Received: 22 February 2021 | Revised: 15 August 2021 | Accepted: 4 September 2021

IET Image Processing

DOI: 10.1049/ipr2.12343

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Disentangled and controllable sketch creation based on disentangling the structure and color enhancement

Nan Gao | **Hui Ren** | **Jia Li** | **ZhiBin Su**

State Key Laboratory of Media Convergence and Communication; Key Laboratory of Acoustic Visual Technology and Intelligent Control System, Ministry of Culture and Tourism (Communication University of China), Beijing Key Laboratory of Modern Entertainment Technology (Communication University of China), School of Information and Communication Engineering, Communication University of China, Dingfuzhuang Street(E), Beijing, China

Correspondence
Nan Gao, State Key Laboratory of Media Convergence and Communication; Key Laboratory of Acoustic Visual Technology and Intelligent Control System, Ministry of Culture and Tourism (Communication University of China); Beijing Key Laboratory of Modern Entertainment Technology (Communication University of China); School of Information and Communication Engineering, Communication University of China, Dingfuzhuang Street(E), Beijing, 100024, China.
Email: gaonan-nan@cuc.edu.cn

Funding information
Research Project of the Communication University of China, Grant/Award Number: CUC200D058; National Key Research and Development Plan, Grant/Award Number: YS2018YFB1403703

Abstract

Existing sketch-based image processing methods include sketch recognition, sketch synthesis and sketch-based image retrieval. For sketch creation, a meaningful task is proposed namely disentangled and controllable sketch creation (DCSC) based on disentangling the structure and color enhancement. Specifically, as the first subtask, sketch structure enhancement (SSE) is used to enhance a non-professional sketch (NPS) and obtain a professional sketch (PS), which is a process denoted as NPS2PS. A data set named SketchMan is first provided, consisting of NPSs and PSs with various postures in different scenes. SSE is trained as a conditional image-to-image translation problem, and there are three models: direct sketch-to-sketch (SS), grayscale guided SS and contour guided SS. Multiple IOU metrics are proposed based on Corner Point Map (CPM), Straight Line Map (SLM) and Segmented Area Map (SAM). As the second subtask, sketch color enhancement (SCE) is trained as a two-stage framework containing a topology enhancement network (TE-Net) that maps a sketch to the corresponding grayscale domain and a color injection network (CI-Net) that injects the global color feature to the AdaIN residual blocks to perform adaptive sketch colorization. The TE-Net and CI-Net disentangle the topological and color features to perform more controllable and diverse SCE results. Experimental results demonstrate that our proposed methods are effective to address the challenging and meaningful DCSC task compared with other state-of-the-art methods.

1 | INTRODUCTION

Sketches have been studied in image classification, generation and retrieval [1–6]. A group of human-drawn strokes are supposed to depict an recognizable object. There are some existing datasets such as QMUL Shoe-V2 dataset [7], TU-Berlin dataset [8] and QuickDraw [6]. These collected sketches often consist of non-professional strokes and lack high-quality visual performance. Professional drawings usually belong to experienced artists. We present some examples in Figure 1a where the painter [9] gives a new life to his son's simple drawings by supplementing more important details and improving the abstract strokes. It is more difficult for a machine to model the refinement from NPS with ambiguous semantics to PS. More and more tasks have been studied well based on GANs [10–13]. It is also promising for the sketch structure enhancement (SSE), where NPS2PS is achieved, as shown in Figure 1b.

In this work, we propose disentangled and controllable sketch creation (DCSC) based on disentangling structure and color enhancement, which aims to convert a NPS to the corresponding PS by means of deep learning. We disentangle DCSC into two subtasks, that is, sketch structure enhancement (SSE) and sketch color enhancement (SCE). SSE learns the NPS2PS mapping by exploring three different routines, and the two-stage SCE consists of the topology enhancement network (TE-Net) and color injection network (CI-Net), which are used to disentangle the topological structure and color features to achieve more controllable and diverse sketch colorization. Note that the

(a) NPS (left) and color PS (right) drawn by human (b) NPS (upper) and PS (lower) generated by computer

FIGURE 1 Sketch structure enhancement (SSE) is an NPS2PS task conducted to synthesize many delicate works. We show some NPS and PS images drawn by human and computer

FIGURE 2 Sketch color enhancement (SCE) results based on various reference images. We show 6 synthesized colored sketches with respect to the different color styles

data distribution of the enhanced results of challenging SSE is still far from that of the original sketches. We verify the performance of our model using the original sketches in the SCE subtask, as shown in Figure 2.

Edge information is vital for many image translation tasks, and SSE task is more special. First, many color or semantic rendering tasks [11, 14] need to generate more details based on the large number of object edges. As for the inpainting tasks [5, 15] with removed contents, the edges of the input image are slightly distorted. This is similar to the superresolution task [16] as well. Moreover, scene generation tasks [17, 18] generally translate a new scene from a caption, scene graph or detection box, which may have many structural distortions rather than precise edge features.

In an SSE task, the edge information has extensive structural distortions as well, which needs to be completed and optimized. Furthermore, the fault of SSE results are easier to be exposed. After NPS2PS, if there are unreasonable strokes on a white canvas, this will cause an uncomfortable visual effect for human eyes. To benefit the research on SSE and other related works, the established data set SketchMan provides NPS covering diverse appearances and postures in both simple and complex scenes, as shown in Figure 3. Note that SketchMan contains both simple characters and multiperson scenes. The sketch structures are more complex than those of other databases[6, 8, 19]. For each sample, there are two kinds of NPS with different degrees of alignment to PS strokes. As shown in Figure 4, freehand NPS is sparser than the approximate PS. In SketchMan, high-level feature maps contain the background mask maps, grayscale maps as well as color maps. As for low-level features, three kinds of maps are obtained by means of corner detection, straight line detection and segmentation based on superpixel clustering.

For the SCE task, inspired by recently proposed works [20–24], we adopt adaptive instance normalization (AdaIN) [25] to control the color style transformation of the sketch. Specifically, we conduct controllable sketch colorization by disentangling the topological and color factors; that is, the first stage TE-Net is trained to generate a grayscale map of the sketch that represents the topology completion result, on the basis of which the second-stage CI-Net performs diverse color modifications based on a specific reference map. In this way, both the topological and color features are enhanced to generate more controllable and detailed color maps.

In summary, the main contributions are as follows:

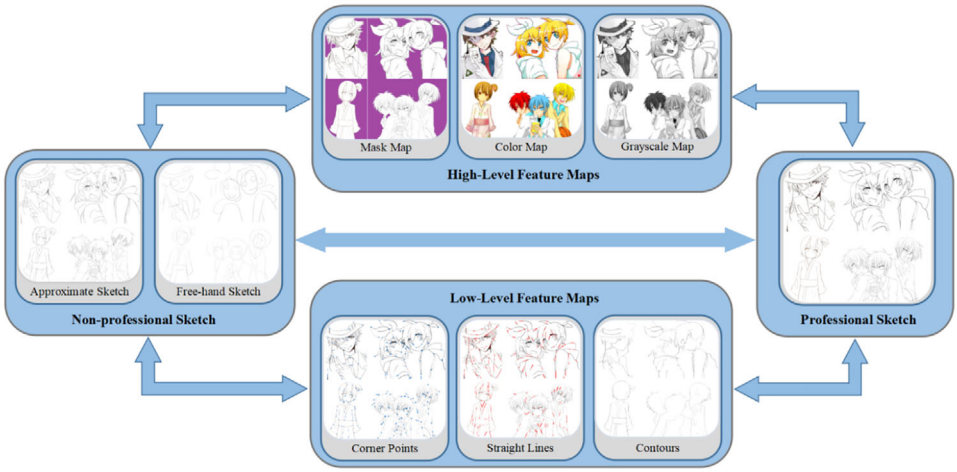

FIGURE 3 An overview of the proposed dataset SketchMan. There are four parts, that is, non-professional sketches, professional sketches, high-level feature maps and low-level feature maps

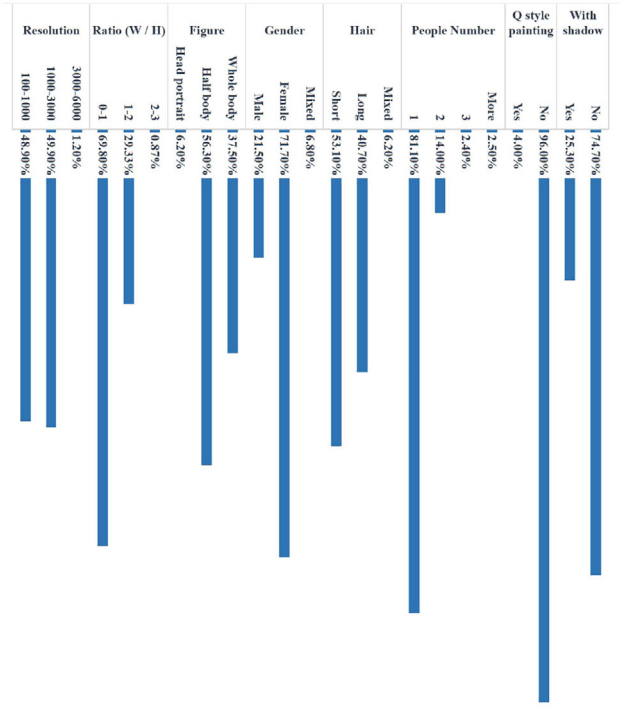

FIGURE 4 Examples of manga creation in our database. The first row is a PS, the second row is an approximate professional NPS, and the last row is a free-hand NPS

