Report

R11944010 黃舒盟 R11944029 黃敬媛 B09902048 王品文

File Structure

```
group_11
|
--------code
|
--------data
|       |
|       -----------styles
|       |
|       -----------image_results
|       |          |
|       |          -------------AesUST_result
|       |          |
|       |          -------------WISE_result
|       |          |
|       |          -------------STROTSS_result
|       |
|       -----------combined_results
|       |
|       -----------video_results
|                  |
|                  -------------AesUST_result
|                  |
|                  -------------Fast_Artistic_result
|                  |
|                  -------------ReReVST_result
|
--------report.pdf
|
--------README.md
```

Styles used

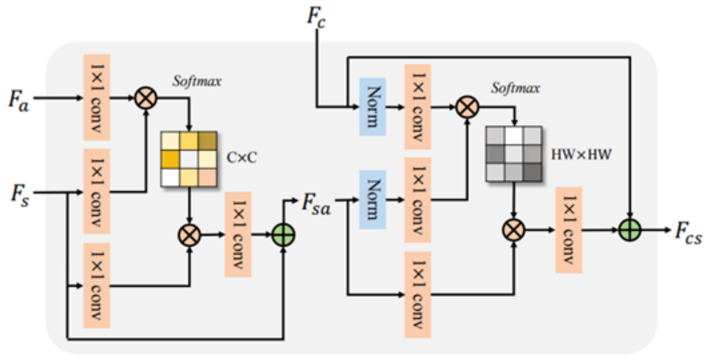
We use different images as input styles, including different paintings, movie posters and other artworks, for final project. Style images are under `data/styles`

Image Method

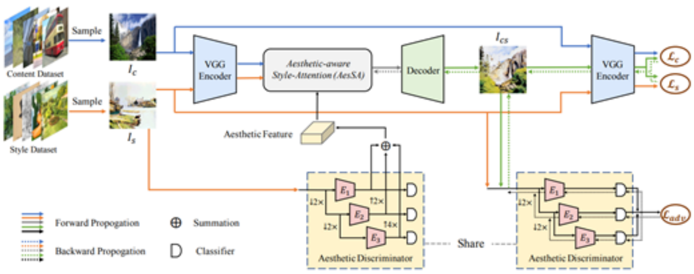
We survey for different methods for single image style transfer, and use them with open source on github.

AesUST:

AesUST is an universal style transfer method that gives promising results. Since it separates style image's content, style and aesthetic part, the content image can performs better with the style given, rather than having contents that doesn't belong to it. (For example, in some early machine learning method for style transfer, the content image might have flowers in the background even though it does not have any at first.) Moreover, by separating the color and style, one can have color-preserved but style transferred output. (So we won't see red lake or purple ocean.) Down below is the part showing how to combine F_a, F_s, F_c and use them for style transferring



The model of AesUST is given below. The discriminators are the same but they performs different roles in different stages. The first stage it performs as an ordinary feature extractor; while at the second stage, it is used as a loss

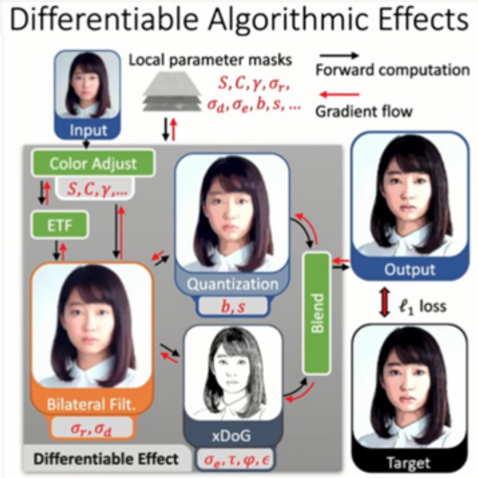


calculator.

