MATH 189R HM Spring 2020 Final Report

Boyuan Chen **Adam Guo** **Marcos Acosta** **Nathan Pappalardo**

Abstract

The utilization of facial shape as a vital element to improve the efficiency of facial recognition is a fresh topic for research. We propose a posterior-like combination of two predictions generated by a facial shape model and a CNN model, respectively. In such a way, the CNN network can save a lot of computations by only looking at the important facial areas, which are the eyes and the mouth.

1 Introduction

The method of convolutional neural network is widely used in facial recognition nowadays. Early since 2014, research teams have been trying to increase the accuracy by designing more and more complicated structures. Nonetheless, with each small amount of improvement in accuracy, the increase in computation is tremendous. We want to find ways to improve the efficiency of the network while not sacrificing too much of accuracy.

One of the most challenging problem of facial recognition is rotation. While humans can easily recognize a person even after they rotate a little bit, the distribution of color on the pixels could become totally different. To solve this problem, many teams have come up with ways to generate a frontal face first, where they they apply normal CNN to. Nevertheless, the accuracy of frontalization is always questionable, and those methods with high accuracy are hard to train by themselves.

We thus propose an easier way to utilize the extracted shape feature to facilitate CNN recognition - a posterior combination. By training a model that extracts facial shape and set the result as input to a clustering model, we can get a prediction of each person P1, based on the shape alone. Then, combining with the prediction P2, generated by CNN, we will have a reliable prediction that takes into account both the face contour and the details on the face. Note that since P2 does not have to cover the face contours, it can train on a much smaller image and thus improve efficiency.

2 Datasets

We used two datasets consisting of images of faces. The first is VGGFace2 (Cao et al. [2018]), a set of over 3.3 million images of 9000+ people, captured "in the wild" with natural poses, emotions, and lighting. We extracted roughly 500 images each of 10 celebrities, and aim to recognise which person any one of these images belongs to. After we downloaded the dataset, we had to remove any "impurities" in the dataset manually, including pictures with tilted faces, incomplete/non-existing faces, or faces covered by hair or other objects, and even pictures containing faces of a different person.

The second is the Head Pose Image Database from the Pointing'04 ICPR Workshop (hereafter referred to as the Pointing dataset) (N. Gourier [2004]). This database provides 15 sets of images, each of which contains 93 images of a single person at different poses. We use this dataset to analyse facial rotations, as well as to provide a set of clean, consistent images of frontal pose.

Preprint. Work in progress.

3 Extracting Face Shape Information

Here we will mostly introduce our attempts on 2D face shape extraction by far. It is proved to be insufficient in the final result, so we will move on to 3D face shape construction in the summer.

3.1 dlib

Dlib for face detection uses a combination of HOG (Histogram of Oriented Gradient) and Support Vector Machine (SVM). The dlib method of using HOG for semi-rigid objects detection was first published in 2005, and refined for multiple times since then (Kazemi and Sullivan [2014]). A dlib HOG model is trained through feeding in images with the items in interest having labeled boxes on the surroundings. According to the publisher, the human face detector only took 6 seconds to train. In comparison, a Haar Cascade detector could take hours, even days to train. In addition, dlib's structural SVM training algorithm is used in the HOG trainer, efficiently reducing the amount of tedious sampling and training data needed.

3.2 Jaw Shape Vector (Rotation of dlib coordinates)

Clustering facial images involves two major steps: rotating input images to face the camera, and using a clustering algorithm to predict probabilities that an image belongs to a person. We first discuss the image rotation process.

We first fix the z-axis (vertical to the paper) rotation by normalizing the coordinates to the center, and then rotating the dots via linear transformation so that the center of the eyes form a line parallel to the x-axis. The scaling of normalization is dependent on the length of the first dot. We set its distance to the origin as 100 pixels, and all the other dots are multiplied by the same scale. This procedure is described in the first row of Figure 1.

Then, we take the bigger half of the face and put it on the xy-plane. We rotate the face around y-axis for an angle that is proportional to the ratio of areas of the bigger half and the smaller half. This process is described in the second row of Figure 1.

Lastly, we project the rotated bigger-half back to the xy-plane. Then, mirror it to the other side to get a full face. Eventually, we find the center of the face and take the lengths of the 17 shoots to form a vector that represents this face shape.

3.3 Next Step: 3D Face Shape Reconstruction

As will be discussed later, the result is not satisfactory because the frontal shape generating function is naive. We will focus on searching for existing methods in generating 3D face shapes. Such is a hot topic in recent years, and there are good models such as Position Map Regression Network (Feng et al. [2018], GANFIT(Gecer et al. [2019]), Unsupervised Training for 3D Morphable Model Regression (Genova et al. [2018]), etc. The ideal method is one that takes multiple inputs from the same person of different angles, so that the 3D-reconstructed mesh should be more accurate.

4 Clustering facial vectors

For this section, our goal is to obtain geometric information from facial images to compute a posterior based on facial shape, which can be multiplied to priors generated by facial features to yield a final prediction of who the given face belongs to. Given a sample of facial images, different types of facial shapes roughly emerge. That is, we can categorise people by how similar their facial shapes are (figure 2). Hence, we used unsupervised probabilistic clustering to learn clusters of face shapes from some input data and compute a probability that any given face belongs to some cluster. This probability gives an effective likelihood that an input face matches some person with a face shape in that cluster, which we can use as our posterior.

We chose to use a Gaussian mixture model (GMM), which consists of a weighted sum of K multivariate normal distributions (Murphy [2012]):

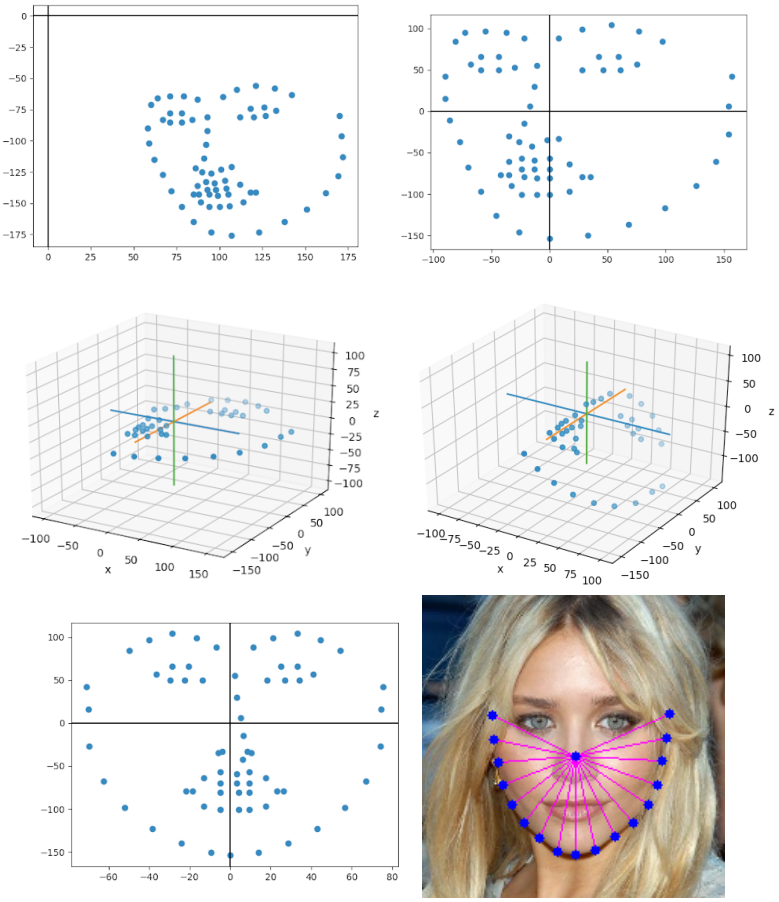

Figure 1: Example of facial shape clusters among 14 people

$$p(x_i : \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i : \mu_k, \Sigma_k)$$

The parameters of each Gaussian and their associated weights can be fitted using expectation maximisation, giving us probabilistic clusters for facial shapes. The Python package `scikit-learn` includes a native GMM model trainer, `GaussianMixture`, which we used to train on the 17-dimensional vectors that represent the facial shape of each image.

4.1 Using GMM clustering for recognition

Our initial approach to clustering faces was to determine whether or not many frontal images of a particular person were enough to discern a distinct shape cluster corresponding to their real face. We manually sorted through images of 5 celebrities, picked images that were frontal-facing (centered along the x, y, and z axes), computed their shape vectors, and clustered them using GMM ($K = 5$).

The 3-dimensional PCA visualisation (figure 3) shows that the clustering of the original data is not well-defined. The GMM fitted 5 naive clusters by effectively slicing the data into 5 chunks, yielding an accuracy of 25% (around random) when predicting which person an image corresponded to. This test confirmed that there is significant variability in the shape vectors of each person depending on minute changes in the angle at which the image is taken. Manually selecting many frontal images of a single person is insufficient in eliminating angle variations.

Figure 2: Example of facial shape clusters among 14 people

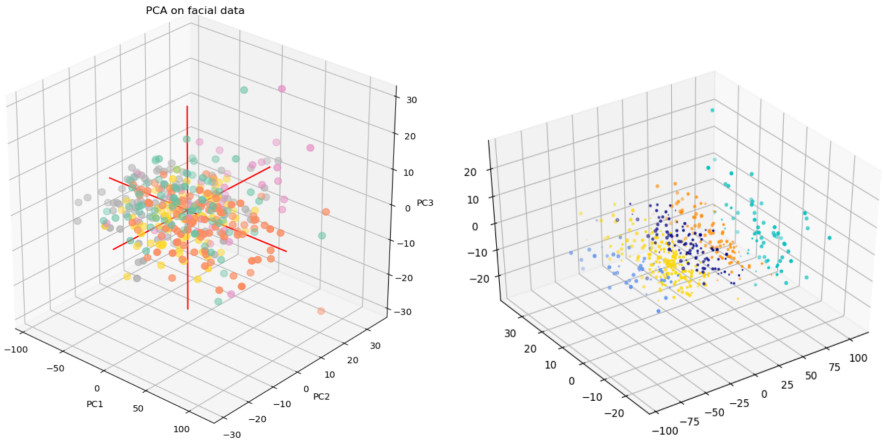

Figure 3: Original and clustered data, shown on 3D PCA of facial vectors

4.2 Using GMM clustering on broad-based sample

The above clustering attempt revealed the inadequacy of geometric information in predicting the identity of an image. This led us to an alternate approach for analysing geometric facial information: instead of training a clustering model on the faces we wish to recognise, we decided to train a model on a broad-based sample of the general human population. The intuition is that by recognising latent face types in the broader population, the model can probabilistically group people in the test database. Then, given an image to infer on, the model can evaluate the likelihood that the image belongs to each cluster, and thereby the likelihood that the image belongs to each person in the test database.

The Pointing dataset provides images of many individuals taken in a standard frontal pose. We trained the same Gaussian mixture model as above on the new set of data, producing the clusters shown in figure 4. Note that the overall structure is similar to figure 3, the difference being that since in this case each vector corresponds to a unique person, we expect a diffusely scattered plot with evenly divided clusters, rather than obvious clusters corresponding to multiple images of a single person.

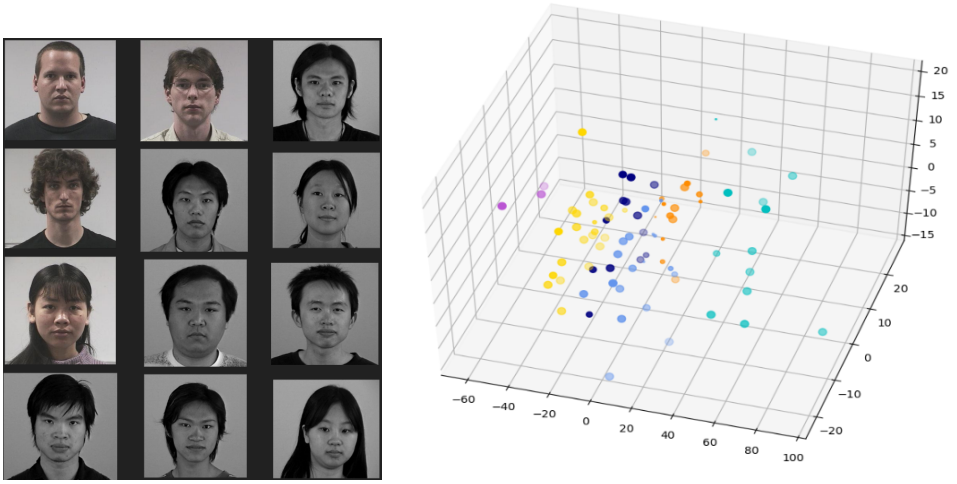

Figure 4: Left: sample of Pointing dataset, right: PCA 3D projection of facial vectors

4.3 Computing posterior

Using the trained Gaussian mixture model, we now compute the posterior: the probability that an input face belongs to any person in the dataset. Note that we are aiming to recognise individual celebrites in VGGFace2, not individuals in the Pointing dataset.

Suppose we have a dataset of n people, where the ith person has m_i images. Let $\mathbf{v}_j^{(i)} \in \mathbb{R}^{17}$ be the vector generated by `dlib` that represents the jawline of the jth image corresponding to person i. Let k be the number of components (i.e. clusters) in the GMM. Let $Z : \mathbb{R}^{17} \to \mathbb{R}^k$ be a function where $Z(\mathbf{x})$ is the probability vector returned by the GMM given input vector $\mathbf{x}$.

