Received December 4, 2019, accepted December 23, 2019, date of publication December 27, 2019, date of current version March 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962579

Two-Stage Sketch Colorization With Color Parsing

HUI REN, JIA LI, AND NAN GAO

Key Laboratory of Visual Acoustic Technology and Intelligent Control System, Ministry of Culture and Tourism, Communication University of China, Beijing 100024, China

Corresponding author: Jia Li (lcxnn@cuc.edu.cn)

This work was supported by the Ministry of Culture and Tourism of China (No. WHB1701), and the National Key Research and Development Plan (No. YS2018YFB1403703).

ABSTRACT We implement high-quality sketch colorization using two-stage conditional generative adversarial network (GAN) training based on different intermediate features. The intermediate features used in autonomous colorization are the grayscale parsing and interval pixel-level color parsing. The autonomous colorization based on grayscale parsing feature can learn the spacial topology of pixels in the first stage to guide the colorization in the second stage. The autonomous colorization based on pixel-level color parsing feature can learn the color information of few feature points in the first stage to guide the colorization of all pixels in the second stage. Additionally, we use the intermediate feature of sampling points as constraint and achieve the color reconstruction using Laplacian mesh editing as a special second stage. Furthermore, the interactive colorization uses the superpixel color parsing as the intermediate feature. Specifically, we use the simple linear iterative cluster (SLIC) to obtain a palette that maintains the edges in the first stage to guide the colorization in the second stage. As for evaluation metrics, we propose a color-coded local binary pattern (CCLBP) score based on color distances from the first-order 8 pixels to the central pixel, to measure the degrees of color blurring and mess. We also propose a light-sensitivity (LS) score based on the reversed grayscale map, to measure the degrees of auto painting and overfitting of the color hint. According to the L1 distances between the original and generated color images based on these scores, compared with state-of-the-art methods including one stage approaches such as pix2pix and PaintsChainer and two-stage approaches such as Style2Paints and DeepColor, our model can achieve the highest-quality autonomous colorization. Moreover, compared with pix2pix, PaintsChainer and Style2Paints with color hints, according to the proposed objective evaluation as well as the user visual study, our model can achieve the highest-quality interactive colorization as well.

INDEX TERMS Colorization, sketch, line arts, generative adversarial network.

I. INTRODUCTION

Similar to the gray-to-image problem [1], [2], the sketch-to-image problem requires the color information to make the generated results vivid. Coloring is an ill-posed problem that requires generating reasonable colors and textures based on a sketch, which is an attractive issue in the field of non-photorealistic rendering. Traditional energy-based methods such as LazyBrush [3] can only handle some lines with low shape complexity, which makes it prone to generating unnatural color in hair and other detailed parts and causing vacancies and other phenomena. With the development of deep learning, especially the generative adversarial networks, many painting tasks have been implemented using adversarial learning [4]–[6] by employing the adversarial loss to improve the effect of the color features, and it has achieved good visual effects. PaintsChainer [7] is the first online project of sketch colorization using a deep learning method. There are three automatic colorization methods including PaintsChainer1 [8], PaintsChainer2 [9] and PaintsChainer3 [10].

There are some important issues in sketch colorization. First, sketch colorization is based on the sketch contour, and so the paired data are used for training. The sketch can provide the conditional information to fit the different colors in the different body spaces. Moreover, the color hints can be

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Piuri.

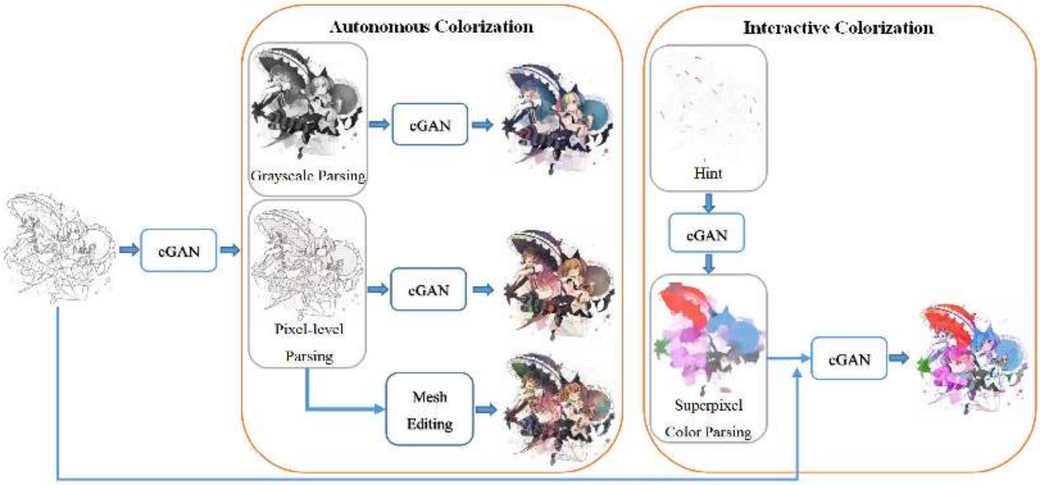


FIGURE 1. Two-stage sketch colorization with color parsing.

used as the conditional information to help the colorization. Therefore, the quality of sketch edge and color hint is critical to colorization.

Second, sketch-to-image is an ill-posed problem. It is difficult to get vivid colorization and avoid the mode collapse. To reduce the complexity of training while introducing more conditional information, it is better to implement it by generating some intermediate feature, and following that with another refinement stage. There are many two-stage approaches that can be used to refine the generation results. The GAN can contain a single generator or discriminator, or it can be a concatenation of multiple generators and discriminators, each of which comprises one decomposed task such as a Gaussian pyramid module [11], palette module [4] or others [12].

To obtain abundant features during the deep learning, there are some typical structural forms of the GAN. First, DCGAN [13], [14] is a set of convolutional neural network structures that enable tricks such as BatchNorm to be successfully used. Second, Pix2pix [15] uses the advantages of the UNet structure, which has become widely used. Pix2pixHD [16] generates high-resolution visually appealing results using a novel adversarial loss, as well as the architecture of a multi-scale generator and discriminator. The residual network is used to get more important features. Our colorization networks use pix2pix as well as pix2pixHD.

