Division of the work of each team member

蕭延儒:

Integration of the facial expression and style transfer algorithm. Style transfer algorithm implementation.

林祐霆:

Style transfer algorithm development and implementation.

方郁婷:

Communication of front-end and back-end using socket
Facial feature extraction and facial expression detection
Integration of Facial Expressions with live2d Web sdk

Paper title

Interaction with live2d using facial landmark detection, facial expression recognition and image stylization.

Motivation

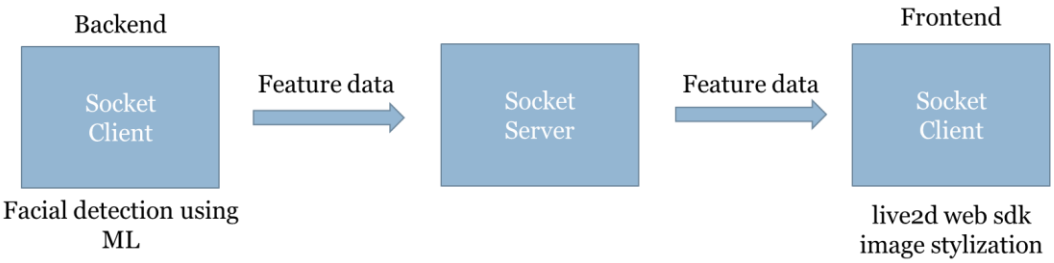
Vtuber(Virtual Youtuber) and machine learning is the trend of the times. We combine them with traditional algorithms, such as image stylization, to make the result more diverse.

Problem definition

We want to build a system that users can control the 3D model through a camera, and the output motion picture will depend on users' facial movement.
We will focus on the facial expression and the overall image style.

Algorithm

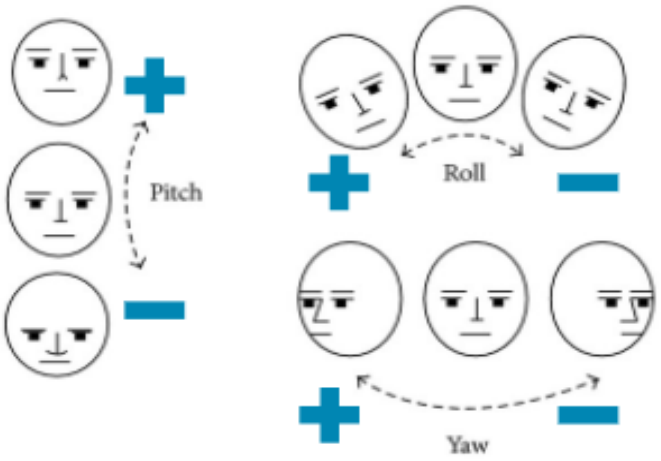
- Architecture.



We choose socket because socket can keep the communication, so feature data can be sent continuously.
We choose web instead of unity because web is easier to modify.

- Facial Detection
  Using Mediapipe to find landmark.
  Mediapipe is made by Google. It can provide excellent, gpu-standard face detection and landmarks detection using cpu only, providing 30 FPS smooth detection.

- Feature data
  - Rotation Vector-> Face Direction
    Using solvePnP function of Opencv and landmark to estimate rotation vector.
    Convert the rotation vector to roll-pitch-yaw.



