DIP Term Project Report: Style Transfer Experiment

Team members: 符嘉文 R10944057, 陳允玟 R11922051, 童嬿瑜 B09902031

Summary

In previous paper presentation, we explored the concept of style transfer from Johnson et al.(2015) and performed some simple style transfer tasks. In this final project, we further utilized other different methods for image style transfer and video style transfer, with the helps of code and model provided in GitHub. We compared the qualitative and quantitative results of these methods, and implemented a simple graphical user interface for the style transfer tasks.

1. Methods for Image and Video Style Transfer

We selected 5 different style transfer methods. Figure 1 below shows our categorization of methods used in the experiments.

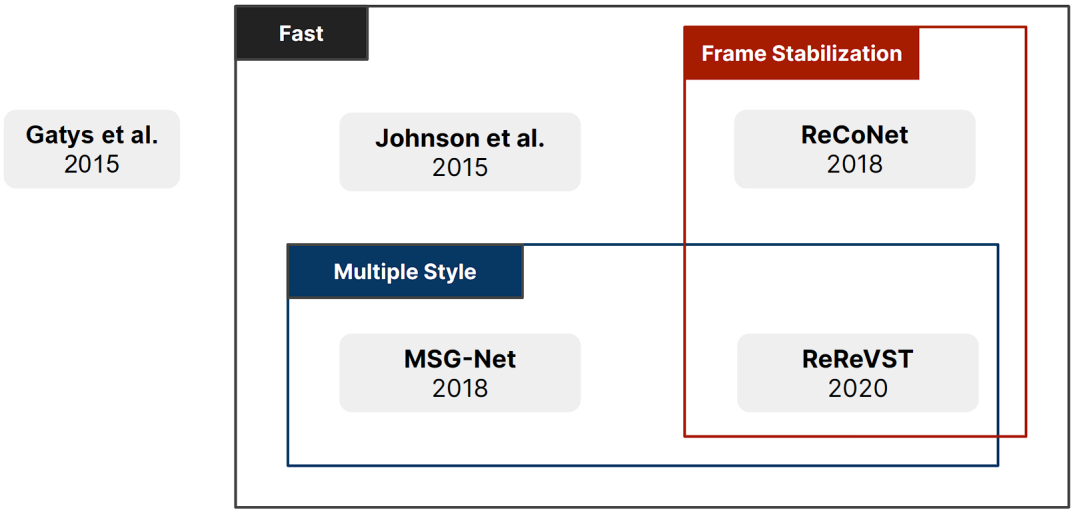
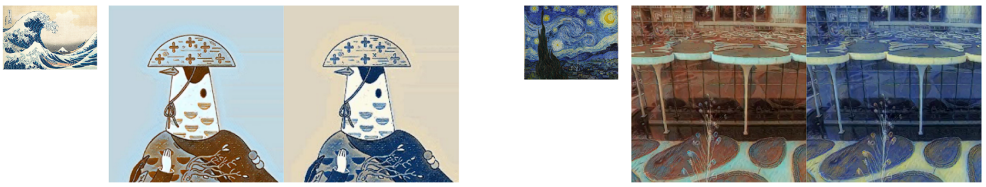


Figure 1: Methods used in this experiment categorized by (1) transfer speed, (2) multiple style support and (3) Implementation of frame stabilization

Among these methods, we used **Gatys et al.(2015)**, **Johnson et al.(2015)**, and **MSG-Net(2018)** for single image style transfer experiments. We want to compare performance of slow and fast methods, as well as results from models trained for multiple styles and single styles.
For video style transfer, the method from Gatys et al.(2015) is excluded because it takes longer time for a single frame. We used methods from **Johnson et al. (2015)** and **MSG-Net(2018)** which feature fast style transfer. We also added methods that featured frame stabilization from **ReCoNet(2018)** and **ReReVST(2020)**. ReReVST supports multiple style transfers while ReCoNet does not.

