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

Jinning Li[1] and **Yexiang Xue**[2]
[1]Shanghai Jiao Tong University
[2]Purdue University
lijinning@sjtu.edu.cn, 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 the 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 state-of-the-art models both visually and quantitatively. 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 style transfer, which converts photos into artistic oil paintings [Gatys *et al.*, 2016; Isola *et al.*, 2017; Zhu *et al.*, 2017]. Style transfer is among the first few commercialized deep learning technologies [Tanno *et al.*, 2017]. Users are excited at these applications, which allow them to generate self-portraits and paintings as if they were painted by famous artists. However, these applications *require a photo* to generate an artistic painting. A 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(a) 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 photo. 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 style transfer, where the neural network is to *summarize, reduce and adapt* the information within a real-world photo into an artistic painting. The aforementioned differences are fundamental, precluding existing style transfer approaches such as Pix2pix [Isola *et al.*, 2017] and Scribbler [Sangkloy *et al.*, 2016b] to succeed in our scribble-to-painting application.

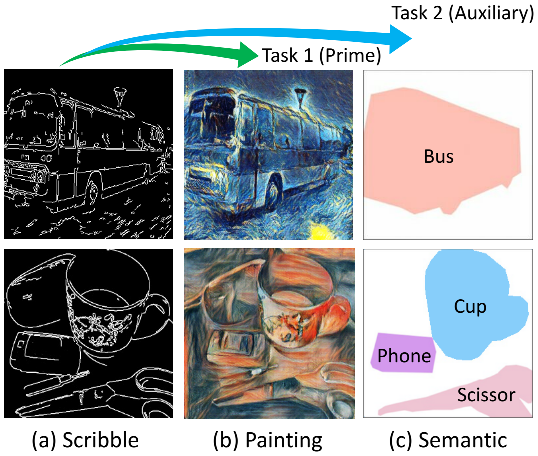


Figure 1: Examples of scribble-to-painting transformation. (a) Scribble images as inputs. (b) Artistic paintings generated by DSP-Net. (c) An auxiliary object detection and segmentation task helps generate better paintings in (b). In the first example, recognizing the object as a bus helps to color and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

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 [Vyshedskiy and Dunn, 2015]. 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.

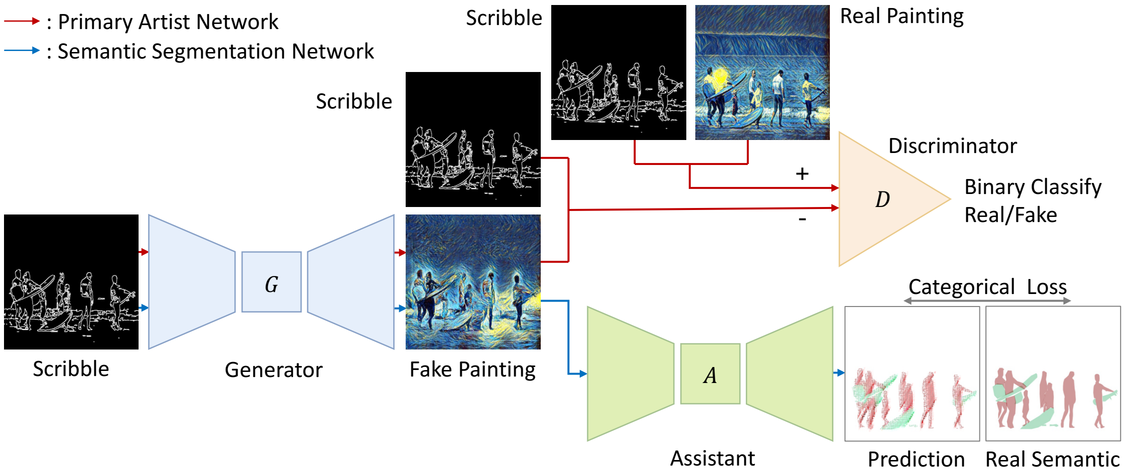


Figure 2: Overview of the multi-task Dual Scribble-to-Painting Network (DSP-Net). It contains a primary artist network and a secondary semantic segmentation network. The primary artist network is a conditional generative adversarial network, which generates paintings in artistic style based on scribble 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).

