Scribble-to-Painting Transformation with Multi-task Generative Adversarial Networks

Jinning Li
Department of Computer Science and Engineering
Shanghai Jiao Tong University
lijinning@sjtu.edu.cn

Yexiang Xue
Department of Computer Science
Purdue University
yexiang@purdue.edu

Abstract

We propose the Dual Scribble-to-Painting Network (DSP-Net), which is able to produce artistic paintings based on user-generated scribbles. In scribble-to-painting transformation, a neural net has to infer additional details of the image, given relatively sparse information contained in the outlines of the scribble. Therefore, it is more challenging than classical image style transfer, in which the information content is reduced from photos to paintings. Inspired by human cognitive process, we propose a multi-task generative adversarial network, which consists of two jointly trained neural nets – one for generating artistic images and the other one for semantic segmentation. We demonstrate that joint training on these two tasks brings in additional benefit. Experimental result shows that DSP-Net outperforms previous models both visually and quantitatively. DSP-Net is also less sensitive to the mode-collapse problem and trains faster. In addition, we publish a large dataset for scribble-to-painting transformation.

1. Introduction

Recent advancements in deep neural networks have brought tremendous successes in neural style transfer, which converts photos into artistic oil paintings, mimicking the styles of Vincent van Gogh or Claude Monet [10]. Neural style transfer is among the first few deep learning technologies which are commercialized as cellphone applications [32]. Users are excited at these applications, since they allow them to generate self-portraits as if they were painted by famous artists. However, these applications require a photo to generate an artistic painting. An more interesting task is to eliminate this requirement, giving users full opportunities to *create* paintings on scenes that do not exist in the real-world.

In this paper, we consider a novel application, which produces artistic paintings based on user-generated scribbles (see Figure 1 for an example). In this way, users can *create* paintings as they like, far beyond the restrictions posed by real-world photos. Scribble-to-painting transformation poses novel challenges, which are not encountered in neural style transfer before. As shown in Figure 1(a), a scribble often only contains strokes that outline the objects in a scene, whose information content is much more sparse than a real-world image. As a result, a scribble-to-painting neural network has to *infer* missing details, such as the colors of the surfaces or the geometric shapes of the objects, based on outlines from the scribble. This is different from the case of photo-to-painting neural style transfer, where the neural network is to *summarize, reduce and adapt* the information

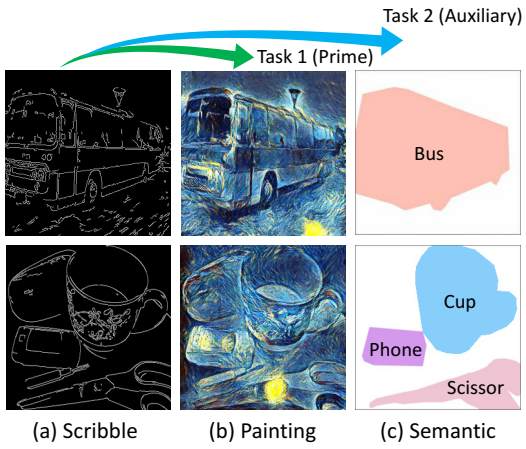


Figure 1: Examples of scribble-to-painting transformation. (a) Scribble images as inputs. (b) Artistic paintings as outputs. (c) An auxiliary object detection and segmentation task that helps to generate better paintings in (b). In the first example, recognizing the object as a bus helps to color its wheels and windows. In the second example, the rectangular object is recognized as a phone given the semantic information of other objects, which helps to color it properly.

