Eric Fischer

PCA, Autoencoder, and FLD for Analyzing Human Faces

The human face is a very important pattern and has been extensively studied in the past 30 years in vision, graphics, and human computer interaction, for tasks like recognition, human identification, expression, animation, etc. An important step for these tasks is to extract effective representations from the face images. This project compares three types of representation in the context of dimension reduction: two generative methods, Principal Component Analysis (linear) and Autoencoder (non-linear), and one discriminative method, Fisher Linear Discriminants.

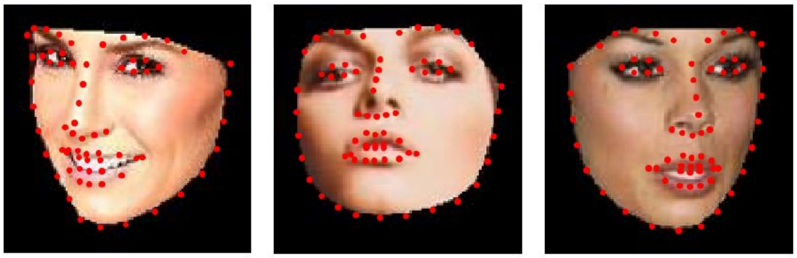


Figure 1: Example faces with 68 landmarks from CelebA. The data set contains 1000 images from CelebA, and they are cropped to 128 × 128 pixels by the OpenFace. These faces have different colors, illuminations, identities, viewing angles, shapes, and expressions.

Principal Component Analysis (PCA)

Part 1: Reconstruct images by appearance

First we compute the mean and first K = 50 eigen-faces of the 800 *training* images with no landmark alignment. We use these eigen-faces to reconstruct the 200 *test* faces.

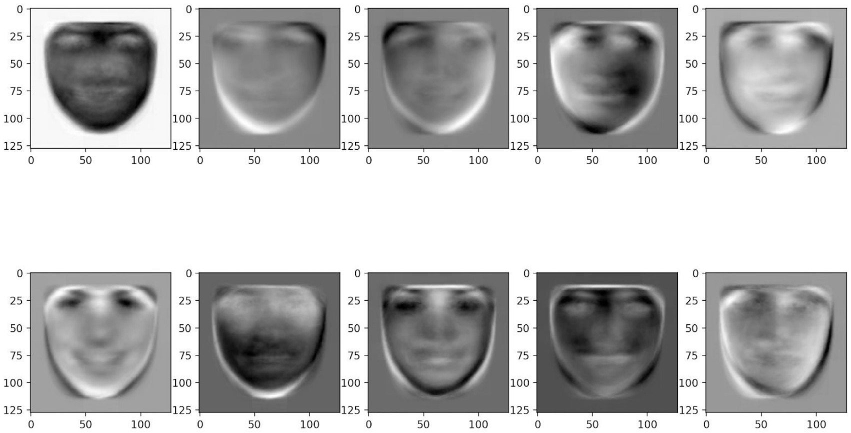


Figure 1: First 10 eigen-faces for the training images (with no landmark alignment)

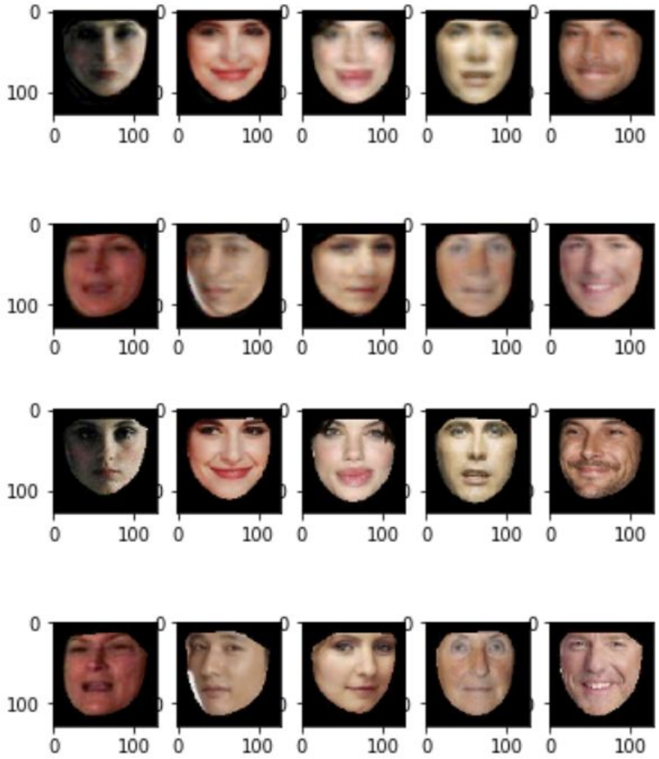


Figure 2: 10 reconstructed *test* images and the corresponding original images

We can plot the total reconstruction error (squared intensity difference between the reconstructed images and their original ones) per pixel. To do this, we sum the errors over all pixels for all images and normalize (i.e. divide) by the number of pixels per image and the number of images. This lends the average PCA reconstruction error (without landmark alignment) per pixel.

