I2S^2: Image-to-Scene Sketch Translation Using Conditional Input and Adversarial Networks

Daniel McGonigle[1], Tianyang Wang[2], Juefei Yuan[1], Kai He[1], Bo Li[1*]
[1]University of Southern Mississippi, Long Beach, USA
[2]Austin Peay State University, Clarksville, USA

***Abstract*—Image generation from sketch is a popular and well-studied computer vision problem. However, the inverse problem image-to-sketch (I2S) synthesis still remains open and challenging, let alone image-to-scene sketch (I2S^2) synthesis, especially when full-scene sketch generations are highly desired. In this paper, we propose a framework for generating full-scene sketch representations from natural scene images, aiming to generate outputs that approximate hand-drawn scene sketches. Specifically, we exploit generative adversarial models to produce full-scene sketches given arbitrary input images that are actually conditions which are incorporated to guide the distribution mapping in the context of adversarial learning. To advance the use of such conditions, we further investigate edge detection solutions and propose to utilize Holistically-nested Edge Detection (HED) maps to condition the generative model. We conduct extensive experiments to validate the proposed framework and provide detailed quantitative and qualitative evaluations to demonstrate its effectiveness. In addition, we also demonstrate the flexibility of the proposed framework by using different conditional inputs, such as the Canny edge detector.**

***Index Terms*—image generation, scene sketch, image-to-scene sketch translation, conditional input, generative adversarial networks, edge map**

I. Introduction

Image-to-Image (I2I) translation has received a lot of attentions [1], [2], [3], [4] due to its many applications, including generating new data for training deep learning models. If we consider human-drawn sketches as a special type of image, then this problem comprises two subproblems: Sketch-to-Image (S2I) and Image-to-Sketch (I2S) translation. However, till now researchers have mainly focused on the S2I problem, including all the aforementioned research works, and also only considered single object-based sketches. According to our knowledge, there is no published research work in the Image-to-Scene-Sketch (I2S^2) research direction.

However, there is an urgent need to curate a large-scale scene sketch dataset in order to train deep learning models for related applications, such as 2D scene sketch-based 3D scene retrieval [5]. Currently available and related scene sketch/contour datasets [6], [7] are either too small in terms of size or suffer in quality due to limited intra-class variations. For example, Berkeley Segmentation Dataset and Benchmark (BSDS500) [6] has only 500 natural images, and 2,500 contour sketches in total, while the Photo-Sketching dataset [8] has 5,000 contour images for 1,000 outdoor scene images. The SketchyScene dataset [7] composed each scene sketch by selecting among a limited number of pre-defined object sketches, thus can not meet our requirement in generating realistic scene sketches. Due to lack of available high-quality 2D scene sketch data, collecting/generating a large number of scene sketches for training deep learning models for related applications is also a challenging task, even by using Amazon Mechanical Turk. Therefore, we are considering an automatic way to generate 2D scene sketches by using the existing large amount of 2D natural images by training a Generative Adversarial Network (GAN) model [9], that is developing a GAN-based Scene Sketch generation approach, dubbed **SceneSketchGAN**.

Fig. 1: Example sketches generated by our method based on given images. Row 1: given images. Row 2: generated sketches.

The main challenges involved in human-drawn scene sketch generation are mainly related to the inherent characteristics of human sketching: people draw sketches in different styles and at different levels of abstraction. This poses a highly under-constrained problem for us. Motivated by the success of CycleGAN [3] in handling a similar problem: generating images from unpaired data, we adopt a similar framework. However, we found it is still challenging to develop an end-to-end solution which generates satisfactory results due to the problem's much larger domain shift between the images and sketches. Then, to add more constraints to the CycleGAN model to solve this under-constrained problem, we need to provide a conditional input, instead of the original image. Using different types of conditional inputs will generate human-drawn sketches with different styles and/or levels of abstraction. This motivates us to further investigate the role

*Corresponding author. E-mail: bo.li@usm.edu or li.bo.ntu0@gmail.com.

