LINE ARTIST: A Multi-style Sketch to Painting Synthesis Scheme

Jinning Li, Siqi Liu
Shanghai Jiao Tong University
lijinning, magi-yunan@sjtu.edu.cn

Mengyao Cao
Shanghai Jiao Tong University
cmy_0622@sjtu.edu.cn

Figure 1: Stylizd images synthesized by LINE ARTIST with sktch images

Abstract

Drawing a beautiful painting is a dream of many people since childhood. In this paper, we propose a novel scheme LINE ARTIST *to synthesize artistic style paintings with freehand sketch images, leveraging the power of deep learning and advanced algorithm.*

Our scheme includes three models. The Sketch Image Extraction (SIE) model is applied to generate the training data. It includes smoothing reality images and pencil sketch extraction. The Detailed Image Synthesis (DIS) model trains a conditional adversarial net to generate reality detailed information. The Adaptively Weighted Artistic Style Transfer (AWAST) model is capable to combine multiple style image with content via VGG19 network and PageRank algorithm. The appealing stylized images are then generated by optimization iterations.

Experiments are operated on the Kaggle Cats dataset and The Oxford Buildings Dataset. Our synthesis results are artistic, beautiful and steadier. Real sketch tests prove that our scheme performs very well on reality environments. With different artist styles, our scheme can generate paintings of various styles. With our scheme, everyone can draw a beautiful picture only by drawing a sketch and then feeding it into our system.

1. Introduction

Deep learning has been applied to many scientific and engineering fields. However, researches on artist production are still limited. In our world, there are many people who want to paint beautiful paintings but feel depressed at not good at coloring. So we are considering creating a method using machine learning to help make some people's dream come true.

In this paper, we propose a new scheme, LINE ARTIST, to paint like a well-known painter. The only thing need to be done is to draw some sketch lines. Then, LINE ARTIST will drawing a meaningful and elegant painting for you like images in Fig.1. Everybody can become a skilful painter!

Much work have been done to transform an real-world image into a stylized one. However, in our work, we aim at generating an artist style painting from a freehand sketch.

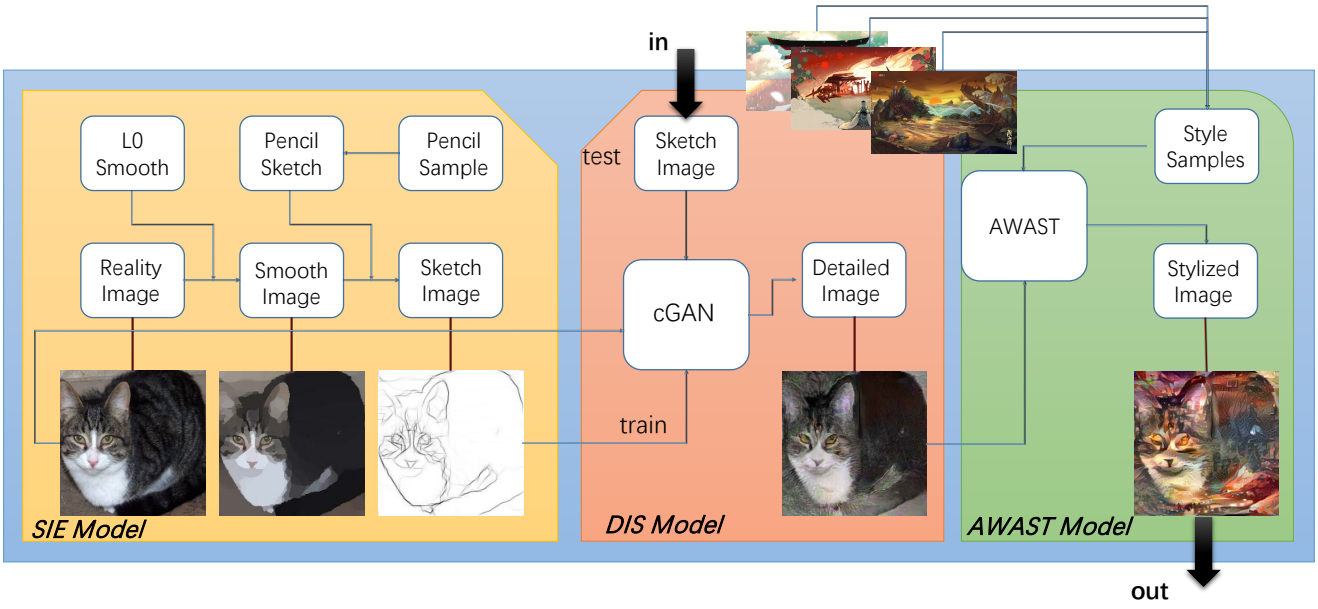


Figure 2: Overview of LINE ARTIST

Our scheme is able to extract the information from the sketch draft, generate a detailed image with new information using GAN. Then, we apply an Adaptively Weighted Artistic Style Transfer [5] algorithm based on VGG19 [20] convolutional neural network and PageRank algorithm to transform our synthesized detailed image into a stylized one.