FIGURE 5 The statistics of character attributes in SketchMan

- We propose disentangled and controllable sketch creation (DCSC) based on disentangling sketch structure enhancement (SSE) and sketch color enhancement (SCE).
- We establish SketchMan that covers various high-resolution anime characters. The statistics of character attributes in SketchMan is shown in Figure 5. Furthermore, the feature maps in high-level and low-level domains are provided for comprehensive analysis.
- We propose SS, grayscale guided SS and contour guided SS to address the SSE task, respectively. In addition to ODS (Optimal Dataset Scale), OIS (Optimal Image Scale) and AP (Average Precision), we propose to conduct the intersection over union (IoU) evaluation of the NPS and PS with respect to points, straight lines, areas and strokes, which established an important benchmark.
- We propose a disentangled and controllable sketch colorization method that includes a topology enhancement network (TE-Net) and a color injection network (CI-Net). Compared with state-of-the-art methods, including pix2pix [10], PaintsChainer [26], Style2Paints [14] and DeepColor [27], the proposed SCE model can achieve higher-quality automatic sketch colorization.

This paper is an extension of our previous conference version [28]. In addition to providing more extensive experiments and in-depth analysis, there are three major differences between this

paper and its previous version: 1) DCSC is proposed based on disentangling SSE and SCE. 2) Another new variant of SS task with self attention is proposed and the corresponding experiments are conducted. 3) An efficient SCE framework is proposed to achieve controllable sketch colorization based on arbitrary reference maps.

2 | RELATED WORK

2.1 | Sketch generation

Sketch generation based on variational model [6] can automatically produce novel stroke sequences. Sketch abstraction [4, 29] can generate sparser sketch whose semantic still could be recognized correctly. The sketch completion task [5] is a sketch inpainting task in white canvas. Casually drawn strokes are refined by color rendering in SketchyGAN [2]. SSE for animated characters is far more complex and challenging than these works.

Sketch colorization usually is based on edges whose distribution is approximate with the real professional sketches. Some examples include PaintsChainer [26], Scribbler [30], Style2paints [14] and Comi-colorization [31].

2.2 | Generative models based on adversarial learning

Generative adversarial networks (GANs) [32, 33] have been widely used to improve many tasks, such as neural rendering [2], content completion [5], attribute editing [11] and style transfer [12, 13]. Compared with other generative models, such as variational autoencoders (VAEs) and flow-based models, GANs achieve higher-quality results by means of adversarial learning.

2.3 | Styled image synthesis

There are some related works on styled image synthesis [13, 20–24, 34–36]. StyleGAN [34] can generate photorealistic faces from noise after disentangling the attribute factors using a feature-mapping network. U-GAT-IT [13] conducts unsupervised image-to-image translation with adaptive layer-instance normalization and class activation map (CAM) loss. SimSwap [35] proposes a simple face-swapping framework using AdaIN [25] to inject the identity feature of the source face. FaceShifter [36] adopts SPADE [37] for the attributes of the target and identity of the source to adaptively generate the swapped face.

2.4 | Dataset

There are some widely used datasets for sketch-based tasks, for example, Sketchy database [19], QuickDraw [6], QMUL Shoe/Chair SBIR datasets [7], TU-Berlin [8], CUHK Face Sketches [38] and SketchyScene [39]. Some examples of these datasets are shown in Figure 6. These datasets are not as complex as the anime characters in Japanese cartoons.

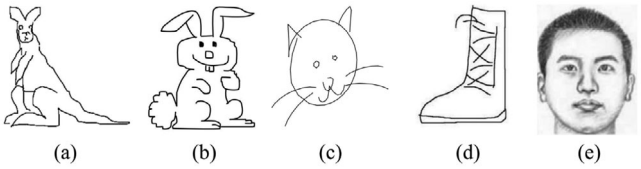

FIGURE 6 Some examples of previous datasets. (a) Sketchy database [19], (b) TU-Berlin [8], (c) QuickDraw [6], (d) QMUL SBIR dataset [7], and (e) CUHK Sketches [38]

3 | APPROACH

In this section, we introduce DCSC based on disentangling the structure and color enhancement, as shown in Figure 7. For SSE, we first present our dataset. Then, we explore three SSE routines considering different pixel distributions. As the second subtask, we introduce a novel approach to perform disentangled and controllable SCE, described below.

3.1 | Dataset

SketchMan selects 2120 high-quality PS samples from pixiv [40], covering approximately 2690 animation characters. The attribute distributions of SketchMan are shown in Figure 5. We invited students to mimic these professional sketches, obtaining corresponding free-hand NPS images and approximate professional sketch (APS) images. Free-hand NPS aims to simulate the random and ambiguous semantic layout. APS is drawn using a fixed brush size. We study the challenging and meaningful SSE based on free-hand NPS. In the high-level sketch domain, we filter the background content using the corresponding mask based on PS and maintain the main body areas of the anime characters, as shown in Figure 8.

3.1.1 | sketch abstraction based on SLIC

The low-frequency information of the original sketch is the abstract sketch. Unlike previous works [4, 29], we utilize simple linear iterative clustering (SLIC) [41] based on superpixel clustering of the real PS rather than a color image of PS, to obtain the shape parsing map in the low-level sketch domain, as shown in Figure 9. Specifically, SLIC is formulated as:

$$D_c = \left[\left(L_j - L_i\right)^2 + \left(a_j - a_i\right)^2 + \left(b_j - b_i\right)^2 \right]^{1/2}, \quad (1)$$

$$D_s = \left[\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2 \right]^{1/2}, \quad (2)$$

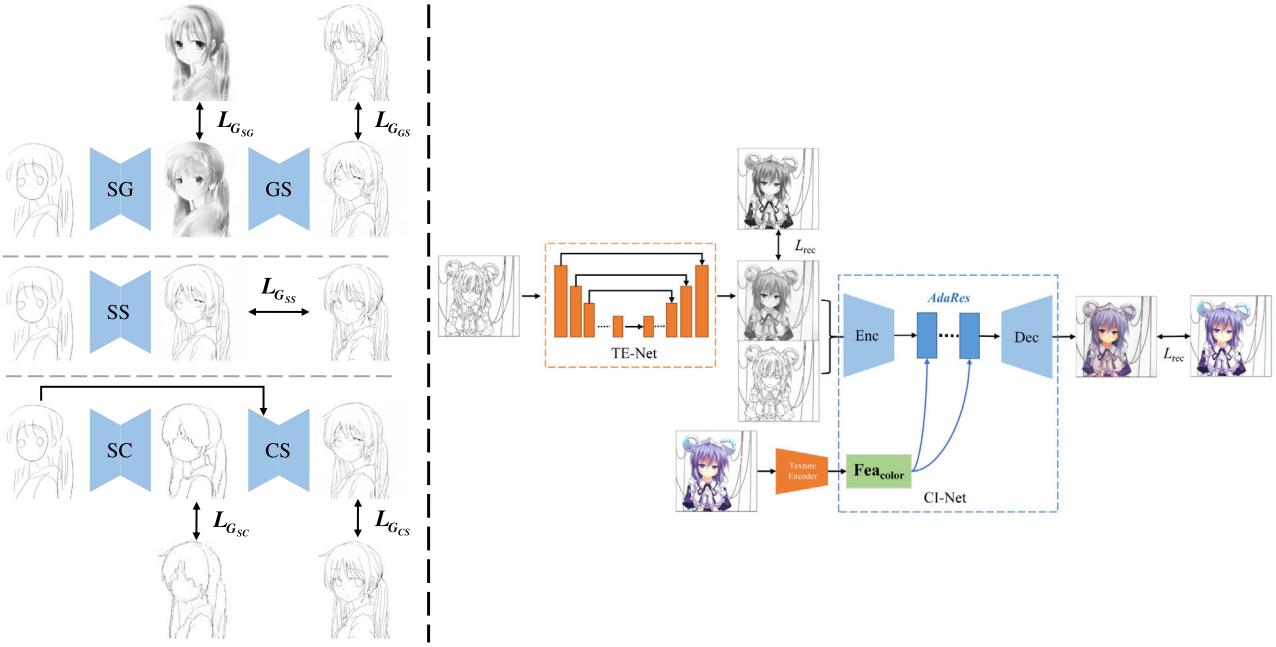

FIGURE 7 An overview of the proposed method. (a) SSE contains three models: (1) SS, that is, an end-to-end sketch translation to enhance a PS. (2) Grayscale guided SS, that is, the grayscale PS is used as an intermediate supervised signal. SG means the sketch-to-grayscale module, and GS means the sketch extraction module. (3) Contour guided SS, that is, fitting the sketch abstraction of PS in the first stage. SC means the sketch-to-contour module, and CS means the sketch refinement module. (b) SCE contains a topology enhancement network (TE-Net) and a color injection network (CI-Net), which aims to complete the topological structure of the sketch and adaptively transfer the color style based on the reference color map in a self-supervised manner. Considering the quality of SSE results is still low, SSE and SCE are independently trained in this paper