2. Code Implementation for different methods

1. **A neural algorithm of artistic style. (Gatys et al. 2015):**
   We use code from tjwhitaker for this method in image style transfer tasks. We added transfer time calculation in the code for this experiment. Code modified is available as chenyunwen/a-neural-algorithm-of-artistic-style
2. **Perceptual losses for real-time style transfer and super-resolution. (Johnson et al. 2015):**
   We cloned code repository from tyui592 for this method in image and video style transfer tasks. This method only support single image style transfer, we trained four models for the selected style images. For video style transfer tasks, we modified the code to implement frame to frame transfer. Our modified code and models is available at chenyunwen/Perceptual_loss_for_real_time_style_transfer
3. **Multi-style generative network for real-time transfer. (MSG-Net)**
   We cloned code repository from zhanghang1989 for this method in image and video style transfer tasks. The original code take camera input for video style transfer and we altered it to use the videos provided by this course. This method supports multiple style transfer so we run model from the original repository and does not trained our model. The code is available at chenyunwen/PyTorch-Multi-Style-Transfer
4. **Reconet: Real-time coherent video style transfer network. (ReCoNet)**
   We cloned code repository from liulai for this method in video style transfer tasks. The original code load sequence images for video style transfer, and we altered it to take video as input. Our code also detect the framerate and frame size of the input video for saving the output. We trained four model for the selected style images, (which take very very long time). Our code and model is available at chenyunwen/reconet-torch.
5. **Consistent video style transfer via relaxation and regularization. (ReReVST)**
   We cloned code repository from daooshee for this method in video style transfer tasks. We use the model provided in the repository as it support multiple style transfer. We encountered insufficient GPU memory when doing video style transfer, so we use CPU for the tasks. Besides, we found that the original code produced unexpected results due to missing BGR-RGB convert upon saving. Figure 2 below shows the unexpected results and the results fixed after RGB conversion. Code modified is available at chenyunwen/ReReVST-UX-Wrapper



(a)Wave+video_0 (b)Starry Night+video_1

Figure 2: Unexpected results in original ReReVST code and results fixed for video style transfer

3. Results of Image and Video Style Transfer Experiments

We selected the four artworks as style image for our image and video style transfer experiment, as shown in Figure 3:

1. The Muse, 1935, Pablo Picasso
2. The Great Wave off Kanagawa (神奈川沖浪裏), 1831-1833, Katsushika Hokusai (葛飾北斎)
3. The Starry Night, 1890, Vincent van Gogh
4. Arles: View from the Wheat Field, 1888, Vincent van Gogh

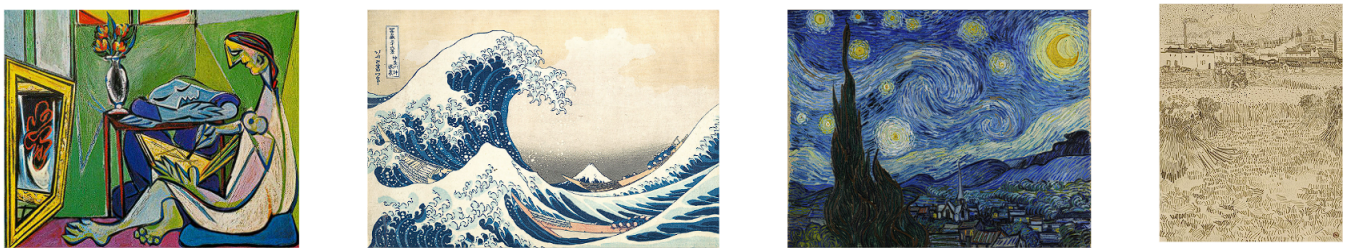


Figure 3: Artworks selected as style images

We found these artworks are often selected in style transfer experiment because they have very distinct visual styles, textures and brushstroke.

3.1 Qualitative Results Comparison of Image Style Transfer

Figure3.1 below show the results of image_0 and image_1 transferred under 3 different methods using 4 style images:

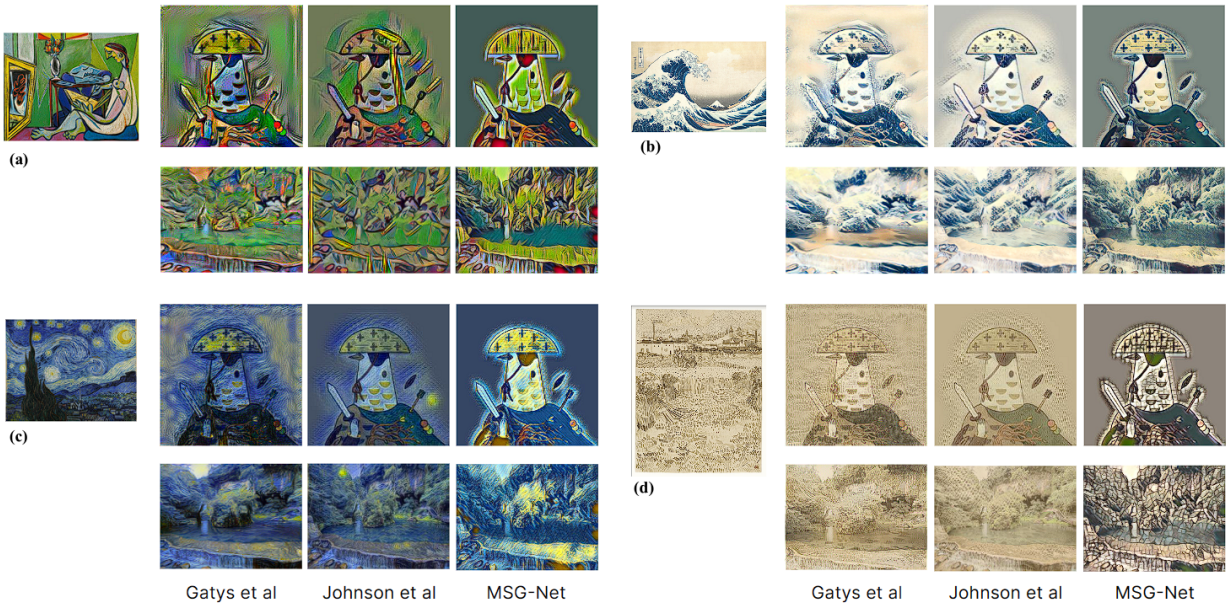


Figure 3.1 results of image style transfer

The detailed images are available in the repository submitted. From the results above, we can see that all three methods could produce pleasing style transfer results that take color schemes, textures and brush strokes from the style images provided and apply on the contents, except in (d) , we observed that results from MSG-Net have some unexpected texture on the output. We also noticed MSG-Net increases the contrast of the images after transfer.

3.2 Qualitative Results Comparison of Video Style Transfer

Figure3.2 below shows the results of video_0 and video_1 transferred under 4 different methods using 4 style images:

Figure 3.2 results of video style transfer

The animated videos are available in the repositories submitted. From the results above, we can see that ReCoNet and ReReVST produce smoother video results with less flickering, but have less obvious textures and brush strokes from the style images, compared to Johnson et al.(2015) and MSG-Net. We also found some unwanted magenta noise in videos results from ReCoNet, due to some unknown bugs in the code. We noticed that results from ReReVST are almost based on the color scheme from the style images and lose color from the contents.

3.3 Quantitative Results Comparison of Image and Video Style Transfer

We also compare the transfer speed of different methods for images and videos style transfer. Table 3.3a below list time needed for single image style transfer:

Time needed for single image style transfer (s)

Method	**Gatys et al.**		**Johnson et al.**		**MSG-Net**	
	image_1	image_0	image_1	image_0	image_1	image_0
The muse	5.59	5.74	0.84	0.84	1.76	1.82
Wave	5.73	5.56	0.84	0.84	1.72	1.85
Starry Night	6.02	6.08	0.97	0.93	1.70	1.76
Arles	5.88	6.02	1.07	1.31	1.69	1.79
Average	**5.80**	**5.80**	**0.84**	**0.84**	**1.74**	**1.83**

Table 3.3a Time needed for single images style transfer of three methods

Figure 3.3 below is a visual chart presentation of above data.

