CIS 5810: Project Report

Abhishek Goyal, Adwait Agashe, Muskaan Beriwal

11th December 2023

Project Title: Video to Video Artistic Style Transfer

1 Project Summary

In our project, we aim to build upon Neural Style Transfer (NST) by extending its capabilities from static images to dynamic video content. Neural Style Transfer is a deep learning technique that leverages convolutional neural networks to merge the content of one image with the artistic style of another. This process results in the creation of visually striking images that combine the content of one source with the aesthetics of another. Our novel contribution lies in adapting this technique to work with video data, allowing for the extraction of styles from one video and applying them to another. This innovative approach opens up new possibilities for artistic expression and content manipulation in the realm of video content, creating dynamic, style-transformed videos.

To achieve this, we plan to develop and compare custom neural network architectures that can process video frames, capturing the stylistic features from one video and transferring them onto another. This project will involve a multi-step process: first, extracting multiple style frames from a source video, and then applying styles to the target video in a seamless and temporally consistent manner and then transitioning between the styles to generate a single output video. Our goal is to ensure that the style transfer process works smoothly across frames, maintaining the artistic quality of the source style throughout the target video.

2 Goals and Objectives

The primary goal of our project is to transform ordinary videos into visually appealing, unique, and artistic pieces of content. We define the following objectives within the scope of our project:

1. **Develop a foundational model:** A fundamental aspect of our project involves the creation of a foundational model capable of extracting the artistic style from an input image and then applying that style to an output image or video. This foundational model will serve as the core engine for style transfer in videos, enabling the infusion of artistic elements into otherwise ordinary video content.

2. **Implement and experiment with video style transfer algorithms:** Our primary goal is to extract styles from a video input which can then serve as the style input to be applied on another plain video and then merge these videos to get the output video.. We will implement a few video style transfer algorithms, drawing inspiration from established image-based style transfer techniques. These algorithms will be adapted and extended to work seamlessly on video frames, addressing the unique challenges and considerations associated with video data. Key elements of this phase include preserving temporal coherence, maintaining video quality, and minimizing artifacts in the transformed content.

3. **Efficiency and Performance Optimization:** An essential objective is the optimization of the efficiency and performance of the video style transfer algorithms. This optimization process aims to ensure that style transfer can be performed quickly, allowing users to experience dynamic style changes without significant delay.

4. **Quality Evaluation:** We will implement comprehensive metrics and assessment criteria to evaluate the quality of the generated videos. This evaluation process will include both automated metrics and manual assessments to determine the visual appeal and artistic coherence of the transformed content.

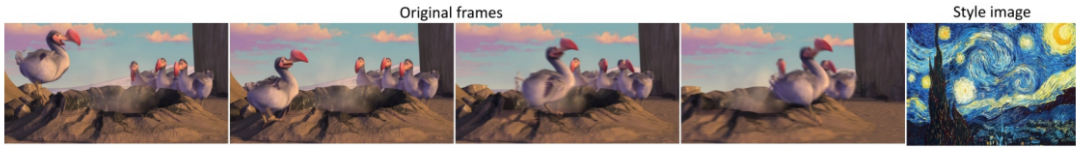

Figure 1: Input for the video style transfer model, based on [2]

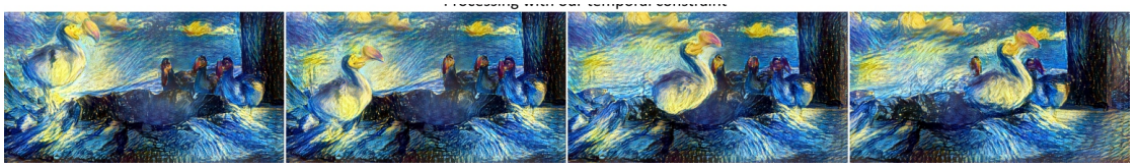

Figure 2: Example output for the video style transfer model, based on [2]

3 Related Works

The paper by Gatys et al. [1] introduces a technique for neural style transfer, which combines the content of one image with the artistic style of another. The paper leverages deep neural networks to separate and recombine content and style features in images, enabling the generation of visually appealing, stylized images. The authors propose a loss function that distinguishes the content and style of images, and an optimization process that minimizes this loss, resulting in the creation of stylized images that mimic the artistic qualities of the style image. This method has had a significant impact on the field of computer vision and has applications in art, design, and image processing, enabling the generation of artwork and stylized images in a unique and creative manner.

The paper by Manuel Ruder et al. [2] builds upon the neural network approach to perform style transfer and extends it to video sequences. To maintain smooth transitions in video style transfer and mitigate flickering, a temporal constraint is introduced that penalizes deviations along point trajectories, considering the optical flow from the original video. A basic approach involves initializing the optimization for frame (i + 1) with the previous stylized frame, allowing unchanged areas to retain their desired appearance; however, in the presence of motion, this approach is improved by considering optical flow and two optical flow estimation algorithms, DeepFlow and EpicFlow, were experimented with for this purpose. In addition, the paper explores two extensions to this work - The first one aims on improving the consistency over larger periods of time using long term motion estimates. Secondly, the authors developed a multi-pass algorithm, which processes the video in alternating directions using both forward and backward flow which results in a more coherent video. Their approach was evaluated on the Sintel benchmark. The authors were able to successfully eliminate most of the temporal artifacts and could create smooth and coherent stylized videos.

