NATURAL-LOOKING ADVERSARIAL EXAMPLES FROM FREEHAND SKETCHES

Hak Gu Kim[1,*] *Davide Nanni*[2,*] *Sabine Süsstrunk*[2]

[1]Immersive Reality and Intelligent Systems Lab, Chung-Ang University, South Korea
[2]School of Computer and Communication Sciences, EPFL, Switzerland

ABSTRACT

Deep neural networks (DNNs) have achieved great success in image classification and recognition compared to previous methods. However, recent works have reported that DNNs are very vulnerable to adversarial examples that are intentionally generated to mislead the predictions of the DNNs. Here, we present a novel freehand sketch-based natural-looking adversarial example generator that we call *SketchAdv*. To generate a natural-looking adversarial example from a sketch, we force the encoded edge information (i.e., the visual attributes) to be close to the latent random vector fed to the edge generator and adversarial example generator. This preserves the spatial consistency of the adversarial example generated from the random vector with the edge information. In addition, by employing a sketch-edge encoder with a novel sketch-edge matching loss, we reduce the gap between edges and sketches. We evaluate the proposed method on several dominant classes of SketchyCOCO, the benchmark dataset for sketch to image translation. Our experiments show that our *SketchAdv* produces visually plausible adversarial examples while remaining competitive with other adversarial attack methods.

Index Terms— image translation, image synthesis, image classification, adversarial examples, generative adversarial network

1. INTRODUCTION

Deep neural networks (DNNs) are widely used tools in various signal processing fields, such as image classification [1, 2], speech recognition [3, 4], and natural language processing [5, 6]. However, recent studies have shown that DNNs are vulnerable to adversarial examples. Here, we define adversarial examples to be natural scenes or images that are manipulated by either adding noise or perturbations [7, 8], which make the DNN classifier fail to predict the correct class. Such miss-classification is a critical issue in security or privacy-related applications [9, 10].

Most of the adversarial attack methods that generate adversarial examples have been proposed for a *digital attack* scenario, i.e., the digital image is altered. Well-known methods are the Fast Gradient Sign Method (FGSM) [7], Projected Gradient Decent (PGD) [11], and Carlini and Wagner (CW) attacks [12]. Although it is enough to add subtle perturbations in the l_p ball [13] for a successful digital attack, it is insufficient for *physical-world* attacks. Larger or unrestricted perturbations are needed to mislead DNNs, since subtle perturbations are usually too small to be captured by a digital camera in the physical-world [14]. As a consequence, several physical-world attack methods have been proposed, such as robust physical perturbations (RP_2) [9] and adversarial patch (AdvPatch) [15]. However, they produce large and noticeable perturbations and thus do not satisfy the stealthiness constraint, which is one of the key factors of adversarial attacks.

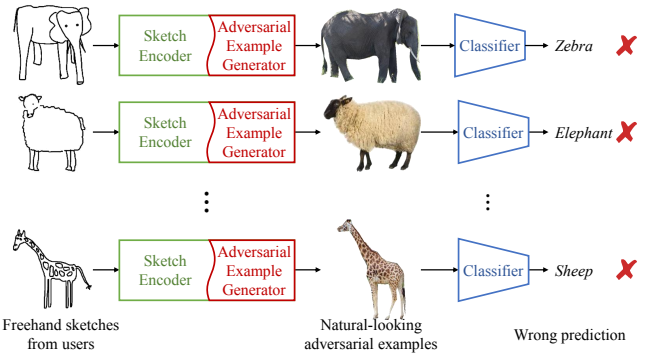

Fig. 1: Synthesizing natural-looking adversarial examples from freehand sketches. Given a freehand sketch from a user, the edge encoder firstly encodes the edge information of the sketch image. With the encoded edge information and the object's class, our generator then synthesizes a natural-looking adversarial example preserving the sketch information while fooling the DNN classifier.

Most recently, a few methods have been proposed that deal with both digital and physical-world attack scenarios. In [13], the authors propose adversarial camouflage by fusing the target image and the target style via neural style transfer. However, this is not practical because the target style and target region mask information are required to generate the adversarial images. In [16], a semantic adversarial example generation method was proposed by embedding the attribute information to the target image. However, it requires to encode all of the attribute channels to embed the semantic level adversarial perturbation.