In order to train a supervised system receiving a sketch image and generating a more informative detailed image in Section 3.2, a corresponding training dataset is needed. However, there is no such dataset published and building a new one will cost a large scale of time and human resources. To solve this problem, we introduce the *Sketch Image Extraction* (SIE) model to synthesize sketch-like images efficiently, whose color is yellow in Fig.2.

To extract the sketch more accurately, int the SIE model we employ the L_0-Smooth [24] algorithm to smooth the initial image in order to make the edges more distinct and wipe out the unnecessary veins. Then, we adapt the pencil drawing [13] method to extract the sketch image from the smooth one. Through the SIE model, we obtain the dataset containing pairs of initial reality images and the corresponding sketch images.

To generate an informatively detailed image from a given sketch image, the *Detailed Image Synthesis* (DIS) model is introduced whose color is orange in Fig.2. We use the dataset generated by the SIE model to train a system that receive the skech image extracted and output a detailed image by generating more information. In our test, by input the sketch draw by human, this system will also synthesize a good result. In this paper, we adopt the Pix2Pix [10] algorithm to the do the synthesis operation.

Finally, the *Adaptively Weighted Artistic Style Transfer* (AWAST) model, whose color is green in Fig.2. solves the problem of artist style transfer with multiple painting samples. Artist Style [5] is a efficient tool to combine two images with both low-level features and high-level features extracted from CNN. In this paper, we introduce the *Adaptive Style Weight* (ASW) based on the feature similarity network and PageRank algorithm. Using ASW, multiple painting samples can be combined more naturally and beautiful. After these process, we obtain a colorful painting result only from a line sketch by an user.

The contributions of this paper are summarized as bellow:

- A delicate sketch image synthesis scheme and two elaborate datasets containing pairs of reality images and corresponding sketch images.
- An efficient DIS model achieving more reality details from sketch images.
- A novel algorithm to adaptively combine multiple painting samples and synthesize appealing stylized images.

2. Related Works

Sketch Extraction Many researches [6, 11, 4]focus on the edge extraction based on algebraic algorithm like sobel

operator and fuzzy mathematics. These traditional edge extraction methods are good substitutions of sketch and usually run fast. However, the trends and continuity of extracted edges are not as natural as man-made ones. In [2], the author propose a random forest based method to detect edge. In the reference [23], the author propose a CNN-based edge detection algorithm, which performs better than the traditional ones. In the paper [22], the author propose a RSRCNN to extract roads from aerial images, which can also be applied to the sketch extraction. However, the CNN-based methods are highly relied on the training datasets and cost a lot to train a network. In [26] the author propose a face sketch synthesis scheme base on greedy search, this technique can synthesis sketch for other objects. However, its effect is more like a grey transfer than sketch extraction. In [25, 24], efficient methods of image smoothing are proposed based on wave pattern, which is helpful preprocessing for extraction. In [13], Lu propose a fast scheme to synthesize pencil drawing sketch image. The result is very appealing.

Detail Synthesis In [19, 10, 15], methods extended from GAN are used to synthesize detailed images with more information from given materials. These models are usually easier to train while the result are more fuzzy to some extent. Chen proposed an end-to-end cascaded refinement networks in [1] to synthesize large-size reality image from semantic layouts, whose result has a high resolution and accuracy. But this method is highly dependent on training datasets. In [8], the author propose an image synthesis model based on Laplacian pyramid, which has a lower computation complexity. In [17], Context Encoders, a GAN based model, is promoted to generate more information from the surroundings, in which artist style could be applied.