Another important issue is the mode collapse. The regularized GAN [17] applies an encoder to a real picture to construct an implicit feature space, which then helps the generator to perform better. The GAN without an encoder can hardly fit the multiple distributed modes, but the RegGAN can do better. The EBGAN [18] and BEGAN [19] are similar works. In [6], the Wasserstein distance is used to train the cGAN to overcome model collapse and enable the model to converge much better. However, the ability to fit complex distributions is reduced. The image palette that we use from the SLIC can be seen as color regularization information that is obtained by an encoder for a real color map, making it easy to fit more complex color schemes. Moreover, each segment area has the same color, which is convenient for constructing a pencil line hint, similar to the real situation.

As for the visual performance, current sketch-to-image algorithms and models have two common problems with the coloring results. One is dirty content, such as the background diffusion of PaintsChainer or the locally dirty colorization of Style2paints v3. Another problem is the lack of color, such as [12]. However, the model that we proposed improves the colorization quality by adding intermediate learning features and is better than other existing models according to objective and subjective evaluations.

Motivated by the idea that feature parsing can improve the quality of generated images [20], [21]. We implement the sketch colorization task with color parsing. Our approach is divided into two parts: two-stage autonomous colorization based on grayscale parsing feature and interval pixel parsing feature and two-stage interactive colorization based on superpixel color parsing, as shown in Fig 1. The compared state-of-the-art algorithms include pix2pix, PaintsChainer, DeepColor and Styel2paintsV3, which are compared using CCLBP score, LS score and subjective visual evaluation. Our contributions can be summarized as follows.

1) The grayscale parsing feature and interval pixel-level color parsing feature are used to implement high-quality autonomous sketch colorization.
2) SLIC is used to obtain the superpixel color parsing that maintains the sketch edges, and through the two-stage cGAN training, high-quality interactive colorization is achieved.
3) We propose two novel objective evaluation criteria based on the multiscale-CCLBP score and LS score to evaluate the colorization performance. Furthermore,

based on the regional obedience, color obedience and visual quality proposed by Style2paints v3, two new subjective criteria regarding partial cleanliness and colorization completion are added to make a more scientific visual evaluation of the sketch colorization.

II. RELATED WORKS

A. GRAY-TO-IMAGE METHODS

As gray-to-image methods, [1] proposes a fully automatic approach that produces vibrant and realistic colorization. Reference [2] uses a CNN to directly map a grayscale image, along with the sparse, local user hints or the global histogram of the LAB and HSV information, to produce a colored output.

B. SKETCH-TO-IMAGE METHODS

Especially for the sketch based image generation problem, the colors of anime are more varied and the color matching is more complicated than grayscale based image generation.

As the first generation of PaintsTransfer, Style2paints [5] is a project that is similar to PaintsChainer in which the natural colorization based on the reference picture is well performed. However, because the fused VGG feature maintains the topological feature of the reference image, so the color structure tend to be overfitting. The improved Style2paints v3 can generate better results [22].

Liu *et al.* [6] improved pix2pix by adding the high-level feature matching loss of VGG and total variation loss, which are used by [23] too. It employed Unet [24] and PatchGAN.

Frans and Kevin [4] proposed a novel network, which designs a color scheme generation network and a shading network. But the approach can not improve the diversity of automatic colorization [25].

Reference [26] firstly resolve the colorization of an entire manga page. It has both global and local functions for interactive revision. An image processing task such as getting the super resolution [27] version of an image is also applied.

C. CONDITIONAL GAN METHODS

Understanding things well needs guidance in certain fields, and so fitting the probability distribution becomes fitting the conditional probability distribution. The typical works include the conditional GAN [28], improved GAN [29] and GVM [30]. Pix2pix [15] can translate an ill-posed map into a map rich in details.

D. FINE GRAINED GAN METHODS

The fine-grained GANs separate generative process into multiple steps. The LAPGAN [11] is the first work to apply hierarchical or iterative generation. The StackGAN [31] can generate 256×256 images from the caption. The PPGN [32] also advocated using an iterative process to continuously adjust and improve images.

E. EDGE DETECTION METHODS

To get abundant sketches, edge extraction from real anime images is important. Reference [4] use OpenCV to extract edges, but the results have too much noise and are visually far from professional line-art. References [6] and [23] use an XDoG [33] filter, which makes it easy for edge extraction to generate black shades or miss important edges, and the amount of noise is more. The HED [34] extracts edges with messy and thick lines, which usually misses many high-frequency details. Reference [5] and [12] use sketchKeras [35], which can stably extract edges with an appropriate ratio of low and high frequency information as well as lines with suitable thick. The results are the visually closest to an artificial sketch.

F. HINT DESIGN

The design of the color hints is divided into two categories.

1) LOCAL HINT

The first is the color patch with a random position and size after random white-out and Gaussian blurring [4], which can easily result in local color overfitting. In the test phase, the results are highly dependent on the user-guided specified color attributes, and it easily results in color diffusion. This kind of scheme involves the spatial grayscale characteristics and has negative effects on the correct topology. The second type is the small pixel patch hint [2], [26], which makes color diffusion difficult, but it can avoid the problem of color hint overfitting. Reference [26] uses the pixel-level hint, and the patch size of dots in [2] ranges from 1×1 to 9×9. Although the second type of scheme has a smaller number of color hints, it makes the overall colorization more robust. In our autonomous colorization, we use an 8×8 patch because a 1×1 hint will encourage an average, grayish generated hint during the gray prediction process, and the same precise color hint patches with the sketch and real color image are supposed. On the other hand, larger hints will cross adjacent edges and will easily cause hint overfitting.

2) GLOBAL HINT

Interactive colorization needs to predict the global hint based on a small number of local hints, which can increase the scope and effect of the hint. The global color feature such as the histogram or palette can be extracted from the reference image.

Reference [26] predicts the color histogram that represents the color distribution, but the corresponding colorization location is implicit. Reference [4] predicts the dominant color palette of an area block, but it loses important edge information. The details of the coloring are messier, and the saturation is poor. Another method is Style2Paints v3 [12], which generates the global color draft by randomly splashing various colors. However, this color parsing is too casual and diffused, and there is no scientific region constraint for the colorization.