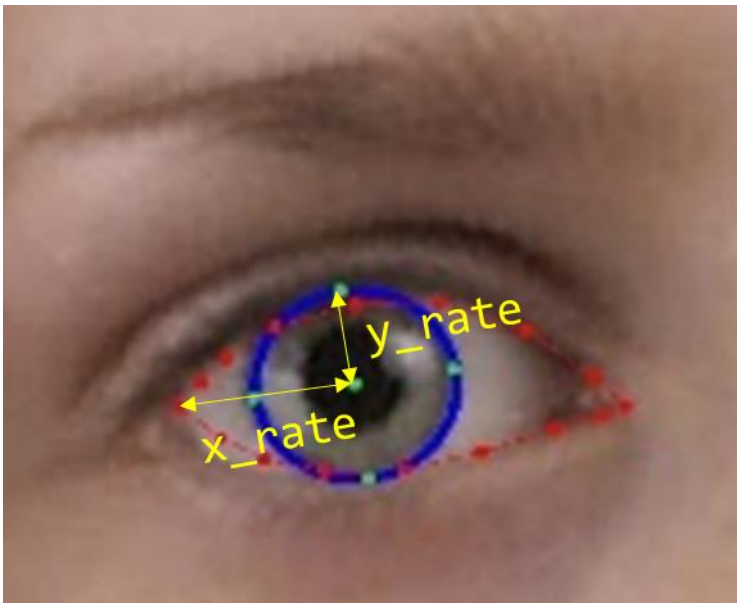
  - Eye aspect ratio -> Blinking

$$EAR_{Lt} = \frac{\|LM_{37} - LM_{41}\| + \|LM_{38} - LM_{40}\|}{2 * \|LM_{36} - LM_{39}\|} * \frac{\|LM_{19} - LM_{33}\|}{\|LM_{27} - LM_{33}\|},$$

$$EAR_{Rt} = \frac{\|LM_{43} - LM_{47}\| + \|LM_{44} - LM_{46}\|}{2 * \|LM_{42} - LM_{45}\|} * \frac{\|LM_{24} - LM_{33}\|}{\|LM_{27} - LM_{33}\|}.$$

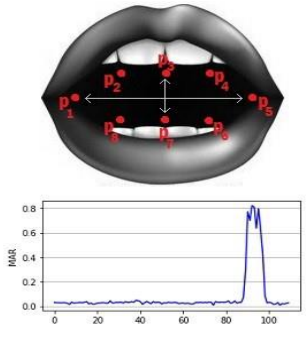
Ref: Using Eye Aspect Ratio to Enhance Fast and Objective Assessment of Facial Paralysis

- Iris detection -> Iris movement



x_rate: how much the iris is toward the left.
0 means totally left and 1 is totally right.
y_rate: how much the iris is toward the top.
0 means totally top and 1 is totally bottom.

- Mouth aspect ratio -> Speaking (open、close)



$$\mathbf{MAR} = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2\,\|p_1 - p_5\|}$$

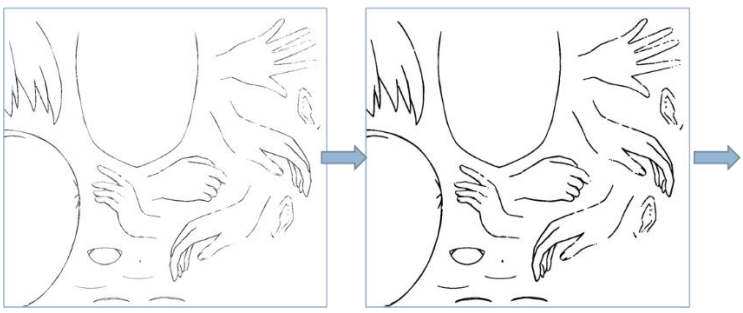
Ref: Asleep at the Wheel: A Computer Vision and Deep Learning Approach to Detecting Drowsiness | by Jaynish P. Vaghela | Medium

- Mouth Distance -> Mouth form (shape)
  Mouth Distance = || *p1- p5* ||
  The feature point is referenced from previous image.

■ Style transfer

- Happy

Our idea is to make the character more energetic.
Step 1: Extract dark color edges from the model
Step 2: Thicken extracted edges and put it back to model
Step 3: (Real-time) Enhance background intensity and overall saturation



Extract dark color edges

Dilate dark color edges

Output
(With intensity and saturation enhancement)

