EECS 4404 Assignment 3

Anton Sitkovets

(1) $p(y|\theta) = \sum_{j=1}^{k} \pi_i N(y|\mu_i, \Sigma_i)$ $\theta = \{(\pi_i, \mu_i, \Sigma_i)\}_{j=1}^{K}$

$E(\theta) = -\sum_{i=1}^{N} \log p(y^{(i)}|\theta)$

a) $E(\theta) = -\sum_{i=1}^{N} \log\left(\sum_{j=1}^{K} \pi_j N(y|\mu_j, \Sigma_j)\right)$

$$\frac{\partial E(\theta)}{\partial \mu_j} = -\sum_{i=1}^{N} \frac{\pi_j \left(\frac{\partial N(y|\mu_j, \Sigma_j)}{\partial \mu_j}\right)}{\sum_{j'=1}^{K} \pi_{j'} N(y|\mu_{j'}, \Sigma_{j'})}$$

$j' \rightarrow$

$$= \sum_{i=1}^{N} \frac{\pi_j \left(N(y|\mu_j, \Sigma_j)\Sigma_j^{-1}(y-\mu_j)\right)}{\sum_{j'=1}^{K} \pi_{j'} N(y|\mu_j, \Sigma_j)}$$

Eqn (393) in the Matrix Cookbook says the derivative of a Multivariable Gaussian is

$$\frac{\partial \ln p(s)}{\partial \mu_j} = \frac{\rho_j N_s(\mu_j, \Sigma_j)\left[\Sigma_j^{-1}(s-\mu_j)\right]}{\sum_k \rho_k N_s(\mu_k, \Sigma_k)}$$

b) $$\frac{\partial E(\theta)}{\partial \pi_j} = -\sum_{i=1}^{N} \frac{k N(y|\mu_j, \Sigma_j)}{\sum_{i=1}^{K} \pi_i N(y|\mu_i, \Sigma_i)}$$

by eqn (391) in the matrix cookbook

$$\frac{\partial \ln p(s)}{\rho_j} = \frac{\rho_j N_s(\mu_j, \Sigma_j)}{\sum_k \rho_k N_s(\mu_k, \Sigma_k)} \frac{1}{\rho_j}$$

c) Sub $\pi_j = \frac{\exp(w_j)}{\sum_{k=1}^{K} \exp(w_j)}$ into the partial derivative

$$\frac{\partial \pi_j}{\partial w_k} = \frac{\partial \frac{\exp(w_j)}{\sum_{k=1}^{K} \exp(w_k)}}{\partial w_k} = \frac{\exp(w_j)\sum_{k=1}^{K}\exp(w_k) - \exp(w_j)\exp(w_j)}{\left(\sum_{k=1}^{K} \exp(w_k)\right)^2}$$

$$= \frac{\exp(w_j)}{\sum_{k=1}^{K} \exp(w_k)} - \frac{\exp(w_j)\exp(w_j)}{\sum_{k=1}^{K}\exp(w_k) \cdot \sum_{k=1}^{K}\exp(w_k)}$$

$$= \pi_j - \pi_j^2 = \pi_j(1-\pi_j) \text{ when } j=k$$

When $j \neq k$

$$\frac{\partial \pi_j}{\partial w_k} = -\frac{\exp(w_j)\exp(w_k)}{\left(\sum_{k=1}^{K} \exp(w_k)\right)^2} = -\pi_j \pi_k$$

d) Sub values from (b) and (c) back into $E(\theta)$

$$E(\theta) = -\sum_{i=1}^{N} \frac{\frac{\partial \left(\frac{\exp(w_j)}{\sum_{k=1}^{K} \exp(w_k)} N(y|\mu_k, \Sigma_k) \right)}{\partial w_k}}{\sum_{j=1}^{K} \frac{\exp(w_j)}{\sum_{k=1}^{K}\exp(w_k)} N(y|\mu_k, \Sigma_k)}$$

$\left(\frac{\exp(w_j)}{\sum_{k=1}^{K} \exp(w_k)} N(y|\mu_k, \Sigma_k) \right) = \underline{\underline{A}}$

if $j=k$ then $\underline{\underline{A}} = N(y|\mu_k, \Sigma_k)\pi_j(1-\pi_j)$

$$\frac{\partial E(\theta)}{\partial w_k} = -\sum_{i=1}^{N} \frac{\pi_j(1-\pi_j) N(y|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_j N(y|\mu_j, \Sigma_j)}$$

if $j \neq k$ then $\underline{\underline{A}} = -N(y|\mu_k, \Sigma_k)\pi_j\pi_k$

$$\frac{\partial E(\theta)}{\partial w_k} = \sum_{i=1}^{N} \frac{\pi_j \pi_k N(y|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_j N(y|\mu_j, \Sigma_j)}$$

