Painting Process Predict

1.Background

Outstanding painter and artist are sharing their drawing process on internet. There are thousands of painting style and techniques when creating a same scene or object by different artist. Although build one's own art style and drawing habit is important, many new beginners would start their career by imitating famous artists' drawing path and techniques. What if those artists didn't share any clues of their painting or illustration? And how we can predict the reverse process if by just giving a single finished image? By giving a certain finished image, how will another artist recreate this work? If we can train a model which can predict the problems mentions above, it will be helpful for those beginners who wants to learn the time lapse of an illustration and explore more potential possibility of the whole process.

Predecessors already define this as a "new video synthesis problem", that is "given a painting, can we synthesize a time lapse video depicting how an artist might have painted it", which came up by Amy Zhao Et al at [1]. They raised several challenges when dealing with the similar task, like different artists will paint the objects in different order even they are under a same scene; some Non-paint effects such as local blurring, smudging will be added during the digital painting process while during the physical painting such as watercolor paintings, artists sometimes use the pigment diffusion effect to represent the light and shadow. For addressing those problems, they design a model which use the blank image and the last frame of the videos (which mean the single original finished image as the input during inference) as the initial input, relying on the previous frame generate by the generator and the last frame to generate all the frames sequentially. The model uses a latent vector which sampling from a distribution that is close to the standard normal, then fit it with the concatenates of the previous frame and the last frame to encourage the model generate the whole process of a finished image. During training, they train an encoder to restrict the data obey normal distribution. They also use the U-Net structure to improve the performance. They get a quite good result and the model can succeed in generating more realism process of the process of an art painting.

If we define the predicting process problems as a video synthesis problem plus a single image synthesis problem, as the during the generating process, the model not only need to generate a more realistic single frame, but also need to consider of the continuous factor such as the light flow and the natural feeling between frame by frame. The model

required the capacity of “capture” the track of a certain artist. Base on the above discussion, there are many advanced designing we can refer and combine. Like Vid2Vid[2], which come up by Ting-Chun Wang Et al at. They define a video synthesis problem — from a video convert to another video, encourage the model generate high resolution and more realistic result. And SPADE[3] from Taesung Park Et al at, which provide a Multimodal method which the GAN model are hardly output. We refer to predecessors’ advanced design (more details see below) to try to improve the recent problems and see if it can address some shortage of Amy Zhao Et al at work.

2.Related work

To see the method better to review the problems and some predecessors’ solutions.

2.1 Painting process dataset challenge

Amy Zhao Et al at raised the challenges of the Painting process prediction task in [1]. Apart from the problem they raised, there are more challenge during the collection and pre-process stage of the dataset:

Move and Scale:
In digital painting, many artists not sure the position or the size of the main object, unlike the physical painting, it’s easy to move or rescale the object in the digital painting software nowadays. And this situation happened all the time as artists need to modify the whole perception of the painting so reach the best effect and proportion. If we extract each frame from the time lapse video, the positions sometimes will not match to previous frame, and this is not the main problem the model need to learn. The “rescale” operation generally happened in some parts of the whole object rather than overall rescale. It means sometimes the next stage will show up the not “subtle strokes process” but the dramatic changes between two frames or stages.

Non-uniform change
We collected over 100 digital painting process videos in early stages (we didn’t use all of them, see 4.1, extracted every 5 frames of the videos and total 300 frames of the videos. The result shows that the changes between each frame are not at uniform amount. The change between two frames or stages are more significant in the early time and gradually decline in the later frames. This situation is not about the painting rates, but the base on habit and common sense. Because in the early stages artists trend to draw the large

color block to “shape” the object, then add the details on it. Although for human being details are more difficult than contour, but the field of the change is actually decrease, this will cause the with the difference between the current state and the final state fewer and fewer, it’s harder and harder for the model to learn the slightly changes. If we extract more frames in the early stage but less frames in the later stage, this situation will slightly improve, but also means will loss the information in the later stage. In different videos, this non-uniform change situation shows different expression, it’s hard to set a criterion of when and how much to decrease the frames in different stages.

Base on the above problems, We manually restrict the object position and size in a maximum degree base on the last frame and limited the maximum frames no more than 300. But this method still cost a huge of time. There are many automatic image registration algorithms like Sift, Surf, Fourier-Mellin algorithm[4] can restrict the dislocation object but we didn’t use that in our experiment. Limited by time and the amount of our dataset, the manually restriction is faster and have a better performance in pre-process stage. The second problems can be solved in somewhat by using our auto degree labeling method, which can see at 3.3.

2.2 Generating image difference challenge

We used the difference between two frames as the one of the inputs to a GAN model, to see if the generator can generate the next frame. See Equation 1.

$$difference = abs(X_{T+1} - X_T) \quad \text{Equation 1}$$

X_{T+1} represents the next frame or stage while X_T represents the current frame or state.

The result is obvious good enough as the model only need to “plus” the difference at different channel. Then our target transfer to giving a certain finished image and the current state, can a GAN or a VAE generate a difference image? We trained a GAN and a VAE to generate the difference between two frames by giving the finished image and the current state as the input. The result shows that even the loss was at a very low value, but the result was still not good, the model even can’t learn anything from the input, see Figure 1.

