Artistic Image Painting Using Artificial Intelligence

Arfy Slowy

slowy.arfy@gmail.com

https://www.github.com/slowy07/neuralPainting

Abstract

This paper proposes an image to painting translation method that generates vivid and realistic painting artworks with controllable styles. Different from previous image-to-image translation methods that formulate the translation as pixel-wise prediction. We deal with such an artistic creation process in a vectorized environment and produce a sequence of physical meaningful stroke parameters that can be further used for rendering. Since a typical vector render is not differentiable, we design a neural renderer which imitates the behavior of the vector renderer and then frame the stroke prediction as a parameter searching process that maximizes the similarity between the input and the rendering output. We explored the zero gradient problem on parameter searching and propose to solve this problem on parameter searching and propose to solve this problem from an optimal transportation perspective. We also show that previous neural renderers have a parameter coupling problem and we re-design the rendering network with a rasterization network and shading network that better handles the disentanglement of shape and color. Experiments show that the paintings generated by our method have a high degree of fidelity in both global appearance and local textures. Our method can be also jointly optimized with neural style transfer that can further transfer visual style from other images.

1. introduction

Creating artistic paintings is one of the defining characteristics of humans and other intelligent species. In recent years, we saw great advancements in generative modeling of image translation or style transfer which utilizes neural networks as a generative tool. Previous image-to-image translation and style transfer methods typically formulate the translation either as a pixel-wise mapping or a continuous optimization process in their pixel space. However, as an artistic creation process the paintings usually proceed as a sequentially instantiated process that creates using brushes, from abstract to concrete, and from macro to detail. This process is fundamentally different from how neural networks create artwork that produces pixel-by-pixel results. To fully master professional painting skills, people usually need a lot of practice and learn domain expertise. Even for a skilled painter with years of practice, it could still stake hours of days to create a realistic painting artwork.

In this paper, we explore the secret nature of human painting and propose an automatic image-to-painting translation method that generates vivid and realistic paintings with controllable styles. We refer to our method as “Neural Painting”. Instead of manipulating each of the pixels in the output image, we simulate human painting behavior and generate vectorized strokes sequentially with a clear physical significance. Those generated stroke vectors can be further used for rendering with arbitrary output resolution. Our method can “draw” in a variety of painting styles, e.g oil-painting brush, watercolor ink. marker-pen, and tape art. Besides, our method can also be naturally embedded in a neural style transfer framework and can be jointly optimized to transfer its visual style based on different style reference images.

In our method, different from the previous stroke-based rendering methods that utilize step-wise greed search, recurrent neural network, or reinforcement learning, we reformulate the stroke prediction as “parameter searching” process that aims to maximize the similarity between the input and the rendering output in a self-supervised manner. Considering that a typical graphic render is not differentiable, we take advantage of the neural rendering that imitates the behavior of the graphic rendering and make all components in our method differentiable. We show that previous neural stroke rendered may suffer from the parameter coupling problem when facing complex rendering scenarios, e.g brushes with real-world textures and color-transition.We, therefore, re-design the neural renderer and decomposed the rendering architecture into a rasterization network and a shading network, which can be jointly trained and rendered with much better shape and color fidelity. We also found interestingly that the pixel wise similarity like $\ell 1\ or\ \ell 2$pixel loss, may have an intrinsic flaw of zero-gradient on optimizing over the vectorized parameters, although these losses have been widely used in a variety of image translation tasks. We show that this problem lies in the different nature of stroke parameterization and rasterization and propose to solve this problem from the perspective of optimal transportation. Specifically, we consider the movement of a stroke from one location to another as a transportation process, where we aim to minimize the efforts of that movement.

We test our method on various real-world images and photos, including human portraits, animals, scenery, daily objects, art photography, and cartoon images. We show that our method can generate vivid painting with a high degree of realism and artistic sense in terms of both global visual appearance and local texture fidelity.

