DOI: 10.1111/cgf.14797

COMPUTER GRAPHICS *forum*
Volume 42 (2023), number 6, e14797

Adversarial Interactive Cartoon Sketch Colourization with Texture Constraint and Auxiliary Auto-Encoder

Xiaoyu Liu,[1,2] Shaoqiang Zhu,[1,2] Yao Zeng[1,3] and Junsong Zhang[1,3]

[1]Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen, China
zhangjs@xmu.edu.cn
[2]NERCEL, Central China Normal University, Wuhan, China
[3]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen University, Xiamen, China

Abstract

Colouring cartoon sketches can help children develop their intellect and inspire their artistic creativity. Unlike photo colourization or anime line art colourization, cartoon sketch colourization is challenging due to the scarcity of texture information and the irregularity of the line structure, which is mainly reflected in the phenomenon of colour-bleeding artifacts in generated images. We propose a colourization approach for cartoon sketches, which takes both sketches and colour hints as inputs to produce impressive images. To solve the problem of colour-bleeding artifacts, we propose a multi-discriminator colourization framework that introduces a texture discriminator in the conditional generative adversarial network (cGAN). Then we combined this framework with a pre-trained auxiliary auto-encoder, where an auxiliary feature loss is designed to further improve colour quality, and a condition input is introduced to increase the generalization ability over hand-drawn sketches. We present both quantitative and qualitative evaluations, which prove the effectiveness of our proposed method. We test our method on sketches of varying complexity and structure, then build an interactive programme based on our model for user study. Experimental results demonstrate that the method generates natural and consistent colour images in real time from sketches drawn by non-professionals.

Keywords: Image and video processing

CCS Concepts: • Computing methodologies → Image generation; Neural networks; Computer vision

1. Introduction

Cartoon colourization can be one of the most straightforward ways to develop the intelligence, imagination and creativity of little children. Different from common anime sketches or photo sketches, cartoon sketches have two main characteristics: (1) simple line structure with less edge and semantic information. Cartoon sketches, which usually use a simple line to describe a complicated object, strip away most of the details and retain only general structural features. (2) Cartoon sketches drawn by non-professionals may be more irregular and abstract with various changes. These features make the task of interactive cartoon sketch colourization challenging, it is mainly reflected in the low quality results with colour-bleeding artifacts and inconsistent colours when colouring various cartoon sketches.

In computer graphics to achieve interactive colourization of sketches, early Lazybrush [SDC09] achieved perfect colour filling with strict area control and accurate colour propagation. However, such a method based on traditional interactive image segmentation cannot generate details from sketches, such as the colour gradient transition and shadow effect in blank areas of sketches. Likewise, this method, requiring users to provide colour scribble for every influential area to be coloured, undoubtedly creates a huge workload for users.

Inspired by deep generative model, especially the generative adversarial networks (GANs) [GPAM*14], most research works have explored learning-based interactive colourization in a variety of domains, including anime [ZLW*18, YSS21, KJPY19, CMW*18], art [LSZE20], photo [SLF*17] and scene [GLX*20, ZMG*19].

Sketches in these works usually have suitable line composition and contain plenty of structural information that can be used to paint a satisfactory visual effect. In some researches, however, some colour-bleeding artifacts also appear along the edges. For instance, the commercial product, PaintsChainer [Yon17], provides three methods (Tanpopo, Satsuki and Canna), which generate three different styles of images for amine line arts with scribble colours as hints. However, when cartoon sketches of children are provided sparser lines, results usually show plenty of colour-bleeding artifacts that lose the realistic texture of real colour images and are difficult for the user to control. To improve the generalization ability in various sketches, Ci *et al.* [CMW*18] introduce a local feature network upon a very deep conditional generative adversarial network (cGAN), thereby achieving high-quality sketch colourization and overcoming overfitting when synthesizing sketches. However, some issues still remain in terms of colour coherency with user hints and harmonious colour results. Although Style2paints [lll18], as a mature software, produces stable high-quality images with relatively consistent colours from what a user specifies, the results may generate mixed colour artifacts along the image boundaries.

Compared with the above works, cartoon sketches contain **much** simpler line structure with less content information; thus, their colouring results are more prone to colour artifacts in large blank areas of the sketches. Likewise, also a challenge for colourization is the generalization ability in different sketches that is affected by the irregular characteristics of cartoon sketches.

As we know, the shadow, material and texture information of greyscales help the greyscale image colourization models produce more realistic and natural colour images. In our previous work on anime sketch colourization, the proposed UGSC-GAN [ZZLL21], which contains two networks, the greyscale generator network and the colour generation network, utilizes the greyscale image generated by the first network to provide supplementary texture information to generate colour image in the second stage. Nonetheless, the greyscale image generation process cannot be controlled by user hints, and the quality of generated greyscale image will directly affect the results of interactive colouring.

In this paper, we designed a colourization model of cartoon sketches with colour hints. We consider that a realistic greyscale image provides profitable texture information, that is, the grey distribution of the pixel and its surrounding spatial neighbourhood pixels, which can be used as a supplementary condition for our sketch colourization. In order to apply grey information reasonably and make the generated colours guided by colour hints is not affected by the grey value and the greyscale image quality like previous work [ZZLL21]. We adopted a texture discriminator to assist our structure discriminator. As we know, greyscale images do not have colour information, but retain pixel intensities that sketches do not have, taking the greyscale image as a constraint, designing a texture discriminator can help the discriminators better distinguish the authenticity, thereby guiding the generator to generate an appropriate greyscale distribution similar to greyscale image, and the relationship between pixels will be denser and more accurate. For the cartoon sketch colourization, it can keep the colouring smooth and free of artifacts, especially at the edges and large blank areas. In addition, we combined the multi-discriminator with a pre-trained auxiliary auto-encoder (AE). Thus, both the texture and structure discriminator are conditioned on the intermediate representation of AE to strengthen colouring ability in different types of cartoon sketches. To further improve image quality, especially in colour harmony, we also designed an auxiliary feature loss based on auxiliary AE.

To evaluate our approach, we implement ablative studies of our proposed module, compare it with existing methods, and demonstrate the efficiency of our approach in a variety of complex sketches, which include simple non-professional hand-drawn, sketches in the Sketchy [SBHH16] dataset and our synthetic sketches. Similarly, we also develop a user interactive colourization programme used in our two-stage user study. Experiments show that our proposed colourization method achieves superior performance. Our contributions are summarized as follows:

- We propose a cartoon sketch colourization task with colour hints and develop a user interactive programme for real-time colourization with high practically.
- To solve colour-bleeding artifacts, we propose a texture discriminator to help reconstruct pixel greyscale. The combination of texture discriminator and structure discriminator effectively improves colouring quality.
- We employ an auxiliary AE for feature-level generative supervision, which further improves the quality of the generated images.