Scanned by CamScanner

(2)

a) $E[y] = \int y \sum_{i=1}^{K} \pi_i \mathcal{N}(y|\mu_i, \Sigma_i)\, dy$

$= \int \sum_{i=1}^{K} y \pi_i \mathcal{N}(y|\mu_i, \Sigma_i)\, dy$

$= \sum_{i=1}^{K} \pi_i \underbrace{\int y \mathcal{N}(y|\mu_i, \Sigma_i)\, dy}_{\text{mean } \mu_i}$

$= \sum_{i=1}^{K} \pi_i \mu_i$

b) $Cov(y) = E[(y-E[y])(y-E[y])^T]$

$= \int (y-E[y])(y-E[y])^T \sum_{i=1}^{K} \pi_i \mathcal{N}(y|\mu_i, \Sigma_i)\, dy$

$= \sum_{i=1}^{K} \pi_i \underbrace{\int (y-E[y])(y-E[y])^T \mathcal{N}(y|\mu_i, \Sigma_i)\, dy}_{\text{covariance } \Sigma_i}$

$= \sum_{i=1}^{K} \pi_i \Sigma_i = E[yy^T] - E[y]E[y]^T$

$\cancel{E[yy^T] - E[y]E}$

$\therefore E[yy^T] = \sum_{i=1}^{K} \pi_i \Sigma_i + E[y]E[y]^T = \sum_{i=1}^{K} \pi_i \Sigma_i + \left(\sum_{i=1}^{K} \pi_i \mu_i\right)\left(\sum_{i=1}^{K} \pi_i \mu_i\right)^T$