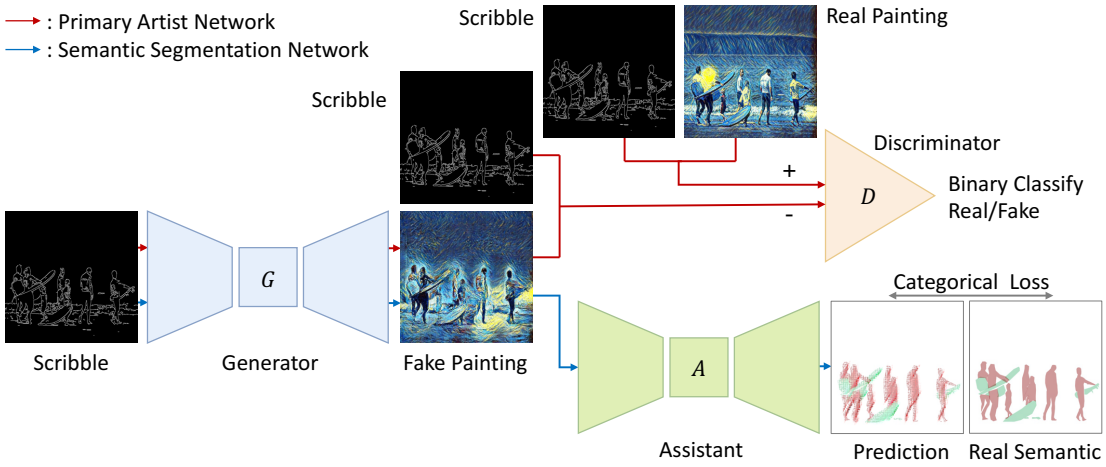


Figure 2: Overview of the multi-task Dual Scribble-to-Painting Network (DSP-Net). It contains a primary artist network and a secondary semantic segmentation network. The primary artist network is a conditional generative adversarial network, which generates paintings in artistic style based on scribble images. The secondary semantic network learns to recognize the semantic segmentation of scribbles. These two networks reuse the parameters of the first few layers (shared layers are marked as the generator here).

within a real-world photo into an artistic painting. We believe the aforementioned differences are fundamental, precluding existing neural style transfer approaches to succeed in our scribble-to-painting application.

We use multi-task learning to address the task of scribble-to-painting transformation. We first ask ourselves *how human painters work from the basis of a scribble*. Human painters first *recognize* each object based on the scribble, and then start to add details. For example, in the first example of Figure 1, recognizing the entire object as a bus helps to color its wheels and windows. Similarly, in the second example, a human being has to recognize the rectangular object as a cell-phone before he can color the object properly. Inspired by this, we build *Dual Scribble-to-Painting Network* (DSP-Net), which simultaneously solves the problems of scribble-to-painting transformation and object detection and segmentation. The whole architecture of DSP-Net is shown in Figure 2, which composes of two subnetworks. A primary network – the artist network – generates oil paintings based on scribble images, while the secondary network – the semantic network – recognizes and segments objects from the scribble image. The primary and secondary networks share the first few layers for feature extraction. The core idea is that the training of the secondary network helps the shared layers to capture a better semantic representation, therefore assisting the task of the primary network.

We demonstrate that joint training on these two tasks brings in additional benefits. Experimental result shows that DSP-Net outperforms previous models both visually and quantitatively. Visually, as shown in Figure 3, the paintings produced by DSP-Net contains more detailed textures which match better with the styles of the target paintings. Quantitatively, compared to previous models, our model matches better to the target artistic paintings under the metrics of the content-mismatching loss and the style-mismatching loss based on a pre-tained VGG-19 network. DSP-Net is also less sensitive to the mode-collapse problem and trains faster.

We make another contribution by generating and publishing a benchmark dataset for scribble-to-painting transformation. Another problem in building a scribble-to-painting transformer is to obtain large and high-quality datasets. Unfortunately, the number of available oil-paintings and scribble images online are far less than the number of photos. Our insight is to rely on large photo image datasets and existing approaches for line extraction and neural style transfer. In particular, we start with the COCO dataset. We extract lines of the photos to form the scribble images, and then we use existing neural style transfer algorithms [16] to transform photos in the COCO dataset to oil paintings. We collected a large dataset of more than 5000 pairs of scribble and painting in this way. We will publish this dataset online as an additional contribution.

2. Scribble to Painting Transformation

Scribble-to-painting transformation is an interesting but challenging task. See Figure 1 for an example, where scribble images in (a) are transformed into stylized paintings in (b). Users can produce beautiful artistic paintings based on simple scribbles using this application, without restricting

to real-world scenes captured by a camera.

Mathematically, we use x_i to denote one scribble image as an input, which is in binary scale. Scribble images are drawn from an underlying space $\mathcal{X}$. An artistic painting as an output is denoted as y_i, which is drawn from an underlying space $\mathcal{Y}$. The training data $d = \{(x_1, y_1), ..., (x_m, y_m)\}$ is drawn from the joint space $\mathcal{X} \times \mathcal{Y}$. The objective of scribble-to-painting transformation is to find a mapping $h \in \mathcal{H}$ from $\mathcal{X}$ to $\mathcal{Y}$, which best mimics the mappings in training dataset. Let $\mathcal{L}$ be a loss function that measures the difference between $h(x)$ and the target y. $\mathcal{L}$ can be, for example, the L1 loss. The objective of scribble-to-painting transformation is to find a function $h \in \mathcal{H}$, which minimizes the expected loss:

$$\min_h \ \mathbb{E}_{(x,y)\sim P_{xy}(x,y)} \left[\mathcal{L}(h(x), y)\right]. \tag{1}$$

Scribble-to-painting transformation is related to neural style transfer, where real photos are transferred into artistic paintings. Nonetheless, it is conceivably more challenging. The main challenge comes from the sparse information content of the scribbles. See Figure 1(a). There are many blank areas in the scribbles, which contain no information.

At the same time, many state-of-art style transfer models work from photos, which are rich in terms of information content. Therefore, the classical style transfer application is a *information reduction* task, where the information-rich photos are converted into information-sparse paintings.