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 sub-networks. 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 training of the primary network. We demonstrate that joint training on these two tasks brings in additional benefits. Similar ideas can be found in CoupleGAN [Liu and Tuzel, 2016] and S^2-GAN [Wang and Gupta, 2016] but with notable differences. CoupleGAN learns the joint distribution of multi-domain images using two parallel Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014] while S^2-GAN cascades two GANs in a series manner. Our secondary segmentation task is supervised directly from annotated results, which is different from the second GANs as in the previous works.

The 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-trained VGG-19 network. The experiments on Sketchy database [Sangkloy *et al.*, 2016a] prove that our approach generalizes well to real human-drawn scribbles. Moreover, the human evaluation results show that the majority (62.44%) of participants picks our results as the best in terms of details and aesthetics.

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. There are some painting datasets for classification such as the Paintings Dataset [Crowley and Zisserman, 2014] and Painting 91 [Khan *et al.*, 2014]. However, they do not come with the corresponding scribbles and are not labeled semantically as well. The number of available oil-paintings online is also far less than the number of photos. Our insight is to rely on large image datasets which were already semantically labeled and the existing approaches for line extraction and neural style transfer. In particular, we start with the COCO dataset [Lin *et al.*, 2014]. We extract lines of the photos to form surrogates for scribble images. Line extraction images resemble real scribbles well. In fact, we showed in the experimental section that DSP-Net trained using line extraction images can be directly applied to real scribbles with satisfactory results. We then use existing neural style transfer algorithms [Johnson *et al.*, 2016] to transform photos in the COCO dataset to oil paintings. We collect a large dataset of more than 5000 pairs of scribble and painting in this way.

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 $x_i \in \mathcal{X}$ to denote one scribble image as input. An artistic painting as output is denoted as $y_i \in \mathcal{Y}$. 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. The objective mapping h could be found by minimizing 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. Therefore, a neural net has to *infer* missing details from the scribble, such as the colors of surfaces and the geometric shapes of objects, etc, in scribble-to-painting transformation. It is a process of *adding information*.

On the contrary, many state-of-art style transfer models work from photos, which are rich in terms of information content. The classical style transfer application is a *information reduction* task, where the information-rich photos are converted into information-sparse paintings. 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 information in the scribble images poses a significant challenge in scribble-to-painting transformation. To solve this problem, we get inspired by 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 image $A(G(x_i))$, 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 [Girshick, 2015] used in our paper. Combining these two neural networks, the objective of DSP-Net can be formulated as:

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

Note that both the scribble x and painting y are fed to the discriminator D. This design allows the discriminator to detect abnormal "content switching" behavior of the generator. Suppose the generator G transforms 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 is interleaved until convergence.

3.1 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 conditional GAN introduced in [Isola *et al.*, 2017; Radford *et al.*, 2015] as our primary network.

The generator and discriminator of the primary network are shown in Figure 1. The generator $G : \mathcal{X} \to \mathcal{Y}$ is a mapping from scribbles to paintings. The discriminator $D : \mathcal{Y} \to \{0, 1\}$ 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 [Isola *et al.*, 2017] as the generator, which first encodes the feature of a scribble image 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 [Li and Wand, 2016] as our architecture. The min-max objective for both the generator G and the discriminator D in the primary artist network is:

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

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

3.2 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 [Girshick, 2015] 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. 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.

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 a 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 multi-task 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 [Lin *et al.*, 2014]. COCO dataset provides the pairs of the real photo p_i and its corresponding semantic image z_i. We build scribble x_i shown in Figure 3a from p_i with a simple but effective Canny edge extracting algorithm [Canny, 1986]. The painting y_i shown in Figure 3b is generated with a fast neural style network [Johnson *et al.*, 2016] pre-trained on COCO dataset. y_i is obtained by feeding real photos p_i into the pre-trained network. Since the photos p_i contain full information, the quality of synthesized paintings is good enough for them to serve as the training ground truths. Along with the semantic image z_i which is manually labeled for photo p_i by COCO dataset, the triples (x_i, y_i, z_i) are induced.

