Video Style Transfer

1st Shubham Das
M.Tech CSIS - 2018202004
IIIT Hyderabad

2nd Rajneesh Singhatiya
M.Tech CSIS - 2018202018
IIIT Hyderabad

3rd Sayan Ghosh
M.Tech CSIS - 2018202002
IIIT Hyderabad

4th Sai Sukrutha Chamakoora
M.Tech CS - 2018201054
IIIT Hyderabad

***Abstract*—This project aims to explore various ways of transferring the style of an artistic image to a video clip, based on computer vision and machine learning.**

***Index Terms*—Video style transfer · Optical flow · Real-time processing.**

I. Introduction

Video Style Transfer has been a widely researched topic for a while. The Artistic Image Style Transfer by Gatys et al. was one of the first papers that brought style transfer into the notice of other researchers. Since then many methods have come up and it also inspired researchers to work on video style transfer because a video is made up of frames, which brought a lot of challenges for quality style transfer that gave rise to other techniques being used for the purpose.

II. Challenges

A. Maintaining color consistency

While transferring style of an image to all video frames it is often the case that colors of an object change drastically which may be because of the model being not suitable or the light intensity of that part of the object changed significantly between the frames.

B. Maintaining pattern consistency

Like colors while transferring style of an image to all video frames, patterns mimic-ed for a particular object in the video doesn't remain constant throughout the video. Sometimes the pattern of the object doesn't move along with the object properly.

C. Difference in resolution between style image and video

Reference image or the image from which the style is taken may have different resolution as compared to the video's resolution to be stylized.

III. Approaches taken

Till now 3 approaches have been tried which are discussed as follows:

A. Approach 1

The approach of artistic neural style transfer by Gatys et al. has been implemented here for frame wise processing.Here the aim is to reduce the loss in a typical deep learning method. the loss function we minimise during image synthesis contains two terms for content and style respectively, that are well separated .We can therefore smoothly regulate the emphasis on either reconstructing the content or the style. A strong emphasis on style will result in images that match the appearance of the artwork, effectively giving a textured version of it, but hardly show any of the photograph's content. When placing strong emphasis on content, one can clearly identify the photograph, but the style of the painting is not as well-matched. For a specific pair of source images one can adjust the trade-off between content and style to create visually appealing images. But the main challenge that remains is that a video is made up of many different frames for which different content and style weights may seem correct.We used the feature space provided by the 16 convolutional and 5 pooling layers of the 19 layer VGGNetwork.

To visualise the image information that is encoded at different layers of the hierarchy we perform gradient descent on a white noise image to find another image that matches the feature responses of the original image. So let $\vec{p}$ and $\vec{x}$ be the original image and the image that is generated and P^l and F^l their respective feature representation in layer l. We then define the squared-error loss between the two feature representations as:

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F^l_{ij} - P^l_{ij} \right)^2 .$$

On top of the CNN responses in each layer of the network we built a style representation that computes the correlations between the different filter responses, where the expectation is taken over the spatial extend of the input image. These feature correlations are given by the Gram matrix $G^l \epsilon R^{N_l \times N_l}$, where $G^l{}_{ij}$ is the inner product between the vectorised feature map i and j in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

To generate a texture that matches the style of a given image we use gradient descent from a white noise image to find another image that matches the style representation of the original image. This is done by minimising the mean-squared distance between the entries of the Gram matrix from the original image and the Gram matrix of the image to be generated. So let $\vec{a}$ and $\vec{x}$ be the original image and the image that is generated and A^l and G^l their respective style representations in layer l. The contribution of that layer to the total loss is then:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - A_{ij}^l\right)^2$$

and the total loss is:

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

where w^l are weighting factors of the contribution of each layer to the total loss.

B. Approach 2

This approach is based on ”Perceptual Losses for Real-Time Style Transfer” paper by Johnson et al.

Our implementation uses TensorFlow to train a fast style transfer network. We use roughly the same transformation network as described in Johnson, except that batch normalization is replaced with instance normalization. We use a loss function close to the one described in Gatys, using VGG19 instead of VGG16 and typically using ”shallower” layers than in Johnson’s implementation (e.g. we use relu$1_1 ratherthanrelu1_2$).

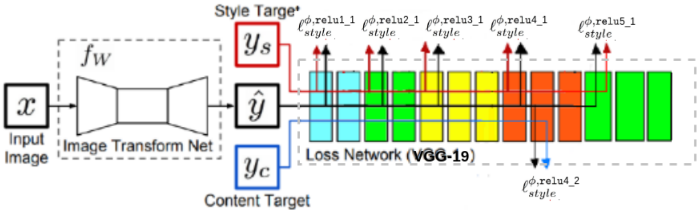
Style Targe'
$\ell_{style}^{\phi,relu1\_1}$
$\ell_{style}^{\phi,relu2\_1}$
$\ell_{style}^{\phi,relu3\_1}$
$\ell_{style}^{\phi,relu4\_1}$
$\ell_{style}^{\phi,relu5\_1}$
y_s
f_W
x
$\hat{y}$
Input Image
Image Transform Net
y_c
Loss Network (VGG-19)
$\ell_{style}^{\phi,relu4\_2}$
Content Target

