Group member: Qingnian Meng(qm4), Wenshan Xiong(wenshan2), Yuqun Wu(yuqunwu2)

Motivation

We want to automate the process of images alignment and morphing given a set of facial images. We also want to explore the visual effects of combining the cartoonization and morphing.

The impact of this project can be broad. To us personally, it tested and strengthened our knowledge in deep learning, computer photography, and software engineering. From application point of view, our project can be used in instagram filters, cartoon production and more.

Approach

We first trained a ResNet model to detect facial keypoints. Then, we fed the detected key points to our morphing algorithm. After obtaining the morphing transformation between 2 facial images, we did a cross-fade between original images and cartoonized images. Finally, we compose the cross-fade and morphing to generate a GIF image (see the Final GIF result in the Results section).

Implementation details

Facial Key-Points Detection

- Language: Python
- Packages: PyTorch, tqdm, functools, matplotlib, numpy, PIL, scipy
- Reference:
    1. https://stackoverflow.com/questions/41707229/tqdm-printing-to-newline
    2. https://inst.eecs.berkeley.edu/~cs194-26/fa20/hw/proj4/download_ibug_dataset.txt
    3. https://inst.eecs.berkeley.edu/~cs194-26/fa20/hw/proj4/load_ibug_dataset.py
    4. https://gist.github.com/LyleScott/e36e08bfb23b1f87af68c9051f985302

Morphing

- Language: Python
- Packages: cv2, pandas, matplotlib, numpy, scipy
- References:

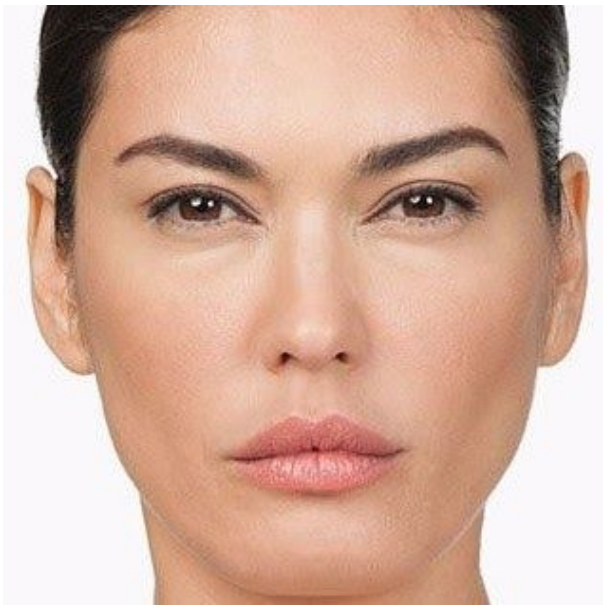
1. https://www.w3resource.com/python-exercises/basic/python-basic-1-exercise-40.php

Cartoonization and combination

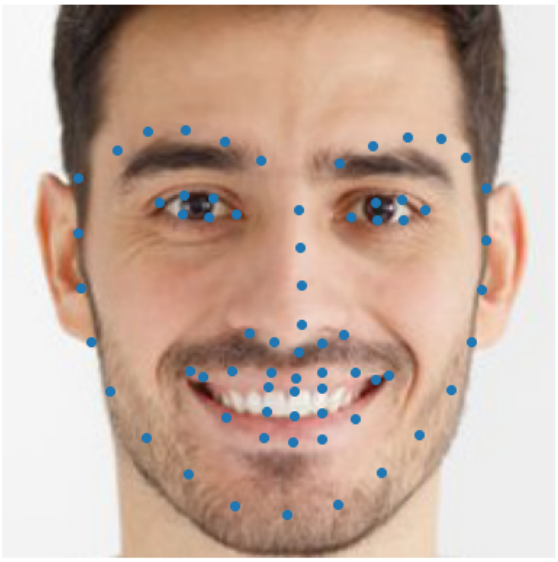
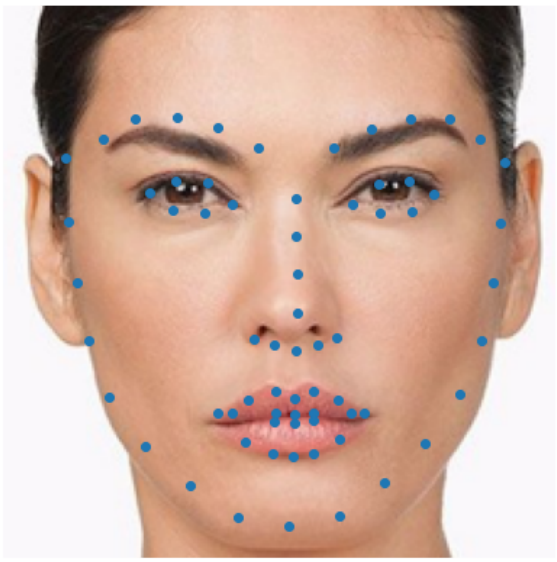
- Language: Python
- Fundation: Morphing functions, Face detection networks, Cartoon Gan
- Reference:
    1. Cartoon Gan: https://github.com/mnicnc404/CartoonGan-tensorflow