We construct a matrix $M \in \mathbf{M}_{n \times k}$ as follows:

$$M = \begin{bmatrix} \frac{1}{m_1} \sum_{j=1}^{m_1} Z(\mathbf{v}_j^{(1)})^T \\ \frac{1}{m_2} \sum_{j=1}^{m_2} Z(\mathbf{v}_j^{(2)})^T \\ \vdots \\ \frac{1}{m_n} \sum_{j=1}^{m_n} Z(\mathbf{v}_j^{(n)})^T \end{bmatrix}$$

The (i, j) element of M is the likelihood that person i belongs to cluster j, which we interpret as the weight of cluster j in predicting person i. We divide by the number of images that person has to prevent biasing towards people with more images in the dataset. Given an input vector $\mathbf{x} \in \mathbb{R}^{17}$ we wish to predict, we use the model to infer $\mathbf{p} = M(\mathbf{x}) \in \mathbb{R}^k$, then compute

$$\mathbf{y} = M\mathbf{p} \in \mathbb{R}^n$$

Normalising $\mathbf{y}$ such that the entries sum up to 1, we have the posterior:

$$\tilde{\mathbf{y}} = \frac{1}{\sum_{i=1}^{n} y_i} \mathbf{y}$$

$\tilde{\mathbf{y}}$ gives the probability that the input $\mathbf{x}$ is an image of each of the n people. Specifically, $\tilde{y}_i$ gives the probability that the input is an image of the ith person.

4.4 Result

Running the full posterior computation pipeline, we experienced more issues related to noise in the facial vectors. We trained a GMM M_f on the Pointing dataset of frontal images, using $k = 6$

components, and fed celebrity images from the VGGFace2 dataset to predict on. However, regardless of which image is given as input, M_f always returned a prediction of $[0 \quad 0 \quad 0 \quad 0 \quad 1.0 \quad 0]$. The model is essentially 100% confident that every input image belongs to the same cluster, with zero probability of belonging to any other cluster. As a result, the posterior computation yielded a uniform probability that any input image belongs to each person.

As a sanity check, we repeated the process using another GMM M_c, trained on VGGFace2 itself (akin to Section 4.1). While the results this time made more sense and did not predict that every image belonged to the same cluster, this approach suffered from the same problems faced in Section 4.1, which were that the predictions were very inaccurate (25%).

Our posterior computation method is quite straightforward and seemed to be working. Instead, the problem lies with the results predicted by the GMM. Examining the reconstructed frontal facial vectors that were generated in Section 3.2, we concluded that the coordinate transformation process distorts each face in a consistent manner by stretching each face horizontally and also creating a perfectly symmetrical face across the vertical axis. While these differences were not apparent to human viewers, they could very well have been perceptible and consistent enough for M_f, which was originally trained on non-transformed faces, to discern a common structure and assign them all to the same cluster, regardless of what the original face was.

This suggests that our clustering approach does not go well with our coordinate transformation process. To compute a usable posterior, we need to re-evaluate our strategy these two steps of the pipeline.

5 Future steps

Though our result is far from satisfying, we hold faith in the general structure and are quite certain about our next steps. The clustering method appears functional and thus should be largely kept. However, there are some important changes that can be made. First, the clustering approach assumes that there are discrete categories that faces can be grouped into. Our data reveals that this is not really true: faces are largely continuous structures, and while as humans we can say that some faces have distinct shapes, there is no clear boundary between these categories. Hence, instead of a clustering algorithm, we can move to computing weights based on the raw distances between vectors (with an appropriate distance function), allowing us to bypass the problems we have had with Gaussian mixture models.

Generating face shape vectors has significant room for improvement. As mentioned in Section 3.3, we will search for more recent approaches in 3D face shape construction. The most popular models take only 1 image as the input. Though good enough on its own standard, we aim to find ways to incorporate multiple images of one person's face to generate a more accurate 3D mesh.

Furthermore, recent advances have also provided a better possibility for generating the second possibility via CNN. Several papers have offered ways to generate frontal face images via GAN. Jian Zhao finished the whole pipeline of using the frontal face generated by GAN for CNN facial recognition, and is proved to have good result (Zhao et al. [2018]). We would like to use these works as methods to generate the second probability in the posterior.

References

Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

J. L. Crowley N. Gourier, D. Hall. Estimating face orientation from robust detection of salient facial features. In *Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK*, 2004.

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network, 2018.

Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction, 2019.

Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression, 2018.

Kevin P. Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, Cambridge, MA, 2012. ISBN 9780262018029.

Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, Shuicheng Yan, and Jiashi Feng. Towards pose invariant face recognition in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.