2. Related Work

2.1. Traditional interactive colourization

The potential use of interactive sketch colourization has encouraged a new research direction in the field of sketch synthesis. Prior interactive colourization methods strongly rely on low-level similarity metrics to propagate scribble or colour hint. Levin *et al.* [LLW04] assumed that adjacent pixels with similar luminance should have a similar colour. Qu *et al.* [QWH06] proposed colourization methods based on texture and gradient information existing in the photos. The famous Lazybrush of Sýkora *et al.* [SDC09] achieved perfect colour filling with strict area control and accurate colour propagation. However, these methods focus on the strokes where local control cannot generate more details from sketches, like colour gradient transition and shadow effect in the blank areas of sketches. Also, these methods usually require users to provide colour scribbles for every influential area to be coloured, which undoubtedly creates a huge workload for users.

2.2. Learning-based interactive colourization

In recent years, many learning-based colourization methods, which are mainly divided into greyscale and sketch colourization, have been proposed. Greyscale colourization [KLP*21, XWF*20, HCL*18] has been studied to create very realistic colour images with varying degrees of exploration in overcoming colour-bleeding. For example, Su *et al.* [SCH20] proposed an instance-aware approach to extract image features at an instance and full-image level and then fuse them to predict the final colour image. Kim *et al.* [KLP*21] proposed a special edge enhancement framework where users can correct colour-bleeding effects through strokes. Unlike greyscale colourization, which relies heavily on the texture and

gradient information of greyscale, sketch colourization, requiring synthesizing colour images from sketches with little content information, is more prone to noticeable colour bleeding and artifacts. Our work focuses on using greyscale information to assist our sketch colourization to overcome colour-bleeding artifacts.

Not only must interactive sketch colourization automatically synthesize colour images. The synthesized images must comply with user-specified colour conditions, including natural language [ZMG*19], text [KJPY19], reference images [LSZE20, LKL*20, CZG*20, ZJLL17], colour scribbles [SLF*17, ZLSS*21], colour points [ZZI*17, CMW*18, ZLW*18] *etc*. To guide the colouring of scene-level drawings of children, Zou *et al.* [ZMG*19] used natural language as conditional colouring information. Using text labels, Kim *et al.* [KJPY19] guided the network to paint specific colours in the corresponding regions of objects. Because these colouring conditions are relatively complex, a user must not only specify the colour, but also provide the corresponding position or colouring object. Paired data are difficult to collect, and it is difficult to establish dense semantic correspondence between sketches and reference images with large information differences. Consequently, reference image-based sketch colourization is still under-explored. Lee *et al.* [LKL*20] used an image with geometric distortion as a visual reference image and designed an attention mechanism to accomplish colour transfer. Chen *et al.* [CZG*20] used an active-learning-based framework to colourize a set of images with a single reference image. Liu *et al.* [LSZE20] through a pre-trained feature extraction network achieved artistic style sketch colourization by transferring the reference image style to sketch. These works require that users only provide images as references, which cannot give users good interactive experiences. Therefore, colour hints based on colour scribbles or colour points are more suitable for cartoon sketch colourization for children. Furthermore, colour points may be more friendly for users in experimental observations.

Several solutions exist for colouring sketches with colour hints. Sangkloy *et al.* [SLF*17] learned to generate realistic results from sketches of specific categories based on the scribble of users. Yonetsuji [Yon17] provided three models (Tanpopo, Satsuki and Canna), and realized three styles of colourization. Unfortunately, their results usually contain plenty of colour-bleeding artifacts with little practicability. Ci *et al.* [CMW*18], who achieved impressive colouring results by a very deep network, improved the generalization ability of the network on sketches by using a local feature network. However, unreliable textures of random locations and blurred, colour-bleeding artifacts along edges, as well as colour dissonance and colour inconsistency, may occur in the above methods. Zhang *et al.* [ZLW*18] divided the colourization problem into two stages: (1) a network to obtain a colour draft and (2) colour hints to make adjustment to obtain impressive results. Their style2paintsV4.5 [lll18] achieves better colouring quality. However, the colouring result commonly contains more saturated colours than colour hints and is obviously time-consuming, which makes it inconvenient for users to adjust colours in real time. The above methods tend to reconstruct pixel-level texture by network learning from a training dataset. To a certain extent, the occurrence of colour-bleeding artifacts cannot be avoided. We use the greyscale information of a greyscale image to prompt our cGAN to focus on texture constraint and use an auxiliary AE for further quality improvement. Our generator realizes real-time colourization with higher practicality.

3. Method

Given a colour hint, our network automatically generates a colourized image of a cartoon sketch, in which colour references the colour hint, and the structure depends on the sketch. An overview of our approach is shown in Figure 2.

3.1. Data preparation

During the data preparation phase, using the Baidu image search engine, we collected from the website 15,000 colour images containing different categories (e.g. plant, animal and transportation) We trained and evaluated our method on the self-made cartoon dataset, where, along with simulated colour hints, we synthesized the sketch and greyscale for each cartoon colour image. Specifically, to obtain sketches paired with I_{gt}, we adopted the sketch extraction method, XDoG [WKO12]. For colour hints, we adopted the approach of Ci *et al.* [CMW*18]. The locations of the points were determined by a binary mask, $I_{mask} = R > | \xi |$ where $R \in 1C \times H \times W$ and $\forall r \in R, r \sim U(0,1), \xi \sim N(1, 0.005)$. The tensors form colour hint $I_{hint} = \{I_{gt} \times I_{mask}, I_{mask} \in 4C \times H \times W$. In addition, a greyscale image was generated automatically during data loading. It is an auxiliary condition that guides our generator to produce a high-quality image with a clear texture (see Section 3.2 for details). Figure 1 shows the pipeline of data processing.

3.2. Network architecture

The main aim of our work is to generate high-quality cartoon images from sketches. As shown in Figure 2, our model consists of three parts: a generator (*Gen*), two discriminators (Dis_s, Dis_g) and a separately pre-trained auxiliary AE.

3.2.1. *Generative network*

Our generator architecture is based on U-Net [RFB15] and Efficientnet [TL19]. Specifically, to generate colours with quality details, *Gen* adopts encoder and decoder structures with skip-connection aiming to preserve low-level feature information. For the encoder, we used the Efficientnet without using the final pooling operation and FC layer. Some current sketch colouring methods improve the quality of generated images, while increasing the complexity of the models, which greatly limits the efficiency and practicability of the colouring systems. Efficientnet corresponds to a model scaling method that significantly improves model efficiency, making our model smaller and faster. While SEnet [HSS18] applied to MBConvBlock [HZC*17] of Efficientnet, which emphasizes improving the receptive field on the feature dimension with less cost and higher model performance, helps generate colours with overall quality.

