Reference Based Sketch Extraction via Attention Mechanism

AMIRSAMAN ASHTARI[*], Visual Media Lab, KAIST, South Korea
CHANG WOOK SEO[*], Visual Media Lab, KAIST, South Korea
CHOLMIN KANG, Visual Media Lab, KAIST, South Korea
SIHUN CHA, Visual Media Lab, KAIST, South Korea
JUNYONG NOH, Visual Media Lab, KAIST, South Korea

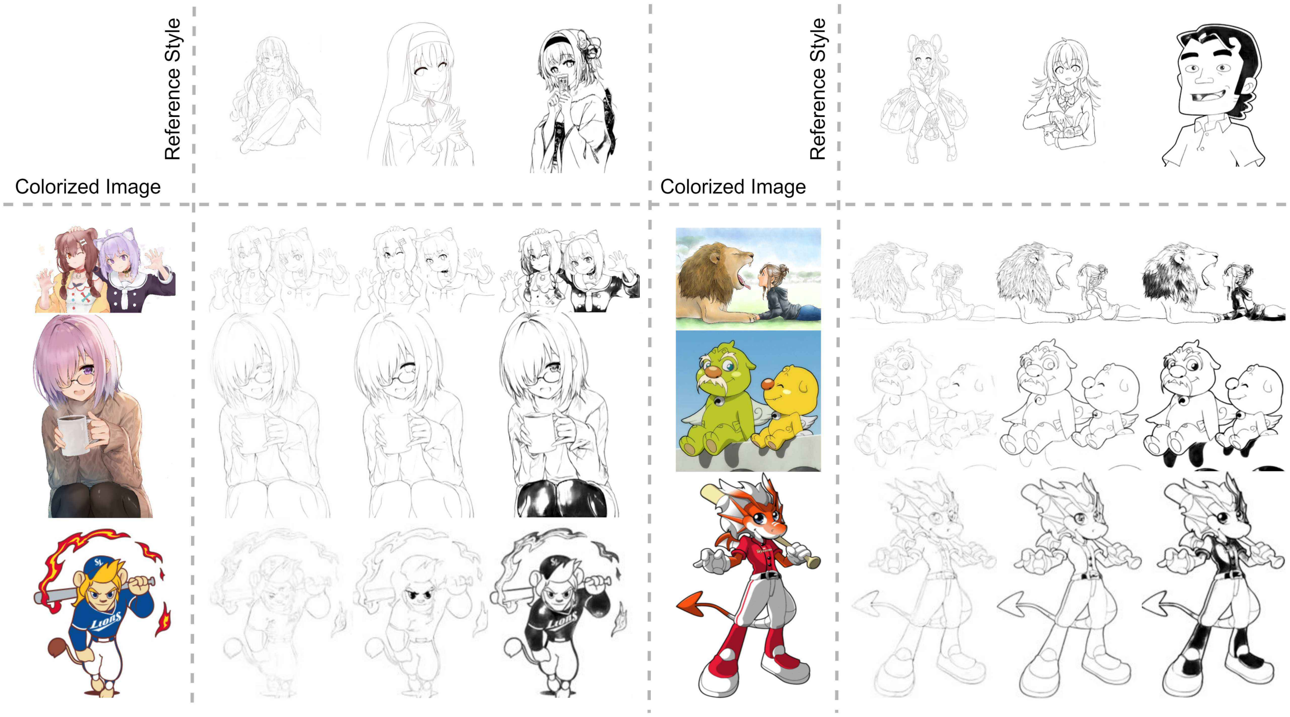


Fig. 1. We propose a method to extract a sketch from a colorized image with the style similar to that of a reference sketch and with the content identical to that of the colorized image. © left to right in reference style: Comete_atr, TK_painter, Ayul_oekaki, Chobi_chu, Comete_atr, SeoulCityBrand; up to down in colorized image: Comete_atr, Ayul_oekaki, SamsungLions, AkaneNagano, SeoulCityBrand, SSG Landers Professional Baseball Team.

We propose a model that extracts a sketch from a colorized image in such a way that the extracted sketch has a line style similar to a given reference sketch while preserving the visual content identically to the colorized image. Authentic sketches drawn by artists have various sketch styles to add visual interest and contribute feeling to the sketch. However, existing sketch-extraction methods generate sketches with only one style. Moreover, existing style transfer models fail to transfer sketch styles because they are mostly designed to transfer textures of a source style image instead of transferring the sparse line styles from a reference sketch. Lacking the necessary volumes of data for standard training of translation systems, at the core of our GAN-based solution is a self-reference sketch style generator that produces various reference sketches with a similar style but different spatial layouts. We use independent attention modules to detect the edges of a colorized image and reference sketch as well as the visual correspondences between them. We apply several loss terms to imitate the style and enforce sparsity in the extracted sketches. Our sketch-extraction method results in a close imitation of a reference sketch style drawn by an artist and outperforms all baseline methods. Using our method, we produce a synthetic dataset representing various sketch styles and improve the performance of auto-colorization models, in high demand in comics. The validity of our approach is confirmed via qualitative and quantitative evaluations.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**; **Computer vision problems**.

Additional Key Words and Phrases: Sketch-extraction, Auto-colorization, Image-to-image translation

[*] These authors contributed equally to this work and are listed alphabetically.

Authors' addresses: Amirsaman Ashtari, Visual Media Lab, KAIST, South Korea, a.s.ashtari@kaist.ac.kr; Chang Wook Seo, Visual Media Lab, KAIST, South Korea, lgtwins@kaist.ac.kr; Cholmin Kang, Visual Media Lab, KAIST, South Korea, stevekang@kaist.ac.kr; Sihun Cha, Visual Media Lab, KAIST, South Korea, chacorp@kaist.ac.kr; Junyong Noh, Visual Media Lab, KAIST, South Korea, junyongnoh@kaist.ac.kr.

0730-0301/2022/12-ART207 $15.00
https://doi.org/10.1145/3550454.3555504

ACM Reference Format:
Amirsaman Ashtari, Chang Wook Seo, Cholmin Kang, Sihun Cha, and Junyong Noh. 2022. Reference Based Sketch Extraction via Attention Mechanism. *ACM Trans. Graph.* 41, 6, Article 207 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555504

1 INTRODUCTION

Sketches play an important role in manga and anime, widely studied in the computer graphics literature [Li et al. 2017; Liu et al. 2013, 2015; Sasaki et al. 2017, 2018; Simo-Serra et al. 2018b; Xie et al. 2020, 2021]. Sketching is the fundamental first step for expressing and communicating artistic ideas, which reflects the main structure and content of drawn images [Liu et al. 2015; Peng et al. 2021; Simo-Serra et al. 2018a, 2016a]. A sketch is a relatively simple construction, centering on the properties of its constituent lines. The thickness, angle, continuity and shape of each of the lines typically contribute to the unique style of a sketch, designed to appeal to a viewer [Fish et al. 2020]. Drawing with one basic line for which the width does not change is monotonous and boring.

In order to make sketch drawings more interesting, artists add variety to each line by varying the line quality or equivalently the line weight[1]. Line quality refers to the thickness or thinness of the line, which contributes to the style of a sketch. Line quality is an essential element for creating engaging line art or sketches and is known to be one of the most important aspects of manga and comic storytelling, but is often disregarded[2]. Using many different types of lines is a way to add feeling or mood to a drawing[3] (e.g., smooth and easy or rough and aggressive). To this end, each artist has his/her unique style when drawing sketches for anime characters, as presented in the various line styles from authentic sketches shown in Figure 2.

Authentic sketches drawn by artists have various sketch styles. However, existing sketch-extraction methods such as Canny [Canny 1986], XDoG [Winnemöller 2011], SketchKeras [lllyasviel/sketchKeras 2018], and Anime2Sketch [Xiaoyu Xiang 2021] extract a sketch from a colorized image with only one style, as shown in Figure 2. For example, the Canny method extracts sketches with noisy dotted lines. Line styles extracted by SketchKeras are similar to pencil strokes, which are uniformly thicker or thinner than lines drawn in authentic sketches. The XDoG method mis-paints some parts of the character's body in black, visually different from the original design. The Anime2Sketch model fails to imitate the eyelash style or black inked areas of the authentic sketches.

In addition to sketch-extraction methods, there are several general-purpose edge-detection techniques [Bertasius et al. 2014; Liu et al. 2019a; Xie and Tu 2015] required for various classical computer vision processes [Isola et al. 2017; Li et al. 2019; Yang et al. 2002; Zhang et al. 2016]. The main focus of these studies is not on anime characters as in our case, and similar to sketch-extraction techniques, these methods extract edges or sketches from a colorized image with only one style.

One naive solution by which to extract sketches from colorized images with various styles is to use style transfer models such as the model proposed by Gatys et al. [2016] and MUNIT [Huang et al. 2018]. Style transfer aims to modify the style of an image while preserving its content, which is closely related to image-to-image translation [Huang et al. 2018]. A style transfer algorithm should be able to extract the semantic image content from a target image (i.e., a colorized image in our case) and then inform a texture transfer procedure to render the semantic content of the target image in the style of the source image (i.e., a reference sketch in our cause) [Gatys et al. 2016]. However, when we trained style transfer models such as the model proposed by Gatys et al. and MUNIT using authentic sketches, they failed to generate satisfactory results, as shown in Figure 2. The underlying reason is that sketches are different from colorized style images in that they are constructed by lines which are more spatially sparse in the image space. In addition, in a sketch, the line quality contributes more than the texture to award the sketch its unique style [Fish et al. 2020]. However, in most style transfer models, the goal is to learn to transfer the texture from a given reference style image.

In this article, we propose a model that extracts a sketch from a colorized image in such a way that the extracted sketch has a line style similar to that of a given reference sketch while preserving the visual content identically to the colorized image, as shown in Figure 1. Imitating a reference sketch style is a challenging task. Such a method must precisely detect edges from a colorized image and reference sketch yet must also learn to imitate the line quality. In addition, the method must be able to find visual correspondences between the colorized image and reference sketch for the sketch style transfer because anime characters in these two images may be drawn in different poses or shapes.

Embracing these challenges, we propose a GAN-based solution for sketch style imitation. Lacking the necessary volumes of data for the standard training of translation systems, at the core of our method lies a self-reference sketch style generator that produces various reference sketches with a similar style but different spatial layouts for each pair of a colorized image and its corresponding sketch. In addition, our method leverages three independent attention modules to detect the edges of a colorized image and reference sketch separately as well as the visual correspondences between them. We apply several loss terms to imitate the style and enforce sparsity in the extracted sketches.

