Neural Style Transfer

Aditya Singh Bisht

What is Neural Style Transfer?

Neural style transfer is an optimization technique used to take two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but “painted” in the style of the style reference image.

Content Image

Style Image

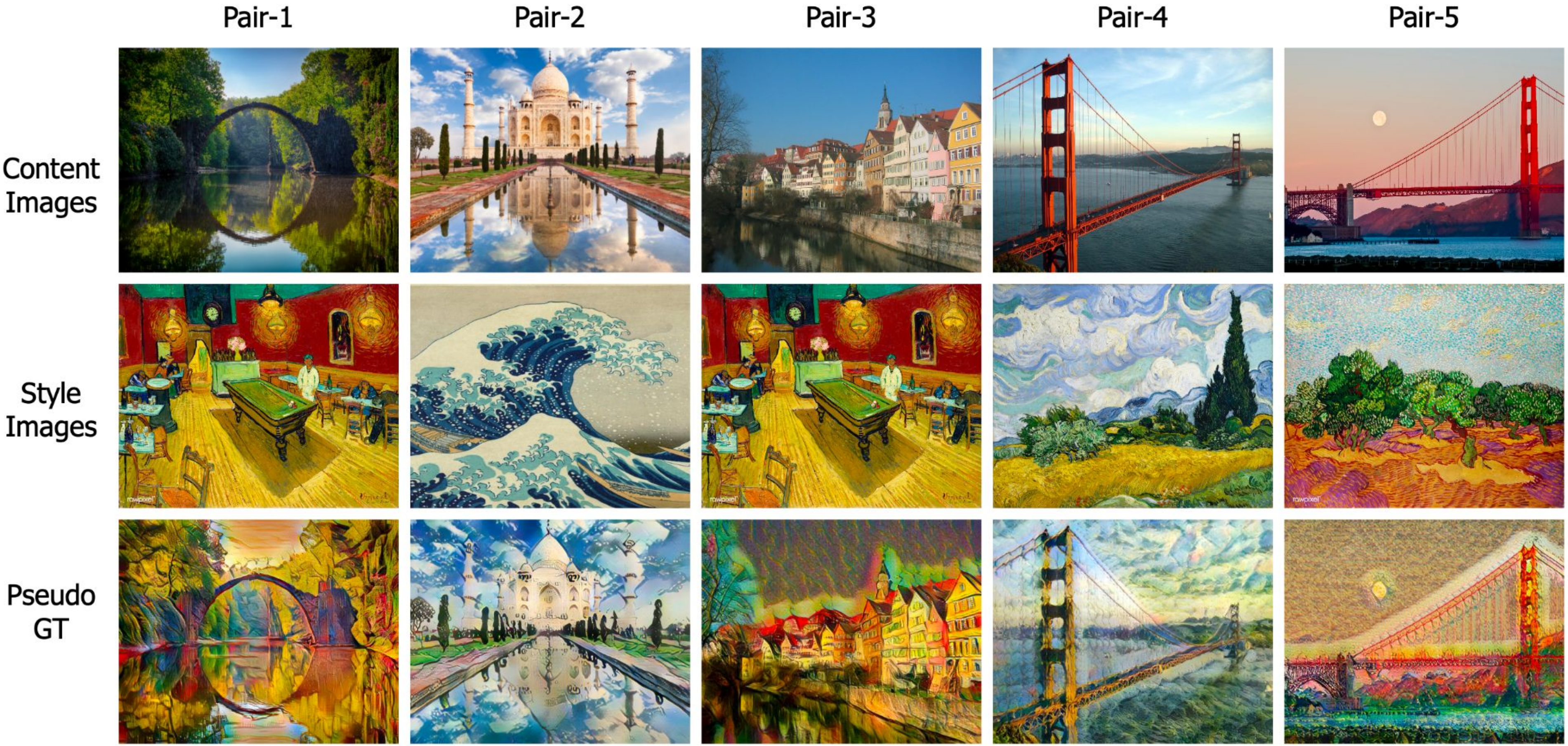
Neural Style Transfer

Methodology

I first extract the content and style (gram matrices) features from any given image by using a deep neural network (like VGG16 or VGG19 or ResNet-18).

The shallow layers in feature extractor store pixel colour information, and style information and the deeper layers store content (structure) related information.

We learn the input image with supervision from a weighted sum of content loss: between the features from content and input image, and style loss: between the features from style and input image


Pair-1
Pair-2
Pair-3
Pair-4
Pair-5
Content Images
Style Images
Pseudo GT

Implementation

- PyTorch.
- Torchvision model zoo for our feature extractors.
- In case of VGG-19, we use relu4_2 layer for content loss and [relu1_1, relu2_1, relu3_1, relu4_1, relu4_2, relu5_1] for style loss.
- When using ResNet-18, we use features from layer1 for content loss and the gram matrices from [conv1, layer1, layer2, layer3] for style loss.

Optimizers

I experimented with different optimizers among LBFGS, Adam and AdamW using the ResNet- 18 extractor.

In order to find the best learning rate (lr) in case of Adam and AdamW, we tune the lr hyper-parameter on Pair-2 . We find lr = 1e−1 and lr = 1e−2 to be the best setting for Adam and AdamW, respectively.

Evaluation for tuning the learning rate was also done using the PSNR, SSIM and RMSE metrics.

We find LBFGS to show the best performance and hence, use LBFGS for further experiments

Metric	LBFGS	Adam	AdamW
RMSE $\downarrow$	**0.01187**	0.01270	0.01238
PSNR $\uparrow$	**38.528**	38.001	38.235
SSIM $\uparrow$	**0.8864**	0.8785	0.8801

Number of Iterations

I also experiment with the number of iterations for the ResNet-18 feature extractor and LBFGS optimizer.

I selected three numbers 1000, 1500, 2000 and used the same methodology in order to compare and evaluate the three models using different numbers of iterations.

I found that the 1000 iterations gives the best performance. Thus, we set the number of iterations to 1000 for further experiments.

Metric	**1000**	**1500**	**2000**
RMSE ↓	**0.01187**	0.01191	0.01191
PSNR ↑	**38.528**	38.504	38.501
SSIM ↑	**0.8864**	0.8859	0.8860

Conclusion

I presented an extensive experimental analysis considering various Neural Style Transfer task factors.

I also proposed to use pseudo ground truth to evaluate and compare different experimental settings.

Additionally, we conduct ablations on various critical performance factors like optimizers and loss terms.