Fig. 1. System overview

As shown in the figure, our system consists of two components: an image transformation network f$_W$ and a loss network ϵ that is used to define several loss functions l$_1$, . . . , l$_k$. The image transformation network is a deep residual convolutional neural network parameterized by weights W; it transforms input images x into output images $\widehat{y}$ via the mapping $\widehat{y}$ = f$_W$(x). Each loss function computes a scalar value l_i($\widehat{y}$, y$_i$) measuring the difference between the output image $\widehat{y}$ and a target image y$_i$. The image transformation network is trained using stochastic gradient descent to minimize a weighted combination of loss functions:

$$W^* = \arg\min_W \mathbf{E}_{x,\{y_i\}} \left[\sum_{i=1} \lambda_i \ell_i(f_W(x), y_i) \right]$$

The loss network φ is used to define a feature reconstruction loss $l^{\varphi}$$_{\text{feat}}$ and a style reconstruction loss $l^{\varphi}$$_{\text{style}}$ that measure differences in content and style between images. For each input image x we have a content target y$_c$ and a style target y$_s$. For style transfer, the content target y$_c$ is the input image x and the output image $\widehat{y}$ should combine the content of x = y$_c$ with the style of y$_s$; we train one network per style target. We used COCO dataset for training the network.

C. Approach 3

This approach is based on ”ReCoNet” paper by Gao et al. ReCoNet is a feed-forward neural network which can generate coherent stylized videos with rich artistic strokes and textures in real-time speed.It stylizes videos frame by frame through an encoder and a decoder, and uses a VGG loss network to capture the perceptual style of the transfer target.

One of the critical issue of Video Style Transfer is Temporal inconsistency, or sometimes called incoherence, can be observed visually as flickering between consecutive stylized frames and inconsistent stylization of moving objects. To mitigate this effect, we use optimization methods guided by optical flows and occlusion masks and calculate the temporal loss. Optical flow is the pattern of motion of objects caused by the relative motion between an observer and a scene. Occlution mask refers to understanding of shading and ambience of the scene.We consider binary pixels values based on the illumination of every pixel(1 at traceable pixels or 0 at untraceable pixels).

Also in real-world videos there exist luminance differences on traceable pixels between consecutive image frames. Luminance differences cannot be captured by temporal losses based on optical flows as optical flow estimation is done by using brightness constancy assumption. To consider this luminance difference, we further introduce a luminance warping constraint in our temporal loss.

With an intuition that same moving object should have same color appearances in consecutive frames, we apply a feature-map-level temporal loss to the encoder such that same object should possess the same features in high-level feature maps.

ReCoNet consists of three modules:

- Encoder
- Decoder
- VGG-loss network

Encoder

Encoder encodes image frames (I$_t$) to feature maps (F$_t$) with aggregated perceptual information. The feature-map-level temporal loss is computed on its output F$_t$ and previous feature-map F$_{t\text{-}1}$. We used three Convolutional layers and four Residual Blocks in the encoder.

Decoder

The decoder is designed to decode feature maps (F$_t$) to a

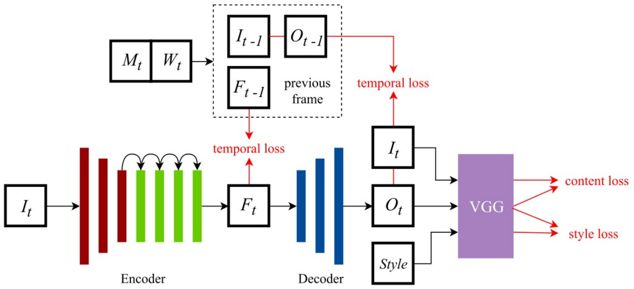

Fig. 2. ReCoNet: I_t , F_t , O_t denote the input image, encoded feature map and stylized output image at time frame t. M_t and W_t denote the occlusic mask and the optical flow between time frames t 1 and t.

stylized image (O_t).Here we compute the output-level temporal loss. We used two Up-sampling Convolutional layers with a final Convolutional layer in the decoder. We adopt instance normalization after each convolution process to attain better stylization quality and Reflection padding is used at each convolutional layer.

VGG – loss network

We use VGG-16 network pretrained on ImageNet to compute the perceptual losses. For each iteration, the VGG network processes each of the input image frame (I_t), output image frame (O_t) and style target independently. The content and style losses are then computed based on the generated image features.

In the inference stage, only the encoder and the decoder will be used to stylize videos frame by frame.

Loss Functions

A two-frame synergic training mechanism is used in the training stage. Temporal losses are calculated between the two frames, current and previous frames at feature maps and at stylized output. Perceptual Losses is calculated on each frame independently and summed up.