Style Transfer Most researches about style transfer focus on the combination of content and single style. In [9], initial content is transformed into oil style by pixel-level analysis. Gatys et al. [5] proposed a style transfer scheme based on CNN whose results are quite appealing. The model in [12] combines markov random fields with [5]. In [21], a faster feed-forward style image synthesis network is proposed. Texture synthesis is used to transfer image style in [3]. In this paper, multiple styles are combined together to synthesize appealing results with PageRank [16] in undirected graph [14].

3. Formulation

Our scheme includes three models: the *Sketch Image Extraction* (SIE) model, the *Detailed Image Synthesis* (DIS) model ,and the *Adaptively Weighted Artistic Style Transfer* (AWAST) model.The overall goal of our scheme is to receive a freehand sketch image $\hat{\boldsymbol{s}}$ and then generate a synthesized painting $\boldsymbol{p_s}$ with artist style. The overview of our scheme is shown in Fig.2.

3.1. Sketch Image Extraction Model

The SIE model receives a reality image $\boldsymbol{r}$. After the L0-Smooth process denoted by $l_0(\boldsymbol{x})$, $\boldsymbol{r}$ is transformed to the smooth image $\boldsymbol{m}$. Then, the sketch extraction algorithm $ext(\boldsymbol{x})$ receives the smooth image and produce the sketch image s.

$$\boldsymbol{S} = ext(l_0(\boldsymbol{R})),$$

where $\boldsymbol{S}$ and $\boldsymbol{R}$ are sets of $\boldsymbol{s}$ and $\boldsymbol{r}$.

Pairs of $\boldsymbol{r}$ and $\boldsymbol{s}$ is denoted by $[\boldsymbol{R}, \boldsymbol{S}] = \{(\boldsymbol{r}, \boldsymbol{s}) | \boldsymbol{r} \in \boldsymbol{R}, \boldsymbol{s} \in \boldsymbol{S}\}$, where $\boldsymbol{S}$ is the set of $\boldsymbol{s}$.

3.1.1 Image Smoothing

Actually, if you take a photo $\boldsymbol{R}$ of something, say, a tree, there are complex edges. For example, the veins of the leaves. However, when we draw a picture, for most people, we cannot draw so well and so subtle. Instead, we only draw the overall edges. So our goal is to generate a painting from a simple sketch. The first task is to abandon the detail veins in $\boldsymbol{R}$.

Smoothing is a useful method to abandon the detail veins. When we smooth a picture, we erease those detailed edges. In this paper, we adopt L_0-Smooth [24], an image smoothing algorithm via L_0 gradient minimization. For every pixel in $\boldsymbol{R}$, we denote the pixel map of the objective smooth image as $\boldsymbol{M}$, then the gradient in $\boldsymbol{M}_{i,j}$ is $\nabla \boldsymbol{M}_{i,j} = (\partial_x \boldsymbol{M}_{i,j}, \partial_y \boldsymbol{M}_{i,j})^T$. The gradient measure in $\boldsymbol{M}$ is:

$$cnt(\boldsymbol{M}) = \#\{(i,j) \mid |\partial_x \boldsymbol{M}_{i,j}| + |\partial_y \boldsymbol{M}_{i,j}| \neq 0\}, \quad (1)$$

which counts the number of pixels in $\boldsymbol{M}$ where $|\partial_x \boldsymbol{M}_{i,j}| + |\partial_y \boldsymbol{M}_{i,j}| \neq 0$. The objection of the optimization process is:

$$\min_{\boldsymbol{M}} \left\{ \sum_{i,j} (\boldsymbol{M}_{i,j} - R_{i,j})^2 + \lambda \cdot cnt(\boldsymbol{M}) \right\} \quad (2)$$

In L_0-smooth algorithm, auxiliary variables $h_{i,j}$ and $v_{i,j}$ are introduced to calculate the minimization of Eqn.2:

$$\min_{\boldsymbol{M},h,v} \left\{ \sum_{i,j} (\boldsymbol{M}_{i,j} - R_{i,j})^2 + \lambda \cdot cnt(\boldsymbol{M}) + \epsilon(h,v) \right\} \quad (3)$$

with $\epsilon = \beta((\partial_x \boldsymbol{M}_{i,j} - h_{i,j})^2 + (\partial_y \boldsymbol{M}_{i,j} - v_{i,j}))$. Then with $\boldsymbol{M}$ initialized by $\boldsymbol{R}$, the iteration will converge to the smooth image $\boldsymbol{M^*}$

3.1.2 Canny Edge Extraction

With the smooth image $\mathbf{M}^*$ extracted, we can extract the contributive edge image $\mathbf{C} = canny(M^*)$ by Canny operator. Here we adopt the Canny operator with Sobel method and double threshold. The result is shown in Fig.3. We can see that the overall shape of the tree is extracted without unexpected noise, which is much more like human sketch.