$= \sum_{i=1}^{K} \pi_i (\Sigma_i + \mu_i \mu_i^T)$

Scanned by CamScanner

EECS 4404
Assignment 3
Anton Sitkovets

Step 1

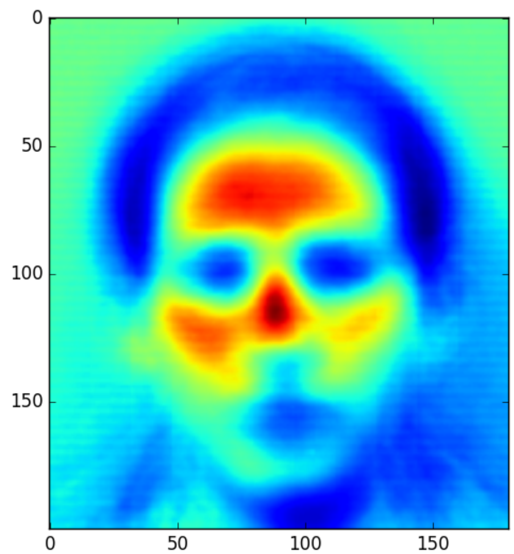


Figure 1. Mean Face of Dataset

Step 2

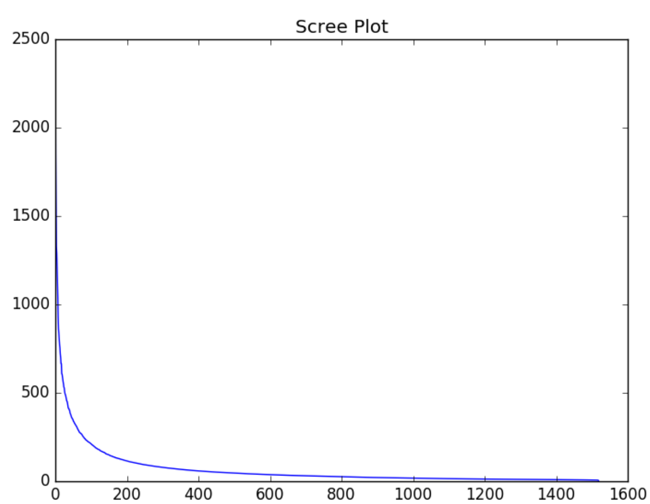


Figure 2. Scree Plot

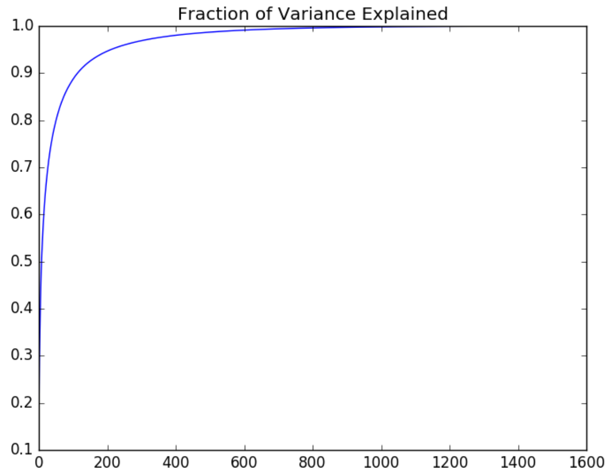
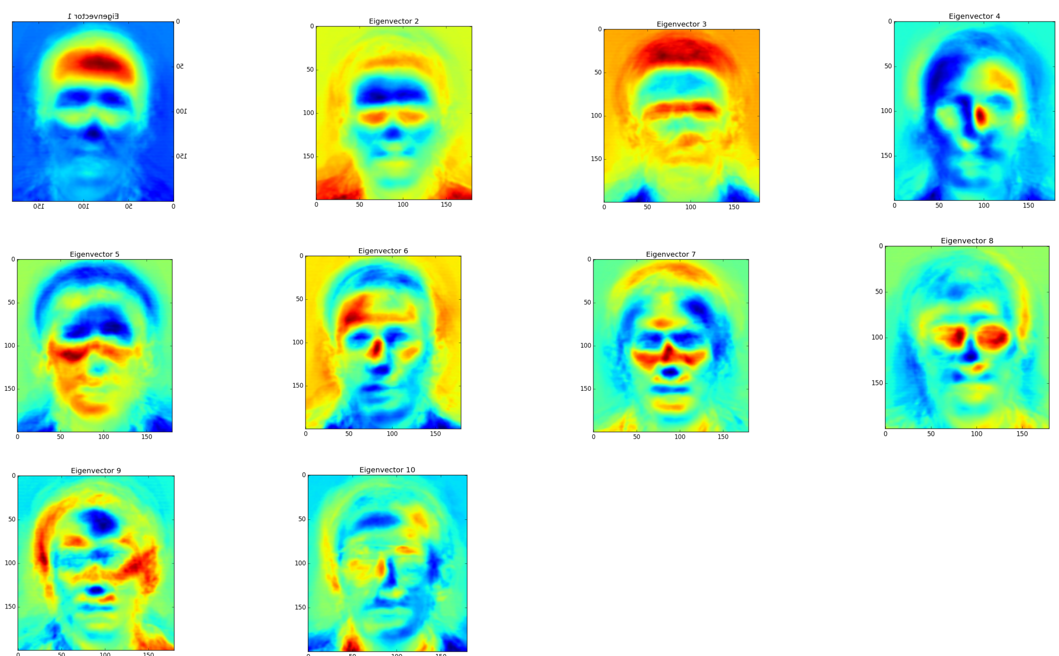


Figure 3. Fraction of Variance for Eigenvalues



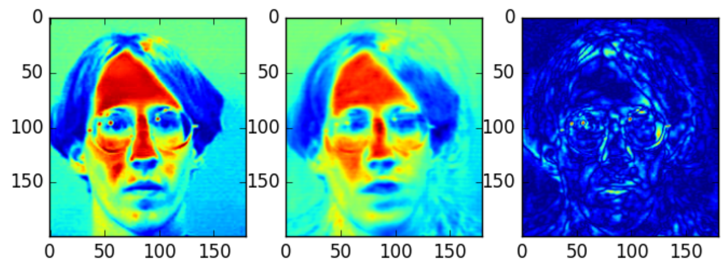
The subspace dimensionality used was 114.

Looking at the principle components above, you can see that each component describes a facial feature that is descriptive of a person. The first component shows that the forehead is very red meaning that the forehead has a lot of variance among individuals. These principle components also outline what the system believes are features of a face in general, so if it sees an image that is not a face it will refer to these components and check if it has these features. This means that the variations in foreheads of individuals can be used as a discriminative factor in identifying an individual. Each principle component isolates a discriminative feature of an individual, where the next ones are the eyes, hair, nose jaw, cheeks, etc. These components also outline the most likely location of each of these features in an image. This means that the model has been trained to assume that people are looking forward and not tilting their heads.

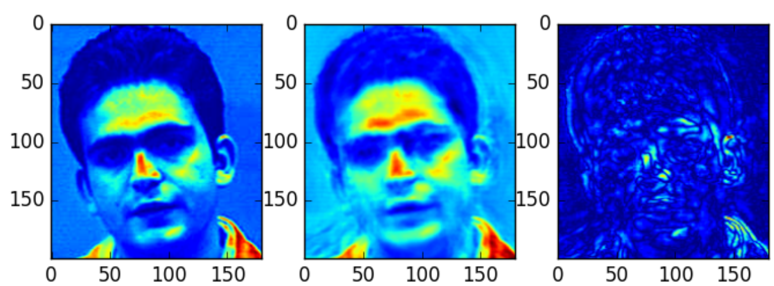
Step 3

Face Reconstruction Results

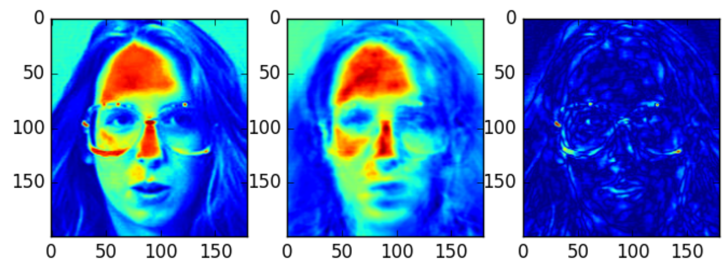
Square Reconstruction Error: 2197.4316165