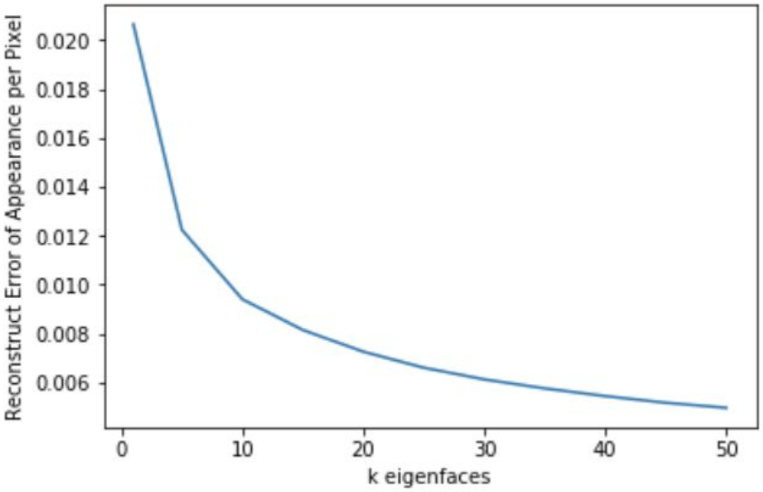


Figure 3: Total reconstruction error per pixel (for the HSV V-channel) over the number of eigen-faces k = 1, 5, 10, 15, …, 50

Part 2: Reconstruct landmarks by geometry

Here we compute the mean and first K = 50 eigen-warpings of the 800 training landmarks. We use these eigen-warpings to reconstruct the 200 test landmarks.

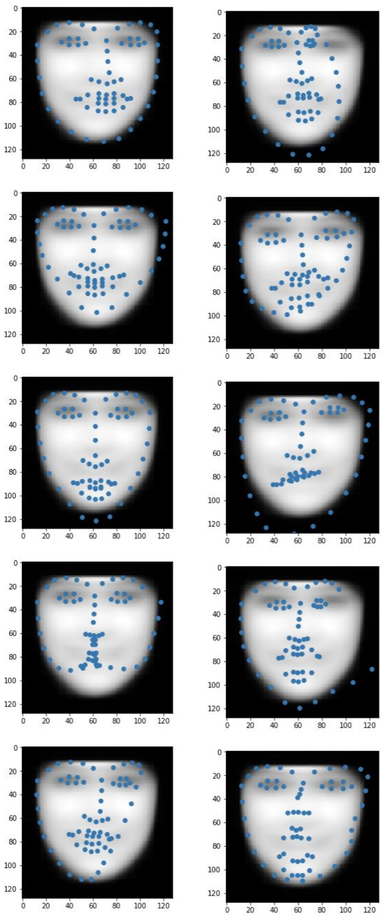


Figure 4: First 10 eigen-warpings for the training landmarks

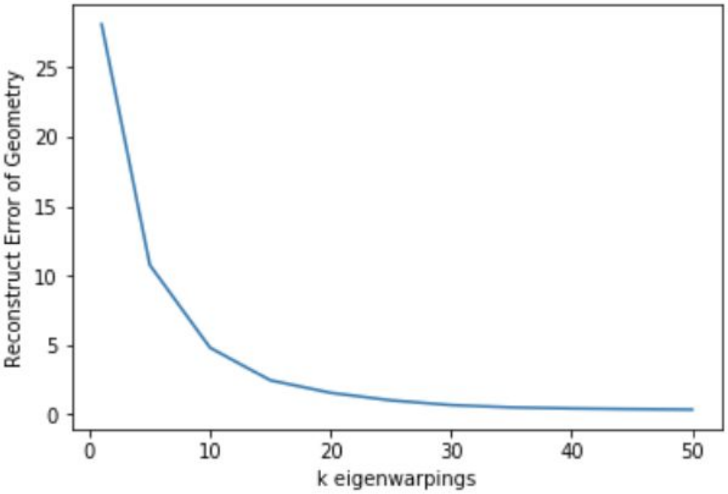


Figure 5: Reconstruction error (distance) over number of eigen-warpings k = 1, 5, 10, 15, …, 50

Part 3: Reconstruct images by appearance and geometry

This part combines Parts 1 and 2 to reconstruct the images by appearance and geometry, using the previously calculated eigen-faces and eigen-warpings for the training images and landmarks. The reconstruction is based on the top 50 eigen-faces and the top 10 eigen-warpings.