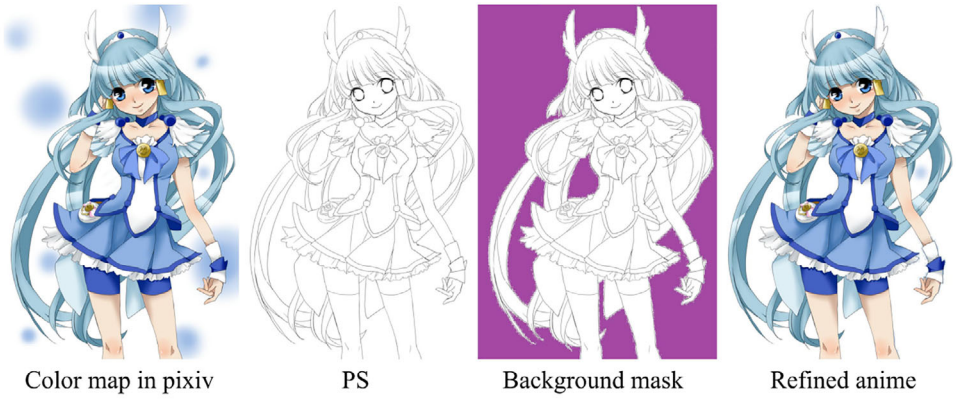

FIGURE 8 Background of color map is filtered by the mask based on the sketch

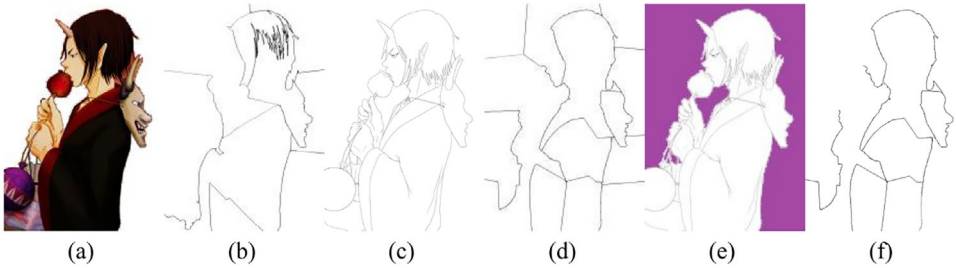

FIGURE 9 The intermediate results of the segmented area map based on SLIC. (a) The color map; (b) the SLIC result of (a). (c) Original PS; (d) SLIC result of (c); (e) background mask; (f) the filtered result of (d) using (e)

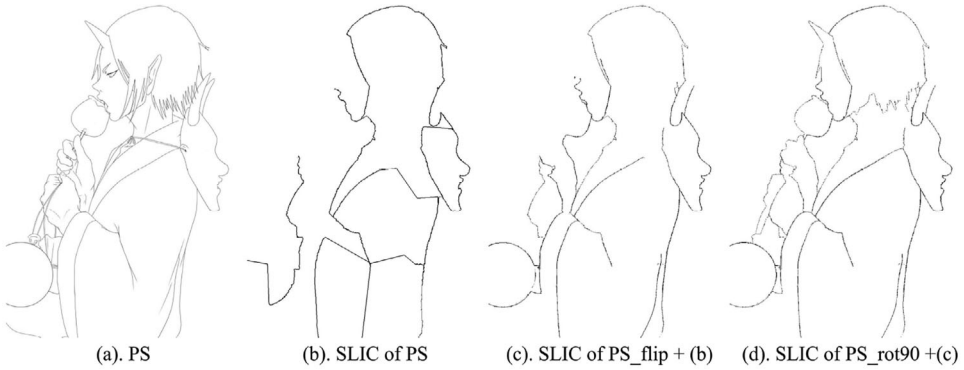

FIGURE 10 Sketch abstraction considering simple linear iterative clustering (SLIC) for the flipped and rotated sketch

FIGURE 11 More examples of sketch abstraction based on SLIC

$$D = \left[(D_c/N_c)^2 + (D_s/N_s)^2 \right]^{1/2}, \tag{3}$$

where the distances in the L-ab color space and pixel space are denoted as D_c and D_s, respectively. The final clustering considers both of them, as shown in Equation 3, where N_c and N_s are used to normalize D_c and D_s, respectively.

Additionally, as shown in Figure 10, we find that sketch abstraction is not ideal while directly using once SLIC operation on the original PS; thus, we exploit another two SLIC after flipping the PS horizontally or rotating the PS 90 degrees along the anticlockwise direction, and then rotate the figure back to its original orientation. Finally, all SLIC maps are combined in the element-wise to obtain the integral regional segmentation map. We show more examples of sketch abstraction in Figure 11.

3.1.2 | NPS augmentation

We use Laplacian mesh editing [42] to generate more NPS images. In Figure 12, the first row shows a PS and four movement situations of 9 feature points, the second row is the initial NPS and the deformed NPS images.

Specifically, vertex i is denoted as $v_i = (x_i, y_i, z_i)$ in the Cartesian coordinate space. The differential coordinates based on the Laplacian operator are defined as follows:

$$L_s(i,j) = \begin{cases} d_i & i = j \\ -1 & (i,j) \in E_a \\ 0 & otherwise \end{cases}, \tag{4}$$

$$\delta_i = \left[\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)} \right] = v_i - \frac{1}{di} \sum_{j|(i,j)\in E_a} v_j, \tag{5}$$

$$L_s x = D\delta^{(x)}, \tag{6}$$

where L_s is the Laplacian metric, E_a indicates whether two vertices are on one edge, and D is a diagonal matrix consisting of d_i values, which are the numbers of vertices adjacent to each vertex. The large linear equation for NPS deformation is formulated as:

$$\left[\frac{L_s}{I_{m\times n}} \right] x = \left[\frac{D\delta^{(x)}}{b_{1:m}^{(x)}} \right], \tag{7}$$

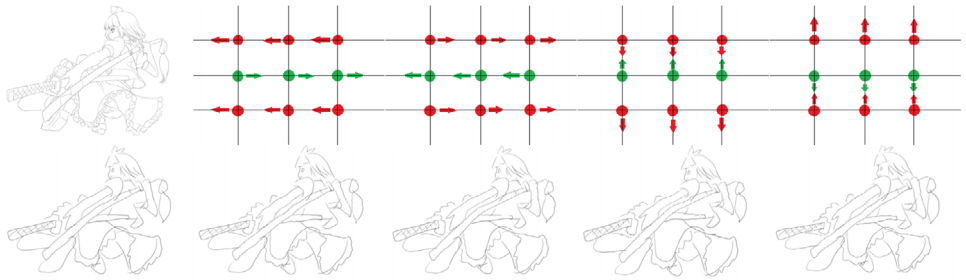

FIGURE 12 NPS augmentation based on Laplacian mesh editing in SketchMan

where $I_{m\times n}$ denotes the identity matrix where the number of moved feature points is m, and b^x represents the positions of the feature points in the Cartesian coordinate space along the x axis.

3.2 | Method

In this subsection, we first present the three pipelines of the SSE task and then introduce the network architecture of SCE, described below.

For the SSE task, we propose three NPS2PS pipelines, that is, SS, grayscale guided SS and contour guided SS, as shown in Figure 7a.

3.2.1 | SS task

A conditional GAN (cGAN) conducts a mapping from observed image x and random noise vector z to a target-domain image $y : G : \{x, z\} \rightarrow y.p_z$ and p_{data} represent the prior distributions for z and the target domain, respectively. And p_G represents the distribution of the synthesis domain. The specific loss is formulated as follows:

$$\mathcal{L}_{cGAN}(G, D) = E_{y\sim p_{data}}[logD(x, y)] + E_{G(x)\sim p_G, z\sim p_z}[log(1 - D(x, G(x, z)))]. \tag{8}$$

Generator will be optimized via:

$$G^* = arg\min_G \max_D \mathcal{L}_{cGAN}(G, D). \tag{9}$$

We train SS model to directly translate an NPS to a PS in a supervised manner, which is a naive benchmark of the SSE task. We use L1 distance and SSIM [43] loss together with the GAN objective. The former is used to decrease blurring, and the latter enhances the structural similarity of $G_{SS}(x)$ and the real PS y.

$$\mathcal{L}_{L1}(G_{SS}) = E_{x,y}\left[\left\|y - G_{SS}(x)\right\|_1\right], \tag{10}$$

$$SSIM(G_{SS}) = E_{x,y}[SSIM(y, G_{SS}(x))]. \tag{11}$$

Moreover, multilevel visual geometry group (VGG) loss is used for preserving perceptual consistency between the generated and real data, where Φ is a pretrained VGG-19 model [44] considering the features of the rectified linear unit (ReLU) {1_1, 2_1, 3_1, 4_1, 5_1}.