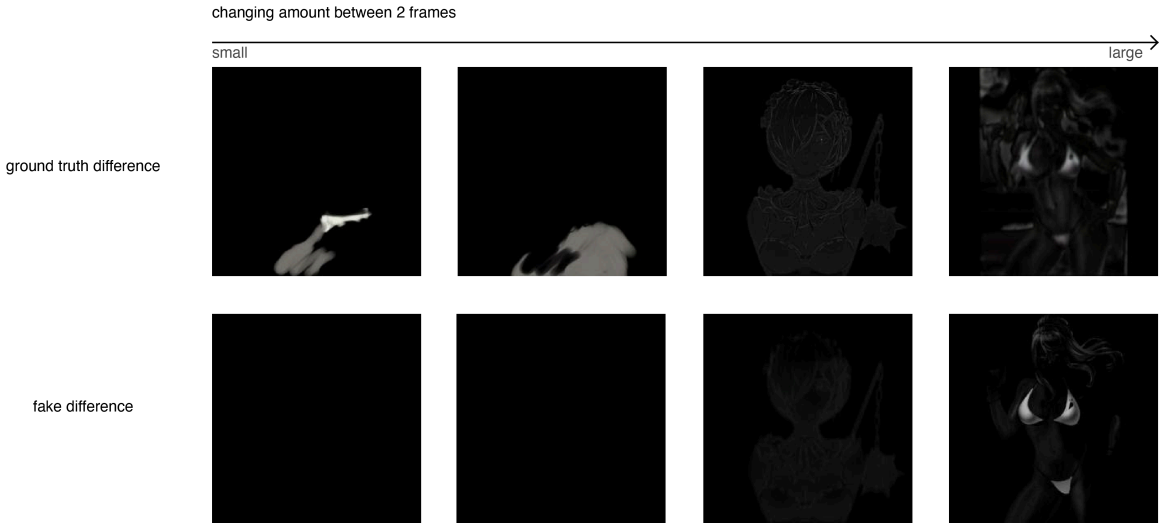

Figure 1

One reason could be the ground truth difference images itself are almost all 0, only some certain field has a feature value. This will cause the model trend to generate the all-zero image when using pixel wise difference loss. We can see with the improve of the changing amount between 2 frames, the model can generate a closer result as now the values are evenly distributed on each position. But it's still hard for the model to learn which information can be thrown away. We can regard the target of the model is to learn how to "get rid of some information". Obviously in the current structure didn't capacity of this.

We finally give up the idea of encouraging model to generate the difference directly although the difference is match to human intuition, but maybe not the best understanding for the machine. We keep the difference image as an input to the VAE also but not verify the loss directly. Next step we will introduce our model structure and will have a further explanation.

3. Method

We first extracted no more than 300 frames of each painting process videos $\{x_1, x_2, \dots x_T\}$. Our model task is giving a finished painting x_T to generate the past frames $\{X_1, X_2 \dots X_{T-1}\}$, which should match the ground truth frames $\{x_1, x_2, \dots x_T\}$.

3.1 Model

Our model is similar with [1], [2], [3], [5]. Base on this structure, we modify the condition branch but also introduced a condition labeling which w.r.t degree input for the CVAE structure, and also used a discriminator to improve the realistic of the result. The model is

with the structure of CVAE + GAN.

During training, we used an encoder took the { $x_{difference}, x_T, x_t, x_{t+1}, x_{label}, x_{degree}$ } as input, generate a latent vector Z to the decoder. Beside the random vector Z, decoder also takes the { $x_T, x_t, x_{label}, x_{degree}$ } as the input.

During inference, we abandon the encoder but sampling vector Z from normal distribution. The x_{label} represents the semantic segmentation label map of the finished image, and the x_{degree} represents the condition label of the amount of difference, which will have a further discuss at 3.5.

Our encoder uses the similar down-sampler like [3] but without any modification at weight and bias. Different from the baseline, the decoder doesn't have the U-net structure, but similar with the pix2pixHD's generator[5]. The condition branch will be passed though the down sampler and concatenate with the reshape vector Z. Then the feature will pass by two up samplers respectively then output an image with 3 channels $x_{stroke}$and a matrix x_{weight} with 1 channel. These two output will base on the Equation 2 to combine the previous frame x_t as the fake next frame $\hat{x}_{t+1}$.

$$\hat{x}_{t+1} = x_{weight} * x_{stroke} + (1 - x_{weight}) * x_t \quad \text{Equation 2}$$

where the x_{weight}, x_{stroke} are the output of the decoder.

Our discriminator is the same with the patchGAN in pix2pix [6]. The model structure shows at Figure 2.