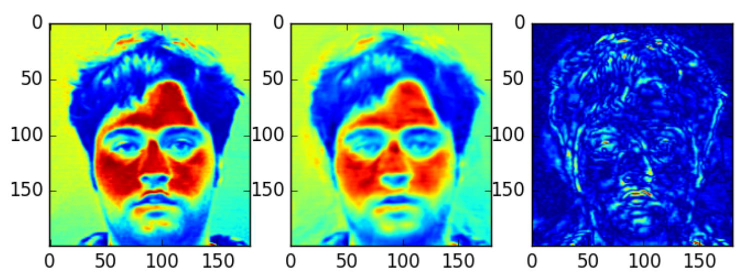
Square Reconstruction Error: 2067.48563296



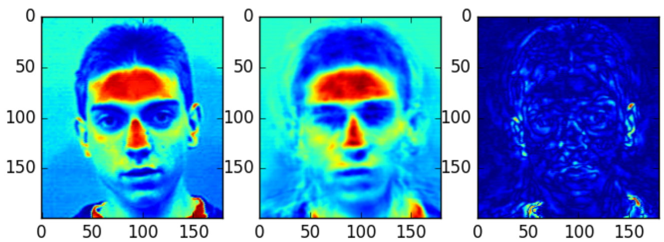
Square Reconstruction Error: 2483.07436851



Square Reconstruction Error: 1427.65997478

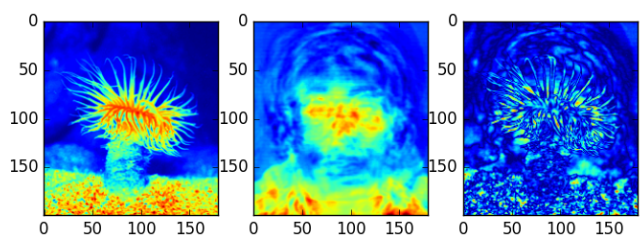


Square Reconstruction Error: 3786.1614095

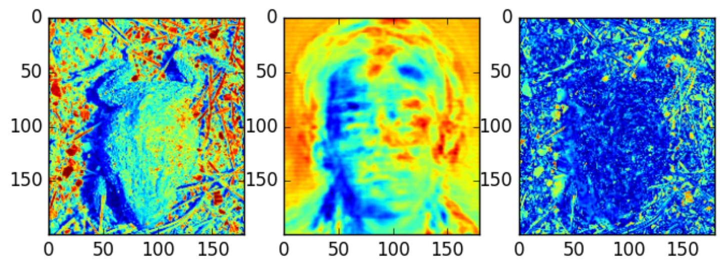


Non face Image Reconstruction

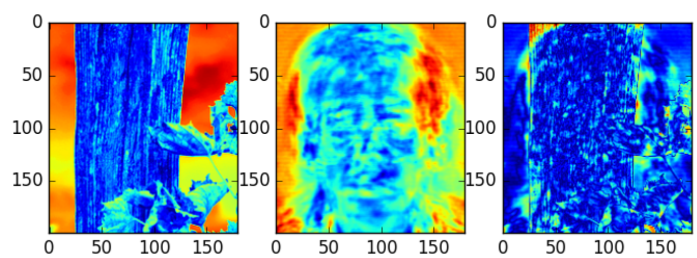
Square Reconstruction Error: 7416.04587379



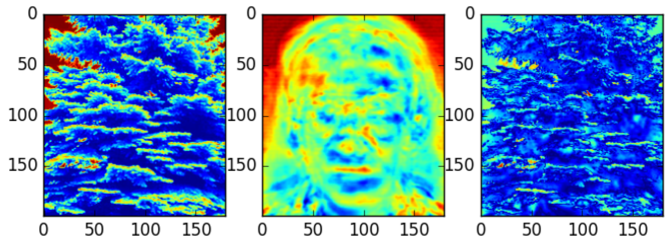
Square Reconstruction Error: 26829.6715679



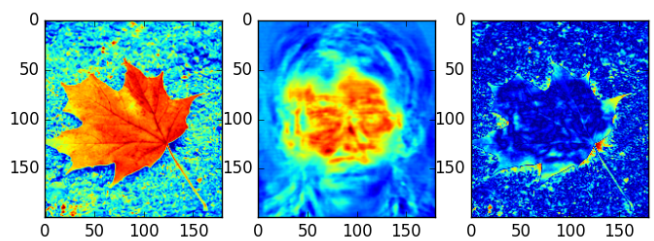
Square Reconstruction Error: 12166.5792411