We also use our sketch-extraction method to improve the performance of auto-colorization models. Sketch colorization is an expensive, time-consuming, and labor-intensive task in the illustration industry [Kim et al. 2019]. The colorization of sketch images is facing strong demand in relation to comics, animation, and other content-creation applications. However, the industry suffers from information scarcity of authentic sketch images (i.e., drawn by an artist) and their corresponding colorized images [Lee et al. 2020], as there is no well-known large public dataset containing both authentic sketches and their corresponding colorized images. Therefore, most auto-colorization techniques [Kim et al. 2019; Seo and Seo 2021; Zhang et al. 2017, 2018b] and industrial software [style2paint 2018] train their models using synthetic sketches extracted by Canny, XDoG, or SketchKeras from colorized images. Unfortunately, these methods generate sketches in a fixed style, different from authentic sketches that may be used for colorization. Using our method,

[1] https://thevirtualinstructor.com/line-quality-cross-contour.html
[2] https://www.clipstudio.net/how-to-draw/archives/163108
[3] http://www1.udel.edu/artfoundations/drawing/linequality.html

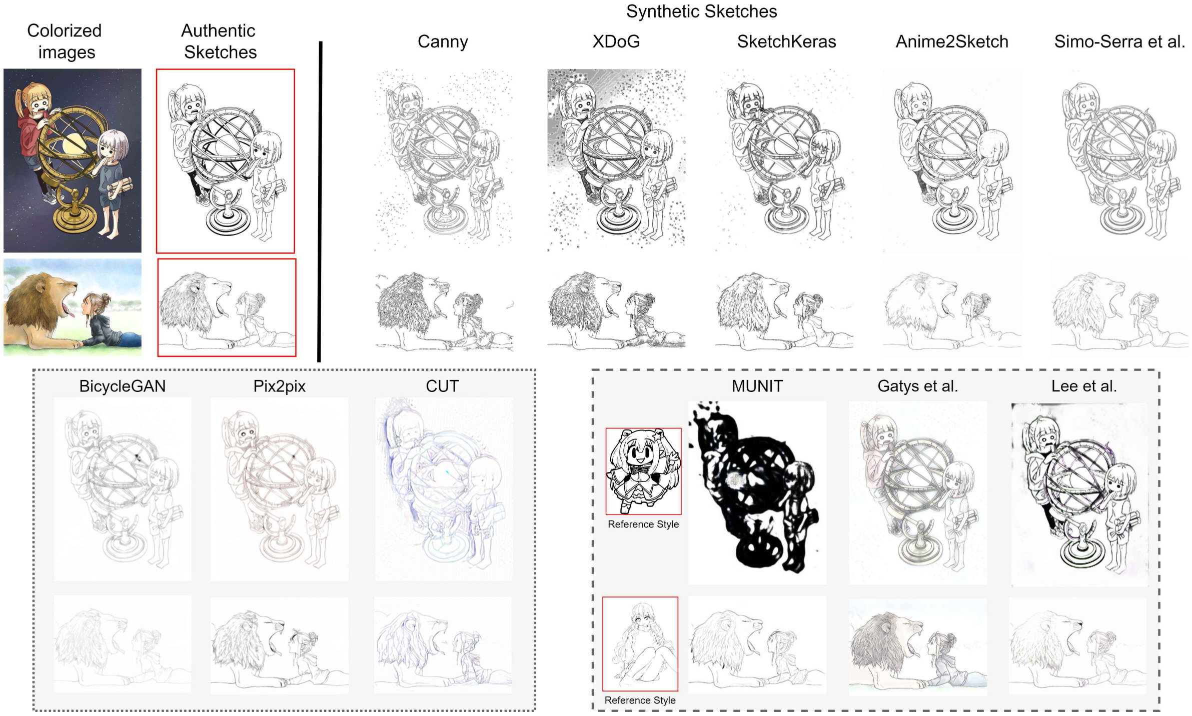


Fig. 2. Authentic sketches drawn by artists have various sketch styles. However, existing sketch-extraction methods extract a sketch from a colorized image with only one style. Moreover, existing image-to-image translation and style transfer models fail to transfer the line style of a reference sketch. © up to down: Dolphilia, AkaneNagano, AonoriwaKame, Comete_atr.

we produce a synthetic dataset with various sketch styles, more representative of real-world scenarios. We show that using our method to produce a synthetic dataset improves the performance of auto-colorization results.

Our sketch-extraction method results in a close imitation of a reference sketch style drawn by an artist while providing a user more flexibility in terms of generating sketches with various styles in a short time. We evaluate our method in a number of qualitative and quantitative experiments, the results of which prove that our method outperforms all existing sketch-extraction techniques. Furthermore, we introduce an evaluation metric which measures how faithfully the model preserves a reference style in multiple consecutive sketch extractions. Finally, a large perceptual study with 200 participants suggests that people easily distinguish the visual differences between various sketch styles and prefer our method over existing baselines. The study confirmed that our method successfully extracts a sketch similar to a given reference style.

2 RELATED WORK

General-purpose edge detectors: Edge detection includes a variety of methods that aim to identify edges at which the image brightness changes sharply. Several general-purpose edge-detection techniques have been proposed for various classical computer-vision tasks such as segmentation [Zhang et al. 2016], image recognition [Yang et al. 2002], image-to-image translation [Isola et al. 2017], and photo sketching [Li et al. 2019]. For a detailed review, see Ziou and Tabbone [2000] and Gong et al. [2018]. General-purpose edge detectors can be categorized into four groups based on the technique used to process the given image [Soria Poma et al. 2020] as follows: 1) Several studies leverage *low-level features* such as brightness or color to detect edges by convolving the image with a Gaussian filter or manually performed kernels [Canny 1986, 1983; Li et al. 2020; Perona and Malik 1990; Winnemöller 2011]. 2) Based on edge formation analyses of the vision systems of monkeys, cats, and humans, several studies have proposed *brain-biologically inspired edge detectors* by modeling the retina or simple cells, or by using Gabor filters or derivatives of Gaussian filters [Akbarinia and Párraga 2018; Grigorescu et al. 2003; Mély et al. 2016; Yang et al. 2015]. 3) Some methods are based on *classical learning algorithms* such as sparse representation learning [Mairal et al. 2008], dictionary learning [Xiaofeng and Bo 2012], gradient descent [Arbeláez et al. 2011], and decision trees [Dollár and Zitnick 2014]. 4) Recent methods employ *deep learning algorithms* based on CNNs to improve the quality of returned edges [Bertasius et al. 2014; Ganin and Lempitsky 2014; Liu et al. 2019a; lllyasviel/sketchKeras 2018; Wang et al. 2017; Xiaoyu Xiang 2021; Xie and Tu 2015].

While all of these interesting methods indeed improve edge-detection capabilities, the edges or sketches extracted from a colorized image are of *only a single style*, unlike our approach, which imitates various sketch styles specifically for anime characters. In addition, the goal of these studies differs from ours in that they neither focus on improving the synthetic data quality required for the training of auto-colorization models nor return aesthetically pleasing sketches similar to an artist sketch style.

Sketch simplification: Sketch simplification methods convert rough sketches into clean line drawings [Simo-Serra et al. 2018a]. Among many studies related to the sketch simplification [Arvo and Novins 2000; Bae et al. 2008; Favreau et al. 2016; Grabli et al. 2004; Hilaire and Tombre 2006; Liu et al. 2015; Noris et al. 2013; Qi et al. 2015; Shesh and Chen 2008; Wilson and Ma 2004], deep CNN based approaches have shown great potential in improving the sketch simplification [Simo-Serra et al. 2018a, 2016b; Xu et al. 2021]. Specifically, Simo-Serra et al. [2016b] proposed a fully-convolutional network to simplify sketches by using paired data and minimizing the MSE loss. Simo-Serra et al. [2018a] further improved their method by employing GANs and incorporating unlabeled real sketches into the learning process. Xu et al. [2021] adopted the multi-layer perceptual loss to preserve semantically important global structures and fine details without incurring blurriness. Inspired by these studies, we incorporate the perceptual loss in our network design and propose a semi-supervised method to deal with a small size of authentic paired data.

Auto-colorization models: Several studies have focused on developing auto-colorization models [Cao et al. 2021; Ci et al. 2018; Fang et al. 2021; Furusawa et al. 2017; HATI et al. 2019; Huang et al. 2005; Iizuka et al. 2016; Kim et al. 2019; Lee et al. 2020; Qu et al. 2006; Seo and Seo 2021; Thasarathan and Ebrahimi 2019; Yuan and Simo-Serra 2021; Zhang et al. 2017, 2018b]. Due to the scarcity of paired sketches and colorized images, most auto-colorization techniques train their models using synthetic sketches extracted by 1) Canny [Huang et al. 2005; Seo and Seo 2021; Thasarathan and Ebrahimi 2019], 2) XDoG [Ci et al. 2018; Fang et al. 2021; HATI et al. 2019; Kim et al. 2019; Lee et al. 2020; Yuan and Simo-Serra 2021], or 3) SketchKeras [Cao et al. 2021; Kim et al. 2019; Seo and Seo 2021; Yuan and Simo-Serra 2021]. However, these sketch-extraction methods generate sketches with only one style, different from authentic sketches with various styles that might be used for auto-colorization. Using our sketch-extraction method, we improve the performance of auto-colorization models by generating a synthetic dataset containing various sketch styles.

Image-to-image translation: One naive solution to extract sketches from colorized images is to use image-to-image translation methods that aim to transfer images from a source domain (e.g., a colorized image) to a target domain (e.g., a sketch) while preserving the content representations. Image-to-image translation models have been trained in both supervised [Isola et al. 2017; Park et al. 2019; Shaham et al. 2021; Zhou et al. 2020; Zhu et al. 2017b] and unsupervised [Cho et al. 2018; Huang et al. 2018; Kim et al. 2020; Park et al. 2020; Zhu et al. 2017a] settings. These image-to-image translation models [Park et al. 2020; Zhu et al. 2017b] based on both supervised and unsupervised learnings only accept a colorized image as their input. Therefore, it is very difficult for a user to manipulate the style of an extracted sketch. In contrast, our method allows a user to control the style of an extracted sketch by providing an example of the desired style as an extra input.

Example-guided style transfer: Example guided style transfer aims to transfer the style of an example image to a target image. Deep neural networks have been widely used for this purpose [Chang et al. 2018; Gatys et al. 2016; Gu et al. 2018; Huang and Belongie 2017; Huang et al. 2018; Johnson et al. 2016a; Liao et al. 2017; Luan et al. 2017; Ma et al. 2019; Yoo et al. 2019; Zheng et al. 2020]. While these methods allow one to control the style of the result by providing an example [Gatys et al. 2016; Huang et al. 2018], they are mostly designed to transfer the texture style of a source image. In contrast, our method is specifically designed to imitate the line quality which contributes more than texture to award the sketch its unique style.

Domain gap: To fill the gap between synthetic and real domains, domain adaptation methods have been employed in many computer vision applications [Liu et al. 2020; Roberts et al. 2021] including semantic segmentation, person re-identification, and object detection [Deng et al. 2018; Hoffman et al. 2018; Sankaranarayanan et al. 2018; Tsai et al. 2018]. There are various methods that aim to solve the domain gap problem, such as learning domain-invariant representations [Ganin and Lempitsky 2015; Ganin et al. 2016] or pushing two domain distributions to be close [Gretton et al. 2012; Sun et al. 2016; Sun and Saenko 2016; Tzeng et al. 2014]. See Wang and Deng [2018] for a review of existing methods. In case of solving the domain gap between synthetic and authentic sketches, imitating the sketch style of human drawings and increasing the style variation of synthetic sketches are required. Our method imitates various sketch styles and generates sketches that better resemble the authentic sketches drawn by artists.

3 METHOD

Training models related to sketches suffers from scarcity of authentic paired data. To address this problem, as one example Simo-Serra et al. [2018a] uses a small set of paired data and a large set of samples only for the target domain to improve the sketch simplification. Similar to this case, training a reference based sketch-extraction model requires a large number of paired data for each sketch style. Unfortunately, only few samples might be available for a specific sketch style. To address this problem, we propose a semi-supervised method that can generate a large number of similar style sketches with different positional layouts using a small set of paired data.

3.1 Overview

Our model extracts a sketch from a colorized image in such a way that the style of the extracted sketch is visually similar to a given reference sketch. As illustrated in Figure 3, the input to our model is a colorized image I_c and reference sketch S_r. The output is an extracted sketch with a similar style to the given reference sketch. Our algorithm performs the following steps to train our sketch-extraction model imitating a reference sketch style:

(A) To generate the reference sketch style S_r using an authentic sketch, we first apply the thin plate splines (TPS) transformation or random flips to a ground truth sketch S_{gt} (Section 3.2). The transformed sketch has a similar style to the ground truth sketch with a spatially different layout.