2375-0197/20/$31.00 ©2020 IEEE
DOI 10.1109/ICTAI50040.2020.00123

of conditional inputs in training a generative model for the problem of scene sketch generation. Finally, we utilize a feature selection process by providing the Holistically-nested Edge Detection (HED) [10] map of a natural scene image as the conditional input, rather than using the raw natural image directly. Therefore, our framework can be generalized as **Edge Map + CycleGAN**, as demonstrated in Fig. 3. We conduct extensive experiments including ablation studies to evaluate the proposed framework, and both quantitative and qualitative results demonstrate the effectiveness and competitiveness of our method. We illustrate several generated sketch examples in Fig. 1. More results can be found in the experiments section and our project homepage[1].

In a word, our contributions can be concluded into three-fold:

- We propose a new research problem image-to-scene sketch (I2S^2) translation, which has an urgent need in building large-scale benchmarks to train deep models in advancing related scene sketch-based 3D scene retrieval, recognition, and processing applications.
- We evaluate different conditional inputs for image-to-scene sketch generation and demonstrate that edge-map is suitable for this task in terms of distribution mapping.
- We present a simple yet effective framework to leverage HED edge map-based feature selection (input conditioning) and a CycleGAN-based distribution mapping to generate appealing hand-drawn scene sketches.

II. Related Work

Sketch-to-Image and Image-to-Sketch Synthesis. Based on sketches, we can generate corresponding images of different styles, which can be found in recent image-to-image translation algorithms based on different types of GANs: deep convolutional GAN (DCGAN) [1], conditional GAN (cGAN) [2], CycleGAN [3], and SketchyGAN [4]. People also developed related GAN evaluation metrics, such as Fréchet Inception Distance (FID) [11] to quantitatively compare the generated results of different GAN-based approaches.

However, there is much less research work on the other direction: image-to-sketch synthesis. Berger et al. [12] proposed to generate portraits coming from the same artistic style at different levels of abstraction based on an image. Li et al. [13] proposed an algorithm to perform perceptual grouping of the semantic parts of a sketch, and then generated sketches from an image by utilizing a human stroke dataset and a deformable stroke model-based optimization approach.

Edge Detection. Edge detection refers to finding extreme gradient values relative to neighboring pixels. It is for this reason that Li et al.'s Photo-Sketching project [8] is the most closely-related research to the work presented in this paper. In addition, we tried to gain insights from some of the more successful models whose aim was to map sketches to photo-realistic images. Recently, an approach named Holistically-nested Edge Detection (HED) [10] has been developed to detect good edge images for the holistic image training and prediction, as well as multi-scale and multi-level feature learning vision problems. It adopts an image-to-image translation approach based on a deep learning model.

Related Paired Image-Sketch Datasets. To train our GAN-based model, we need to provide paired image-sketch datasets, such as Berkeley Segmentation Dataset and Benchmark (BSDS500) [6], the Sketchy Database [14], the SketchyScene dataset [7], and the Photo-Sketching dataset [8]. We can also pair the 30 classes of scene images and sketches [15] in the Eurographics Shape Retrieval Contest (SHREC) 2019 2D Scene Sketch-Based [5] and Image-Based [16] 3D Scene Retrieval Benchmarks to form a new image-sketch pair dataset.

However, among all of the aforementioned datasets, only BSDS500 and the Photo-Sketching datasets have the best 2D image-2D sketch (in fact, sketch images are contour images) matching quality (i.e. accuracy in feature correspondence). For other datasets, either the matching quality is low such as the Sketchy and SketchyScene datasets, or the images and sketches are only matching at the category level, instead of at the appearance level, such as the SHREC-based generated one. BSDS500 has 500 natural images, while in average each image has five different early aligned contour images annotated by five subjects. The recently built Photo-Sketching dataset is much bigger. It has 5,000 roughly aligned contour images for 1,000 outdoor scene images.

Sketch Style, Abstraction, and Quality. In the paper, we define "sketch" as an abstract picture drawn by a non-professional human using certain sketching techniques to represent an object or a scene. It is difficult to quantitatively measure the styles and abstraction levels of different human sketches. Therefore, most of existing related research works adopt a data-driven approach [12], [13] to learn different models for them. Muhammad et al. [17] regarded the sketch abstraction level of a sketch as a tradeoff between its recognizability and the number of strokes it contains, and proposed a sketch abstraction model through a stroke removal process guided by reinforcement learning. Kudrowitz et al. [18] proposed that we can measure the sketch quality of a sketch image based on its line work, perspective, and proportions and then found that higher quality sketches contribute to a higher ranking of their creativity levels.