Square Reconstruction Error: 25033.3309596



Square Reconstruction Error: 15207.9467755



Histograms of Reconstruction Error for Face and Non Face Images

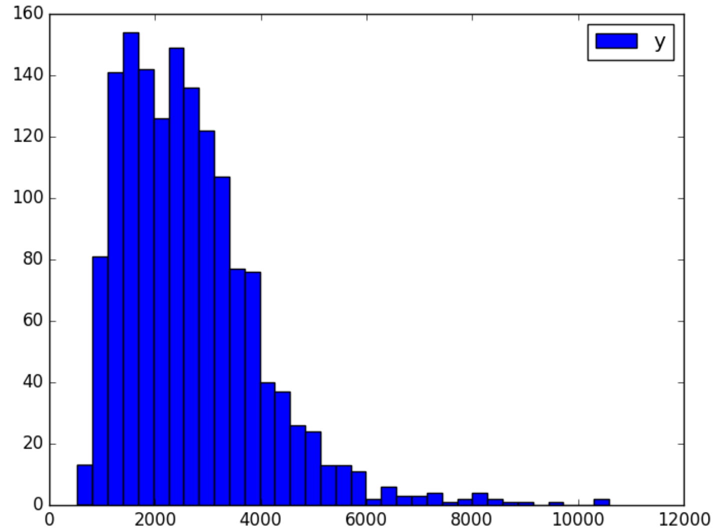


Figure 4. Histogram of Reconstruction Error for Face Images

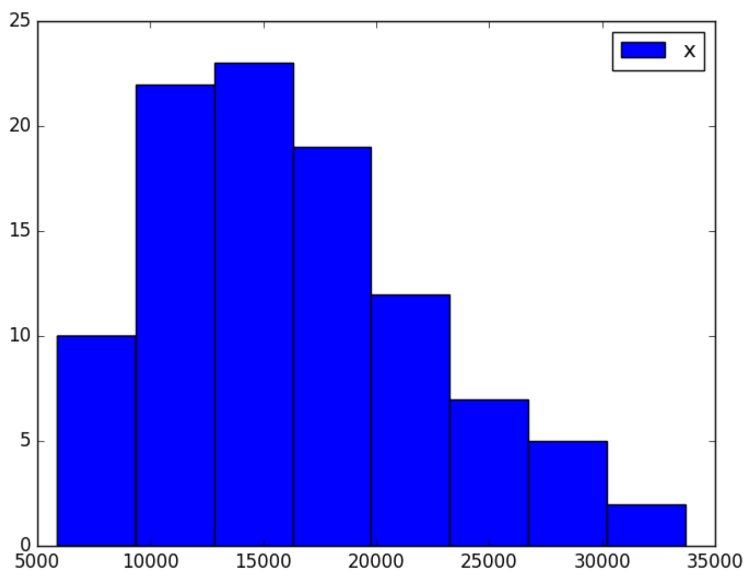


Figure 5. Histogram of Reconstruction Error for Non Face Images

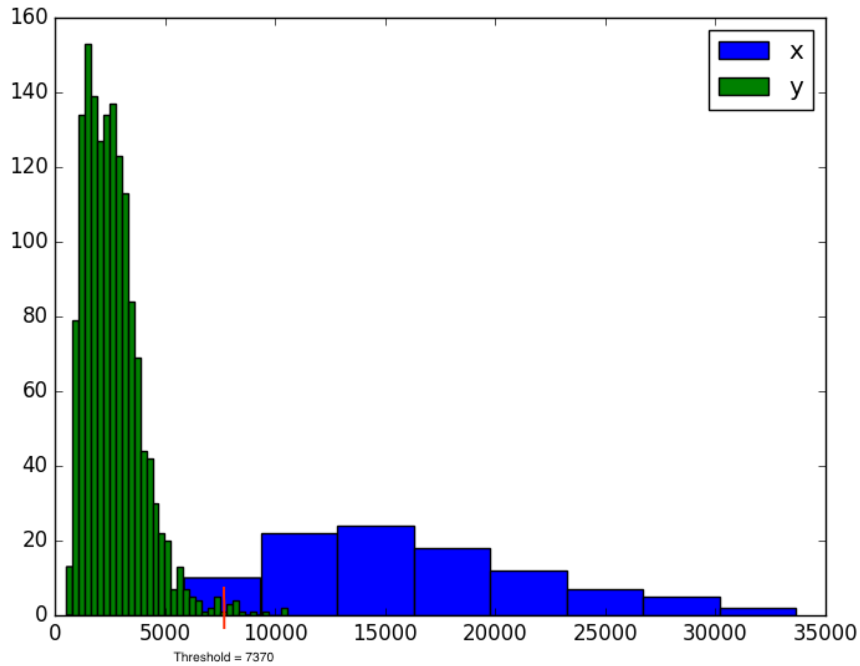
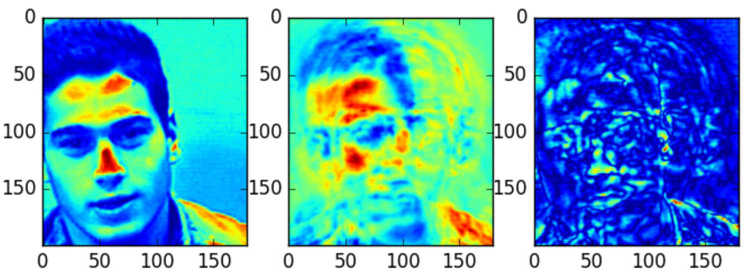


Figure 6. Histogram with both Face and Non Face Reconstruction Errors. Threshold value is chosen to be 7370

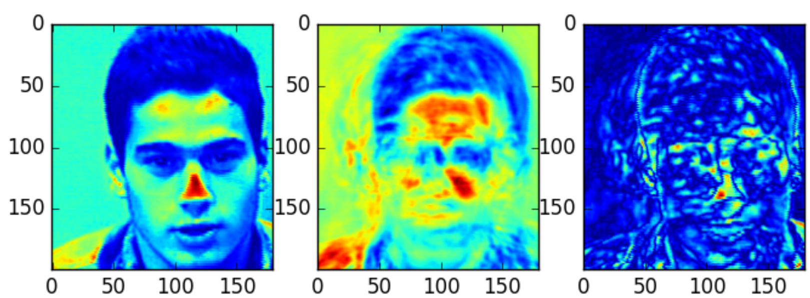
The chosen threshold value was 7370 after inspecting the above histogram.

Classification of Faces as Non Face Examples

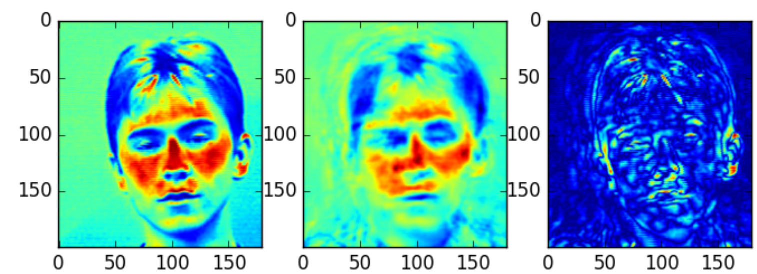
Square Reconstruction Error: 9688.00372802



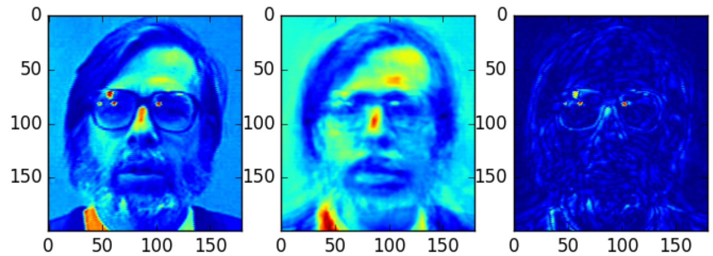
Square Reconstruction Error: 8284.79109436



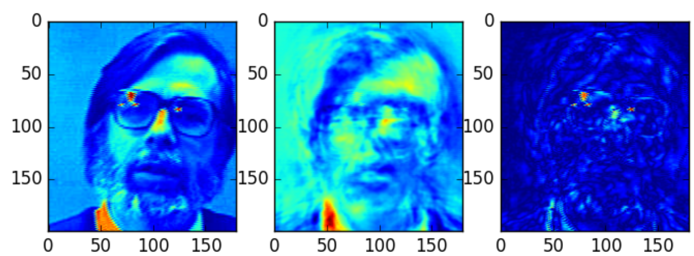
Square Reconstruction Error: 8039.13787216



