1. *Briefly,* what does your tool/classifier do?

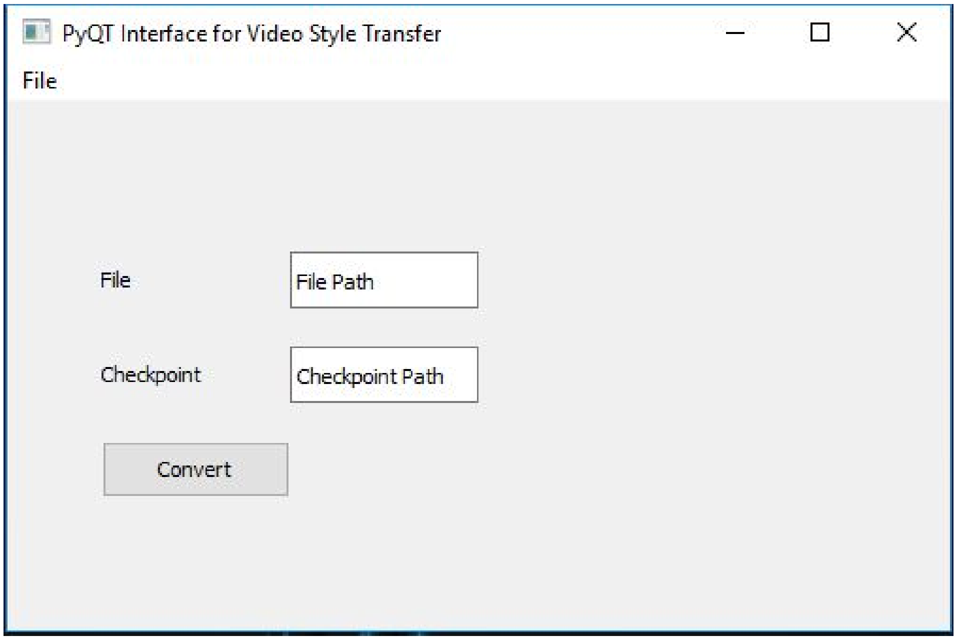
The tool is a modified version of Lengstrom's Fast Style Transfer Library, which is an implementation of Gatys' A Neural Algorithm of Artistic Style, Johnson's Perceptual Losses for Real-Time Style Transfer and Super-Resolution, and Ulyanov's Instance Normalization.

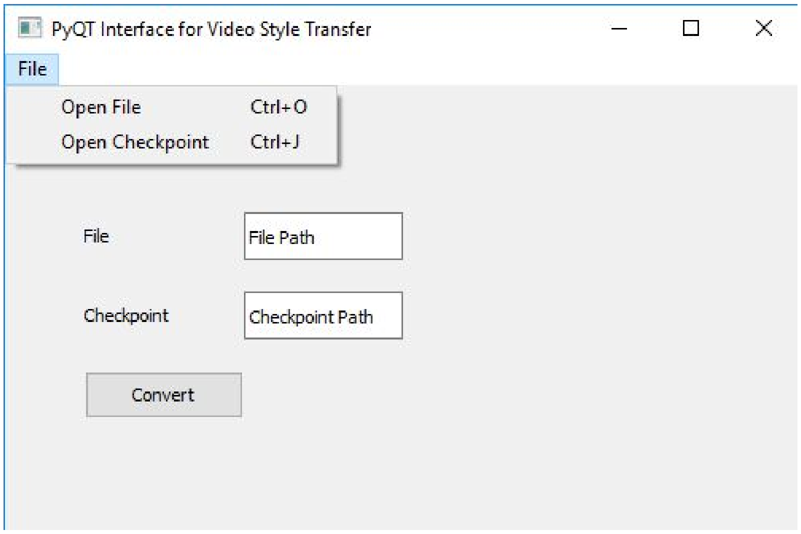
There were two key issues with the library.
1) The library was using deprecated code, as the library was last updated two years ago.
2) The library did not support for preservation of original colors.

I fixed those two issues and created a basic interface that allows you to select the Checkpoint file and the Video to create a final video that merges the two styles.

