Paint Transformers

Manogna Sreenivas

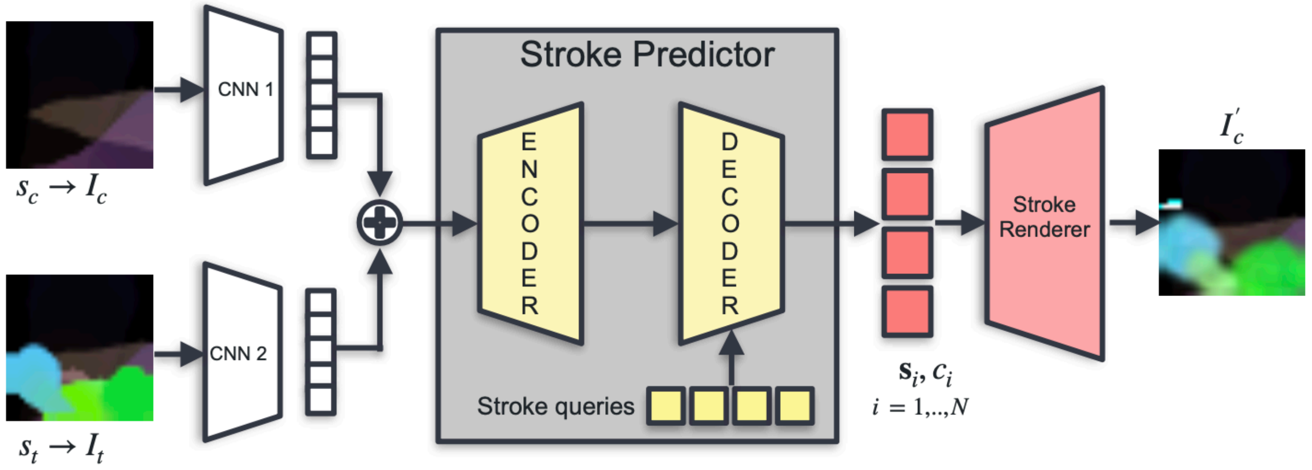
Introduction

- Neural Image painting is the task of generating a painting given a reference image.
- GANs model this a translation task from image to painting at a pixel level.
- Inspired from way humans paint, recent methods [1][2][3] model this task as a creative stroke rendering process onto a canvas.
- Given an intermediate canvas and target image, the task requires:
  - Stroke predictor: Strokes represented by parametrisable geometric objects
  - Stroke rendering: Draw stroke onto canvas, given stroke parameters
- Previous works for stroke prediction come from different paradigms of AI: Using RNNs, RL agent[3], Optimisation based stroke searching[2].
- Renderers need to be differentiable to allow back propagation of pixel level errors.
- Paint Transformers propose an efficient way to predict stroke parameters using a Transformer encoder decoder module. They also propose impressive self training pipeline to synthesise canvas and target images by generating random strokes on the fly.

Proposed Approach

- **Paint Transformers[1]**
  - Stroke Predictor: Transformer encoder decoder module.
  - Stroke renderer: Affine transformation to draw rectangular brush strokes parameterised by $(x, y, w, h, \theta, r, g, b)$.
- **Stylised Neural Painting[2]**
  - Models strokes of oil paint brush, watercolour brush, marker pen.
  - Stroke renderer: A neural renderer (an FCN like DCGAN generator architecture)
  - Stroke prediction process is cumbersome as they use and optimisation objective to search for parameters in the stroke parameter space during inference.
- Inspired by the above, I propose to integrate neural stroke renderers with Paint transformer leveraging the efficient stroke predictor [1] and complex stroke renderer from [2]. This eliminates the overhead during inference in [2],[3].
- In specific, I model strokes as bezier curves (watercolour brush in [2]) in this project.
- Bezier Stroke Renderer with Paint Transformer:
  - Bezier stroke renderer is learnt as shown in Fig (b).
  - The weights of stroke renderer is frozen when integrated with Paint Transformer. However, the pixel loss is propagated through the Neural renderer to supervise the learning of stroke predictor.
  - Once the stroke predictor is trained, the neural renderer can be discarded during inference.
  - Given predicted stroke parameters, simple image manipulation techniques suffice to draw strokes on the canvas.

