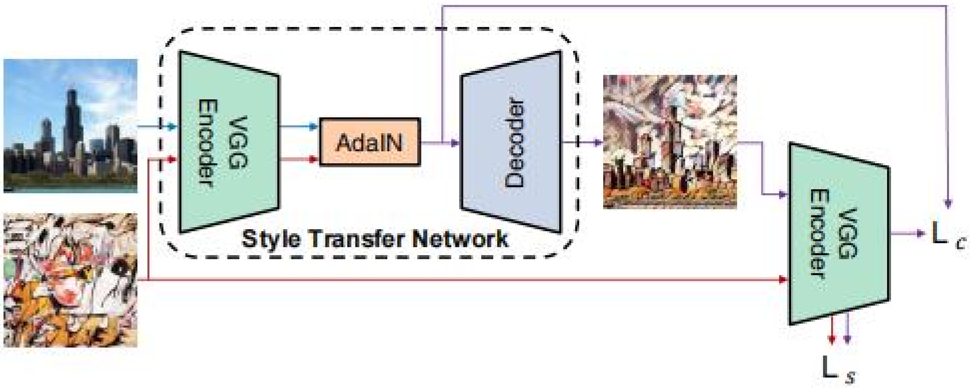
CEG5304 C A 3

Zhang Bokun

A0263813E

1. Architecture - AdaIN



Net structure:

The network architecture consists of an encoder-decoder architecture with residual blocks. The encoder is a pre-trained VGG-19 network that extracts features from the input content image and style image. The decoder network uses upsampling layers to reconstruct the stylized image from the extracted features.

The AdaIN operation is used to combine the content features and style features. It performs a normalization of the mean and variance of the content features with the statistics of the style features.

Code and pretrained models are available at: https://github.com/xunhuang1995/AdaIN-style

2. How it works

The pipeline of this method consists of several steps, including preprocessing, encoding, AdaIN, decoding, and postprocessing. Here is a detailed explanation of each step:

Preprocessing: The input content image and style image are first preprocessed by resizing them to a fixed size and normalizing them to have a mean of 0 and standard deviation of 1. This step ensures that the images have a consistent size and pixel value range for further processing.

Encoding: The VGG-19 network is used to extract feature maps from the preprocessed content image and style image. The feature maps represent different levels of abstraction of the images and contain information about texture, color, and shape.

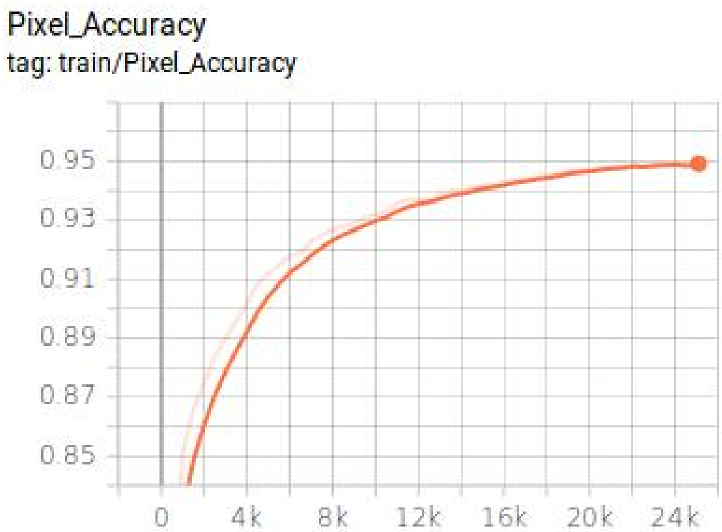
AdaIN: The adaptive instance normalization (AdaIN) operation is then performed on the content feature maps. AdaIN is a normalization technique that adjusts the mean and standard deviation of the feature maps to match those of the style feature maps. This operation aligns the statistics of the content and style feature maps to enable the transfer of the style from the style image to the content image. The resulting stylized feature maps are then fed into the decoder.

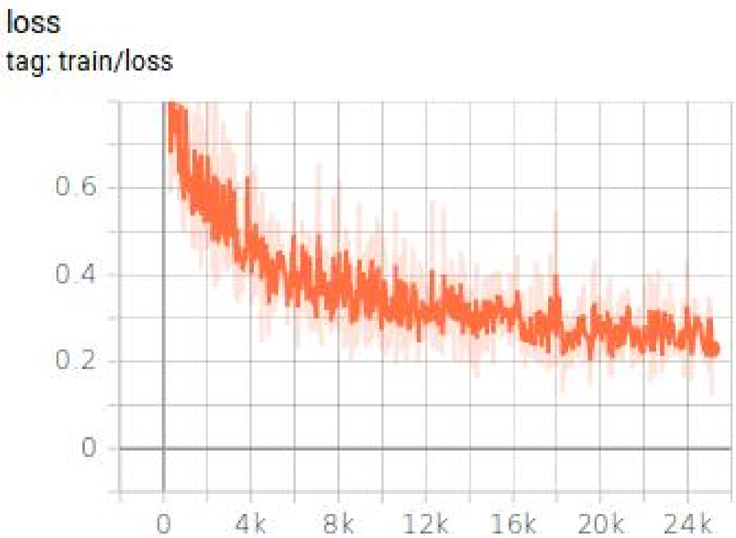
Decoding: The stylized feature maps are then decoded using an upsampling network to reconstruct the stylized image. The upsampling network uses upsampling layers to increase the spatial resolution of the feature maps and a series of convolutional layers to reconstruct the image. The decoder network is trained to reconstruct the stylized image from the stylized feature maps while preserving the content information of the input image.