5 Experiments

In this section, we are going to evaluate the experimental performances of DSP-Net and several existing models widely used in image transformation, including Neural Style [Gatys *et al.*, 2016], Fast Neural Style [Johnson *et al.*, 2016], Pix2pix [Isola *et al.*, 2017], and CycleGAN [Zhu *et al.*, 2017]. We train the GAN based models for 200 epochs and neural style for 1000 iterations. We keep the other hyper-parameters and basic settings as the recommended value mentioned in their original code.

We use the triples of scribbles, paintings, and semantic images obtained using the methods introduced in Section 4. 5000 images from COCO dataset [Lin *et al.*, 2014] are randomly sampled to build the dataset. We split the dataset into 4500 images for training and 500 images for a test. The oil paintings *The Starry Night* by Vincent van Gogh and *The Scream* by Edvard Munch are used as the style images. The code and dataset could be found at https://github.com/jinningli/DSP-Net.

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

A DSP-Net without the secondary task is equivalent to a conditional GAN (Pix2pix model), the result of which is shown in Figure 3c. The Pix2pix model can generate reasonable colors and contents. However, it fails to generate a vivid style and brushwork of oil paintings. Comparing to Pix2pix, DSP-Net in Figure 3g generates more realistic patterns of oil painting and semantic contents. This is because the secondary semantic segmentation task forces the generator to generate varieties of paintings which satisfies the constraints of semantic information.

The sparsity of scribble images causes serious blurring and blank space (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* and *The Scream* in the areas where enough information is provided, it failed to generate the patterns in sparse areas. Similar problems also happen to Fast Neural Style model. This model can only infer the color and patterns to apply within a limited distance from the drawn scribble lines. So the uncertain area is left blank.

The performance of the popular neural style algorithm in Figure 3e is also quite limited in the scribble-to-painting transformation task. It is hard to figure out the objects in