Square Reconstruction Error: 8018.08056518

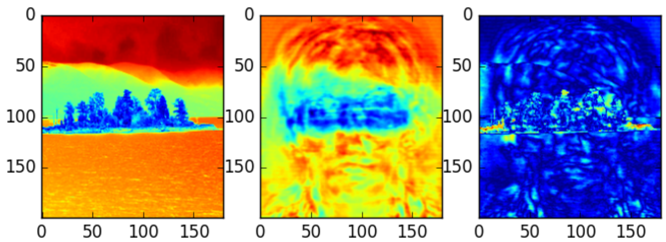


Square Reconstruction Error: 10561.3084232

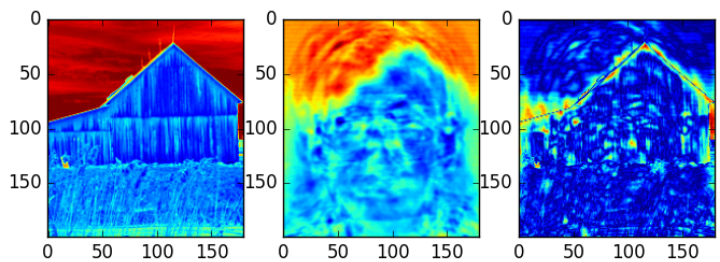


Classification of Non Face Images as Faces Examples

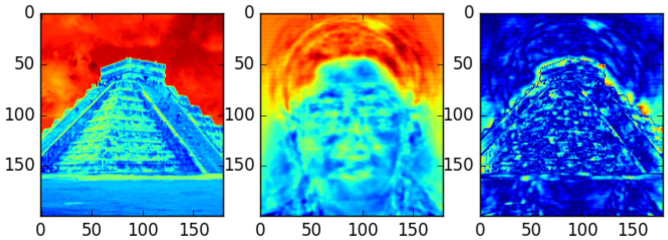
Square Reconstruction Error: 6880.56111118



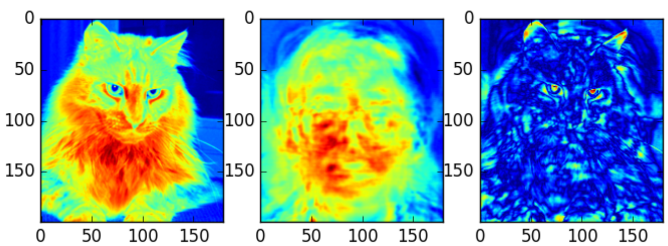
Square Reconstruction Error: 6196.20077575



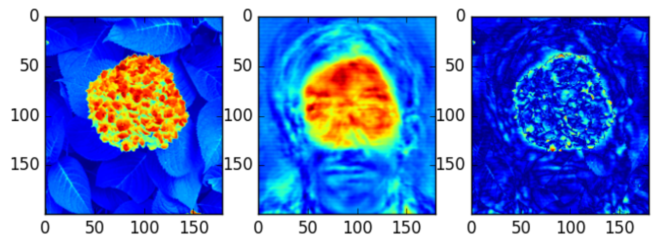
Square Reconstruction Error: 5874.88903204



Square Reconstruction Error: 6569.28742907



Square Reconstruction Error: 6715.53989688



The images have been misclassified because the model has been trained to recognize faces in a very controlled environment where almost all the faces are in the centre of the image, looking straight and head not slanted. So, when there is a face that defies any of these situations, the model will have a larger reconstruction error. Looking at the first and second face false classification, we can see that since the face is not centred, looking down and slanted, there is a high reconstruction error. The third face is also slanted and has closed eyes, whereas the next two faces have very large beards and glasses. However, when looking at the falsely classified non-face images, we can see that all the images have a very central focal point, such as the cat or the pyramid. Since object are symmetric and evenly framed in the picture, this causes the classifier to think these images are faces. To prevent such an issue from occurring more faces should be included in the database with differing lighting, scale and translation.

When looking at the non-face reconstructions we can see that the reconstruction tends to look like a face. This makes sense as we are using the W and b values derived from the face data set. Since we are using the eigenvectors found from the face set, any time something revolving a facial feature is found in the non-face images, the reconstruction will try to form a face out of it. This is evident in all the examples above, but especially in the one just above, where a group of berries were turned into a face after reconstruction. There is such a large difference in reconstruction quality between the face and non-face images, because the face data set has little variance to it and is relatively controlled. With the face data set, the faces are all mostly centred in the frame, looking straight and all follow a template of two eyes, a nose, a mouth, ears, etc. Whereas, with the non-face images, there is no general structure of what the image will be like, so it is harder to generalize the data set and find the principle components that will create a seamless reconstruction.