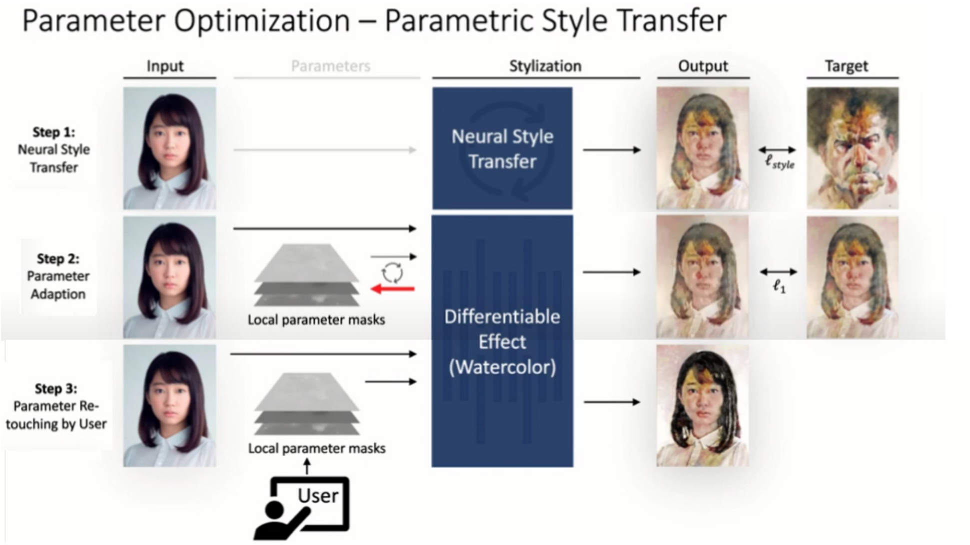
We decide to use this method because it is fast and the results are promising.

WISE:

WISE is a new style transfer paradigm, which combines algorithmic image processing filters and machine learning based approaches.
Below is an example which generated a cartoon style image through filter-based image stylization pipelines. We can notice that there are many parameters values for these filters. WISE's target is automatically optimizing these parameter values via gradient descent to make the ouput image match the target image.



The process of WISE is down below. First, we execute an Neural style transfer method to generate a stylized image. Second, we train local parameters of image processing filters to make the output image matching the target stylized image. Third, Since the parameters of image filters can still be adapted by the user, the style transfer results remain editable.



Becuase wise is an editable style transfer method and its author provide a beautiful user interface, We choosed it as one of our image method.

STROTSS:

STROTSS is used by the authors of WISE as its Neural style transfer method, so we also list it here.

Figure 1: Examples of our output for unconstrained (left) and guided (right) style transfer.Images are arranged in order of content, output, style. Below the content and style image on the right we visualize the user-defined region-to-region guidance used to generate the output in the middle.

STROTSS is a new optimization-based style transfer algorithm that allows user-specified region-to-region control over visual similarity between the style image and the output.

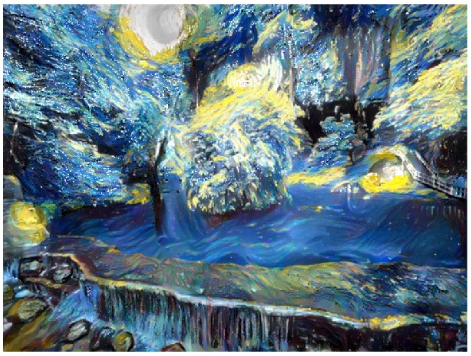
WISE vs. STROTSS:

As mentioned above, The output image of WISE will be approximated to The stylized image of STROTSS.
For Samurai Chicken image, the WISE's result is quite well.

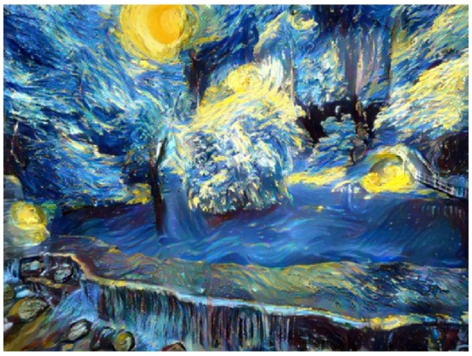
WISE

STROTSS

For Scenery Image, the WISE's result is worse. There're some undesirable white noises. So, directly use STROTSS is better if you don't need to edit the image.