Paint Transformer to render Bezier curves



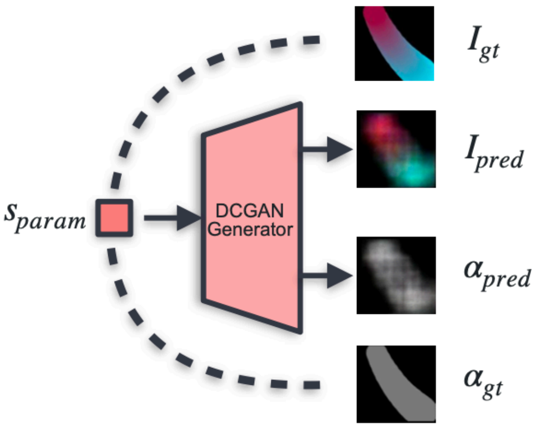
$$s_{gt} = s_t \backslash s_c$$

$$\mathcal{L}_{\text{pixel}} = \| I_c' - I_t \|_1$$

$$\mathcal{L}_{\text{stroke}} = \| \mathbf{s}_{\mathbf{pred}} - \mathbf{s}_{\mathbf{gt}} \|_1 + BCE(c_{pred}, y_{gt})$$

$$min \mathcal{L} = \mathcal{L}_{\text{stroke}} + \mathcal{L}_{\text{pixel}}$$

(a) Self training a Paint Transformer to predict N strokes given intermediate canvas and target image



$$min \mathcal{L} = \| I_{gt} - I_{pred} \|_2 + \| \alpha_{gt} - \alpha_{pred} \|_2$$

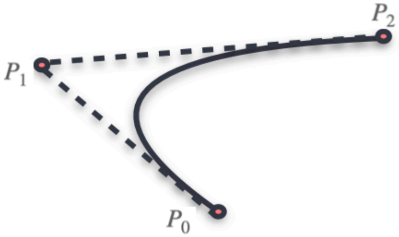
(b) Learning Neural Stroke renderer

Contributions (Novelty)

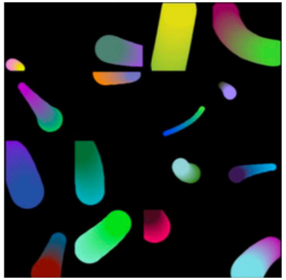
- Bezier Curves for stroke rendering:
  - A quadratic bezier curve is represented by

    $$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2$$

    where $P_i = (x_i, y_i), i = 1,2,3$ represent the control points.
  - Strokes like watercolour brush can be synthesised adding thickness (r_0, r_2) and colour$(R_0, G_0, B_0, R_2, G_2, B_2)$ parameters at P_0 and P_2 respectively which is linearly interpolated to obtain the colour and thickness along the bezier curve as shown in (b)
- I proposed a way to seamlessly integrate neural stroke renderers with Paint Transformer to learn to render complex strokes.
- Using the model definitions from the inference code provided by the authors [1], I implemented the self training pipeline and reproduced the results.
- Implemented Bezier stroke renderer and integrated with Paint Transformer.
- For comparison with [2], code provided by the authors was used.