*Equal contribution

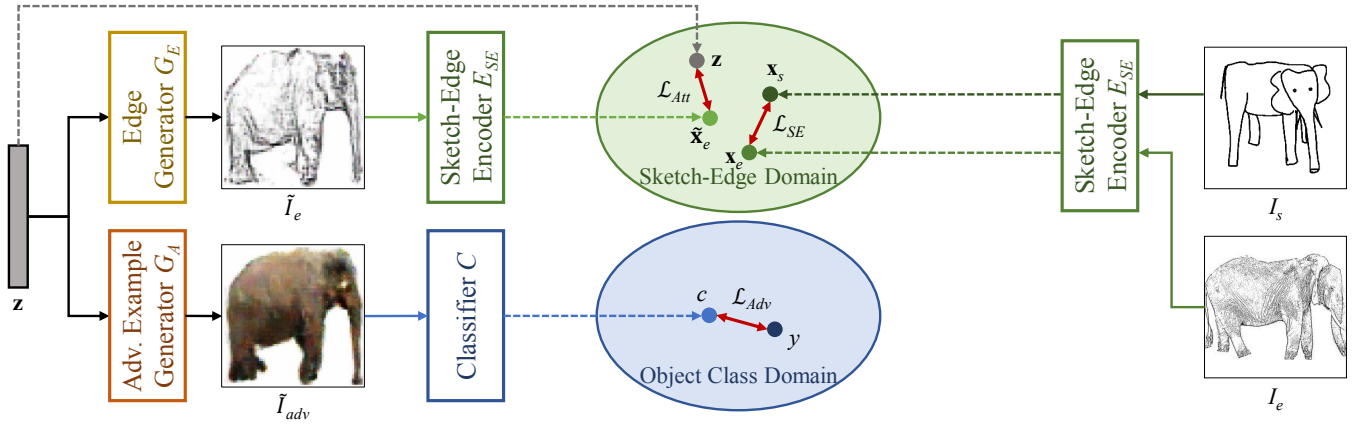

Fig. 2: Overall architecture of the proposed *SketchAdv* in training. From the random latent vector $\mathbf{z}$, edge image $\tilde{I}_e$ and natural-looking adversarial example $\tilde{I}_{adv}$ are generated by the edge generator G_E and the adversarial example generator G_A, respectively. To match the edge image and the natural image, the generated edge image is mapped onto the sketch-edge domain by the sketch-edge encoder E_{SE}, $\tilde{\mathbf{x}}_e$. Then, the latent vector $\mathbf{z}$ can learn the encoded edges of the sketches with $\mathcal{L}_{Att}$ (Section 2.1) to make the generator synthesize the natural-looking adversarial example from the freehand sketch. The generator G_A can produce the adversarial example $\tilde{I}_{adv}$ by maximizing the distance between the predicted class c and the ground-truth class label y, $\mathcal{L}_{adv}$ (Section 2.1). Simultaneously, the sketch-edge encoder E_{SE} helps to link the sketch image to the edge image, and therefore to the corresponding natural-looking adversarial example, by leveraging the loss $\mathcal{L}_{SE}$ between the computed embeddings $\mathbf{x}_e$ and $\mathbf{x}_s$ (Section 2.2).

To overcome the limitations of the existing methods, we propose a novel freehand sketch-based natural-looking adversarial example generator (*SketchAdv*). Figure 1 shows the key intuition of method. Given a freehand sketch from the user, the sketch-edge encoder firstly encodes the sketch information and maps it to the edge information. Then, the adversarial example generator, trained with the relation between the edges and natural images, synthesizes the natural-looking adversarial examples that are able to fool the DNN classifier.

Our contributions can be summarized as follows:

- We propose a novel natural-looking adversarial example generation framework from freehand sketches. By embedding the edge information (i.e., the visual attributes) into the input latent vector in the sketch-edge domain, the adversarial example generator can learn the mapping from the edge image to the natural adversarial image.
- To mitigate the difference between freehand sketches and the edges, we propose a sketch-edge matching loss function in the sketch-edge domain. By matching the encoded sketch and the encoded edge in the latent space, the adversarial image generator can provide visually plausible adversarial examples from freehand sketches directly.
- To verify the naturalness and attack ability of our adversarial examples, we perform the evaluations on SketchyCOCO, the benchmark for sketch to image translation. We achieve both good perceptual quality and attack success.

2. PROPOSED METHOD

Figure 2 depicts the overall architecture of the proposed *SketchAdv*. Our goal is to synthesize the natural-looking adversarial example $\tilde{I}_{adv}$ from the freehand sketch image I_s.