$$\mathcal{L}_{vgg}(G_{SS}) = E_{x,y}\left[\left\|\Phi(y) - \Phi(G_{SS}(x))\right\|_1\right]. \tag{12}$$

The min-max function of the SS task is:

$$G^*_{SS} = arg\min_{G_{SS}} \max_{D_{SS}} \mathcal{L}_{cGAN}(G_{SS}, D_{SS}) + \lambda_{L1}\mathcal{L}_{L1}(G_{SS}) + \lambda_{vgg}\mathcal{L}_{vgg}(G_{SS}) + \lambda_{SSIM}SSIM(G_{SS}). \tag{13}$$

We also use the Huber loss to improve the SS model, as shown in Equation 14, where δ is a threshold used to choose the $\mathcal{L}1$ or $\mathcal{L}2$ loss, as shown in Equation 15. We denote this variant of SS as SS+Huber.

$$\mathcal{L}_\delta(G) = \begin{cases} \frac{1}{2}\left|y - G(x)\right|^2 & \left|y - G(x)\right| \le \delta \\ \delta \cdot \left[\left|y - G(x)\right| - \frac{1}{2}\delta\right] & otherwise \end{cases}, \tag{14}$$

$$G^*_{Huber} = arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda_\delta\mathcal{L}_\delta(G) + \lambda_{vgg}\mathcal{L}_{vgg}(G) + \lambda_{SSIM}SSIM(G). \tag{15}$$

To improve the effect of discrimination, we train another advanced SS model. In addition to the L1 loss at the image level, the L1 distances belonging to the intermediate feature space of the discriminator are considered, as shown in Equations (16) and (17). We denote this variant of SS as SS+L1+FM.

$$\mathcal{L}_{FM}(G) = E_{x,y}\left[\left\|D(y) - D(G(x))\right\|_1\right], \tag{16}$$

$$G_{FM}^{*}=\min_{G}\max_{D}\mathcal{L}_{cGAN}(G,D)+\lambda_{L1}\mathcal{L}_{L1}(G)+\lambda_{FM}\mathcal{L}_{FM}(G)+\lambda_{vgg}\mathcal{L}_{vgg}(G)+\lambda_{SSIM}SSIM(G). \tag{17}$$

Additionally, inspired by Transformer [45], we add the self attention layer in the encoder and decoder of the global generator of Pix2pixHD [11]. Specifically, following with the conv, batchnorm and ReLU layers, the attentional channels are selected in the propagation by learning the query, key and value features. We denote this variant of SS as SS+L1+FM+Att.

3.2.2 | Grayscale guided SS task

In the grayscale guided SS model, the contextual information of a complete topological structure of the intermediate grayscale map is more abundant and diverse than the sparse black-and-white content of the original NPS. Concretely, the grayscale guided SS task includes two stages.

(1) The sketch-to-grayscale module, which predicts the topology distributions of the dense pixels under the supervision of grayscale PS, that is, the SG stage learns the mapping from NPS x to the corresponding grayscale PS image $y_G: G_{SG}:\{x\}\rightarrow y_G$:

$$G_{SG}^{*}=\arg\min_{G_{SG}}\max_{D_{SG}}\mathcal{L}_{cGAN}(G_{SG},D_{SG})+\lambda_{L1}\mathcal{L}_{L1}(G_{SG})+\lambda_{vgg}\mathcal{L}_{vgg}(G_{SG})+\lambda_{SSIM}SSIM(G_{SG}). \tag{18}$$

(2) The sketch extraction module, which learns the mapping from the predicted $\hat{y_G}$ in the first stage to the real PS y. Since $\hat{y_G}$ is an improved intermediate feature map whose boundary distribution is closer to the PS domain, we input $\hat{y_G}$ to G_{GS} to implement sketch extraction, and the ground truth is the final PS, denoted as $y: G_{GS}:\{\hat{y_G}\}\rightarrow y$. The generator learns to remove the color pixels around the sparse sketch. The second stage is formulated as:

$$G_{GS}^{*}=\arg\min_{G_{GS}}\max_{D_{GS}}\mathcal{L}_{cGAN}(G_{GS},D_{GS})+\lambda_{L1}\mathcal{L}_{L1}(G_{GS})+\lambda_{vgg}\mathcal{L}_{vgg}(G_{GS})+\lambda_{SSIM}SSIM(G_{GS}). \tag{19}$$

3.2.3 | Contour guided SS task

Sketch abstraction removes the most internal details and maintains the overall outline of the object, which makes it easier to fit the data distributions of the abstract sketches in the sketch-to-contour (SC) stage. Moreover, in the contour-to-sketch (CS) stage, the details of the PS object are mainly synthesized, and the fitting difficulty of NPS2PS is reduced in this coarse-to-fine way. Specifically, the contour guided SS model consists of two stages.

(1) Sketch abstraction, which is trained for the purpose of guiding SSE optimization with the object contour as a shape parsing clue of PS. We input NPS x to SC stage to predict the corresponding PS contour $y_C: G_{SC}:\{x,z\}\rightarrow y_C$, where y_C is extracted by means of SLIC.

$$G_{SC}^{*}=arg\min_{G_{SC}}\max_{D_{SC}}\mathcal{L}_{cGAN}(G_{SC},D_{SC})+\lambda_{L1}\mathcal{L}_{L1}(G_{SC})]+\lambda_{vgg}\mathcal{L}_{vgg}(G_{SC})+\lambda_{SSIM}SSIM(G_{SC}). \tag{20}$$

(2) Sketch refinement, where NPS and the predicted $\hat{y_C}$ of the SC stage are concatenated as the input of CS stage to generate the real PS. Since the NPS usually contains richer edge details compared with the $\hat{y_C}$, we utilize both $\hat{y_C}$ and NPS x as the inputs of the AS network to conduct NPS2PS, denoted as $y: G_{CS}:\{\hat{y_C},x\}\rightarrow y$.

$$G_{CS}^{*}=arg\min_{G_{CS}}\max_{D_{CS}}\mathcal{L}_{cGAN}(G_{CS},D_{CS})+\lambda_{L1}\mathcal{L}_{L1}(G_{CS})+\lambda_{vgg}\mathcal{L}_{vgg}(G_{CS})+\lambda_{SSIM}SSIM(G_{CS}). \tag{21}$$

3.2.4 | SCE task

As shown in Figure 7b, the SCE model contains a TE-Net and a CI-Net. The TE-Net has a U-net structure [46]. Let X_s be the sketch obtained based on SketchKeras [47]. The output of the TE-Net is the generated grayscale map $\hat{X}_g$. The L1 distance is determined as follows:

$$\mathcal{L}_{rec}=\left\|\hat{X}_g-X_g\right\|_1. \tag{22}$$

We further add the perceptual loss to improve the feature matching between $\hat{X}_g$ and the target grayscale map X_g.

$$\mathcal{L}_{per}=\frac{1}{N}\sum_{i=1}^{N}\|F_{vgg}^{(i)}(\hat{X}_g)-F_{vgg}^{(i)}(X_g)\|_2, \tag{23}$$

where $F_{vgg}^{(i)}$ denotes the i^{th} convolution layer of the VGG19 model, that is, Conv{2_1, 3_1, 4_1, 5_1}.

We utilize the contextual loss [48] to measure the feature similarity between $\hat{X}_g$ and X_g. This loss reduces the texture distortions after sketch colorization. It is formulated as

$$\mathcal{L}_{CX}=-\log(CX(F_{vgg}^{l}(\hat{X}_g),F_{vgg}^{l}(X_g)), \tag{24}$$

where l is the $ReLU\{3\_2, 4\_2\}$ layer of the pretrained VGG19 network.

In the second stage, consisting of the CI-Net, the color feature extracted by the VGG19 model is injected into the grayscale code to control the affine transform parameters in the AdaIN layers. Similar to the losses of the TE-Net, the reconstruction loss, perceptual loss and contextual loss between the generated

color image $\hat{X}_c$ and the target color image X_c are as follows:

$$\mathcal{L}'_{rec} = \left\|\hat{X}_c - X_c\right\|_1, \tag{25}$$

$$\mathcal{L}'_{per} = \frac{1}{N}\sum_{i=1}^{N} \|F^{(i)}_{vgg}(\hat{X}_c) - F^{(i)}_{vgg}(X_c)\|_2, \tag{26}$$

$$\mathcal{L}'_{CX} = -\log(CX(F^l_{vgg}(\hat{X}_c), F^l_{vgg}(X_c)). \tag{27}$$

Let $\mathcal{L}_{GAN}$ be the adversarial loss used to discriminate the triplet $\{X_s, \hat{X}_g, \hat{X}_c\}$ and the real pair $\{X_s, X_g, X_c\}$ with

$$\begin{aligned}\mathcal{L}_{GAN}(G_{SCE}, D_{SCE}) = \mathbb{E}_X[\log D(X_s, X_g, X_c)] \\ + \mathbb{E}_{\hat{X}}[\log(1 - D(X_s, \hat{X}_g, \hat{X}_c)].\end{aligned} \tag{28}$$

The total loss of our SCE model is

$$\begin{aligned}G^*_{SCE} = arg\min_{G_{SCE}}\max_{D_{SCE}} \mathcal{L}_{cGAN}(G_{SCE}, D_{SCE}) \\ + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per} + \lambda_{CX}\mathcal{L}_{CX} \\ + \lambda'_{rec}\mathcal{L}'_{rec} + \lambda'_{per}\mathcal{L}'_{per} + \lambda'_{CX}\mathcal{L}'_{CX}.\end{aligned} \tag{29}$$

4 | EXPERIMENTS

4.1 | Implementation details

We uniformly resize the sketch images in the training set of SketchMan. Specifically, the short edge is set to 256 and the long edge is adaptively resized according to the ratio of the width and height. The NPS and corresponding PS are randomly cropped to 256×256 in the training stage. Generally, when dealing with higher resolution such as 512×512, SSE models will easily fit more short and noisy strokes. Moreover, if dealing with a lower resolution such as 128×128, the synthesized blur sketches will be low-quality. Our SSE and SCE models use the Adam [49] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$.