As illustrated in Figure 2, the colour hint (I_{hint}) is first concatenated with the sketch (I_{sketch}) as the input of *Gen* and then transformed to the feature map via the encoder module. We **obtain** the intermediate representation (the grey rectangle of *Gen* in Figure 2), which is then upsampled with five transposed convolutional layers to produce the final colourized image. Specifically, we used batch normalizations and ReLU activations for each convolutional layer, except for the last convolutional layer that uses tanh activation.

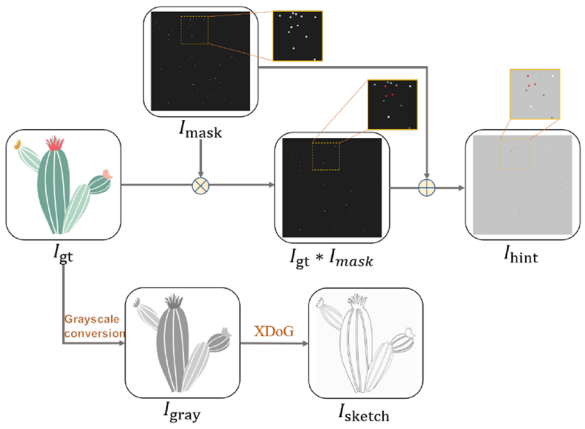


Figure 1: *Data processing. Colour hint (I_{hint}) is simulated from I_{gt} by the binary mask (I_{mask}). I_{gt} is converted to greyscale (I_{grey}), and I_{sketch} is extracted by using XDoG sketch extraction.*

3.2.2. *Discriminator network*

To further improve colourized image quality in the generator architecture, we introduced a multi-discriminator framework. As illustrated in Figure 2, by removing Dis_g, our architecture is similar to Pix2pix [IZZE17], which adds additional sketches to both Dis_s and *Gen* as input. Here we named the first discriminator as the structure discriminator. To be able to fool the Dis_s, the images generated by *Gen* must be sufficiently realistic and match the sketch structure. For the second discriminator (Dis_g) shown in Figure 2, we used the greyscale image as the additional 'texture' input, which provides not only edge information, but also implicitly reserves 'region category' information (grey value of each pixel, regional texture, *etc*.). Therefore, in addition to obtaining the relationship between I_{sketch} and I_{gen} or I_{sketch} and I_{gt}, our discriminator also obtains the relationship information between I_{grey} and I_{gen} or I_{grey} and I_{gt}. The additional 'texture' condition enhances the attention of the discriminator to the texture and details of the corresponding region, and also encourages the *Gen* to generate images that match the information of the greyscale images, which greatly improves image quality, especially on solving the problem of colour-bleeding artifacts.

3.2.3. *Auxiliary auto-encoder*

Our auxiliary AE consists of the encoder and decoder. The architecture of our auxiliary AE is the same as that for *Gen*. Different from *Gen*, the task of the auxiliary network is to achieve colour image reconstruction. During the training of *Gen* and *Dis*, auxiliary AE is fixed and provides for intermediate representations ($40C \times 28H \times 28W$) and the final encoded feature map ($1280C \times 7H \times 7W$) (the grey rectangle in Figure 2).

Zhang *et al.* [ZJLL17], in their anime sketches style transfer work, mention that the output quality of a cGAN-based network depends on the degree of information gap between the input and output. The conditional discriminator leads the generator to focus excessively on the relationship between the sketch and the colour image, and to some extent ignores the composition of the colour image, resulting in inevitable overfitting. Therefore, our hope is that the auxiliary AE provides fine-grained level information of a colour image to the conditional discriminator to alleviate overfitting. We argue that the encoded feature map carries limited information about image details. Thus, it is more effective to use an intermediate representation ($40C \times 28H \times 28W$) of our auxiliary AE as the condition for input of Dis_g and Dis_s for an information supply (see illustration in Figure 2).

The auxiliary AE encodes the low-dimensional input into a high-level latent code and forces the network to learn the most

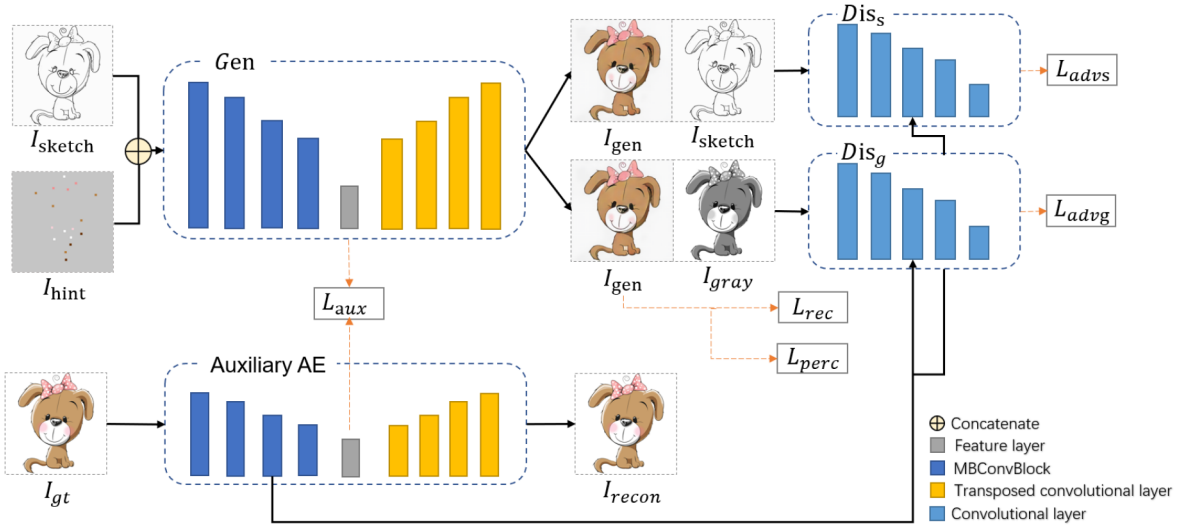


Figure 2: *Overview of our proposed approach. The colourization model consists of a generator (Gen), two discriminators (Dis_s and Dis_g) and an auxiliary auto-encoder. The input of Gen is the sketch (I_{sketch}) and the colour hint (I_{hint}). Output of the model is I_{gen}. The white box with the orange dashed arrow pointing to it represents our loss function.*

informative features. We applied the auxiliary AE to refine the intermediate representations of our generator, constrain *Gen* to embed the input sketch into high-dimensional latent space, and represent the colour information for better colouring. Specifically, to further improve generated image quality, especially in colour harmony and colour coherency, we designed an auxiliary feature loss by computing the distance between the final encoded feature map ($1280C \times 7H \times 7W$) of AE and our encoded feature map (details are given in Section 3.3).