In training, first, the edge generator G_E and adversarial example generator G_A learn the edge image $\tilde{I}_e$ and natural-looking adversarial image $\tilde{I}_{adv}$, respectively. To learn the relation between the freehand sketch I_s and the edge I_e, we design the sketch-edge encoder E_{SE} so that we can match their embedding in the sketch-edge domain. Simultaneously, by encoding the edge feature with the generated edge image $\tilde{I}_e$ with the same E_{SE}, we can embed the edge representation of the sketch in the latent vector $\mathbf{z}$ to achieve natural-looking images from the freehand sketch. Finally, the adversarial example generator G_A learns how to manipulate the adversarial example $\tilde{I}_{adv}$ to fool the classifier C by maximizing the classification distance between the predicted class c and the ground-truth class label y.

2.1. Adversarial Example Generator from Edges

As mentioned in SketchyCOCO, it is difficult to directly map to natural images from sketches. To deal with that, we leverage an EdgeGAN [17]-like architecture, which consists of two generators (G_A and G_E) for adversarial example and edge generation, three discriminators (D_A, D_E, and D_J) for adversarial image, edge, and joint learning, a sketch-edge encoder E_{SE} and an image classifier C. At first, both G_A and G_E take the random latent vector $\mathbf{z}$ as input, together with

an one-hot encoded vector to represent the object class. The two discriminators D_A and D_E distinguish the generated adversarial example $\tilde{I}_{adv}$ and edge images $\tilde{I}_e$ from real natural I_n and edge I_e images, respectively. The joint discriminator D_J encourages $\tilde{I}_{adv}$ and $\tilde{I}_e$ to be in the same object class. In order to insert adversarial perturbation to $\tilde{I}_{adv}$, we design a novel objective function of a natural-looking adversarial example generator with the adversarial loss term, $\mathcal{L}_{Adv}(c)$:

$$\mathcal{L}_{G_A} = \mathcal{L}_G(D_A) + \mathcal{L}_G(D_J) - \eta\mathcal{L}_{Adv}(c) \tag{1}$$

where η is the weight for adversarial loss, $\mathcal{L}_{Adv}$ and c is the predicted class by the classifier C. $\mathcal{L}_G(D(\tilde{x}))$ is equal to $\mathbb{E}_{\tilde{x}\sim P_g}[-D(\tilde{x})]$.

$$\mathcal{L}_{Adv}(c) = -\mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[(1 - P(C = c|y)^{\gamma}) \log P(C = c|y)] \tag{2}$$

where γ=2 and $P(C = c|y)$ is the probability of the true class for $\hat{I}_{adv}$ as predicted by the classifier C. The adversarial loss is based on the focal loss.

In the discriminator loss, we consider the zero-centered gradient penalty (the third term in Eq. 3) for its characteristic generalization and training stability.

$$\mathcal{L}_D(D) = \mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] + \mathop{\mathbb{E}}_{x\sim P_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}\left[\|\nabla_{\hat{x}} D(\hat{x})\|_2^2\right] \tag{3}$$

2.2. Sketch-Edge Encoder

The sketch-edge encoder is designed to 1) encourage the encoded attribute information of the edge image to be close to the latent vector $\mathbf{z}$ fed to G_A and G_E with edge attribute embedding loss ($\mathcal{L}_{Att}$, the first term in Eq. 4) and 2) to reduce the gap between the freehand sketch images and the edge information of natural images with the sketch-edge matching loss ($\mathcal{L}_{SE}$, the second term in Eq. 4). By considering both of the edge attribute embedding and sketch-edge matching, we can produce more natural-looking images from freehand sketches. The loss for E_{SE} consists of the edge attribute embedding loss ($\mathcal{L}_{Att}$) and the sketch-edge matching loss ($\mathcal{L}_{SE}$).

$$\begin{aligned}\mathcal{L}_E = &\mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[\|E_{SE}(\tilde{\mathbf{x}}_e) - \mathbf{z})\|_2] \\ &+ \mathop{\mathbb{E}}_{\tilde{x}\sim P_r}[\|E_{SE}(\mathbf{x}_s) - E_{SE}(\mathbf{x}_e)\|_2]\end{aligned} \tag{4}$$

where $\tilde{\mathbf{x}}_e$, $\mathbf{x}_e$, and $\mathbf{x}_s$ are the encoded features of $\tilde{I}_e$, I_e, and I_s. The architecture of the sketch-edge encoder E_{SE} has the same structure as bicycleGAN [18], which is based on WGAP-GP [19].