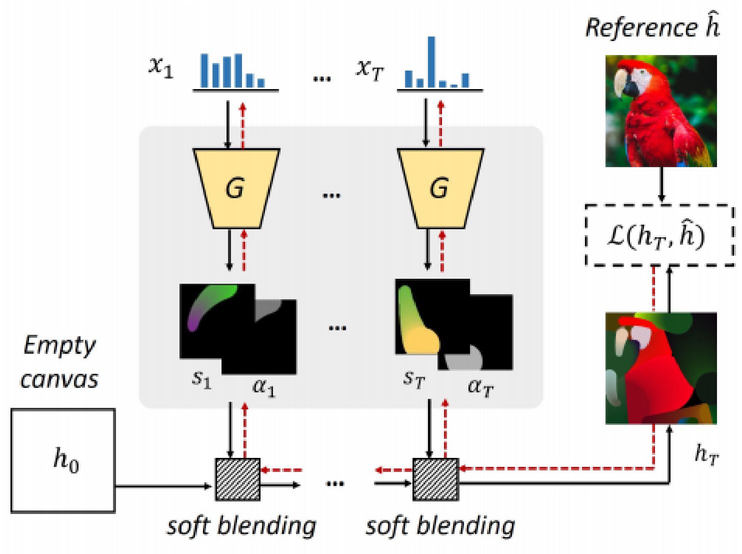
the contribution of our paper is summarized as follows:

- We propose a new method for stroke base image-to-painting translation. We re-frame the stroke prediction as a parameter searching process. Our method can be jointly optimized with neural style transfer in the same framework
- We explore the zero-gradient problem on parameter searching and view the stroke optimization from an optimal transport perspective. We introduce a differentiable transportation loss and improve stroke convergence as well as the painting results.
- We design a new neural renderer architecture with a dual-pathway rendering pipeline (rasterization + shading). The proposed renderer better deals with the disentanglement of the shape and color and outperforms previous neural renderers with a large margin.

2. Related Work

Image translation and style transfer. Image translation, which aims at translating images from one domain (e.g., real photos) to another (e.g., artworks), has drawn great attention in recent years. GAN based image translation as Pix2Pix, CycleGans and their variants has played and important role in tasks like image synthesis, sematic editing, style transfer, and also has been applied to computer-generated arts. In addition to the GAN based method, neural style transfer has also made breakthroughs in stylized image synthesis and is widely used for artwork creation. Besides, the Deep Dream, a method that was initially designed to help visualize deep neural networks, also has become a new form of psychedelic and abstract art. Despite the above applications, these methods all generate paintings in a pixel-by-pixel manner, which deviates from the fact that humans use brushes to paint.

Differentiable rendering. Rendering is a fundamental problem in computer graphics that converts 3D models into 2D images. Traditional rendering pipelines typically involve discrete operation called rasterization, which makes the rendering non-differentiable. Differentiable rendering breaks such limitations and allows calculation of the derivative from the rendering output to the input parameters such as shape, camera, pose and lighting. Since deep neural networks are naturally differentiable.



PIC 1 start from an empty canvas and then render stroke-by-stroke with soft blending. we use gradient descent to find a set of "optimal" stroke parameters that minimize loss $\mathcal{L}$. Here black arrow lines and red ones mean back-propagation of the gradient

In their topology, a new research topic called "neural rendering" quickly emerged, which bridges the gap between graphic rendering and deep neural networks.

image sketching/painting. Humans are born with the ability to abstract and sketch object instances. Early methods on image sketching deal with this problem by using stepwise greedy search or require user interaction Recent approaches typically train recurrent neural networks and Reinforcement Learning (RL), or integrate adversarial training to generate non deterministic stroke sequences. More recently, thanks to the recent advances of neural rendering, computers are now able to generate more realistic painting artworks. Among these methods, "Learning to paint" has a similar research motivation to ours, both dedicated to generating stroke based realistic paintings, however our method differs from this method in several aspects. First, this method generates strokes by using RL while we formulate this process as stroke parameters research since training RL agents is computationally expensive. Second, we focus on stroke-base style transfer, which is a rare studied problem.Finally, also we redesign the neural renderer and introduce optimal transport methods to this problem.

3. Methodology

Our method consist of three technical modules : 1) a neural renderer that is trained to generate strokes given a set vectorized stroke parameters; 2) a stroke blender that combines multiple rendering strokes in differentiable menner; 3) a similarity measurement module that enforces the reconstruction of the input image. In the following, we introduce how each module works accordingly.