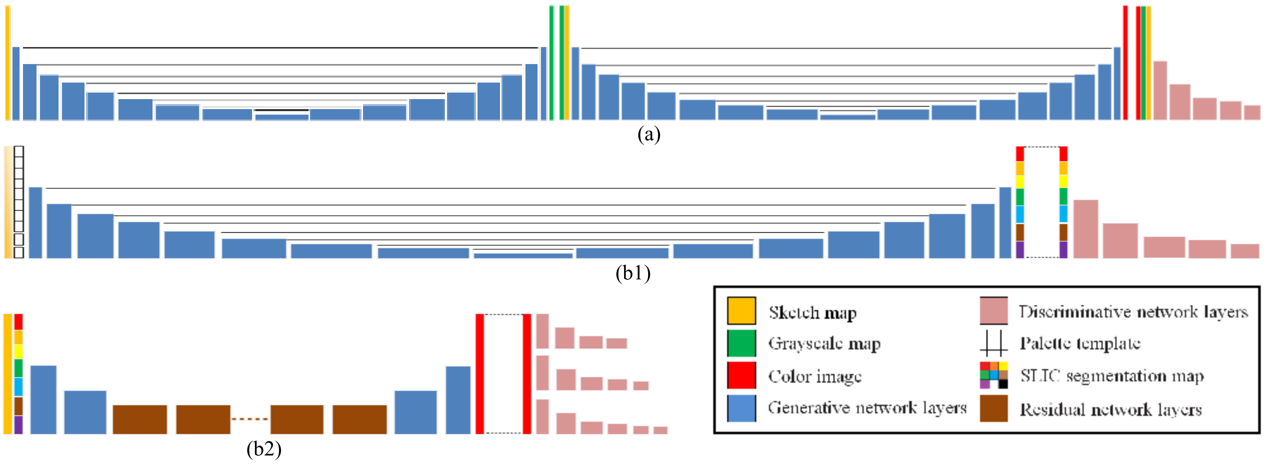


FIGURE 2. Network structure. (a) Autonomous colorization, (b1) The SLIC stage, and (b2) The interactive colorization stage.

However, our approach uses the superpixel segmentation algorithm, which is a simple linear iterative cluster (SLIC) [36], to obtain a color segmentation map that maintains the edge as the referenced global hint to guide the interactive sketch colorization.

Recently, image segmentation has been used in a variety of computer vision tasks, such as depth prediction [37], virtual try-on [20], scene understanding [38], [39] and image generation [40]. The use of SLIC can obtain color parsing to extend the line hint to the local area, thereby providing more color information to the interactive colorization.

III. METHODS

Our approach is divided into three parts: autonomous colorization based on a grayscale image, autonomous colorization based on sampling points and interactive colorization based on color clustering. The reasons why these two algorithms are not combined are as follows. (1). Autonomous colorization and interactive colorization are different. Autonomous colorization is supposed to fill color in a wide range of an image and leave no white area. On the other hand, interactive colorization focuses on high color obedience and an interactive user experience. (2). In the interactive colorization process, the grayscale feature conflicts with the hint and they constrain each other, which is not conducive to sketch colorization.

A. AUTONOMOUS COLORIZATION

1) COLORIZATION BASED ON GRAYSCALE PARSING

The Wnet that is used in this paper is equivalent to the combination of two Unets, which are subjected to both forward and backward propagation, and the two generated parts are synchronously conducted at the same stage, which is different from the StackGAN scheme. Wnet makes two related generation processes supervise each other and conduct joint optimize. For example, the grayscale image and the color image that is generated from the grayscale image both come from the same sketch. During autonomous colorization, the grayscale image expands the spatial information of the anime edges, and the color image refines the color distribution of the anime lines on the basis of the grayscale image.

Our network of autonomous colorization use the Unet including the convolution, BatchNorm and ReLU layer, which constitute the Wnet. We first extract the edges of the color images in the training set and give them sample hints. Then, these line-arts are sent to the first half of Wnet whose aim is to generate the grayscale maps with corresponding color hints. Second, the predicted grayscale maps and sketches with the same random blocks from real color maps are sent to the last half of the Wnet to obtain the automatic colorization results, as shown in Fig 2 (a). The grayscale and color maps are associated with each other by means of sharing hints, and they contain the same semantics as the sketch. Additionally, the PatchGAN size of D is 70×70 and a 62×62 probability matrix is obtained. The input of D is a set including a sketch, grayscale map and color map, which are specifically divided into two random groups: a real sketch, a fake grayscale map, and a fake color map or a real sketch, a real grayscale image and a real colored image. The generator will more precisely fit the shadows and highlights of the manga, as well as effectively avoid regional color diffusion.

The GAN needs to accumulate trust. In other words, as long as it generates a picture that the discriminator thinks is acceptable, it will no longer be willing to take risks to try new pictures to become more satisfactory because this process will bring more punishment. It becomes easier for our generator to obtain the trust of discriminator and to change and update modes, which can alleviate the meaningless gradient problem by making the distributions of the generated data and real data more similar.

Compared with the traditional end-to-end network structure, the end-mid-terminal structure is step-by-step color-filled, and the middle end corresponds to an encoder of the

FIGURE 3. Results of autonomous and interactive sketch colorization. (a). From left to right are the sketch, PaintsChainer, Style2Paints v3, DeepColor, our model of autonomous colorization and real images. (b). The first row is PaintsChainer, the second row is ours, the last row is Style2Paints v3.

topological space, which makes parameter training relatively easy. Moreover, the synchronous training progress strengthens the discrimination, and the meaning of the basic semantics is continuously enriched, unlike previous works such as AutoPainter, DeepColor and Style2paints v1, which are all based on sets of sketches and color maps.