(B) The colorized image and the generated reference sketch are then fed into two independent encoders $E_c(I_c)$ and $E_r(S_r)$, followed by two independent Convolutional Block Attention Modules (CBAM) [Woo et al. 2018]: $CBAM_c(E_c(I_c))$ and $CBAM_r(E_r(S_r))$. The attention modules are designed to learn

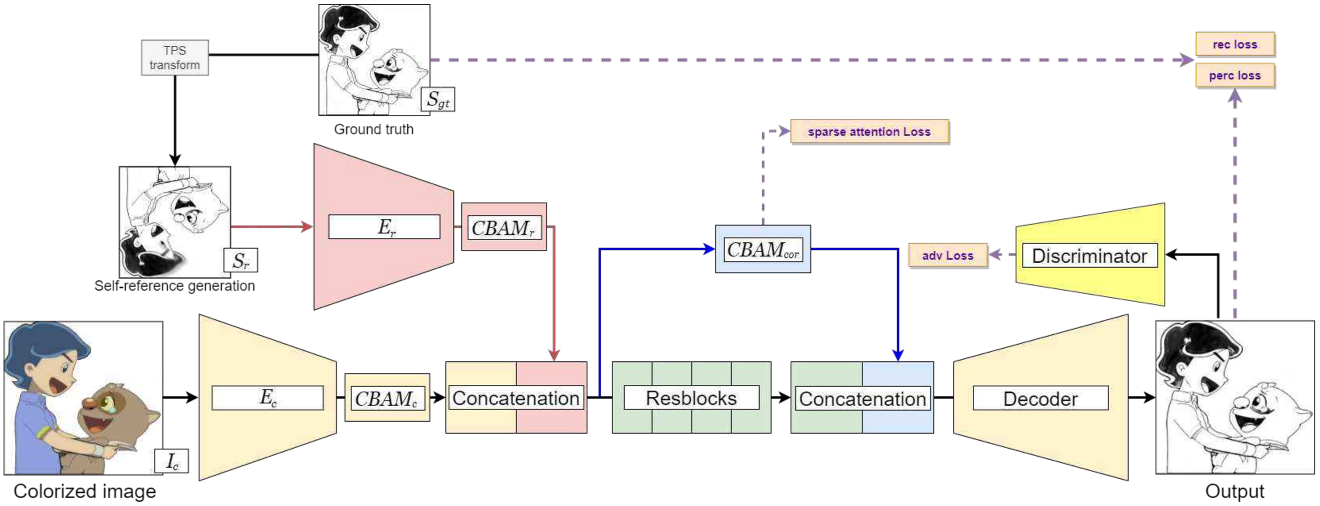


Fig. 3. Overview of our sketch style imitation. We apply the thin plate splines (TPS) transformation to a ground truth sketch for the self-reference generation. We use three independent Convolutional Block Attention Modules (CBAM) to detect edges of the colorized image and reference sketch as well as the visual correspondences between them. © SeoulCityBrand.

spatially important features of the colorized image and reference sketch style such as their edges.

(C) The encoded features are then concatenated and in parallel, passed through several residual blocks as well as another CBAM attention module $CBAM_{cor}$. Here, the attention module learns to encode spatial correspondences between the colorized image and reference sketch style features (Section 3.2).

(D) Finally, the output of the residual blocks and the attention module are concatenated and fed into a generator to extract a sketch similar to the reference sketch style S_r. The discriminator, as an adversary of the generator, has an objective to distinguish the generated sketch images from the real ones.

3.2 Self Reference Sketch Style Generation

Inspired by Yu et al. [2017] and Lee et al. [2020], we use augmentations to increase the volume and diversity of sketches for training. To generate reference sketch styles S_r in the training phase, we randomly apply one of two augmentations to ground truth sketches S_{gt}: 1) a random flip, or 2) a TPS transformation. While the generated reference sketches have a style similar to the ground truth sketches (e.g., thickness of lines), they have different spatial layouts (e.g., the position of eyes). These spatial variations in the generated reference sketches help our network to deal with unseen reference sketches that might have a similar line style but different position layouts or character body shapes. See the supplementary material for more details about the augmentations. When we did not apply any transformations during the training phase, the reference sketch became identical to the ground truth sketch. Therefore, the trained model lazily learned to generate output identical to that of the reference sketch without learning any content from the colorized image.

3.3 Attention Mechanism

Our model extracts a sketch from a colorized image in such a way that the extracted sketch has a line style similar to that of a given reference sketch while representing the same visual content as the colorized image. To this end, our model should have mechanisms to learn three essential features: 1) the content of the colorized image, 2) the style of the reference sketch, and 3) the visual correspondences between the colorized image and reference sketch. Inspired by the use of attention modules in example-guided image-to-image translation models [Iizuka and Simo-Serra 2019; Lee et al. 2020], we use the attention mechanism to learn these features. We chose the CBAM [Woo et al. 2018] attention module because it is computationally more efficient and requires less learnable parameters in comparison with others [Iizuka and Simo-Serra 2019; Lee et al. 2020]. See the supplementary material for details.

To extract the content from the colorized image and style from the reference sketch, we apply an independent CBAM attention module to each of the last convolutional layers of the encoded colorized image $CBAM_c(E_c(I_c))$ and reference sketch style $CBAM_r(E_r(S_r))$. These two CBAM attention modules independently learn to attend more to edges drawn in the colorized image and reference sketch. We sequentially apply the channel and spatial attention modules of CBAM so that our network adaptively learns which spatial information to emphasize or suppress on the given colorized image and reference sketch.

To learn visual correspondences between the colorized image and reference sketch, we apply another $CBAM_{cor}$ attention module to the concatenated features of the colorized image and reference sketch encoders. This attention module learns to attend more to corresponding edges between the colorized image and reference sketch. One example would be eyes drawn in both of the colorized image and reference sketch but depicted in various positions and line styles. The following equation summarizes the CBAM attention modules used in our network design:

$$\begin{aligned} x_c &= CBAM_c(E_c(I_c)), \\ x_r &= CBAM_r(E_r(S_r)), \\ x_{cor} &= CBAM_{cor}([x_c; x_r]), \end{aligned} \tag{1}$$

where the $x_{(.)}$ symbols are encoded features in which subscripts c, r, and cor denote the colorized image, the reference sketch style, and the visual correspondences between them, respectively. In addition, each $CBAM_{(.)}$ attention module sequentially computes and applies channel $M^{ch}_{(.)}$ and spatial $M^{sp}_{(.)}$ attention maps, as follows:

$$\begin{aligned} CBAM_c &= M^{sp}_c \otimes M^{ch}_c \otimes E_c(I_c), \\ CBAM_r &= M^{sp}_r \otimes M^{ch}_r \otimes E_r(S_r), \\ CBAM_{cor} &= M^{sp}_{cor} \otimes M^{ch}_{cor} \otimes [x_c; x_r], \end{aligned} \tag{2}$$

where ⊗ indicates element-wise multiplication.

Our ablation study described in Section 5.3 confirms both qualitatively and quantitatively that without using any attention mechanism, our sketch-extraction performance will not be satisfactory. Moreover, as shown in Table 1, the ablation study also quantitatively confirms that our sketch extraction model using three independent CBAM attention modules outperforms the same model trained with only one CBAM attention module employed after feature concatenation. In addition, the visualization of attention maps presented in Section 5.4 and in the supplementary material suggest that two CBAM attention modules used after E_r and E_c attend more to the edges of the colorized image and reference sketch, while the one employed after feature concatenation attends more to visual correspondences between the colorized image and reference sketch.

3.4 Objective Functions

Sparse Attention Loss: Sketches usually have more white space than colorized images, and hence lines making a sketch are depicted sparsely in the image space. Therefore, we also apply the L1 regularization loss to the spatial attention map of our $CBAM_{cor}$ attention module to encourage the elements of x_{cor} to be spatially sparse, as follows:

$$\mathcal{L}_{sparse} = ||M^{sp}_{cor}||_1. \tag{3}$$

This sparse attention loss encourages our model to attend only to the most important spatial areas of the concatenated colorized image and reference sketch feature maps. In addition, we show that this objective function helps our generator to extract sketches with sharp lines and without unnatural artifacts such as faded lines. See our ablation study described in Section 5.3 and Figure 4 for more details.

Reconstruction Loss: Given a colorized image and a reference sketch, generator G is trained to produce a sketch that is similar to the ground truth in terms of a pixel-wise L1 loss function. We use the L1 loss instead of L2 because it encourages less blurring [Isola et al. 2017]. This reconstruction loss penalizes the network for the pixel-wise difference between the generated sketch $G(I_c, S_r)$ and ground truth sketch S_{gt}, as follows:

$$\mathcal{L}_{rec} = \mathbb{E}_{S_{gt},I_c,S_r}[||G(I_c, S_r) - S_{gt}||_1]. \tag{4}$$

Adversarial Loss: The discriminator D, as an adversary of the generator, has the objective of distinguishing the generated sketches from the real ones. The output of real/fake discriminator D computes the probability that an arbitrary sketch is a real one. On the other hand, the generator G attempts to deceive the discriminator D by creating an output sketch $G(I_c, S_r)$ that looks similar to a ground truth sketch S_{gt} via a conditional adversarial loss function. The loss for optimizing the discriminator D is formulated as the standard crossentropy loss, as follows:

$$\begin{aligned} \mathcal{L}_{adv} = &\ \mathbb{E}_{S_{gt},I_c}\left[\log\left(D(S_{gt}, I_c)\right)\right] \\ &+ \mathbb{E}_{I_c,S_r}\left[\log\left(1 - D\left(G(I_c, S_r), I_c\right)\right)\right]. \end{aligned} \tag{5}$$

Perceptual Loss: Training with the perceptual loss allows the model to better reconstruct fine details and edges [Johnson et al. 2016b]. Because the perceptual loss depends on high-level features from a pretrained network and measures image similarities more robustly than per-pixel losses, it is less sensitive to pixel-wise shifts. The use of the perceptual loss also encourages a network to produce an output that is more perceptually plausible as evidenced by sketch simplification methods [Xu et al. 2021]. Therefore, we employ a form of perceptual loss that penalizes the differences in intermediate activation maps between the generated sketch $G(I_c, S_r)$ and ground truth sketch S_{gt} from the ImageNet [36] pretrained network. We use activation maps from both high-level and low-level layers of the pretrained network to penalize the corresponding high-level semantic and low-level style differences between the generated and ground truth sketch, as follows:

$$\mathcal{L}_{perc} = \mathbb{E}_{I_c,S_r,S_{gt}}[\sum_l ||\phi_l(G(I_c, S_r)) - \phi_l(S_{gt})||_1], \tag{6}$$

where ϕ_l denotes the activation map from the l^{th} layer of a VGG16 network, pretrained on ImageNet.

In summary, the overall loss function for the generator G and discriminator D is defined as follows:

$$\begin{aligned} \min_G \max_D \mathcal{L}_{total} = &\ \lambda_{sparse}\mathcal{L}_{sparse} + \lambda_{perc}\mathcal{L}_{perc} \\ &+ \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}, \end{aligned} \tag{7}$$

where the $\lambda_{(.)}$ symbols are the penalty weights that define to which degree we want to enforce each loss term. We empirically derived the penalty weights used in our experiments based on the quality of extracted sketches when imitating a reference sketch style. We set $\lambda_{sparse} = 0.1$, $\lambda_{perc} = 1$, $\lambda_{rec} = 10$, and $\lambda_{adv} = 1$. λ_{sparse} helps generate a sparse sketch and eliminate inaccurate dense edges from the colorized image. The supplementary material shows that if we increase λ_{sparse}, the sparsity of the extracted sketch will increase.

3.5 Implementation Details

The sizes of the input colorized images and reference sketches are fixed at 256x256 for every dataset. We use the Adam optimizer [Kingma and Ba 2014] with $\beta 1 = 0.5$, and $\beta 2 = 0.999$. The learning rates for both the generator and discriminator are initially set to 0.0002. The total number of epochs was 1500, and the learning rate decayed slightly after 750 epochs. We applied augmentations during training. The detailed network architectures of our discriminator and generator are described in the supplementary material.

4 DATASET

Twitter Dataset: We collected over 2000 pairs of colorized images and their corresponding authentic sketches (i.e., manually drawn by line artists) from Twitter using a tag-based search. We manually eliminated mis-pairs and low-quality image pairs. After filtering, the final dataset contained 1300 pairs of sketches and colorized images, partitioned into 1000 pairs for training and 300 pairs for validation of our sketch-extraction model. In this dataset, artists were anonymous and drew sketches with various sketch styles. According to Twitter's regulation [twitter policy 2021], one can legally use this dataset for research purposes. This dataset was used to train the model and for the experiments described in Section 6 (*Evaluation 2*), Section 5.4, Section 5.5, and Section 6.1. Examples of our Twitter dataset are shown in the supplementary material.