WISE

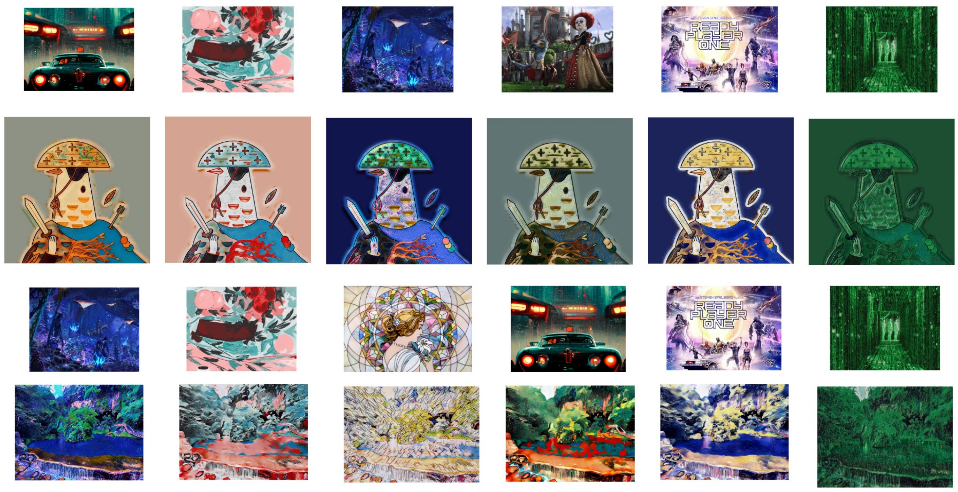


STROTSS

Image Result

Here is the results of the above two method. The high resolution version are placed in `data/image_results` .

AesUST



WISE

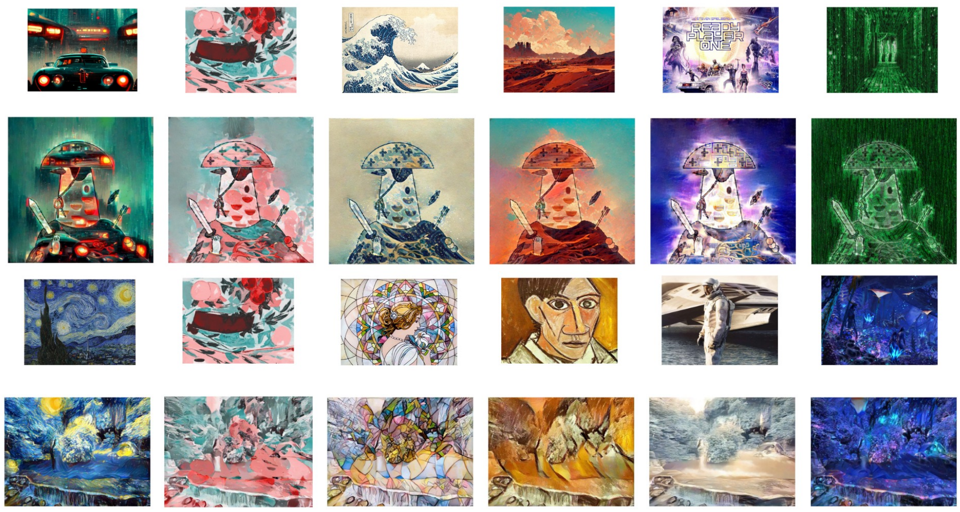


Image Method Comparison

- AesUST: transfer rate fast; color conversion well; style strokes not obvious
- WISE: can be applied to any style transfer method; editable; takes extra time for optimizing the parameter values of image filters.
- STROTSS: transfer rate slow; obvious change in color and stroke; The strong style might distort content’s object’s edges