References [41]–[43] were based on the pipeline of dual learning, but they are more inclined to handle style conversion. Wnet has an edge encoder and a grayscale encoder, but there is no color encoder, and so the mode collapse phenomenon is alleviated by adding the prior color information of dots. The cross-entropy loss function of our autonomous colorization is as follows, where yu is the true grayscale image, yw is the corresponding true color image, and x_d is the sketch edge x with the corresponding user-defined color hints d.

$$L_{cGAN}(G, D) = E_{x_d, yu, yw}[\log D(x_d, yu, yw)] + E_{x_d, z}[\log(1 - D(x_d, Gu, Gw))] \quad (1)$$

The input of the first Unet is x_d, z is the random noise. Gu is the fake grayscale image.

$$Gu = G_u(x_d, z) \quad (2)$$

The input of second Unet is x_d with noise z and the result of gray prediction network, x_d can make the colors of dots clearer when generating the color map. Gw is the corresponding fake color image.

$$Gw = G_w(x_d, Gu, z) \quad (3)$$

Discriminating these two Unet at the same time will strengthen and optimize the discriminant effectively, as shown in (1). On the other hand, the generator G tries to play against the discriminator until achieving the final balance.

The L1 distance of the grayscale and color maps is taken into consideration.

$$L_{L1}(G) = E_{x_d, yu, z}[||yu - Gu||_1] + E_{yu, yw, z}[||yw - Gw||_1] \quad (4)$$

To encourage the generated image to be more visually similar to the real image, we employ a pretrained VGGNet to extract the high-level information of the image. The L2 distance from ReLU_4.3 of the VGG16 for the predicted grayscale map and the real gray map and that between the predicted color map and the real color map are respectively calculated as the feature constraints, and the formula is as follows.

$$L_{F4}(G) = E_{x_d, yu, z}[||\varphi_j(yu) - \varphi_j(Gu)||_2] \quad E_{yu, yw, z}[||\varphi_j(yw) - \varphi_j(Gw)||_2] \quad (5)$$

By optimizing the following loss function ($\lambda_1 = \lambda = 10$), as shown in (6), the final optimal generator is obtained.

$$G^* = \arg\min_G \max_D L_{cGAN}(G, D) + \lambda_1 L_{L1}(G) + \lambda L_{F4}(G) \quad (6)$$

The total variation loss is used in [6], [23] in order to make the color smoother and enhance the color correlation between adjacent pixels, but it also easily leads to color diffusion and overlap. Predicting the grayscale map can effectively avoid traditional problems such as the color spreading across regions, the lack of color and low color saturation.

In the training phase, for a certain color image in every epoch, the 8 × 8 color feature blocks have random spatial positions that are randomly selected from 0 to 60, and then added to the sketch and the grayscale image. The control points of the sketch, grayscale and color map remain the same throughout the learning process. In the test stage, the user can usually give any number of color hint dots. However, in our autonomous colorization, we provide no hint. The results are shown in Fig 3(a). In the test comparison of interactive coloring, we provide hints to the model using the grayscale intermediate feature and implement an objective evaluation as one of the baselines of interactive colorization.

2) COLORIZATION BASED ON PIXEL-LEVEL COLOR PARSING

Similar to the cGAN based on intermediate feature of grayscale images, we trained the cGAN based on the intermediate feature of sampled feature points, without adding prior color information of dots in the first stage. The second stage can be a cGAN or Laplacian mesh editing [44]. Specifically, we take the reconstructed color image using the Laplacian differential mesh deformation algorithm in the RGB color space as the baseline representing the efficiency and accuracy of the deep learning algorithm. Considering the sketch as a plane with some gullies, the intermediate feature points that are obtained by the GAN are regarded as the applied force, and then the colored image can be regarded as groups of mountains.

3D mesh deformation method generally contains three steps. First, transform the Descartes coordinates of the mesh into differential coordinates. Second, change the coordinates of some of the characteristic points to predict the desired deformation effect. Third, implicitly solve the linear equation Ax=B and then restore the differential coordinates as Descartes coordinates to accomplish the mesh editing.

Command M = (V, Ea, F) is a triangular mesh with n vertexes. V is the set of vertexes, Ea is the set of edges, and F is the set of mesh faces. The vertex adjacency matrix A of the mesh is represented as follows.

$$A_{ij} = \begin{cases} 1 & (i,j) \in E_a \\ 0 & otherwise \end{cases} \tag{7}$$

$$D_{ij} = \begin{cases} d_i & i = j \\ 0 & otherwise \end{cases} \tag{8}$$

$$K_{ij} = \begin{cases} d_i & i = j \\ -1 & (i,j) \in E_a \\ 0 & otherwise \end{cases} \tag{9}$$

The graph Laplacian matrix is K=D-A in (9). And d_i is the number of vertexes that are adjacent to the vertex i in (8).

Vertex i is represented by an absolute coordinate in the Descartes coordinate system: vi= (xi, yi, zi). The differential coordinates are defined as follows:

$$\delta_i = [\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)}] = v_i - \frac{1}{d_i} \sum_{j|(i,j) \in E_a} v_j \tag{10}$$

$$L_s x = D\delta^{(x)} \tag{11}$$

The differential coordinates of the mesh can be obtained according to the following formula (11), where matrix L_S =K, x is an n-dimensional vector that contains the absolute coordinate components of all vertexes, and y and z are similar to x.

The solution requires solving a reversible linear system. The large equations can be effectively obtained by the linear algebra operation and then the topological information of the deformed mesh can be obtained by the decoupling and recoupling of these equations. Take the X space as an example:

$$\begin{bmatrix} \frac{L_s}{I_{m\times n}} \end{bmatrix} x = \begin{bmatrix} \frac{D\delta^{(x)}}{XF_{1:m}} \end{bmatrix} \tag{12}$$

In the equation, n is the number of all mesh points, and m is the row of feature vectors, whose corresponding coordinates are stored in XF, YF and ZF respectively. $I_{m\times n}$ represents some lines to be added under L_S, whose elements are zeros except for 1 in the position whose column is the index of the corresponding feature point. During practical applications, we usually select some constraint points on the 3D grid and then use the least squares method to realize the mesh editing.

(a) N=9. (b) N=8. (c) N=7. (d) N=3.

