CS 4701 Status Report

Video Neural Style Transfer

**Team Name:** NStylist, **Members:** Roberto Halpin (rgh224), Shirley Thomas (smt244), Meiqi Wu (mw849)

Neural style transfer(NST) is the technique of combining the artistic style of one image to another using deep learning networks. It is one of the most fun techniques in deep learning.

It merges two images, namely the “content” image (Fig 1. A) and “style” image (Fig 1. C), to create a “generated” image (Fig 1. B).

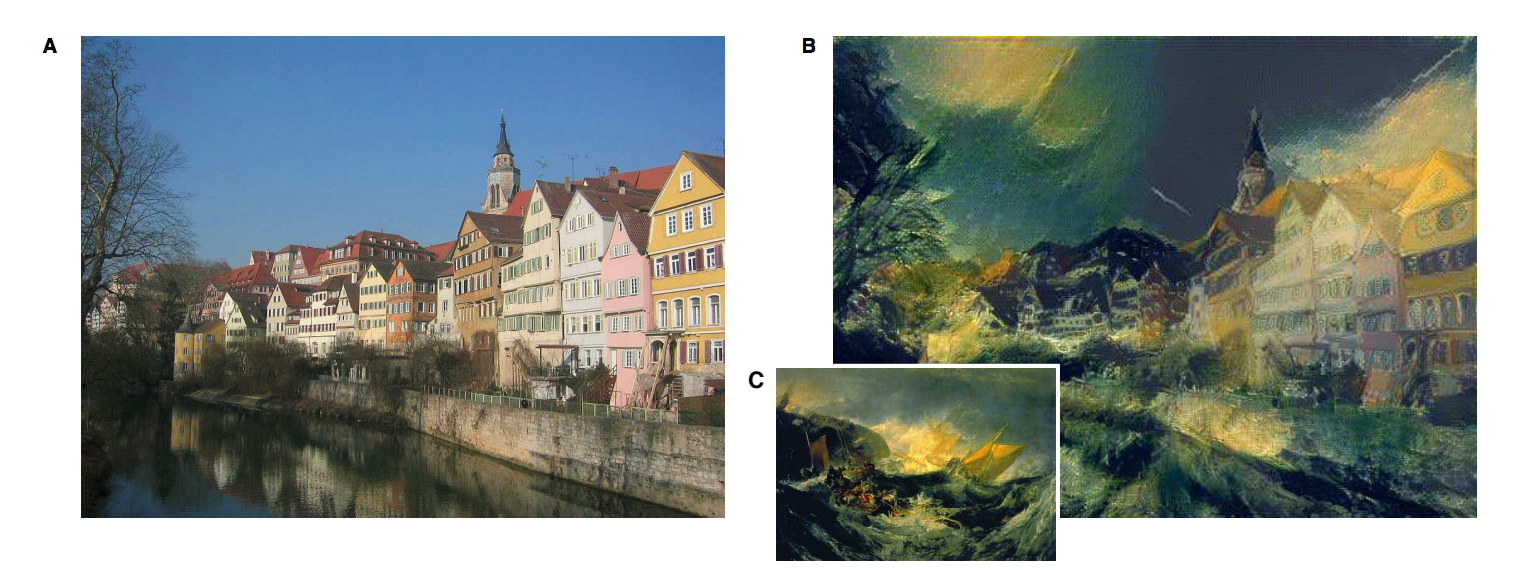


Fig 1. Example of art generation with neural style transfer. (A) “Content” image: photograph depicting the Neckarfront in Tubingen, Germany (Photo: Andreas Praefcke). (B) “Generated” image. (C) “Style” image: The Shipwreck of the Minotaur by J.M.W. Turner, 1805.

In this project, we would be exploring the neural style transfer on videos. Here, we would be taking the style from an image and applying it to a content video to generate a new stylized video. The simple strategy would be to implement the style framewise, but this would bring discontinuities among the frames, leading to an unappealing flickering effect. To remedy this, when training the model, we would be adding consistency loss to maintain the flow of the video.

We target to build a browser based application which allows users to upload ~~any style image~~ and a content video to view a generated video.

After implementing the framewise model, we discovered that the architecture we based our model on was very slow, not functional for any demonstration. Thus, we did a literature search and found an interesting [paper](http://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Real-Time_Neural_Style_CVPR_2017_paper.pdf) that addressed this problem on video style transfer. The paper claims real-time video style transfer, so we have shifted our focus to implementing this paper.

This paper requires us to train models conditioned on style images offline. Thus, when performing inference, we will load trained models conditioned on certain style images. Meaning that users must choose one of the predefined style images, but are still free to upload their own content videos. Also it is important to note that we have GPU access to fully train all our models, and if a GPU is also required during inference, we will demo the application on a machine with GPU access.

For our project, we will be exploring the following aspects of AI: convolutional neural network, deep learning, computer vision, and (maybe) transfer learning.

**Evaluation Criteria:**

**Qualitative evaluation:**

Due to the nature of our project, a main component of the qualitative evaluation condenses into how interesting, or “cool” the generated video appears. It is important that the produced video maintains the structural form of the content video, but also adapts the style from the designated style image. In addition, the generated video should look smooth, it should not introduce unpleasant jaggedness that makes the video unappealing. Clearly qualitatively measuring these characteristics has a great deal of subjectivity, thus in addition to evaluating them ourselves, we plan on showing the generated videos to others. By doing this we can receive a form of user opinion that can give alternative insights. Even though our project mainly consists of qualitative results with no clear, informative quantitative metric, we have some ideas of evaluating our project quantitatively.

We can also compare our qualitative results of our new, real-time video style transfer model to our previous framewise model to identify any key differences. Additionally, we have identified existing apps that perform video style transfer, so when showing results to users we will show them our output as well as the apps’ outputs and ask them to decide their preference. Of course we will make sure that the users cannot tell the source of the stylized videos a priori (i.e. removing watermarks).

**Quantitative evaluation:**

We can quantify the consistency between adjacent images to make sure that the video generated video is smooth across time. For example, we plan to use dynamic range (difference between minimum and maximum pixel values), and correlation between adjacent images, as metrics to quantify the consistency. Another way of quantitative evaluation is to plot the loss function with learning. With a good model, we should see the loss function in a decreasing trend with the progress of training. Most likely the loss function we will use will consist of three terms: content loss, style loss, and consistency loss. Showing the loss values overtime on the trained content video gives an idea of the qualitative characteristics mentioned previously. In addition, we can also show how the loss value deconstructs into the three terms (normalized by any weighting), which gives us a sense of how the model is performing with respect to each characteristic. Lastly, we can take another content video that the model was not trained on, evaluate the model on that content video with the given style image, and find the corresponding loss of the generated video.

For quantitative evaluation, we will use temporal error to quantify the temporal consistency. Temporal error is defined as the average pixel-wise Euclidean color difference between stylized frames and the predictions based on their former stylized frames. A pre-trained forward optical flow will be utilized to predict the next frame based on the former one. We will compute the temporal error of our model and compare it to the published results. If our results can match the published results, then our implementation is correct.

**Project Plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Description** | **Date** | **Status** |
| Literature & coding review | Read any relevant papers or alternative information sources. Know the basics of the coding languages/frameworks we plan to utilize. | 10/2 | Done |
| Implement basic model | Implement the neural video style transfer frame-by-frame model, most likely on Pytorch. ~~Making sure it can output a generated video given a content video and a style image.~~ (Image only) | 10/18 | Done |
| Obtain Dataset | Download 100 videos from videvo.net. Download style images. | 10/18 | Done |
| Train & evaluate on basic model | Train the basic model, and conduct basic qualitative ~~and quantitative~~ evaluation on the trained basic model. (Image only) | 10/25 | Done |
| Status Report | Submit status report | ~~10/30~~  11/1 | Done |
| Preprocess Dataset | Convert the videos in the dataset to frames for calculating the temporal loss. Implement a dataloader to shuffle the frames and provide a frame and the next sequential one for calculation. | 11/6 | In progress |
| Implement the framewise model to work for videos | Making sure frame-by-frame can output a generated video given a content video and a style image. | 11/6 | In progress |
| Modify model | ~~Consider and implement variants of consistency loss and any other modifications to the base model.~~  Implement new model architecture for videos. | ~~11/6~~  11/13 | In progress |
| Train & evaluate on ~~modified~~ all models | Train ~~modified~~ all models, and conduct qualitative and quantitative evaluation on trained ~~modified~~ all models. | ~~11/13~~  11/25 |  |
| Draft design for website/user interface | Design a basic UI to connect to our model. | ~~11/18~~  10/24 | Done |
| Create website | Create a basic UI for users to select ~~upload~~ a style image and upload a content video to view/download the generated video. | 11/25 | In progress |
| Qualitative (final) Evaluation | Evaluate application with users | 11/30 |  |
| Final Report | Finish final report | 11/30 |  |
| Final Report deadline |  | 12/16 |  |

**Prerequisites:**

Pytorch is planned to be used for implementing the deep neural network. For the user interface, HTML and javascript can be used. The Python Flask framework is required to make this a Python web app.

~~Extensive dataset is not required.~~ Video dataset is needed for calculating temporal loss, as mentioned in the paper. Also, some sample paintings to study the style (style image) and content videos to transfer the style to are required.~~, should suffice. The UI will be made flexible to allow the user to input a style image and a content video.~~

This is not related to any work previously done by any of the members in this team.

**References:**

Haozhi Huang et al.; Real-Time Neural Style Transfer for Videos. In CVPR, 2017. <http://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Real-Time_Neural_Style_CVPR_2017_paper.pdf>

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge; A Neural Algorithm of Artistic Style arXiv:1508.06576 [cs] (2015). <https://arxiv.org/pdf/1508.06576> ArXiv: 1508.06576.

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge; Image Style Transfer Using Convolutional Neural Networks. In CVPR, 2016. <https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf>

Manuel Ruder, Alexey Dosovitskiy, Thomas Brox; Artistic Style Transfer for Videos arXiv:1604.08610 [cs] (2016). <https://arxiv.org/abs/1604.08610> ArXiv: 1604.08610.