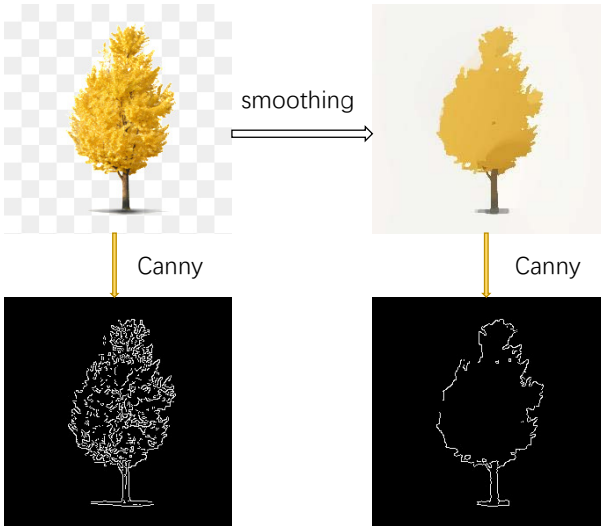


Figure 3: The different Canny result before and after smoothing. The upper left image is the origin image. The upper right is the smoothing one. The left lower is the edge of the origin image and the right lower is the edge of the smoothing one.

3.1.3 Pencil Sketch Extraction

Although a sketching-like image is already generated in Section 3.1.2, the line in the generated image $\mathbf{C}$ is not natural enough compared with real freehand sketching by humans. For most people, we cannot draw a line so long. There is always some break point.

In this way, the Canny generating picture is not what we want. To solve this problem, we use the the algorithm of producing pencil drawings from natural images [13]. We mainly use the line drawing with strokes method, for we do not need pencil to draw the shadow.

In the line drawing from strokes method, there are some things we need notice. One thing is that artists always draw lines with breaks, not a long line without any break. Another thing is that there are always crosses at the junction of two lines. Based on the two important thing, the method of drawing lines from strokes was born. When drawing strokes at a point, we determine the direction, length, and width in a pixel classification and link process based on a unified convolution framework.

Classfication We first transform the input image into grayscale version. Then compute the gradients of it, yielding magnitude

$$G = ((\partial_x I)^2 + (\partial_y I)^2)^{\frac{1}{2}} \tag{4}$$

where I is the grayscale input, ∂_x and ∂_y are gradient operators in two directions, implemented by forward difference. Then we use local information to do the classification. We choose 8(depending on the case, it may changed to 6 or other value) directions, each is 45 degrees apart and denote the line segment as $L_i (i = 1, 2, ..., 8)$ representing the 8 directions and the length of L_i is $\frac{1}{30}$ of the height or width. So the responding map for a certain direction i is

$$C_i = L_i * G \tag{5}$$

where $*$ is the convolution operator. Finally, the classification is performed by choosing the maximum value among the response map in the 8 directions. This step is written as

$$C_i(p) = \begin{cases} G_p, & if argmin_i[G_{i](p)} = i \\ 0, & otherwise \end{cases} \tag{6}$$

where p indexes pixels and C_i is the magnitude map for direction i.

Line Shaping We generate lines also by convolution given the map set C_i and this is expressed like this

$$S^{'} = \sum_{i=1}^{8} (L_i \otimes C_i) \tag{7}$$

Convolution aggregates nearby pixels along direction L_i, which links edge pixels that are even not connected in the original gradient map.

Using these two method, we get freehand sketches, which are more like what ordinary people draw than using Canny.

3.2. Detailed Image Synthesis Model

In DIS model, We adapt the *conditional Generative Adversarial Networks* in [10], which is denoted by $cGAN(\cdot)$. This architecture fits our work because cGAN runs fast and its precision is high enough for artistic synthesis. In the training process, $cGAN$ receive the pairs of reality and sketch images, $[\bar{\boldsymbol{S}}, \boldsymbol{R}]$, generated in Section 3.1 to train the model. In the test process, the DIS model receives the real freehand sketch image $\hat{\boldsymbol{s}}$ and then generates a detailed informative image $\boldsymbol{d}$.