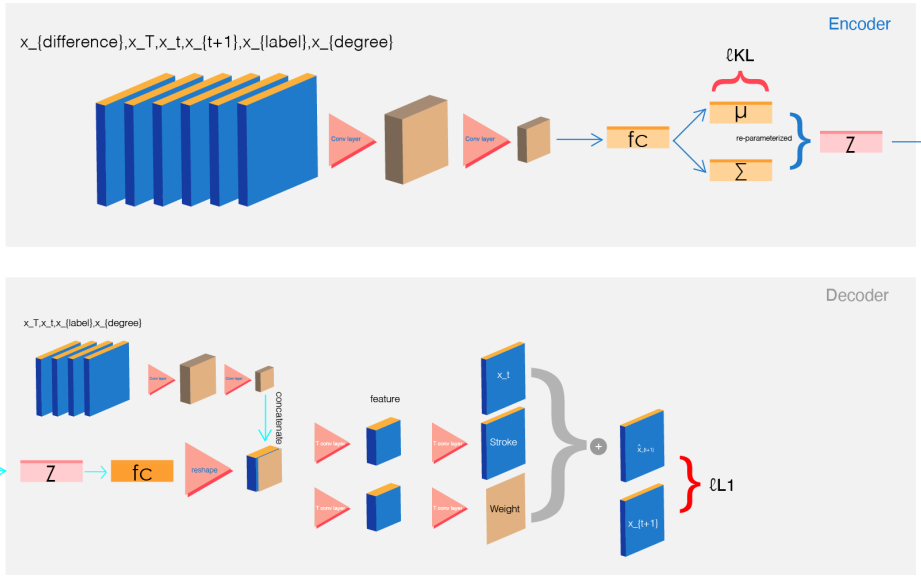

Figure 2

3.2 loss function

we use a similar loss function like most of the image generating task[7], [8], [9]. During training, we use KLD loss to measure the difference between the distribution of latent variables and the unit Gaussian distribution. We used L1loss to measure the pixel-wise difference between the ground truth image and synthesized image. We used same GAN loss to improve the realistic of the image. Beside that We also used the TV loss to decrease the noise during synthesis. Perceptual loss is commonly used to improve the result, we used the same Vgg16 perceptual loss structure from pix2pixHD.

3.3 Auto degree labeling function

It's common to used condition branch for the VAE to achieve controllable effect. Base on the condition branch from [1], we add an extra condition one-hot label which w.r.t the amount of changing between each frame.

At first from our observation we found that if we give a one hot label of the information of which part will change in the next frame, the model can have a better understanding which parts can focus on. The changing part is bases on the semantic segmentation map of the finished image. The reason why we used the semantic segmentation map as the one of our label is now a days a semantic segmentation map synthesis model is common and there are been widely used in object detection like [10], [11], so during test we can use another model to generate a semantic segmentation map and make it one-hot format. We manually labeled the one-hot vector by judging every time which part will change in the next frames, see the Figure 3.

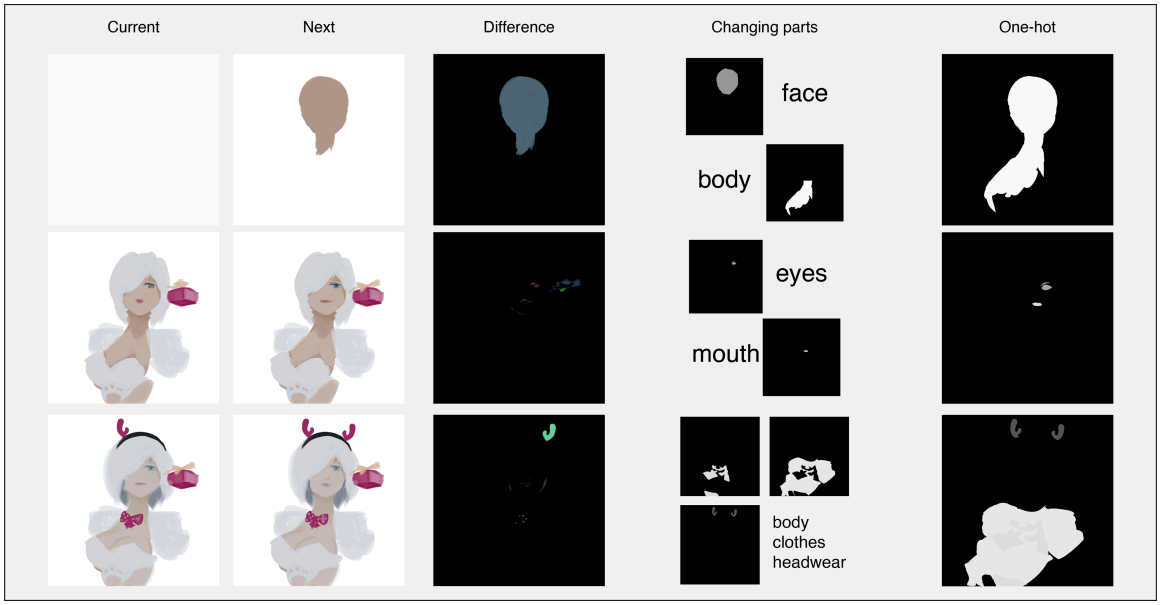

Figure 3

By doing this, we can modify the condition during test. If we randomly pick different parts of the semantic segmentation and regroup them as a new input, we can get a different result from the original dataset. See the result at Figure 4.

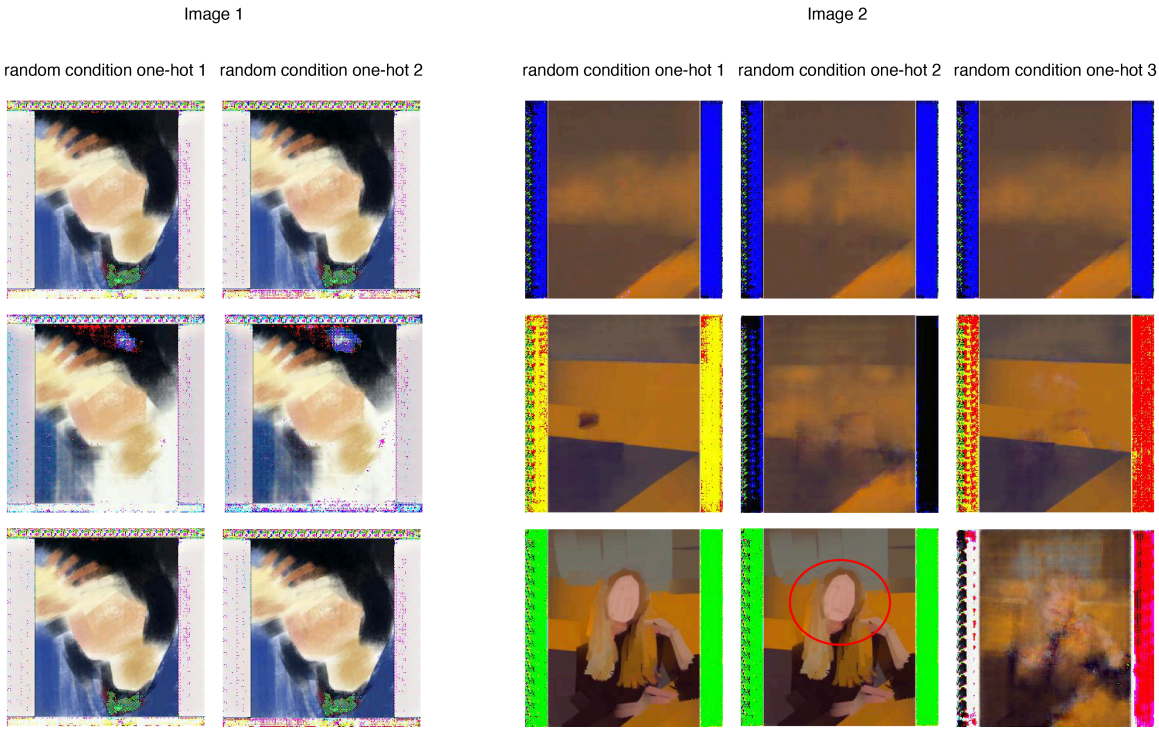

Figure 4

Later we realized that it's a huge time consuming work to label every pair frame of the all the dataset. We first try an auto function depend on the difference image's index to

determine which parts of the semantic segmentation will be regrouped. Like the challenge mentioned above, many artists will “reshape” or “rescale” their object during painting, which mean some of the changing position will overlapping each other. If a difference happened at the edge or the junction of two or more parts, only depend on which index changed are not accurate enough. Because we based on the changing parts to make the condition branch, it’s easy for human to judge which parts the artist is painting at, but it’s unavoidable happened overlapping. This situation will have a greatly improved if the painting dataset which are use the wireframes to make the sketch or flat coating, i.e., [12]. The explanation at Figure 5.

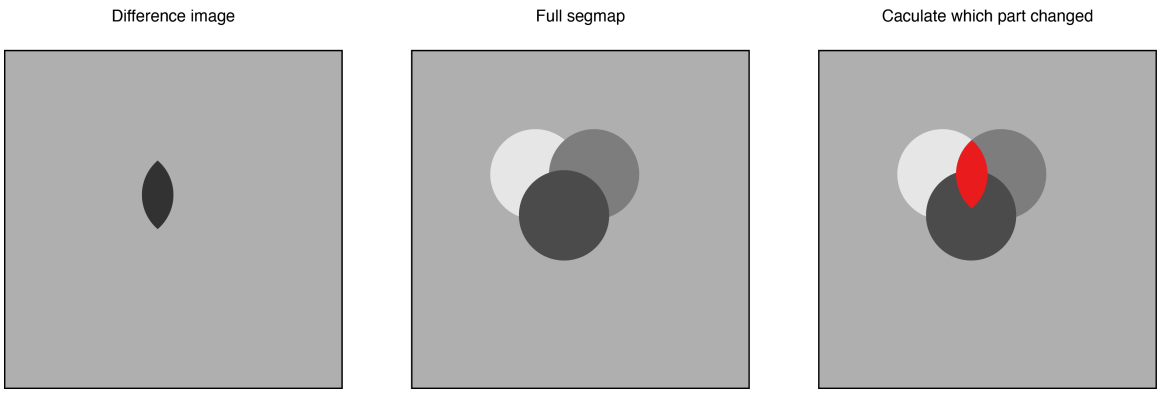

Figure 5

YipinZhou Et al ’s model[13] has a similar task, giving a object image to synthesize the past or future physical state of this object. They at used a condition branch base on “degree”, which means with the by giving a certain value, the model will generate the correspond to “how far” from the current state. Inspired by this, we found a faster and more accurate to make the condition one-hot label. We based on how much “contribution” of each part to the difference image to make the one-hot degree condition branch. With respect to the “degree”, this condition branch provides a more meticulous granularity and also can be calculate by algorithm. See the function at Figure 6.

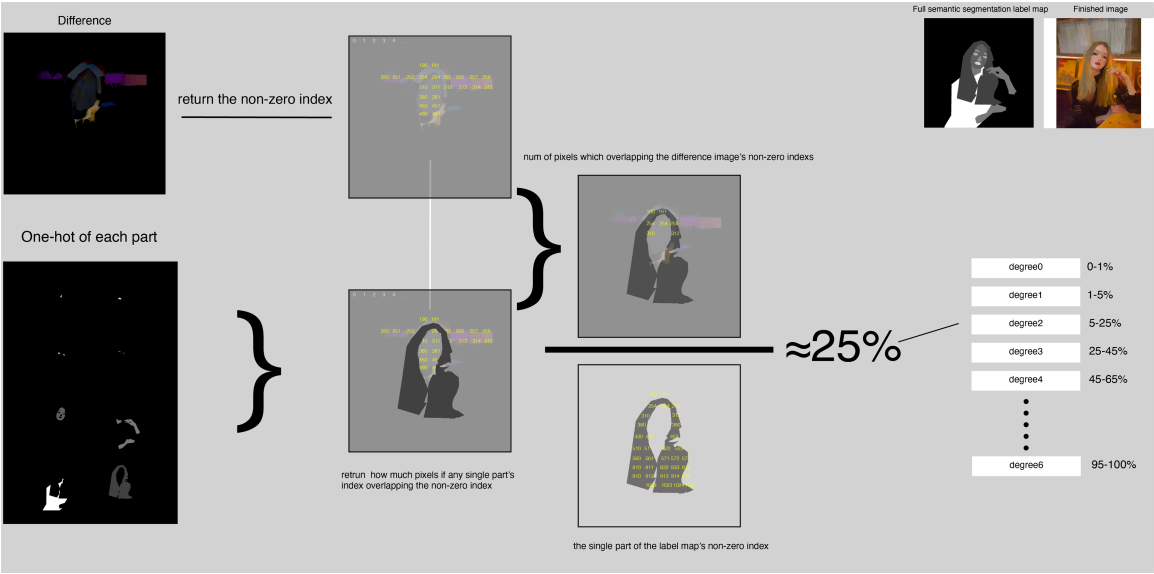

Figure 6

3.4 optimization

Followed by [1], [13], we used both pairwise optimization and sequence optimization. During pairwise optimization, we extract frames from the original video dataset in pairs, which can see at 4.1.

According to Equation 3 and Equation 4:

$$\mathcal{L}_{\mu,\sigma^2} = \frac{1}{2}\sum_{i=1}^{d}\mu_{(i)}^2 + \frac{1}{2}\sum_{i=1}^{\mathrm{d}}\left(\sigma_{(i)}^2 - \log\sigma_{(i)}^2 - 1\right) \text{Equation 3}$$

$$\mathcal{L}_{KL}(N(\mu,\sigma\text{^}2)||N(0,1) = \frac{1}{2}(\log\sigma^2 + \mu^2 + \sigma^2 - 1) \text{Equation 4}$$

The target of this step is to minimize the $\mathcal{L}_{KL} + \mathcal{L}_{L1(\hat{x}_{t+1},x_{t+1})} + \mathcal{L}_{Vgg(x_{t+1},x_{t+1})} + \mathcal{L}_{TV}$, we take the $\hat{x}_{t+1}$ to be the next "fake" x_{t+1}, See the details at Figure 7.

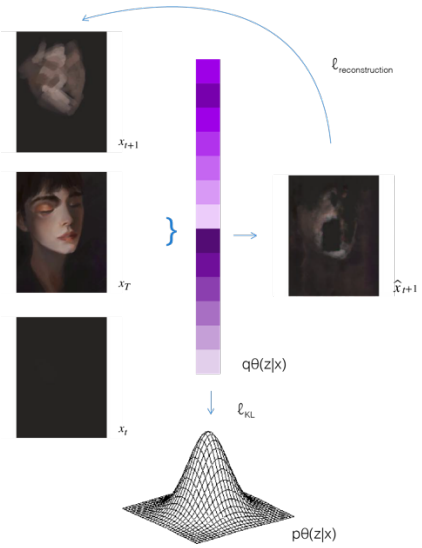

Figure 7

We also used the sequence optimization, this process is similar with the pairwise optimization, while we take the $\hat{x}_{t+1}$ to be the next "fake" x_{t+1}, which mean we encourage model to keep generating the fake x_{t+N} which will be the next input data of itself. At the same time, the latent vector will sampling from $p(z)\sim N(0,1)$. See details at Figure 8.

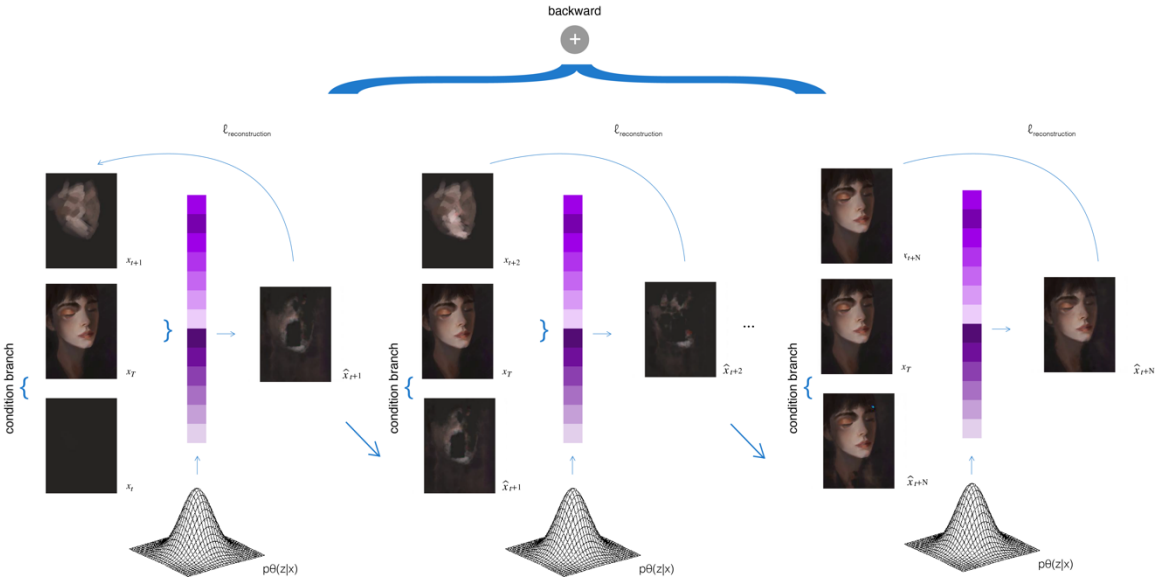

Figure 8

Limited by time and dataset We didn't let the model to generate the $\hat{x}_1$ by giving the blank image as input. It's better to use an extra model to generate the first frame which is a more effective way like [2].

3.5 Inference

During inference, we remove the encoder and sampling latent z from $p(z)\sim N(0,1)$, and use the random condition label degree to synthesize the frames. There are also two model remain to be trained. One is mention in section3.5, a model which synthesizes the first frame by giving a semantic segmentation map which include the information of the background and the edge of the paintings. Another model will synthesize the semantic segmentation map which will provide to the first frame synthesize model and our video synthesize model by giving the single painting. Limited by time and the amount of the dataset, we didn't train these two model. The full flow of the inference at Figure 9.

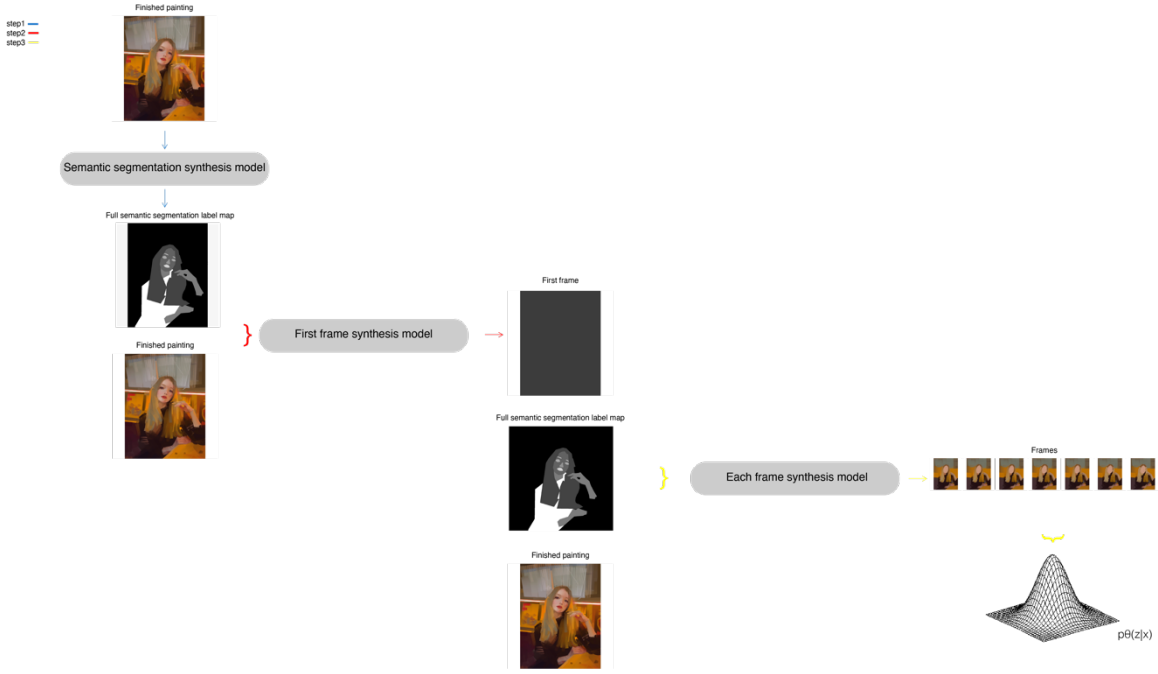

Figure 9

4.Experiment

4.1Dataset

We collected the painting time lapse videos from web art forum or video platform like Youtube or Bilibili where users share their painting process. For copyright we won't release our dataset. The style of the painting we used was digital and water color, which mean those painting were from digital painting software and without wireframe or sketch but with large or small stroke with color blocks while being painted.

Frist, we extract no more than 300 frames from each video in total. The reason why we do that was there were a lot static frames in the original videos. We want to make sure the frames we extract have an obvious change between each other.

Then we sampled frames in these 300 frames for each video in different intervals, i.e., we extracted frames every i.e., {1,3,5} interval, and made them into pairs and sequences. See Figure 10.

We used portrait painting for our dataset as we can make a semantic segmentation map obviously. We made our semantic segmentation map by Photoshop (The similar semantic segmentation generating task can be easier access by good trained model in the future). Our semantic class are {edge[1] ,background, hair, clothes, eyebrows, eyes, nose, mouth, face, body}.

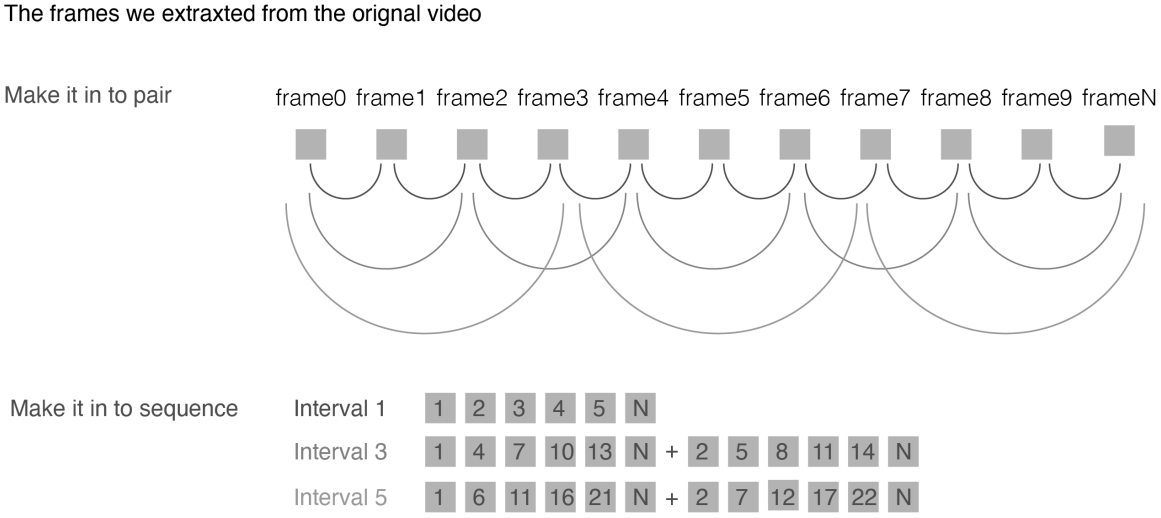

Figure 10

The reasons we need to make the dataset into different interval are:

1. If the original videos amount is not enough, it's better to make the dataset into different granularity
2. If we want to use the random condition degree label during inference, it's better to provide as much as possible amount of different step base on the current status, which mean provide more possibility of changing base on the current status.

In theory, if the amount of the original videos is much enough, the way we mentioned above can be omit. But once it be provided a much enough possibility of different step, the

[1] In order to make sure all of our dataset at the same size, we mark those edge no matter they were from the original video or pre-process.

performance will be improved obviously. In [1], the random normal standard distribution take charge of this problems. But if we want the result is controllable base on our condition degree label, we recommend this method to finetune the model.

In our experiment, for saving time and limited by device, we only make our dataset in interval 5. The result is at 4.3.

We recommend not to over sampling too large interval to make the dataset. As we base on the percentage of the changing field of the single part semantic segmentation label map, if the interval is too large, the information will properly loss or conflict with other granularity. We recommend try to keep the condition degree label maps of all dataset to match a uniform Distribution. In other words, make sure all the label will evenly place at different degree.

4.2 Implementation

Our code borrows heavily from https://github.com/NVlabs/SPADE. We implement our code used Pytorch and open-cv to pre-process the dataset. Our L1 loss lambda was be set to 100 while the KLD loss lambda be set to 0.05 initially and improved 10% every 20 epochs. We trained our model in 600 epoch, while the first 400 epoch was pair-wise optimization. Our Vgg loss lambda is 10.

Limited by the memory of our device, during sequence -wise optimization, we split the sequences which were too long.

The code and other hyperparameter see our Github: https://github.com/waihinchan/scar

4.3 Result

Base on the limitation amount of our dataset, for the best explanation of our model. Our result was from training set but we will also provide some result outside of our dataset. We also provide a failed example to evidence the importance of the granularity in pre-process the data.