Four Artists Dataset: We additionally gathered paired sketches and their corresponding colorized images, drawn and colorized by four different artists using illustration software. In this dataset, the identities of the artists are known, and each artist has his/her own unique sketch style. For each individual artist, the dataset contains up to 20 pairs of sketches and their corresponding colorized images. Because the identities of the artists are known, it is possible to group the sketches based on the artist's identities and explore the differences between their sketch styles. For example, while artist B uses more details to depict his/her sketches, artist A's sketches are more abstract and sparse. Artist C paints eyelashes with black colors while artist D draws just the borderlines of eyelashes. The four artist dataset can be used for research purposes upon an agreement with the original artists. This dataset was only used as a test set and for conducting the experiments described in Section 6 (*Evaluation 1*), Section 5.2, Section 5.3, and Section 6.1. Examples of the Four Artists Dataset are shown in the supplementary material.

5 EXPERIMENTS

In this section, first we introduce the quantitative evaluation metrics used in our experiments (Section 5.1), after which we compare our method with several baselines (Section 5.2) both qualitatively and quantitatively, and ablate the loss functions as well as attention modules to analyze their effects (Section 5.3). Finally, we visualize the attention maps of our network design (Section 5.4) and introduce our cyclic evaluation metric to measure the quality of our sketch style transfer method (Section 5.5). In the supplementary material, we show that our method can deal with imitating different poses, e.g., the face is in a very different location between the colorized image and the reference sketch.

5.1 Evaluation Metrics

For a quantitative analysis, we utilized the widely used Peak Signal-to-Noise Ratio (PSNR) [Wang et al. 2004], the Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al. 2018a], and the Frèchet Inception Distance (FID) [Grigorescu et al. 2003] metrics that assess the pixel-wise difference, the perceptual distance based on neural network features, and the Frèchet distance between two data collections, respectively. We consider LPIPS as our main evaluation metric to measure the perceptual similarity because it learns representations of images that correlate well with perceptual judgments [Zhang et al. 2018a]. We used both low-level and high-level features for computing LPIPS. Moreover, in *Evaluation 1* of our user-study, people preferred sketches extracted by our method, and the average LPIPS score of these sketches outperforms others similar to the human preference in sketch selections. PSNR focuses on pixel-wise differences that fail to account for many nuances of human perception [Wang et al. 2004], and FID requires a large number of test sets (e.g., 50k) to compare the statistics of generated samples to those of real samples accurately [Borji 2021].

Proposed cyclic evaluation metric: We also propose a novel evaluation metric to measure how faithfully our model preserves the style of a reference sketch in multiple consecutive sketch-extraction steps. The key idea behind this is that if our model extracts a sketch using a *reference style*, one should also be able to produce an identical *reference style* using the extracted sketch from the previous step. We provide a detailed explanation of the proposed evaluation metric as well as its corresponding results in Section 5.5.

5.2 Comparison with Baselines

All the baseline models reported in Table 1 were trained using our Twitter dataset and tested on our Four Artist Dataset when official training codes or details were available. We used the original code and augmentations of each baseline. For training, we used 1000 pairs from the Twitter training set. We only used pre-trained models of SketchKeras and Anime2Sketch. In the supplementary material, we provide more details about the relation between our method and these baselines regarding the technique used to process a given colorized image and incorporated loss terms.

We compared our sketch-extraction method against four sketch-extraction approaches: 1) Canny [Canny 1986], 2) XDoG [Winnemöller 2011], 3) SketchKeras [lllyasviel/sketchKeras 2018], and 4) Anime2Sketch [Xiaoyu Xiang 2021]. To verify the effectiveness of our method, we conducted both qualitative and quantitative comparisons on four different datasets. Each dataset contains sketches drawn and colorized by an artist based on his/her own artistic style.

Figure 4 shows the overall qualitative results of our sketch-extraction model and the four baselines. The supplementary material contains more examples. The Canny method extracts sketches with noisy dotted lines while the XDoG and Anime2Sketch approaches mis-paint some parts of the character's cloth, head, or eyelashes in black. Line styles extracted by SketchKeras are similar to pencil strokes, which are thicker than the lines of the ground truth sketch drawn by the artist. In contrast, the sketch extracted using our algorithm more closely resembles the ground truth drawn manually by the artist. For example, our method draws only the borderlines of eyelashes and pupils, similar to the ground truth, and avoids mis-painting the character's body. This suggests that our method is superior at establishing visual correspondences between the extracted sketch and the reference style. In addition, the thickness of the lines extracted by our method varies at different parts of the character's body, making the result more similar to the ground truth.

We report in Table 1 the LPIPS, PSNR, and FID scores calculated over the four different datasets. On average, our sketch-extraction method outperforms the existing baselines on all reported scores, demonstrating that our method is robustly capable of producing sketches similar to the original style of an artist, on the four different datasets examined here.

We also compared our model both qualitatively and quantitatively against out-of-the-box image-to-image translation models, in this case pix2pix [Isola et al. 2017], BicycleGAN [Zhu et al. 2017b], and CUT [Park et al. 2020] as well as the example-guided style

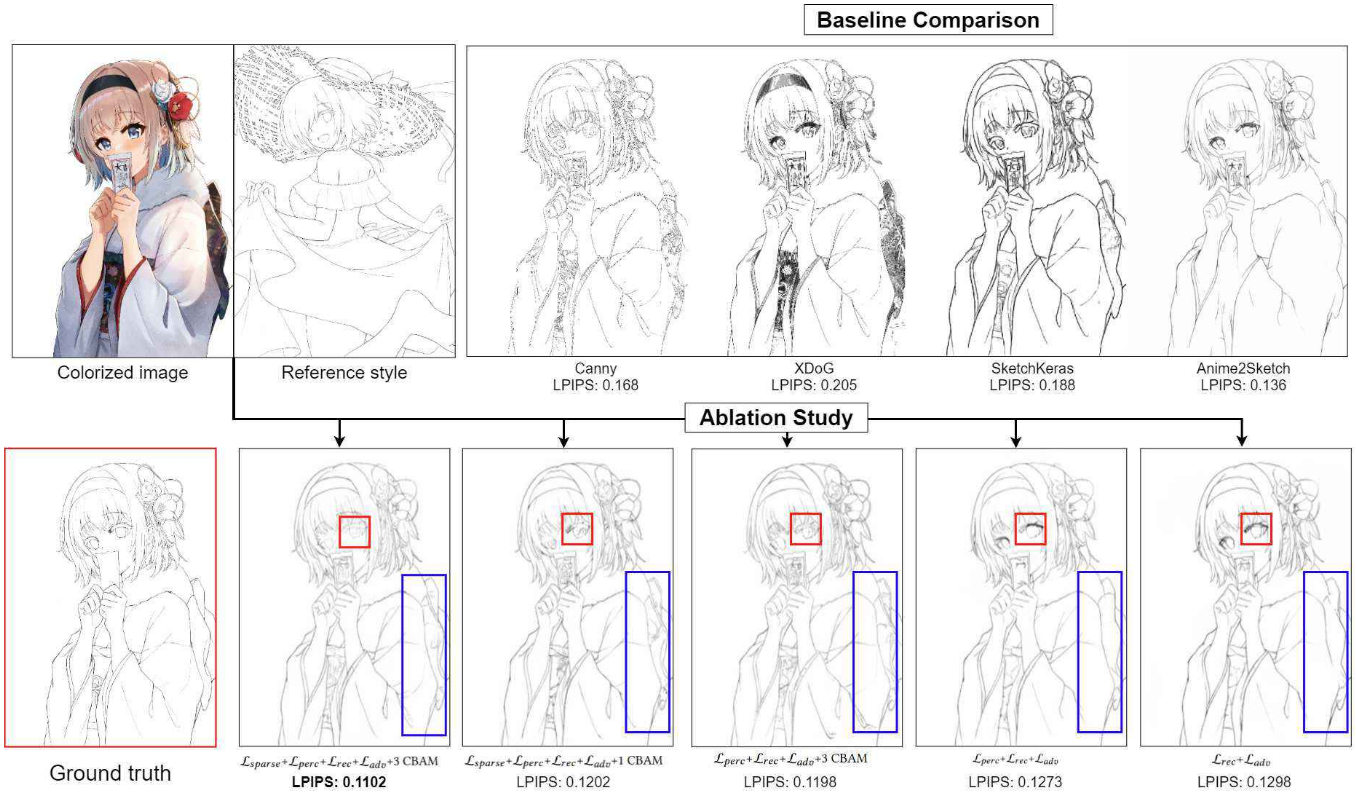

Baseline Comparison
Colorized image
Reference style
Canny
LPIPS: 0.168
XDoG
LPIPS: 0.205
SketchKeras
LPIPS: 0.188
Anime2Sketch
LPIPS: 0.136
Ablation Study
Ground truth
$\mathcal{L}_{sparse}+\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}$+3 CBAM
LPIPS: 0.1102
$\mathcal{L}_{sparse}+\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}$+1 CBAM
LPIPS: 0.1202
$\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}$+3 CBAM
LPIPS: 0.1198
$\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}$
LPIPS: 0.1273
$\mathcal{L}_{rec}+\mathcal{L}_{adv}$
LPIPS: 0.1298

Fig. 4. Qualitative results of comparison with baselines and ablation study. Our method draws only the borderlines of eyelashes, similar to the ground truth. Without sparse loss, the output sketch contains inaccurate dense lines. Without our attention mechanism, the character's eyelashes are mis-painted in black. Without the perceptual loss, the output sketch contains incomplete edges. See the supplementary material for more examples. © Ayul_oekaki.

Table 1. Quantitative results of comparison with baselines and ablation study. The best LPIPS score is annotated in bold while other best scores are underlined.

	Dataset A	Dataset B	Dataset C	Dataset D	AVERAGE
Methods	**LPIPS**↓ / PSNR↑ / FID↓				
$\mathcal{L}_{sparse}+\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}+3\ CBAM$	**0.1340**/34.71/66.76	**0.1350**/34.85/100.95	0.0986/34.54/63.16	**0.1950**/32.84/81.80	**0.1406**/34.23/78.16
$\mathcal{L}_{sparse}+\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}+1\ CBAM_{cor}$	0.1407/34.66/73.36	0.1429/34.53/109.01	**0.0974**/34.25/69.99	0.2191/32.37/82.29	0.1500/33.95/83.66
$\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}+3\ CBAM$	0.1350/34.65/69.10	0.1408/34.80/116.89	0.1027/34.42/62.27	0.2043/32.74/82.88	0.1457/34.15/82.78
$\mathcal{L}_{perc}+\mathcal{L}_{rec}+\mathcal{L}_{adv}$	0.1361/34.59/72.50	0.1385/34.71/105.88	0.1059/34.35/69.68	0.2117/32.78/90.92	0.1480/34.10/84.74
$\mathcal{L}_{rec}+\mathcal{L}_{adv}$	0.1381/34.53/77.41	0.1399/34.72/105.17	0.1051/34.31/63.03	0.2164/32.64/95.01	0.1498/34.05/85.15
Canny [Canny 1986]	0.1682/33.94/103.55	0.1569/34.77/116.58	0.1207/34.54/124.17	0.2253/32.55/94.62	0.1677/33.95/109.73
XDoG [Winnemöller 2011]	0.2311/33.98/146.61	0.1718/34.91/175.28	0.1008/34.50/104.69	0.2174/32.55/125.63	0.1802/33.80/138.05
SketchKeras [lllyasviel/sketchKeras 2018]	0.2112/34.99/97.95	0.1667/34.55/131.30	0.1093/34.26/81.19	0.2063/32.35/92.78	0.1733/34.03/100.80
Anime2Sketch [Xiaoyu Xiang 2021]	0.1633/34.51/95.28	0.1572/35.05/131.60	0.1171/34.02/71.08	0.2328/32.58/80.96	0.1676/34.04/94.73
Pix2pix [Isola et al. 2017]	0.1532/34.81/113.71	0.1733/34.35/119.90	0.1115/34.04/100.40	0.2154/32.35/114.52	0.1633/33.88/112.13
BicycleGAN [Zhu et al. 2017b]	0.1756/34.62/114.52	0.1964/34.71/131.59	0.1257/34.32/83.79	0.2375/32.38/122.48	0.1838/34.00/113.09
CUT [Park et al. 2020]	0.2614/34.19/111.74	0.2460/34.29/181.31	0.2817/34.24/176.99	0.3335/32.15/100.76	0.2806/33.71/142.70
Gatys et al. [2016]	0.3128/33.26/140.52	0.3321/33.20/151.14	0.3022/33.42/122.87	0.3487/31.12/166.87	0.3239/32.75/145.35
MUNIT [Huang et al. 2018]	0.4112/31.98/189.45	0.4250/31.81/201.11	0.4017/31.04/210.99	0.4042/31.04/198.54	0.4105/31.46/200.02
Lee et al. [2020]	0.1638/34.27/107.58	0.1774/34.16/120.34	0.1289/33.55/111.46	0.2357/32.14/113.06	0.1764/33.53/113.11
Simo-Serra et al. [2016a]	0.2143/31.16/146.86	0.2279/30.54/143.59	0.1637/31.46/123.80	0.2363/29.92/99.52	0.2105/30.77/128.44

