Photo-realistic Style Transfer for Videos

Michael Honke, Rahul N. Iyer and Dishant Mittal

University of Waterloo, ON, Canada

Abstract. In this paper we present an algorithm to do photo-realistic style transfer on a video given a reference image. Photo-realistic style transfer is a technique which transfers colour from one reference domain to another domain by using deep learning and optimization techniques. Here, we present a technique which we use to transfer style and colour from a reference image to a video. Optical flow is used to maintain consistent style transfer over multiple frames of video.

Keywords: style transfer, video, optimization, image segmentation, photo-realistic style transfer, videos, optical flow

1 Introduction

Parameters such as time of day and season often have significant impact on the meaning of images. They don't change the content of an image, but are part of the image's style. Adjusting these details after an image is captured is non-trivial. Furthermore, adjusting such parameters for a video sequence adds additional complexity to the problem. We show how to transfer the style from one photorealistic image to a whole video sequence using deep learning based style transfer. We make use of two prominent works in style transfer.

Ruder et al. [1] presented a technique which enables the transfer of style from an artistic image (for instance, a painting) to a complete video sequence. They extended static style transfer developed by Gatys et al. [2] and Johnson et al.[3] to complete video sequences. They demonstrated that a naive implementation, independently processing each frame of the video, leads to glimmering and untrue irregularities. This was because the solution of the style transfer task was not steady. With the goal of stabilizing the transfer process and preserving an even transition between independent video frames, they established a temporal constraint that penalized divergence between two consecutive frames. This temporal constraint used the optical flow present in the original video and applied a penalty to the divergence between frames along pixel trajectories. Non-concealed regions and the boundaries of the motion were eliminated from the penalizer. This imparted the liberty to rerender the unconcealed regions and deformed motion boundaries while conserving the look of the rest of the image.

Luan et al. [4] developed a deep-learning technique particularly for style transfer in photorealistic images. They reliably transplanted style from a reference image to a content image while preventing the content from being distorted. Their primary contribution was to restrict the style transfer to be locally affine

in colorspace, and to depict this restriction as a custom and completely differentiable loss function. They proved that their technique successfully prevents distortions and produces convincing photorealistic style transfers over a wide variety of content, including transfer of parameters like time of day, season, weather and artistic edits. They utilised the Matting Laplacian to restrict the conversion from the input to the output to be locally affine in colorspace. Furthermore, they used semantic segmentation to enable style transfer between like content only.

In our implementation we merge the loss functions and optical flow method used by Luan et al.[4] and Ruder et al. [1] to get a cumulative loss function. This is used to transfer style from a photorealistic reference image to an entire video sequence.

2 Algorithm

Ruder et al.'s [1] main aim is to render a video of stylized images x describing the style of an image a and the content of video frames p. Gatys et al. [2] derived an energy minimization problem consisting of a content loss and a style loss. The central concept is that attributes extracted by a convolutional neural network (CNN) contain details about the content of the image, whereas the correlations of these attributes store the details related to style. The expected loss can be minimized by iteratively updating an initially randomized noise image. If each frame is independently processed than this iterative optimization process doesn't consistently style the same objects across frames. To solve this problem Ruder et al. [1] created a temporal consistency loss using optical flow. A forward-backward consistency check of optical flow is performed so that style transfer can be initialized for recently disoccluded objects. After which further deviations in these objects' appearances will be penalized. A similar process is done for motion boundaries. The temporal loss given is

$$\mathcal{L}_{temporal}(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{c}) = \frac{1}{D} \sum_{k=1}^{D} c_k \cdot (x_k - \omega_k)^2 \tag{1}$$

where $\boldsymbol{c} \in [0,1]^D$ is the weight per pixel, zero for recently disoccluded pixels, or those at motion boundaries, otherwise it is one. D is the total number of pixels (across all colour channels). $\boldsymbol{\omega}$ is the optical flow of a previous frame warped to the current one $\boldsymbol{x}$.