The loss function of CGAN is:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{(s,r)\sim[\bar{\boldsymbol{S}},\boldsymbol{R}]}[logD(s, r)] + \\ & \mathbb{E}_{s\sim\bar{\boldsymbol{S}},z}[log(1 - D(s, G(s, z)))], \end{aligned} \tag{8}$$

where $G(\cdot)$ represents the generator, $D(\cdot)$ represents the discriminator.

In our work, G is set to use the U-Net [18], which is similar to the Encoder-decoder architecture.

In order to make the results closer to reality, we adapt the objective cGAN with L_1 distance inspired by [17]. By doing this, the generator is able to generate image more similar to the ground truth.

The compound objective is:

$$\mathcal{L} = \mathcal{L}_{cGAN}(G, D) + \lambda \mathbb{E}_{(s,r)\sim[\bar{\boldsymbol{S}},\boldsymbol{R}],z}[\|r - G(s, z)\|_1] \quad (9)$$

Then, the objective generator by the optimization is:

$$\bar{G} = \arg\min_G \max_D \mathcal{L}(G, D) \quad (10)$$

with the objective $\bar{G}$ by training, we can generate the detailed image d by $d = G(\hat{s})$.

3.3. Adaptively Weighted Artistic Style Transfer

The *Adaptively Weighted Artistic Style Transfer* (AWAST) model recieves the detailed informative image d synthesized in Section 3.2 and then given set of artistic style samples A. The features of the detailed image d and the artistic samples $a \in A$ are extracted from the pretrained VGG19 [20] net. An novel weight calculating algorithm based on PageRank is used to combine the features. Then, a random noise image x was optimized to our objection image g, which combines both the low and high level features of d and A.

The first step of AWAST is to feed d and every a into the pretrained VGG19 network inspired by the artistic style algorithm [5]. After this process, we obtain an activated feature map $\boldsymbol{F}_l$ on each convolutional layer$l$. Every activated feature map represents the different levels of features. In the beginning convolutional layer, more low-level information about color, pattern and details are extracted. And the high-level information like the distribution and shape is store in the latter convolutional layer.

We denote the set of activated feature maps for every artistic sample $a \in A$ as $\mathbb{F}_a$: $\mathbb{F}_a = \bigcup\{\boldsymbol{F}_{a,l}\}$. Similarly, we denote the extracted features of informative image d as $\mathbb{F}_d = \bigcup\{\boldsymbol{F}_{d,l}\}$ and $\mathbb{F}_x = \bigcup\{\boldsymbol{F}_{x,l}\}$ for noise image x.

Our approach is optimizing x by iteration t to make sure it is similar to both d and a in A. So, we define the loss function of the optimization as:

$$\mathcal{L}(d, A, l, t) = \alpha \mathcal{L}_1(l, d, x_t) + \beta \sum_{l, a\in A} \bar{\omega}(l, a) \cdot \mathcal{L}_2(l, a, x_t), \quad (11)$$

where $\mathcal{L}_1(d, x_t)$ is the content loss between d and x_t, $\mathcal{L}_2(a, x_t)$ is the style loss between a and x_t. $\bar{\omega}(l, a)$ is the *adaptive style weight* (ASW) of sample a and layer l. α is the overall content weight and β is the overall style weight.

The content loss $\mathcal{L}_1$ is defined as the squared error loss between d and x_t:

$$\mathcal{L}_1(l, d, x_t) = \frac{1}{2} \sum_{i,j} (F^l_{i,j,d} - F^l_{i,j,x_t}). \quad (12)$$

The style loss is defined with the Eccentric covariance matrix(Gram) matrix G^l in different layer l, $G^l_{i,j} = \sum_k F^l_{i,k} F^l_{j,k}$. Our optimization problem is converted to minimize the difference between the Gram matrix of a and x_t. We define the style loss as

$$\mathcal{L}_2(l, a, x_t) = \frac{1}{4 n_l w_l h_l} \sum_{i,j} (G^l_{i,j,a} G^l_{i,j,x_t}), \quad (13)$$

where $\frac{1}{4 n_l w_l h_l}$ is the normalization coefficient in order to make $\mathcal{L}_2$ compatible with $\mathcal{L}_1$

Recall that different $\boldsymbol{F}_{a,l}$ can represent different features, in another word, the painting skills of different artists. For example, in Pablo Picasso's painting, the high-level feature would contain more sharp, irregular geometry and lines while the low-lever feature would include deeper colors and smooth pattern. For Claude Monet's paiting, the features are almost opposite.