transfer models MUNIT [Huang et al. 2018], Gatys et al. [2016], and Lee et al. [2020]. We used the BicycleGAN and CUT models for supervised and unsupervised learning settings, respectively, as these models show the best performance on the UT-Zap50K dataset [Yu and Grauman 2014] on the edge2shoes task [Pang et al. 2021] compared to other supervised and unsupervised image-to-image translation methods [Cho et al. 2018; Huang et al. 2018; Isola et al. 2017; Kim et al. 2020; Zhu et al. 2017a]. The model proposed by

Gatys et al. is considered as the standard in style transfer [Jing et al. 2020], and MUNIT also shows better performance on the UT-Zap50K dataset on the edge2shoes task [Pang et al. 2021] compared to the alternatives [Cho et al. 2018]. As shown in Figure 2, the sketches extracted via these methods are not visually similar to authentic sketches. In addition, the quantitative results shown in Table 1 confirm that our method outperforms all of these methods. In addition, our attention mechanism based on CBAM outperforms the attention mechanism of Lee et al. [2020]. We also compared our approach with a sketch-simplification method [Simo-Serra et al. 2016a] and showed the superiority of our method. See the supplementary material for more qualitative comparisons with the baselines.

5.3 Ablation Study

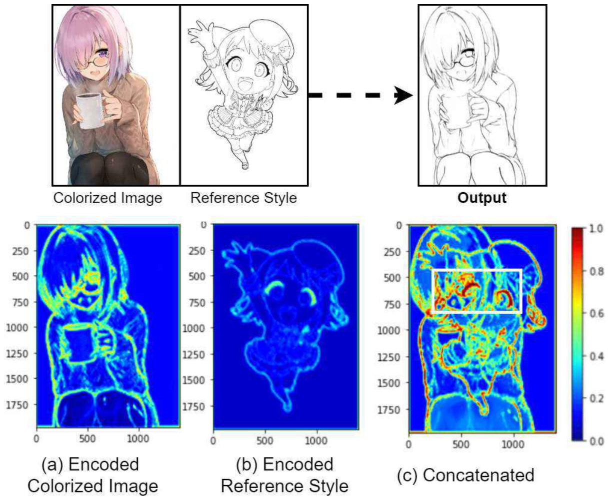
We ablate the attention module and loss functions individually to analyze their effects qualitatively, as shown in Figure 4, and quantitatively, as shown in Table 1. When we remove the sparse loss, the output sketch contains inaccurate dense lines emerging in the eye and shoulder areas that are not visually similar to the original sketch, which is depicted with more sparse lines. Without our attention mechanism and with only one $CBAM_{cor}$ attention module, the character's eyelashes are mis-painted in black, which does not comply with the reference sketch style depicting only the borderlines of the eyelashes. Without the perceptual loss, the output sketch contains incomplete edges on the character's cloth and shoulders. The supplementary material contains more qualitative examples of the ablation study.

Quantitatively, on average, our method using all losses and the attention module performs best out of all others. Table 1 shows that removing each of the losses (perceptual or sparse loss) or the spatial attention module (rows 2 and 4) will adversely affect the sketch-extraction performance. In our ablation study, adding the sparse loss leads to the greatest improvement in the LPIPS, PSNR, and FID scores, followed by adding the attention modules and then the perceptual loss.

We conducted the ablation study on the size of our training set. In the supplementary material, we showed that if we train our model with 30% of our dataset (i.e., 300 pairs), our method can extract sketches similar to reference styles with fine quality, thanks to the TPS augmentation.

5.4 Visualization of Attention Maps

As shown in Figure 3, we leveraged three CBAM attention modules in our network design. Figure 5 shows an example of spatial attention maps $M^{sp}_{(.)}$ learned by these CBAM attention modules, which are located in our network design after the colorized image encoder (a), reference style encoder (b), and concatenated features of the colorized image and reference style encoders (c). First, we observed that the spatial attention modules of the encoded colorized image (a) and reference style (b) learn to attend more to the edges of the colorized image and reference style, respectively. In addition, the spatial attention module applied to the concatenated features (c) learns to attend more to visual correspondences between the colorized image and the reference style (e.g., eyelashes). The ability of our network to find visual correspondences between the colorized image and reference style leads to the extraction of a sketch from the colorized image with thick eyelash lines, similar to the reference style. The supplementary material contains more examples of our visualization of attention maps.



(a) Encoded Colorized Image (b) Encoded Reference Style (c) Concatenated

Fig. 5. Visualization of our attention. Detecting edges of the colorized image (a), reference sketch (b), and visual correspondences (c). © Ayul_oekaki.

5.5 Cyclic Evaluation Metric

Our goal is quantitatively to assess how our model preserves the reference sketch style in multiple consecutive sketch extractions. Our assumption is that if we use a *reference style* to extract a sketch from a colorized image, the extracted sketch should have a style similar to that of the *reference style*. Therefore, one should be able to use the extracted sketch as a reference to generate precisely the same *reference style* used during the first sketch-extraction step. To this end, first we randomly select a pair consisting of a colorized image I_{c_1} and its corresponding ground truth sketch S_{gt_1} from our dataset. Second, we use the ground truth sketch S_{gt_1} as a reference to extract a sketch from another random colorized image I_{c_2}. The extracted sketch $G(S_{gt_1}, I_{c_2})$ should have a style similar to S_{gt_1}. Third, we use the extracted sketch $G(S_{gt_1}, I_{c_2})$ as a reference to extract a sketch from the colorized image I_{c_1}, which results in generating a sketch $G(G(S_{gt_1}, I_{c_2}), I_{c_1})$ with the same content and style as S_{gt_1}. Finally, we compare $G(G(S_{gt_1}, I_{c_2}), I_{c_1})$ and S_{gt_1} using LPIPS, FID, and PSNR metrics. A representative example of our cyclic evaluation process is shown in Figure 6 and is summarized as follows:

$$\begin{aligned} S_{gt_1}, I_{c_2} &\xrightarrow{G} G(S_{gt_1}, I_{c_2}), \\ G(S_{gt_1}, I_{c_2}), I_{c_1} &\xrightarrow{G} G(G(S_{gt_1}, I_{c_2}), I_{c_1}), \\ \text{Cyclic LPIPS} &= \text{LPIPS}(G(G(S_{gt_1}, I_{c_2}), I_{c_1}), S_{gt_1}). \end{aligned} \tag{8}$$

We report in Table 2 the cyclic LPIPS, PSNR, and FID scores calculated using our model when imitating a reference sketch style as well as two example-guided style transfer models: Gatys et al. [2016], Lee et al. [2020], and Munit [Huang et al. 2018]. In all reported scores, our model outperforms the general-purpose example-guided style transfer models in imitating and preserving a reference sketch style in multiple consecutive sketch extractions.

Table 2. Quantitative results of our cyclic and dissimilarity evaluations.

Cyclic Evaluation	**LPIPS**↓/PSNR↑/FID↓
Ours	**0.1931**/33.97/114.41
Lee et al. [2020]	0.2455/31.22/182.04
Gatys et al. [2016]	0.4119/27.79/253.41
MUNIT [Huang et al. 2018]	0.4138/27.50/244.81
Dissimilarity Evaluation (ours)	0.2495/33.30/153.02

Because general-purpose example-guided style transfer models are not specifically designed for the sketch style transfers, unlike our method, we conducted another experiment to justify the computed scores of our cyclic metrics. Specifically, we use a similar process to generate sketches with content identical to that used to compute the cyclic metric but with different sketch styles. To this end, the experiment proceeded as follows: First, given the same pair of the colorized image I_{c_1} and its corresponding ground truth sketch S_{gt_1} from our dataset, we extracted a sketch from the colorized image I_{c_1} using another random reference style S_{gt_2}, i.e., $G(S_{gt_2}, I_{c_1})$. Second, we used the extracted sketch $G(S_{gt_2}, I_{c_1})$ as a reference to extract a sketch from the colorized image I_{c_1}, which results in the generation of a sketch $G(G(S_{gt_2}, I_{c_1}), I_{c_1})$ with the content identical to and a style different from those of S_{gt_1}. Finally, we compared $G(G(S_{gt_2}, I_{c_1}), I_{c_1})$ and S_{gt_1} using the LPIPS, FID, and PSNR scores. Because the sketches generated using this evaluation process have different sketch styles while representing the same content, we refer to this evaluation process as the *dissimilarity metric*. We applied our sketch-extraction model twice to the colorized image I_{c_1} because we wanted to ensure a fair comparison with our cyclic metric, which also uses two sketch-extraction steps. A representative example of our dissimilarity evaluation is shown in Figure 6 and is summarized below.

$$\begin{aligned} S_{gt_2}, I_{c_1} &\xrightarrow{G} G(S_{gt_2}, I_{c_1}), \\ G(S_{gt_2}, I_{c_1}), I_{c_1} &\xrightarrow{G} G(G(S_{gt_2}, I_{c_1}), I_{c_1}), \\ \text{Dissimilarity LPIPS} &= \text{LPIPS}(G(G(S_{gt_2}, I_{c_1}), I_{c_1}), S_{gt_1}). \end{aligned} \tag{9}$$

As shown in Table 2, the cyclic LPIPS, PSNR, and FID scores outperform the corresponding dissimilarity scores while both cyclic and dissimilary metrics are computed over the generated sketches with content identical to that in the ground truth sketches. In the cyclic evaluation, the style of the generated sketches are preserved to be similar to the ground truth sketches. However, in the dissimilarity evaluation, the generated sketches have slightly different sketch styles but the exactly same content as the ground truth. This quantitative result suggests that our sketch-extraction model reflects small changes of a reference style in the extracted sketches.

6 PERCEPTUAL STUDY

Several previous studies have leveraged anonymous human evaluators to validate the quality of sketches drawn by a network model [Chen et al. 2017; Liu et al. 2019b]. To qualitatively assess our sketch-extraction algorithm, we conducted two evaluations, as described below.