Chen et al. [3] implemented video style transfer using feed-forward style transfer networks. They improved consistency by reusing features from the intermediate layers of the style transfer network from the previous frame: they were warped using optical flow and combined with feature maps extracted from the current frame in non-occluded regions before being passed to the decoder part of the network.

4 Approach

Our project is structured around the following key steps:

1. **Creation of a Stylized Image Model:** We will initiate the project by developing a foundational model. This model will accept an input image and a style image, and subsequently generate a stylized image as its output. To optimize our model's performance, we will explore a variety of style transfer networks that have been documented in prior research papers. Our selection criteria will be based on the quality of the output stylized images.

2. **Extension to Video Inputs:** In the next phase, we aim to expand the model's functionality to accommodate video inputs. This enhancement will enable us to generate a coherent and synthesized stylized video. To achieve this, we will experiment with different loss functions to minimize variations in objects across various video frames, ensuring a consistent stylized output throughout the video sequence.

3. **Video Style Extraction and Application:** In the final stage, we will further refine our model by allowing video input for the extraction of multiple styles from the video. These extracted images will then serve as the style input for our previous model. Thes videos will then be merged together such that there is a good transition between the different style images that are extracted from the input style vido.

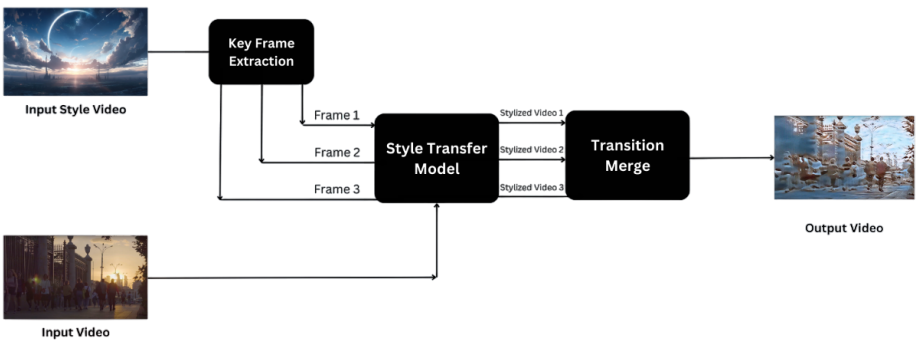

Figure 3: Flowchart of Overall Approach

Technology Stack and Dataset

For the implementation of our project, we will employ Python in conjunction with Keras and TensorFlow. Additionally, we will leverage the WikiArt Dataset to obtain style images that will be used

as input for our model. To train the model effectively, we will utilize extensive image datasets such as the COCO dataset or the ImageNet dataset.

5 Implementation

5.1 Baseline Neural Style Transfer Model:

1. **Creation of Image to Image Style Transfer Model:**

   After thoroughly reviewing the research papers on the subject of style transfer, we understood the methodologies proposed in each study. Based on the techniques discussed in these papers, we implemented the following model for image-to-image style transfer based on the paper by Gatys et al:

   In this model, we use a backbone of VGG-19 CNN model which is used for feature extraction. We generate these features for the input image and the style image and after that, we try to minimize the sum of two loss functions:

   (a) Style Loss: This loss takes into account the difference in style in the style image and the content image and reducing this ensures that the output image takes in the corresponding style of the give style image.

   (b) Content Loss: This loss takes into account the difference in the actual input image and the output image. Reducing this loss ensures that the output image has similar content to the input image.

   For every pair of input image and style image, the model reconstructs a output image that will minimize the sum of the above two losses by solving an optimization problem. The output image that minimizes these losses is our final stylized image.

2. **Extending the Image to Image style transfer model to Image to Video style transfer:**

   After completion of the Image-to-Image style transfer model, we tried using the same model to apply the style of an image to the entire video. We split up the video into different frames and applied the same model to each of the frames of the video. After the output frame of each of these input frames is computed, we merge these individual frames together to get an output stylized video.

3. **Model Results:**

   Output Image:

   The diagram below shows the output of our model wherein a grid plot is constructed, showcasing the application of various styles represented in the columns onto different input images displayed in rows. The resulting grid exhibits a series of transformations where the original source style images from the columns are imposed upon the target images in the row, illustrating the diverse combinations and effects achieved through the neural style-transfer process.

   Output Video:

   In the image below, a grid plot is constructed, showcasing the application of various styles represented in the rows onto a video, displayed as 3 frames of the video in columns. The resulting grid exhibits a series of transformations where the original source style images from the rows are imposed upon the target video frames in the columns, illustrating the diverse combinations and effects achieved through the neural style-transfer process.