First we align the training images by warping them into the mean position, and then compute the eigenfaces from these *aligned* images. Note the difference from Part 1: in this case, the eigenfaces are computed from the training images after mean alignment.

Now for each test image, we project its landmarks onto the top 10 eigen-warpings. This lends the reconstructed test landmarks (for which we lose some geometric precision from reconstruction). Next, we warp the images to the mean position and project them onto the top 50 eigen-faces. This lends the reconstructed images (for which we lose some appearance accuracy from reconstruction) at the mean position.

Lastly, we warp these reconstructed images to the positions of the reconstructed test landmarks. A reconstructed image is thus constructed from 60 scalar projections, 50 from its projections onto the top 50 eigen-faces and 10 from its landmark projections onto the top 10 eigen-warpings.

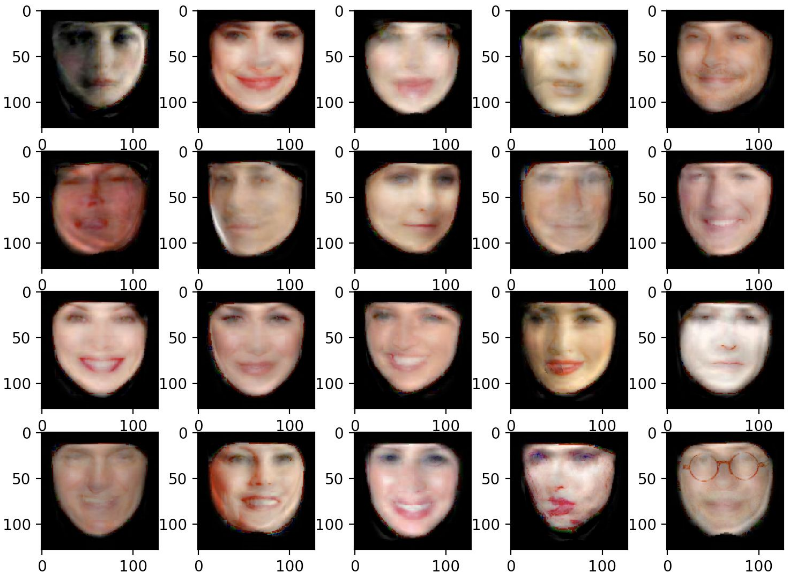


Figure 6: 20 reconstructed faces based on the top 50 eigen-faces and top 10 eigen-warpings