3.3. Loss function

Auxiliary feature loss: There is a large information gap between sketches and colour images, making it very difficult to map from sketch domain to colour image domain. Therefore, we introduce an auxiliary feature loss to encourage our *Gen* to produce impressive results with reasonable constraint from the features of the auxiliary AE. Our auxiliary feature loss promotes the sketch-hint feature space to coincide with the corresponding colour image feature space as much as possible and helps *Gen* extract useful features more efficiently. Specifically, we take the final encoded feature map from auxiliary AE and calculate the distance between it and the corresponding feature of the generator as the auxiliary feature loss. The loss is defined as follows:

$$L_{aux} = E_{s,c,u\sim P_{data}(s,c,u)}[||f_{gen}(s,u) - f_{aux}(c)||_1] \quad (1)$$

where $f_{gen}(s,u)$ denotes the intermediate representations of our *Gen*, $f_{aux}(c)$ denotes the final encoded feature map of the auxiliary pretrained AE. Sketches, colour hints and colour images are represented by s, u and c, respectively.

Reconstruction loss: Reconstruction loss penalizes the network for the difference between a generated image and ground truth. Reconstruction loss is essential here as it forces the network to be more precise with colour by paying more attention to a colour hint. Reconstruction loss is defined as follows:

$$L_{rec} = E_{s,c,u\sim P_{data}(s,c,u)}[||Gen(s,u) - c||_1] \quad (2)$$

where $Gen(s,u)$ denotes the generated image. Sketches, colour hints and colour images are represented by s, u and c, respectively.

Perceptual loss: To encourage our network to produce a convincing image that conforms to human perception, perceptual loss [JAFF16] penalizes the model to bridge the semantic gap by sending a generated image and ground truth to the pre-trained model VGG-16 [SZ14] in ImageNet [DDS*09]. Perceptual loss helps the model obtain corresponding low-high feature maps from different convolutional layers. Perceptual loss is defined as follows:

$$L_{perc} = E_{c_{gen},c\sim P_{data}(c_{gen},c)}\left[\sum_{i=1}^{L}\frac{1}{N_i}||\phi_i(c) - \phi_i(c_{gen})||_1\right] \quad (3)$$

where c denotes ground truth, c_{gen} denotes a generated image, ϕ_i denotes the feature map of the layer i from the VGG-16 and N_i denotes the size of the layer i.

Structure adversarial loss: The structure discriminator (Dis_s), as one of the opponents of the generator, has the task not only to determine the authenticity of the input image, but also to force the generator to generate a colour image with the same structure as the input edge image. Based on structure constraint, structure adversarial loss is defined as follows:

$$\begin{aligned} L_{advs} = {} & E_{s,c,u\sim P_{data}(s,c,u)}[\log Dis_s(s, Gen(s,u), f'_{aux}(c)) \\ & + \log(1 - Dis_s(s, c, f'_{aux}(c)))] \\ & - E_{s,u,c\sim P_{data}(s,u,c)}[Dis_s(s, Gen(s,u), f'_{aux}(c))] \end{aligned} \quad (4)$$

where Dis_s is the structure discriminator that determines whether the generated colour image $Gen(s,u)$ and edge image, s, have the same structure, and $f'_{aux}(c)$ is the intermediate representations of the auxiliary AE.

Texture adversarial loss: The texture discriminator (Dis_g), similar to the Dis_s, as the second opponent of the generator, is used not only to determine the authenticity of the input image, but also devoted to forcing the generator to produce the colour image whose texture is similar to the original colour image. Texture adversarial loss helps *Gen* reconstruct texture information of images and prevents the appearance of colour-bleeding artifacts. Based on texture constraint, texture adversarial loss is defined as follows:

$$\begin{aligned} L_{advg} = {} & E_{g,c,u\sim P_{data}(g,c,u)}[\log Dis_g(g, Gen(s,u), f'_{aux}(c)) \\ & + \log(1 - Dis_g(g, c, f'_{aux}(c)))] \\ & - E_{c,g,u\sim P_{data}(c,g,u)}[\log Dis_g(g, Gen(s,u), f'_{aux}(c))] \end{aligned} \quad (5)$$

where Dis_g is the texture discriminator that determines whether the generated colour image, $Gen(s,u)$ and greyscale image, g have a similar texture and $f'_{aux}(c)$ is the intermediate representations of the auxiliary AE.

4. Experiments

4.1. Experimental settings

Dataset: We collected a cartoon paintings dataset with 15,000 clean colour images and a ratio of training sets to test sets of 9:1. We generated sketch image pairs using the boundary detection filter, XdoG. The parameters of XdoG were set empirically to $t = 0.95$, $k = 4.5$ and $f = 10$. To verify the adaptability of our model in the future, we collected some sketches from the authoritative Sketchy [SBHH16] dataset and hand-drawns of non-professionals.

Network training details. Our model was trained in the PyTorch framework through a NVIDIA GeForce GTX 2080ti GPU. We trained our model using the Adam optimizer. The batch size was set at 32, the learning rate was 0.0001, and the training epochs were 125. We performed one gradient descent through the discriminator; the generator executed five times during training. Due to memory limitation, the resolution of the image was set to 224×224.

4.2. Ablation study

We conducted ablation studies in both qualitative and quantitative evaluations to analyse how the components proposed in our framework contribute to the final performance of cartoon sketch colourization.

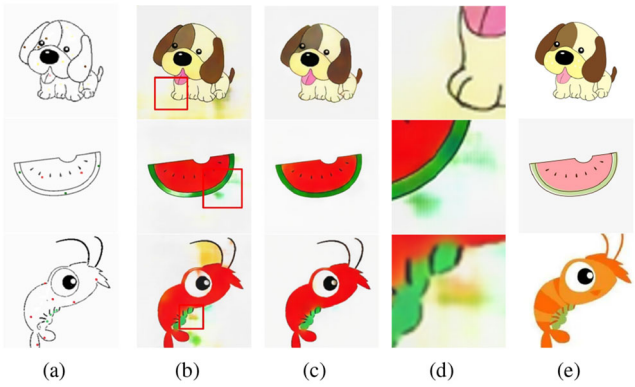


Figure 3: *Validation of the structure discriminator. (a) Sketch; (b) without texture discriminator; (c) with structure discriminator; (d)* ***partial*** *zoom in red frame; (e) ground truth.*

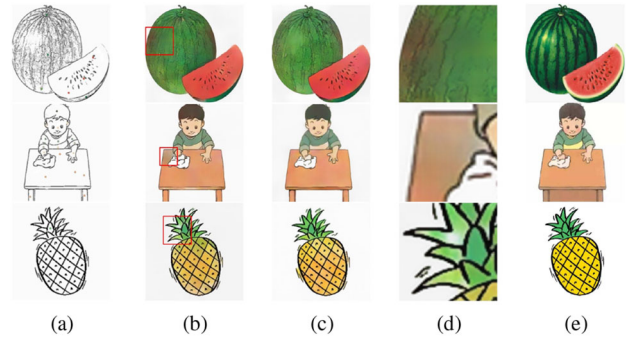


Figure 4: *Validation of the texture discriminator. (a) Sketch; (b)without texture discriminator; (c) with texture discriminator; (d)* ***partial*** *zoom in red frame; (e) ground truth.*

Qualitative comparison for structure constraint: To verify the effect of structure constraint, we compared the performance of the model with and without the structure discriminator. As shown in Figure 3, without the structure discriminator, there is a significant overflow in some local regions of the images.