III. Methodology

In this section, firstly we introduce our motivation for input conditioning, and then discuss a feasible solution to extract desired conditional input. Secondly, we analyze the methods used for distribution mapping for sketch generation, and propose to use CycleGAN [3] to perform such mapping. Finally, we present a framework to leverage each component for full-scene sketch generation.

A. Conditional Input

Sketch generation from a given arbitrary input image can be regarded as a conditioned-generation task. Formally, given an input image x, the corresponding sketch y can be obtained

[1]URL: https://github.com/I2S2/Image-to-Scene-Sketch-Translation/.

Fig. 2: Sketches generated by the CycleGAN using the given images as direct inputs. Row 1: given images. Row 2: generated sketches.

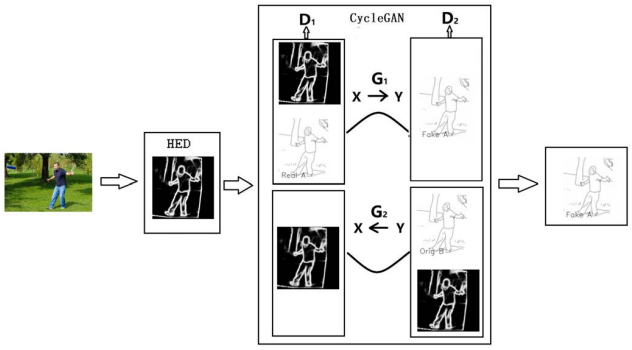

Fig. 3: The proposed framework I2S^2 for full-scene Image-to-Scene Sketch translation. A natural image goes through two stages: HED edge detection-based feature selection and CycleGAN-based distribution mapping. G_1 and G_2 are two generators, while D_1 and D_2 are two discriminators.

by mapping x to y using a distribution mapping function g, having $y = g(x)$.

Nevertheless, image-to-sketch generation is quite different from regular generation tasks. Using a regular input image directly may lead to poor performance. As an example, we adopt regular images as the inputs for a generative model (e.g. CycleGAN), and train the model to generate sketches. The results are unsatisfactory, as shown in Fig. 2. In traditional image generation tasks [1], [2], [3], [4], the generated images contain ample information, and it is relatively less challenging to perform a mapping from randomly sampled inputs to the generated results in light of GAN [9] or Variational Auto-Encoder (VAE) [19] theories. However, as a sparse image, our target sketches usually contain much fewer clues than regular images and are far from sources in terms of details. This makes the traditional image generation pipeline not an ideal candidate for image-to-sketch generation.

In fact, the above analysis indicates that using a regular image as the input for sketch generation is not a good option. Therefore, we are motivated to explore using some other format of an image as the input, namely conditional input in this paper. To leverage GAN or VAE models to generate satisfactory sketches, one natural solution would be using a conditional input that has fewer minor details than its original image. In this paper, we empirically observe that using the edge map of an arbitrary image as the input can help the model to generate appealing sketches. The rationale can be generalized into two-fold. Firstly, edge detection is well-studied and an edge map can be conveniently extracted from a given arbitrary input image. Secondly, edge detection is similar to sketch generation in terms of functionality. As a result, we exploit an edge map as conditional input for the generative model in this work.

B. Edge Detection-Based Conditional Input

Edge detection is a well-studied and widely used technology in image processing. Typical methods include the Canny detector, Sobel detector, Prewitt detector, etc. Technically, any edge detector can be employed to provide a conditional input for our task. However, these traditional edge detection methods have a common issue: lack of ability to produce edges at different scales and levels for images that may have a lot of variations in properties such as contrast and hue. This problem becomes immediately apparent when one applies an edge detection method such as the Canny or Sobel detector to an entire dataset, as some images may yield a good edge map but others may not. In addition, traditional methods, such as the Canny detector, may need additional thresholds to specify the sensitivity of edge detection to determine appropriate thresholds.