Our result Figure 11.shows the diversity of the results and also shows a different way of painting compare with the original process. There were some artifact noise on our result, like we mentioned above, we sample our dataset in interval 5, the meticulous of our granularity is not enough which mean during inference there were some information the dataset never show up before (another reason is lacking of the dataset cause the impoverished of the information w.r.t random vector z).

Figure 11

The failed result shows at Figure 12. We sampled from frames in interval {1,3,5,10,12}, among them the interval 3 and interval 10 and 12 cause a conflict. As the interval 5 is large enough w.r.t changing field, it turned up the phenomenon that same condition degree label but different result in interval 10 and 12. In other words the model trend to generate unpredictable result from the same input but different output.

Figure 12

The Figure 13 showed our condition degree label effect. We Can see by assign different parts to different degree, we can get the part changing what we expected for. The changing will also be affected by the current frame, in theory if the current state is very close to the last frame, even give a certain part into a very high degree, the generated image will not change obviously.

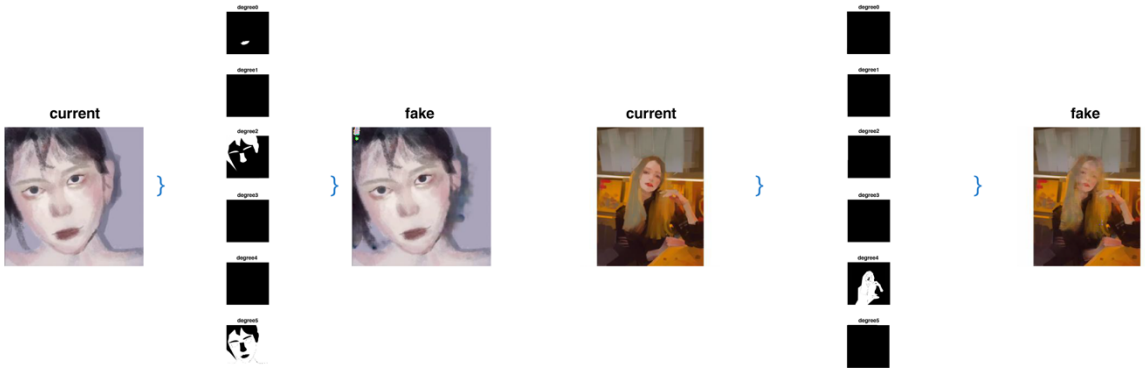

Figure 13

The Figure 14 showed the result of inference with a new data. The sequence collapsed very fast in the early stage and trend to generate a blur result.

Figure 14

5.Conclusion

Base on predecessor's work, we introduce another method to control the next status of generating in video synthesis task. Although our result showed a lot of artifact and our model still facing the overfitting problems, the result shows by modifying the condition label map we can generate something totally different from original painting process and shows the possibility of how machine will paint an art piece. We introduced a modular model and came up an modular inference system ideas when dealing with the similar problems.

Base on the experiments we made above, there are some key factor we want to place here.

1. It's important to cover as much as possible of the different status base on the current status when sampling dataset and keep the dataset without conflict at the same time.
2. If it's hard to make sure the condition degree label of the dataset is close to uniform

distribution, it's good to sampling from the distribution of the dataset we used to make the random label degree during inference.

6. Acknowledgments

[1] Zhao, Amy, G. Balakrishnan, Kathleen M. Lewis, F. Durand, J. Guttag and Adrian V. Dalca. "Painting Many Pasts: Synthesizing Time Lapse Videos of Paintings." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 8432-8442.

[2] Wang, T., Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, A. Tao, J. Kautz and Bryan Catanzaro. "Video-to-Video Synthesis." NeurIPS (2018).

[3] Park, T., Ming-Yu Liu, T. Wang and Jun-Yan Zhu. "Semantic Image Synthesis With Spatially-Adaptive Normalization." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 2332-2341.

[4] Álvaro Ordóñez, Francisco Argüello, Dora B. Heras. (2017) Fourier–Mellin registration of two hyperspectral images. International Journal of Remote Sensing 38:11, pages 3253-3273.

[5] Wang, T., Ming-Yu Liu, Jun-Yan Zhu, A. Tao, J. Kautz and Bryan Catanzaro. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 8798-8807.

[6] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 5967-5976.

[7] Kingma, Diederik P. and M. Welling. "Auto-Encoding Variational Bayes." CoRR abs/1312.6114 (2014): n. pag.

[8] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 5967-5976.

[9] Dosovitskiy, A. and T. Brox. "Generating Images with Perceptual Similarity Metrics based on Deep Networks." ArXiv abs/1602.02644 (2016): n. pag.

[10] Shelhamer, Evan, J. Long and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017): 640-651.

[11] Ronneberger, O., P. Fischer and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." MICCAI (2015).

[12] Wikipedia contributors, "Cel," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Cel&oldid=984468954 (accessed November 20, 2020).

[13] Zhou, Yipin and T. Berg. “Learning Temporal Transformations from Time-Lapse Videos.” ECCV (2016).