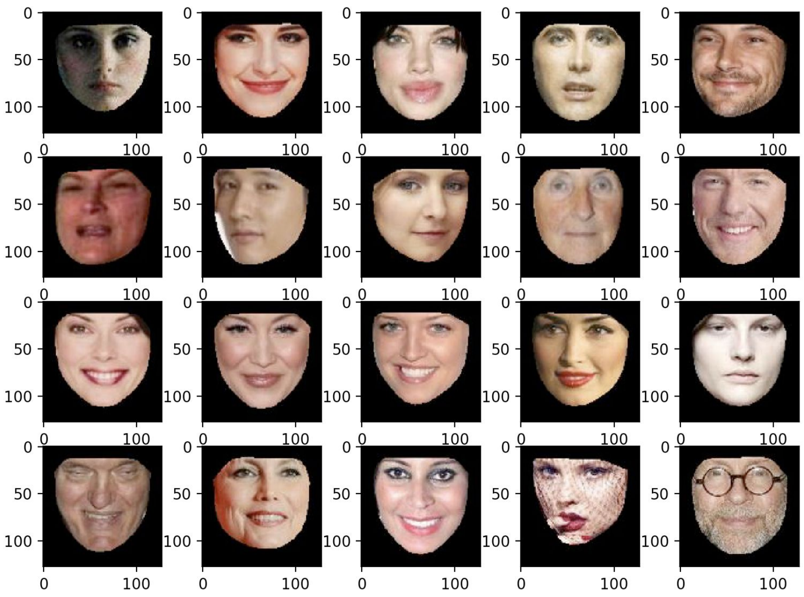


Figure 7: Corresponding original faces for the 20 reconstructed faces

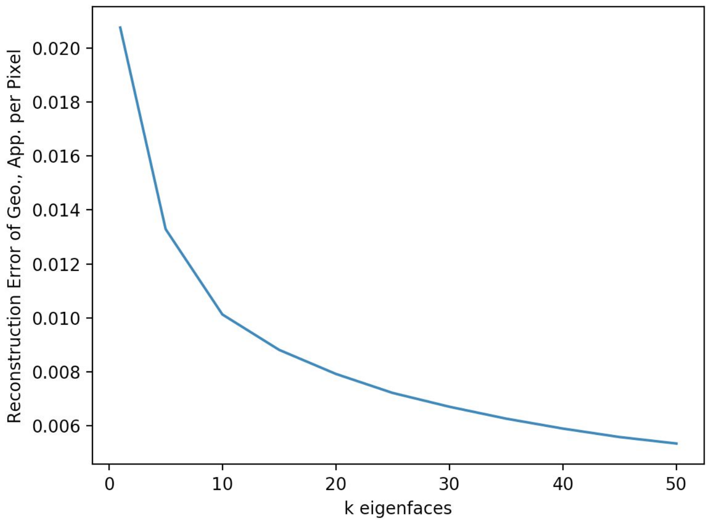
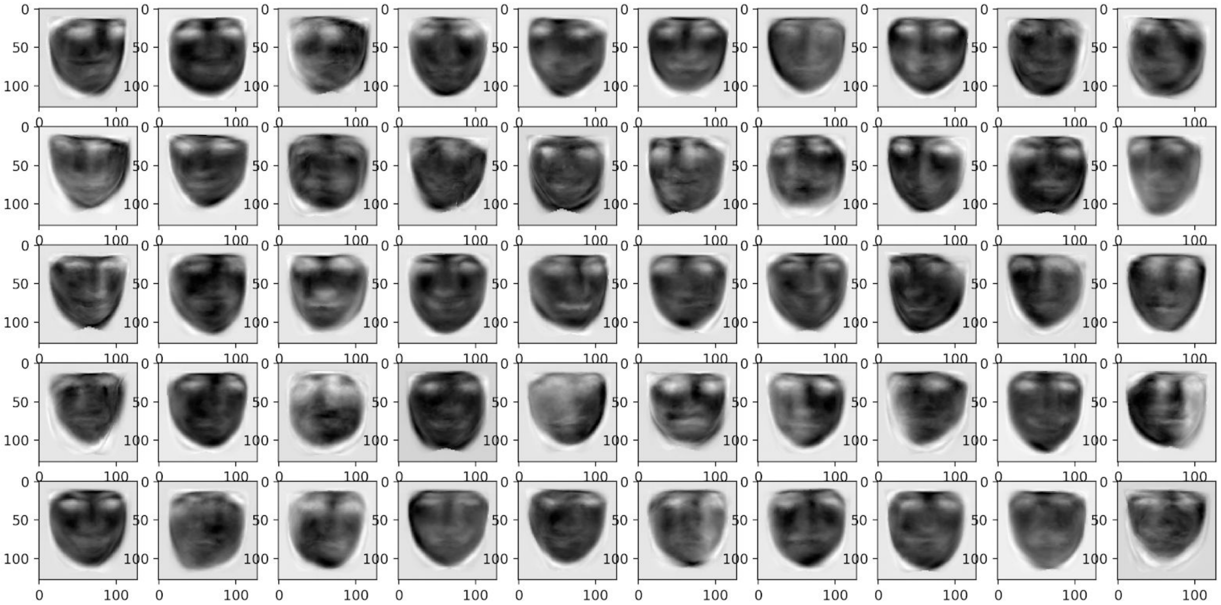


Figure 8: Reconstruction errors per pixel against number of eigen-faces k = 1, 5, 10, 15, …, 50

Part 4: Synthesize Random Faces

In this part, we synthesize random faces by random sampling of the landmarks and appearance separately. The eigen-faces and eigen-warpings can be seen as basis functions for the generated faces, while we need to sample the coefficients at each eigen-axis (i.e. eigenvector). Each eigen-axis has its own unit, $\sqrt{\lambda_i}$, the square root of the i-th eigenvalue belonging to the i-th eigen-axis.

For the random sampling of coefficients at each eigen-axis, we sample b_i, or the scalar projection parameter, from a normal distribution with a mean of 0 and a variance of λ_i (which corresponds to a standard deviation of $\sqrt{\lambda_i}$). After obtaining the parameters, we reconstruct 50 images which are shown below in grayscale.



Autoencoder

PCA reconstructs an image linearly with an orthonormal basis using eigenvectors learned from the covariance matrix of the training data. Autoencoder is a nonlinear extension of PCA, which extracts features and reconstructs data using a deep neural network. An autoencoder learns to compress data from the input layer into a short code and then uncompresses that code into something that closely matches the original data. This forces the autoencoder to engage in dimensionality reduction. The figure below shows an illustration of an autoencoder with convolutional structures.