Qualitative comparison for texture constraint: To verify the effect of texture constraint, we compared the performance of the model with and without the texture discriminator. As shown in Figure 4c, under the constraint of texture information, the results are vividly coloured with smooth texture and reasonable colour propagation. On the contrary, as seen in Figure 4b, colouring results are prone to show artifact and colour blank without texture constraint. For example, there are some grey artifacts in the upper left corner of the table, and the pineapple leaves lost colour in some areas. According to Figure 4, we can see that the texture discriminator effectively improves the quality of generated images, especially when eliminating colour-bleeding artifacts.

Qualitative comparison for L_{aux}: We compared our method with a clone version, but without L_{aux}. The results are shown in the left panel of Figure 5. It can be seen that L_{aux} helps the model generate more natural and realistic images with proper saturation and smooth highlights. By contrast, without L_{aux}, the model tends to produce incorrect colours and shadows, *e.g.* the dinosaur's teeth and legs appear inappropriately coloured, the snake's nose area is darker, the freckled shadow on the boy's face should appear in the freckled area instead of under his hair, *etc*. In addition, it is seen that without L_{aux}, the edge areas of the generated images usually become blurred with some illusory colours, such as in the third image in the last row of Figure 5. Using L_{aux} eliminates these effects.

Table 1: *Quantitative results of the ablation study.*

Methods	FID↓	PSNR↑	SSIM↑
W/o dis_s	37.11	21.54	0.865
W/o dis_g	38.01	21.20	0.862
W/o L_{aux}	37.63	21.37	0.867
W/o *AuxCI*	38.26	21.44	0.858
Ours	35.20	21.67	0.870

Qualitative comparison for AuxCI: The generalization ability is crucial for a model to be used for a variety of practical colouring tasks in line art. To verify the capability of the auxiliary conditional input to improve robustness, we test our model on the Sketchy Dataset [SBHH16]. In the right panel of Figure 5, it can be seen that our proposed method achieves gratifying results. However, when the AuxCI is removed, some problems such as incomplete colouring, colour divergence along the edge, and uneven colour distribution appear in the generated results.

Quantitative evaluation: As our evaluation metrics, we took peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [WBSS04], and Frechet inception distance (FID) [HRU*17]. Specifically, we used PSNR to measure the similarity between two images at the pixel level, SSIM to evaluate the distortion degree of the generated image content relative to the ground truth, and FID to assess the overall quality and diversity of a generated image. Table 1 evaluates the effectiveness of individual components. Experimental results demonstrate that our total work has the lowest FID, highest PSNR and SSIM. In addition, it is worth noting that our colour hints are simulated from ground truth, which means that our generated images are highly relevant to ground truth. Thus, PSNR and SSIM are also valid in the current situation.

4.3. Comparison with prior methods

We compared our model against previous colourization methods using the same sketches with a relatively consistent colour gamut in terms of colour hints. These methods include the online application PaintsChainer (Canna, Satsuki, Tanpopo) [Yon17], the commercial product Style2paints [lll18], UGSC-GAN [ZZLL21] and Ci *et al.* [CMW*18], which is a user-guided colouring model using cGAN. We trained the models Ci *et al.* [CMW*18] and Zhang *et al.* [ZZLL21] on our cartoon dataset.

Figure 6 shows the qualitative results of our model and others. It can be seen that our model can generate more natural and impressive

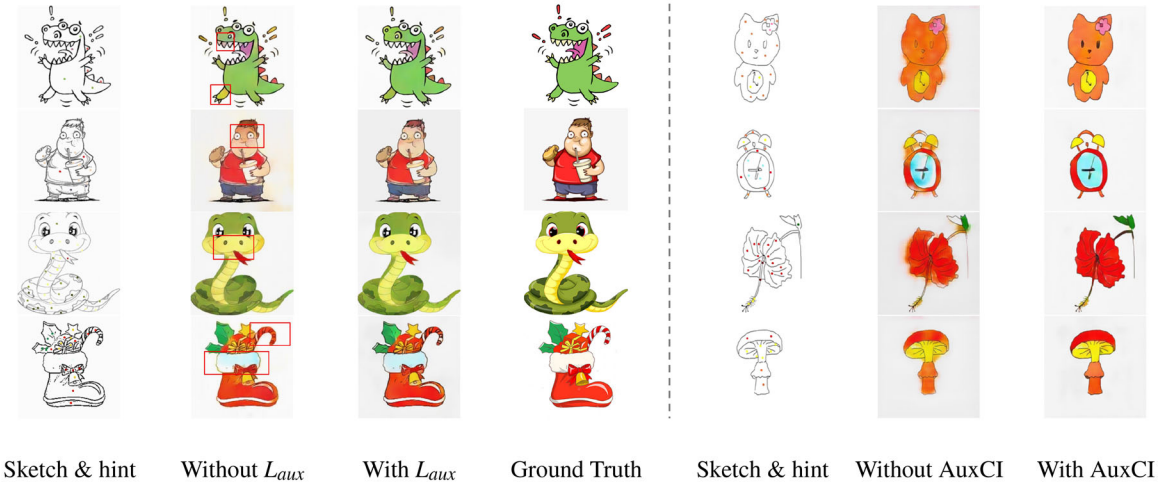


Figure 5: *Validation of the auxiliary feature loss (L_{aux}) and the auxiliary condition input (AuxCI). The left panel corresponds to the qualitative comparison for L_{aux}; the right panel corresponds to the qualitative comparison for AuxCI. The sketches on the right are from the Sketchy Dataset [SBHH16] and are abstractions of natural images, so we do not provide useless ground truth images.*

Figure 6: *Comparing results with prior works in our test dataset. (a) Sketch and hints; (b) ours; (c) UGSC-GAN; (d) Ci* et al.*; (e) Style2paints; (f) Canna; (g) Satsuki; (h) Tanpopo; (i) ground truth.*

images with less colour bleeding and fewer artifacts than the other methods, and the image colours are also consistent with the colour hints. In contrast, other methods have some quality problems. As Figure 6, PaintsChainer suffers from **severe** colour bleeding, which is **difficult** to correct with **proper** colour hints.