On the contrary, in scribble-to-painting transformation, a neural net has to *infer* missing details from the scribble, such as the colors of surfaces and the geometric shapes of objects, etc. It is a process of *adding information*. We believe that this is a fundamental difference, which precludes direct application of successful approaches in style transfer to scribble-to-painting transformation.

3. Dual Scribble-to-Painting Network

As discussed above, the sparsity of the information content in the scribble images poses a significant challenge in scribble-to-painting transformation. To solve this problem, we get inspired from human cognitive process: *how do human painters work from the basis of a scribble*? Before human painters start adding details, they first *recognize* each object from the scribble. The recognition process helps human painters to decide the correct colors, geometry and textures of the artistic painting.

Inspired by this, we explore *multi-task learning* for scribble-to-painting transformation. We employ a secondary semantic segmentation task to assist the primary task of scribble-to-painting transformation. In this setting, besides the input scribbles x_i's and the output paintings y_i's of the primary task, there is an additional output z_i, which is a semantic map of detected objects in the scene (See Figure 1(c)). As a result, the dataset consists of a set of triples $d = \{(x_1, y_1, z_1), ..., (x_m, y_m, z_m)\}$ drawn from an underlying distribution $P_{x,y,z}(.)$ defined over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

We propose the *Dual Scribble-to-Painting Network* (DSP-Net), where both the primary and the secondary tasks are trained simultaneously (See Figure 2). The primary neural network – the artist network – is a generative adversarial network. It contains a generator G to transform the input scribble image x_i to an *fake* artistic painting $G(x_i)$. The goal of the generator is to fake $G(x_i)$ so that it is hard for the discriminator D to tell apart from the real painting y_i. The discriminator, on the other hand, tries to separate the real paintings from the fake ones. After the generator and discriminator networks reach an equilibrium, the generated paintings will look similar to the real ones.

The secondary neural network – the semantic network – share the generator network G with the first neural net. Its task is to detect and segment objects from the scribble images, therefore assisting the primary artist network to generate paintings with correct semantics. The semantic network contains a sub-network named assistant A, which maps the output of the generator $G(x_i)$ into a semantic segmentation map $A(G(x_i))$, which best mimics the output of true z_i. In other words, we would like to minimize the distance between $A(G(x_i))$ and z_i with respect to a criterion $\mathcal{R}$, such as the smooth L1 loss function [11] used in our paper. Combining these two neural networks, the objective of DSP-Net can be formulated as:

$$\begin{aligned}\min_A \max_D \min_G \ & \mathbb{E}_{(x,y)\sim P_{xy}(x,y)}[\log(D(x,y))]+ \\ & \mathbb{E}_{x\sim P_x(x)}[\log(1 - D(x, G(x)))]+ \\ & \mathbb{E}_{(x,z)\sim P_{xz}(x,z)}[\mathcal{R}(A(G(x)), z)].\end{aligned} \tag{2}$$

Notice that both the scribble x and painting y are fed to the discriminator D. This design allows the discriminator to detect abnormal "switching" behavior of the generator. Suppose the generator G transform a scribble of a dog into a painting of a cat. The discriminator D is able to discover this fact because the original dog scribble is also fed into D.

The artist and semantic networks cooperate with each other by sharing the parameters of G. G can be treated as a feature extraction network for the semantic task while it is the backbone of the artist network. Figure 2 illustrates the general architecture. When training the primary artist network, D tries to maximize the first two terms and G tries to minimize the second term in Eqn.2 adversarially. When training the secondary semantic network, both the assistant network A and the generator G are optimized cooperatively with respect to the third term in Eqn.2. The training of the two networks are interleaved until convergence.

3.1. Multitask Learning

Our DSP-Net architecture is motivated by human cognitive process. When human painters are given a scribble image, they first make sense of the scribble, identifying each

object from the image. In this way, they can decide the correct colors, textures, etc, for each object in the image.

For example, in the example shown in the second row of Figure 1, the mobile phone on the left is hard to be recognized unless given the correct context information. In fact, its rectangular shape can be interpreted as a microwave or a television. It is the relative size of the box compared to other objects that makes us believe that it is a cellphone. Therefore, without the assistance of the semantic network, the primary artistic network will get confused on the correct way to color this object. We believe this is the reason why we often see unreal or blurred paintings in baseline models.

Empirically, the secondary semantic segmentation network also speeds up the training because the semantic segmentation task appears to be easier to train than the primary task. The semantic task can also serve as a regularizer that helps reduce overfitting.

3.2. Primary Artist Network

The objective of the primary artist network is to synthesize an oil painting given a scribble image. We use an adaptation of the network introduced in [15, 27] as our primary network, which is a generative adversarial network [12] (GAN).