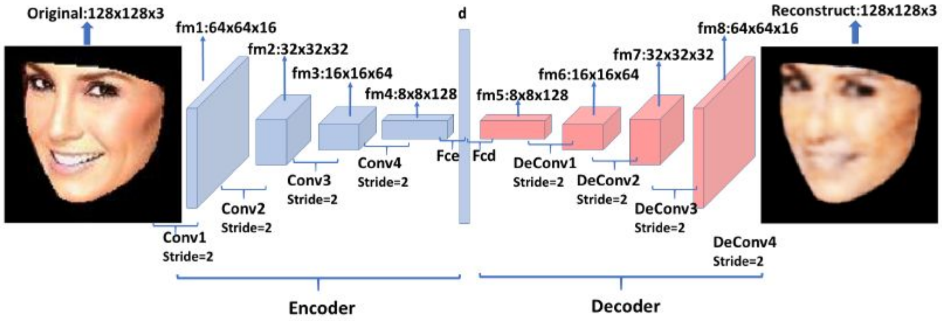
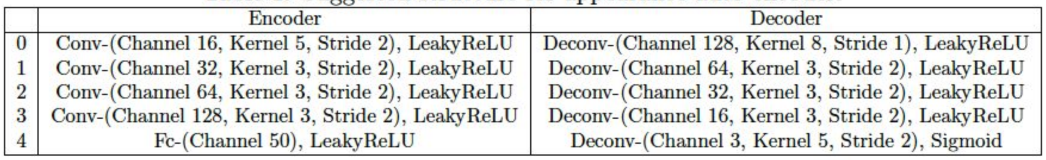


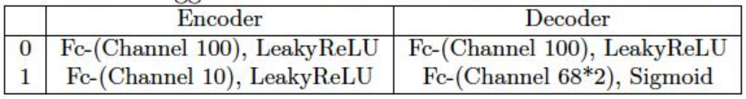
Figure 2: Illustration of Auto-encoder with convolutional architectures. An auto-encoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.

Table 1: Suggested structure for appearance auto-encoder.



	Encoder	Decoder
0	Conv-(Channel 16, Kernel 5, Stride 2), LeakyReLU	Deconv-(Channel 128, Kernel 8, Stride 1), LeakyReLU
1	Conv-(Channel 32, Kernel 3, Stride 2), LeakyReLU	Deconv-(Channel 64, Kernel 3, Stride 2), LeakyReLU
2	Conv-(Channel 64, Kernel 3, Stride 2), LeakyReLU	Deconv-(Channel 32, Kernel 3, Stride 2), LeakyReLU
3	Conv-(Channel 128, Kernel 3, Stride 2), LeakyReLU	Deconv-(Channel 16, Kernel 3, Stride 2), LeakyReLU
4	Fc-(Channel 50), LeakyReLU	Deconv-(Channel 3, Kernel 5, Stride 2), Sigmoid

Table 2: Suggested structure for landmark auto-encoder.



	Encoder	Decoder
0	Fc-(Channel 100), LeakyReLU	Fc-(Channel 100), LeakyReLU
1	Fc-(Channel 10), LeakyReLU	Fc-(Channel 68*2), Sigmoid

Reperforming the experiments in the PCA analysis, landmarks can be reconstructed and generated by an auto-encoder with a fully-connected architecture, while two-dimensional face images can be reconstructed and generated by an auto-encoder with a convolutional architecture. Below are 20 reconstructed faces using the autoencoder and their corresponding original faces.

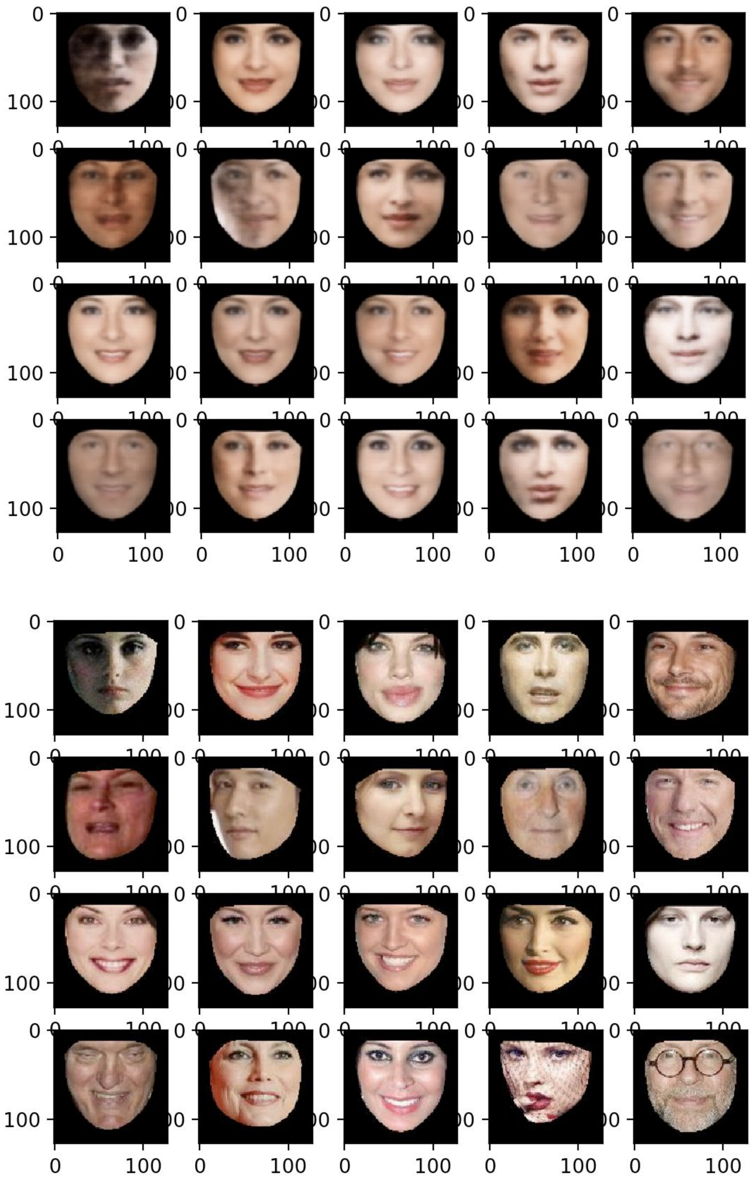


Figure 1: 20 reconstructed faces based on landmark and appearance autoencoders and the corresponding original faces

The autoencoder loss for reconstructing the geometry of test landmarks is 0.0008, and the loss for reconstructing the appearance of warped test images is 0.0068.

Now we can try an interpolation experiment. We choose the first 4 dimensions of the latent variables of appearance that the maximal variance, and below are the interpolation results on each dimension while keeping the other dimensions fixed.

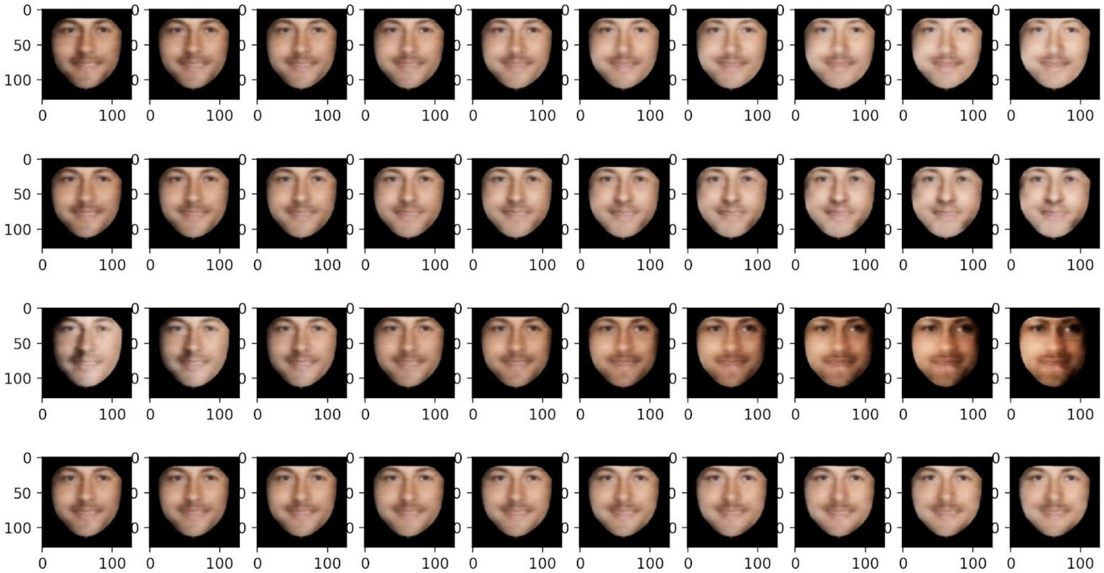
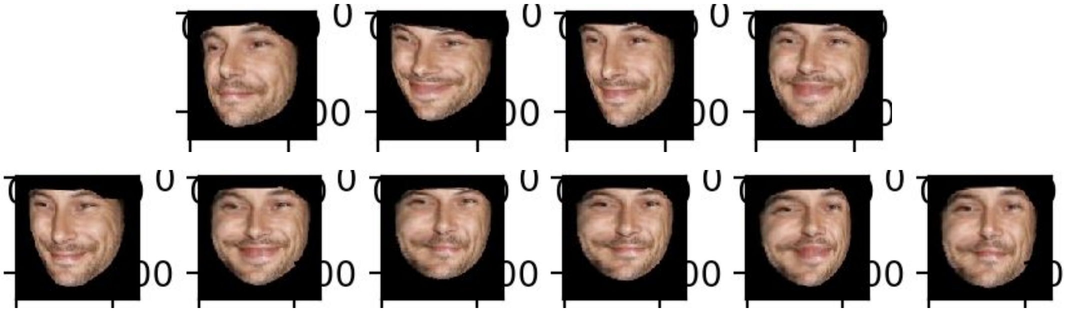