3. EXPERIMENTS

We evaluate our approach on the SketchyCOCO dataset [17]. We verify the performance of on the foreground object images while placing the natural-looking adversarial example into the other target (background) image. As such, to an observer, our adversarial examples are indistinguishable from other natural objects in the image, but they lead to wrong classification of the DNN classifier.

Table 1: Performance comparison (classification error, %) of attack success rate on SketchyCOCO.

Methods	*Elephant*	*Giraffe*	*Sheep*	*Zebra*
No Attack	0.00	0.00	4.19	0.00
FGSM [7]	98.30	99.25	100	100
PGD-10 [11]	100	100	100	100
AdvPatch [15]	78.12	86.63	89.03	88.32
Ours	88.75	98.42	99.17	100

Dataset SketchyCOCO is a large scale composite dataset for supporting and evaluating image generation from scene-level freehand sketches [17]. It covers instance freehand sketches including 14 foreground classes (around 700 sketches for each class) and 3 background classes. It also contains 14,081 natural images from the MS COCO dataset [20]. In our experiments, we focused on 4 foreground object classes, which are *elephant*, *giraffe*, *sheep* and *zebra*. For each class, the sketch and natural images are split into two groups: 80% for training and the remaining 20% for testing.

Implementation We select EdgeGan [17] as our backbone generator. We also employ the classifier in EdgeGAN paper, which consists of four masked residual unit (MRU) blocks [21]. In our training, RMSProp is used for EdgeGAN with learning rate of 2×10^{-4} and $\alpha = 0.9$. Adam optimizer is used for the classifier with learning rate of 1×10^{-4} and an exponential decay of 0.95. All modules are implemented with PyTorch on TiTan X.

3.1. Quantitative Results

Table 1 shows the quantitative results on the SketchyCOCO validation set. To evaluate performance, we compare to three well-know adversarial attack methods, which are FGSM [7] and PGD-10 with $\epsilon = 16/255$ [11] for digital adversarial attacks, and AdvPatch [15] with patch size of 16×16 for a physical world attack. As seen in Table 1, our approach is competitive with the digital attacks and outperforms the other physical world attack.

3.2. Qualitative Results

Figure 3 shows some qualitative results of our $SketchAdv$ for *zebra* and *elephant* classes. The first column shows the original freehand sketch images I_s, the second column shows the edge image $\tilde{I}_e$ generated by edge generator G_E, and the third column shows the natural-looking adversarial example $\tilde{I}_{adv}$ generated by our generator G_A. A freehand sketch from a user is converted into its latent space representation in the

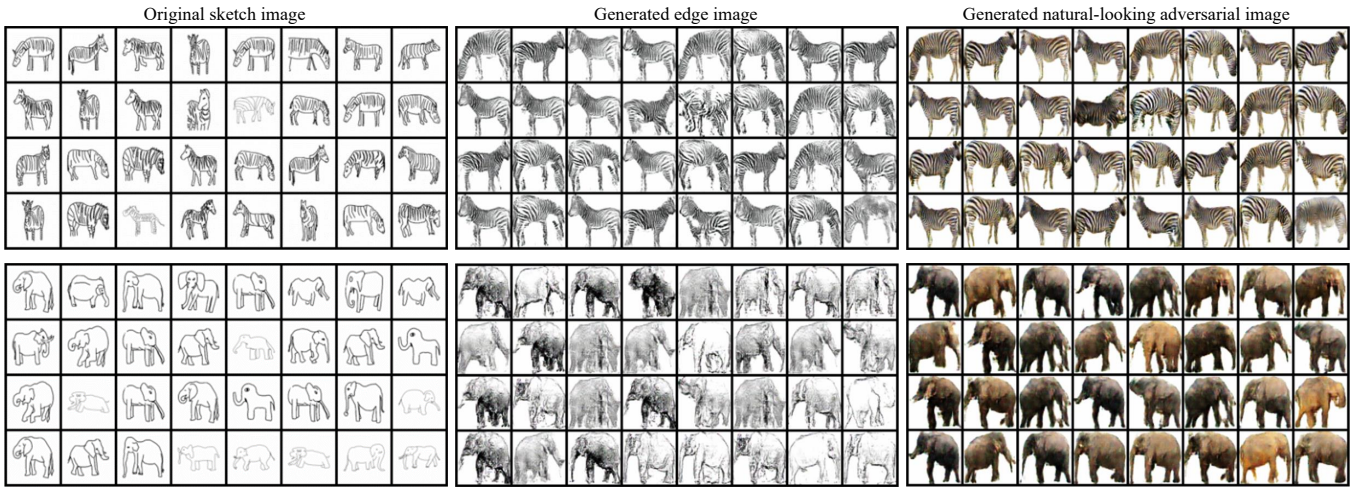

Fig. 3: Some examples of our SketchAdv for the *zebra* and *elephant* classes of SketchyCOCO.