- **Overview**

PIC 1 shows an overview of our method. Given and empty canvas h_0 , We draw step-by-step and superimpose those strokes a rendered at each step iteratively. In each drawing step t, a trained neural renderer $\mathcal{G}$ takes in a set of stroke parameters xt (e.g., shape, color, transparency, and texture), and produces a stroke foreground and alpha matte. We then use soft blending to mix the canvas, the foreground, and alpha mate each step and make sure the entire rendering pipeline is differentiable. The soft blending is defined as follows:

$$h_t + 1 = \alpha_t s + (1 - \alpha_t) h_t$$

where ($\propto_t \boldsymbol{s}_t$) = $G(\boldsymbol{x}_t)$. We finally gather the stroke parameters from all the T steps and optimize them by searching within the stroke parameter space. The searching is conducted under a self-supervised manner i.e., We enforce the final rendered output similar to reference image

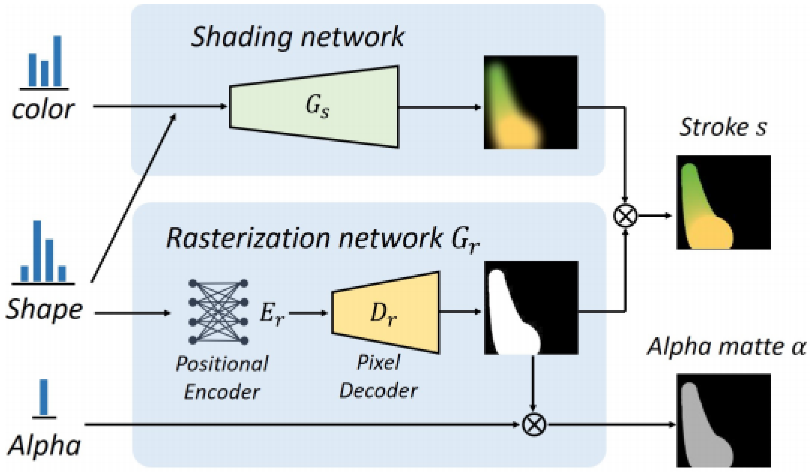
$$h_t = f_{t=\ 1\ \sim\ T}(\hat{x}) \approx \hat{h}$$

where $h_t = f_{t=\ 1\ \sim\ T}()$ is a recursive mapping from stroke parameters to the rendered canvas $\overline{x}$= [x_t........x_T] are the collection of stroke parameters at t= 1,2….T drawing steps.

Suppose L represent a loss function that measure the similarity between the canvas h_t, we optimize all the input strokes at their parameter space and minimize the facial similarity loss. We use gradient decent to update the strokes as follows

$$\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{x}} - \mu \frac{\partial \mathcal{L}(h_T, \hat{h})}{\partial \tilde{\boldsymbol{x}}},$$

- **Disentangle neural rendering**
  To build a neural renderer, a general practice is to build a deep convolutional network and train it to imitate behavior of a graphic engine. Previous researches proposed to use stacked transposed convolutions or position encoder + decoder architecture. These approaches can work well in simple stroke rendering scenarios. However, we find these renderers may suffer from a coupling of shape and color representations when testing with more complex rendering settings like color transition and some textures. We propose to solve this problem by designing a dual pathway neural renderer that disentangles color and shape/texture through the rendering pipeline.



PIC 2 We design a dual-pathway neural renderer which consists of a shading network and a rasterization network. Our renderer takes in a group stroke parameters (color, shape, and transparency) and produces the rasterized foreground map and alpha matte

As shown in *PIC 2,* the proposed neural renderer consists of two networks, a shading network and a rasterization network. We divide the parameters of a stroke into three groups, color,shape and transparency. We build as a stack of several transposed convolution layers, which takes in both the color and shape parameters and generates stroke with faithful foreground color. We design the *Gr* as a positional encoder + a pixel decoder, which simply ignores the color but generates stroke silhouette with a clear shape boundary. We finally generate the output stroke foreground s by masking the color map with the stroke silhouette and generate the final alpha matte by rescaling the silhouette using the input alpha value.