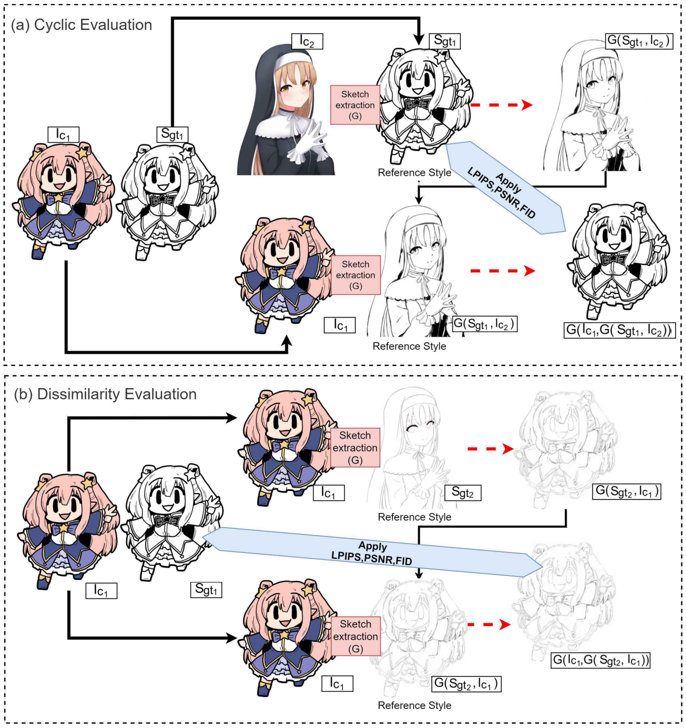


Fig. 6. Proposed cyclic and dissimilarity evaluation metrics. We measure how faithfully our model preserves the style of a reference sketch in multiple consecutive sketch-extraction steps. © AonoriwaKame (I_{c_1}), Tk_painter (I_{c_2}).

Evaluation 1 [Ours vs. Baselines]: We compared sketches drawn by our algorithm with five different sketch-extraction baselines: 1) Canny, 2) XDoG, 3) SketchKeras, 4) Anime2Sketch, and 5) BicycleGAN. To this end, we asked participants to choose a sketch that best resembles an artist's original sketch style (i.e., ground truth) among six sketches extracted by the five baselines as well as our algorithm that imitates the artist's original sketch style. See Figure 7 for an example of this comparison and the supplementary material for more comparisons conducted in this study.

Evaluation 2 [Style imitation performance]: The goal is to check whether our sketch-extraction method can imitate various sketch styles. To this end, given a colorized image, we compared four sketches drawn by our algorithm, with each extracted to imitate the specific sketch style represented in a reference style. Given one of the four reference styles as the ground truth, we asked participants to choose the sketch that best resembled the ground truth reference style among four extracted sketches: one imitating the same style as the ground truth and three imitating other styles. These styles were randomly selected. Figure 7 presents an example of this comparison. The supplementary material contains more comparisons used in this study.

Evaluation 1 and 2 setup: For a fair visual comparison of the sketches, we used the same image size for all of the extracted sketches. For each comparison, we placed extracted sketches in a side-by-side configuration, each to a randomly assigned slot in

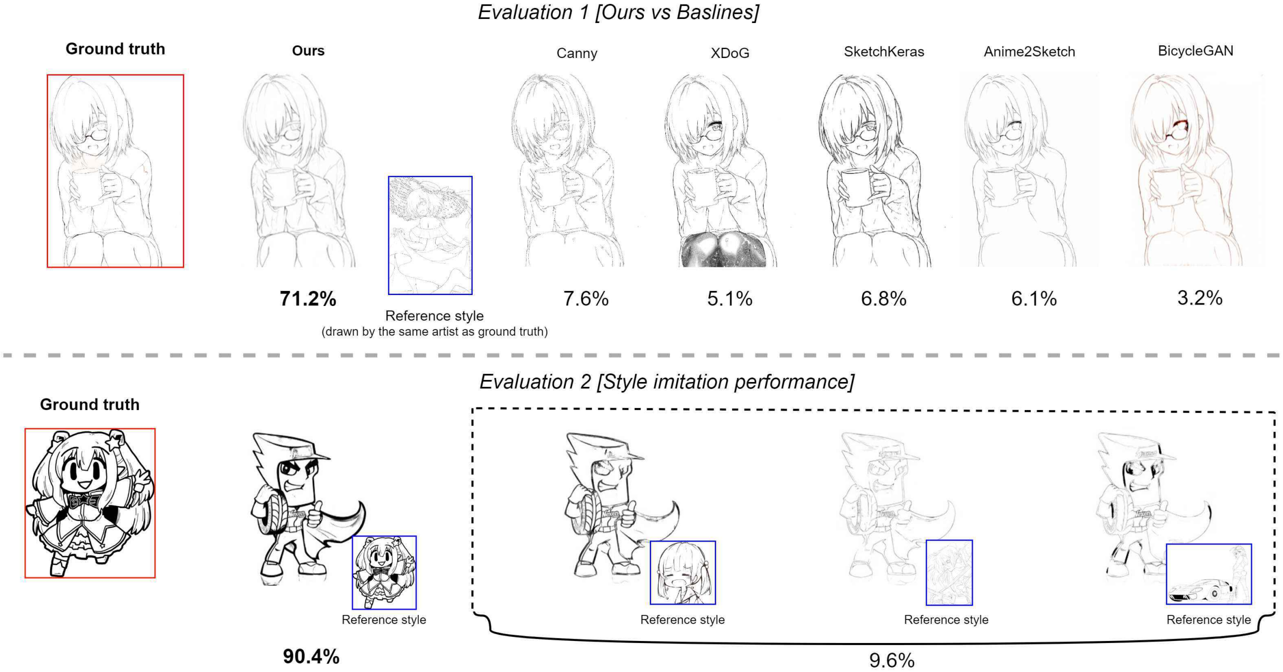


Fig. 7. A representative example of our perceptual study and its results. Note that we did not show the reference styles, presented here in the blue boxes, to the participants. We asked the participants to choose the sketch that best resembles the ground truth. In *Evaluation 1*, we compared our sketch-extraction method against five baselines. In *Evaluation 2*, we checked whether our sketch-extraction method can imitate various reference styles. © *Evaluation 1*: Ayul_oekaki; *Evaluation 2*: AonoriwaKame, KiwoomHeroes, Comete_atri, Chobi_chu.

a group of extracted sketches. Each participant made twenty comparisons for each evaluation. The sketches in each comparison set were extracted from the same colorized image. The participants were asked: “Which sketch represents the ground truth sketch style the best?”. They had to select one from six and four sketches in each comparison set for *Evaluation 1* and *Evaluation 2*, respectively. There were 100 participants for each evaluation (200 participants in total). The results of our evaluations should not depend on the specifics of any demographic distribution and we did not therefore focus on any specific groups of people. We used the four artists dataset in *Evaluation 1* because we required an example of each artist’s sketch to imitate his/her style. In *Evaluation 2*, we used the Twitter dataset because the participants compared the style similarity between the extracted sketch and the reference sketch. These sketches were selected from the Twitter test-set (300 pairs), and styles were randomly selected.

6.1 Results of the Perceptual study

Evaluation 1 presented in Figure 7 shows that 71.2% of the participants selected sketches extracted by our algorithm, which focuses on imitating the artist’s style over the five baseline methods. This suggests that people can easily distinguish visual differences between various sketch styles. It also confirms that the state-of-the-art methods cannot imitate a specific sketch style due to the fact that they extract the same style for all given colorized images.

For *Evaluation 2*, presented in Figure 7, 90.4% of the participants selected the intended sketch targets corresponding to the ground truth. This shows that our method can imitate various sketch styles that are visually similar to given ground truth sketch styles. Regarding the 9.6% who selected different sketch targets, these likely stemmed from certain challenging comparisons in which the *ground truth sketch style* was indeed visually similar to *other sketch styles*, making it difficult for the participants to distinguish the small differences between them. All of the comparisons used in *Evaluation 2* are presented in the supplementary material.

7 MORE APPLICATIONS

7.1 Improving automatic sketch colorization

As mentioned in Section 1, training auto-colorization models suffers from information scarcity of authentic sketches (i.e., drawn by an artist) and their corresponding colorized images. Therefore, existing auto-colorization methods use sketch-extraction techniques to produce synthetic sketches from colorized images for training their models. In this experiment, we used our sketch extraction technique to improve the performance of auto-colorization models by producing more realistic synthetic sketches in comparison with the synthetic sketches generated by four baselines: Canny, XDoG, SketchKeras, and Anime2Sketch. We evaluate the impact of sketch-extraction techniques on the quality of auto-colorization models, both qualitatively, as shown in Figure 8, and quantitatively, as shown in Table 3. In this evaluation, we used synthetic sketches for training and authentic sketches for testing of the auto-colorization models. See the supplementary material for more details. Both qualitative and quantitative results suggest the following.

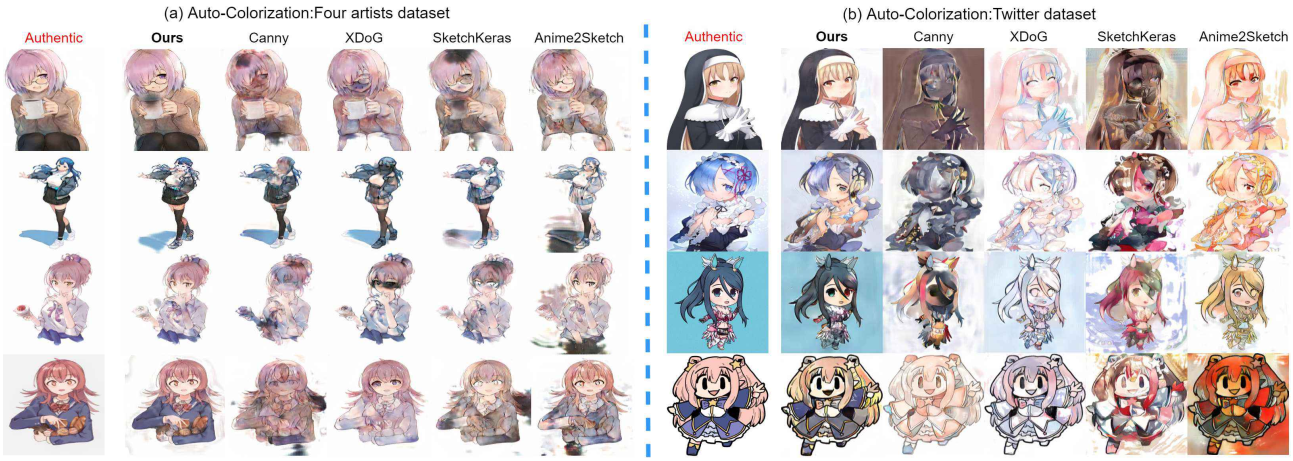


Fig. 8. Our sketch-extraction method improves the quality of auto-colorization models by improving the quality of synthetic sketches used for training them. We improve the synthetic sketches by imitating an artist's sketch style (a), or by generating sketches with various styles similar to authentic ones (b). © up to down (a): Ayul_oekaki, Okera_sz, Comete_atr (rows 3 & 4); (b): Tk_painter, Maromayu, BioTroy, AonoriwaKame.

Table 3. Quantitative results of improving auto-colorization models using our sketch-extraction technique. The best LPIPS scores are bolded while other best scores are underlined.

	Four artist dataset	Twitter dataset
Methods	**LPIPS**↓/PSNR↑/FID↓	
Ours	**0.1895**/31.81/122.50	**0.4360**/29.03/148.59
Canny	0.2629/30.94/224.43	0.5834/28.24/214.43
XDoG	0.2477/30.99/231.20	0.5223/28.40/213.10
SketchKeras	0.2501/30.91/198.25	0.5278/28.50/207.40
Anime2Sketch	0.2764/30.46/219.90	0.5294/28.40/211.34

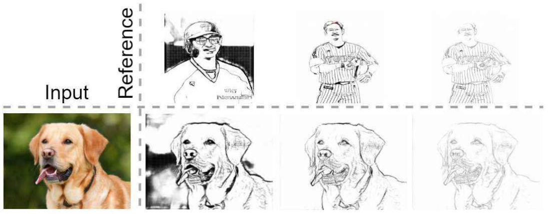


Fig. 9. Our method can be applied to non-anime characters, such as actual photos, if a proper dataset is provided.

(1) The quality of an auto-colorization model is highly dependent on the sketch-extraction technique used to produce the synthetic data.
(2) Using our sketch-extraction method improves the quality of auto-colorization models when we have only one sample of authentic sketches drawn by an artist for whom we want to colorize his/her sketches automatically, as shown in Figure 8(a). For this evaluation, we used the four artists dataset and randomly chose only one authentic sketch from each artist as a reference style to imitate his/her sketch style for the synthetic dataset generation.
(3) Our sketch-extraction method in general improves the quality of auto-colorization models by improving the quality of synthetic data, as shown in Figure 8(b). For this evaluation, we used our method to generate sketches with ten random styles for each colorized image of the Twitter dataset.

