**Geometrical Modified Facial Recognition**

Boyuan(Jack) Chen

Team members: Adam Guo, Marcos Acosta, Nathan Pappalardo

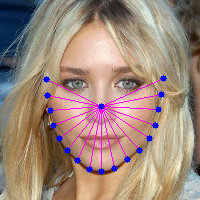
In the spring semester, I recruited three students from 5C to work with me on this project. I was in charge of team discussions, work distribution and research directions. I am aware of every detail in the project.

The principle of our project was to incorporate intuitive shape information into CNN face recognition. For us humans, our recognition system is hierarchical: we first examine the shape and contour of a face, and then we move on to check the details, such as eyes, mouth and nose. Nonetheless, a CNN model treats each pixel equally from the very start. Furthermore, it is a black box. Through forward propagation, we can see that a trained model might not be looking for what we would expect. Also, models can be very sensitive to expanded data, and we usually need reinforcement learning to deal with this problem. However, if we can teach our intuition to the model, then we will not only save training iterations, but also take control over the training direction.

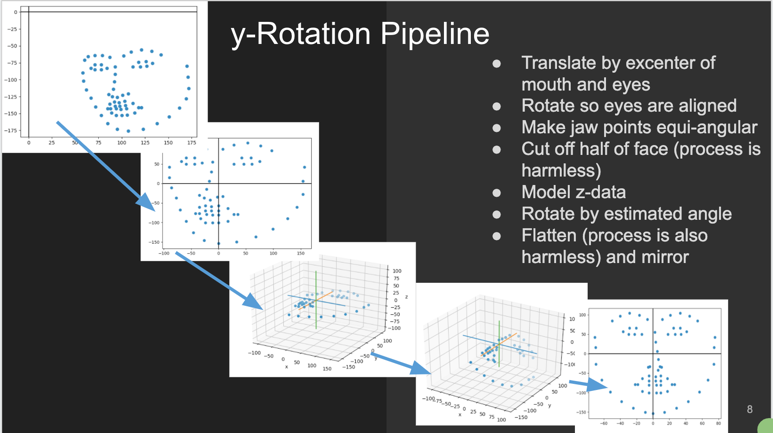
Our ultimate goal is to ensemble the prediction of a shape-recognition model, which takes the shape information of a face and returns the person it belongs to, with a normal CNN model. The desired result is either that the ensembled model has a higher accuracy than CNN alone, or that the ensembled model takes fewer data and training time to give a good prediction than CNN alone.

In the spring semester, we focused on taking the 2D shape information value. We first found dlib, a relatively precise model that extracts 68 edge points, called facial landmarks, on a face. We then decided that not all the points should be used in recognition, because the same person could open their mouth or shut it in different pictures, and so are the eyes. Nonetheless, the position of mouth and eyes are still valuable information. I thus decided to follow your suggestion of taking the weight centers of the two eyes and one mouth to form a triangle. It will be used later on to determine the rotation, but here we used its center to shoot out lines at the borders, and then take the lengths of the 17 lines to form a 17-dimensional vector, as the shape information of the face.

A close up of a person with the mouth open

Description automatically generatedA close up of a person

Description automatically generated



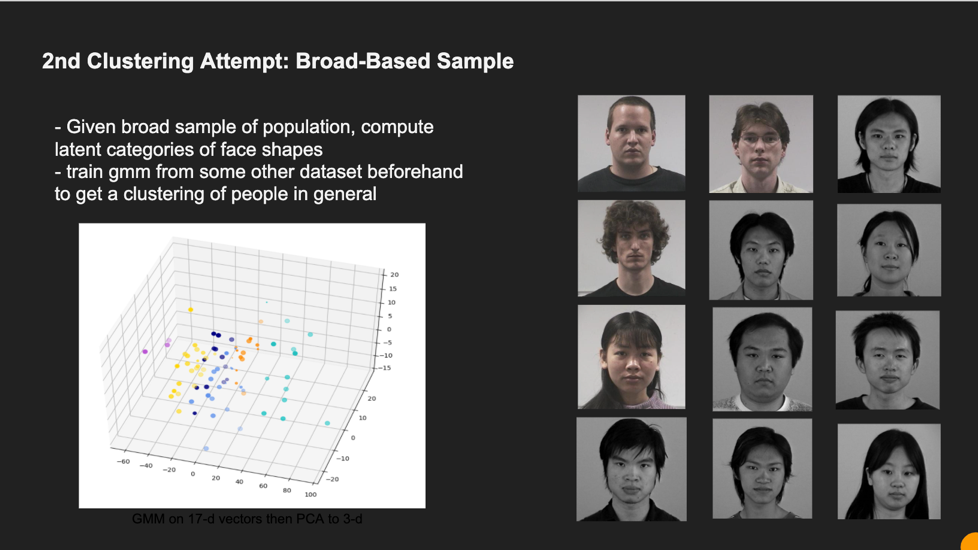
One big problem for 2D shape feature is the rotation. Once the face rotates along y-axis, one side of the face becomes much larger than the other side. We dealt with this problem by getting the rotation value from the eyes-mouth triangle with a ratio between the areas of left and right halves. Then, we rotated the larger side of the face by a certain degree and mirrored it back to get the whole face.

Once the shape value is extracted, I used Gaussian clustering, learned in Math189R, to get the predictions. The result was colorful, but the precision was rather low. It simply divided the points into plates. The precision of this classification is 25.3%.

A close up of a piece of paper

Description automatically generated

So here comes our second clustering attempt - we didn’t apply clustering on the dataset that we wanted to recognize, but some other dataset in the wild having one frontal face for each person. In this way, we get a general distribution of human frontal face, and then we fit the training dataset to this clustering. As a result, what the shape model will tell is that this person belongs to the fifth kind of face, rather than this face is Jack Chen’s face.



Unfortunately, this plan did not work, either. The model predicted all the rotated faces in one class. Our explorations on 2D shape extraction stopped here, as the semester came to an end. We then discussed why it did not work. First, our rotation function was naïve. It lost the information of half of the face, and the coordinate transformation was not robust. Second, the shape vector was only 17 dimensional, but the face shape can have much more information than that.

We then deduced that these two flaws were inevitable in 2D shape extraction, though we spent a very long time thinking about good algorithms for extraction and rotation. Thus, we decided to take your advice to move on to 3D shape extraction. I then searched the various methods, and found the best one to be GANFIT, a model that generates a skin layer from one facial image. Though it had already achieved better result than all its predecessors, we believed that a good shape generation should come from multiple angles. As your anecdote on Sunzi Bingfa goes, it is all clear when she moves. At this step, we are implementing the code and looking for ways to synthesize the output from multiple images.

Unfortunately, we have to pause the project during this hard time, yet we plan to rework on it in the spring semester. We will hopefully publish a paper by next year April.