On each iteration the image is processed by the CNN to update the loss functions. Ruder et al.'s complete loss function is then

$$\begin{aligned} \mathcal{L}_{video}(f^{(i)}, a, x^{(i)}) = \alpha \mathcal{L}_{content}(f^{(i)}, x^{(i)}) + \beta \mathcal{L}_{style}(a, x^{(i)}) \\ + \gamma \sum_{j \epsilon J, (i-j) \geq 1} \mathcal{L}_{temporal}(x^{(i)}, w^{i}_{i-j}(x^{i-j}), c^{(i-j,i)}) \end{aligned} \tag{2}$$

where i denotes the index of a frame, $f^{(i)}$ is the i^{th} frame of the video, a is the style image, $x^{(i)}$ is the i^{th} stylized frame to be generated, c denotes temporal weight,

ω is a function that warps a given frame using the optical flow field that was estimated between two images. J denotes the set of indices each frame should take into account relative to the frame number, e.g., with J = 1, 2, 4, frame i takes frames i−1, i−2, and i−4 into account. Taking the temporal loss over multiple previous frames allows for a long term temporal loss to be established. $\mathcal{L}_{\text{content}}$ and $\mathcal{L}_{\text{style}}$ are defined by Gatys et al. [2]

Luan et al. [4], apart from $\mathcal{L}_{\text{content}}$ and $\mathcal{L}_{\text{style}}$ described above, explain how to regularize these losses to preserve the structure of the input image and generate outputs that are photorealistic. Rather than directly imposing this constraint on the output image they applied it on the transformation which has been applied to the input image. Describing the space of photorealistic images is a problem that remains unsolved. However, the authors proposed that we don't actually need to solve it if we leverage the fact that the input that would be fed into the model is already photorealistic. Their plan was to assure that this fact should not get lost during the transfer process by attaching a term to the equation of the original loss function which penalizes image deformations. The proposed solution was to search for the transform of an image that is locally affine in color space. That means to search for a function such that for every output there exists an affine function that maps the corresponding RGB components in input to their output equivalents. They build upon the Matting Laplacian of Levin et al. [5]. They proposed the following loss function

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^T M_I V_c[O]$$

where M_I is a standard linear system that only depends on the input image I and $V_c[O]$ is the vectorized version (N 1) of the output image O. M_I is calculated by first finding the Matting Laplacian of the content image.

One limitation of the style loss presented by Gatys et al. [2] was that the Gram matrix is computed over the entire image. A precise distribution of feature maps is completely encoded by the Gram matrix up-to an isometry. However, this can cause spillovers as that limits its power to adapt to variations in semantic context. The solution to this problem was that by keeping the set of labels constant (i.e. sky, buildings, water, etc.) Luan et al. [4] generated image segmentation masks for both the input as well as reference images. They included the masks to the input image as supplementary channels and by appending the segmentation channels they built up the neural style algorithm. Finally that style loss was updated as follows

$$\mathcal{L}^l_{s+} = \sum_{c=1}^{C} \frac{1}{2N^2_{l,c}} \sum_{i,j} (G_{l,c}[O] - G_{l,c}[S])^2_{i,j}$$

where C is the number of channels in the semantic segmentation mask, S is the style image and G is the Gram matrix. Finally they formulated the photorealistic style transfer loss function by combining all 3 components losses together as

$$\mathcal{L}_{total} = \sum_{1=1}^{L} \alpha_l \mathcal{L}_c^l + \tau \sum_{1=1}^{L} \beta_l \mathcal{L}_{s+}^l + \lambda \mathcal{L}_m$$

where L is the total number of convolutional layers and l indicates the l_{th} convolutional layer of the deep neural network, τ is a weight that controls the style loss. α and β are the weights to configure layer preferences. λ is a weight that controls the photorealism regularization.

To implement the style transfer from one photorealistic image to a whole video sequence using deep learning we merged the concepts from both the works i.e from Ruder et al. [1] and Luan et al. [4]. Precisely, we accomplished this in two parts:

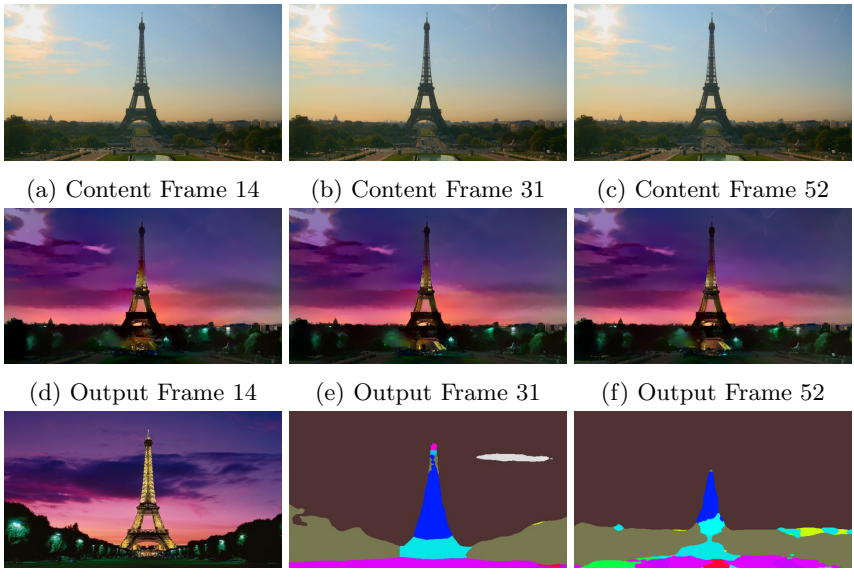
- We integrated the loss functions used in photorealistic style transfer to the loss functions used in artistic video style transfer. After reviewing their implementations, our program was written using the framework of the video transfer algorithm. The style loss of the video method was replaced with that of the photorealistic one. New to the video transfer algorithm was the photorealistic affine transform constraint.
- We used a neural network to perform semantic segemnetation on the reference image and all the frames of the video. This helped us to automate the task of segmentation which can then be used in computing the overall loss for the videos. The style loss term in artistic style transfer in videos is replaced completely with the semantic segmentation loss used by Luan et al. [4]. The final loss function can be written as

$$\begin{aligned} \mathcal{L}_{final} &= \sum_{1=1}^{L} \alpha_l \mathcal{L}_c^l + \tau \sum_{1=1}^{L} \beta_l \mathcal{L}_{s+}^l + \lambda \mathcal{L}_m \\ &+\gamma \sum_{j \epsilon J, (i-j) \geq 1} \mathcal{L}_{temporal}(x^{(i)}, w_{i-j}^i(x^{i-j}), c_{long}^{(i-j,i)}) \end{aligned} \tag{3}$$

where the symbols are as defined in the previous sections.
Deepflow [6] was used to compute the optical flow of the different frames in the video. To get the segmentation of a particular image we trained a neural network on the ADE20K dataset [7] which provides 20,210 annotated images for training, 2000 for validation and 3000 images for testing for 900 different types of categories defined in SUN database [8]. We trained a Dilated ResNet-34 architecture. By training this we get an IOU of 74.9% on the validation set. As our style transfer algorithm is dependent on the segmentation, if the model has problem segmenting the image the algorithm also struggles.