Streamlit User Interface

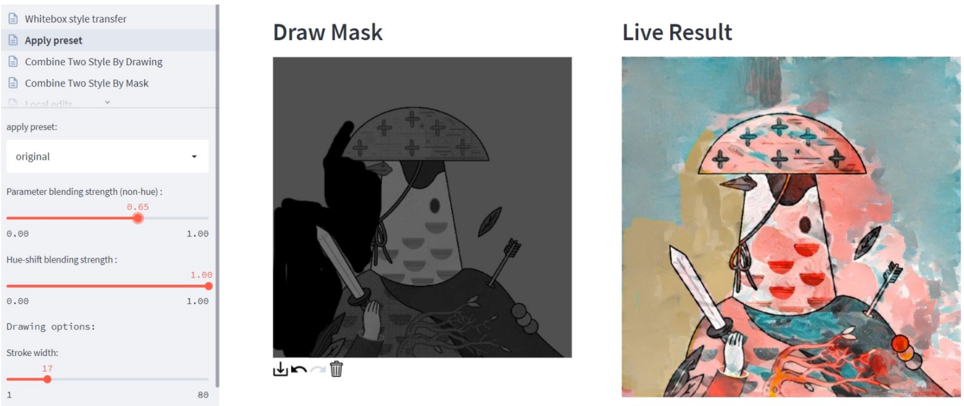
The Interface is based on the editing demo UI which provided by WISE’s authors. We also add other extensions, like **Combine Two Style** Pages.

Here is the Demo Video Link: **https://www.youtube.com/watch?v=QDhb_q-CWzY** (https://www.youtube.com/watch?v=QDhb_q-CWzY)

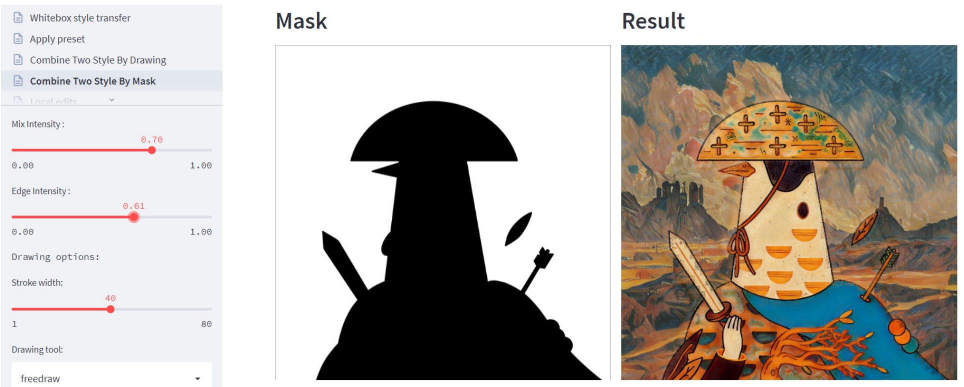
The usage is in readme file: **https://github.com/yuan7765/dipfinal/blob/main/README.md** (https://github.com/yuan7765/dipfinal/blob/main/README.md)

Image Method Combination – Stylized Image with Stylized Background

We're inspired by WISE's Apply Preset Page that user can draw a mask to make some local changes in it as a preset.



If we can apply another stylized background image in mask part, we believe that the output image will be more interesting. Below is some examples, and the high resolution results are placed in `data/combined_results/`



Video Method

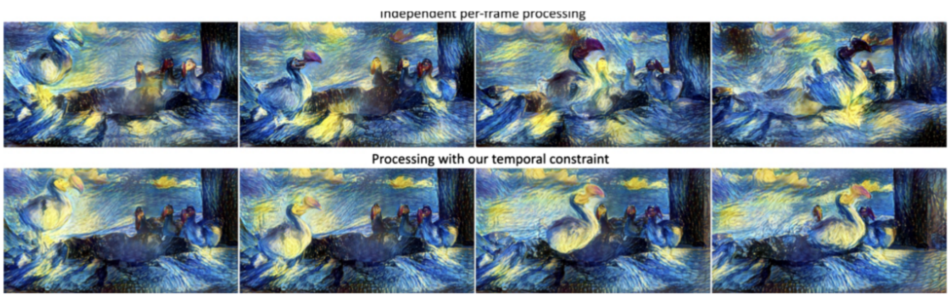
What we are trying to do is to find some methods with obvious style differences and no flickering problems. Try different styles to find a suitable style video. In the report, the file type will be displayed as a screenshot, and the complete result will be placed in `data/video_results` .

