Udnie, Image Source: Wikipedia

NYU TANDON

FAST NEURAL STYLE TRANSFER

Submitted by:

Avinash Veershetty (apv280)

Rajat Ravindrakumar Bapuri (rrb398)

Shiva Sanketh Ramagiri Mathad (srm714)

EL-GY 6123: Introduction to Machine Learning (Graduate) course project

Problem Statement

We aim at using Neural Style Transfer to produce artistic transformation of images. A task which a human would take days or months to complete is done within seconds using neural networks.

Content

+

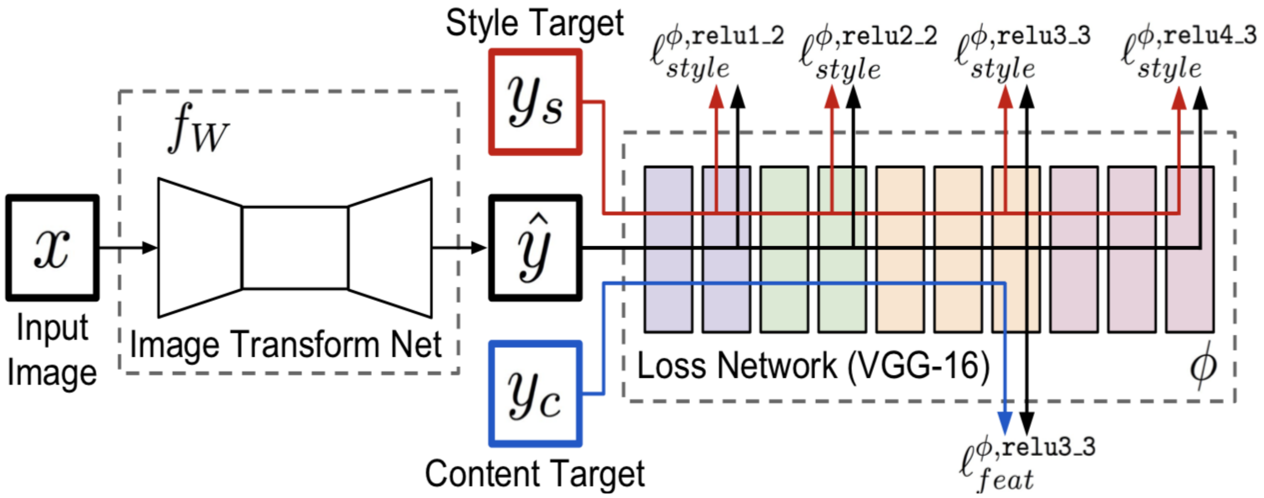
Style

=

Transformed image

Architecture

- Following is the architecture used in the Fast Neural Style Transfer.
- Image Transform net is a 16 layer network which has residual connections.
- A Deep Convolutional Neural network such as VGG-16, VGG-19, Inception network or Resnet pre-trained on Imagenet dataset is used as loss network.



Perceptual Losses for Real-Time Style Transfer and Super-Resolution by Johnson et al

Methodology

NYU TANDON

- The *image transformation network* f_w is a deep residual convolutional neural network parameterized by weights, i.e. it transforms input images x into output images $\hat{y}$ via mapping $\hat{y} = f_w(x)$.

$$W^* = argmin_w E_{x,\{y_i\}}[\sum_{i=1} \lambda_i l_i(f_w(x), y_i)]$$

- Loss network is used to define a *feature reconstruction loss* and *style reconstruction loss* that measure differences in content and style between images.

Dataset and training

- We use MS-COCO dataset to train the network.
- The loss net is loaded with weights pretrained on Imagenet dataset and are frozen and untrainable. The Image Transform Net are trainable.
- We use the following hyperparameters to train the network:
  - Adam optimizer with learning Rate : 1e−3
  - Total variation regularization whose strength can be set between 1×10^{-6} and 1×10^{-4}
  - 2 Epochs on the whole dataset.
- We train different models by changing the loss network to different CNNs such as VGG 16, VGG 19, Inception Net, ResNet etc.
- More detailed information is provided in the report.

Improvements

- The base code we have opted incorporates VGG 16 architecture, we included the VGG 19 network.
- Base code implementation derives its inference on images. Our implementation extends it to provide fast style transfer on videos.

Results - Images

- It is observed that VGG 19 outperforms when compared to all other architectures.
- For video results, please check the readme of the GitHub repo.

Style Image

Resulting image using VGG 19

Content Image

Resulting image using VGG 16

NYU
TANDON

Results - Texture + Style images on the content image

- It is also possible to add texture to the content image along with style image.

Content Image

Style Image

Texture image

Resulting image without texture using VGG 16

Resulting image with texture using VGG 16

Results - Texture + Style images on the content image

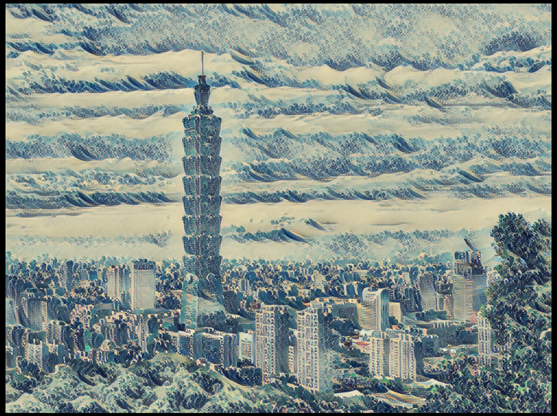
- It is also possible to add texture to the content image along with style image.