Running this interface successfully requires access to a Linux Machine with the following :
TensorFlow 0.11.0 - Python 2.7.9, Pillow 3.4.2, scipy 0.18.1, numpy 1.11.2 - A decent NVIDIA GPU (8GB or higher) supporting cuDDN - All the required NVIDIA software to run TF on a GPU (cuda, etc) - ffmpeg 3.1.3 if you want to stylize video
- PyQt4 compatible with your version of python
(These are backend requirements for the program itself to run)

Code : https://github.com/risshiki/fast-style-transfer-master

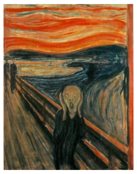




In Windows : run new_interface.exe
In Linux : Run setup_python.sh and then click on new_interface.py
Choose a file and checkpoint and hit convert. A new terminal will execute the given command and display the output. The program will notify you when the conversion is done on the convert button.

Results :

Pewdiepie vs T-Series rendered in Scream style



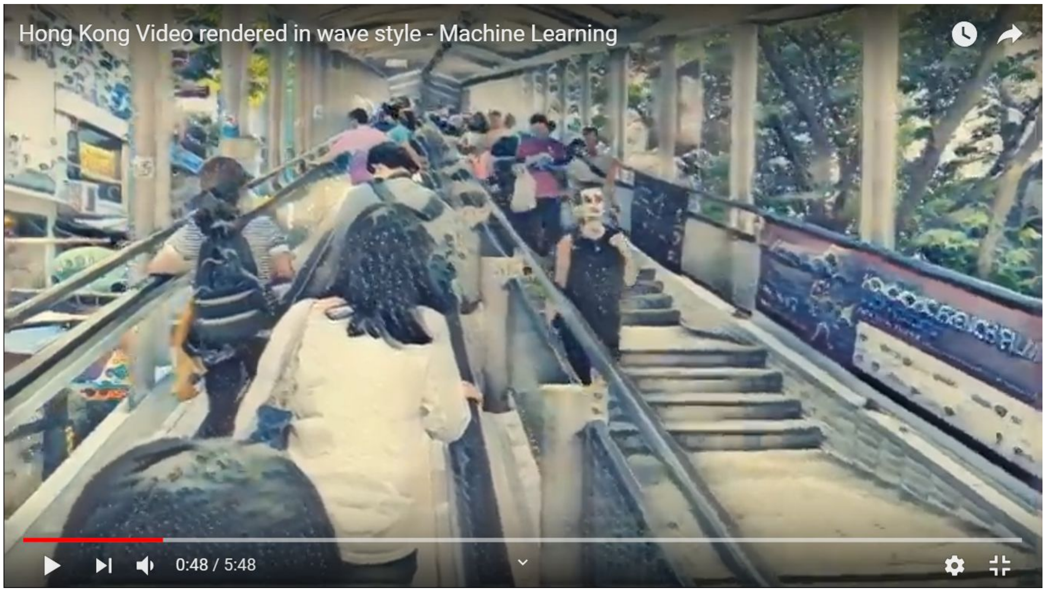
Queen - I want to break free rendered in Rain Princess style

Hong Kong Skyline rendered in Wave painting preserving some of original colors

2. How much better than random is it? Include baseline rates and multiple model fitness statistics, as applicable
   In comparison to the initial approach of image style transfer, which retrains the model each time, the native approach in the library does image style transfer in nearly real time. My added approach of preserving real colors takes approximately thrice the time from the original approach as it performs multiple operations on each frame before and after being passed to the checkpoint. This is to be expected as the frame is being processed thrice for each iteration.

3. Where does your dataset come from, and what are its biases?

I used the VGG neural net and trained the model over the 2014 Coco dataset. The coco dataset has negative connotations in terms of it's associations of certain objects with gender (Kitchen/spoons - women, keyboards/sports equipment - men)
However, this had no impact on the way I applied the dataset. It rendered men and women in identical fashion.



No apparent difference in the way men and women are rendered

4. Who is the tool designed for?

The tool is designed to recreate high quality videos in an artistic style. This is useful for artists and animators who tend to typically have powerful computers and simply want to automate some of the work associated with recreating images and do not want to waste time using apps like Prisma which take up a lot of time for processing a single image.

5. Who is the tool *not* designed for?
   The tool is not designed for artists who want to recreate videos with lower resolution into artistic styles. The Queen video (second link) is a very good example of why that is a bad idea.

6. If the tool were widely adopted, how would it change the lives of users?

a. How might it change the lives of people who don’t use it?

The application can change how we interpret the concept of animation. Although I did not have the time to fully flesh out this component of the application, it would be incredibly cool if this were to be fleshed out with an interface that allowed you to segment the video into smaller chunks and apply specific checkpoint GAN’s to each image. This could revolutionize the way we think about animation.