To make the artistic style painting synthesized from multiple artist painting samples $a \in A$ being more delicate and appealing, we introduce the *adaptive style weight* (ASW) to balance the style futures from different samples and layers. Since multiple painting samples are input to our system, various styles and skills from different artists will pile up. In order to emphasize the most common used skills, factions, and color preference, etc. and weaking some minor and unimportant skills from these artists , we use ASW to balance the importance of them.

To calculate the ASW, we build a style similarity network and use the undirected PageRank algorithm [7] to define the ASW. We define the difference matrix Δ between different $F_{a,l}$ by the squared error:

$$\Delta^l_{p,q} = \frac{1}{2} \sum_{i,j} (F^l_{i,j,a_p} - F^l_{i,j,a_q})^2. \quad (14)$$

We define the similarity matrix Γ as:

$$\Gamma^l_{p,q} = \frac{\max_{i,j} \Delta^l_{i,j}}{\Delta^l_{p,q}}. \quad (15)$$

The matrix Γ^l is symmetric, full-rank and the diagnal element is zero. We build a fullly connected undirected graph according to Γ^l. The weight of edges in this graph is defined as: $\mu^l_{i,j} = \Gamma^l_{i,j} / \sum_j \Gamma^l_{i,j}$. The weight of nodes in this graph is denoted as PR, representing the importance of this node in the PageRank algorithm.

The iteration formulation of undirected PageRank is:

$$PR_t(p_i) = \frac{1-\theta}{N_s} + \frac{\theta}{N_s} \sum_{p_j \in E(p_i)} \mu_{p_i,p_j} PR_{t-1}(p_j), \quad (16)$$

where p_i and p_j are nodes in the graph. θ is the damping factor of PageRank algorithm. $N_s = |A|$ is the number of nodes. $E(p_i)$ is the set of nodes that connecting with p_i

Then, the matrix $\overline{PR}$ is obtained after the PageRank algorithm converged. Then, we define the ASW with a sigmoid mapping:

$$\omega(l,a) = (1+exp(2-\frac{4(\overline{PR} - \min(\overline{PR}))}{\max(\overline{PR}) - \min(\overline{PR})}))^{-1}, \quad (17)$$

By normalization, the ASW $\bar{\omega}(l,a) = \frac{1}{N_l N_s}\omega(l,a)$ is obtained.

Then, by minizing the overall loss in Eqn.3.3, the random noise image x will converge to the objection image g.

4. Experiments

To evaluate the performance of our scheme, experiments of our scheme and baselines are set as below.

Datasets: For the SIE and the DIS model, we use the Cats dataset on Kaggle. However, we didn't use the whole dataset, we only choose 2620 images in the dataset, extracting some images on which there are mainly human beings or something like that but not mainly cats. We also use the Oxford Buildings Dataset. We randomly choose 4202 building images in this dataset.

For the AWAST model, we mainly choose style images from Google Image for the ink painting style, the Picasso style, the Van gogh style, the watercolor style and Ukiyo-e style. For the Onmyoji style, we choose images of a mobile game named Onmyoji of Netease Company in China.

Environment: All the experiments are conducted on a PC device (Intel(R) Core(TM) i7-6900K 3.2GHz, 16GB memory, NVIDIA GeForce(R) GTX 1080 Ti) and are implemented in Python 3.6.2[1].

4.1. Training

Because our model is the combination of the three models mentioned above, we now describe the training process in three main steps corresponding to the three models.

In the SIE model, for smoothing, we set the parameters lamda to be 0.02, kappa to be 1.2. And for pencil sketching, we set the length of convolution line to be 7, the width of the stroke to be 0.1, the number of directions to be 10 for the cat dataset. For the building dataset, we set the length of convolution line to be 7, the width of the stroke to be 0.1, the number of directions to be 10. Because for some image in the two datasets, there are some pictures too complex for freehand drawing, we deleted these complex pictures.

The DIS model receives the training datasets from SIE, which are pairs of sketch images and the reality images. These images are resized to 256×256 to accelerate our training speed. Then, we set the learning rate $lr = 0.0002$, the generator use the unet 256 network. The initialization of network is set to use xavier initializer. Both the input and output channels are set to be 3.

The AWAST model receives the 256×256-size images synthesized by the DIS model and the style image samples illustrated in Section 4. The parameters α and β in Eqn.3.3 are set as 8 and 500. The dumping factor of PageRank θ in Eqn. 3.3 is set to be 0.85 and the convergent accuracy of PageRank is 0.0001. The learning rate is 1 and parameters β_1 and β_2 for Adam optimizer are 0.9 and 0.999. The iteration of optimization is set to be 2000.