Content Image

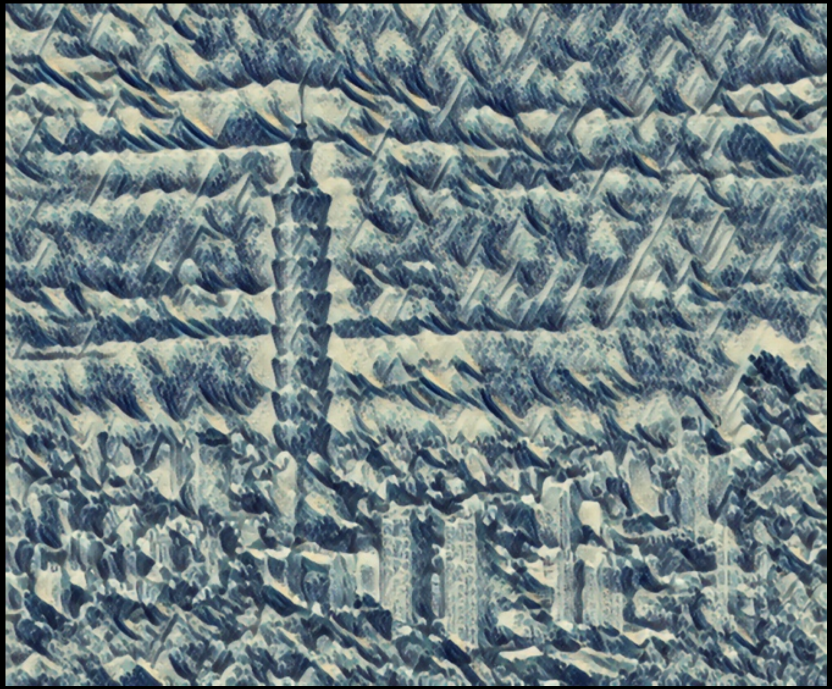
Style Image

Texture image

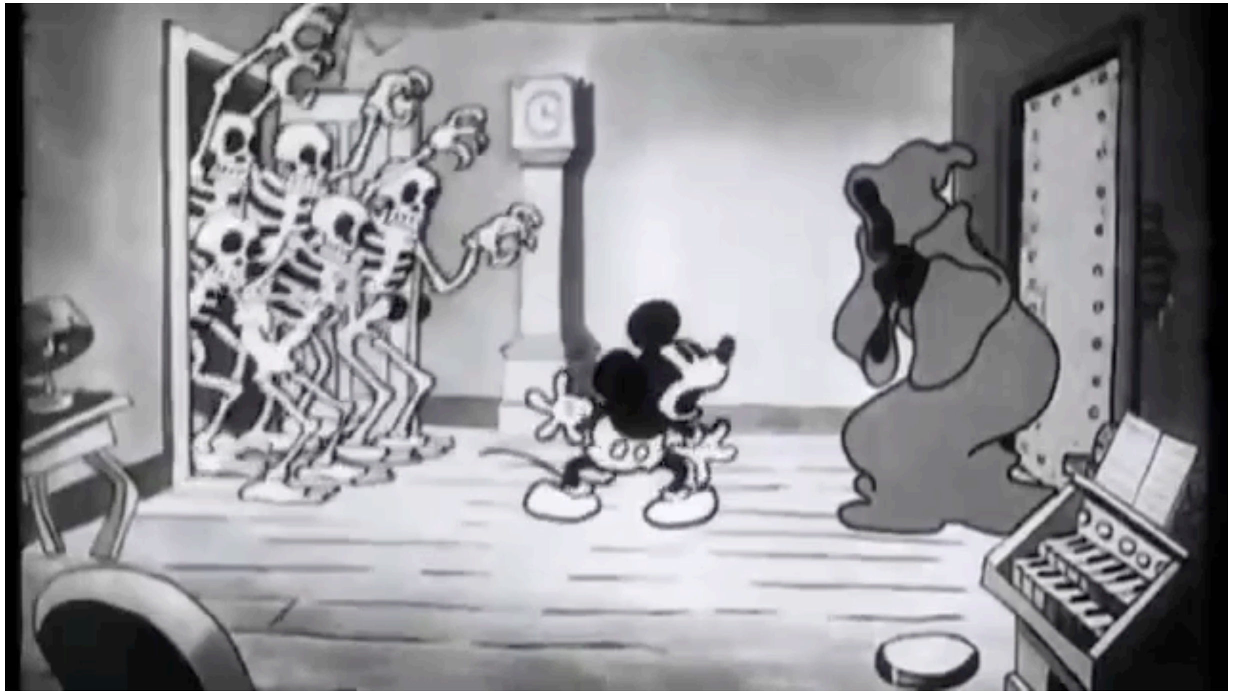
Resulting image without texture using VGG 16



Resulting image with texture using VGG 16



Results - Video

Video results can be viewed in the GitHub repo.

Video: courtesy of Walt Disney

Conclusions

- Thus, we would like to make an inference that VGG-19 outperforms VGG-16 in the Style Transfer Task.
- This task has enabled us to explore other interesting problems such as adding texture to images along with style, stylizing the videos.
- The style transfer task does not need many epochs over entire dataset to train the network as opposed to classification tasks which require many epochs.
- As future work we would like to extend this approach to other forms of input such as audio and text.

References

- Johnson et al (2016), Perceptual Losses for Real-Time Style Transfer and Super-Resolution.
- Simonyan et al (2014), Very Deep Convolutional Networks for Large-Scale Image Recognition
- [4] - Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context
- [5] - Misgod, fast-neural-style--transfer, (2017), Github Repository, https://github.com/misgod/fast-neural-style-keras
- [6] - https://opencv.org/
- [7] Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR. (2015)
- [8] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: ICLR Workshop. (2014)
- [9] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. In: ICML Deep Learning Workshop. (2015)
- [10] Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: NIPS. (2015)
- [11] Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
- [12] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)