- Angry

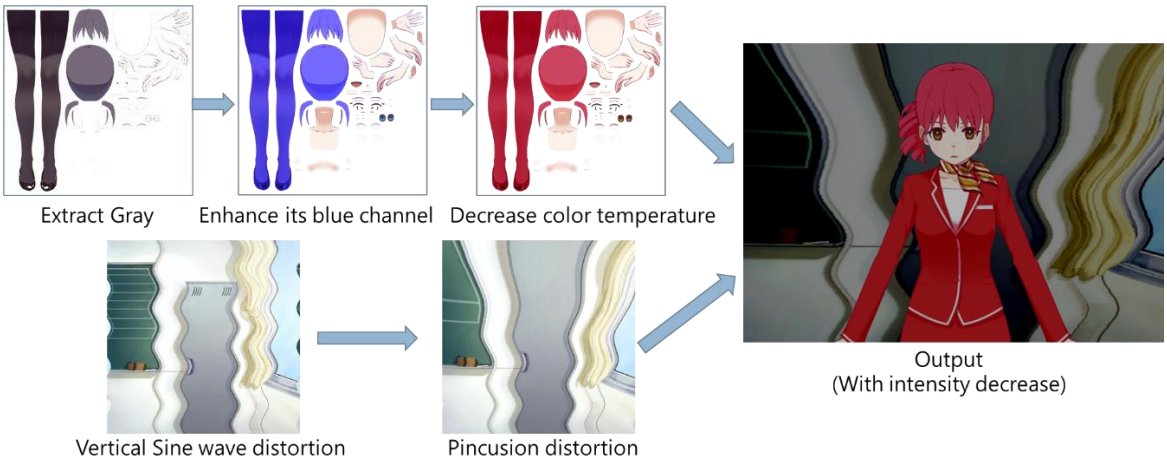
Our idea is to make the image dramatic and powerful.

Step 1: Extract gray color from model and turn them into blue.

Step 2: Decrease color temperature.

Step 3: Distort the background image using sine wave and Pincusion distortion.

Step 4: (Real-time) Decrease overall intensity and enhance contrast.



- Surprise

Our idea is to make the cartoonish shock effect .

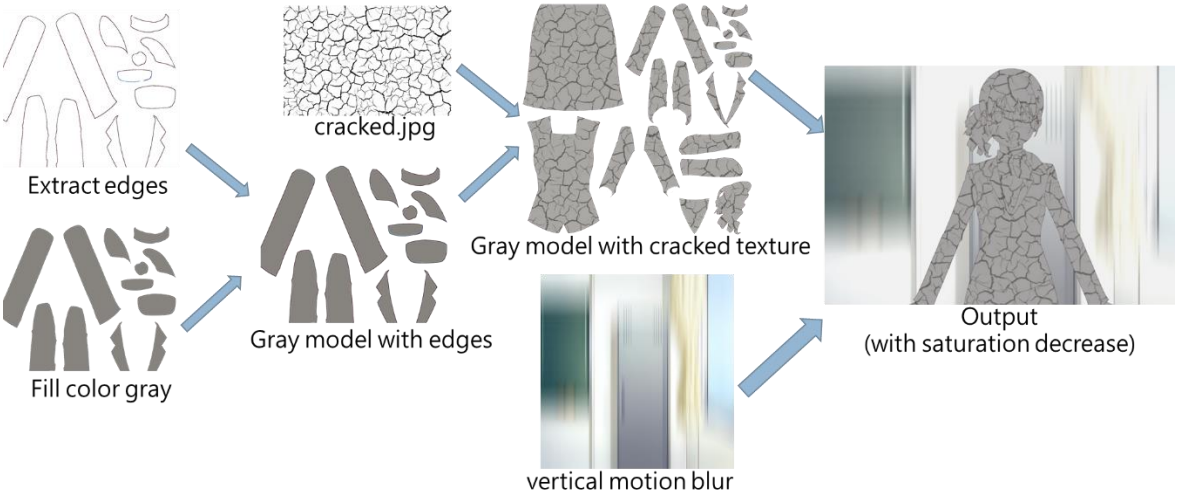
Step 1: Extract edges from the model

Step 2: Turn the whole model into color gray except for edges

Step 3: Add cracked texture on the model

Step 4: Add vertical motion blur to the background image

Step 5: (Real-time) Decrease the overall saturation



- Closing Eyes

Our idea is to make the image artistic.