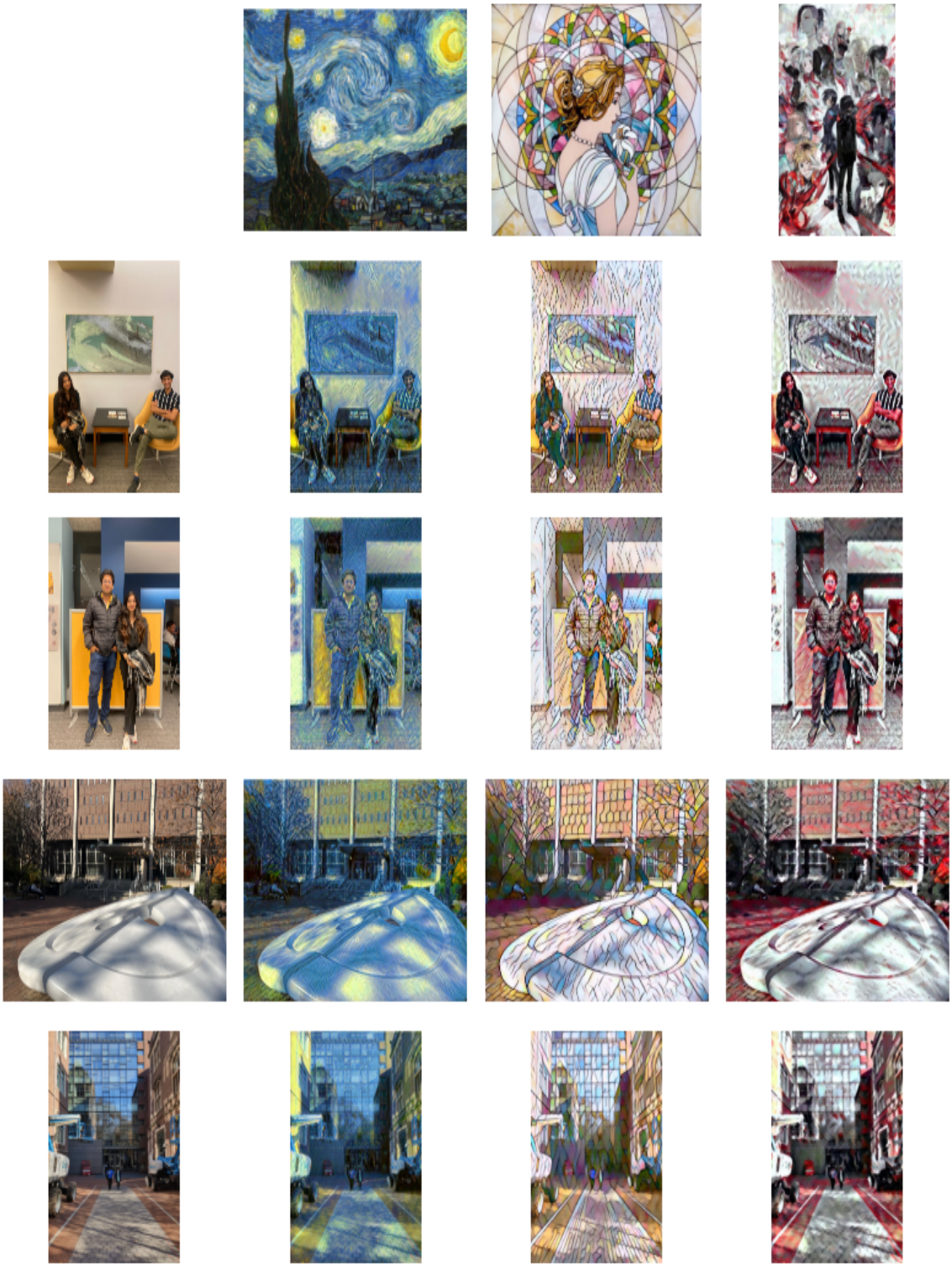

Figure 4: Model outputs for Image Input

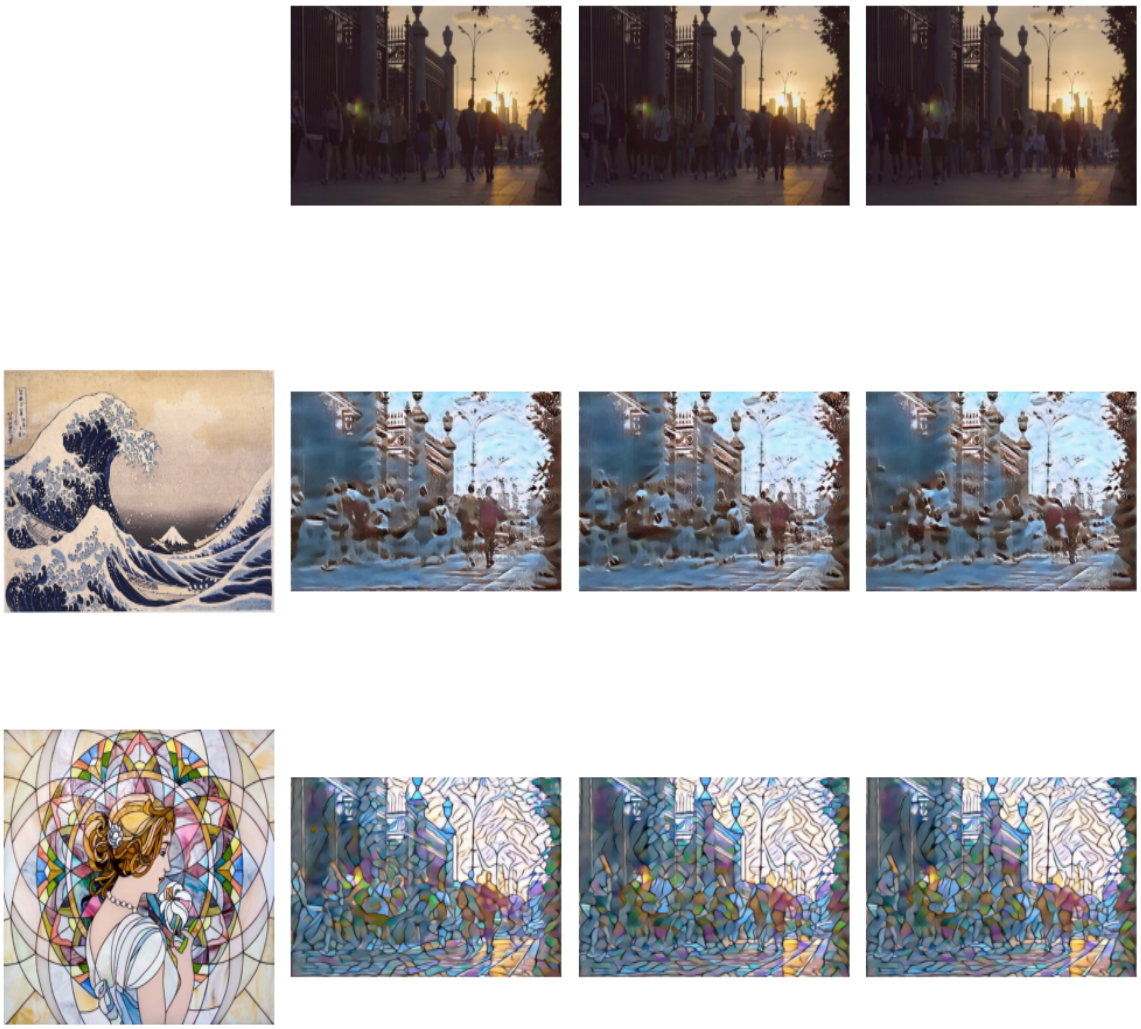

Figure 5: Model outputs for different frames of the video

4. **Drawbacks:**

   There were two significant drawbacks of using this model. Firstly, the model does not work well for applying a style to videos as it has to solve an optimization problem to compute the stylized image for each frame which is very slow. Secondly, there is a lot of flickering between different frames of the video as objects move around in the video.

5.2 Segmentation based approach

During our analysis, we identified flickering in the video when applying styles to individual frames. Notably, given the movement or changes in the foreground or background across frames, applying a consistent style to the same object proved challenging. To address this, we proposed a solution involving segmenting the foreground and background of each frame. By applying styles separately to these segments before combining them, our goal was to ensure a consistent style across objects throughout the frames.

Methodologies Implemented:

1. **Segmentation with Blurred Edges:**
    - Employed the GrabCut algorithm to segment foreground and background.
    - Blurred edges for smoother segments and applied style individually, combining them using masking.
    - **Observations:** Noticed distinct edges around objects due to overlap of styles on both foreground and background.
2. **Segmentation with Soft Edges:**
    - Modified GrabCut to avoid style overlap by utilizing sharper edge boundaries.
    - **Observations:** Reduced style overlap but encountered clear distinctions between foreground and background objects, lacking seamless style blending.
3. **Segmentation with Re-sizing:**
    - Addressed halo effects by resizing the foreground to eliminate blackened backgrounds.
    - Applied style to the resized version and combined both using masking.
    - **Observation:** Slight improvement observed, yet inconsistencies persisted between foreground and background frames. We had to think of other ways to apply style to objects consistently. One of the natural ways is to apply style to the whole image with the background blurred and the foreground blurred, and then combining both images to achieve consistency.
4. **Segmentation with Gaussian Blur:**
    - Generated masks for foreground and background and applied Gaussian blur on frames.
    - Created foreground with a blurred background and vice versa.
    - Applied style to each image and combined using the masks to generate the stylized frame.
    - **Observation:** Produced better results but did not entirely resolve the issues.

Application of Segmentation with Gaussian Blur in Image-to-Video Style Transfer:

- Achieved consistency in style across foreground and background objects in video frames.
- Observed a lingering halo effect around objects and noticed persistent flickering in the background of the images.

Results:

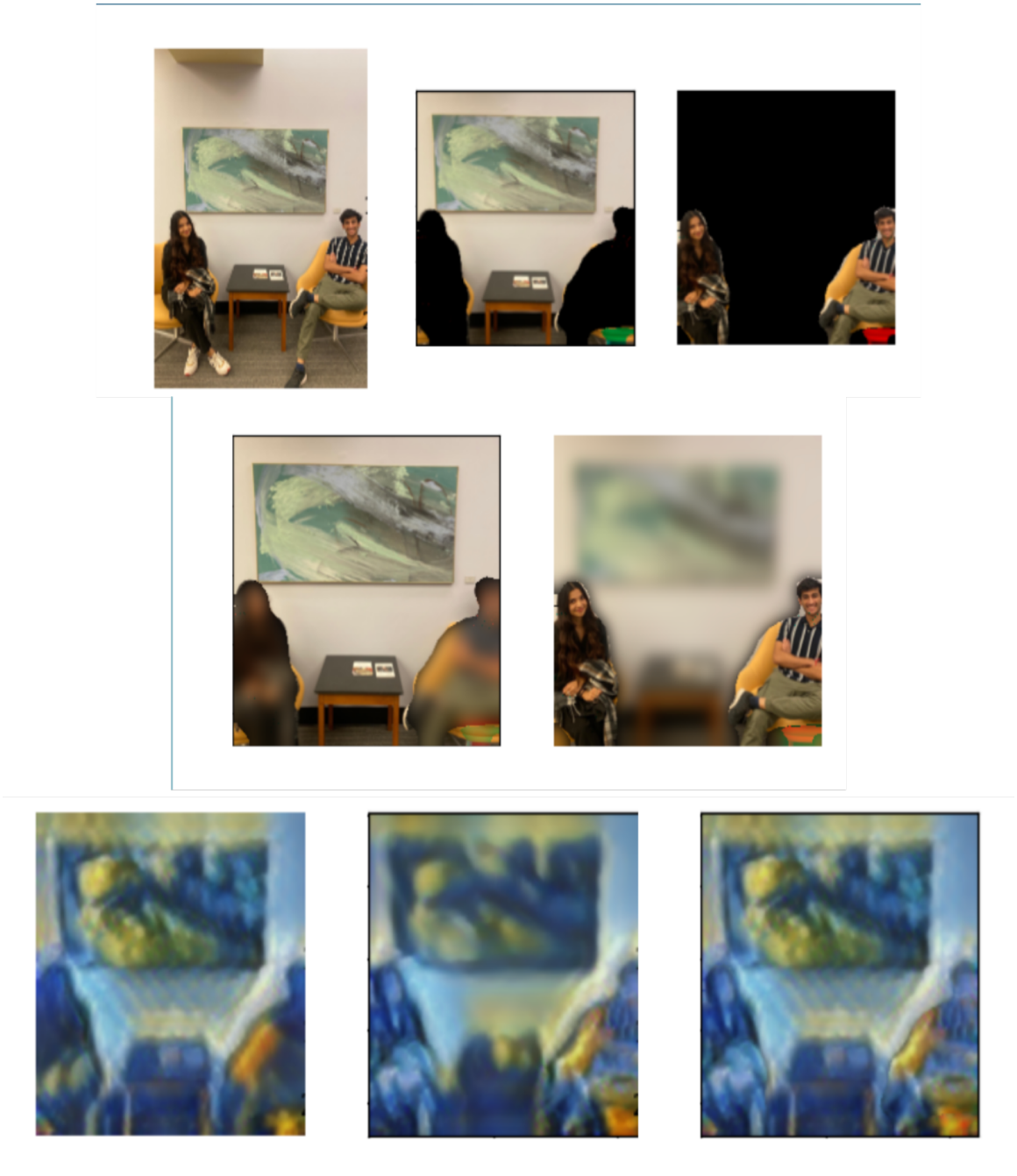

Figure 6: Styles applied separately to images after segmenting out the foreground and background

Figure 7: Segmentation Stylization applied to video gives more consistent styles on the person as visible in the image

5.3 Arbitrary Image Stylization Model

Having encountered limitations with the segmentation approach, we shifted focus towards addressing the temporal consistency of styles applied to objects across video frames using a Neural Network. However, the methodology discussed earlier, particularly in 5.1, demonstrated a significant drawback—it necessitated training the neural network for each frame, rendering the process exceedingly slow. Consequently, it became evident that this approach was not ideal for image-to-video stylization. Thus, we transitioned to utilizing the Arbitrary Image Stylization model, designed to train the neural network only once, thereby promising significantly improved speed and efficiency.

Model Architecture Overview:

The adopted model leverages a style transfer network structured as a conventional encoder/decoder architecture, with a specialization in normalizing parameters specific to individual painting styles. This approach, termed conditional instance normalization, involves normalizing each unit's activation using the formula:

$$\tilde{z} = \gamma_s \frac{z - \mu}{\sigma} + \beta_s$$

Here, μ and σ represent the mean and standard deviation across spatial axes in an activation map. γ_s and β_s form a linear transformation that defines the learned mean (β_s) and standard deviation (γ_s) unique to each painting style s.

Traditionally, "N-Style Networks" were constrained to work solely with explicitly trained styles. However, our objective extends beyond this limitation by aiming to enable stylizations for previously unseen painting styles. This extension is pivotal, as the network's ability to generalize to unfamiliar painting styles signifies the breadth and diversity of the network's representation of various styles. Particularly for our application, establishing a generalized network to extract style information is crucial, as it facilitates the extraction and transitional application of different styles from a source image onto the input video.

We coded the architecture from scratch to train it on the COCO dataset. However, during training, we realized it would take a lot of computational resources and time and we went ahead with pre-trained models. This model architecture relies on a pre-trained Inception-v3 architecture to compute the mean across each activation channel of the Mixed-6e layer, resulting in a 768-dimensional feature vector. Subsequently, two fully connected layers are applied to generate the final embedding $\tilde{S}$.

5.4 Video to Video Style Transfer:

1. **Style Extraction from image:**
   (a) The initial step in video-to-video style transfer involves computing style frames derived from the given video.
   (b) This process begins by extracting frames from the video content.
   (c) The video is divided into five segments to extract a total of five distinct style frames.
   (d) Within each section, the set of middle frames is collected and designated as the output set of frames for subsequent sections.
   (e) These selected frames are then utilized in the following stages of the style transfer process.
2. **Style Keyframes to Video Transfer:**
   Upon extracting style keyframes from the source video, the subsequent challenge was to apply these style transfers sequentially in a transition manner while ensuring a smooth transition