We train our neural renderer with standard pixel regression losses on both the rendered stroke foreground and the alpha matte. During the training, we minimize the following objective function:

$$\mathcal{L}_G(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}\sim u(\boldsymbol{x})}\{\|s-\hat{s}\|_2^2 + \|\alpha-\hat{\alpha}\|_2^2\},$$

where $\bar{s}$and $\hat{\alpha}$ are the ground truth foreground foreground and alpha matte rendered by graphic engine. $x \sim u(x)$ are stroke parameters randomly sampled within their parameter space.

- **Pixel similarity and zero-gradient problem**

There are many ways to define the similarity between the rendered output and the reference, and perhaps the most straight-forward one is to define as pixel-wise loss, e.g ℓ_1 *or* $\ell 2$ losses. Note that when we manipulate the image by directly optimizing from their pixel space, using the pixel-wise loss can work pretty well. However, when it comes to optimizing stroke parameters, we show that pixel loss does not always guarantee effective gradient descent particularly, when the rendered stroke and its ground truth do not share overlapped regions, there will be a zero optimizing the pixel-loss and the transportation loss. We see that pixel ℓ_1 loss, the stroke fails to move along the right direction since there is no gradient on its parameterized locations while using transportation loss make the stroke nicely converges to the target

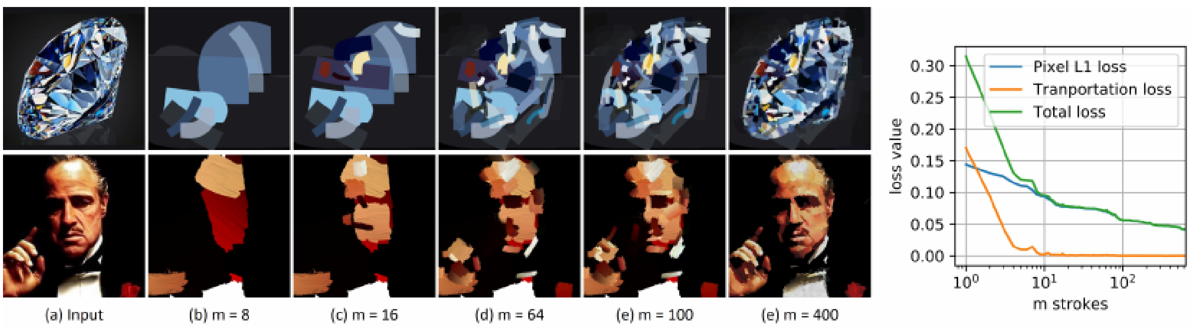
- **Optimal transport for stroke searching**

  We define the minimum transportation efforts, i.e., the wasserstein distance, as an effective measure of similarity loss between the canvas and the reference image.in this paper we employ a smoothed version of the classic optimal transport distance with an entropic regularization term, which yields the celebrated sinkhorn distance is differentiable and enjoys benign mathematical properties that can result in much lower computational cost than the original one. The Primary idea is to consider extra entropic constraints on the joint probability matrix. Further using the lagrange multiplier.

- **Implementation details**

  Network architecture. We build our shading network similar to DCGAN's generator, which consist of six transposed conv layers. We remove the Tanh activation from the output layer and observe a better convergence. In our rasterization network, we first build a positional encoder with four fully connected layers then build a pixel decoder with six conv layers and the pixel-shuffle layers. We also experiment with the architecture of "UNet", wherein this case, we tile the stroke parameters on their spatial dimension to a 3D tensor as the input of the network.