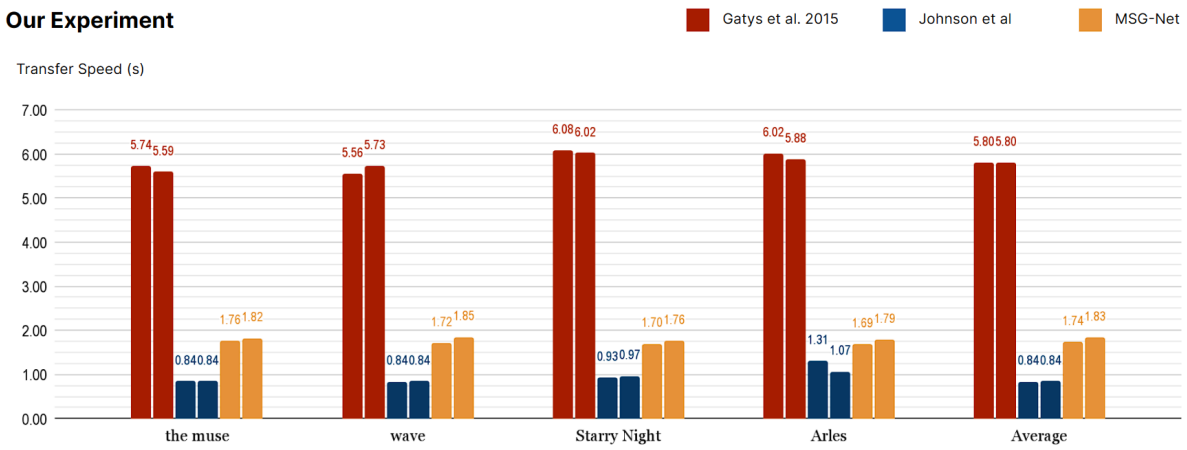


Figure 3.3a Time needed for single images style transfer of three methods

We found that Johnson et al. perform best in speed and MSG-Net is slightly shower. Gatys et al.take much more longer time which make it nearly impractical to use this method for video style transfer (would take more than half an hour to process each video with 360 frames)

Table 3.3b below list time needed for single image style transfer:

Time needed for video style transfer (s)

Method	**Johnson et al.**		**MSG-Net**		**ReCoNet**	
	video_0	video_1	video_0	video_1	video_0	video_1
The muse	19.88	24.05	17.14	24.57	8.66	11.58
Wave	19.77	23.78	17.36	24.67	8.16	11.24
Starry Night	19.57	23.78	17.71	24.90	7.92	11.06
Arles	19.47	23.87	17.00	24.97	7.97	11.46
Average	**19.67**	**23.87**	**17.30**	**24.78**	**8.18**	**11.33**

Table 3.3a Time needed for videos style transfer of three methods

Figure 3.3 below is a visual chart presentation of above data.

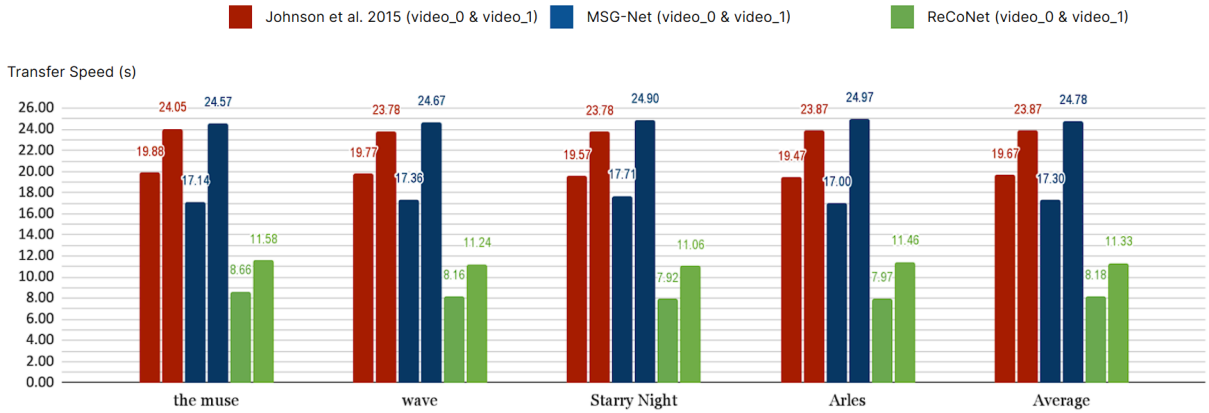


Figure 3.3b Time needed for single images style transfer of three methods

When it comes to video style transfer, we found that ReCoNet takes the least time to transfer the same video, while MSG-NET and Johnson et al. have approximate speed. We excluded ReReVST from this comparison because we used CPU to perform the task, which drastically increased the time needed to process those videos.

