmentation maps are given to the discriminators to classify them as real or fake.

We designed three variants of our model to embed different levels of segmentation feedback to the sketch to image translation pipeline.

The first variant utilizes the full segmentation map of an image where all foreground and background classes – a total of 135 classes – are considered. In this model, ground-truth color image I_{real} and the generated color image I_{fake} are fed to *Seg* which outputs full segmentation maps for both images. Then, these two outputs are given to a discriminator network D_M to discriminate between real and fake segmentation maps. We call this model as *Multi-class* in the rest of the paper.

As a higher level of abstraction, grouping objects only as background and foreground may yield sufficient information. In the second variant of our model, we only used two classes (background and foreground) in the segmentation map by grouping all foreground classes into one and all background classes into another class. As with our original multi-class model, binary segmentation outputs for

Fig. 2: Sample segmentation results on different datasets.

real and fake images are fed to a discriminator network D_B to discriminate between real and fake ones. We refer to this model as *Binary*.

Finally, our third variant is the union of the above two. It contains both discriminators, and is named as *Both*. Overall loss function for our model is the summation of losses of the baseline GAN model (L_G) and the two additional discriminators' (L_B and L_M). Formally, the objective function is $\mathcal{L} = w_g L_G + w_b L_B + w_m L_M$. We run ablation experiments to set the best values of w_g, w_b and w_m. We keep w_g fixed and search for w_b and w_m on the bedroom dataset using the *Both* model setting. Results show that using 1.0 for w_g, w_b and w_m yields the best FID score (see Table 2).

We used PyTorch [26] to implement our models. We use sketch images as source domain, and color images as target domain. Datasets are described in Section 4. All training images (i.e. color and sketch images) are resized to 256×256 pixels. We train all models for 200 epochs using the Adam optimizer [27] with a learning rate of 0.0002. We conducted all our experiments on a Nvidia Tesla V100 GPU.

4. DATASET

We evaluated our models on four challenging datasets. The first dataset consists of bedroom images from the Ade20k indoor dataset [28], with 1355 train and 135 test images. The second dataset contains children's book illustrations by Alex Scheffler [22], with 659 train and 131 test images. The third and fourth datasets were curated by us from the COCO dataset. We collected images which contain elephant or sheep instances. Note that these images contain not only elephants and sheeps but objects/regions from other foreground and
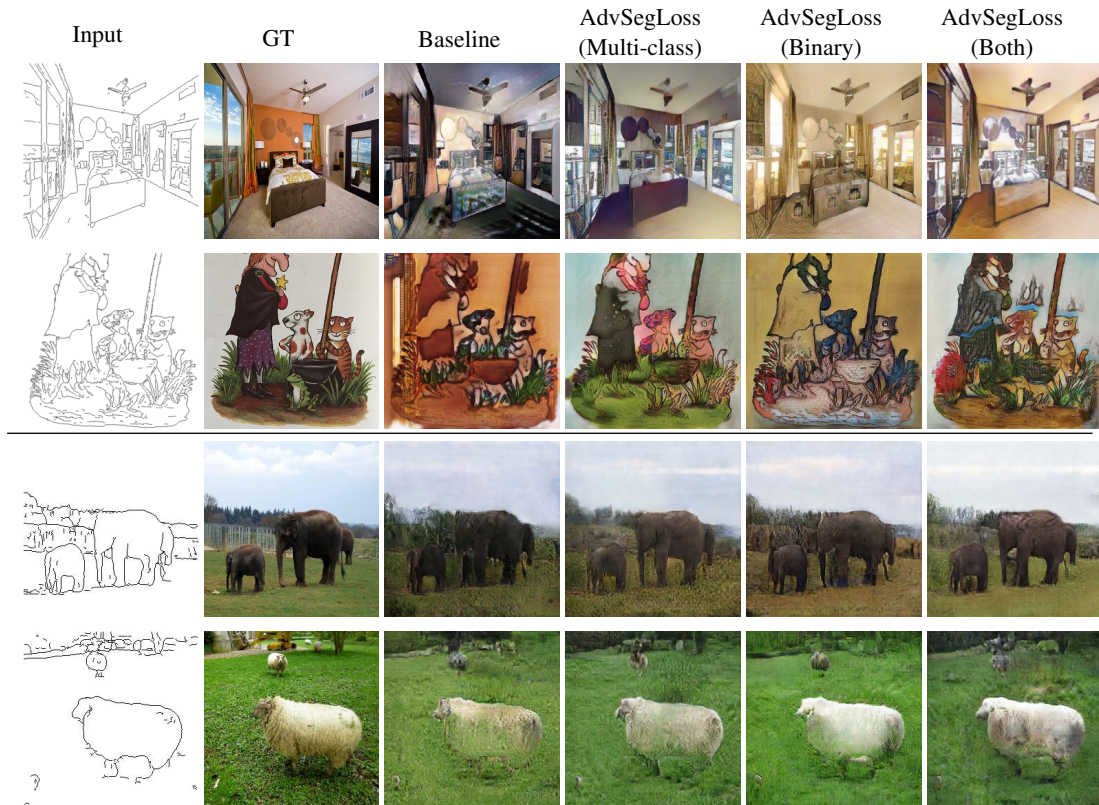
| Input | GT | Baseline | AdvSegLoss (Multi-class) | AdvSegLoss (Binary) | AdvSegLoss (Both) |