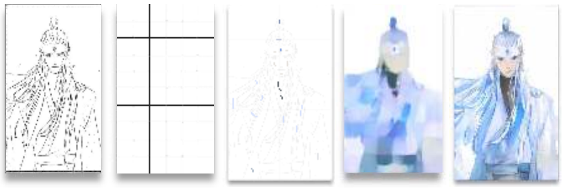
FIGURE 4. Reconstruction based on the Laplacian mesh deformation corresponding to different sampling interval parameters N.

The solution process can use advanced linear system solvers such as TAUCS or SUPERLU [44]. The algorithm steps that are used in this paper are as follows.

1) The 512 × 512 image is divided into 64 64 × 64 subimages. For engineering reasons, 128 × 128 and 256 × 256 subimages will overload the memory and cause errors when reading the vertex information using OpenMesh.
2) The GAN obtains the color feature points of the image based on big data. Considering the edge connection and internal filling performance of the subimage, 22 × 22 feature points with an interval of 3 are selected, which is approximately 10 percent of all pixels. Other intervals cannot obtain perfect reconstruction results, as shown in Fig 4.
3) Map each subimage onto the 64 × 64 3D mesh topology and transform it into the differential mesh space to reconstruct the colored image using Laplacian surface editing in the RGB space. We obtain the symmetric positive definite matrix A and the right hand color distribution matrix B, and then send them to the SuperLU_DIST [45].
4) Integrate the overall colored image using the segmented subimages.

B. INTERACTIVE COLORIZATION

Interactive colorization contains two stages. The first stage obtains the SLIC color segmentation feature map S of the real anime color map as the target image. Then, the translucent sketch image is merged with the 10 × 10 checkerboard image to get input image I. The checkerboard image is used to guide the transform from the sketch to the SLIC map. Considering that color parsing is the main purpose in the first stage, we make the sketch and checkerboard image translucent, as shown in Fig 5. Then, we randomly extract 20-30 color hint lines with random thicknesses from 3-5 and random locations from S and attach them to I to obtain I_h, which is used as the input of the Unet. In the learning process, we randomly change the hue to prevent the overfitting of colors. The model of the first stage is marked as A.



(a) (b) (c) (d) (e)

FIGURE 5. Interactive colorization based on color parsing. (a) sketch, (b) checkerboard image, (c) hint map with translucent sketch and checkerboard, (d) color parsing result, and (e) final result generated from (a) and (d).

The reason why the SLIC algorithm is used to obtain the global color hint is because the palette that is obtained after segmentation seeks to retain the sketch edges, and each segment area has the same color, which is convenient for constructing a pencil line hint similar to the real situation. The SLIC function is as follows. The measured distance is divided into Dc and Ds. In the CIELIB color space, Dc calculates the lab color distance between pixel i and cluster center C_j. The measurement distance of Ds is in the x and y coordinate space, and it calculates the position distance between pixel i and cluster center S_j. The color proximity and spatial proximity are standardized through their maximum distances Nc and Ns in the clusters as (15). Eventually, color clustering is achieved.

$$D_c = \left[(L_j - L_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2\right]^{1/2} \quad (13)$$

$$D_s = \left[(x_j - x_i)^2 + (y_j - y_i)^2\right]^{1/2} \quad (14)$$

$$D = \left[(D_c/N_c)^2 + (D_s/N_s)^2\right]^{1/2} \quad (15)$$

The cross-entropy loss function of the first stage is as follows.

$$L_{cGAN_1}(G_1, D) = E_{I_h,S}[\log D(I_h, S)] + E_{I_h,z}[\log(1 - D(I_h, Gs))] \quad (16)$$

The L1 distances of the SLIC map are then considered.

$$L_{L1}(G_1) = E_{I_h,S,z}[||S - Gs||_1] \quad (17)$$

The generator is optimized by using the following loss function ($\lambda_1 = 10$).

$$G_1^* = \arg\min_{G_1}\max_{D} L_{cGAN_1}(G_1, D) + \lambda_1 L_{L1}(G_1) \quad (18)$$

In the second stage, all samples in the training set of the first stage are subjected to model A, and the soft target palette Ps is obtained, which concatenates with the real sketch K as the input of the second stage. Then, the real anime color map T is used as the target image. In the learning process, we randomly change the hue of Ps and T to prevent the overfitting of colors. The colorization training is implemented by means of pix2pixHD, whose generator uses the progressive strategy and discriminator uses multiple di scrimination with different receptive fields. The model of the second stage is marked as B. The cross-entropy loss function of B is as follows.

$$L_{cGAN_2}(G_2, D) = E_{K,Ps,T}[\log D(K, T)] \quad E_{K,Ps,z}[\log(1 - D(K, G_T))] \quad (19)$$

The L1 distances of the color map are then considered.

$$L_{L1}(G_2) = E_{K,T,z}[||T - G_T||_1] \quad (20)$$

The perception loss of VGG19 is taken into consideration, including relu1_1, relu2_1, relu3_1, relu4_1 and relu5_1.

$$L_{Fj}(G_2) = E_{K,T,z}[||\varphi_j(T) - \varphi_j(G_T)||_2] \quad (21)$$

Moreover, the total variation loss is added to ensure smooth colorization.

$$TV(x) = \sum_{i,j}\left[(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2\right]^{1/2} \quad (22)$$

The generator is optimized by using the following loss function ($\lambda_1 = \lambda_{Fi} = 10, \lambda_{TV} = 1$).

$$G_2^* = \arg\min_{G_2}\max_{D_1,D_2,D_3} \sum_{k=1,2,3} L_{cGAN_2}(G_2, D_k) + \lambda_1 L_{L1}(G_2) + \sum_{Fi=1\sim5} \lambda_{Fi} L_{Fi}(G_2) + \lambda_{TV} L_{TV}(G_2) \quad (23)$$

IV. EXPERIMENT RESULTS

A. DATASET

Our original database comes from Safebooru, Baidu and Danbooru2017 [46]. After web crawling, image cropping and data filtering, we collected 20,000 512 × 512 colored manga images for training, and 200 images for test. After using sketchKeras to obtain the corresponding original sketch, we get the corresponding grayscale map by rgb2gray transformation, and then constitute the sketch-gray-color image sets.