The generator and discriminator of the primary network are shown in Figure 1. The generator $G : \mathcal{X} \rightarrow \mathcal{Y}$ is a mapping from scribbles to paintings. The discriminator $D : \mathcal{Y} \rightarrow \{0, 1\}$ is a mapping from artistic paintings to a binary outcome. Generator G receives a scribble image x and try to generate a fake painting $G(x)$ that cannot be distinguished as a real painting by the discriminator. In this paper, we use a variant of the encoder-decoder architecture in [15] as the generator, which first encodes the features of scribble images into a low-dimensional representation and then decodes it to an artistic painting.

Discriminator D is trained to differentiate whether a given painting is synthesized by generator or it is a real one. The fake painting should be classified as 0 and the real painting should be classified as 1. In this paper, we adapt the Markov discriminator introduced in [19] as our architecture. The min-max objective for both the generator G and the discriminator D in the primary artist network is:

$$\begin{aligned}\max_D \min_G \; & \mathbb{E}_{(x,y)\sim P_{xy}(x,y)}[log(D(x,y))]+ \\ & \mathbb{E}_{x\sim P_x(x)}[log(1-D(x,G(x)))].\end{aligned} \tag{3}$$

When the competitive training between the generator and the discriminator achieves an equilibrium, the generator will learn to generate relatively realistic paintings.

3.3. Secondary Semantic Segmentation Network

Leveraging multi-task learning, we introduce a secondary semantic segmentation network to detect and segment objects based on scribble images. Both the primary and secondary tasks are trained simultaneously while sharing the parameters of G. This secondary semantic segmentation task is shown using blue arrows in Figure 2.

We apply another encoder-decoder architecture for the secondary task, which is named as an assistant A. Given a scribble image x, the secondary network is trained by minimizing the segmentation loss between the generated semantic segmentation $A(G(x))$ and the ground truth z:

$$\min_A \mathbb{E}_{(x,z)\sim P_{xz}(\mathcal{X},\mathcal{Z})}[\mathcal{R}(A(G(x)), z)]. \tag{4}$$

Here, $\mathcal{R}$ is a loss function that penalizes the difference between $A(G(x))$ and z. In this paper, we use smooth L1 loss proposed in [11] as our loss function. Different from the case in the primary task where generator and discriminator have competing objectives, the generator G and A share the same objectives in the secondary task.

During training, the primary artist network and the secondary semantic segmentation network are optimized in an interleaved manner.

4. Dataset Generation

Another challenge in scribble-to-painting transformation is to obtain large and high-quality dataset. However, the number of available oil paintings and scribbles are quite limited. It is even harder to find the pairs of scribble and painting for a supervised transformer. What is more, in order to train the semantic segmentation task simultaneously in the multitask learning setting, the semantic images of corresponding paintings are necessary. However, currently there is no manually labeled semantic segmentation for oil paintings.

In this paper, we build a triple dataset including scribbles, paintings and semantic images based on COCO dataset [21]. COCO dataset provides the pairs of photo p_i and its corresponding semantic image z_i. We build scribble x_i from p_i with a simple but effective Canny edge extracting algorithm [1]. The built scribbles are shown in Figure 3a. Painting y_i is generated with a fast neural style network [16] pre-trained on COCO dataset. y_i is obtained by feeding photos p_i into the pre-trained network. The synthesized paintings are shown in Figure 3b. Along with the semantic image z_i which is manually labeled for photo p_i on COCO dataset, the triples (x_i, y_i, z_i) are induced.

5. Related Works

Interaction between Art and Machine Learning The interaction of computer science and art is always a charming research topic. There are already many interesting applications of machine learning on art, such as interactive music palettes [28], sketch generating [13], and autocoloring [17]. In this paper, we focus on a artistic task transforming scribbles into paintings. This task is related to style

transfer task since some current style transfer models can also be applied to transform the scribbles into other images such as Neural Style [9] and Pix2pix [15]. However, the performance of these models is limited because the sparse representation of scribbles are different from other images. SketchyGAN in [3] adopts a data augumentation and techniques and a MRU network module to synthesize photos with sketches, which improves the realism on sketch-to-photo synthesis task. However, SketchyGAN needs users to claim a label of the object they are going to draw, which limits the application of SketchyGAN.

Image Style Transfer Before 2015, almost all the style transfer models used for painting synthesis are based on statistics of image textures such as [26, 30]. In 2015, Gatys et al. proposed an amazing neural style algorithm [9] using pre-trained very deep convolutional networks [31] as a feature extraction tool to recognize the texture and transfer the styles. The following models such as [33, 18] try to train a feedforward neural network to transfer the style of given images. There are also other variants. In [7] the author explored the combinations of multiple styles. In [14] the author used the adaptive instance normalization to constraint the style textures. Other models in [2, 16] studies how to transfer style in real time.