4.2 | Quantitative metric of SSE

The performances of the SSE models are evaluated by three criteria: the ODS, OIS and AP [50]. The precision/recall curves for the original and refined NPS is shown in Figure 13. Different from the edge detection task [51–54] whose edges are more aligned to the object boundaries than the hand-drawn sketch, which has more offset near boundaries. Therefore, the recall of SSE results is relatively low. As shown in Table 1, compared with the other SSE models, grayscale guided SS has better F-score (ODS=.56, OIS=.56, AP=.34).

Sketch has three important structual elements including points, lines and areas. Therefore, we consider their corresponding feature maps, that is, corner point maps (CPM), straight line maps (SLM), and segmented area maps (SAM), described below.

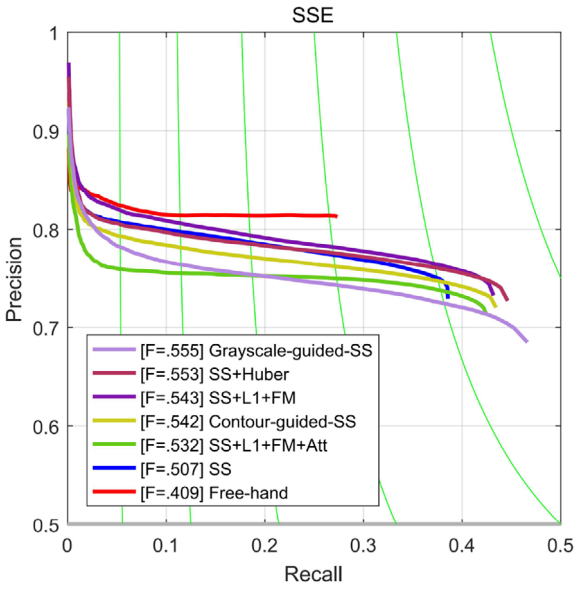

FIGURE 13 Precision/recall curves for NPS and our SSE approaches. Note that we report the indicators of our previous work [28]

TABLE 1 Evaluation results in terms of fixed contour threshold (ODS), per-image best threshold (OIS) and average precision (AP) for different algorithms on the test set. Note that we report the indicators of our previous work [28]

	ODS	OIS	AP
NPS	0.41	0.41	0.22
SS	0.51	0.51	0.30
SS+L1+FM+Att	0.53	0.54	0.32
Contour-guided SS	0.54	0.54	0.33
SS+L1+FM	0.54	0.55	0.34
SS+Huber	0.55	**0.56**	0.34
Grayscale-guided SS	**0.56**	0.56	**0.34**

We use pix2pixHD [11] as our backbone. In order to deal with images with any resolution, our SSE model improve pix2pixHD to a fully convolutional network by using the neural group of upsampling operations and a scale-invariant convolutional layer as the decoder architecture rather than the deconvolutional layers. The training set contains 10,600 NPSs where 2120 initial NPSs are deformed to four groups of augmented NPSs, as illustrated in Figure 12. It should be noted that in our protocol, different deformed NPSs can simulate drawn sketches by different drawer, which are supposed to be mapped to one professional sketch image. There are 132 images in our test set, as the first benchmark for implementing quantitative evaluation of SSE task.

While drawing a sketch, changing the drawing direction on the corner points has a great impact on the professional level of the sketch. We use [55] to detect the corner points according

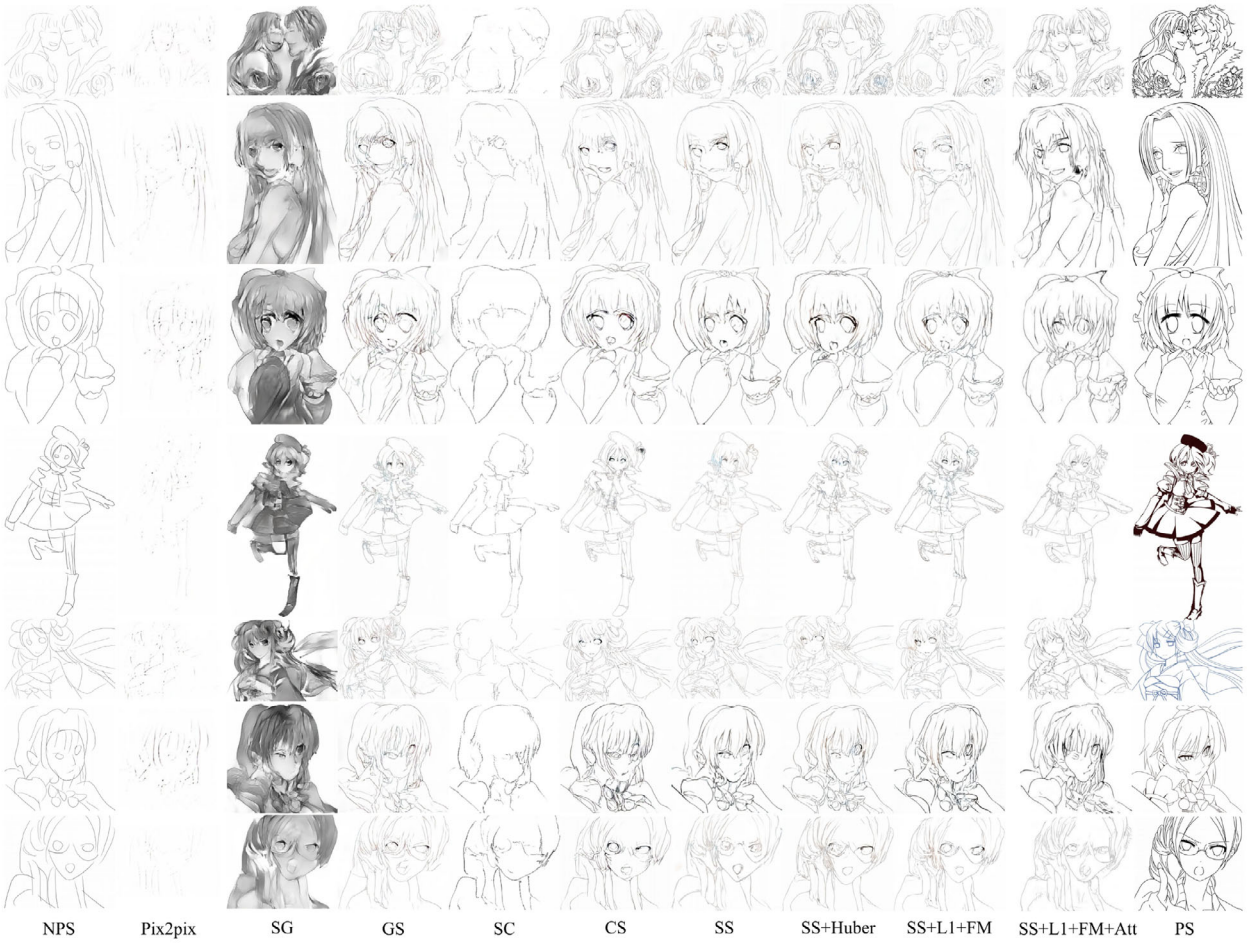

FIGURE 14 Examples of corner points distribution map

to the multi-scale distance D. D10 means the minimum Euler distance of the adjacent corner points is 10 pixels. As shown in Figure 14, D20 indicates the sum of the CPM results of D10 and D20, and the other notation is similar. We locate and extract the straight lines by means of probabilistic Hough transform [56]. As mentioned in Section 3.1.1, the high-resolution original PS images are segmented to 20 superpixel regions by implementing SLIC, to obtain the segmented area map. Note that we filter the redundant segmentation edges of the SLIC results that do not belong to the sketch abstraction, by using the background mask mentioned in Section 3.1.

Inspired by [57], which classifies different strokes into corresponding categories, we use the segmentation metric IOU to conduct the quantitative evaluation based on semantic discrepancy of pixels. Equation (30) shows the proportion of different combinations between real value i and predicted value j under k category. Concretely, p_{ij} represents the amount of pixels with a ground truth label of category i but with a predicted category of j. Therefore, TP, FN, and FP are denoted by p_{ii}, p_{ij}, and p_{ji}, respectively. We set k as 2.

$$IoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ij}-p_{ii}}. \tag{30}$$

The IoU is calculated for both the foreground and background, that is, the canvas and sketch in the SSE task, to represent the proportion of pixels successfully enhanced in the NPS. Note that the mIoU is the mean of these two IoUs.

We separate the sketch image into the background part and the foreground part using two thresholds, that is, 225 and 250. A lower threshold means that there are less details in the sketch. The IOU-based quantitative evaluation results are shown in Table 2, and the qualitative evaluation results are shown in Figure 15.

4.3 | Quantitative metric of SCE

We apply two major quantitative metrics to evaluate the performance of SCE subtask.

4.3.1 | Light sensitivity map

We use the light sensitivity map proposed in [58] to evaluate the colorization performance for the SCE subtask. This score focuses on the ability of autopainting and overfitting to the color hint.

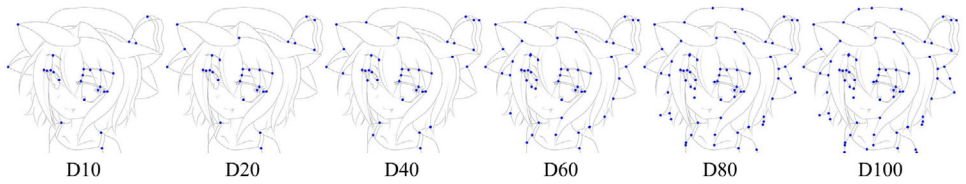

FIGURE 15 Structure enhancement results by means of different SSE methods

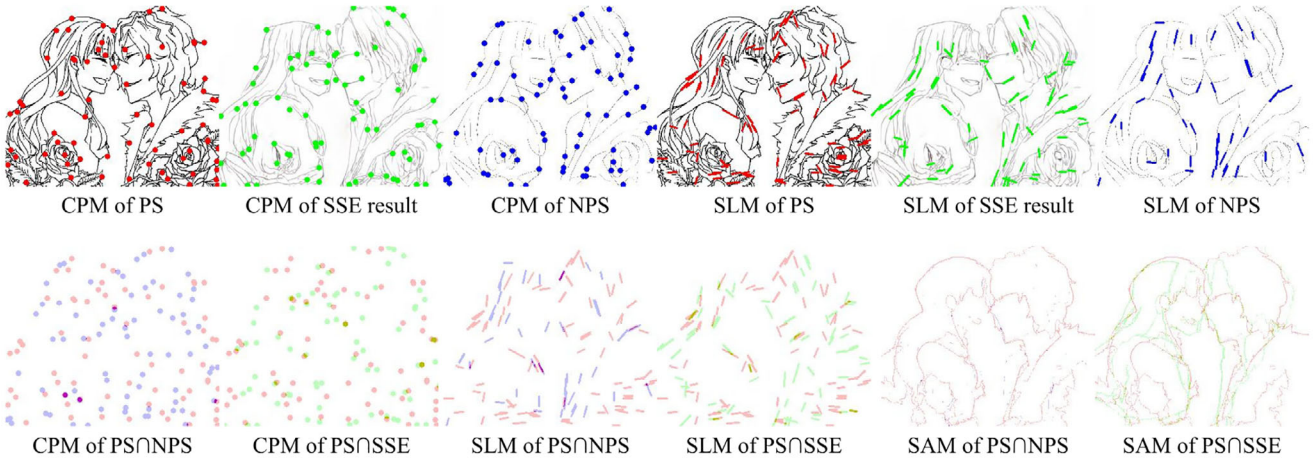

FIGURE 16 The color-coded LBP, B3 and R1 share the same binary value

4.3.2 | Color-coded local binary patterns (CCLBP) map

Furthermore, we use the color-coded local binary patterns (CCLBP) map proposed in [58] to evaluate the colorization performance. It reflects the rendering effect, for example, the smoothness and cleanness of the anime paintings. It is calculated as follows:

$$T = (D_{Rij}^2 + D_{Gij}^2 + D_{Bij}^2)^{1/2}, \tag{31}$$

where D_{Rij} is the distance between the current pixel i and the adjacent pixel j in the R channel. T represents the color difference, which is used to determine the binary value of the corresponding location of pixel j. As shown in Equation (32), for instance, if the distance value T_{R_2} between I(i,j) and I(i − 1, j + 1) is larger than T_{th}, the corresponding binary value R_2 is 1, as shown in Figure 16.

$$R_{c=1:3} = \begin{cases} 1 & T_{R_c} > T_{th} \\ 0 & otherwise \end{cases}. \tag{32}$$

After obtaining the coded binary values, we transfer them to the color space to obtain the CCLBP map based on Equation (33).

$$I_{CCLBP}(R) = 256 \times \frac{(4 \times R_1 + 2 \times R_2 + R_3)}{8}. \tag{33}$$

4.4 | Experimental results

As shown in Table 2, compared with NPS, the sk-IoU value with 250 threshold for the SS task has around 6.5% improvement. As the sketch image naturally has large white areas, it is challenging to enhance the sparse strokes. The contour guided SS approach is also superior over the grayscale guided SS approach, because the grayscale guided SS model usually produces results with more randomly distributed strokes and noises. Overall, the performance of SS+L1+FM has the best IOU performance.

In the test set, SSE models have completed some critical areas of the NPS to some extent, and the structure has been optimized, for example, the head area. However, the overall cleanliness, sketch aesthetics and stroke distribution are still relatively inferior to those of the real PS, especially when it comes to some challenging and complex scenes, for example, Figure 17. Moreover, as shown in Figure 18, the overlaps of the CPM, SLM as well as SAM concerning the SSE results and the original PS are

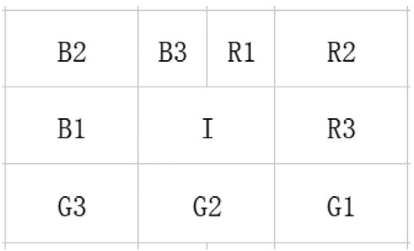

FIGURE 17 Example of SSE of complex scene

TABLE 2 IoU evaluation on diverse domains. The average IOU of CPM, SLM and SAM is denoted as PLA. The best score among six proposed SSE methods is represented using bold indexes. Note that we report the indicators of our previous work [28]

Domain	**Sketch**						**CPM**			**SLM**			**SAM**			**PLA**		
IoU	**mIoU**		**sk-IoU**		**can-IoU**		**mIoU**	**P-IoU**	**can-IoU**	**mIoU**	**L-IoU**	**can-IoU**	**mIoU**	**A-IoU**	**can-IoU**	**mIoU**	**PLA-IoU**	**can-IoU**
Threshold	**225**	**250**	**225**	**250**	**225**	**250**	-	-	-	-	-	-	-	-	-	-	-	-
Free-hand NPS	48.1	48.0	8.7	14.1	87.4	81.9	49.4	5.2	93.6	47.4	5.9	88.9	49.9	2.0	97.8	48.9	4.4	93.4
Pix2pix	45.6	44.4	1.5	5.1	89.7	83.7	48.8	4.1	93.4	48.5	1.6	95.4	49.0	0.8	97.2	48.8	2.2	95.3
Grayscale guided SS	47.7	47.8	10.6	19.1	84.9	76.6	49.0	5.4	92.7	46.5	5.7	87.3	49.0	3.2	94.8	48.2	4.8	91.6
Contour guided SS	48.1	48.9	11.0	19.8	85.2	78.0	49.3	5.9	92.7	46.6	5.9	87.3	49.2	3.4	95.0	48.4	5.1	91.7
SS	48.3	49.1	10.4	**20.5**	**86.2**	77.7	49.4	**6.1**	92.7	46.7	6.0	87.4	49.1	3.4	94.8	48.4	5.2	91.6
SS+SL1	48.2	49.2	11.0	20.4	85.5	78.0	**49.4**	6.1	**92.8**	46.5	6.0	87.3	49.2	3.4	95.0	48.4	5.2	91.7
SS+L1+FM	**48.3**	**49.2**	11.0	19.9	85.6	**78.4**	49.4	6.0	92.8	46.8	**6.2**	87.4	**49.3**	**3.5**	95.0	48.5	**5.2**	91.7
SS+L1+FM+Att	48.1	48.6	**11.4**	19.5	84.7	77.8	48.9	5.2	92.6	**48.9**	4.8	**93.0**	49.2	3.3	**95.2**	**49.0**	4.4	**93.6**

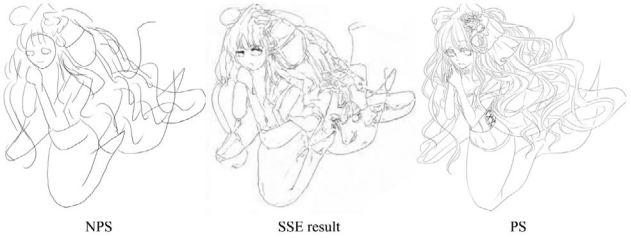

FIGURE 18 Overlap of the corner points, straight lines and segmented areas considering the real PS (red), generated SSE results (green) and free-hand NPS (blue)

TABLE 3 Quantitative comparison with PaintsChainer [26], Style2Paints [14], DeepColor [27] and Pix2pix [10] based on L1 scores of CCLBP and light-sensitivity

Methods	**LS** ↑	$CCLBP_5$ ↑	$CCLBP_{10}$ ↑	$CCLBP_{15}$ ↑
PaintsChainer1	−1.316	0.687	2.277	1.844
PaintsChainer2	−1.092	−4.150	−1.074	−0.660
PaintsChainer3	**−0.379**	−6.813	−6.687	−6.179
Style2Paints v3	−2.339	−3.819	−2.931	−2.529
DeepColor	−3.451	−8.019	−6.288	−5.982
Pix2pix	−0.588	−12.717	−5.626	−2.429
Ours	−0.384	**1.156**	**3.072**	**1.899**

still sparse. Generally, the more the overlap, the more professional the predicted PS is.