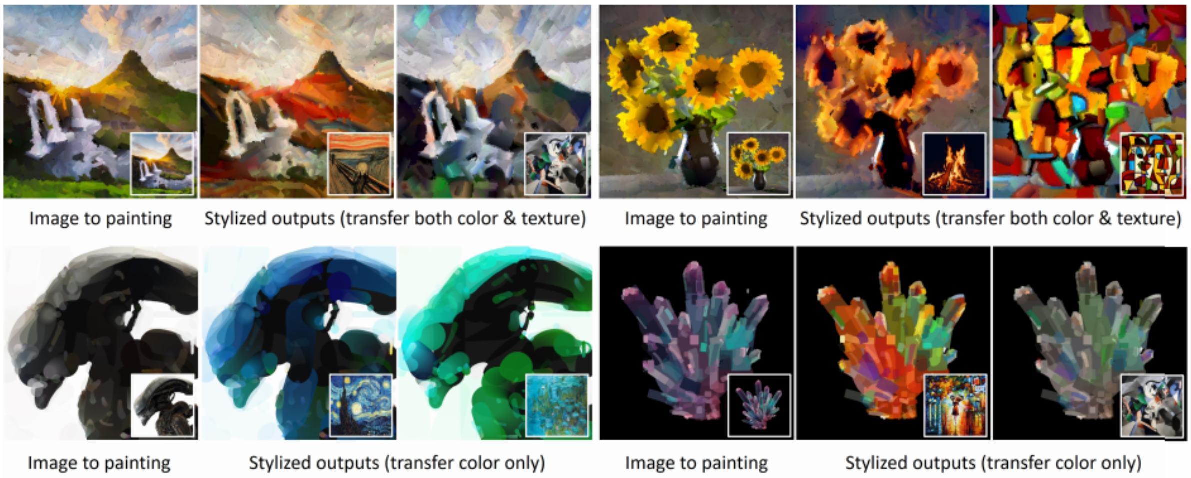
  Training details. We train our renderer by using Adam optimizer. we set batch size to 64, learning rate to 2e-4, and beats to (0.9, 0.999). we reduce the learning rate to its 1/10 every 100 epochs and stop training after 400 epochs. In



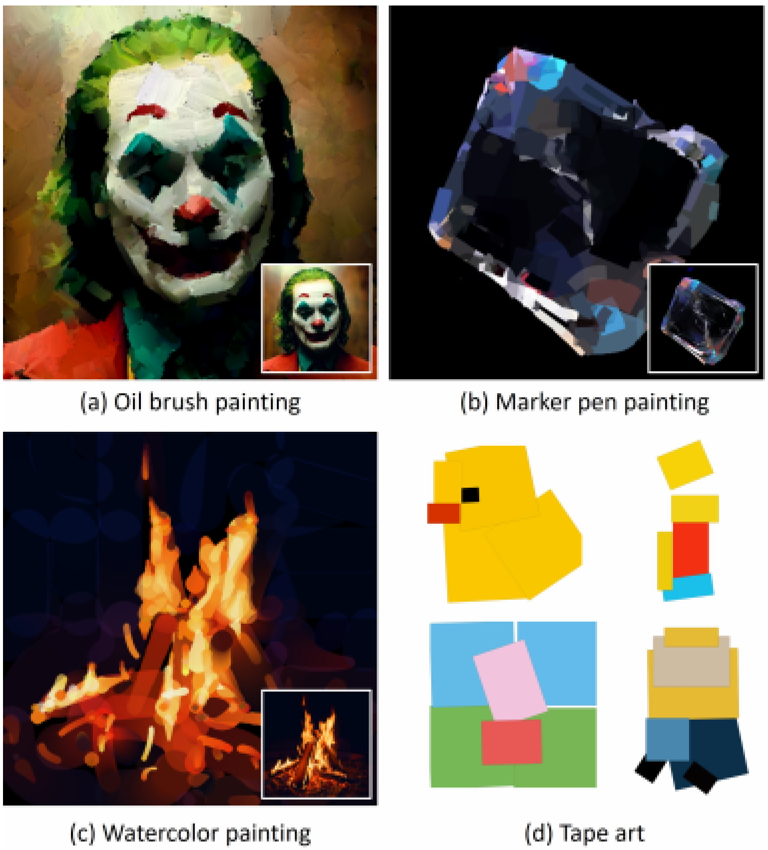
PIC 3 Stroke-by-stroke painting of our method with marker-pen (1st row) and oil-paint bush (2nd row). On the right, we also plot the loss curves as the painting preceded.

each epoch we randomly generated 50,000 x 64 ground truth strokes using a vector engine. we set the rendering output size to 128 x 128 pixels. We traint renderers separately for each stroke type.

Progressive rendering. To render with more details , we design a preogressive rendering pipeline in both the scale and action dimension. We first start from searching parameters on a single 128 x 128 canvas and then divide the canvas into m x m blocks (m = *2,3,4....*) with overlaps and search on each of them accordingly. In each block scale, we gradually add new strokes to an "active set" and update the strokes progressively. In each update, we optimize all strokes within the active seat at the same time. We run gradient descent for 20 x N steps for each block, where N is the number of strokes in each block, same for all blocks regardless of their scales.