4.2. Baselines and Analysis

4.2.1 Baselines

We set some baselines to show our method of assembling the model works best:

Baseline 1: Shown in Fig.4: reality image → Canny image → style image.

Baseline 2: Shown in Fig.5: reality image → smoothing image → sketch image → style image.

Baseline 3: Shown in Fig.6: reality image → smoothing image → Canny image → detailed image → style image.

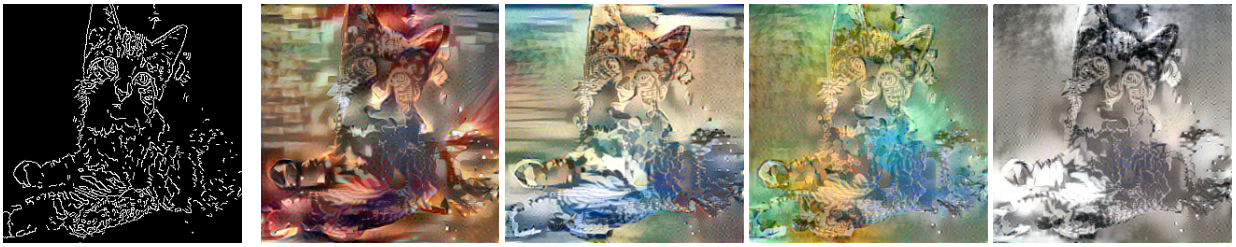
Our scheme: In Fig.7: reality image → smoothing image → pencil sketch → detailed image → style image.

4.2.2 Analysis of baselines

As shown in Fig.4, the synthesized stylized image is quite blurring. There are many noisy points that can't converge to the optimizing object. And the features of the content are not extracted successfully. So there are some areas just copy the pattern of the style samples. For example, in Fig.4c, near the cat's feet, the pattern is just like the tide which comes from the style samples of Ukiyo-e style. This is because the information in the edge picture is not enough to push the AWAST synthesize the objetive result, so as to people's freehand sketch. So, the information generating process, DIS, is needed.

In Fig.5, we can see that by smoothing and pencil sketch extraction, the results are much better. The problems of pattern copying like Baseline 1 are improved and the shape of the cat looks more clear. At the same time, the style image is not so blurring as Baseline 1. These improvements suggest that the smoothing and pencil sketch extraction are quite helpful to extract the key information in the reality image while weak the noise. By comparing Fig.4a and Fig.5a,

[1] All the code will be publicly available after the review process.



(a) Canny edge (b) Onmyoji (c) Ukiyo-e (d) Van Gogh (e) Chinese painting

Figure 4: Baseline 1: Extract edge with canny algorithm and feed AWAST directly

(a) Sketch (b) Onmyoji (c) Ukiyo-e (d) Van Gogh (e) Chinese painting

Figure 5: Baseline 2: Smoothing, extracting pencil sketh, and feeding the sketch to AWAST directly

(a) Detailed image (b) Onmyoji (c) Ukiyo-e (d) Van Gogh (e) Chinese painting

Figure 6: Baseline 3: Smoothing, Canny, DIS, and feeding AWAST with the detailed image

(a) Detailed Image (b) Onmyoji (c) Ukiyo-e (d) Van Gogh (e) Chinese painting

Figure 7: Our scheme: Smoothing, extracting of pencil sketch, DIS, and feeding AWAST with the detailed image

the sketch in Baseline 2 is more like human freehand sketch on the continuity and directions.

Fig.6 represents the final results of Baseline 3, which includes smoothing, canny, DIS, and AWAST. Compared to Baseline 1, the results are much more beautiful. The shape of the cat is more clear and the problem of pattern copying is solved. The shade of the styles is also more natural. The details are also more clear. In Fig.6e, for example, the eyes of the cat become yellow which is different from the color of its body. This means the detail of the cat's eyes is noticed and emphasize by the AWAST model.

Our final scheme is shown in Fig.7 including smoothing, pencil sketch extraction, DIS, and AWAST. The performance improves a lot comparing to Baseline 3. For example, in Fig.7e, eyes of the cat are much more vivid and the patterns of fur and beard on its head and body are detailed.