Step 1: Extract edges from model

Step 2: Calculate the x, y-axis gradient of model

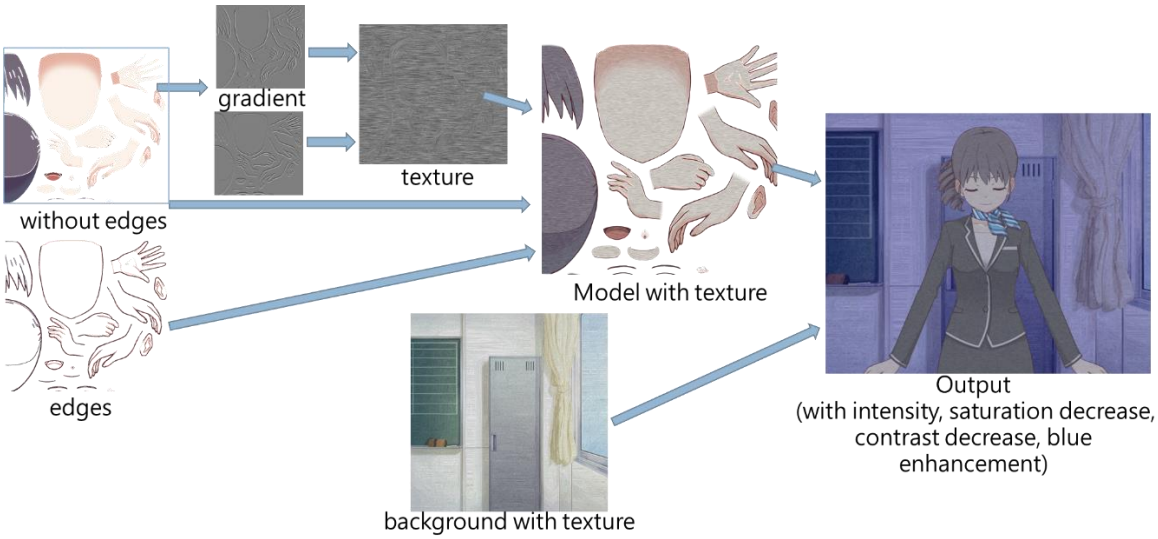
Step 3: Use gradient to calculate LIC(Line Integral Convolution)

Step 4: Use LIC as texture and add it to model

Step 5: Paste edge back to model

Step 6: Repeat Step 2~4 for the background image.

Step 7: (Real-time) Lower overall intensity, contrast and saturation; enhance color blue



Experimental results

User can control live2d model smoothly.

