Contents:

Table of Figures:

1. Introduction:

It is possible to achieve artistic style transfer within a purely image processing paradigm. This is in contrast with deep neural networks to learn the difference between "style" and "content" in a painting. Style transfer is the technique of recomposing images in the style of other images. It all started when Gatys et al. published an awesome paper on how it was actually possible to transfer artistic style from one painting to another picture using convolutional neural networks. The "style fusion" concept is used that help the algorithm to follow broader structures of style at a higher level while giving it the freedom to make its own artistic decisions at a smaller scale.

Specifically, we want to create an image filter that transfers the style of one painting to the content of another image. Our results are comparable to the neural network approach, while improving speed and maintaining robustness to different styles and contents.

2. Dataset:

The dataset for image processing is obtained randomly. All input images are 400x400 pixels, with 3 RGB color channels. We take two types of images to achieve our final output as Artistic Style Image as:

i. **Content Image:** The pure realistic image on which we want to transfer style and make it an artistic image.
ii. **Style Image:** The texture style image whose texture we want to transfer on content image.

3. Implementation:

We implement our project in MATLAB by using CNN algorithm.

4. Algorithm:

In our project, we use Convolutional Neural Network model. Basically, the convolutional neural network (CNN) is a class of deep learning neural networks. Instead of a fully connected network of weights from each pixel, a CNN has just enough weights to look at a small patch of the image. Convolutional Neural Networks (CNNs) are a category of Neural Network that have proven very effective in areas such as image recognition and classification. CNNs have been successful in computer vision related problems like identifying faces, objects.

The core of the algorithm consists of six steps: style fusion, patch matching, style synthesis, content fusion, color transfer, and denoising.

4.1. Initialization:

Some pre-processing and initialization is done on the input images prior to running the algorithm.

Figure 1: Left>Content Image; Right>Style Image

4.2. Color Transfer:

The first step is to apply a color transfer from the style image to the content image. This color transfer process is also used inside the algorithm. After this, a strong Gaussian Noise is added into the picture. The purpose of Gaussian noise is to mimic the effect of many random processes that occur in nature.

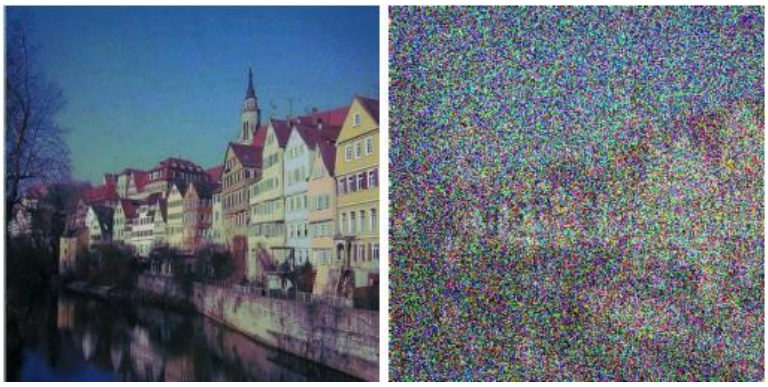

Figure 2: Left>Adding Gaussian noise; Right>Color Transfer

4.3. Style Fusion:

Image fusion is the process of combining relevant information from two or more images into a single image. The resulting image will be more informative than any of the input images.

Figure 3: Left>Style Image; Right>Result obtained by Style Fusion

4.4. Patch Matching:

The objective of patch matching is to extract areas of the style image that mimic areas of the content image. To do this, we first choose a patch in the content image that we want to find a style-match for. Then, we perform a nearest-neighbor (NN) search in the style image : we iterate through patches in the style image and select the patch which has the lowest L -norm with respect to the content patch.

Figure 4: Patch Matching from Style to Content Image

4.5. Style Synthesis:

We now attempt to use the style patches, found using NN, to estimate the output image. In essence, each patch in the output image should roughly match the corresponding NN style patch.

Figure 5: Style Synthesis

4.6. Content Fusion:

To ensure that the original content is preserved, we use a pixel-by-pixel weighted average of the content image and the estimated style image. In areas where there is more "content," the weight will be heavier on the content image. In areas where there is less "content," the weight will favor the estimated image. The pixel-by-pixel weight mask is calculated by performing a series of segmentation algorithms on the content image.

The content is usually in the foreground of the image, we assign the background of the content image a small weight and the foreground a larger weight. Lastly, to make transitions between high content areas and high-style areas smoother, we apply a Gaussian smoothing to the mask.

Figure 6: Content Fusion

4.7. Style Transfer:

The color palette of the final output image can be chosen to be anything desired. For example, we could either keep the color palette of the original content image or take the colors from the style image. Since, the colors used in the style image can be interpreted as part of the style, we chose the latter approach. This also results in more visually pleasing results since there is generally more diversity in

the color palette of the style image. So, we are transferring style from style image to the content image by giving it an Artistic Style Transfer look.

4.8. Denoising:

The last step of the inner-most loop is to restore our image, which has been heavily processed, to the prior statistics of a general image.

5. Result:

At the end of implementation, we obtained the results based on our algorithm. Our algorithm obtains fairly robust style transfer for a variety of different style and content images.

5.1. Failure Cases:

- **Incompatible style image:** One case where our algorithm may not produce pleasing results is when the style image does not provide a good representation of the style.
- **Incompatible content image:** The most common failure mode for incompatible content images is when our segmentation does not successfully detect areas of relevant content.

6. Conclusion:

We reached at the conclusion, that CNN algorithm successfully helps us in style transferring to content image and give it an Artistic look which is a branch of deep learning. However, our algorithm takes 20-30 min for style transferring. This computational speed may vary but 15 min at least required for artistic look otherwise we require GPUs for robustness.

7. Outputs:

We take the two input images for our algorithm testing. The first image is content image that we want to transfer into Artistic image that is taken randomly of dimension 400*400 in RGB colors and the second one is style image whose style we want to apply on another image. After passing both images through the core of the algorithm step by step, the final output image after adapting Artistic Style looks like:

*Figure 7: **Left**>Content Image of 400*400 in RGB; **Middle**>Starry Night Style Image; **Right**> Artistic Style Image*

8. References:

[1] E. Wang and N. Tan, "Artistic Style Transfer," *Digit. Image Process.*, pp. 1–7, 2018.

[2] A. Handa, P. Garg, and V. Khare, "Masked Neural Style Transfer using Convolutional Neural Networks," *2018 Int. Conf. Recent Innov. Electr. Electron. Commun. Eng. ICRIEECE 2018*, pp. 2099–2104, 2018.

[3] H. Hua, K. Teo, G. Colas, and A. Deng, "Drawings from Photos.", pp 1-5, 2017.

[4] M. Elad and P. Milanfar, "Style Transfer Via Texture Synthesis," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2338–2351, 2017.