Arbitrary Style Transfer

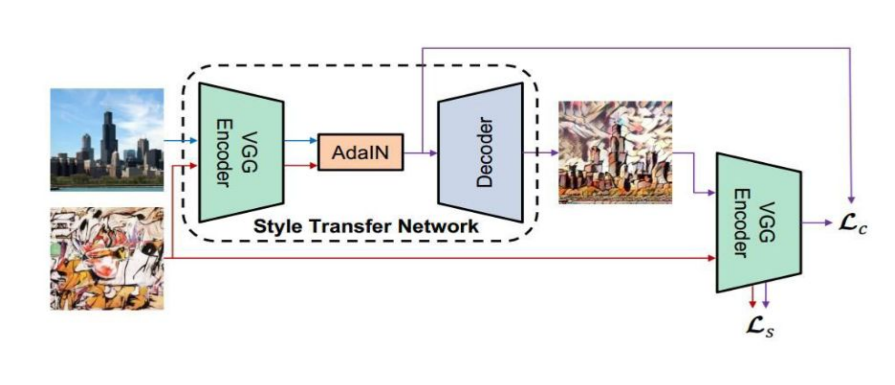
By:

Abinash Maharana
Aditya Gupta
Mohsin Hafiz
Kartik Agrawal

Contents

- Overview
- Components
- Progress
- Remaining work
- Final deliverables
- Results

Overview



In our project, we give 2 images, a content image and another style image which defines what style we want to give to our content image.

Our paper requires us to encode the content and style images using vgg19 network.

Then we put a middle AdaIN (Adaptive Instance Normalisation) layer. In this layer we try to change the statistics of content class like mean and variance to that of style images.

Then we use decoder whose network is opposite of encoder. Now we train our model on COCO dataset and then performed experiments.

Components

Main components of our architecture:

1. Encoder
2. Adaptive Instance Normalization train layer
3. Decoder and other utilities.

Our architecture is then trained on COCO dataset. After completion, our model would be run to get output to our content and style images.

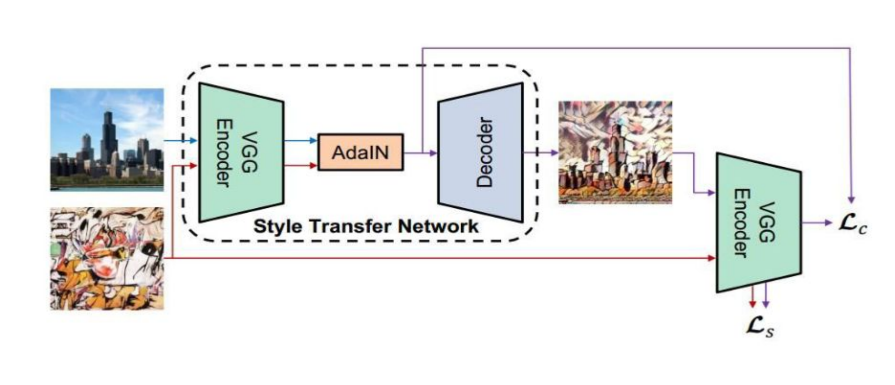
Progress

- Build encoder to decoder pipeline without the middle layer.
- Model training utilities like loss function,etc.
- Trained pipeline on COCO dataset.
- Performed experiments on our images and results are attached in this presentation.

Remaining work

- Writing AdaIN middle layer and integrating with our pipeline.
- Training our model again along with AdaIN layer on COCO dataset.
- Now experiment our complete architecture on our images.
- Perform follow-up experiments like style interpolation, spatial and color controls, content-style tradeoff, etc.

Final Deliverables



- Trained End-to-End model
  - Style Transfer Network

- Results of our architecture experimented on our content and style images.

- Results and observations from conducting experiments.
  - Content-style tradeoff
  - Style Interpolation
  - Spatial and color control

Results: