A Real-Time, Stable, Style-Invariant Architecture for Artistic Style Transfer of Videos

Michelle Bao, Ankush Swarnakar

{baom, ankushsw}@stanford.edu

Overview

- Style transfer is the procedure of **reproducing a content image as a pastiche** in the style of another image
- Architectures for single-image style transfer have progressed to be real-time, content-invariant, and style-invariant
- **Video** style transfer is **comparatively slower** and **produces temporal inconsistencies** between frames called “popping”
- We present a novel architecture that produces **stable, stylized videos in real-time for any content and style input** using a noise-resilient loss function and adaptive instance normalization

Related Work

Single-Image Style Transfer

- *Johnson, et al.* proposed the **Image Transformation Network (ITN)** which uses a CNN to produce a pastiche from an input content image for a given style image by iteratively minimizing the pastiche’s loss
- *Huang, et al.* introduced an **Adaptive Instance Normalization layer (AdaIN)** between the encoding and decoding layers to learn the instance normalization parameters for any style input, enabling style- and content-invariance

Video-Style Transfer

- Naive approaches independently apply style transfer to individual frames, creating **inconsistencies between frames** that reduce perceptual quality
- *Ruder, et al.* proposed the use of **optical flow estimation** to enhance temporal consistency across frames but such methods are computationally expensive and slow

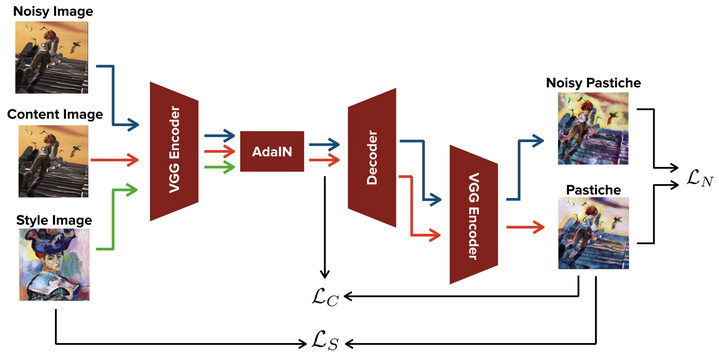
Approach

Dataset:

- MPI-Sintel: animated videos from open-source film *Sintel*, each with 20-50 frames and 1024 x 436 pixels per frame
- 23 training sequences, 12 testing sequences

1. Encoder and Decoder (CNN)

- **Encoder to extract features** from input images using convolutional & max-pool layers from a pre-trained VGG-19 from *Simonyan, et al.*
- **Decoder to map features** back to image space **with stylization**



2. Adaptive Instance Normalization (AdaIN)

- Instance normalization parameters (γ, β) largely determine style of pastiche
- We use **AdaIN** operator proposed by *Huang, et al.* between the encoder & decoder to allow for **stylization in any style**

$$\text{AdaIN}(x, y) = \sigma(y)_{N,C}\frac{(x - \mu(x)_{N,C})}{\sigma(x)_{N,C}} + \mu(y)_{N,C}$$

- Operator shifts content features to match statistics of style features

3. Noise-Resilience

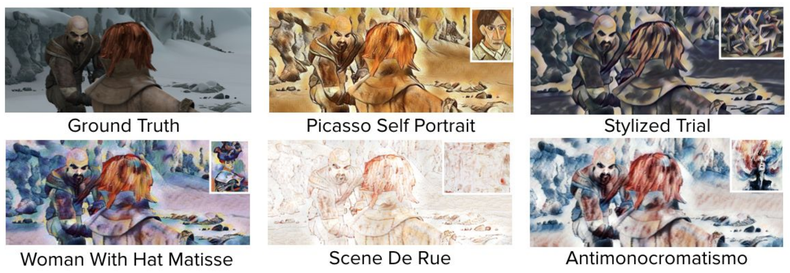
- Because most temporal inconsistencies arise from noise between frames, our operator **adds random noise** to the content image to generate a “noisy pastiche”
- ITN is trained to **minimize per-pixel loss between “noisy” and original pastiche**

$$\mathcal{L}_N = \frac{1}{N}\sum_{j,k}(g(n(x))_{j,k} - g(x)_{j,k})^2$$

Results

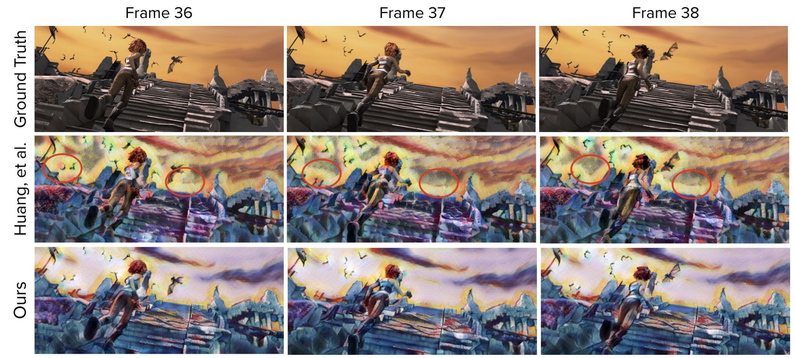
Style-Invariance

- ITN produces **pastiches of high style- and content-quality** while preserving encoding and decoding parameters across style images
- AdaIN effectively stylizes input images and **guarantees style-invariance** instead of relearning CNN for each style



Temporal Consistency

- ITN with the style loss operator **significantly reduces popping effects and motion stylization artifacts** in videos in sequential frames with the style loss operator compared to without, **enhancing perceived consistency**



- Used Structural Similarity Index (SSIM) and temporal error to quantify temporal & perceptual consistency

$$E_{temporal}(\hat{x}, \hat{y}) = \sum_{t=1}^{T-1}\frac{((\hat{x}^t - \hat{x}^{t+1}) - (\hat{y}^t - \hat{y}^{t+1}))^2}{\hat{x}_k^t - \hat{x}_k^{t+1} + \epsilon} \qquad E_{tempAvg}(\hat{x}, \hat{y}) = \frac{E_{temporal}(\hat{x}, \hat{y})}{T-1}$$

- Our model **decreased the temporal error by 60+%** for select videos and averaged a temporal error decrease of 6.69%
- Our model **increased SSIM by 70+%** for select videos and averaged an SSIM increase of 3.20%

Run-Time Efficiency

- Our ITN runs with **comparable speed to single-image style transfer** and is roughly **25x faster** than the standard optical flow approach towards video style transfer
- ITN can process nearly **20 frames-per-second**

Conclusion

- We present a **real-time, stable, and style-invariant architecture** for video style transfer using **adaptive instance normalization** and a **noise-resilient loss objective**
- Observed quantitative (60+%) and qualitative improvements in temporal consistency and substantial increase in run-time (25x)
- In future work, we hope to investigate how selectively adding noise with masks during training can preserve higher-degrees of stylization
- We enable applications in camera filters, data augmentation, computer graphics, etc.