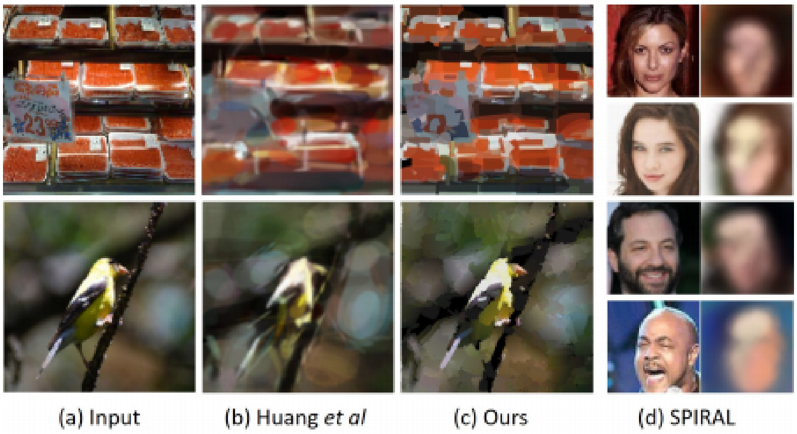
PIC 4 Style transfer result by using our method. The 1st row shows the results that transfer the style of both color and texture. The 2nd row shows the results that transfer color only.



PIC 5 (a) -(c) Stylized paintings generated by our method. in (d), we also snow some highly abstract tape arts of cartoon characters by our method.

4. Experimental Anlysis

Painting generation. PIC 5 shows as group of stylized paintings generated by using our method with different stroke types. in (d), we also show several tape-artworks of well-known characters that are automatically created by our method with a minimum number of strokes. We can see that our method successfully learns high -level abstractions of the characters and vividly portrays their shape and color. **PIC 3** shows the stroke-by-stroke painting result by using different stroke brushes. On the right of this figure, we also plot the changes in the loss values as the painting proceeded. We can see that in the very first few drawings steps, our method nicely captures the global appearance of the object, and the drawing then gradually goes from macro to detail. **PIC 4** shows more examples of our painting results as well as their style transfer result. We can see by integrating the style loss in (7), both color and textures can be successfully, transferred to the paintings with their content remaining unchanged.



PIC 6 A comparison of the paintings created by our method (400 strokes), "learning-to-Paint" (400 strokes), and Spiral (20 strokes). The result in (b) and (d) are from their papers

Comparison with other methods. in **PIC 6**, we compare our method with two recently proposed stroke-based image-to-painting translation methods: 1) "Learning-to-Paint", and 2) "SPIRAL", where both of them train RL agent to paint. We can see our method generates more vivid results with a clear distinction on brush textures, while other methods tend to produce blurred results. We also compare the stylized artworks created by our method with those created manually

conclusion

We explore the nature of human painting using differentiable stroke rendering. We consider this artistic creation process under a stroke parameter searching paradigm that maximizes the similarity between the sequentially rendered canvas and the reference image. Our method can generate highly realistic and painting artwork in vector format with controllable styles. We dealt with the image similarity measurement from the perspective of optimal transportation and tackle the disentanglement of the color and shape with a dual-pathway neural renderer. Controlled experiments suggest th effectiveness of our design.

REFERENCES

[1] Henry Abbott. Creating art using generative adversarial networks. 2019.

[2] Xuelin Chen, Daniel Cohen-Or, Baoquan Chen, and Niloy J Mitra. Neural graphics pipeline for controllable image generation. arXiv preprint arXiv:2006.10569, 2020.

[3] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in neural information processing systems, pages 2292–2300, 2013.

[4] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. Science, 360(6394):1204–1210, 2018.

[5] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, SM Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. arXiv preprint arXiv:1804.01118, 2018.

[6] 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.

[7] David Ha and Douglas Eck. A neural representation of sketch drawings. arXiv preprint arXiv:1704.03477, 2017.

[8] Paul Haeberli. Paint by numbers: Abstract image representations. In Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pages 207– 214, 1990.

[9] A Hertzmann. A survey of stroke-based rendering. IEEE Computer Graphics and Applications, 23(4):70–81, 2003.

[10] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14(8), 2012.

[11] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 8709–8718, 2019.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125-1134, 2017