We also compared training speed of methods which only support single style transfer. The time needed to train models for each style is shown in Table 3.3c.

Time needed for model training (s)

Method	**Johnson et al.**	**ReCoNet**
The muse	0:46:35	2:38:30
Wave	0:50:02	3:18:23
Starry Night	0:47:37	2:31:31
Arles	0:47:05	2:30:47
Average	0:47:50	2:44:48

Table 3.3c Time needed for model training of each styles

ReCoNet takes a very long time for model training, compared to Johnson et al. We think that is why universal styles transfer is preferred in many use cases.

4. Graphic User Interface Implementation

Based on the models and codes above, we implemented a web graphical user interface using Gradio for images style transfer. The system supports uploading content images and styles images, as shown in Figure 4a. For methods that only support single style transfer, the system provides 4 styles options with corresponding trained models to choose from, as shown in Figure 4b.

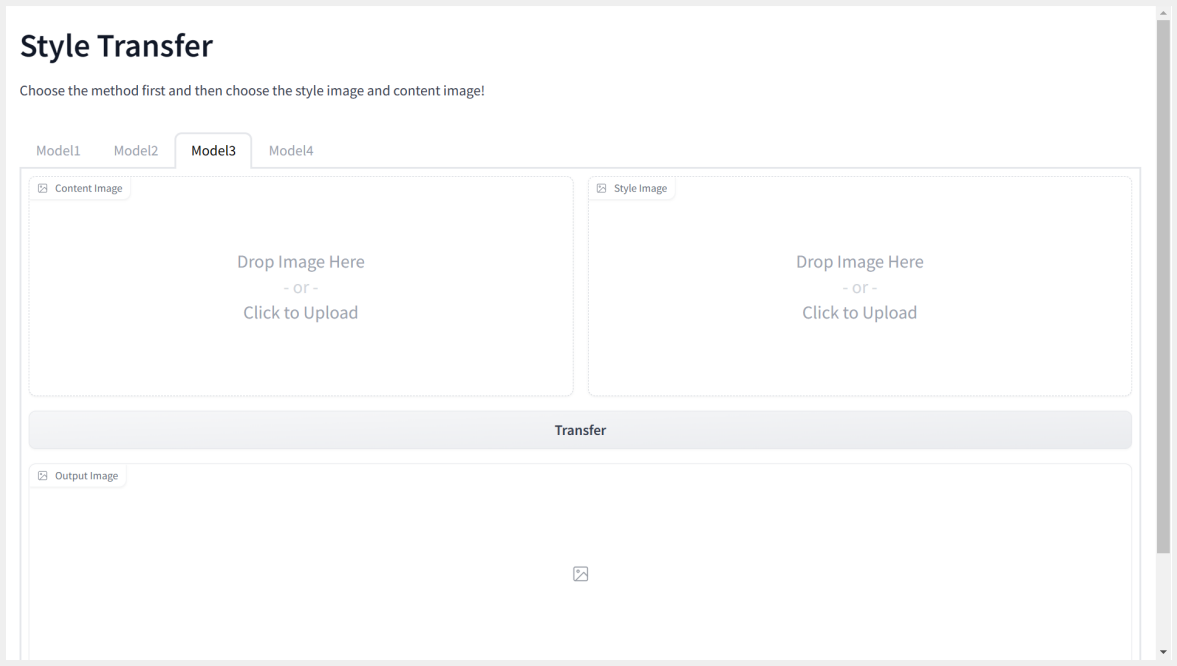


Figure 4a

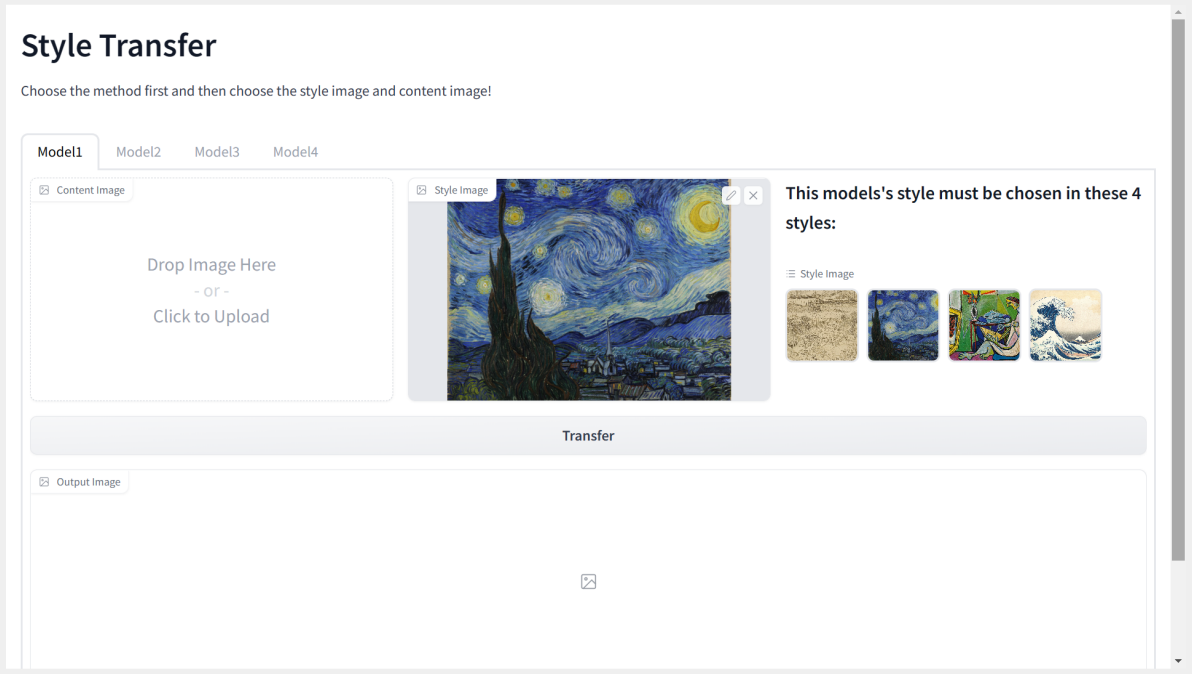


Figure 4b

Conclusion

We concluded the comparisons of the methods in Table 5.

	Gatys et al	Johnson et al	MSG-Net	ReCoNet	ReReVST
Year	2015	2015	2018	2018	2020
Transfer Speed	Slow (5s)	Fast (0.8s)	Fast (1.7s)	Very Fast	N/A
Video deflickering	No	No	No	Yes	Yes
Training Speed per Style	-	40-50 min per style (10000 epoch)	-	~750 min per style (100 epoch)	-
Style supported	1 style	1 style	Multistyle	1 style	Multi-style