3 Results

The presented method is used to convert a daytime time-lapse of the Eiffel tower to a sunset equivalent using a sunset reference image of the tower. The



(a) Content Frame 14 (b) Content Frame 31 (c) Content Frame 52

(d) Output Frame 14 (e) Output Frame 31 (f) Output Frame 52

(g) Reference Image (h) Reference Image Segmentation (i) Content Frame 52 Segmentation

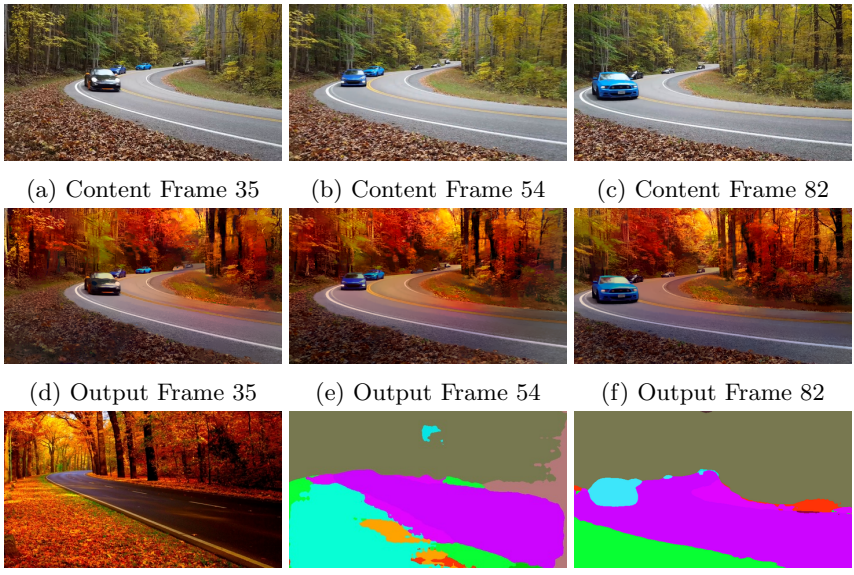
Fig. 1: Eiffel Tower photorealistic video style transfer results.

algorithm successfully transfers the general colour of the sky, and the ground lighting. The lighting of the tower is only partially transferred. This was due to the segmentation method failing to classify the top of the tower as a building (dark blue), instead losing it amongst the sky segment as seen in Figure 1i. The output frames successfully track the motion of the sky, as well as vehicles around the base of the tower. While general motion is minimal, we show that our algorithm produces consistent output without temporal artifacts.

Having shown consistent landscape photorealistic style transfer, the algorithm was applied to a scene with a higher degree of movement. Here, a summer video of cars racing down a road has its season transformed later into fall. The leaf colour of the trees was successfully transferred from the reference image. Also the tree trunk colour was changed as the tree species are different between the reference image and the content video. More importantly, the cars were left unaffected as no reference cars were contained in the reference image. For this example the reference segmentation was manually touched up to improve results. This requires minimal time to do as there is only one reference frame.

4 Conclusion

Two previous style transfer works were introduced. We then produced a novel combination of these two paper's loss functions to bring photorealistic style transfer to video content. This is our most significant contribution. We know of



(a) Content Frame 35 (b) Content Frame 54 (c) Content Frame 82

(d) Output Frame 35 (e) Output Frame 54 (f) Output Frame 82

(g) Reference Image (h) Reference Image Segmentation (i) Content Frame 82 Segmentation

Fig. 2: Cars in a forest photorealistic video style transfer results.

no previous work that achieves this same task. By successfully demonstrating our algorithm we also further validate the results of Ruder et. al. [1] and Luan et al. [4]. Ruder's video style transfer algorithm is shown to be robust enough to be used within other types of style transfer algorithms. Luan's photorealistic style transfer method also produces consistent enough results to successfully stylize multiple frames of video. Issues seen in our results are typically due to poor segmentation. This is a critical area of improvement for future work.

References

1. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In Rosenhahn, B., Andres, B., eds.: Pattern Recognition, Cham, Springer International Publishing (2016) 26–36
2. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2414–2423
3. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, Springer (2016) 694–711
4. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. CoRR, abs/1703.07511 **2** (2017)
5. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 61–68

6. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1385–1392
7. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017)
8. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE (2010) 3485–3492