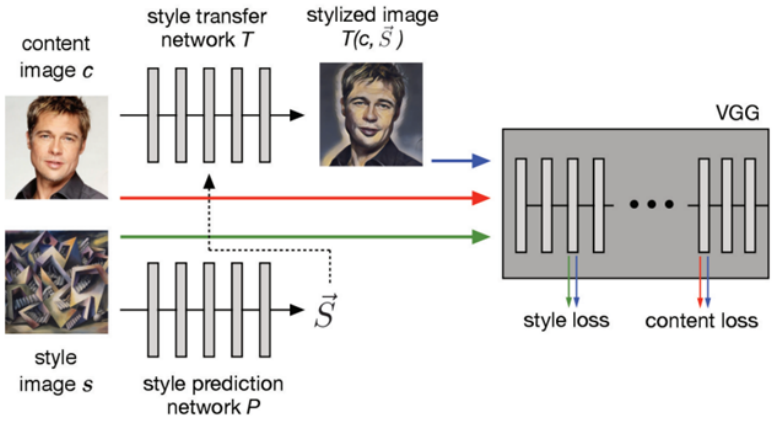

Figure 8: The style prediction network P(·) predicts an embedding vector S from an input style image, which supplies a set of normalization constants for the style transfer network. The style transfer network transforms the photograph into a stylized representation. The content and style losses are derived from the distance in representational space of the VGG image classification network.

between styles with minimal flickering. To achieve this, our approach focused on maintaining temporal consistency throughout the transitioning of styles.

One method employed to ensure temporal consistency in transitioning styles involved applying consecutive styles to each frame collectively and dynamically adjusting the degree of each style applied over time. Initially, at the video's onset, 100% of style keyframe 1 was applied to a frame. However, as the video progressed, the intensity of style 1 was gradually diminished while proportionally amplifying the influence of the subsequent keyframe's style. This transition strategy involved segmenting the input video based on the number of styles to be applied and calculating the number of frames within each segment. Within these segments, the styles were blended using the aforementioned dynamic mixing ratio.

For the application of each style to every frame, we relied on the previously implemented Arbitrary Image Stylization model discussed in section 5.2. This model served as a crucial component in achieving transitional style transfers between the source video and the input video, culminating in the generation of a stylized target video.

This approach allowed us to systematically transition between different styles extracted from the source video, ensuring a visually cohesive and aesthetically pleasing evolution of styles across the output video frames.

Results:

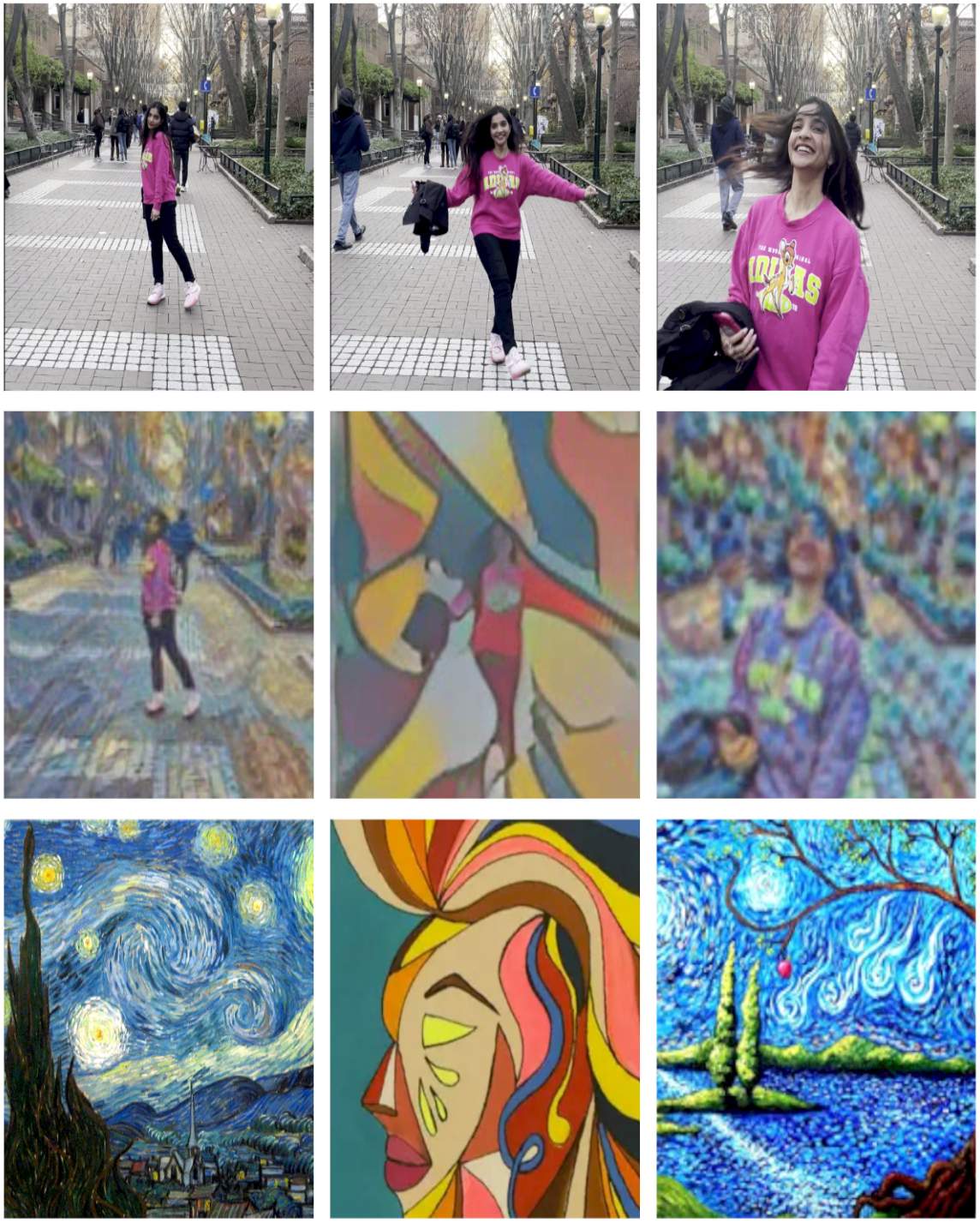

Figure 9: Applying multiple styles using multiple style images.

