DIP Final Group 19 Report

Group members

B08902061 資工大四 郭柏廷
B08902070 資工大四 戴培凱

Division of Work

There's no specific division of work, it's about 50%-50%

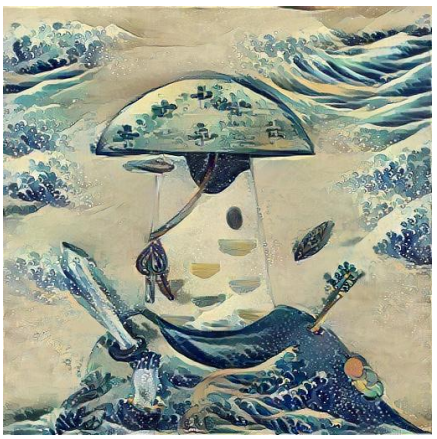
Methods we used

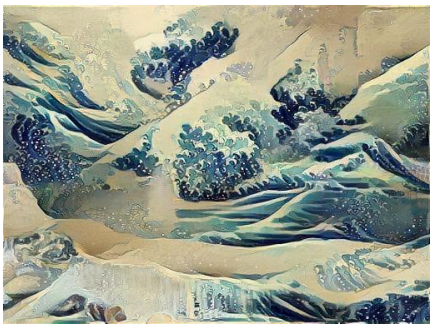
- Style Transfer Algorithm
  - Image only
    - Neural Style Transfer (by Leon Gatys et al.)
    - SANet
  - Video and Image
    - MCCNet
    - Linear Style Transfer(Li et al.)

- Additional functions added
  - Preserve Color Transfer
    - 2 ways to implement
      1. Color Histogram Matching
         - Do Color Transfer on style image with content image, or
         - Do Color Transfer on stylized result with content image
         - Downside
           - Get poor color preserve result if style image has few color
      2. Luminance only transfer
         - RGB -> YUV
         - Y is luminance, UV is colors
         - Do style transfer only on Y channel
  - Reversible Transform

        - Do forward transform on content images before style transfer, and backward transform the stylized result
        - If a transform can be recovered(backward transform) without losing much details, it can be called a Reversible Transform
        - Types of reversible transform we implement
            - Resize
            - Rotate
            - Swirl
            - Wavy
- Procedure of Our Program
    - For image style transfer
        - add noise
        - reversible forward
        - style transfer
        - reversible backward
        - preserve color
    - For video style transfer
        - add noise
        - reversible forward (only resize and rotate)
        - style transfer
        - reversible backward
        - preserve color
    - How to run
        - see github README

Results

- Images
    - Image 1
        -  

    -  
  - Image 2
    -  
    -  
- For videos and more result, please see Presentation and Github
  - Presentation
    - https://docs.google.com/presentation/d/1UF8GscKSB5iiydNLauxmdw-UiMvXuGbfV8lEyJQsEmU/edit#slide=id.p
  - Github
    - https://github.com/B08902070/DIP_final

Discussions

- Ways to improve stylized results
  - Modifying Style Transfer Models
    - pre-train VGG using style transfer data instead of original pretrain weight
    - Modify weights in each layers of VGG by multiplying some coefficients or do normalization

        - Do normalization on loss function
        - Use exponential moving average to reduce noise
    - Pre-processing content and style images
        - add noise to on single color background
        - Do normalization on lightness of style image
        - Add some effect on content images like reversible transform or color preserve
- What could be better?
    - Style Transform effect on video is not obvious
    - Still some flickering in stylized video result
    - The artifact of add noise method on single color background is still obvious in some style images
    - Since our program is designed to apply customize operations on each content-style pair, the style transfer model have to be loaded again after switching to another pair, which largely increase the process time.

Reference

- https://github.com/crowsonkb/style-transfer-pytorch
- https://github.com/diyiiyiii/MCCNet
- https://github.com/sunshineatnoon/LinearStyleTransfer/blob/master/README.md
- https://github.com/MaxReimann/stroke-adjustable-nst-transforms
- https://github.com/GlebSBrykin/SANET