The Holistically-nested Edge Detection (HED) method [10] addresses the mentioned issues by using multiple receptive fields of various sizes to produce multiple edge maps in parallel, and deep supervision to weigh each output map appropriately. As a result, it can effectively extract edge features in image regions having sharp contrast, thus producing a more complete edge map. Such ability makes HED more suitable for our image-to-sketch generation task. Firstly, the training process of the discriminator of our adopted GAN model during the adversarial learning will benefit from a more complete and accurate edge map, because the discriminator cannot be easily cheated unless the generated sketches are also complete. This will in turn boost the performance of the generator to generate better quality sketches. Secondly, with complete edge information, the generator is found to be more likely to produce reasonable full-scene sketches, while incomplete edge information often fails to provide sufficient conditioning and constraints for the generator's inference.

Based upon the above analysis, we adopt the HED method as the conditioning input function g (see Section III-A). Specifically, to accommodate it in an end-to-end fashion, we utilize the pre-trained HED model to generate an edge map for a given input natural image, and then feed the produced edge map into the generative model which will be detailed in the following section. During training, we freeze the weights of the HED model, and only update the weights of the generative model.

C. Generative Model-Based Scene Sketch Generation

There are two branches of generative models, namely GAN and VAE. In this work, we exploit a GAN structure to generate sketches, but our framework can be easily changed to accommodate a VAE structure as the generative model. We employ a dataset in which one image corresponds to multiple sketch labels. One option is still using a GAN structure that favors a 1:1 match for the image pairs, and designing a new loss to measure the average distance for all labels. In our work, we argue that CycleGAN [3] is more suitable for the selected dataset, because it was developed to map an image from an input domain to a target domain without having to be a 1:1 match for the image pairs [3].

As shown in Fig. 3, CycleGAN utilizes two pairs of generators (G_1, G_2) and discriminators (D_1, D_2) to map back and forth between source (images) and target (sketches) domain feature spaces X and Y. During training, the original input image x is mapped to the target domain by generator G_1, and then back to the original source domain by generator G_2. Meanwhile, the target sketch is also being mapped to the source image domain and then back again to the target domain. The introduced cycle consistency loss $\mathcal{L}_{\mathrm{CyC}}$ measures the L_1 loss between the original images / target sketches and their respective reconstructions via both generators,

$$\begin{aligned}\mathcal{L}_{\mathrm{CyC}}(G_1, G_2) =& \mathbb{E}_{\mathrm{x}\sim \mathrm{p_{data}(x)}}\|G_2(G_1(x)) - x\|_1 + \\ & \mathbb{E}_{\mathrm{y}\sim \mathrm{p_{data}(y)}}\|G_1(G_2(y)) - y\|_1.\end{aligned} \tag{1}$$

The loss function of CycleGAN is thus defined as follows,

$$\begin{aligned}\mathcal{L}(G_1, G_2, D_1, D_2) =& \mathcal{L}_{\mathrm{GAN}}(G_1, D_2, X, Y) + \\ & \mathcal{L}_{\mathrm{GAN}}(G_2, D_1, Y, X) + \\ & \lambda \mathcal{L}_{\mathrm{CyC}}(G_1, G_2),\end{aligned} \tag{2}$$

where λ is a hyperparameter indicating the relative weight of the cycle consistency loss compared with the GAN loss.

We empirically determine the optimal architecture and configuration of the CycleGAN for the purpose of generating appealing sketches. Each generator is implemented as a 9-layer ResNet [20]. The discriminators adopt the PatchGAN structure [2]. We train the entire generative model by following the CycleGAN pipeline.

It is worth noting that the original identity mapping is used to prevent the model from making any drastic changes when the image or target is close to their respective counterparts. However, we observe that it also helps to prevent producing too many details in our generated sketches. This is highly expected since sketches should be clear and simple, which is essentially different from edge detection. Moreover, it is convenient to control the quality of generated sketches by adjusting the cycle loss weight, leading to more realistic sketches. We detail the analysis in Section IV.

D. Framework

Our entire framework for full-scene sketch generation is illustrated in Fig. 3. Our framework only exploits HED and CycleGAN. However, it is general enough to be easily replaced with other methods for other purposes. For instance, we investigate the combination of Canny and CycleGAN in Section IV, and observe that it can also produce acceptable sketches. We would like to highlight that our work aims to explore an effective framework for image-to-scene sketch generation. We illustrate an example of an input image, its conditional input, and the output result in Fig. 4.