Auto-colorization model trained using our synthetic dataset outperforms all existing baselines both qualitatively and quantitatively.

7.2 Extracting sketches from real-world and face images

Our method can be used as a general-purpose edge detector to extract edges from any colorized photo as long as we provide a proper training dataset. Therefore, one direction worthwhile of exploration is to train our model as a general-purpose edge detector using a dataset that focuses not only on anime characters. Then, our method can also be used to generate artistic sketches from real-world photos. As a representative example, we trained our model using only 100 pairs of real-world images and their corresponding sketches. As shown in Figure 9, our method successfully extracts sketches with various styles from the unseen real-world images. Moreover, we trained our model on real face images. Using CUHK face-sketch dataset [Wang and Tang 2009], we made 3 different styles of portrait sketches for 30 real face images (i.e., only 90 pairs in total for training). As shown in Figure 10, our method can generalize on extracting sketches from unseen real-face images, similar to the given reference styles.

8 LIMITATIONS AND FUTURE WORK

Our model extracts a sketch from a colorized image in such a way that the extracted sketch has a line style similar to that of a given reference sketch. We showed both qualitatively and quantitatively that our method is capable of imitating various sketch styles, commonly used in drawing anime characters (*Evaluation 2* in Section 6, Section 5.2, and Section 5.5). However, our method sometimes fails if the style of the reference sketch is not a line art (e.g., pointillism sketches), as shown in Figure 11. One might address this problem by providing examples of these styles in our training set. Moreover, if we increase the penalty weight of the sparse loss λ_{sparse} in Equation (7), the sparsity of the extracted sketches will increase.

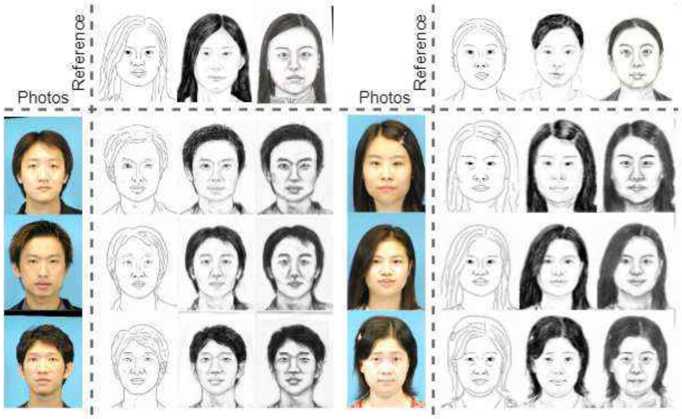


Fig. 10. Our method can generalize on extracting sketches from unseen real-face images [Wang and Tang 2009], similar to the given reference styles.

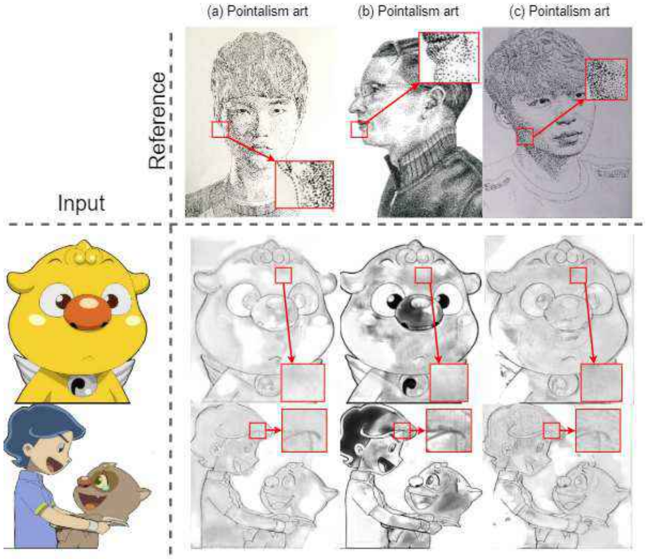


Fig. 11. Failure cases occurred when reference sketches are the pointalism art, considerably different from the line art. © SeoulCityBrand (Input), Kang (Reference).

Therefore, the extracted sketches sometimes might contain inaccurate dense lines or missing lines, if the penalty weight of the sparse loss λ_{sparse} is not tuned correctly with exceedingly low or high values, as shown in Figure 12 and the supplementary material.

Given a reference sketch style, we applied our model to extract sketches from colorized videos (see the supplementary video). While the extracted sketches from most of the video frames are satisfactory, in some consecutive video frames with sudden motions or scene changes, our method fails to achieve temporally coherent results. This occurs because our model is designed to extract a sketch from a single frame and therefore does not explicitly enforce any temporal consistency constraints. A future direction worthwhile to explore is to enforce temporal consistency to expand our method into the video domain. Moreover, most learning-based sketch-extraction approaches suffer from resulting low-resolution sketches. Therefore, another possible research direction is to increase the resolution of the extracted sketches.

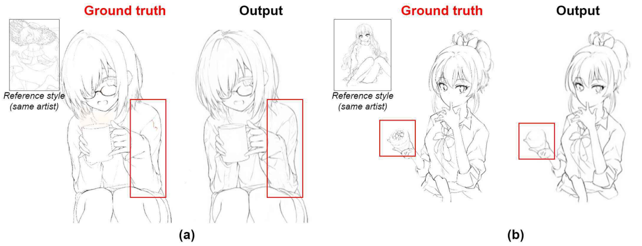


Fig. 12. If we increase the penalty weight of the sparse loss λ_{sparse}, the sparsity of the extracted sketches will increase. Therefore, without tuning the sparse loss weight, extracted sketches might contain inaccurate dense lines (a) or missing lines (b) with exceedingly low or high λ_{sparse} values. © left to right: Ayul_oekaki, Comete_atr.

We showed both qualitatively and quantitatively that the quality of an auto-colorization model depends on the sketch-extraction technique used to produce synthetic datasets for training. In addition, using our Twitter dataset, we proved that our model can improve auto-colorization performance by extracting sketches with various styles. As future work, similar to extracting sketches with various styles from our Twitter dataset, one can use our method to extract sketches with various styles from different datasets commonly used for training auto-colorization models, such as in Aizawa et al. [2020] and Danbooru database [DanbooruCommunity 2021; Zhang et al. 2020]. Then, the synthetic dataset generated by our method can be used as a benchmark training set to train auto-colorization models.

9 CONCLUSION

We presented the first approach to extract a sketch from a colorized image with the style similar to the given reference sketch and with content identical to that in the colorized image. Lacking the necessary volumes of paired authentic sketches and colorized images data, we proposed a novel training scheme by integrating a self-reference sketch style generator to produce various reference sketches with a similar style but different spatial layouts. In our network design, we use three independent attention modules to enable our model to detect edges of a colorized image, learn the line style of a reference sketch, and transfer the line style of the reference sketch to visually corresponding parts of the colorized image edges. We apply several loss terms to imitate the sketch style and enforce sparsity in the extracted sketches. We used our sketch-extraction technique to improve the performance of auto-colorization models by producing a realistic synthetic dataset to train these models. We evaluated our method in a number of qualitative and quantitative experiments. The results suggest that our method outperforms all existing sketch-extraction techniques. Moreover, we introduced a new cyclic evaluation metric to measure how our model preserves a reference sketch style in multiple consecutive sketch extractions. Finally, our user-study results confirmed that participants can easily distinguish the visual differences between various sketch styles and that they preferred our method over existing baselines.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00450, A Deep Learning Based Immersive AR Content Creation Platform for Generating Interactive, Context and Geometry Aware Movement from a Single Image). Special thanks to Ayul_oekaki, Okera_sz, Stellar_sound, Chobi_chu, Maromayu, Dolphilia, Tk_painter, BioTroy, AkaneNagano, AonoriwaKame, Meisaki, Kwanggyoon Seo, Joonkyu Park, and Minju Kim.

REFERENCES

Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, and Hikaru Ikuta. 2020. Building a Manga Dataset "Manga109" With Annotations for Multimedia Applications. *IEEE MultiMedia* 27, 2 (2020), 8–18. https://doi.org/10.1109/MMUL.2020.2987895

Arash Akbarinia and C. Alejandro Párraga. 2018. Feedback and Surround Modulated Boundary Detection. *International Journal of Computer Vision* 126 (12 2018). https://doi.org/10.1007/s11263-017-1035-5

Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2011. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5 (2011), 898–916. https://doi.org/10.1109/TPAMI.2010.161

James Arvo and Kevin Novins. 2000. Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. In *Proceedings of the 13th annual ACM symposium on User interface software and technology.* 73–80.

Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology.* 151–160.

Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. 2014. DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection. (12 2014).

Ali Borji. 2021. Pros and Cons of GAN Evaluation Measures: New Developments. (03 2021).

J Canny. 1986. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (June 1986), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

John F. Canny. 1983. Finding Edges and Lines in Images. *Theory of Computing Systems Mathematical Systems Theory* (1983), 16.

Ruizhi Cao, Haoran Mo, and Chengying Gao. 2021. Line Art Colorization Based on Explicit Region Segmentation. *Computer Graphics Forum* (2021). https://doi.org/10.1111/cgf.14396

Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. 2018. PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 40–48. https://doi.org/10.1109/CVPR.2018.00012

Yajing Chen, Shikui Tu, Yuqi Yi, and Lei Xu. 2017. Sketch-pix2seq: a model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121* (2017).

Wonwoong Cho, Sungha Choi, David Park, Inkyu Shin, and Jaegul Choo. 2018. Image-to-Image Translation via Group-wise Deep Whitening and Coloring Transformation.

Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. 2018. User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks *(MM '18).* Association for Computing Machinery, New York, NY, USA, 1536–1544. https://doi.org/10.1145/3240508.3240661

DanbooruCommunity. 2021. Danbooru2020: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset. https://www.gwern.net/Danbooru2020. https://www.gwern.net/Danbooru2020 Accessed: 2021/11/03.

Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 994–1003.

Piotr Dollár and C. Zitnick. 2014. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (06 2014). https://doi.org/10.1109/TPAMI.2014.2377715

Tzu-Ting Fang, Duc Minh Vo, Akihiro Sugimoto, and Shang-Hong Lai. 2021. Stylized-Colorization for Line Arts. In *2020 25th International Conference on Pattern Recognition (ICPR).* 2033–2040. https://doi.org/10.1109/ICPR48806.2021.9412756

Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. 2016. Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–10.

Noa Fish, Lilach Perry, Amit Bermano, and Daniel Cohen-Or. 2020. SketchPatch: Sketch stylization via seamless patch-level synthesis. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.

Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. 2017. Comicolorization: Semi-Automatic Manga Colorization. In *SIGGRAPH Asia 2017 Technical Briefs* (Bangkok, Thailand) *(SA '17).* Association for Computing Machinery, New York, NY, USA, Article 12, 4 pages. https://doi.org/10.1145/3145749.3149430

Yaroslav Ganin and Victor Lempitsky. 2014. N^4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms. https://doi.org/10.1007/978-3-319-16808-1_36

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning.* PMLR, 1180–1189.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Xin-Yi Gong, Hu Su, De Xu, Zhengtao Zhang, Fei Shen, and Hua-Bin Yang. 2018. An Overview of Contour Detection Approaches. *International Journal of Automation and Computing* 15 (06 2018), 1–17. https://doi.org/10.1007/s11633-018-1117-z

Stéphane Grabli, Frédo Durand, and Francois X Sillion. 2004. Density measure for line-drawing simplification. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.* IEEE, 309–318.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.

Cosmin Grigorescu, Nicolai Petkov, and Michel Westenberg. 2003. Contour detection based on nonclassical receptive field inhibition. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 12 (02 2003), 729–39. https://doi.org/10.1109/TIP.2003.814250

Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. 2018. Arbitrary Style Transfer with Deep Feature Reshuffle. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 8222–8231.