Generative Adversarial Networks Some models focus on using generative adversarial network to generate images such as the DCGAN in [27] and the following models in [25, 6]. Different from previous works, Pix2pix model [15] uses images as input instead of a random Gaussian vector to train a conditional GAN. After that, the following works try to improve the performance of conditional GAN by changing the architectures. In [34], the author proposed an architecture with multi-scale generator and discriminator to increase the resolution of synthesized images. CycleGAN in [37] and DualGAN in [36] build an cycle-consistent architecture to train two GANs in both direction. Its variants include StarGAN in [4] training multiple generative adversarial networks simultaneously to transfer the images across multiple domains.

Multi-task on Generative Adversarial Networks Some other works introduce the idea of multi-task learning to improve the performance of generative adversarial networks. CoupleGAN in [23] trains two GANs simultaneously to solve two different tasks sharing the parameters of both generators and discriminators. In [22], the author extended the CoupleGAN into a unsupervised training settings. In [35], the author concatenated two GANs for structure image generating task and indoor photo generating task.

6. Experiments

In this section, we are going to evaluate the experimental performances of DSP-Net and some existing models widely used in image transformation. We will compare the synthesized paintings both visually and quantitatively.

6.1. Dataset

We use the triples of scribbles, paintings, and semantic images obtained using the methods introduced in Section 4. 5000 images from COCO dataset [21] are ramdomly sampled to generate them. We split the dataset into 4500 images for training and 500 images for test. The oil painting *The Starry Night* by Vincent van Gogh is used as style image.

6.2. Baselines

Neural Style Neural style algorithm is proposed in [10], where a pre-trained VGG-19 network is used to extract the features of an image within different levels. A random noise is optimized by iteration to look alike the extracted content features and style features. In the experiment, we use the scribble image as the content and the painting *The Starry Night* as the style. All the hyper-parameters are taken as the default settings.

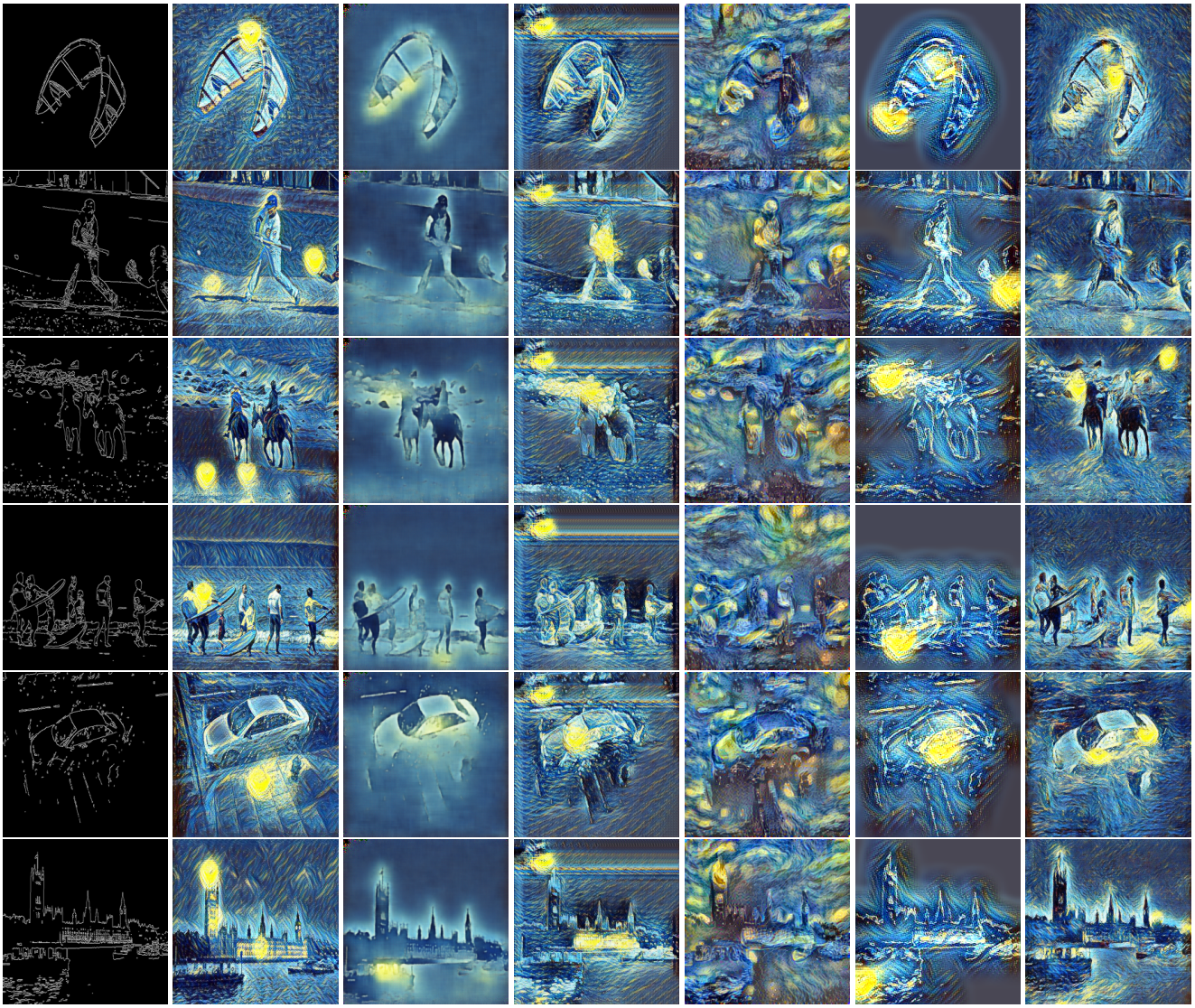
Fast Neural Style We use the scribble as the content and the oil painting *The Starry Night* as the style. We train the Fast Neural Style in [16] to obtain an neural network transformer which receives the scribbles as input to generate artistic paintings. The loss between the generated paintings, content, and style image is calculated in the same way as Neural Style. The hyper-parameters are taken as the default settings.