Feature-map-level Temporal loss:

The feature-map-level temporal loss penalizes temporal inconsistency on the encoded feature maps between two consecutive input image frames:

$$\mathcal{L}_{temp,f}(t-1,t) = \frac{1}{D} M_t \| F_t - W_t(F_{t-1}) \|^2$$

F_{t-1} , F_t - feature maps outputted by the encoder for previous and current input frames

W_t - the ground-truth forward optical flow

M_t - the ground-truth forward occlusion mask

D = C × H × W where C-channel size , H-image height, W-image width W of the encoded feature maps F

Optical Flow and Occlusion masks are downscaled to simulate temporal motions in high-level features.

Output-level Temporal loss:

The output level temporal losses considers changes in luminance of traceable pixels calculates Warping error for both input and output images.

The relative luminance Y = 0.2126R + 0.7152G + 0.0722B, same as Y in XYZ color space, is added as a warping constraint for all channels in RGB color space:

$$\mathcal{L}_{temp,o}(t-1,t) = \sum_c \frac{1}{D} M_t \| (O_t - W_t(O_{t-1}))_c - (I_t - W_t(I_{t-1}))_Y \|^2$$

c ϵ [R, G, B] is each of the RGB channels of the image

Y - relative luminance channel

O_{t-1} , O_t - the stylized images for previous and current input frames respectively

I_{t-1} , I_t - the previous and current input frames respectively

W_t - the ground-truth forward optical flow

M_t - the ground-truth forward occlusion mask

D = H × W where H-height and W-width of the input/output image.

Perceptual Losses:

We adopt the Content loss $L_{content}(t)$, the Style loss $L_{style}(t)$ and the total variation regularizer $L_{tv}(t)$ for each time frame t. The content loss and the style loss utilize feature maps at relu3_3 layer and [relu1_2, relu2_2, relu3_3, relu4_3] layers respectively.

The final **loss function** for the two-frame synergic training is:

$$\mathcal{L}(t-1,t) = \sum_{i \in \{t-1,t\}} (\alpha \mathcal{L}_{content}(i) + \beta \mathcal{L}_{style}(i) + \gamma \mathcal{L}_{tv}(i)) + \lambda_f \mathcal{L}_{temp,f}(t-1,t) + \lambda_o \mathcal{L}_{temp,o}(t-1,t)$$

where alpha, beta, gamma, lamda_f and lamda_o are hyperparameters for the training process.

We used MPI Sintel Dataset for training the network.

IV. Comparison of the approaches

Following are the result of the 3 approaches:

A. Approach 1

A side by side comparison of actual clip and stylized clip is available here .

The stylized video clearly lacks both colour and pattern consistency.

B. Approach 2

A side by side comparison of actual clip and stylized clip is available here

Here the both colour and patterns are quite stable which can be improved by training on bigger data-set. Since the image

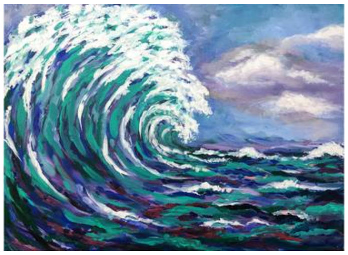

Fig. 3. The reference style image used

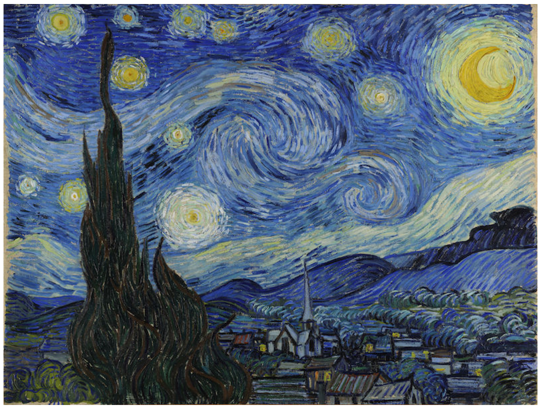

Fig. 4. The reference style image used

and pattern resolutions are different the patterns are a little bit pixel-ted.

C. Approach 3

A side by side comparison of actual clip and stylized clip is available here
The reference style image used here is same as used in Approach 2.
Although color consistency is there in the video but pattern are consistent enough which can be improved by training on larger data-set of optical flow.

Conclusion

Here, three different approaches have been studied and implemented for Video Style Transfer. One shortcoming that can be inferred is that one artistic image can't describe the whole style of the artist which can be solved by fetching similar images to the input reference image before the start of training and video processing, which we plan to implement in the near future. These techniques aim to solve Video Style Transfer problem similar to what the Prisma app does.

References

[1] Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)

[2] Justin Johnson, Alexandre Alahi, Li Fei-Fei : Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv:1603.08155v1 (2016)

[3] Chang Gao, Derun Gu, Fangjun Zhang, Yizhou Yu : ReCoNet: Real-time Coherent Video Style Transfer Network. arXiv:1807.01197v2 (Nov 2018)

[4] Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Advances in Neural Information Processing Systems 28. (May 2015)

[5] Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky : Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv:1607.08022v3 (Nov 2017)