Fig. 3: Sample results from baselines (CycleGAN and Pix2Pix) and our model with different settings. Input images on each row are from bedroom, illustration, elephants and sheep datasets, respectively. First two rows display results of unpaired training, and last two rows show results for paired training. On bedroom and elephant datasets *Binary*, on illustration and sheep datasets *Both* setting gave best results for both training schemes.

background classes such as person, animals, mountains, grass and sky. Elephant dataset contains 1800 train and 343 test images, and the sheep dataset has 1300 train and 229 test images. Example images from these datasets and their segmentation outputs are shown in Figure 2.

Edge images are extracted using the HED [18] method. In the first two columns of Figure 3, we present sample natural and edge images for all the datasets. It can be seen that the images contain a variety of foreground and background objects, also it is hard – even for the trained eye – to figure out the source dataset for some of the edge images. Our code, pretrained models and the scripts to produce the "sheep" and "elephant" datasets, and corresponding sketch images can be found at https://github.com/giddyyupp/AdvSegLoss.

5. EXPERIMENTS

We compared our models with the baseline models: CycleGAN [19], AutoPainter [10] and Pix2Pix [21]. We used their official implementations that are publicly available. Baseline models are trained for 200 epochs.

5.1. Quantitative Analysis

To quantitatively evaluate the quality of generated images, we used the widely adopted Frechet Inception Distance (FID) [29] metric. FID score measures the distance between the distributions of the generated and real images. Lower FID score indicates the higher similarity between two image sets. We present FID scores for all the experiments in the Table 1. FID scores are inline with the visual inspections (see Figure 3), for all the datasets, at least one of the variants of our model performed better than the baseline.

First of all, when we compare FID scores of two training schemes and baseline models, paired training (Pix2Pix) performed better than unpaired training, as expected. However, our "adversarial segmentation loss" affected the results of paired and unpaired cases differently. For instance, on elephant dataset our models improved baseline up to 35 points for unpaired case, but only 5 points for paired case.

Another crucial observation is that segmentation guidance closed the gap between unpaired and paired training results. Best FID scores for unpaired models on bedroom, illustration and elephant datasets become very close to or even better than paired training. For instance on the elephant dataset, the initial 40+ point FID gap (126 vs 83) dropped to 13 (92 vs 79) on *Binary* setting. Here the only exception is the sheep dataset. Since the sheep dataset contains various complex objects, unpaired and paired models failed to generate plausible images.

When we look at the best performing settings on different datasets, structure of the dataset has an effect on the results. For instance, even though one is an indoor and the other one is an outdoor dataset, bedroom and elephant images are composed of similar structure. FG/BG ratios and placements of them in these datasets

Table 1: Comparison with baseline methods in terms of FID scores, lower is better.

Dataset	Unpaired				Paired				
	CycleGAN	+AdvSegLoss (Multi-class)	+AdvSegLoss (Binary)	+AdvSegLoss (Both)	AutoPainter	Pix2Pix	+AdvSegLoss (Multi-class)	+AdvSegLoss (Binary)	+AdvSegLoss (Both)
Bedroom	113.1	111.7	**87.1**	93.2	206.8	100.5	100.0	**95.1**	110.1
Illustration	213.6	206.9	204.8	**189.4**	272.0	180.0	176.9	178.0	**175.7**
Elephant	126.4	103.9	**91.9**	116.9	155.1	83.5	85.8	**78.8**	82.8
Sheep	209.3	207.2	236.1	**196.8**	233.1	157.0	159.9	162.0	**150.5**

are similar across all images, i.e. walls, ceiling and floors in bedroom images are always positioned in the same places on different images. Also elephant images contain very few FG objects, i.e. only elephants most of the time, and large BG areas such as grass, trees and sky. On these two datasets, *Binary* setting which considers FG/BG classes only gave the best FID score. On the other hand, illustration and sheep images got a variety of FG objects and scenes. On such datasets, using only a FG/BG discriminator even degrades the performance.

Table 2:
Ablation results.

Table 3: User Study results.

w_b and w_m	FID
0.1	114.8
0.5	114.5
1.0	**93.2**
5.0	147.8
10.0	104.6

Dataset	CycleGAN	+AdvSegLoss
Bedroom	20.0	**80.0**
Illustration	27.0	**73.0**
Elephant	39.1	**60.9**
Sheep	19.1	**80.9**

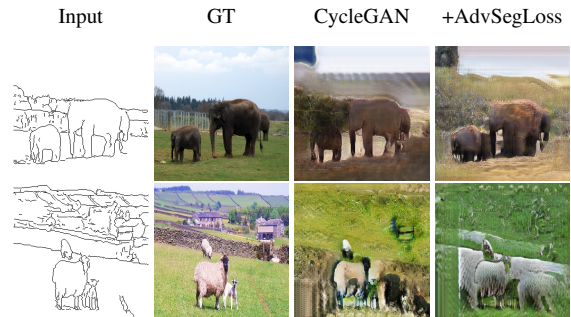Input　　GT　　CycleGAN　　+AdvSegLoss

Fig. 4: Sample results on elephant and sheep datasets for CycleGAN and our best model. Realism of both models are not satisfactory, however, especially colors of BG areas are better in our results.