Results visually observed	Very obvious textures and brushes strokes	Very obvious textures and brushes strokes	Increase contrast, produced unexpected textures	Produce magenta noise	Lose color from content images.

Table 5 Style transfer methods comparison

It is hard to tell which methods perform best. We suggest picking suitable methods for different use cases based on the requirements. There are many interesting methods not included in this experiment and we could explore them in the future.

References

Papers references:

- Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).
- 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.
- Zhang, Hang, and Kristin Dana. "Multi-style generative network for real-time transfer." Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018.
- Gao, Chang, et al. "Reconet: Real-time coherent video style transfer network." Asian Conference on Computer Vision. Springer, Cham, 2018.
- Wang, Wenjing, et al. "Consistent video style transfer via relaxation and regularization." IEEE Transactions on Image Processing 29 (2020): 9125-9139.

Codes references:

- Gatys et al.2015 - https://github.com/tjwhitaker/a-neural-algorithm-of-artistic-style
- Johnson et al. 2015 - https://github.com/tyui592/Perceptual_loss_for_real_time_style_transfer
- MSG-Net - https://github.com/zhanghang1989/PyTorch-Multi-Style-Transfer
- ReCoNet - https://github.com/liulai/reconet-torch
- ReReVST - https://github.com/daooshee/ReReVST-Code
- PyTorch Documentation - https://pytorch.org/docs/stable/index.html
- OpenCV Documentation - https://docs.opencv.org/4.x/
- Gradio Documentaion - https://www.gradio.app/docs/