Style2paints, as a mature colourization tool, yields good colouring results, but, as shown in Figure 6e, its generated results appear oversaturated in colours that differ from the colour hints, with some colour-bleeding artifacts at the same time. For example, the watermelon baby and the Christmas stocking in Figure 6e have problem-

Figure 7: *More colouring results of our method. The upper of the left part is the colouring results of non-professional hand-drawn; the bottom of the left part is the colouring results of the Sketchy dataset; the top right of the image shows the results of our own dataset.*

atic colour spreading near the boundaries between adjacent objects. Also, there are some colours covering image edges in generated images, such as the edge of the wheel is covered by yellow and the black edges of the watermelon seeds almost disappear in Figure 6e.

The method of Ci *et al.* [CMW*18] has fewer artifacts in the generated images, but it does not follow the user-specified colours very well, such as the red quilt of the baby watermelon and the brown bear in the Christmas stocking in Figure 6d. Besides, the colour of images generated by the method may be lighter than the colour provided by the user, such as the leg of red monkey and the blue pillars in the ship in Figure 6d.

The UGSC-GAN, as we can see from Figure 6c, when using colour points as the colour hints, the colour of images is not smooth enough, for example, the red colour on the monkey pants does not sufficiently spread to the boundary of the region. More importantly, the colour coherency is also descended. When providing the darker blue to the ship, it displays a lighter colour; when giving a lighter green to grass, it displays a darker colour. This is largely caused by the low and high grey values of the greyscale images generated in the first stage in the corresponding blue and green region, respectively.

Comparison of colouring results in non-closed sketches. The images shown above all have closed and well-defined boundaries. Figure 9 shows an additional comparison regarding the colouring of non-closed sketches. In addition to the methods presented above, we also added Lazybrush [SDC09], which is a naïve segmentation-based colouring method that can achieve perfect colour filling with strict area control and accurate colour propagation. However, as shown in Figure 9, Lazybrush mainly suffers from two issues: (1) It needs to provide background colours during colouring, while in the case where a few blue lines are provided, there is a significant overflow in the unclosed region, such as the first line in Figure 9. To prevent overflow, we used blue lines to control the degree of overflow, but the result will appear to be insufficiently coloured, as shown in

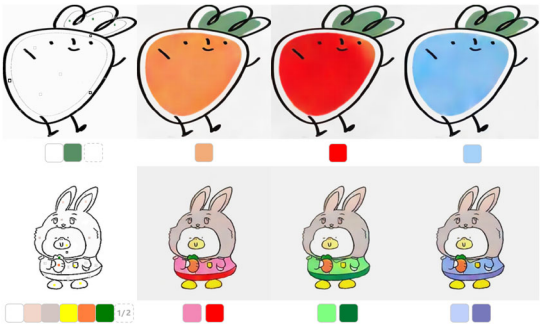


Figure 8: *Colouring results with various colour hints. The colour plates below the sketches show all the colours used in the colouring process, where the dotted boxes indicate the different colours used at a particular location, such as orange, red and blue below the coloured image.*

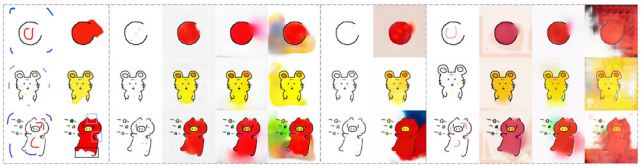


Figure 9: *Colouring results for non-closed sketches. The blue lines in the sketch are used to constrain the background to be white.*

the black box area; (2) some local areas with non-clear boundaries do not overflow, but some jaggedness (a single pixel point) appears (refer to the blue box areas). By comprehensive comparison with some previous methods, our method shows the better anti-spill performance when dealing with non-closed sketches.

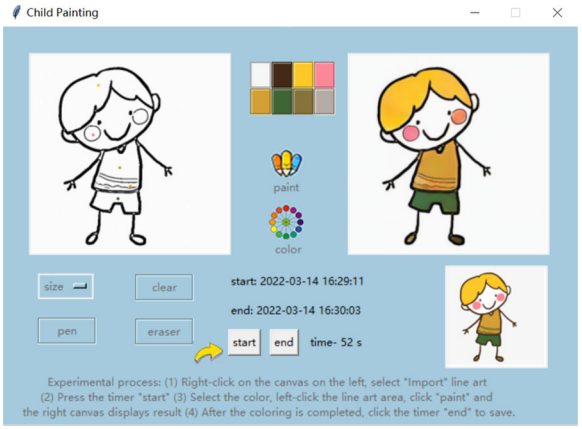


Figure 10: *User interface for colourizing cartoon sketches. The left canvas shows the provided sketch (we can also draw line image in it by using the 'pen' tool), the right shows the painting result in real time. In the middle are colour suggestions, colouring buttons and colour palette. In the bottom right corner is the reference image corresponding to the sketch. We have also set a timer on the user interface.*

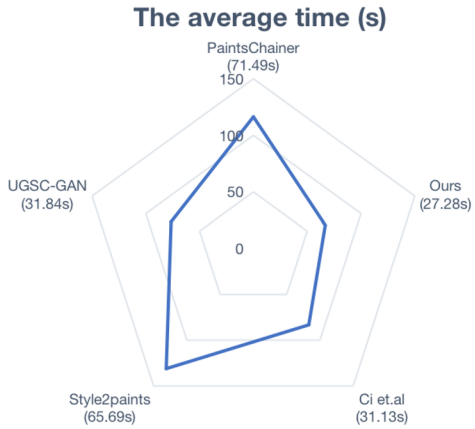


Figure 11: *Visualization of the average colouring time of the five methods. We compared the average time taken by participants to paint all the sketches over PaintsChainer, Style2paints [lll18], the method of Ci* et al. *[CMW*18], UGSC-GAN [ZZLL21] and ours. The value in parentheses indicates the standard deviation.*

4.4. More results

Figure 7 shows more results for simple and complex cartoon sketches, respectively, from non-professional hand-drawn, Sketchy dataset and our dataset. In the supplementary material, we further provided a side-by-side comparison of the aforementioned methods on sketches of varying complexity levels. Figure 8 shows our colouring results when various colour hints are provided.

4.5. User study

User-guided colourization is a highly subjective issue. Consequently, there is no single absolutely suitable evaluation metric to measure our architecture effectiveness. Therefore, we conducted a two-stage user study to evaluate our methods and others.