Figure 4: Interpolation results of one face for the 4 dimensions of the latent variables of **appearance** with maximal variance, while keeping other dimensions fixed

Now we can choose the first 2 dimensions of the latent variables of landmarks that have the maximal variance and obtain the interpolation results on each dimension while keeping the other dimensions fixed. Below are the results for warping a chosen face by the generated landmarks.



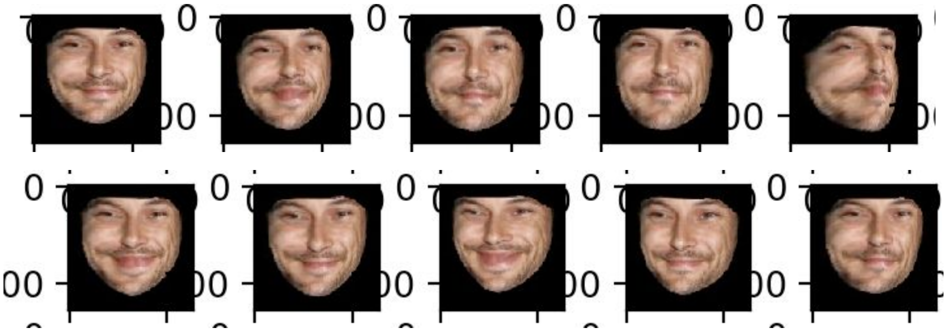


Figure 5: Interpolation results of one face for the 2 dimensions of the latent variables of **landmarks** with maximal variance, while keeping other dimensions fixed

Fisher Linear Discriminants (FLD)

In this part, we can experiment with Fisher face linear decision boundaries to discriminate between testing faces. We will try two different approaches, but in both we find the Fisher face that distinguishes male from female using a *training* set of 800 faces and then test it on the 200 *testing* faces. The Fisher face line has the following form to distinguish between males and females.

$$\omega^T x + \omega_0 > 0 \rightarrow \text{Male}$$

$$\omega^T x + \omega_0 < 0 \rightarrow \text{Female}$$

In the first approach, for less computationally intensive calculations, we represent each image with a 60-dimensional vector derived from PCA, 50 values from the scalar projections of the image onto the top 50 eigen-faces and 10 values from the scalar projections of its landmarks onto the top 10 eigen-warpings. Using this representation, we calculate the male and female class scatter matrices, the within-class scatter matrix, and the Fisher face or linear decision boundary that aims to separate the male and female classes. Below is a scatter plot for the Fisher face with respect to discriminating between males (blue) and females (red) in the 200 testing faces. Each image is represented by one value (i.e. 1-D, the result of plugging in the appropriate values into the Fisher face line equation above. The reported error for the 200 testing faces is 8.50% with a threshold ω_0 set to -.001. The threshold, c in the following equation, was derived as the hyperplane between projections of the two means for male and female faces:

$$c = \vec{w} \cdot \frac{1}{2}(\vec{\mu}_0 + \vec{\mu}_1)$$

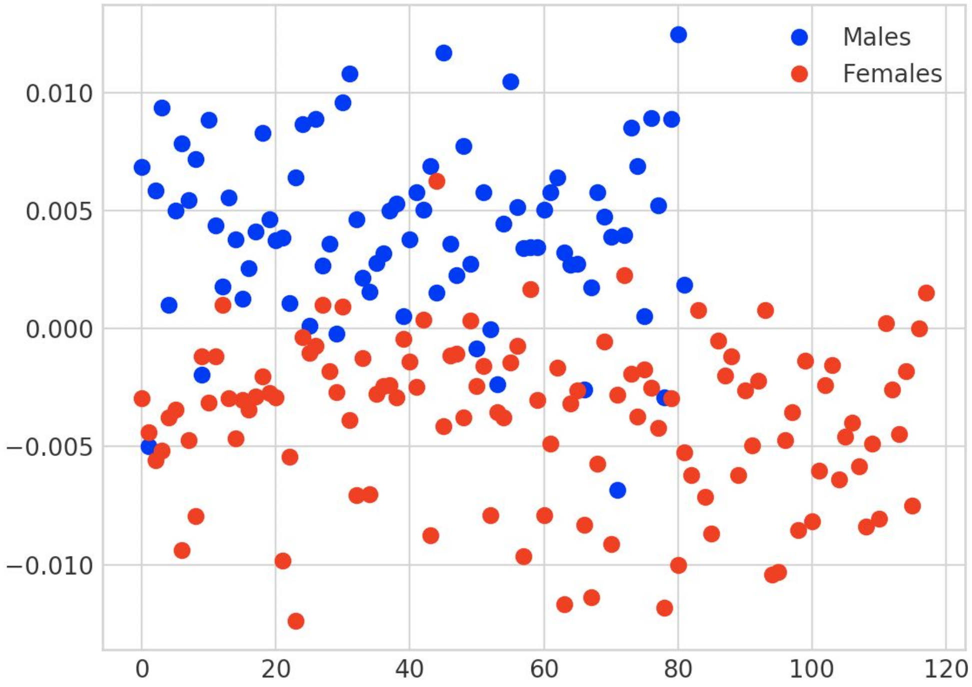


Figure 1: Scatter plot for the 1-D projected points of testing faces

In the second approach, we again represent each image with 60 dimensions, yet we will compute two Fisher face lines for geometric and appearance features, respectively. One Fisher face line is based on the 10 values for the scalar projection of each image's landmarks onto the top 10 eigen-warpings, and the second Fisher face line is based on the 50 values for each image's scalar projection onto the top 50 eigen-faces. Note that in this second approach, we also align images to the mean position before projecting them onto the top 50 eigen-faces.

We will then project the image data points to the 2-D feature space learned by the Fisher faces to visualize how separable the points are in comparison with our first analysis. Below is a scatter plot for the two Fisher faces with respect to discriminating between males and females in the 200 testing faces. Each image is now represented by two values in this approach, the result of plugging in the appropriate values to two Fisher face line equations.

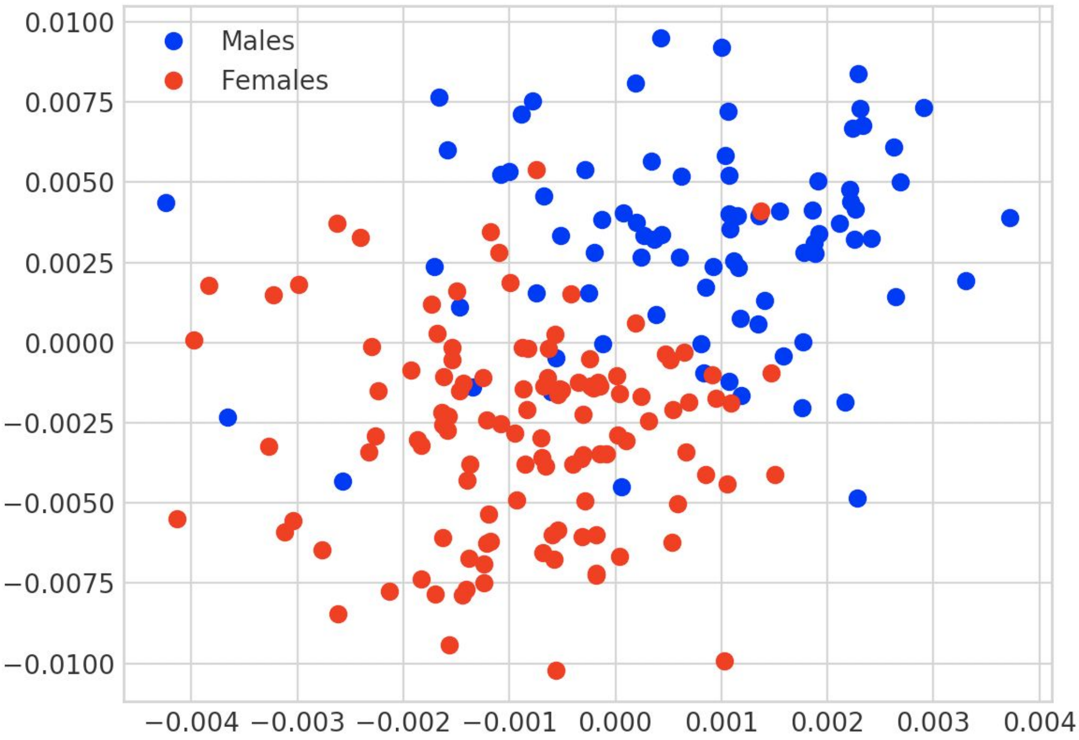


Figure 2: Scatter plot for the 2-D projected points of testing faces