Pix2pix Pix2pix [15] model use a conditional generative adversarial network to train a generator to transfer images from one domain to another domain. We use scribbles $\mathcal{X}$ as one domain and objective paintings $\mathcal{Z}$ as another domain to train the cGAN. Grid search is used to find the best hyper-parameters. We trained it for 300 epochs on the training set of 4500 samples.

CycleGAN CycleGAN [37] trains two conditional generative adversarial network in both directions between two domains. Although CycleGAN can be trained on unsupervised settings, we trained it with paired scribbles x and objective paintings z for better performance. Grid search is used to find the best hyper-parameters. We trained it for 300 epochs on 4500 samples.

6.3. Visual Analysis

Figure 3 shows the painting generated by DSP-Net and other well-known models used in style transfer task. Our DSP-Net model contains more detailed and real textures which match better with the target style. The first column shows the scribble images x_i. Different from photos, these scribbles are greyscale images and we can find that there are many areas left blank. The second column is the ground truth painting y_i synthesized from photos p_i. Because the photos contain full information, it's reasonable that these synthesized ground truths look the most real and beautiful.



(a) Scribble (b) Ground truth (c) Pix2Pix (d) CycleGAN (e) Neural Style (f) Fast Neural (g) DSP-Net (Our)

Figure 3: The experimental results of DSP-Net and baselines. The first column shows the scribble images used as inputs. The second column shows the ground truth synthesized with photos. The following columns are paintings generated with DSP-Net and other models. The blank area of scribble images will cause blurring or even blank in the generated paintings, especially the CycleGAN model in Figure 3f and the Fast Neural Style model in Figure 3d. The sparsity of scribble image also make Pix2pix model in Figure 3c fails to generate style textures for different objects in the scene.

The following columns are the generated images with style transfer methods and DSP-Net.

The sparsity of scribble images causes serious blurring (See CycleGAN model in Figure 3d and Fast Neural Style model in Figure 3f). Although the CycleGAN model successfully synthesizes parts of the style texture of *The Starry Night* in areas where enough information is provided, it failed to generate the patterns in sparse areas. This is because the generator of CycleGAN feels uncertain about what object should be drawn in those area. Similar problems also happen to Fast Neural Style model. This model can only infer what color and patterns should be applied, within a limited distance from the lines drawn in the scribble images. So the uncertain area is left blank.

The neural style algorithm in Figure 3e is one of the most famous algorithms used in image style transfer task. However, its performance is quite limited in the scribble-to-painting transformation task. It is hard to figure out the objects in the synthesized images. They looks like a recombination of the patterns of *The Starry Night*. This is

understandable because Neural Style algorithm relies on a pre-trained VGG-19 network to extract the features of both the scribble images and paintings. However, the VGG-19 network is pre-trained to extract the features of photos on Imagenet dataset, so that its ability to recognize the lines in scribble images is limited.

It is widely acknowledged that Generative Adversarial Networks are hard to train. The most significant problem is mode collapse. It means that the generator may collapse so that only limited varieties of images or patterns can be generated. For example, there is a slight mode collapse area on the upper left side of Figure 3d generated by CycleGAN. The mode collapse area will always be exactly the same in all the output paintings no matter which scribble is inputted. We trained it several times, however this problem always happens. Same problem also happens to the Pix2pix model. By adjusting hyper-parameters, this problem is finally solved. This may be attributed to the variety of objects in COCO dataset, where 118 categories are included. It is hard for these models to generate corresponding textures for every categories.

Mode collapses are less likely to happen in the setting of DSP-Net. Intuitively, this is because the secondary semantic segmentation task force the generator to generate varieties of paintings which satisfies the constraints of semantic images. The gradient of generator is also less likely to vanish because the gradient from the secondary task will break the local optimum of equilibrium.

