Neural Style Transfer

Abhisar Kushwaha, Rohan Teja V, Vikas Bansal
Team 10

November 25, 2018

1 Introduction and Overview

Neural Style Transfer is basically transferring of the style from one image to the another image called content image.We have done it in two stages which are explained below.

Stage 1 :

In this we implemented the naive method which transfers the style without using neural network but with the help of image clustering, feature extraction and blending of images

Stage 2 :

We style our content image with the help of neural network based on minimizing loss functions for content image and style image.

Related work

- This is a technique outlined in Leon A. Gatys paper (link in references), A Neural Algorithm of Artistic Style.
- Some work done by Raymond Yuan (link in references) in which he iterated 1000 times but that's not so optimized.

2 Methods

For Stage-1 we have used the image clustering, feature extraction and blending of images.In this naive method we implemented the image clustering on the style image, edge extraction on the content image and blended those two images with the appropriate alpha value of images to get the styled image.Since this was not dependent on any neural network it was fast.

For Stage-2 we implemented a method which transfers the style using neural network.In this case, we load VGG19 which is pre-trained on the ImageNet dataset with Python and the Keras deep learning library as it is a pretty simple model and the feature maps work better for style transfer, and feed in our input tensor to the model. This will allow us to extract the feature maps of the content, style, and generated images.

In order to access the intermediate layers corresponding to our style and content feature maps, we get the corresponding outputs by using the Keras to define our model with the desired output activations.

Content Loss:

We pass the network both the content image and our base input image. The model returns intermediate layer outputs. Then we simply take the euclidean distance between the two intermediate representations(iterations) of those images. Backpropagation is then done to reduce content loss..

$$L^l_{content}(p,x) = \sum_{i,j} (F_i,^l_j\,(x) - P_i,^l_j\,(p))^2$$

Style Loss:

This works on the same principle but, instead of comparing the raw intermediate outputs, we compare the distance between the style representation using the gram matrices of these images(the correlation between different responses given by the matrices G^l, where $G^l_{i,j}$ is the inner product between the vectorized feature map). We then minimise the mean squared distance between the feature correlation map of the style image and the input image.

Contribution to style loss by each layer:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_i,^l_j - A_i,^l_j)^2$$

total style loss:

$$L_{style}(a,x) = \sum_l^L w_l E_l$$

We then iteratively update our output image such that it minimizes our loss: we dot update the weights associated with our network, but instead we train our input image to minimize loss. A function is used to check the loss and gradients...It records the operations during the forward pass and then is able to compute the gradient of our loss function with respect to our input image for the backwards pass. Finally, we check successive iterations for the best choice of image considering losses.We have used an optimized algorithm "Limited Memory BFGS" to minimize gradient descent which is based on the family of Quasi-Newton method.

3 Experimental Analyses

Datasets

We are not using any dataset but we are using a pre-trained model VGG19 which is trained on the imagenet dataset.

Results

After minimizing value of loss function to some iterations or some threshold value we got our styled image.

4 Discussion and Future Directions

This method can we used to find the new aspects and ideas in the field of paintings and art. Now, everyone can add any type of style to their image.

In future we can further take this project as :

- We can take the user's option about how much or to what extent to style the image
- We can add this method to give live style transfer to any video; and every frame.

- We can also work on optimizing the code and reduce the number of iterations by combining the above two methods i.e Stage-1 and Stage-2 and get a clear styled image with less number of iterations and computation.

References

[1] Leon A. Gatys paper *A Neural Algorithm of Artistic Style.*

[2] https://medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398

[3] https://medium.com/artists-and-machine-intelligence/neural-artistic-style-transfer-a-comprehensive-look-f54d8649c199

[4] https://medium.com/mlreview/making-ai-art-with-style-transfer-using-keras-8bb5fa44b216

[5] https://github.com/anishathalye/neural-style