Single Image Combination (use AesUST as example)

Since AesUST performs well on single image, and most video style transfer methods' output have strange stripes in the background of 武士雞, we decide to see if there is any possibility that using single image combination can improve or at least decrease this situation. Although AesUST performs quite well on each single picture, and the strange stripes in the background disappears, the flickering issue is serious for 社科圖 video. Therefore, we use 2 other methods in order to have better video results.

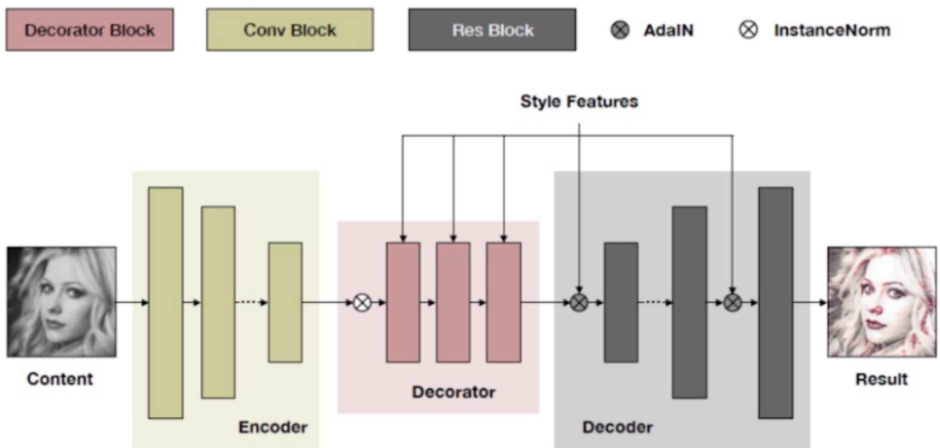
Fast Artistic

Fast Artistic use approaches introduced by Gatys et al. "A Neural Algorithm of Artistic Style "and Johnson et al. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution” which are used as basis for their video style transfer algorithms. Used in the basis of the above two articles to introduce new ways of initialization and new loss functions to generate consistent and stable stylized video sequences even in cases with large motion and strong occlusion.This method optimizes the deep network architecture and training procedure in previous methods to stylize videos of arbitrary length in near real-time in a consistent and stable manner. Transformation results are significantly and quantitatively better than simpler baselines, also slowing flickering issues.The figure shows more consistent results



ReReVst

The method is base on paper "Consistent Video Style Transfer via Relaxation and Regularization ". For the conflict of style transfer and temporal consistency, ReReVST proposes to reconcile this contradiction by relaxing the objective function to make the stylization loss term more robust to motions. Through relaxation, style transfer is more robust to inter-frame variation without degrading the subjective effect. They are using a zero-shot video style transfer framework and a new module that can dynamically adjust inter-channel distributions. Quantitative and qualitative results demonstrate the superiority of our method over state-of-the-art style transfer methods.



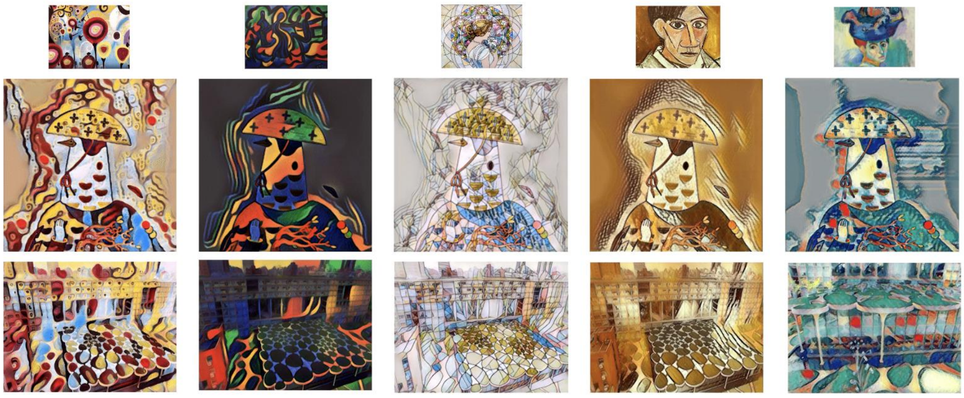
The architecture diagram (Figure 1) shows that ReReVST uses two decodings to reduce the problem of flickering