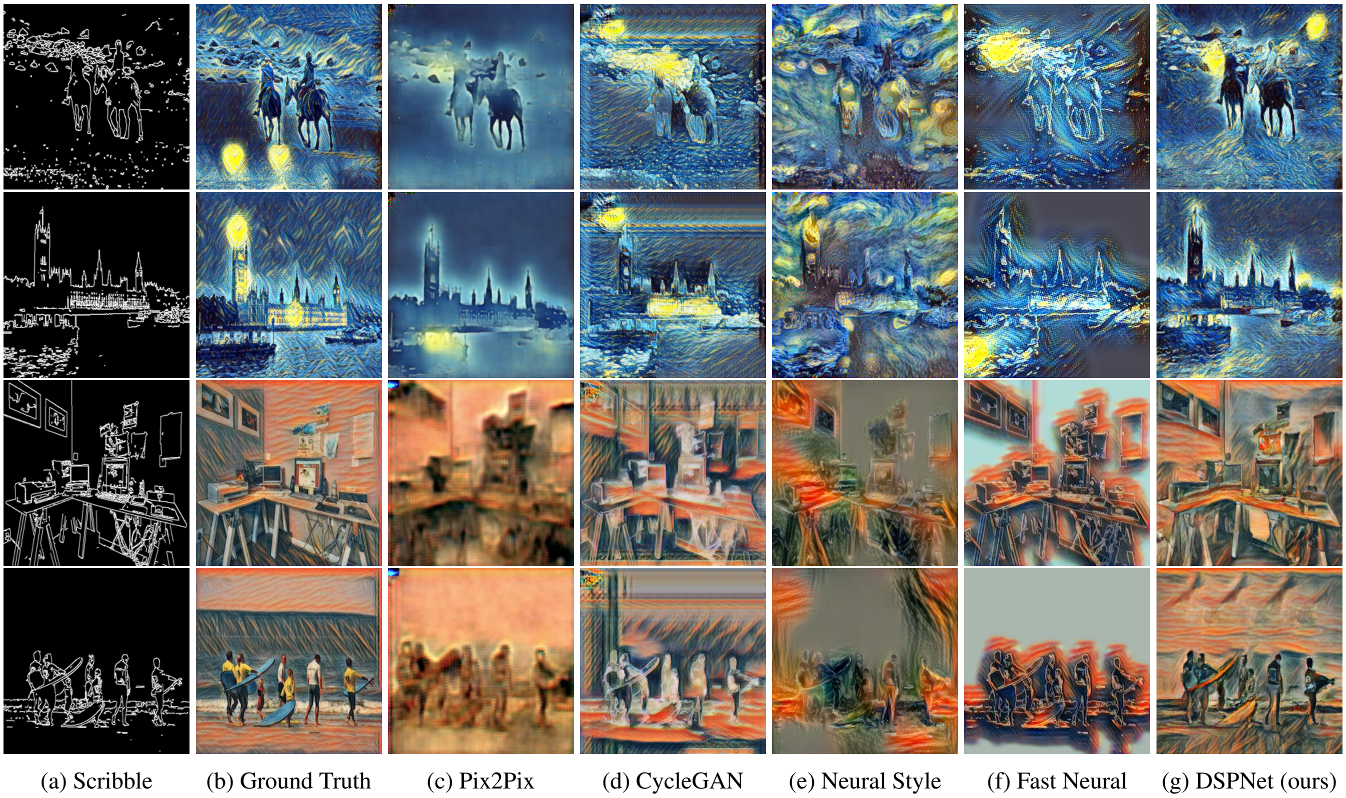


Figure 3: The experimental results of DSP-Net and comparative models. The first two lines are based on the style of *The Starry Night* and the following two lines are trained on *The Scream*. The first column shows the scribble images used as inputs. The second column shows the ground truth built with 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.

the synthesized images. They look like a recombination of the patterns of *The Starry Night* or *The Scream*. The 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.

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

5.2 Quantitative Analysis

We also compare our model quantitatively with some state-of-the-art style transfer models. Experimental result proves that our model performs better under the metrics of content mismatching and style mismatching (See Table 1). This means our results are closer to the objective painting images with respect to both content and style similarity.

Inspired by the idea mentioned in the neural style algorithm [Gatys *et al.*, 2016] 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 real ones. We use a pre-trained very deep convolutional network [Simonyan and Zisserman, 2014] (VGG network) to extract the encoded features. 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, $\delta_c(x, y, h) = \frac{1}{2}\sum_l \sum_{i,j} \left[M^l_{i,j}(y) - M^l_{i,j}(h(x))\right]^2$.

In order to calculate the style mismatching loss, we use Gram matrix [Gatys *et al.*, 2015] 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, $E^l_{i,j}(q) = \sum_k M_{i,k}(q)^l M^l_{j,k}(q)$. After calculating all the gram matrices of y and $h(x)$ on different layer l, the style mismatching δ_s can be described as $\delta_s(x, y, h) = \sum_l \frac{1}{4W_l^2 H_l^2} \sum_{i,j} \left[E^l_{i,j}(y) - E^l_{i,j}(h(x))\right]^2$, where W_l and H_l are the width and height of the feature map in the layer l of VGG-19 network.

Model	Content Loss	Style Loss
Pix2pix	0.185	0.447
CycleGAN	0.194	0.074
Neural Style	0.200	0.105
Fast Neural Style	0.246	0.304
DSP-Net(Ours)	**0.175**	**0.070**

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.

Model	the most realistic & aesthetic?
Pix2pix	1.49%
CycleGAN	4.23%
Neural Style	27.11%
Fast Neural Style	4.73%
DSP-Net (Ours)	**62.44%**

Table 2: Human evaluation in terms of content realism and style aesthetics. We report the percentage of votes that each model is chosen as the most realistic and aesthetic.

With two metrics mentioned above, we calculate the average of content loss δ_c and style loss δ_s on the test set, 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 $\bar{\delta}_s(h) = \frac{1}{m}\sum_m \delta_s(x_m, y_m, h)$.