Figure 10: Grid of Keyframes Extracted from the source Video

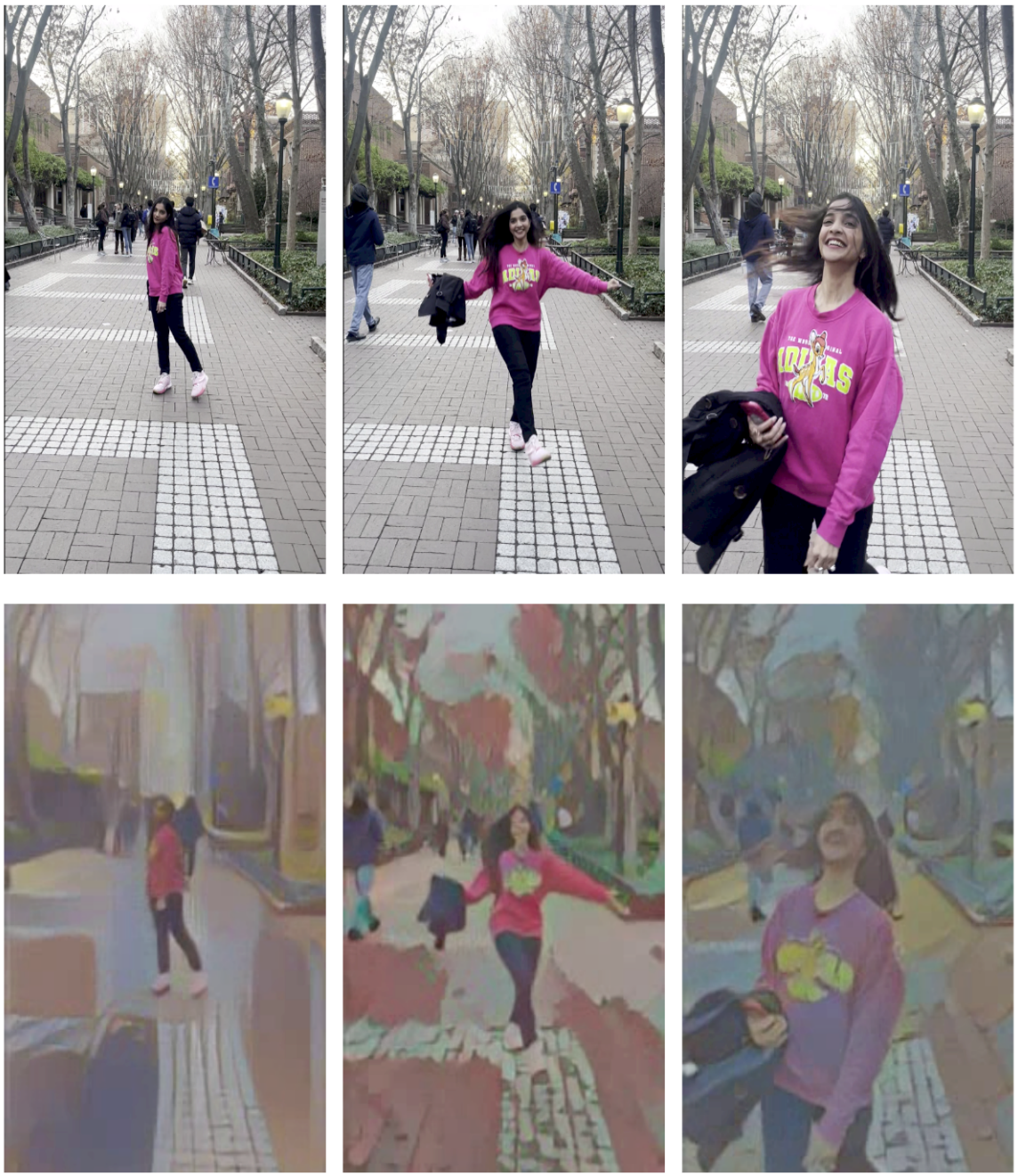

Figure 11: Final Output after applying the styles from the video

6 Qualitative Analysis

For the Baseline Neural Style Transfer Model (which was based on the model by Gatys et al.), we found that the major drawback in terms of the visual quality of the output videos was flickering. There was very little temporal consistency between frames as the objects moved around and this caused quite a lot of flickering in the output videos. Generating output videos was also a very time consuming process as we have to reconstruct the output image for every pair of style and input image.

To address this, our next model, the Segmentation based approach applied styles separately to the segmented foreground and background before combining them. Through this approach, we were able to achieve consistency in style across foreground and background objects in video frames. However, one persistent issue that we found was a lingering halo effect around foreground objects and because we blurred the frames before applying the styles, the flickering in the background had reduced slightly but it was still a noticeable artifact in the output videos.