B. TRAINING SET

We set the batchsize to 4 in the training stage. In addition, we train the network with Adam, and the initial learning rate is 2e-4. Moreover, the instance normalization and the normal initialization of network are used.

C. BASELINE

One-stage colorization based on pix2pix is used as the baseline of autonomous colorization, and the pix2pix trained with hint is used as a baseline of interactive colorization. In addition, the colorization based on grayscale features with hints is used as the baseline of interactive colorization as well.

D. QUANTITATIVE EVALUATION

In this section, we compare our reconstructions with the images that were generated by PaintsChainer, DeepColor and Style2paints V3. We select 200 line-arts of real images from the test set to conduct the quantitative evaluation including light-sensitivity score and the CCLBP score because adaptive

B2	B3	R1	R2
B1	I		R3
G3	G2		G1

FIGURE 6. Color coded LBP, R1 and B3 share the same value.

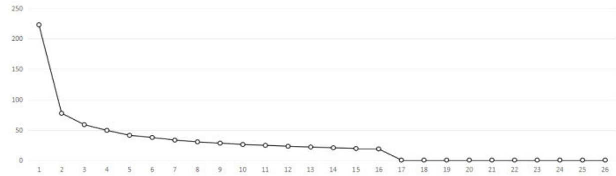


FIGURE 7. Example curve of the CCLBP score, where the horizontal axis represents the threshold of T.

color filling and the quality of the filling result are two aspects of great concern for sketch colorization. The evaluation of the colorization effect is based on the same hue of the same position, although the shapes of the hints that are used by different algorithms are slightly different. For example, these include the slim pencil line of PaintsChainer, which is similar to our method; the thick line of DeepColor; or the circular point of Style2paints V3.

1) LIGHT-SENSITIVITY SCORE

After obtaining the colored image, we get the grayscale image G. The light-sensitivity image is obtained by subtracting G using a white plate with the same size as the colored image.

The L1 distance of the light-sensitivity maps for the colored images and the original color images will represent the degree of auto painting and hint overfitting. That is, the more negative the L1 distance is, the more inefficient auto painting is, and the more ill-posed the spatial structure. Namely, the color distribution is unsaturated. Additionally, it illustrates that colorization is too dependent on the hint, and when the area lacks a hint, it will have insufficient colorization. Namely, the overfitting of hints is serious.

2) COLOR CODED LBP SCORE

We propose the CCLBP to objectively evaluate the manga colorization. By calculating the color distances from the first-order circular 8 pixels to the current central pixel, the CCLBP feature map is obtained based on distance coding in the RGB space. For example, the color value of R channel for location (i,j) in CCLBP map comes from the binaries R1, R2 and R3, as shown in Fig 6 and (26). D_R in (24) is the distance value in R channel between two pixels in original image color space. If the distance value T_{R3} between I(i,j) and I(i,j+1) is larger than T_{th}, the corresponding binary R3 is 1, as shown in (25). Otherwise, R3 is 0. After obtaining R1, R2 and R3, they are decoded into the color space of CCLBP map, as shown in (26). This feature map has a tolerance to color variation. The larger the threshold of T is, the higher the degree of tolerance towards regional colorization errors.

Moreover, the CCLBP score mainly focuses on the rendering details including the smoothness and clearness of the coloration in different areas. The black area represents the similar colors or closed hues, and the edge feature can represent the degree of color diffusion. Namely, the messy edge represents disharmonious colorization. The L1 distances of the CCLBP feature maps between the original images and the generated images is used to represent the coloring effect. The more negative the L1 distance is, the more inefficient the ability of natural and high-quality colorization. We select thresholds of 5, 10, and 15 to get the multi-CCLBP scores, according to Fig 7.

FIGURE 8. Detailed examples of the reasons for positive values of the light-sensitivity score and CCLBP score in the experiment.

FIGURE 9. CCLBP and LS maps of an interactive example. From left to right, they are the sketch, colorization result, LS map, CCLBP5 map, CCLBP10 map and CCLBP15 map. The first row is PaintsChainer v3, followed by Style2paints v3, our model of pix2pixHD+SLIC, pix2pix+SLIC, pix2pix + hint and our model with grayscale + hint.

$$T = \left(D_{R_{ij}}^2 + D_{G_{ij}}^2 + D_{B_{ij}}^2\right)^{1/2} \tag{24}$$

$$R_{c=1:3} = \begin{cases} 1 & T_{R_c} \geq T_{th} \\ 0 & otherwise \end{cases} \tag{25}$$

$$I_{CCLBP}(R) = 256 \times (4 \times R1 + 2 \times R2 + R3)/8 \tag{26}$$

E. RESULTS

As for autonomous colorization, our result is the best compared with the other three state-of-the-art algorithms,

FIGURE 10. Autonomous sketch colorization results and the corresponding LS maps and multiscale-CCLBP maps (T_h = 5, 10, 15).

TABLE 1. L1 scores of autonomous colorization.

L1 Score	LS	CCLBP		
		T_{th}=5	T_{th}=10	T_{th}=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
Style2paints3	-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
Our+Laplacian	-0.301	-1.375	-3.734	-5.183
Our+pixel	**-0.262**	-9.446	0.612	1.576
Our+grayscale	-0.362	**7.461**	**4.334**	**2.613**

as shown in TABLE 1. The values come from the statistical means of 200 images in the test set.

Taking the original color map as a reference, the light-sensitivity score and CCLBP score should be negative, which are lower than the ideal values of the original color map. The reasons for the positive values in the experiment are as follows.

1) For the light-sensitivity score, we hope that the white area is minimized during the coloring process; therefore, if the algorithm automatically colors the white space in the original image, the light-sensitivity score may be positive.
2) For the CCLBP score, the better effect is expected to increase the black area of the CCLBP map as much as possible to indicate the naturally smooth and nondiffusion of the color. If the coloring algorithm reduces the jaggedness in the original image and it has better local details, a positive score will appear. Another reason is that some edge information is missing during edge extraction, resulting in more black areas, as shown in Fig 8.

However, the comparison experiments among PaintsChainer, Style2paints V3, DeepColor and our model are based on the same edge extraction map. Therefore, the relative values absolutely can explain the colorization quality of the corresponding algorithm.