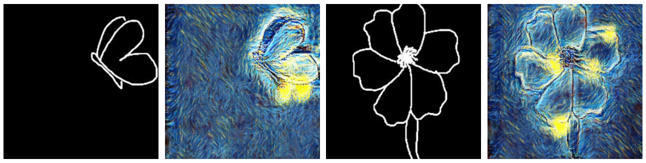
Assume h_k is the hypothesis of the k-th model. We apply normalization to both the content and style mismatching losses by $\Delta_c(h_k) = \bar{\delta}_c(h_k)/\sum_k \bar{\delta}_c(h_k)$ and $\Delta_s(h_k) = \bar{\delta}_s(h_k)/\sum_k \bar{\delta}_s(h_k)$. The results of normalized mismatching losses 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. The content mismatching of Fast Neural Style is the largest, which is 0.246, since Fast Neural Style model fails to generate textures in the area where no scribble is provided.

In the aspect of style mismatching, DSP-Net outperforms other models as well. The style 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. Pix2pix gets the highest mismatching loss because in Figure 3c, almost the whole image remains blur and almost no texture is synthesized.

5.3 Generated Paintings with 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 [Lu *et al.*, 2012; Lim *et al.*, 2013]. Recently, SketchyGAN [Chen and Hays, 2018] has tried to augment the edge extractions with human-drawn sketches in the Sketchy Dataset [Sangkloy *et al.*, 2016a] to improve the model generalization. However, the quality of its synthesized images from sparse sketches is still not satisfactory enough.



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

Figure 4: Syntheses of DSP-Net with real scribbles. It is trained on our dataset and tested on the Sketchy database which consists of real human-drawn scribbles, as is shown Figure 4a and 4c.

Even though our approach is trained on images of edge extraction, it generalizes well to human-drawn scribbles. We train DSP-Net model on our dataset derived from COCO introduced in Section 4, 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 style patterns are realistic.

5.4 Human Evaluation: Realism and Aesthetics

We complete a survey among students from different majors. We show every participant 6 questions. In each question, we show the participant one scribble image (randomly drawn from the testset) and the corresponding six oil paintings synthesized by five baseline models and our proposed model. The order of synthesized paintings is shuffled so that the participant do not know which painting corresponds to which model. In each question, we let participants vote for the most realistic and artistic painting. Then we count the percentage of votes each model gets. We in total receive 402 votes from 67 participants. Results in Table 2 show that the paintings synthesized by DSP-Net are chosen to be the most realistic and aesthetic in 62.44% out of all questions.

6 Conclusion

In this paper, we focus on a novel task that transforms scribbles into artistic paintings. This task is more challenging than classical image style transfer, because the scribbles contain far less information compared with photos. We use multitask 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. Experiments on human-drawn scribbles and human participated surveys also demonstrate the effectiveness of DSP-Net. As an additional contribution, we publish a large dataset for scribble-to-painting transformation. Possible future directions include extending the application of the multi-task generative adversarial networks into other important tasks in computer vision and natural language processing.

References

[Canny, 1986] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986.

[Chen and Hays, 2018] Wengling Chen and James 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.

[Crowley and Zisserman, 2014] Elliot J Crowley and Andrew Zisserman. In search of art. In *Workshop at the European Conference on Computer Vision*, pages 54–70. Springer, 2014.

[Gatys *et al.*, 2015] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

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

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[Khan *et al.*, 2014] Fahad Shahbaz Khan, Shida Beigpour, Joost Van de Weijer, and Michael Felsberg. Painting-91: a large scale database for computational painting categorization. *Machine vision and applications*, 25(6):1385–1397, 2014.

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

[Lim *et al.*, 2013] Joseph J Lim, C Lawrence Zitnick, and Piotr 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.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Liu and Tuzel, 2016] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[Lu *et al.*, 2012] Cewu Lu, Li Xu, and Jiaya 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.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Sangkloy *et al.*, 2016a] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016.

[Sangkloy *et al.*, 2016b] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. *arXiv preprint arXiv:1612.00835*, 2016.

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

[Tanno *et al.*, 2017] Ryosuke Tanno, Shin Matsuo, Wataru Shimoda, and Keiji Yanai. Deepstylecam: A real-time style transfer app on ios. In *International Conference on Multimedia Modeling*, pages 446–449. Springer, 2017.

[Vyshedskiy and Dunn, 2015] Andrey Vyshedskiy and Rita Dunn. Mental synthesis involves the synchronization of independent neuronal ensembles. *Research Ideas and Outcomes*, 1:e7642, 2015.

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

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.