For the SCE task, a quantitative comparison with PaintsChainer [26], Style2Paints [14], DeepColor [27] and Pix2pix [10] is shown in Table 3. Our model exhibits competitive performance compared with other state-of-the-art methods. We use the test set in [58] to evaluate the autonomous sketch colorization in this paper, and some comparative examples are shown in Figure 19. More SCE results based on various reference images are shown in Figure 20.

4.5 | Subjective evaluation

4.5.1 | User study of the SSE task

We conduct the subjective evaluation of the SSE task considering sketch professionalism and AI forgery detection, described as follows.

We invited 20 people where half of them are anime amateurs and the others are artists. After briefly introducing the SSE task, these 20 users are asked to judge the real PS images in terms of its performance with respect to (a) drawing aesthetics, (b) the sketch completeness, (c) line smoothness and (d) noise amount. There were 924 images to be displayed, and each participant observed 500 random images. We randomly show a SSE image, which is scored 1–5 on the basis of the above four metrics. We collected a total of 10,000 human judgments, and the average subjective evaluation results are shown

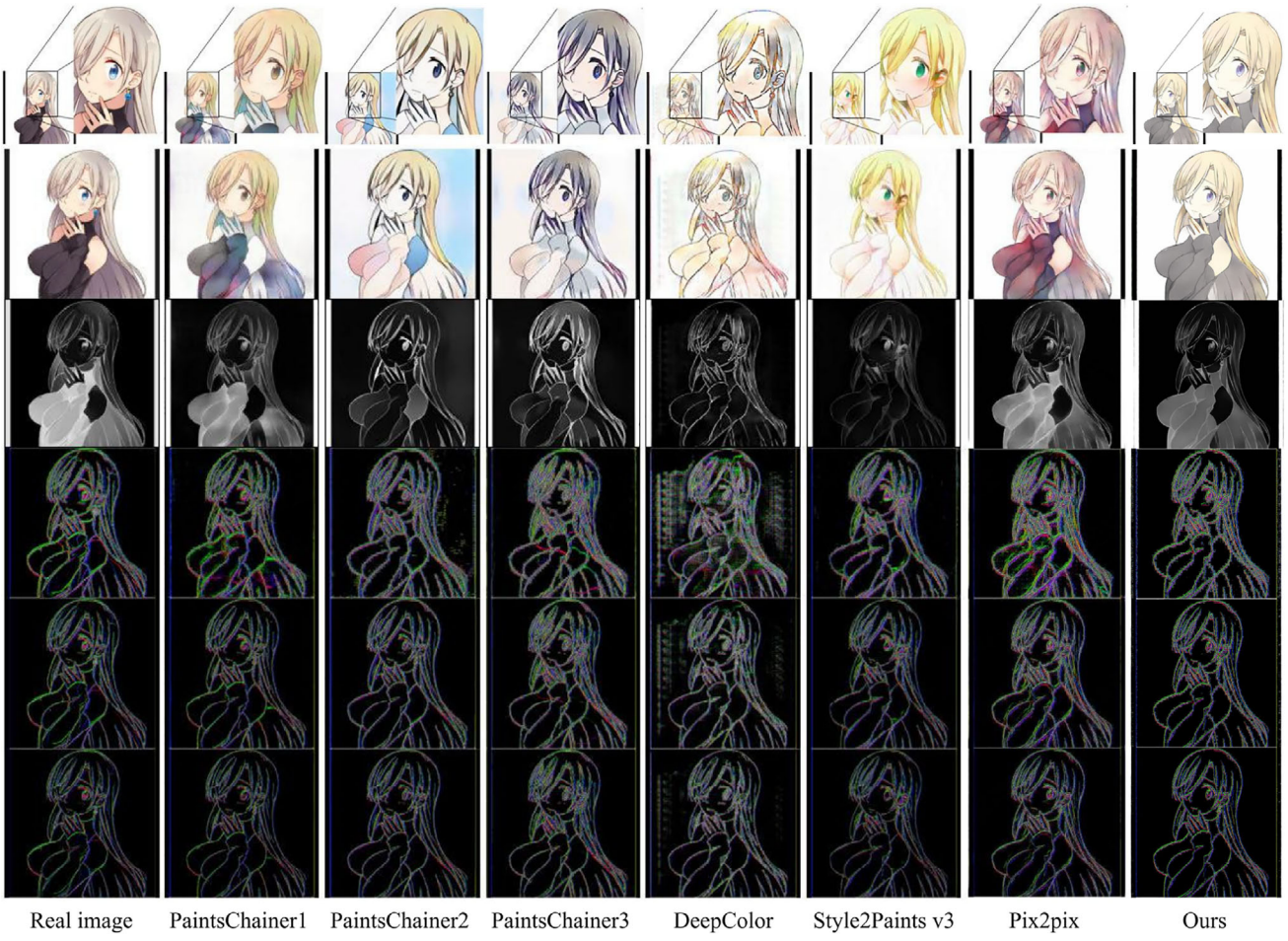

FIGURE 19 Some examples of autonomous sketch colorization results (row 2), the corresponding LS maps (row 3) and the multiscale CCLBP maps ($T_{th} = 5$ (row 4), 10 (row 5), 15 (row 6))

FIGURE 20 More SCE results based on various reference images

TABLE 4 Subjective evaluation results of SSE task

	a ↑	b ↑	c ↑	d ↓	Overall ↑
NPS	1.9	1.8	3.2	0.4	1.6
Pix2pix	0.3	0.4	0.3	0.7	0.1
SS	3.4	**3.8**	3.2	2.5	2.0
SS+SL1	2.8	3.3	3.0	2.9	1.5
SS+L1+FM	3.2	3.4	3.0	2.7	1.7
SS+L1+FM+Att	2.9	3.6	3.3	**3.2**	1.7
grayscale guided SS	2.6	3.1	2.9	3.1	1.4
contour guided SS	**3.5**	3.5	**3.6**	2.0	**2.2**

TABLE 5 Average scores of our user study for SCE task

Methods	a	b	c	d	Mean
PaintsChainer1	5.3	4.2	5.9	6.9	5.6
PaintsChainer2	6.1	4.1	**7.2**	**7.3**	6.2
PaintsChainer3	6.6	4.8	5.9	7.0	6.1
Style2Paints v3	7.0	5.5	5.1	4.9	5.6
DeepColor	3.4	3.6	5.1	5.7	4.5
Pix2pix	6.0	5.5	5.8	5.8	5.8
Ours	**7.1**	**5.9**	6.9	6.9	**6.7**

in Table 4, where $V_{Overall} = \frac{1}{4}(V_a + V_b + V_c - V_d)$. Compared with the other proposed baselines, contour guided SS has a better perceptual performance.

As for AI forgery detection, the 20 users are invited to discriminate the generated fake sketch and real sketch in Pixiv. We found that human observation could detect only 5% of the forgery drawings, which demonstrates that SSE algorithms need to be improved further.

4.5.2 | User study of the SCE task

We implemented a user study of the SCE task based on four criteria: (a) the overall color visual effect, (b) regional obedience, (c) local rendering purity, and (d) colorization completion degree [58]. There were a total of 1400 generated color images, and each participant needed to randomly observe 500 images. The results are shown in Table 5, and our method has better performance than the others.

5 | CONCLUSION

We performed DCSC based on disentangling the structure and color enhancement. SSE task mainly contains three difficulties: (1) randomly distributed lines are difficult to optimize based on the model priors, (2) the large amount of blank background makes the semantic of a large scale of areas ambiguous, and (3) the structural feature of an NPS has serious distortions. To perform this challenging task, we collect plenty of anime characters and draw the corresponding NPS in SketchMan. Moreover, we explored three different pipelines, that is, SS, grayscale guided SS and contour guided SS. We have established an important benchmark for the SSE task. We recommend this challenging and meaningful issue to the research community, hoping to attract more attention. Considering the quality of SSE results is still low, SSE and SCE are independently trained in this paper. The SSE is supposed to be improved in the future. The data that support the findings of this study are openly available in SketchMan2020 at [59].