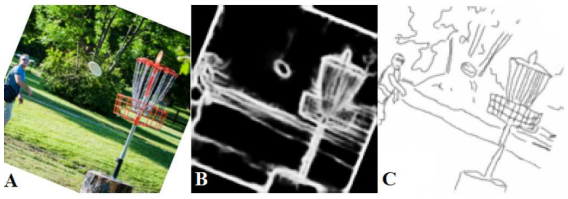

Fig. 4: Sketch generation example with our model. (A) represents a given color image, (B) is the corresponding conditional input, and (C) is a generated full-scene sketch.

IV. Experiments and Discussions

To demonstrate the effectiveness of our framework, we detail our extensive experiments in this section. We firstly introduce the dataset adopted for the experiments. Then, we introduce the evaluation metrics used to quantitatively evaluate the proposed method and training details, followed by a qualitative analysis of the results. Finally, we provide insights through discussions for the potential usage of our framework. Our code will be shared via the project homepage, as well.

A. The Photo-Sketching Dataset

We exploit the dataset curated by Li et al. [8] in our experiments. They crawled a dataset of 1,000 outdoor images from Adobe Stock, and each image is paired with 5 drawings. They selected 5,000 high-quality drawings from this dataset. It is ideal for our task due to two main reasons. Firstly, each image in this dataset corresponds to five targets that include various degrees of details. This property is beneficial for full-scene sketch generation. Secondly, the contour maps cover almost all the objects in the corresponding images, which encourages our model to generate every significant object that is present in the image. We follow a general practice to augment the training images by flipping, rotation, and translation.

B. Evaluation Metrics

To quantitatively evaluate our method, we adopt the Fréchet Inception Distance (FID) [11], Sørensen-Dice coefficient (Dice, a.k.a F-score, F-measure) [21], sensitivity (SN, a.k.a "recall" or "hit rate"), and accuracy (Acc). Except FID, higher values are better. FID uses the output of the third layer of the Inception-v3 network trained on the ImageNet dataset in order to measure the earth-mover distance between the generated distribution and target distribution. One main advantage of using FID for evaluation is that we compare related statistics in the feature space rather than doing that at the pixel-level. This is especially important for the image-to-sketch

generation task, because a sketch image contains insufficient pixel information and most pixels are background. A lower FID score corresponds to a higher degree of similarity between images. The fluctuation due to differences in trained weights is small (less than 10% in instances mentioned in [22]), and the domain of object classes we use to train is also small, making FID an appropriate metric for the evaluation of generated sketches.

For Dice, sensitivity, and accuracy, true positive (TP) pixels represent target sketch pixels; false positive (FP) pixels represent background pixels incorrectly generated as sketch pixels. True negative (TN) and false negative (FN) refer to the truth of whether the pixel belongs to the background and is not part of the sketch. The Dice-Sørensen Coefficient (Dice), sensitivity (SN), and accuracy (Acc) are defined as follows: $Dice = \frac{2TP}{2TP+FP+FN}$, $SN = \frac{TP}{TP+FN}$, $Acc = \frac{TN+TP}{TN+TP+FN+FP}$.

Dice score can also be viewed as a ratio of intersection of predicted sketch pixels to union of predicted and actual sketch pixels. This metric is commonly used in image segmentation. Sensitivity measures the true positive rate, or recall of the generated sketch. Accuracy measures the ratio of correctly-placed pixels to the total number of pixels. These metrics are all pixel-wise evaluations to measure how well the generated sketch matches the target sketch. We strive to avoid using evaluation methods for boundary or contour detection, since for generating sketches our goal is the quality of the sketch, rather than simply extracting the locations and configuration of contours.

C. Training Settings

It is worth noting that in our framework, the first component used to provide conditional input is a pre-trained model, and its results are not subject to change with different training settings. Our training details will only affect the second component of the framework, that is, the CycleGAN part. Here, we only introduce the training settings which lead to the best results we have observed. We train the model using the Adam optimizer with a learning rate of 0.0002. The batch size and the weight of identity loss are set to 1 and 0.5, respectively. We adjust the weights of the cycle consistency loss for the two generators to 20%, that is λ=0.2. The model is trained for 30 epochs. For other settings, we strictly follow the practice of training a CycleGAN for the purpose of fair comparisons.