(a) Quadratic Bezier Curve

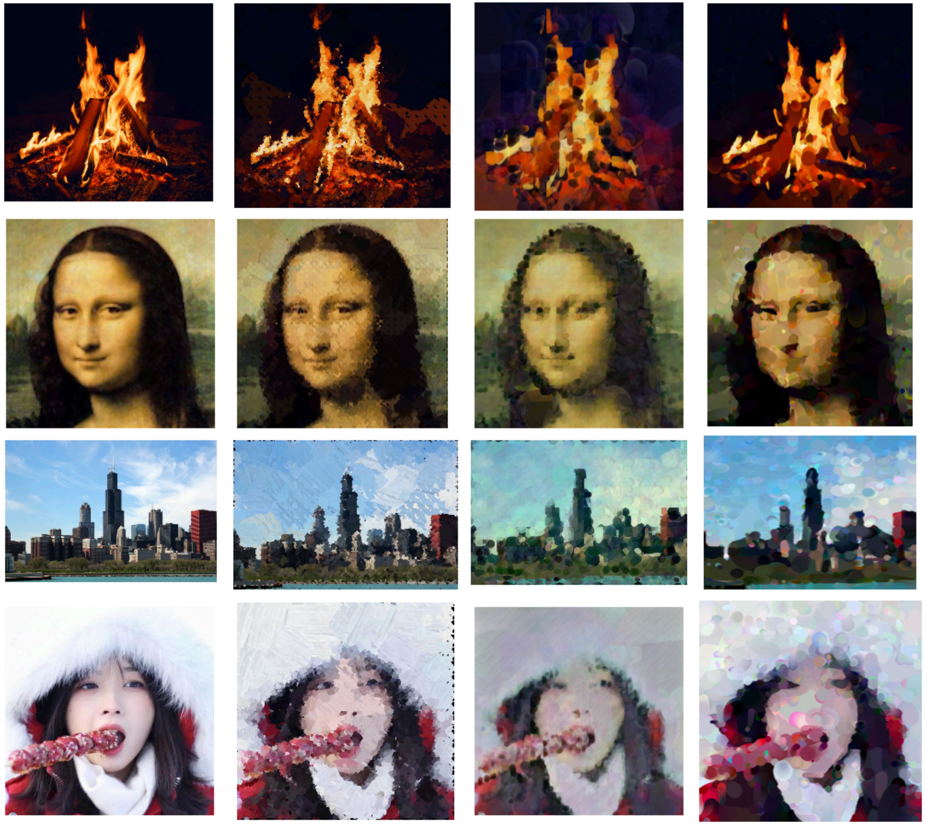


(b) Bezier curve strokes

[1] Songhua Liu et al. “Paint transformer: Feed forward neural painting with stroke prediction”, ICCV 2021
[2] Zhengxia Zou et al. “Stylized neural painting”, CVPR 2020
[3] Zhewei Huang et al. “Learning to paint with model-based deep reinforcement learning”, ICCV 2019

Results & Conclusion

- Paint Transformer trained for oil paint brush strokes(a) render vivid strokes with best quality paintings.
- Results from (d) trained using the same brush strokes as in (c) introduce artefacts and are not smooth always. It is also very expensive during inference.
- For an image of size 1024x1024, the total time for stroke prediction takes about 150s using Stylised Neural Painting, while it takes only about 55ms using Paint Transformer on an NVIDIA Geforce RTX2080 Ti GPU.
- The proposed Paint transformer to render bezier curves(c) is able to render different shapes as can be seen clearly in the fire image. It performs better than (d) though trained to render the same bezier strokes.
- These being results from initial experiments, more extensive fine tuning would produce better results. The idea was validated as a part of this project.
- The proposed method can be easily extended to a variety of strokes. A selection strategy to select from different strokes can also be incorporated.



(a) Target image

(b) Paint transformer with oilpaint brush[1]

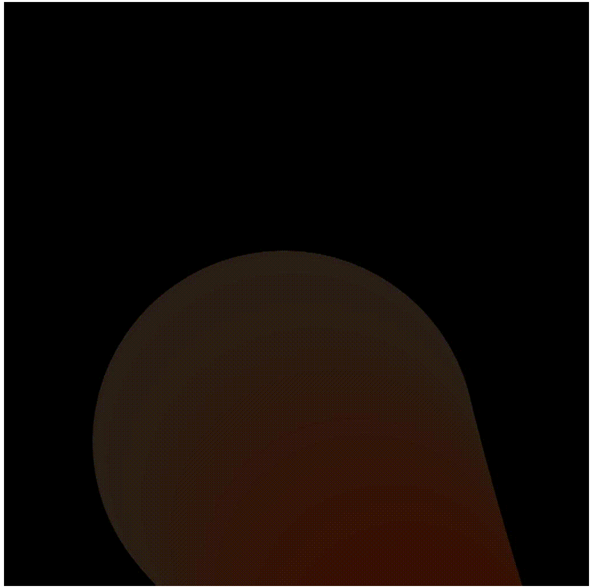
(c) Paint transformer with bezier strokes

(d) Stylised Neural painting

More results at: https://github.com/manogna-s/NeuralPainting

Painting Demo

(a) Paint transformer with rectangular brush strokes



(b) Paint transformer with Bezier curve strokes