

Image Style Transfer

...

Neural Style

Overview

Combine style and content from different images to generate a new image.

Implementation Overview

CNNs trained and existing models

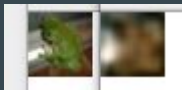
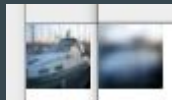
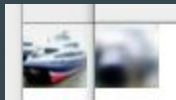
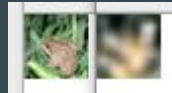
We have trained a small 32x32 autoencoder and a classifier based on the CIFAR10 dataset.

We would be trying style transfer using this network later on to test results based on various networks.

Framework used - Torch (Lua JIT)

Training time: Approx 24hrs on GPU (30 iterations on all images)

Trained Autoencoder (CNN) on 32x32 images



Content Loss modules implemented

A single content loss module has been added in the network and we have tested it with a few images. The 3 most commonly used images for style transfer were used and the results were recorded. The tubingen image, brad pitt's face and the golden gate image.

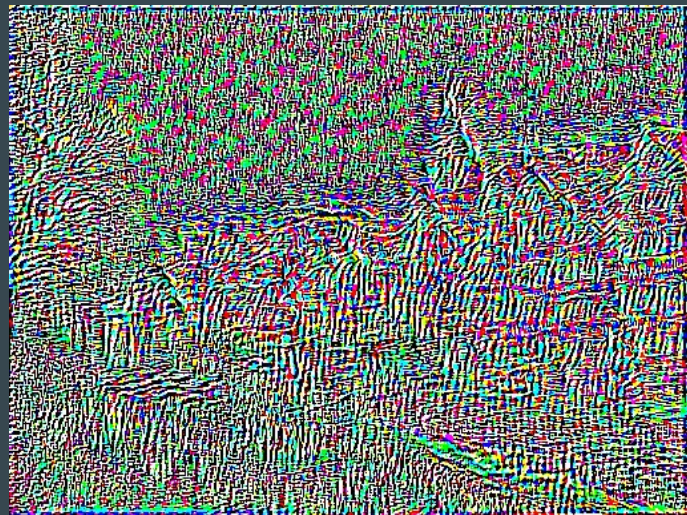
This module just stores an expected content image representation initially when we pass it through the network and then when we pass the white noise image it compares the L2 norm between the noise image representation at the current layer and the expected output of the content image at that same layer and adds this to the gradient while backpropagating.

White noise to Image (Algorithm)

- Using white noise as input(x), propagate through the network to get the output (y).
- Set $dy = 0$, and start back propagation.
- Initially the gradients would be zeros, but as soon as we reach our loss modules that we inserted in the network, the gradients start accumulating. These are now passed all the way till the input as dx .
- We update the input as $x' = x - dx * rate1$ (Gradient descent on the input image)
- Perform this multiple times (10s-100s) and then we would end up minimizing the loss at the input image and our image x would start to have the content of our content image and the style of the style image we used as inputs initially.
- (Currently this loss only comprises of content losses)

Example Outputs - 1

Tubingen.jpg





Example Outputs - 2

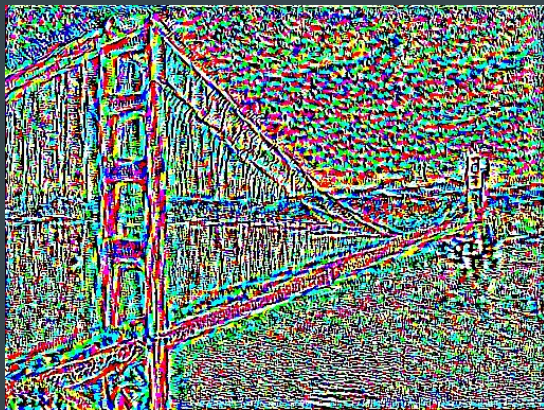
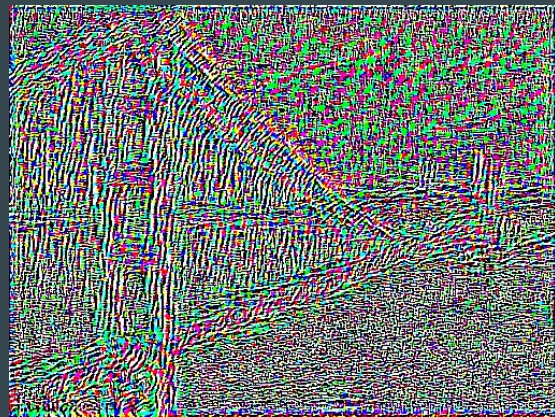
brad_pitt.jpg





Example Outputs - 3

golden_gate.jpg





Past/Current Research

- Style Transfer has been extended to videos as well. Some papers directly apply this on videos frame by frame whereas others calculate differences between frames and apply it explicitly so it's a lot faster and doesn't feel too noisy.
- A few methods similar to style transfer are being compared. http://cs231n.stanford.edu/reports2016/208_Report.pdf
- Style transfer on face portraits was done in 2014 itself, https://people.csail.mit.edu/yichangshih/portrait_web/ and the results were quite impressive. This was extended to videos as well.
- Generate almost painting like images (Crayon, wax, paint, pastels etc) - <http://jaisrael.github.io/ParallelStyleTransfer/>
- We have also been planning on extending this to the automatic coloring of sketches. Consider a sketch as an input, get a corresponding image either via an SBIR system or just using standard class images for each of the available sketch classes. We then segment both the sketch and the image and match corresponding regions. We then finally apply the style transfer on each region independently to ensure that we end up coloring the sketch in a manner similar to its corresponding image.

Work Status

Currently we have implemented the content loss layers and added them to the network.

We still need to add the style loss modules and add them in various parts of the network.

Once that is done, we would be using different networks (preferable a custom trained CNN with a 128x128 input) and then check on how it performs.