D. Results and Discussions

To demonstrate the competence of our framework, we compare it with other methods, such as HED [10], and Photo-Sketching [8]. It is important to note that the HED method was not designed for sketch generation, and to our best knowledge there are very few works focusing on image to full-scene sketch generation. Therefore, we still add HED in our comparison, considering edge detection is one of the closest general image processing tasks to our image-to-sketch generation problem. We present the quantitative results of each method in Table I, based on the Photo-Sketching dataset and the aforementioned metrics. As can be seen, our method

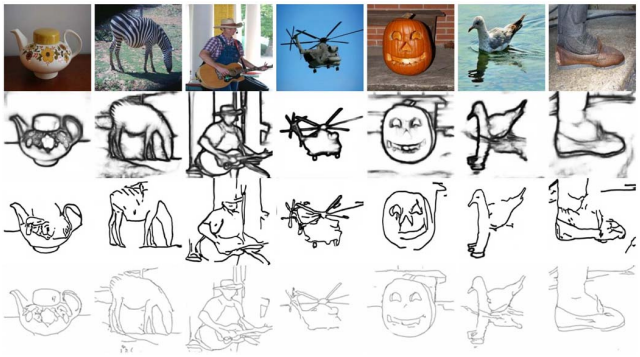

Fig. 5: Qualitative evaluations of different methods. Row 1: given images. Row 2: results of HED [10]. Row 3: results of Photo-Sketching [8]. Row 4: our results.

Fig. 6: Generated sketches when different loss functions are employed to train the generative model. Row 1: given images. Row 2: results of the WGAN-loss [22] (**WGAN+**). Row 3: results of the CycleGAN-loss (Our approach).

always outperforms either Photo-Sketching or HED in terms of FID, Dice Score, Accuracy and Sensitivity. However, when a regular image is directly used as the input of the generator, that is, the **CycleGAN** only method, our framework without conditional input has inferior performance than the competitors in terms of all the four metrics except FID. It indicates the necessity of exploiting an edge map as conditional input. But when the Canny edge detector is adopted to provide the conditional input, giving the **Canny+** approach, even though its performance is still competitive, it greatly falls behind our results. It can be observed that using different loss functions also have an impact on the results. The CycleGAN loss has demonstrated more robust and also better performance than the Wasserstein loss (**WGAN+**) for our framework.

We further compare the qualitative results by giving three sets of typical examples in Figs. 5∼7. We observe that the HED method [10] tends to generate too many edge details, and the results are not like hand-drawn sketches. While the quality of the sketches generated by the Photo-Sketching method [8] is generally better, but they often miss a significant number of important feature lines, as well as critical visual cues. On the

Metric/Method	Photo-Sketching [8]	HED [10]	Canny+	CycleGAN	WGAN+	Ours
FID	103.268	255.942	54.549	47.516	121.575	**32.626**
Dice	0.330	0.293	0.246	0.128	0.197	**0.765**
Acc	0.916	0.842	0.871	0.883	0.912	**0.972**
SN	0.449	0.690	0.445	0.183	0.248	**0.994**

TABLE I: Quantitative evaluations of sketches generated by different methods. For FID, the lower the better, and the higher the better for the other metrics. In **Canny+**, we adopt the Canny detector to detect the edge map from an image, and use this edge map as the conditional input. In **CycleGAN**, we directly use regular images as conditional inputs. In **WGAN+**, Wasserstein loss is used within our framework.

Fig. 7: Generated sketches when different conditional inputs are used. Row 1: given images. Row 2: when the conditional input is provided by the Canny edge detector (**Canny+**). Row 3: when the conditional input is provided by the HED method [10] (Our method).

contrary, our results are much closer to hand-drawn sketches with necessary and proper level of details. Fig. 6 indicates that, compared with the WGAN loss (**WGAN+**), the CycleGAN loss is more helpful to robustly produce appealing results. In Fig. 7, **Canny+**) often generate inferior results than our HED-based approach. All of these further validate our best configurations for our proposed **Edge Map + GAN** image-to-scene sketch framework: **HED + CycleGAN**.