To tackle these limitations with the Gatys based models, we decided to focus on addressing the temporal consistency of styles applied to objects across video frames using a Neural Network. For this, we implemented the Arbitrary Image Stylization model. We found 2 major advantages of this model over the others - Firstly, the different styles that were extracted from the input style video transitioned seamlessly across the frames in the output video. This sensured a visually aesthetic and consistent progression of styles throughout the frames of the output video. Secondly, this approach was significantly faster that the Gatys model, approximately **5x** faster. This was a great improvement from the previous models.

7 Challenges Overcome

1. **Image Resizing:** The input and style images that we used in our model were all of different sizes. We needed them to be of the same size to run our model correctly and also needed to get the output image in the same size as the input to see the actual results. To do this, we compressed all input and style images to 512x512 and then passed them to the model. After computing the output image in this dimension, we then re-scaled the output image back to the original size of the input image.

2. **Video style transfer too slow:** A notable challenge we faced during the implementation of image to video style transfer was that Neural Style Transfer was extremely slow as it retrained the model everytime we passed an input image to it. For a 5-10 second video, our model was taking around 10 minutes which is very slow. The slow execution speed posed a hindrance to achieving real-time or efficient transformations. To address this challenge, we used the Arbitrary Image Stylization model which only trains the model once and uses this trained model on further input images. This enhanced our speed to about 1 minute for a 5-10 second video, which was a significant reduction in time taken.

3. **Video Flickering:** In the Neural style transfer model, when we pass frames of a video to it, there are some frames wherein there is just noise because of how the model processes them and this causes the output video to flicker. We tackled this challenge with Arbitrary Image Stylization model, wherein we applied 2 styles at the same time, transitioning across frames. This leads to the flickering getting hidden in the transition of the styles and causes the output video to have lesser noise and flickering than before.

4. **Multiple Styles Changing Across Frames in Video Style transfer:** When we are trying to extract style from a video, there can be changes in style across the various frames of the video. We handled this in the AIS model by extracting more than one style from the input

style video and having multiple styles changing in the output video corresponding to the input style video that we have.

8 Conclusion & Future Work

Following our experimentation, we discovered that the Arbitrary Image Stylization model excels in video-to-video style transfer. It produces consistent outputs across various frames, markedly reducing flickering. Moreover, as we train the model to accommodate diverse styles, it remarkably accelerates the generation of output videos from input ones.

The following are ideas that we couldn't implement this time due to time constraints, but we plan on incorporating them in the future as they would bring about better and more optimized performance and quality of outputs:

1. **Finding Layer with maximum variation in the neural network and normalizing it:**
    - In the future, we plan to implement more sophisticated algorithms to automatically identify layers with maximum variation. This could include leveraging attention mechanisms, reinforcement learning, or evolutionary algorithms to optimize the selection process.
    - Once identified, normalizing these layers could help enhance the overall visual quality and consistency of the stylized video as it would lessen flickering because we would have normalized the layer which has the maximum changes happening.
2. **Using optical flow to improve temporal consistency:**
    - Optical flow is a technique used in computer vision to track the motion of objects between frames in a video.
    - Applying optical flow to style transfer will help maintain temporal consistency, ensuring that the style smoothly transitions from one frame to the next.
    - It can be used to align features between frames, reducing flickering or abrupt changes in the stylized video.
3. **Inpainting Foreground / Background:**
    - Inpainting involves filling in missing or corrupted parts of an image.
    - Separating the foreground and background in videos and inpainting each separately can enhance the overall visual quality and coherency of the stylized video.
    - This approach could be particularly useful in scenarios where the style transfer might affect the foreground and background differently.
4. **Key Frame extraction using Mean Squared Error:**
    - Key frame extraction involves identifying frames that capture the essence of a video or represent significant changes.
    - Using Mean Squared Error (MSE) as a metric for key frame extraction can help identify frames with the most visual difference from their neighboring frames.
    - These key frames can serve as reference points for the stylization process.

References

[1] Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. CoRR abs/1508.06576 (2015), http://arxiv.org/abs/1508.06576

[2] Ruder, M., Dosovitskiy, A., Brox, T. (2016). Artistic Style Transfer for Videos. In: Rosenhahn, B., Andres, B. (eds) Pattern Recognition. GCPR 2016. Lecture Notes in Computer Science(), vol 9796. Springer, Cham. https://doi.org/10.1007/978-3-319-45886-1_3

[3] Chen, D., Liao, J., Yuan, L., Yu, N., Hua, G.: Coherent online video style transfer. CoRR abs/1703.09211 (2017)

[4] Lu, Haofei, and Zhizhong Wang. "Universal video style transfer via crystallization, separation, and blending." Proceedings of the International Joint Conferences on Artificial Intelligence Organization (IJCAI), Vienna, Austria. 2022.

[5] Wu, Zijie, et al. "CCPL: contrastive coherence preserving loss for versatile style transfer." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.

[6] Kolkin, Nicholas, Jason Salavon, and Gregory Shakhnarovich. "Style transfer by relaxed optimal transport and self-similarity." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[7] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer International Publishing, 2016.

[8] Liu, Xiao-Chang, et al. "Depth-aware neural style transfer." Proceedings of the symposium on non-photorealistic animation and rendering. 2017.

[9] Ghiasi, Golnaz, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. "Exploring the structure of a real-time, arbitrary neural artistic stylization network." arXiv preprint arXiv:1705.06830 (2017).