5.2. Qualitative Analysis and User Study

We present visual results of sketch colorization for our model and the baseline models in the Figure 3. On bedroom and illustration datasets, we show results of unpaired training, and on elephant and sheep datasets we show paired training results.

On the bedroom dataset, the *Binary* setting generates better images compared to baselines and other settings. Colors are uniform across the object parts in this setting. There are defective colors in the CycleGAN results such as the bottom of the bed and floor. On the illustration dataset, the baseline model performed poorly. Objects are hard to recognize and most importantly colors are not proper at all. On the other hand, *Multi-class* and *Both* settings generate significantly better images i.e. generated objects and background got consistent colors.

Finally, on elephant and sheep datasets, although generated images are not very visually appealing for all the methods, segmentation guided images are quite appealing compared to baseline models'. On the elephant dataset *Binary*, on the sheep dataset *Both* setting performed the best.

We conducted a user study[1] to measure realism of generated images. We show two random images (at random positions, left or right) which were generated with CycleGAN and our best setting (lowest FID score) for all four datasets, and asked participants to select the more realistic one.

We collected a total of 115 survey inputs from 39 different users. In Table 3, we present results of the user study in terms of preference

percentages of each model. User study results are inline with the FID score results, on all datasets, images generated by our model were preferred by the users most of the time.

6. CONCLUSION

In this paper, we presented a new method for the sketch colorization problem. Our method utilizes a general purpose image segmentation network and adds an adversarial segmentation loss (AdvSegLoss) to the regular GAN loss. AdvSegLoss could be integrated to any GAN model, and works even if the dataset doesn't have segmentation labels. We used CycleGAN and Pix2Pix as baseline GAN models in this work. We conduct extensive evaluations on various datasets including bedroom, sheep, elephant and illustration images and evaluate the performance both quantitatively (using FID score) and qualitatively (through a user study). We show that our model outperforms baselines on all datasets on both FID score and user study analysis.

Regarding the limitations of our method, although we improve the baseline both qualitatively and quantitatively, especially elephant and sheep results lack realism. Even the paired training results are not visually appealing on these two datasets (last two rows of Figure 3), most probably due to the fact that the baseline models are not very successful at generating complex scenes.

---

[1]Readers could reach our study at http://52.186.136.234:8080/

8. REFERENCES

[1] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu, "Deepfacedrawing: deep generation of face images from sketches," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 72–1, 2020.

[2] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo, "Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence," in *CVPR*, 2020.

[3] Yuhang Li, Xuejin Chen, Feng Wu, and Zheng-Jun Zha, "Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks," in *ACM MM*, 2019.

[4] Wengling Chen and James Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *CVPR*, 2018.

[5] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays, "Texturegan: Controlling deep image synthesis with texture patches," in *CVPR*, 2018.

[6] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang, "Image generation from sketch constraint using contextual gan," in *ECCV*, 2018.

[7] Runtao Liu, Qian Yu21, and Stella X Yu, "Unsupervised sketch to photo synthesis," in *ECCV*, 2020.

[8] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri, "Comicolorization: semi-automatic manga colorization," in *SIGGRAPH Asia 2017 Technical Briefs*. 2017.

[9] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo, "User-guided deep anime line art colorization with conditional adversarial networks," in *ACM MM*, 2018.

[10] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo, "Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks," *Neurocomputing*, vol. 311, pp. 78 – 87, 2018.

[11] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu, "Two-stage sketch colorization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–14, 2018.

[12] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu, "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan," in *Asian Conference on Pattern Recognition*, 2017.

[13] Bingchen Liu, Kunpeng Song, Yizhe Zhu, and Ahmed Elgammal, ": Synthesizing stylized art images from sketches," in *Asian Conference on Computer Vision*, 2020.

[14] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *CVPR*, 2017.

[15] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou, "Sketchycoco: Image generation from freehand scene sketches," in *CVPR*, 2020.

[16] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu, "Language-based colorization of scene sketches," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–16, 2019.

[17] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen, "Xdog: an extended difference-of-gaussians compendium including advanced image stylization," *Computers & Graphics*, vol. 36, no. 6, pp. 740–753, 2012.

[18] Saining Xie and Zhuowen Tu, "Holistically-nested edge detection," in *ICCV*, 2015.

[19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV*, 2017.

[20] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," *ICCV*, 2017.

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

[22] Samet Hicsonmez, Nermin Samet, Emre Akbas, and Pinar Duygulu, "Ganilla: Generative adversarial networks for image to illustration translation," *Image and Vision Computing*, p. 103886, 2020.

[23] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

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

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014.

[28] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[29] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017.