Postprocessing: Finally, the stylized image is denormalized by multiplying the pixel values by the standard deviation of the original input image and adding back the mean. The pixel values are then clipped to ensure that they fall within the valid range of [0, 255] for display or storage.

The pipeline is repeated for each frame of a video sequence to achieve real-time style transfer. Overall, this method combines the strengths of neural style transfer and instance normalization to enable fast and flexible style transfer for a wide range of applications in computer vision, graphics, and multimedia.

3. Training





We use all of the seven thousand content pictures and three hundred style pictures to train the VGG model. And we see that the accuracy is over 95% and the final loss is under 20%.

4. Testing

Figure1 Part2-4-1

Figure2 Part2-4-1-Monet

Figure3 Part2-4-2

Figure4 Part2-4-2-Monet

Figure5 Part2-4-3

Figure6 Part2-4-3-Monet

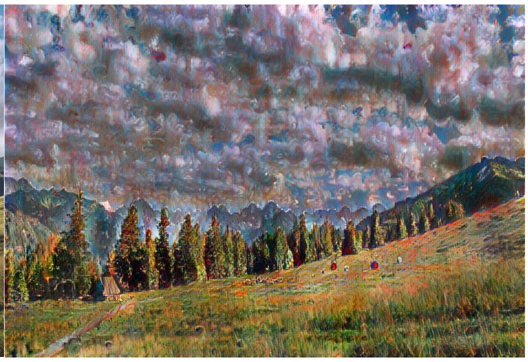


Figure7 Part2-5-1

Figure8 Part2-5-1-Monet

Figure9 Part2-5-2

Figure10 Part2-5-2-Monet

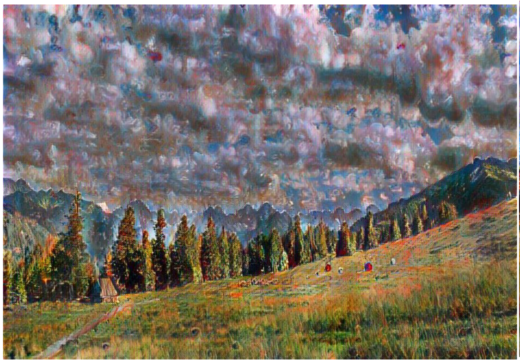
Figure11 Part2-5-3

Figure12 Part2-5-3-Monet

5. Comparison

The following four pictures are transformed:






The following four pictures are randomly picked in Monet’s works:


Advantages:

Actually, we can see the results are so close to the Monet's style of works. His work style emphasizes the effect of color and light, showing strong changes in color and light. So there are many colors and strong changes in the transformed pictures.

Besides, the results saved the origin shape and features in pictures at the same time. Sometimes Monet's work is abstract and unshaped. We cannot tell things. However, with AdaIN method, we saved the original features and make it clear.

Disadvantages:

Limited style transfer quality: While the method can transfer the style of the style image to the content image, the resulting stylized images may not always be of the highest quality. This is because the quality of the stylized images depends on the quality of the style image, and some styles may not be well-suited for certain content images.

Dependence on pre-trained models: The method relies on a pre-trained VGG-19 network to extract features from the content and style images. This means that the quality of the stylized images may be limited by the quality of the pre-trained models and the dataset used to train them.

Limited flexibility: The method is designed for real-time style transfer and may not be as flexible as some other style transfer methods. For example, the method may not be able to transfer multiple styles to a single content image or perform style transfer across different modalities such as text or audio.

Computationally intensive: While the method is designed for real-time style transfer, it still requires significant computational resources to run. This may limit its use on devices with limited processing power.

Solutions:

Limited style transfer quality: One possible solution is to sharpen images to improve the quality of the stylized images. Another possible solution is to use advanced deep learning techniques, such as Generative Adversarial Networks (GANs), to improve the quality of the stylized images.

Dependence on pre-trained models: One possible improvement is to use a more advanced pre-trained model, such as a larger and more diverse dataset for training. Another possible improvement is to use transfer learning techniques to fine-tune the pre-trained model on a specific dataset.

Limited flexibility: One possible solution is to use a more flexible style transfer method, such as neural style transfer or universal style transfer, which can handle multiple styles and modalities. Another possible solution is to combine style transfer with other image processing techniques, such as image segmentation or object detection.

Computationally intensive: One possible improvement is to use more efficient hardware or software optimization techniques to reduce the computational cost of style transfer. Another possible improvement is to use real-time style transfer techniques that are specifically designed for low-power devices, such as mobile phones or embedded systems.

Before After

we sharpen images to improve the quality of the stylized images. And we use Generative Adversarial Networks (GANs), to improve the quality of the stylized images.

It is more clear than before.