The secondary semantic network also speeds up the convergence of primary networks because the semantic segmentation task can be treated as a sub-problem of painting synthesis. It is easier to train. Based on the semantic information provided by this sub-problem, the convergence speed of primary network will be increased in the beginning of training process.

6.4. Quantitative Analysis

We also compare our model quantitatively with some famous style transfer models such as Neural Style, Fast Neural Style, Pix2pix, and CycleGAN. Experimental result proves that our model performs better under the metrics of content mismatching and style mismatching. This means our results are closer to the ground truths with respect to content similarity and style similarity. The result is shown in Table 1.

Inspired by the idea mentioned in the neural style algorithm [10] to use a pre-trained network as feature extractor, we propose two metrics, the content mismatching loss δ_c and the style mismatching loss δ_s, to quantitatively evaluate the results of scribble-to-painting transformation task. The mismatchings of content and style are evaluated between the synthesized paintings and corresponding real paintings. We use a very deep convolutional network [31] (VGG network) pre-trained on Imagenet dataset [5] to encode both the fake and true paintings and extract features. Mismatching losses are calculated on the encoded features of paintings. We use $M^l(x)$ to denote the encoded feature map of x in layer l.

Assume h is the hypothesis of the model to be evaluated. The content loss is defined as the mean squared error between features of $h(x)$ and y extracted from different layers of VGG network,

$$\delta_c(x, y, h) = \frac{1}{2} \sum_l \sum_{i,j} \left[M^l_{i,j}(y) - M^l_{i,j}(h(x)) \right]^2 . \quad (5)$$

In order to calculate the style mismatching loss, we use Gram matrix [8] to capture texture information of different layers between fake painting $h(x)$ and real painting y. The Gram matrix denoted by $E^l_{i,j}$ of an image q on the layer l is calculated as the the inner product between the i-th and j-th vectorized feature maps,

$$E^l_{i,j}(q) = \sum_k M_{i,k}(q)^l M^l_{j,k}(q). \quad (6)$$

After calculating all the gram matrices of y and $h(x)$ on different layer l, the style mismatching δ_s can be described as,

$$\delta_s(x, y, h) = \sum_l \frac{1}{4W_l^2 H_l^2} \sum_{i,j} \left[E^l_{i,j}(y) - E^l_{i,j}(h(x)) \right]^2 , \quad (7)$$

where W_l and H_l are the width and height of the feature map in the layer l of VGG-19 network.

With these two metrics mentioned above, we calculate the average of content mismatching loss δ_c and style mismatching loss δ_s on the test set mentioned in Section 6.1, which contains 500 test samples. We calculate the average mismatching losses on the test dataset by $\bar{\delta}_c(h) = \frac{1}{m} \sum_m \delta_c(x_m, y_m, h))$ and $\bar{\delta}_s(h) = \frac{1}{m} \sum_m \delta_s(x_m, y_m, h)$.

Assume h_k is the hypothesis of the k-th model. We apply normalization to both the content and style mismatching losses by $\Delta_c(h_k) = \bar{\delta}_c(h_k) / \sum_k \bar{\delta}_c(h_k)$ and $\Delta_s(h_k) = \bar{\delta}_s(h_k) / \sum_k \bar{\delta}_s(h_k)$.

The results of normalized mismatching losses of different models and the normalized standard deviation are shown in Table 1, the lower the better.

In the aspect of content mismatching, DSP-Net achieves the lowest mismatching loss of 0.175, which outperforms 5% than the rank-2 Pix2pix model. See the visual result in Figure 3, both the results of Pix2pix and DSP-Net are realistic with respect to the content distinctiveness. However, the blurring and texture mismatching limit the performance of Pix2pix model. The content mismatching of Fast Neural Style is the largest, which is 0.246. This is because Fast Neural Style model fails to generate textures in the area where no scribble is provided.

Table 1: Normalized mismatching losses, the lower the better. The second column is content loss and the fourth column is style loss. Our DSP-Net achieves the lowest mismatching losses of both content and style.

Model	Content Loss	Content Std.	Style Loss	Style Std.
Pix2pix	0.185	0.216	0.447	0.280
CycleGAN	0.194	0.195	0.074	0.146
Neural Sty.	0.200	0.154	0.105	0.103
Fast Neural	0.246	0.270	0.304	0.321
DSP-Net(Ours)	**0.175**	0.166	**0.070**	0.130

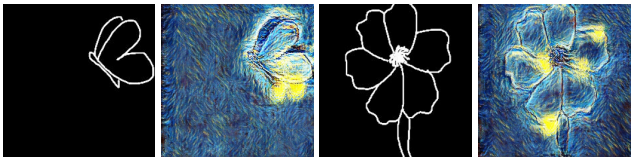
In the aspect of style mismatching loss, the DSP-Net outperforms than other models as well. The mismatching loss of DSP-Net is 5.4% lower than the rank-2 CycleGAN model. This result is also reasonable comparing with the visual result in Figure 3. The textures of results of DSP-Net looks the most similar to the ground truth in Figure 3b. The CycleGAN model also contains some style textures. However, it suffers from the problem of sparse input and mode collapse. The Pix2pix gets the highest mismatching loss because in Figure 3c, almost the whole image remains blur and almost no texture are synthesized.