In TABLE 2, test images from 1 to 100 are evaluated by using PaintsChainer1, 101-150 using PaintsChainer2, and 151-200 using PaintsChainer3. The mean CCLBP score is calculated from T_5, T_{10} and T_{15}, respectively weighted by 0.5, 0.3 and 0.2, because the CCLBP map with T_5 can show more colorization details, which is more important. The overall score is the mean value of LS score and the mean CCLBP score. Our model with the grayscale map performs better than our model with the pixels and the model with the Laplacian mesh editing in autonomous coloring. In the interactive coloring test, we provide the approach based on the grayscale map with hints in the same position and color as in the other algorithms. By comparing the other two-stage algorithms, it is proved that the test results using SLIC as the color parsing features are better than other one-stage method such as PaintsChainer and pix2pix, as well as two-stage methods such as Style2paints v3.

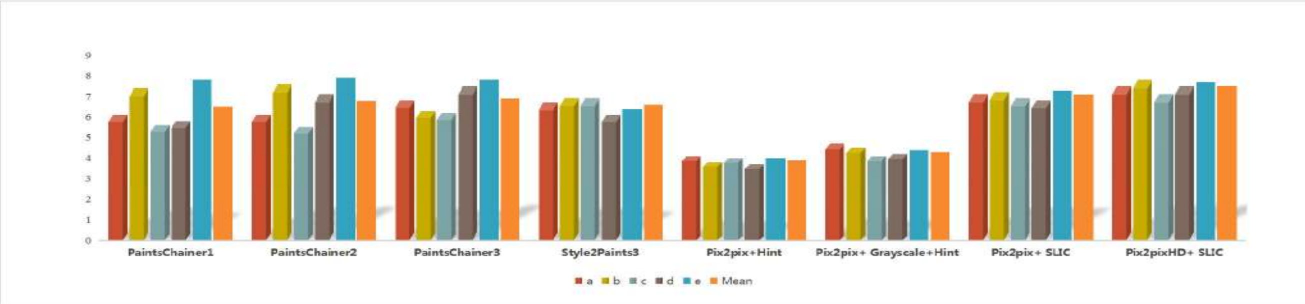


FIGURE 11. Histogram of the subjective evaluation scores.

TABLE 2. L1 scores of interactive colorization.

L1 Score	LS	CCLBP				Overall
		T_5	T_{10}	T_{15}	Mean	
PaintsChainer1	0.71	-11.72	-1.69	**0.58**	-6.25	-2.77
PaintsChainer2	**0.84**	-17.83	-5.04	-1.61	-10.75	-4.95
PaintsChainer3	0.63	-6.77	-7.24	-6.80	-6.92	-3.14
Style2paints3	-1.96	-7.70	-5.35	-3.65	-6.18	-4.07
Pix2pix+Hint	-2.91	-12.65	-6.74	-3.95	-9.13	-6.02
Pix2pix+ Grayscale+Hint	-2.58	-9.75	-3.55	-1.77	-6.29	-4.43
Pix2pix+ SLIC	-2.51	-5.29	**-1.68**	-0.08	-3.16	-2.83
Pix2pixHD+ SLIC	-2.20	**-3.77**	-2.69	-1.54	**-3.00**	**-2.60**

TABLE 3. Average scores of our user study.

	a	b	c	d	e	Mean
PaintsChainer1	6.1	7.4	5.6	5.8	7.8	6.5
PaintsChainer2	6.1	7.6	5.5	7.1	**7.9**	6.8
PaintsChainer3	6.8	6.3	6.2	7.5	7.8	6.9
Style2Paints3	6.7	6.9	6.9	6.1	6.4	6.6
Pix2pix+Hint	4.1	3.8	4.0	3.7	4.0	3.9
Pix2pix+ Grayscale+Hint	4.7	4.5	4.1	4.2	4.4	4.3
Pix2pix+ SLIC	7.1	7.2	6.9	6.8	7.3	7.1
Pix2pixHD+ SLIC	**7.5**	**7.8**	**7.1**	**7.5**	7.7	**7.5**

The subjective evaluation is divided into five aspects. (a). The overall color visual effect is scored according to the evaluator's first feeling of the coloring result, and it describes the overall popularity of the colorization. (b). The color obedience describes whether the resulting color is consistent with the hint color. PaintsChainer3 often has a low color obedience score. (c). The regional obedience describes whether the color spreads within a reasonable range, and whether it spreads to the background or crosses the line boundary to enter another key area. PaintsChainer often spreads color into the background, and Style2Paints3 sometimes can not spread the correct color within the reasonable range. (d). The local rendering purity determines whether the local part is enlarged enough to observe whether the color is messy, or whether the local information is clear. Style2Paints3 often results in messy local images. (e). Colorization completion describes the size of the white areas of the entire image, and white space in the key part is considered to be a colorization failure. Style2Paints3 often leaves large white areas in the colorization. Our results have better visual quality and create more vivid paintings than other approaches.

To evaluate the visual performance in the interactive colorization task, we conducted a user study with 20 participants consisting mostly of students who were interested in manga.

1) LAYOUT OF THE USER STUDY

After a short introduction to the colorization task, users are instructed to observe the original color images with

FIGURE 12. More colorization results of real sketches. From up to down, there are the sketch, sketch with hint, predicted SLIC maps and our results.

wonderful coloring performance. Then, they are instructed to randomly select a hint map from our test set. We show the hint maps all the time during the observation time of the different color images of the compared methods. Since the colorization quality is related to the perception of details and the whole generated images, we set a time limit of 30 seconds, after which we hide the current image. During the observation time, the users were asked to score the five subjective evaluation criteria from a-e. The score is from 1 to 10. There are a total of 1200 generated images, and each participant needs to randomly observe 400 images. We designed the study to only take approximately 4 hours per participant, which results in the collection of 8000 human decisions. Table 3 shows the average scores of our user study. The corresponding histogram is shown in Fig 11.

V. CONCLUSION AND FUTURE WORK

By means of L1 distance evaluation based on light-sensitive-score and CCLBP-score, compared with PaintsChainer, Style2paints V3 and DeepColor, it is shown that our model can achieve high-quality manga colorization both visually and numerically when given the line-art.