Results

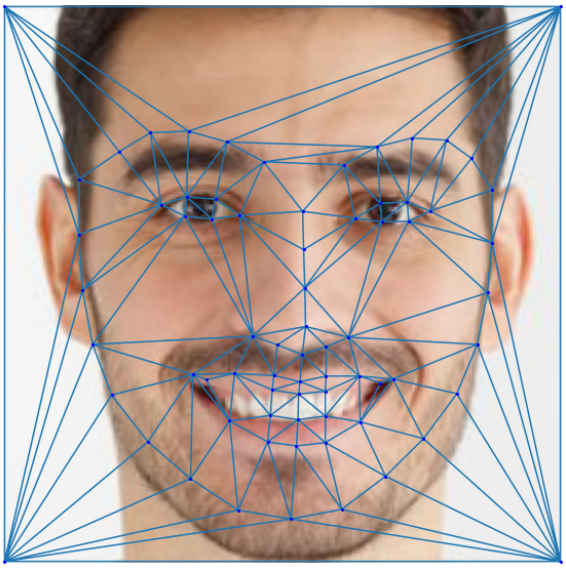
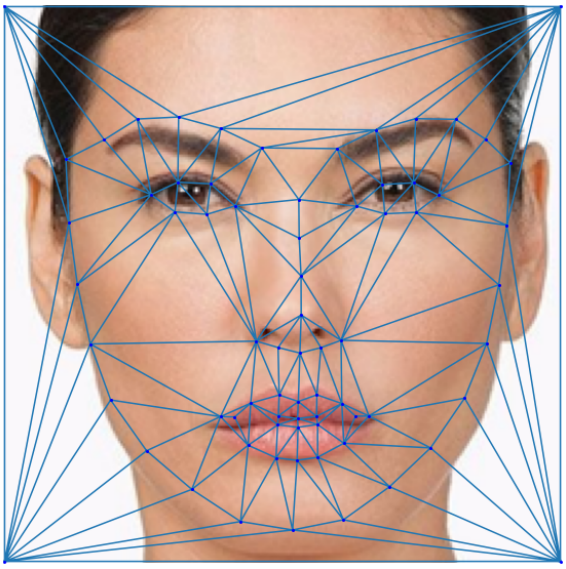
The original images:

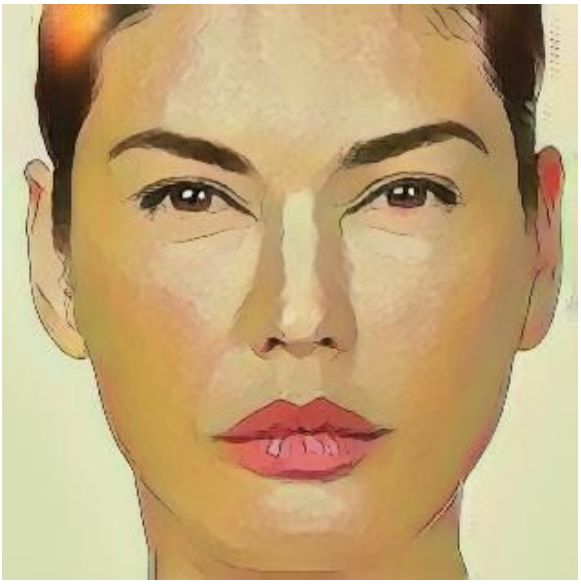
Results from face detection network:

Aligned Images:

Cartoonized Images (`shinkai` style):

Final GIF result:

If GIF is unviewable, please use this link:
https://drive.google.com/file/d/1-o5y0cG5ma1vDixk2CS8ekWllimS1JuV/view?usp=sharing

The GIF consists of 3 parts. First part is the transition of the female face from the original to cartoonized. Second part is the morphing from cartoonized female face to cartoonized male face. The final part is transition from cartoonized male face to the original male face

Challenge

For facial keypoints recognition, our first challenge lies in the data preprocessing. The data we found in Kaggle has 6666 images in total. However, half of the images are mirror versions of the other half. So, the original images are only 3333 in total. To compensate for this, we implemented some data augmentation techniques such as random rotation. Moreover, after careful examination of the given bounding boxes, we found that some boxes are so small that some key-points will lie outside the images. Thus, we decided to define the bounding boxes to be determined by the four outermost (left, right, top, bottom) points plus some custom size boundary.

For the morphing part, we initially used only the key points from the process above for morphing. This resulted in the morphing of the facial features only, and often misaligned with the rest of the head. To solve this problem, we added 4 additional key points at the corners of the picture. This addition ensured that the morphing is applied to the entire image instead of only the face.

For cartoonization, although the GitHub provides a Colab version, it cannot be run directly because it uses TensorFlow and TensorFlow-GPU packages, which are troublesome in Colab. After trying and searching for a while, I have to uninstall the package and install the normal TensorFlow (non-GPU), followed by TensorFlow-GPU, so that it could be used.

We expect to receive 10 points for the challenge since we don’t propose any new techniques.