To evaluate user experience and satisfaction of our method, in the first-stage of the user study, we developed a user interface to enable users to interact (see Figure 10). We provided 15 participants with a tutorial and asked them to colour their respective randomly selected two sketches using our method and others (Style2paints, PaintsChainer, Ci *et al*. and UGSC-GAN). The order of the colouring methods is up to the participants. We limited them to colouring two sketches within 45 min, and the number of times, each sketch is coloured, depends on how satisfied they are with the results. Notably, because non-professionals may struggle with choosing suitable colours, which to some extent will affect a user's interactive experience and painting efficiency, we provided original image references and suitable colour suggestions in the user interface (see Figure 10). Through the experiment, we collected a total of 30 pictures, of which 28 are valid.

During painting, we recorded the time participants took for each sketch. The average time taking for each method is shown in Figure 11. After drawing, we provided each participant with a questionnaire and asked them to complete a multi-dimensional survey consisting of rankings in three dimensions: colour coherency, colour harmony and colouring sensitivity (the order is the same as in the questionary).

Colour coherency: Colour coherency indicates, to a greater extent, whether the model follows the colour tones of the colour hints provided by users. A good user-guided colourization model fills the sketch region using the correct red colour, instead of blue or white.

Colour harmony: In the present case, the broader meaning of colour harmony is whether the colour of an image is natural and realistic.

Colouring sensitivity: Good sensitivity refers to whether the results can be displayed quickly after the colour is given. This determines whether the method is of high practicality.

According to the survey, our method has better user experience and satisfaction, especially in colour coherency. As shown in Figure 13a, our method acquired 86.67% of the Top1 votes, indicating that our method follows more closely the colour hint of the user. Furthermore, compared with other methods, our model obtains more natural and realistic colour images for higher visual quality (see Figure 13b). In addition, as shown by the colouring sensitivity comparison with three methods (see Figure 13c), more than 80% of the participants perceive our method to have better colouring sensitivity, which indicates that, compared with other methods, our method achieves the fastest colouring. Similarly, the time records of participants' drawing (see Figure 11) also confirm the speed strength of our method.

In order to count the number of hints used by the participants, we added a colour hints counting function to our colouring app. We choose two methods that carry the interaction procedure used by our method: Ci *et al.* and UGSC-GAN for comparison. The results are shown in Figure 12.

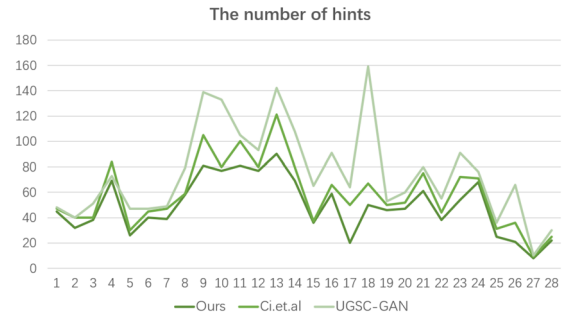
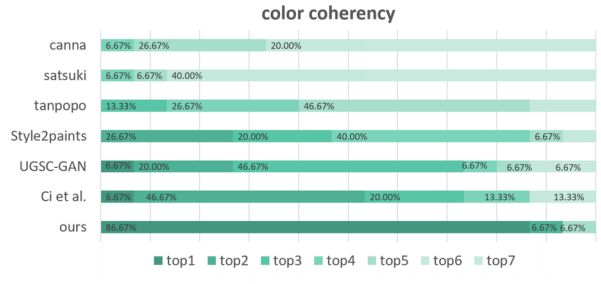
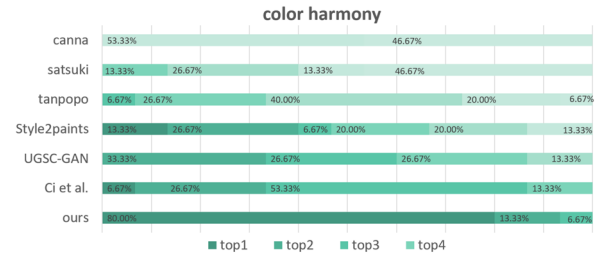


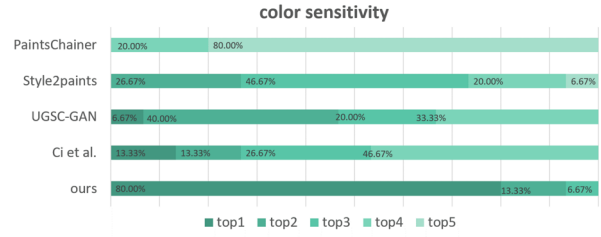
Figure 12: *Visualization of colour hints counts for the three methods when drawing the same sketch. To evaluate the effect of colour hints on colourization, we chose Ci* et al. *[CMW*18] and UGSC-GAN [ZZLL21], which can reasonably and easily count points, to compare with our method.*



(a)



(b)



(c)

Figure 13: *Comparisons in colour coherency, colour harmony and colouring sensitivity. We collected participants' ranking of the six methods on these three dimensions. The three methods of the PaintsChainer are coloured simultaneously when the colour hint is given. Thus, we show only four methods in the colouring sensitivity dimension.*

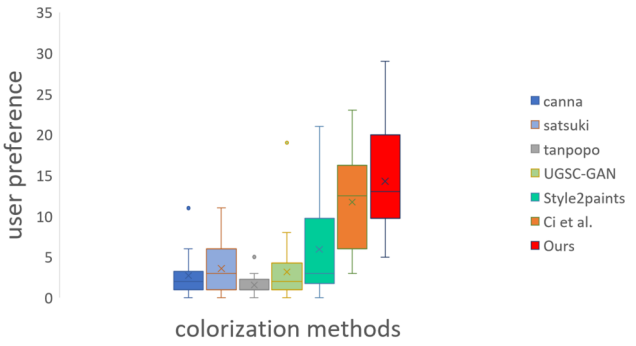


Figure 14: *User preference of different methods. The user preference on the vertical axis represents how many users have chosen the result drawn using the abscissa method.*

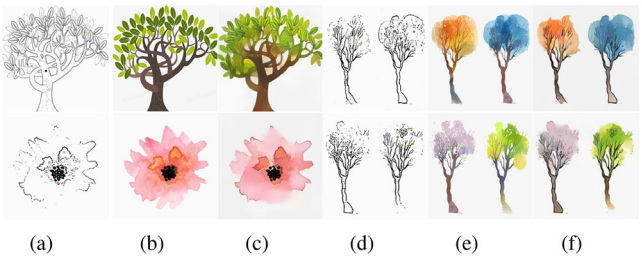


Figure 15: *Limitation analysis. (a) and (d) are sketches; (b) and (e) are ground truth; (c) and (f) are the results.*

From the line graph, it is clear that our method is the least expensive in terms of the hints used. Further combining Figure 13 and the following Figure 14, we can see that our method can achieve higher quality and obtain more user preferences with a minimum number of hints.