For the SCE task, the first-stage TE-Net is trained to generate the grayscale map of the sketch representing the topology completion result, on the basis of which the second-stage CI-Net achieves diverse color modification based on a specific reference map. In this way, both the topological and color features are enhanced to generate more controllable and detailed color maps. Furthermore, the color diversity and regional obedience of the created sketches need to be improved.

ACKNOWLEDGEMENTS

This work was supported by National Key Research and Development Plan (No. YS2018YFB1403703) as well as Research Project of the Communication University of China (No.CUC200D058).

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in SketchMan2020 at https://github.com/LCXCUC/SketchMan2020 [59].

ORCID

Nan Gao https://orcid.org/0000-0002-8277-4100

REFERENCES

1. Yu, Q., Yang, Y., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-Net: A deep neural network that beats humans. Int. J. Comput. Vis. 122(3), 411–425 (2017). https://doi.org/10.1007/s11263-016-0932-3
2. Chen, W., Hays, J.: SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In: CVPR, pp. 9416–9425. IEEE, Piscataway (2018)
3. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch Me That Shoe. In: CVPR, pp. 799–807. IEEE, Piscataway (2016)
4. Riaz Muhammad, U., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.M.: Learning Deep Sketch Abstraction. In: CVPR, pp. 8014–8023. IEEE, Piscataway (2018)
5. Liu, F., Deng, X., Lai, Y.K., Liu, Y.J., Ma, C., Wang, H.: SketchGAN: Joint Sketch Completion and Recognition With Generative Adversarial Network. In: CVPR, pp. 5830–5839. IEEE, Piscataway (2019)
6. Ha, D., Eck, D.: A Neural Representation of Sketch Drawings. Paper presented at ICLR, Vancouver, BC, Canada, 30 April–3 May 2018. https://openreview.net/forum?id=Hy6GHpkCW
7. Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: SketchX!-Shoe/Chair fine-grained SBIR dataset (2011). http://sketchx.eecs.qmul.ac.uk
8. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. (Proc SIGGRAPH) 31(4) 44:1–44:10 (2012)

9. Romain, T.: Amazing drawings (2017). https://www.pinterest.ca/darrenrawlings/thomas-romain/
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-Image translation with conditional adversarial networks. In: CVPR, pp. 1125–1134. IEEE, Piscataway (2017)
11. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In: CVPR, pp. 8798–8807. IEEE, Piscataway (2018)
12. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In: ICCV, pp. 2242–2251. IEEE, Piscataway (2017)
13. Kim, J., Kim, M., Kang, H., Lee, K.H.: U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation. Paper presented at ICLR, Addis Ababa, Ethiopia, 26–30 April 2020. https://openreview.net/forum?id=BJlZ5ySKPH
14. Zhang, L., Li, C., Wong, T.T., Ji, Y., Liu, C.: Two-stage sketch colorization. ACM TOG. 37(6) 261:1–261:14 (2018)
15. Yu, J., Lin, Z.L., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative Image Inpainting with Contextual Attention. In: CVPR, pp. 5505–5514. IEEE, Piscataway (2018)
16. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution. In: European Conference on Computer Vision, vol. 8692, pp. 184–199. Springer, Berlin (2014)
17. Hinz, T., Heinrich, S., Wermter, S.: Generating Multiple Objects at Spatially Distinct Locations. In: ICLR, New Orleans, Louisiana, United States (2019) Available from: https://openreview.net/forum?id=H1edIiA9KQ
18. Johnson, J.E., Gupta, A., Fei-Fei, L.: Image Generation from Scene Graphs. In: CVPR, pp. 1219–1228. IEEE, Piscataway (2018)
19. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies. ACM TOG 35(4), 119:1–119:12 (2016). http://doi.acm.org/10.1145/2897824.2925954
20. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: European Conference on Computer Vision (ECCV), pp. 172–189. Springer, Berlin (2018)
21. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR, pp. 4401–4410. (2019)
22. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., et al.: Few-shot unsupervised image-to-image translation. In: ICCV, pp. 10551–10560. IEEE, Piscataway (2019)
23. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: ICCV, pp. 9459–9468. IEEE, Piscataway (2019)
24. Duan, B., Fu, C., Li, Y., Song, X., He, R.: Cross-Spectral Face Hallucination via Disentangling Independent Factors. In: CVPR, pp. 7930–7938. IEEE, Piscataway (2020)
25. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV, pp. 1501–1510. IEEE, Piscataway (2017)
26. Yonetsuji, T.: PaintsChainer (2017). https://petalica-paint.pixiv.dev/index_zh.html
27. Frans, K.: Outline Colorization through Tandem Adversarial Networks. CoRR. abs/1704.08834 (2017). http://arxiv.org/abs/1704.08834
28. Li, J., Gao, N., Shen, T., Zhang, W., Mei, T., Ren, H.: SketchMan: Learning to Create Professional Sketches. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 3237–3245. ACM, New York (2020)
29. Pang, K., Li, D., Song, J., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep factorised inverse-sketching. In: ECCV, Munich, pp. 36–52. Springer, Berlin (2018)
30. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling Deep Image Synthesis with Sketch and Color. In: CVPR, pp. 5400–5409. IEEE, Piscataway (2017)
31. Furusawa, C., Hiroshiba, K., Ogaki, K., Odagiri, Y.: Comicolorization: Semi-automatic Manga Colorization. In: SIGGRAPH Asia 2017 Technical Briefs, pp. 12:1–12:4. ACM, New York (2017). http://doi.acm.org/10.1145/3145749.3149430
32. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative Adversarial Nets. In: NIPS, pp. 2672–2680. Curran Associates, Inc., Red Hook (2014). http://dl.acm.org/citation.cfm?id=2969033.2969125
33. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv 14111784 (2014)
34. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR, pp. 8110–8119. IEEE, Piscataway (2020)
35. Chen, R., Chen, X., Ni, B., Ge, Y.: SimSwap: An Efficient Framework For High Fidelity Face Swapping. In: MM '20: The 28th ACM International Conference on Multimedia. ACM, New York (2020)
36. Li, L., Bao, J., Yang, H., Chen, D., Wen, F.: Advancing High Fidelity Identity Swapping for Forgery Detection. In: CVPR. IEEE, Piscataway (2020)
37. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic Image Synthesis With Spatially-Adaptive Normalization. In: CVPR. IEEE, Piscataway (2019)
38. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. 31(11), 1955–1967 (2009). https://doi.org/10.1109/TPAMI.2008.222
39. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., et al.: SketchyScene: Richly-annotated scene sketches. In: ECCV, pp. 421–436. Springer, Berlin (2018)
40. jerryli27.: Pixiv dataset (2017). https://github.com/jerryli27/pixiv_dataset
41. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC Superpixels (2010)
42. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. In: ACM SIGGRAPH, pp. 533–540. ACM, New York (2006)
43. Zhou Wang., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13(4), 600–612 (2004)
44. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint, arXiv:180911096 (2018)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is All you Need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., Red Hook (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
46. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 9351, pp. 234–241. Springer, Cham (2015) (available on arXiv:1505.04597 [cs.CV]) Available from: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a
47. Lllyasviel.: sketchKeras (2017) https://github.com/lllyasviel/sketchKeras
48. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The Contextual Loss for Image Transformation with Non-Aligned Data. In: ECCV. Springer, Berlin (2018)
49. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. Springer (2015)
50. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33(5), 898–916 (2010)
51. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV, pp. 1395–1403. IEEE, Piscataway (2015)
52. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV, pp. 1841–1848. IEEE, Piscataway (2013)
53. Shen, W., Wang, X., Wang, Y., Bai, X., Zhang, Z.: Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In: CVPR, pp. 3982–3991. IEEE, Piscataway (2015)
54. Bertasius, G., Shi, J., Torresani, L.: Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In: CVPR, pp. 4380–4389. IEEE, Piscataway (2015)
55. Shi, J., et al.: Good features to track. In: CVPR, pp. 593–600. IEEE, Piscataway (1994)

56. Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic hough transform. Patt. Recogn. 24(4), 303–316 (1991). https://doi.org/10.1016/0031-3203(91)90073-E
57. Yang, L., Zhuang, J.: SketchGCN: Semantic Sketch Segmentation with Graph Convolutional Networks. arXiv preprint, arXiv:200300678 (2020)
58. Ren, H., Li, J., Gao, N.: Two-Stage sketch colorization with color parsing. IEEE Access 8, 44599–44610 (2020)
59. LCXCUC: SketchMan2020 (2020). https://github.com/LCXCUC/SketchMan2020

How to cite this article: Gao, N., Ren, H., Li, J., Su, Z.: Disentangled and controllable sketch creation based on disentangling the structure and color enhancement. IET Image Process. 1–16 (2021). https://doi.org/10.1049/ipr2.12343