As for the model stability, the neural style achieves the lowest standard deviation on both the content and style mismatching value. On one hand, neural style does not train a neural network but optimize a Gaussian random image into the target painting with the help of a pre-trained network. On the other hand, the result of neural style in Figure 3e contains almost the same texture and no content information for most test samples. So it's reasonable that neural style algorithm is more stable. Among neural network besed models, the DSP-Net achieve the lowest standard deviation on both content and style mismatching value. This proves that the multi-task setting in the DSP-Net is helpful to the stability of the primary artist network for scribble-to-painting transformation.

6.5. Performance on Real Scribbles

We use edge extractions as our input because they are easier to obtain in large quantity than human-drawn scribbles. The use of edge extractions to mimic scribbles are seen in multiple publications [24, 20]. Our approach generalizes well to human-drawn scribbles, even though it is trained on images of edge extraction.

We train DSP-Net model on our dataset introduced in 6.1 and test it on the Sketchy Dataset consisting of real human-drawn sketches. The results are shown in Figure 4. Both the synthesized details of objects and the patterns of oil paintings are satisfactory in Figure 4b and 4d. Although our model trained on edge extractions already achieves satisfactory results, we also expect additional improvements if further fine-tuning our model on the Sketchy database.



(a) Butterfly (b) Synthesis (c) Flower (d) Synthesis

Figure 4: Syntheses of DSP-Net which is trained on our dataset and tested on the Sketchy Database [29]. Figure 4a and 4c are human-drawn scribbles.

6.6. Human Evaluation for Realism and Aesthetics

Table 2: Human evaluation in terms of content realism and style aesthetics. We report the frequency of each model to be chosen as the most realistic and stylized.

Model	Picked as more realistic and stylized?
Pix2pix	1.49%
CycleGAN	4.23%
Neural Style	27.11%
Fast Neural Style	4.73%
DSP-Net (Ours)	**62.44%**

We complete a survey among students from different majors. We assigned each student 6 scribbles and the corresponding paintings synthesized by different models without telling them which model they were generated with. We asked students to identify the most realistic painting in terms of both content realism and style aesthetics. We collected 402 samples evaluated by 67 students. Results in Table 2 show that a majority of students (62.44%) agreed that the result of our DSP-Net model is the most realistic.

7. Conclusion

In this paper, we focus on a novel task that transforms scribbles to artistic paintings. This task is more challenging than classical image style transfer, because the scribbles contain far less information compared with photos. We use multi-task learning to solve this problem. We introduce the Dual Scribble-to-Painting Network (DSP-Net), which consists of two jointly trained neural networks. The primary network is trained to generate artistic images based on scribbles and the secondary network is for semantic segmentation. Experimental results show that DSP-Net outperforms previous models both visually and quantitatively. DSP-Net is also less sensitive to the mode-collapse problem and trains faster. As an additional contribution, we publish a large dataset for scribble-to-painting transformation. Possible future directions include using multi-task generative adversarial networks to improve the performance of other tasks such as super-resolution and speech recognition.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 4

[2] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 5

[3] W. Chen and J. Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018. 5

[4] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint*, 1711, 2017. 5

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 7

[6] H. Dong, S. Yu, C. Wu, and Y. Guo. Semantic image synthesis via adversarial learning. *arXiv preprint arXiv:1707.06873*, 2017. 5

[7] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2017. 5

[8] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015. 7

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

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 1, 5, 7

[11] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3, 4

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 4

[13] D. Ha and D. Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 4

[14] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017. 5

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 4, 5

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 2, 4, 5

[17] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016. 4

[18] 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. 5

[19] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016. 4

[20] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3158–3165, 2013. 8

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 5

[22] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 5

[23] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 5

[24] 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. 8

[25] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016. 5

[26] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70, 2000. 5

[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4, 5

[28] A. Roberts, J. Engel, S. Oore, and D. Eck. Learning latent representations of music to generate interactive musical palettes. 2018. 4

[29] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016. 8

[30] A. Semmo, D. Limberger, J. E. Kyprianidis, and J. Döllner. Image stylization by oil paint filtering using color palettes. In *Proceedings of the workshop on Computational Aesthetics*, pages 149–158. Eurographics Association, 2015. 5

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 7

[32] R. Tanno, S. Matsuo, W. Shimoda, and K. Yanai. Deepstylecam: A real-time style transfer app on ios. In *International Conference on Multimedia Modeling*, pages 446–449. Springer, 2017. 1

[33] 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. 5

[34] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 5, 2018. 5

[35] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016. 5

[36] Z. Yi, H. R. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876, 2017. 5

[37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 5