Yliess HATI, GREGOR JOUET, FRANCIS ROUSSEAUX, and Clement DUHART. 2019. PaintsTorch: A User-Guided Anime Line Art Colorization Tool with Double Generator Conditional Adversarial Network. In *European Conference on Visual Media Production* (London, United Kingdom) *(CVMP '19).* Association for Computing Machinery, New York, NY, USA, Article 5, 10 pages. https://doi.org/10.1145/3359998.3369401

Xavier Hilaire and Karl Tombre. 2006. Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 6 (2006), 890–904.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning.* PMLR, 1989–1998.

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. 1510–1519. https://doi.org/10.1109/ICCV.2017.167

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-Image Translation. (04 2018).

Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An Adaptive Edge Detection Based Colorization Algorithm and Its Applications. In *Proceedings of the 13th Annual ACM International Conference on Multimedia* (Hilton, Singapore) *(MULTIMEDIA '05).* Association for Computing Machinery, New York, NY, USA, 351–354. https://doi.org/10.1145/1101149.1101223

Satoshi Iizuka and Edgar Simo-Serra. 2019. DeepRemaster: Temporal Source-Reference Attention Networks for Comprehensive Video Enhancement. *ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA)* 38, 6 (2019), 1.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2016. Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–11.

P. Isola, J. Zhu, T. Zhou, and A. A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 5967–5976.

Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. 2020. Neural Style Transfer: A Review. *IEEE Transactions on Visualization & Computer Graphics* 26, 11 (nov 2020), 3365–3385. https://doi.org/10.1109/TVCG.2019.2921336

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016a. Perceptual Losses for Real-Time Style Transfer and Super-Resolution, Vol. 9906. 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016b. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision.* Springer, 694–711.

Hyuncheol Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. 2019. Tag2Pix: Line Art Colorization Using Text Tag With SECat and Changing Loss. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 9055–9064.

Junho Kim, Minjae Kim, Hyeonwoo Kang, and KwangHee Lee. 2020. U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation. *ArXiv* abs/1907.10830 (2020).

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. 2020. Reference-Based Sketch Image Colorization Using Augmented-Self Reference and Dense Semantic Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Chengze Li, Xueting Liu, and Tien-Tsin Wong. 2017. Deep extraction of manga structural lines. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. 2019. Photo-Sketching: Inferring Contour Drawings From Images. 1403–1412. https://doi.org/10.1109/WACV.2019.00154

Yunhong Li, Yuandong Bi, Weichuan Zhang, and Changming Sun. 2020. Multi-Scale Anisotropic Gaussian Kernels for Image Edge Detection. *IEEE Access* 8 (01 2020), 1803–1812. https://doi.org/10.1109/ACCESS.2019.2962520

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer through Deep Image Analogy. *ACM Trans. Graph.* 36, 4, Article 120 (July 2017), 15 pages. https://doi.org/10.1145/3072959.3073683

Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. 2019b. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5830–5839.

Rui Liu, Chengxi Yang, Wenxiu Sun, Xiaogang Wang, and Hongsheng Li. 2020. Stereogan: Bridging synthetic-to-real domain gap by joint optimization of domain translation and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 12757–12766.

Xueting Liu, Xiangyu Mao, Xuan Yang, Linling Zhang, and Tien-Tsin Wong. 2013. Stereoscopizing cel animations. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10.

Xueting Liu, Tien-Tsin Wong, and Pheng-Ann Heng. 2015. Closure-aware sketch simplification. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–10.

Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. 2019a. Richer Convolutional Features for Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 1939–1946. https://doi.org/10.1109/TPAMI.2018.2878849

lllyasviel/sketchKeras 2018. *sketch keras.* Retrieved 2020-04-22 from https://github.com/lllyasviel/sketchKeras

Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep Photo Style Transfer. (03 2017).

Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. 2019. Exemplar Guided Unsupervised Image-to-Image Translation with Semantic Consistency. *ICLR* (2019).

Julien Mairal, Marius Leordeanu, Francis Bach, Martial Hebert, and J. Ponce. 2008. Discriminative Sparse Image Models for Class-Specific Edge Detection and Image Interpretation. 43–56. https://doi.org/10.1007/978-3-540-88690-7_4

David A. Mély, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. 2016. A systematic comparison between visual cues for boundary detection. *Vision Research* 120 (2016), 93–107.

Gioacchino Noris, Alexander Hornung, Robert W Sumner, Maryann Simmons, and Markus Gross. 2013. Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics (TOG)* 32, 1 (2013), 1–11.

Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. 2021. Image-to-Image Translation: Methods and Applications.

Taesung Park, Alexei Efros, Richard Zhang, and Jun-Yan Zhu. 2020. Contrastive Learning for Unpaired Image-to-Image Translation. (11 2020), 319–345. https://doi.org/10.1007/978-3-030-58545-7_19

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2332–2341. https://doi.org/10.1109/CVPR.2019.00244

Jinye Peng, Jiaxin Wang, Jun Wang, Erlei Zhang, Qunxi Zhang, Yongqin Zhang, Xianlin Peng, and Kai Yu. 2021. A relic sketch extraction framework based on detail-aware hierarchical deep network. *Signal Processing* 183 (2021), 108008.

P. Perona and J. Malik. 1990. Detecting and localizing edges composed of steps, peaks and roofs. In *[1990] Proceedings Third International Conference on Computer Vision.* 52–57. https://doi.org/10.1109/ICCV.1990.139492

Yonggang Qi, Yi-Zhe Song, Tao Xiang, Honggang Zhang, Timothy Hospedales, Yi Li, and Jun Guo. 2015. Making better use of edges via perceptual grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1856–1865.

Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. 2006. Manga colorization. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 1214–1220.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 10912–10922.

Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. 2018. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3752–3761.

Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Joint gap detection and inpainting of line drawings. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 5725–5733.

Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2018. Learning to restore deteriorated line drawing. *The visual computer* 34, 6 (2018), 1077–1085.

Chang Wook Seo and Yongduek Seo. 2021. Seg2pix: Few Shot Training Line Art Colorization with Segmented Image Data. *Applied Sciences* 11, 4 (2021). https://doi.org/10.3390/app11041464

Tamar Rott Shaham, Michaël Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. 2021. Spatially-Adaptive Pixelwise Networks for Fast Image Translation. In *CVPR.*

Amit Shesh and Baoquan Chen. 2008. Efficient and dynamic simplification of line drawings. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 537–545.

Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018a. Mastering Sketching: Adversarial Augmentation for Structured Prediction. *ACM Trans. Graph.* 37, 1, Article 11 (jan 2018), 13 pages. https://doi.org/10.1145/3132703

Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. 2018b. Real-time data-driven interactive rough sketch inking. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. 2016a. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. 2016b. Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. *ACM Trans. Graph.* 35, 4, Article 121 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925972

Xavier Soria Poma, Edgar Riba, and Angel Sappa. 2020. Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection. 1912–1921. https://doi.org/10.1109/WACV45572.2020.9093290

style2paint 2018. *paints chainer.* Retrieved 2020-04-22 from https://style2paints.github.io/

Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision.* Springer, 443–450.

Harrish Thasarathan and Mehran Ebrahimi. 2019. Artist-Guided Semiautomatic Animation Colorization. 3157–3160. https://doi.org/10.1109/ICCVW.2019.00388

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. 2018. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 7472–7481.

twitter policy 2021. *twitter policy.* Retrieved 2021-04-30 from https://developer.twitter.com/en/developer-terms/agreement-and-policy

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135–153.

Xiaogang Wang and Xiaoou Tang. 2009. Face Photo-Sketch Synthesis and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 11 (2009), 1955–1967. https://doi.org/10.1109/TPAMI.2008.222

Yupei Wang, Xin Zhao, and Kaiqi Huang. 2017. Deep Crisp Boundaries. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 1724–1732. https://doi.org/10.1109/CVPR.2017.187

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Brett Wilson and Kwan-Liu Ma. 2004. Rendering complexity in computer-generated pen-and-ink illustrations. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering.* 129–137.

Holger Winnemöller. 2011. XDoG: Advanced Image Stylization with EXtended Difference-of-Gaussians. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering* (Vancouver, British Columbia, Canada) *(NPAR '11).* Association for Computing Machinery, New York, NY, USA, 147–156. https://doi.org/10.1145/2024676.2024700

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV).*

Ren Xiaofeng and Liefeng Bo. 2012. Discriminatively Trained Sparse Code Gradients for Contour Detection. In *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/16a5cdae362b8d27a1d8f8c7b78b4330-Paper.pdf

Xiao Yang Yiheng Zhu Xiaohui Shen Xiaoyu Xiang, Ding Liu. 2021. Anime2Sketch: A Sketch Extractor for Anime Arts with Deep Networks. https://github.com/Mukosame/Anime2Sketch.

Minshan Xie, Chengze Li, Xueting Liu, and Tien-Tsin Wong. 2020. Manga filling style conversion with screentone variational autoencoder. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Minshan Xie, Menghan Xia, Xueting Liu, Chengze Li, and Tien-Tsin Wong. 2021. Seamless manga inpainting with semantics awareness. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–11.

Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 1395–1403. https://doi.org/10.1109/ICCV.2015.164

Xuemiao Xu, Minshan Xie, Peiqi Miao, Wei Qu, Wenpeng Xiao, Huaidong Zhang, Xueting Liu, and Tien-Tsin Wong. 2021. Perceptual-Aware Sketch Simplification Based on Integrated VGG Layers. *IEEE Transactions on Visualization and Computer Graphics* 27, 1 (2021), 178–189. https://doi.org/10.1109/TVCG.2019.2930512

Kai-Fu Yang, Shao-Bing Gao, Ce-Feng Guo, Chao-Yi Li, and Yong-Jie Li. 2015. Boundary Detection Using Double-Opponency and Spatial Sparseness Constraint. *IEEE Transactions on Image Processing* 24, 8 (2015), 2565–2578. https://doi.org/10.1109/TIP.2015.2425538

Ming-Hsuan Yang, D.J. Kriegman, and N. Ahuja. 2002. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 1 (2002), 34–58. https://doi.org/10.1109/34.982883

Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. 2019. Photorealistic Style Transfer via Wavelet Transforms. 9035–9044. https://doi.org/10.1109/ICCV.2019.00913

Aron Yu and Kristen Grauman. 2014. Fine-Grained Visual Comparisons with Local Learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 192–199. https://doi.org/10.1109/CVPR.2014.32

Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. 2017. Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision* 122, 3 (2017), 411–425.

Mingcheng Yuan and Edgar Simo-Serra. 2021. Line Art Colorization With Concatenated Spatial Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 3946–3950.

Kaihua Zhang, Lei Zhang, Kin-Man Lam, and David Zhang. 2016. A Level Set Approach to Image Segmentation With Intensity Inhomogeneity. *IEEE Transactions on Cybernetics* 46, 2 (2016), 546–557. https://doi.org/10.1109/TCYB.2015.2409119

Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. 2017. Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN. 506–511. https://doi.org/10.1109/ACPR.2017.61

Lvmin Zhang, Yi Ji, and Chunping Liu. 2020. DanbooRegion: An Illustration Region Dataset. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 137–154.

Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. 2018b. Two-Stage Sketch Colorization. *ACM Trans. Graph.* 37, 6, Article 261 (Dec. 2018), 14 pages. https://doi.org/10.1145/3272127.3275090

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018a. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 586–595. https://doi.org/10.1109/CVPR.2018.00068

Haitian Zheng, Haofu Liao, Lele Chen, Wei Xiong, Tianlang Chen, and Jiebo Luo. 2020. Example-Guided Image Synthesis Using Masked Spatial-Channel Attention and Self-supervision. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 422–439.

Xingran Zhou, Bo Zhang, Ting Zhang, Pan Zhang, Jianmin Bao, Dong Chen, Zhongfei Zhang, and Fang Wen. 2020. Full-Resolution Correspondence Learning for Image Translation.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward Multimodal Image-to-Image Translation *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 465–476.

Djemel Ziou and Salvatore Tabbone. 2000. Edge Detection Techniques - An Overview. 8 (06 2000).