PAINT TRANSFORMER

Manogna Sreenivas

SR No. 18659

ABSTRACT

Neural image painting is an exciting problem in the area of AI for creativity. Given a target image, the objective here is to generate a painting of the same. Paint transformer [1] employs a transformer module to predict a set of stroke parameters which is then used by a stroke renderer to paint. As a part of this project, I implemented the Paint Transformer[1] and compared it with Stylized Neural Painting[2]. I also propose a way to seamlessly incorporate new stroke renderers into the current pipeline.

1. INTRODUCTION

Stroke based neural image painting is a creative process of imitating humans by painting a series of strokes onto a canvas. This task requires a stroke predictor to obtain stroke parameters and a stroke renderer to draw strokes onto canvas. Previous approaches for this task are spread across different paradigms of AI. [3] design an RL agent to generate strokes. [2] formulate this as an optimization problem to search for an optimal set of strokes. Paint Transformer design an impressive self training pipeline using a transformer as stroke predictor and no external dataset.

2. TECHNICAL DETAILS

2.1. Paint Transformer

Paint Transformers formulate this as set prediction problem. Given an intermediate canvas I_c and target image I_t, a transformer model is learnt to predict a set of stroke parameters to minimize the difference between them. For training, a set of random parameters (s_c, s_t) s.t $s_c \in s_t$ are sampled to generate an intermediate canvas I_c and target I_t. The set of ground truth stroke parameters is then given by $s_{gt} = s_t \setminus s_c$.

$$s_{pred} = StrokePredictor(I_c, I_t) \tag{1}$$

$$S_{pred}, \alpha_{pred} = StrokeRenderer(s_{pred}) \tag{2}$$

$$I_c^{'} = \alpha_{pred} S_{pred} + (1 - \alpha_{pred})\, I_c \tag{3}$$

$$\mathcal{L} = \mathcal{L}_{\text{stroke}}\ (s_{pred}, s_{gt}) + \mathcal{L}_{\text{pixel}}\ (I_c, I_t) \tag{4}$$

The transformer encoder embeds I_c and I_t which is then fed to the decoder along with N learnable stroke query vectors. The decoded queries are further fed into FC layer to predict stroke parameters and confidence $(\boldsymbol{s_{param}}, \boldsymbol{c}) = (s_i, c_i), i = 1, .., N$. Selected params are then used to render stroke and alpha maps which is then used to update the canvas as described in (2) and (3). The Transformer is then optimized to minimize the objective $\mathcal{L}$ defined in (4).

2.2. Stroke Renderer

Paint transformers use a simple differentiable affine transformation to render rectangular brush strokes parameterized by $(x, y, w, h, \theta, r, g, b)$. Alternatively, neural networks can be learnt to render more complex parameterized strokes. A trained differentiable neural renderer can be integrated with the Stroke predictor to propagate gradients from the pixel wise loss during training. However, this can be discarded during inference as basic image functions from libraries like opencv suffice to render the strokes from predicted params. Inspired by [3][2], I propose to integrate a bezier curve renderer with the Stroke Predictor. A quadratic bezier curve is defined as $B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2$ where $P_i = (x_i, y_i)\,, i = 1, 2, 3$ define the control points.

3. RESULTS

Inference is done recursively over K scales in a coarse to fine fashion. Painting on current scale acts as canvas for the next scale. In each scale, the canvas is divided into PxP patches, patch size dependent on the scale.
Paint transformer being a simple feed forward model, is very efficient for inference. Stylized neural painting optimizes the search for stroke parameters which is expensive during inference. For an image of size 1024x1024, the total time for stroke prediction as described above takes about 150s using [2] while it takes only about 55ms using Paint Transformer.

4. CONTRIBUTIONS

Using the model definition from the inference code provided by the authors, I implemented the training pipeline. I compared the complexity and qualitative results with the method [2]. I propose a way to integrate new stroke renderers in an efficient way into the current pipeline. Though the idea was validated as a part of this project, more extensive fine-tuning experiments are to be performed.

5. RESOURCES

1. https://github.com/wzmsltw/PaintTransformer
2. https://github.com/jiupinjia/stylized-neural-painting

6. REFERENCES

[1] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang, "Paint transformer: Feed forward neural painting with stroke prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[2] Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi, "Stylized neural painting," 2020.

[3] 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 (ICCV)*, 2019.