To further evaluate the visual quality, we conducted a second-stage experiment. We collected painting results from the participants of first-stage user study, and asked another 45 participants to choose their favourite images colourized by using each method. To avoid the influence of habitual thinking, we scrambled the order of the results of images of colouring that used the same sketch in different methods. For the sets of images, we counted the number of user preferences for each method (see Figure 14). The highest user preferences show that, in most cases, our model generates images with more favourites and higher quality.

5. Limitation

Although our model obtains positive achievements in colouring quality and practicability, there are also some limitations. In our work, we pay more attention to the underlying feature semantics, including texture and structure, while not considering the high-level semantic information of the input sketch. Figure 15 shows several failure instances, such as the second set of images from the left. When there is no clear boundary in the sketch, our model tends to produce blurry effects in regions without boundaries. However, when the sketch is cluttered with contour information, the colour-

ing result of our model has colour bleeding as shown in the first set of images from the left. In the future, we will try to add a semantic information fusion module for generating more refined colouring results.

In addition, our model has difficulty learning and expressing ink style patterns, such as the sets of images from the right side. The colouring results can barely reproduce colour gradation and shade variation of the groundtruth. We believe that some dedicated generation methods can be developed for ink style images.

6. Conclusion

Our work is a novel study for colourizing cartoon sketches with colour hints. We proposed a GAN based on cGAN with a texture discriminator. The texture constraint allows our model to focus on generating appropriate texture information and helps overcome colour-bleeding artifacts. The auxiliary feature loss based on an auxiliary AE, using a consistency constraint on a feature-map to assist the generator in mapping the latent space to ground truth domain, further improves colouring quality, especially in colour harmony and colour coherency. Likewise, the auxiliary condition input of the two discriminators enhances the generalization ability of our model on different types of sketches. Experimental results and evaluations show that our method outperforms existing methods. Also, our interactive programme realizes real-time colourization, giving children a better painting experience. In the future, as described in Section 5, we will start to explore ink style image colourization and explore how to add more semantic information to our model and further improve colouring quality.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61772440).

References

[CMW*18] Ci Y., Ma X., Wang Z., Li H., Luo Z.: User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM International Conference on Multimedia* (2018), pp. 1536–1544.

[CZG*20] Chen S.-Y., Zhang J.-Q., Gao L., He Y., Xia S., Shi M., Zhang F.-L.: Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics 28*, 2 (2020), 1198–1208.

[DDS*09] Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L.: Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), IEEE, pp. 248–255.

[GLX*20] Gao C., Liu Q., Xu Q., Wang L., Liu J., Zou C.: SketchyCOCO: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 5174–5183.

[GPAM*14] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial nets. In *Proceedings of the 27th Advances in Neural Information Processing Systems* (2014).

[HCL*18] He M., Chen D., Liao J., Sander P. V., Yuan L.: Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–16.

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 30th Advances in Neural Information Processing Systems* (2017).

[HSS18] Hu J., Shen L., Sun G.: Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7132–7141.

[HZC*17] Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H.: MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[IZZE17] Isola P., Zhu J.-Y., Zhou T., Efros A. A.: Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1125–1134.

[JAFF16] Johnson J., Alahi A., Fei-Fei L.: Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision* (2016), Springer, pp. 694–711.

[KJPY19] Kim H., Jhoo H. Y., Park E., Yoo S.: Tag2Pix: Line art colorization using text tag with SECat and changing loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9056–9065.

[KLP*21] Kim E., Lee S., Park J., Choi S., Seo C., Choo J.: Deep edge-aware interactive colorization against color-bleeding effects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14667–14676.

[LKL*20] Lee J., Kim E., Lee Y., Kim D., Chang J., Choo J.: 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* (2020), pp. 5801–5810.

[lll18] lllyasviel: style2paints. https://github.com/lllyasviel/style2paints (2018). Accessed: 2019-03-22.

[LLW04] Levin A., Lischinski D., Weiss Y.: Colorization using optimization. In *Proceedings of the ACM SIGGRAPH 2004 Papers* (2004), pp. 689–694.

[LSZE20] Liu B., Song K., Zhu Y., Elgammal A.: Sketch-to-art: Synthesizing stylized art images from sketches. In *Proceedings of the Asian Conference on Computer Vision* (2020).

[QWH06] Qu Y., Wong T.-T., Heng P.-A.: Manga colorization. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 1214–1220.

[RFB15] Ronneberger O., Fischer P., Brox T.: U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention* (2015), Springer, pp. 234–241.

[SBHH16] Sangkloy P., Burnell N., Ham C., Hays J.: The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 1–12.

[SCH20] Su J.-W., Chu H.-K., Huang J.-B.: Instance-aware image colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 7968–7977.

[SDC09] Sỳkora D., Dingliana J., Collins S.: Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum 28* (2009), 599–608.

[SLF*17] Sangkloy P., Lu J., Fang C., Yu F., Hays J.: Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5400–5409.

[SZ14] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[TL19] Tan M., Le Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (2019), PMLR, pp. 6105–6114.

[WBSS04] Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612.

[WKO12] Winnemöller H., Kyprianidis J. E., Olsen S. C.: XDoG: An extended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics 36*, 6 (2012), 740–753.

[XWF*20] Xu Z., Wang T., Fang F., Sheng Y., Zhang G.: Stylization-based architecture for fast deep exemplar colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 9363–9372.

[Yon17] Yonetsuji T.: PaintsChainer.github.com/pfnet. *PaintsChainer 1* (2017), 2.

[YSS21] Yuan M., Simo-Serra E.: Line art colorization with concatenated spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 3946–3950.

[ZJLL17] Zhang L., Ji Y., Lin X., Liu C.: Style transfer for anime sketches with enhanced residual U-Net and auxiliary classifier gan. In *Proceedings of the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)* (2017), IEEE, pp. 506–511.

[ZLSS*21] Zhang L., Li C., Simo-Serra E., Ji Y., Wong T.-T., Liu C.: User-guided line art flat filling with split filling mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9889–9898.

[ZLW*18] Zhang L., Li C., Wong T.-T., Ji Y., Liu C.: Two-stage sketch colorization. In *Proceedings of the SIGGRAPH Asia 2018 Technical Papers* (2018), ACM, pp. 261.

[ZMG*19] Zou C., Mo H., Gao C., Du R., Fu H.: Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–16.

[ZZI*17] Zhang R., Zhu J.-Y., Isola P., Geng X., Lin A. S., Yu T., Efros A. A.: Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999* (2017).

[ZZLL21] Zhang J., Zhu S., Liu K., Liu X.: UGSC-GAN: User-guided sketch colorization with deep convolution generative adversarial networks. *Computer Animation and Virtual Worlds* (Oct. 2021). https://doi.org/10.1002/cav.2032