Video Method Comparison

Model	AesUST	Fast Artistic	ReReVST
Pros	Use the same set of style parameters	The effect of style transformation is remarkable	Almost no flickering issues
	Remarkable in style	Less flickering issues	Method is efficient
	No weird textures appear in the background	- -	Can replace many different styles
Cons	Serious flickering issues	Long operation time; Can’t add new styles quickly	The effect of style transfer is less obvious

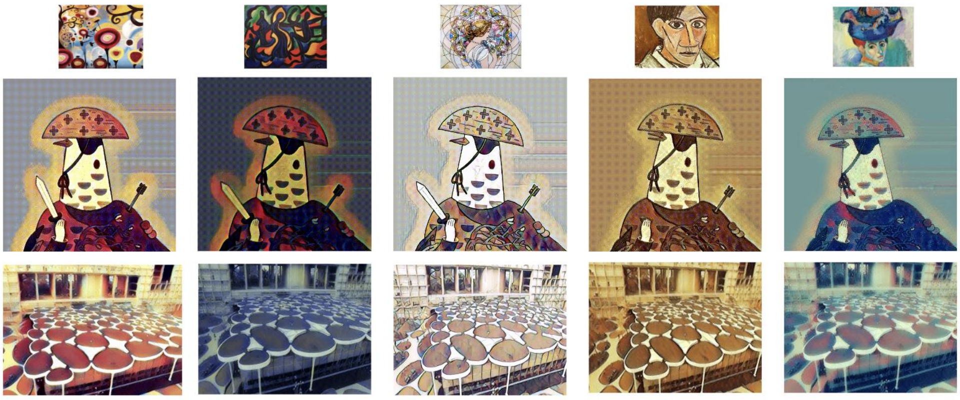
Video Result

Fast Artistic



We think the effect of converting the texture is remarkable. After converting the paintings with strong style and the style image with bright colors, there are reasonable results. In this picture, we can see that the texture and key color of the style image are obvious on the 武士雞 and 社科院空拍 . But in the content with a simple back like the samurai chicken, there will be strange textures in the background. We guess that it is because the result of the previous picture is used for calculation, so the background will be affected by the animation of leaves or swords. If it is not used in the previous picture, go Operations like integrating results with the AesUST method do not have the same problem.

ReReVst



On the ReReVST side, we can find that due to the conversion method, the background has no strange textures, but for style images with complex colors, the conversion results of ReReVST are relatively insignificant. Although the color of the style image cannot be clearly seen from the result, it can still transform the characteristics of the style image with a particularly strong style or a fluorescent color.

Conclusion

- Fast Artistic style conversion effect is remarkable, although it takes a long time, it is suitable for style conversion of images with rich pictures
- The effect of ReReVST conversion is less noticeable, but if choose a style with a large contrast and obvious strokes, you can also get good results

Reference

1. Wang, Z., Zhang, Z., Zhao, L., Zuo, Z., Li, A., Xing, W., & Lu, D. (2022, October). AesUST: towards aesthetic-enhanced universal style transfer. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 1095-1106).
2. Wang, W., Yang, S., Xu, J., & Liu, J. (2020). Consistent video style transfer via relaxation and regularization. IEEE Transactions on Image Processing, 29, 9125-9139.

3. Ruder, M., Dosovitskiy, A., & Brox, T. (2018). Artistic style transfer for videos and spherical images. International Journal of Computer Vision, 126(11), 1199-1219.
4. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. “A neural algorithm of artistic style.” arXiv preprint arXiv:1508.06576 (2015).
5. Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution.” European conference on computer vision. Springer, Cham, 2016.
6. Ripken, Winfried & Reimann, Max & Büßemeyer, Martin & Semmo, Amir & Döllner, Jürgen & Trapp, Matthias. (2022). WISE: Whitebox Image Stylization by Example-based Learning. 10.48550/arXiv.2207.14606.
7. Kolkin, Nicholas & Salavon, Jason & Shakhnarovich, Greg. (2019). Style Transfer by Relaxed Optimal Transport and Self-Similarity.