V. Conclusions

We propose a flexible framework for image to full-scene sketch generation in this paper. We demonstrate that different components can be exploited in this framework to achieve multiple levels of results. We investigate the impact of conditional input and demonstrate the necessity of edge map for appealing sketch generation from a regular image. We also analyze the distribution mapping problem in the context of sketch generation and demonstrate the suitability of CycleGAN for sketch generation. The effectiveness of the proposed framework is validated through extensive experiments, and it is convenient to setup the framework to produce human-drawn like sketches.

References

[1] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: ICLR'16, San Juan, Puerto Rico, May 2-4, 2016.

[2] P. Isola, J. Zhu, T. Zhou, A. A. Efros, Image-to-Image translation with conditional adversarial networks, in: CVPR'17, Honolulu, HI, USA, July 21-26, 2017, pp. 5967–5976.

[3] J. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: ICCV'17, Venice, Italy, October 22-29, 2017, pp. 2242–2251.

[4] W. Chen, J. Hays, SketchyGAN: Towards diverse and realistic sketch to image synthesis, in: CVPR'18, Salt Lake City, UT, USA, June 18-22, 2018, pp. 9416–9425.

[5] J. Yuan, et al, SHREC'19: Extended 2D scene sketch-based 3D scene retrieval, in: 3DOR'19, Genoa, Italy, May 5-6, 2019, pp. 33–39.

[6] D. R. Martin, C. C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: ICCV'01, Vancouver, British Columbia, Canada, July 7-14, Volume 2, 2001, pp. 416–425.

[7] C. Zou, et al, SketchyScene: Richly-annotated scene sketches, in: ECCV'18, Munich, Germany, September 8-14, Part XV, 2018, pp. 438–454.

[8] M. Li, Z. L. Lin, R. Mech, E. Yumer, D. Ramanan, Photo-Sketching: Inferring contour drawings from images, in: WACV'19, Waikoloa Village, HI, USA, January 7-11, 2019, pp. 1403–1412.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, Y. Bengio, Generative adversarial nets, in: NIPS'14, December 8-13, Montreal, Canada, 2014, pp. 2672–2680.

[10] S. Xie, Z. Tu, Holistically-nested edge detection, International Journal of Computer Vision 125 (1-3) (2017) 3–18.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local nash equilibrium, in: NIPS'17,December 4-9, Long Beach, CA, USA, 2017, pp. 6626–6637.

[12] I. Berger, A. Shamir, M. Mahler, E. J. Carter, J. K. Hodgins, Style and abstraction in portrait sketching, ACM Trans. Graph. 32 (4) (2013) 55:1–55:12.

[13] Y. Li, Y. Song, T. M. Hospedales, S. Gong, Free-hand sketch synthesis with deformable stroke models, International Journal of Computer Vision 122 (1) (2017) 169–190.

[14] P. Sangkloy, N. Burnell, C. Ham, J. Hays, The Sketchy database: learning to retrieve badly drawn bunnies, ACM Trans. Graph. 35 (4) (2016) 119:1–119:12.

[15] J. Yuan, H. Abdul-Rashid, B. Li, Y. Lu, Sketch/image-based 3D scene retrieval: Benchmark, algorithm, evaluation, in: MIPR'19, San Jose, CA, USA, March 28-30, 2019, pp. 264–269.

[16] H. Abdul-Rashid, et al, SHREC'19: Extended 2D scene image-based 3D scene retrieval, in: 3DOR'19, Genoa, Italy, May 5-6, 2019, pp. 41–48.

[17] U. R. Muhammad, Y. Yang, Y. Song, T. Xiang, T. M. Hospedales, Learning deep sketch abstraction, in: CVPR'18, Salt Lake City, UT, USA, June 18-22, IEEE Computer Society, 2018, pp. 8014–8023.

[18] B. M. Kudrowitz, P. Te, D. R. Wallace, The influence of sketch quality on perception of product-idea creativity, AI EDAM 26 (3) (2012) 267–279.

[19] D. P. Kingma, M. Welling, Auto-encoding variational bayes (2013). arXiv:1312.6114.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR'16, Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778.

[21] L. R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.

[22] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: ICML'17, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 214–223.