(a) Reality (Golden Gate Bridge) (b) Result of Gatys et al. (c) Result of Ours

Figure 8: Performance analysis of AWAST

(a) Freehand sketch (b) Detailed image (c) Onmyoji (d) Ukiyo-e (e) Van Gogh (f) Chinese painting

Figure 9: Test result of real freehand sketch of cat

On Fig.7b, the style transfer is active and does not break the shape of the cat while on Fig.6b, the style mix the cat's body and the environment.

4.2.3 Analysis of AWAST

We use 6 images in Fig 10 of Ukiyo-e style to analyze the performance of AWAST. However, one image of them is coverd by black color. The experiment result when iterations $iter = 2500$ is shown in Fig.8. The result of Gatys et al. in Fig.8b appears to have many black areas, which is attributed to taking the patterns of black. This is because the blending method of them is simply taking the average of multiple features. Our method in Fig.8c is steadier and more natural, because ASW based on PageRank algorithm weaks the weight of the unusual styles like black images and emphasizes the right styles in the other 5 images.

Figure 10: Images of Ukiyo-e style to analyze AWAST

4.3. Test

For convenience, all the experiment above is based on the test set which is generated by SIE model. However, our goal is to transform a real freehand sketch to an artistic painting. So we take an $A4$ paper and a pencil and draw a cat sketch by hand, which is shown in Fig.9a. This is quite easy actually. We simply take a photo with our mobile-phone and then feed it into the DIS and the AWAST model. To make the result compared fairly, we still use the same style samples. the result is shown in Fig.9.

5. Conclusions

To be an artist is always many people's dream. In this paper, we propose LINE ARTIST to synthesize appealing paintings with freehand sketch. To achieve this goal, we propose the SIE model to smooth the images and extract the sketch to build new datasets. The DIS model based on GAN allow the sketch drew by users to generate more details and looks natural. The AWAST model propose an novel algorithm based on PageRank to adaptively combining styles from multiple painting samples. Our results are prove to be vivid, artistic, and adapted to different styles stably. There are also some issues need to be improved. In the DIS and the AWAST model, the synthesis is not real-time. However, LINE ARTIST still has the potential to become an powerful entertainment APP and assistant of artists.

References

[1] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *arXiv preprint arXiv:1707.09405*, 2017.

[2] P. Dollar and C. L. Zitnick. Structured forests for fast edge detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[3] M. Elad and P. Milanfar. Style transfer via texture synthesis. *IEEE Transactions on Image Processing*, 26(5):2338–2351, 2017.

[4] W. Gao, X. Zhang, L. Yang, and H. Liu. An improved sobel edge detection. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 5, pages 67–71. IEEE, 2010.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[6] C. I. Gonzalez, P. Melin, J. R. Castro, O. Mendoza, and O. Castillo. An improved sobel edge detection method based on generalized type-2 fuzzy logic. *Soft Computing*, 20(2):773–784, 2016.

[7] V. Grolmusz. A note on the pagerank of undirected graphs. *arXiv preprint arXiv:1205.1960*, 2012.

[8] J. Ho Lee, I. Choi, and M. H. Kim. Laplacian patch-based image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2735, 2016.

[9] F. Huang, B.-H. Wu, and B.-R. Huang. Synthesis of oil-style paintings. In *Pacific-Rim Symposium on Image and Video Technology*, pages 15–26. Springer, 2015.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[11] S. Kumar, R. Saxena, and K. Singh. Fractional fourier transform and fractional-order calculus-based image edge detection. *Circuits, Systems, and Signal Processing*, 36(4):1493–1513, 2017.

[12] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

[13] C. Lu, L. Xu, and J. Jia. Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, pages 65–73. Eurographics Association, 2012.

[14] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[15] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[17] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[19] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. *arXiv preprint arXiv:1612.00835*, 2016.

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

[21] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[22] Y. Wei, Z. Wang, and M. Xu. Road structure refined cnn for road extraction in aerial image. *IEEE Geoscience and Remote Sensing Letters*, 14(5):709–713, 2017.

[23] S. "Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[24] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, page 174. ACM, 2011.

[25] F. Zhang, L. Dai, S. Xiang, and X. Zhang. Segment graph based image filtering: fast structure-preserving smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 361–369, 2015.

[26] S. Zhang, X. Gao, N. Wang, J. Li, and M. Zhang. Face sketch synthesis via sparse representation-based greedy search. *IEEE transactions on image processing*, 24(8):2466–2477, 2015.