Table 2: The adversarial loss weight η for each training configuration.

Class	2 epochs	20 epochs	100 epochs
Elephant	49	69	100
Giraffe	165	138	112
Sheep	39	25	22
Zebra	82	57	176

sketch-edge domain. From that, both the edge image and the natural-looking adversarial example are generated. Note that a human would not notice whether these images are adversarial examples or not. However, the DNN classifier is very vulnerable to them as seen in Table 1.

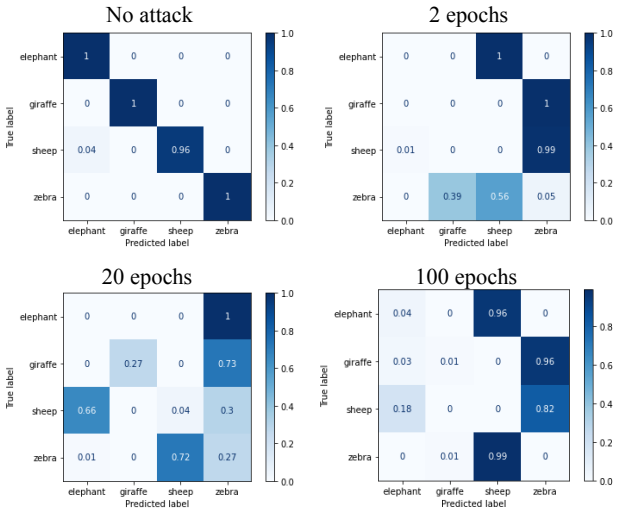

Fig. 4: Visualization of confusion matrix for classification.

3.3. Analysis

Table 2 shows the weight η for the adversarial loss at various training configurations. By exploring the weight for each class according to epochs, we can find the best adversarial loss weight. Note that the proper weight values are different according to the class and training configuration, which is reasonable as each object has different spatial complexity, texture, color, *etc*.

Figure 4 visualizes the confusion matrix between the predicted classes and the ground-truth class labels for different numbers of training epochs. For no attack images (top-left), the classifier provides almost perfect classification performance for the four different classes. However, after only 2 epochs in our training, the classifier is not able to output the correct label as shown in the top-right of Figure 4. Finally, the classifier rarely categorizes properly at 100 epochs (bottom-right).

4. CONCLUSION

We proposed a freehand sketch-based adversarial example generator (*SketchAdv*) to synthesize natural-looking adversarial examples in both the digital and physical worlds. By designing a sketch-edge matching loss, we preserve spatial consistency of the foreground object while increasing the adversarial attack success rate. Future work will expand the applicability; we can naturally embed our adversarial examples in the physical world like on bill boards or T-shirts.

5. ACKNOWLEDGEMENTS

This research was supported by the Chung-Ang University Research Grants in 2021. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2019R1A6A3A12032776)

6. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[3] Li Deng, Geoffrey Hinton, and Brian Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8599–8603.

[4] Jing Huang and Brian Kingsbury, "Audio-visual deep learning for noise robust speech recognition," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7596–7599.

[5] Nariman Farsad, Milind Rao, and Andrea Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2326–2330.

[6] Kyeongpil Kang, Kyohoon Jin, Soyoung Yang, Soojin Jang, Jaegul Choo, and Youngbin Kim, "Restoring and mining the records of the joseon dynasty via neural language modeling and machine translation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 4031–4042.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[8] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[10] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7714–7722.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[12] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[13] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1000–1008.

[14] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al., "Adversarial examples in the physical world," 2016.

[15] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[16] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li, "Semanticadv: Generating adversarial examples via attribute-conditioned image editing," in *European Conference on Computer Vision*. Springer, 2020, pp. 19–37.

[17] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou, "Sketchycoco: Image generation from freehand scene sketches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5174–5183.

[18] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman, "Multimodal image-to-image translation by enforcing bi-cycle consistency," in *Advances in neural information processing systems*, 2017, pp. 465–476.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[20] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209–1218.

[21] Wengling Chen and James Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425.