The creations of PaintsChainer are easily color-diffused, and the results of Style2paints V3 frequently experience a lack of coloring and messy details. Finally, the results of DeepColor are usually varicolored. All the above matters are improved by our automatic colorization method. Future work will implement global feature coloring, style transformation, and specific emotion rendering.

1) Global feature colorization. Our model can colorize the similar grayscale areas in the global perception field. Other global features are the color histogram and image saturation, which can implement the style conversion.
2) Emotion rendering. Different colors or different combination of colors represent different emotional features. Emotion colorization is higher level rendering based on cognitive psychology, which will help to research image emotion recognition and interaction. Changing the facial expressions corresponding to specific emotions can also enrich the emotion rendering effect.

Moreover, the training scheme, which includes using grayscale parsing and random patch hints together as well as the SLIC parsing implementation, are universal for the interactive generation of ink paintings, watercolors, and oil paintings. Ink paintings especially need to be researched and promoted, which will increase public awareness and participation in traditional Chinese painting.

In summary, through subjective and objective evaluations, compared with the results of the state-of-the-art coloring algorithms, the coloring effect of our model with color parsing is the best both in autonomous and interactive colorization, which is shown in Fig 9, Fig 10 and Fig 12. However, the coloring effect of the colorist is better because of the rich coloring experience and a mount of time consuming. Therefore, AI coloring needs to be improved in the future.

REFERENCES

[1] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. ECCV*, Oct. 2016, pp. 649–666.

[2] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017.

[3] D. Sýkora, J. Dingliana, and S. Collins, "LazyBrush: Flexible Painting Tool for Hand-drawn Cartoons," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 599–608, Apr. 2009.

[4] K. Frans, "Outline colorization through tandem adversarial networks," 2017, *arXiv:1704.08834*. [Online]. Available: https://arxiv.org/abs/1704.08834

[5] L. Zhang, Y. Ji, X. Lin, and C. Liu, "Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 506–511.

[6] Y. Liu, Z. Qin, T. Wan, and Z. Luo, "Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks," *Neurocomputing*, vol. 311, pp. 78–87, Oct. 2018.

[7] T. Yonetsuji. (2017). *PaintsChainer*. [Online]. Available: https://github.com/pfnet/PaintsChainer

[8] TaiZan. (2016). *PaintsChainer Tanpopo*, Preferred Networks. [Online]. Available: https://petalica-paint.pixiv.dev/index_zh.html

[9] TaiZan. (2017). *PaintsChainer Canna*, Preferred Networks. [Online]. Available: https://petalica-paint.pixiv.dev/index_zh.html

[10] TaiZan. (2017). *PaintsChainer Satsuki*, Preferred Networks. [Online]. Available: https://petalica-paint.pixiv.dev/index_zh.html

[11] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[12] L. Zhang, C. Li, T. T. Wong, Y. Ji, and C. Liu, "Two-stage sketch colorization," in *Proc. SIGGRAPH Asia Tech. Papers*, Dec. 2018, p. 261.

[13] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[14] H. Heo and Y. Hwang, "Automatic sketch colorization using DCGAN," in *Proc. 18th ICCAS*, 2018, pp. 1316–1318.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[17] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," 2016, *arXiv:1612.02136*. [Online]. Available: https://arxiv.org/abs/1612.02136

[18] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial networks," 2016, *arXiv:1609.03126*. [Online]. Available: https://arxiv.org/abs/1609.03126

[19] D. Berthelot, T. Schumm, and L. Metz, "BeGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*. [Online]. Available: https://arxiv.org/abs/1703.10717

[20] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2019, pp. 2357–2366, Jun. 2019.

[21] R. K. Sarvadevabhatla, I. Dwivedi, A. Biswas, and S. Manocha, "SketchParse: Towards rich descriptions for poorly drawn sketches using multi-task hierarchical deep networks," in *Proc. 25th ACM MM*, 2017, pp. 10–18.

[22] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu. (2018). *Style2Paints*. [Online]. Available: https://github.com/lllyasviel/style2paints/tree/master/V3

[23] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5400–5409.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent.*, Oct. 2015, pp. 234–241.

[25] K. Frans. (2017). *Deepcolor*. [Online]. Available: http://color.kvfrans.com/draw

[26] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, "Comicolorization: Semi-automatic manga colorization," in *Proc. SIGGRAPH Asia Tech. Briefs*, Nov. 2017, p. 12.

[27] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[30] J. Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. ECCV*, 2016, pp. 597–613.
[31] H. Zhang, T. Xu, and H. Li, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE ICCV*, Oct. 2017, pp. 5907–5915.
[32] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 4467–4477.
[33] H. WinnemÖller, J. E. Kyprianidis, and S. C. Olsen, "XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization," *Comput. Graph.*, vol. 36, no. 6, pp. 740–753, Oct. 2012.
[34] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE ICCV*, Dec. 2015, pp. 1395–1403.
[35] Lllyasviel. (2017). *SketchKeras*. [Online]. Available: https://github.com/lllyasviel/sketchKeras
[36] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels," (No. REP_WORK), EPFL, Lausanne, Switzerland, Tech. Rep. 149300, Jun. 2010.
[37] J. Ye, Y. Ji, X. Wang, K. Ou, D. Tao, and M. Song, "Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2829–2838.
[38] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.
[39] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8277–8286.
[40] D. Liang, R. Wang, X. Tian, and C. Zou, "PCGAN: Partition-controlled human image generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Aug. 2019, pp. 8698–8705.
[41] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE ICCV*, Oct. 2017, pp. 2223–2232.
[42] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1857–1865.
[43] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE ICCV*, Oct. 2017, pp. 2849–2857.
[44] O. Sorkine, "Laplacian mesh processing," in *Proc. Eurographics (STARs)*, 2005, pp. 53–70.
[45] X. S. Li, J. W. Demmel, J. R. Gilbert, L. Grigori, M. Shao, and I. Yamazaki, "SuperLU users' guide," Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-44289, Nov. 1999.
[46] DANBOORU. (2017). *A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset*. [Online]. Available: http://www.gwern.net/Danbooru2017