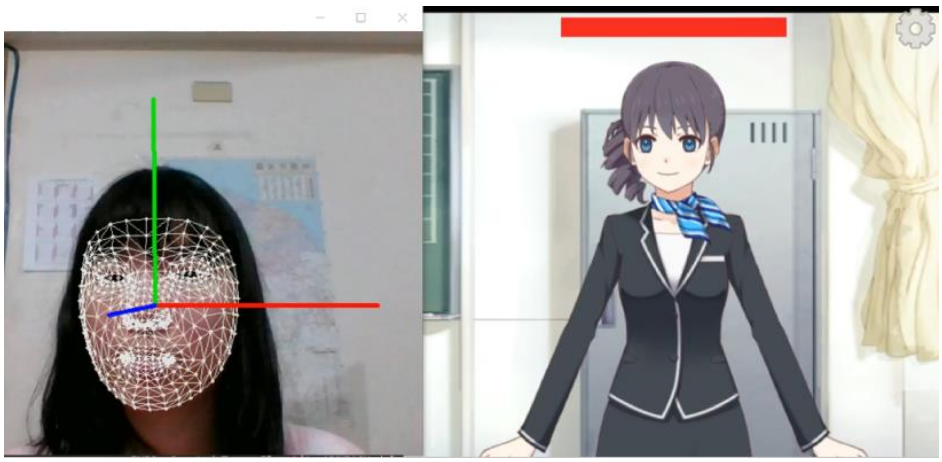
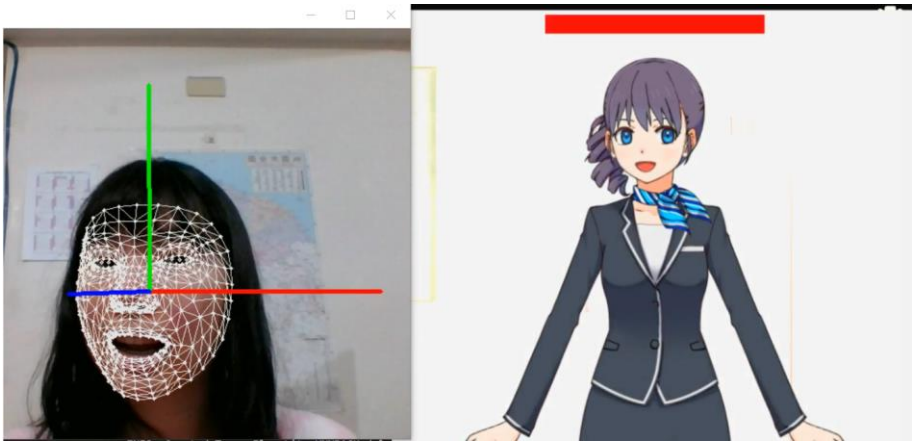
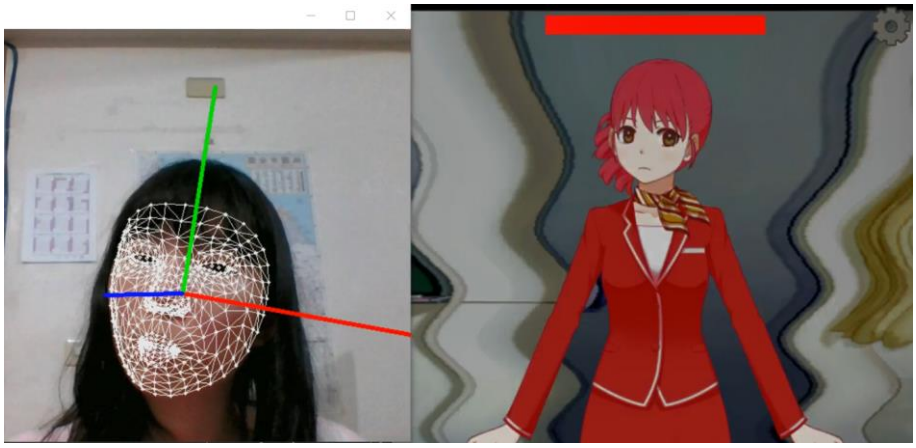


Image style can change through specific expression.

Happy



Angry



Surprise

Closing Eyes

Discussions

Because the landmark detection is unstable, we use Kalman Filter as a point stabilizer to stabilize a 2D point. Otherwise, the model movement will be shaking tremulously. While when I use Kalman Filter in other feature data besides the rotation vector, its performer is not well. Because other features, such as iris movement, are an instantaneous action. If I use Kalman Filter, it will fade the current value, causing my eyes to stop moving. To solve this problem, I use the threshold to replace it. Although the movement of the eyeball finally changed to three values, left, right and middle, it is not obvious because the movement is not large.

As for style transferring part, we realize real-time transferring will lead to frame delay. Thus, we decide to move the most complicated part of the algorithm to preprocessing, and leave simple parts such as image filters to real-time processing. During designing style transferring algorithm, we want to make sure details like eyes pupils and bold lines of the model will not be distorted or eclipsed. So, we decide to extract them before the process, and paste them back after the process.

Reference

Jialing Feng, Zhexiao Guo, Jun Wang and Guo Dan. (2020). Using eye aspect ratio to enhance fast and objective assessment of facial paralysis. *Computational and Mathematical Methods in Medicine.* 1 – 11. https://doi.org/10.1155/2020/1038906

Jaynish P. Vaghela. (2019, Dec 16) *Asleep at the Wheel: A Computer Vision and Deep Learning Approach to Detecting Drowsiness.* Medium. https://medium.com/@jaynishvaghela/driver-drowsiness-detection-using-lstm-network-2966c49a5400

Live2D. (2018). CubismWebSamples [Source code]. https://github.com/Live2D/CubismWebSamples

Mmmmmm44. (2021). VTuber-Python-Unity [Source code]. https://github.com/mmmmmm44/VTuber-Python-Unity