Step 4

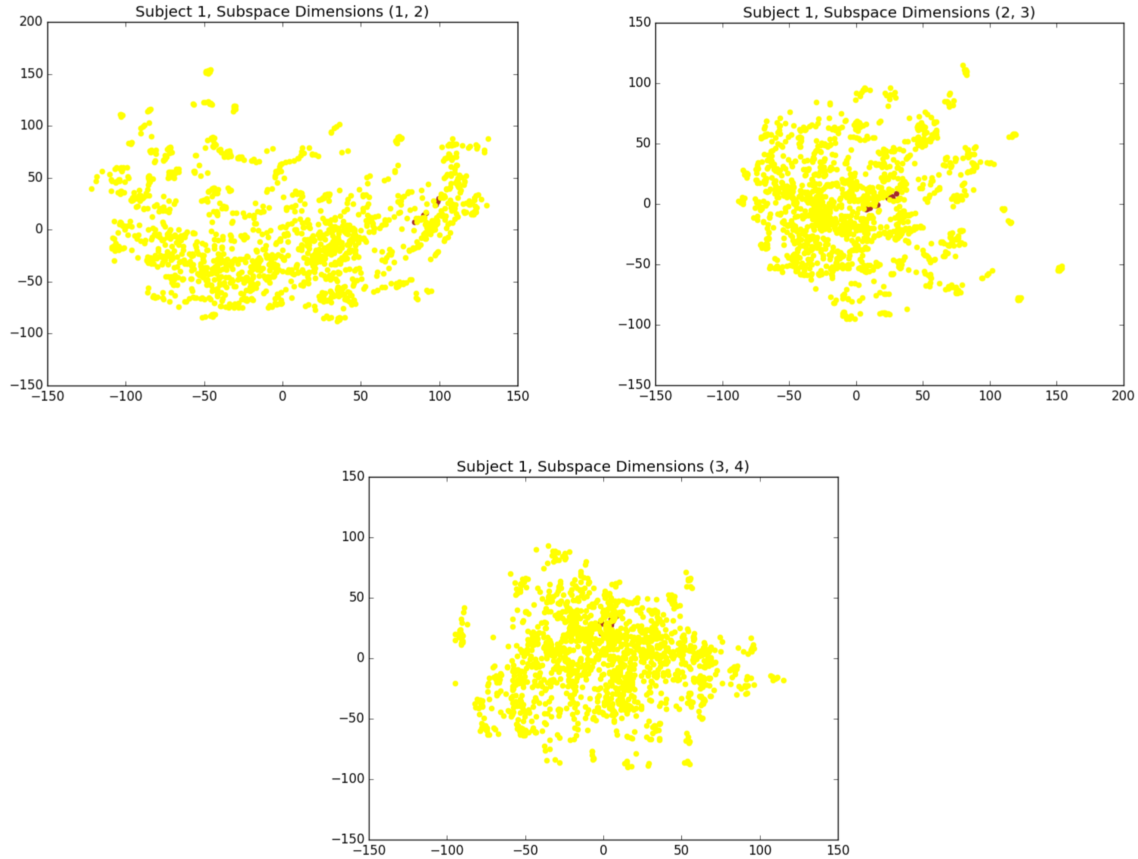
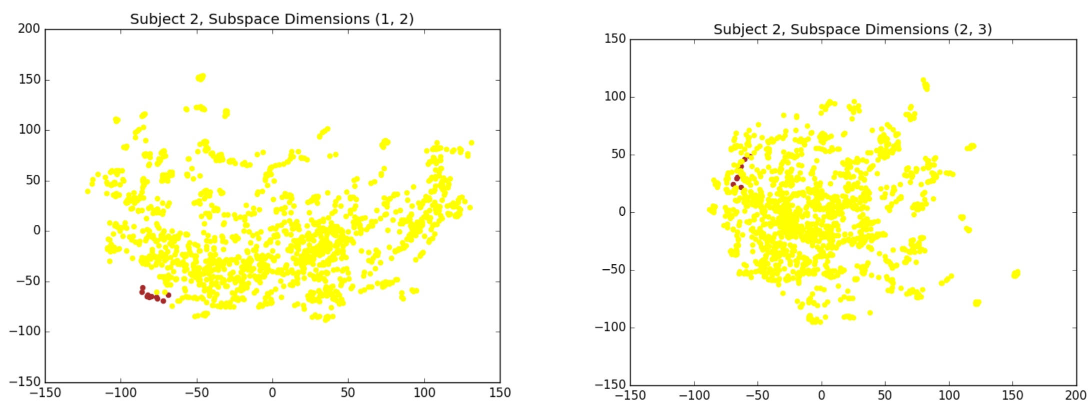


Figure 7. Scatter plots for Subject 1



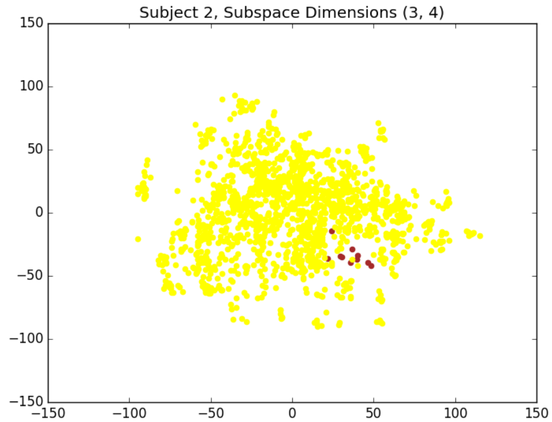
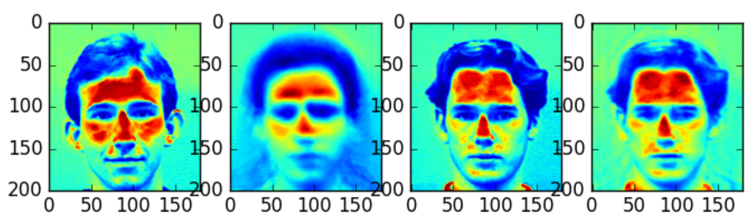
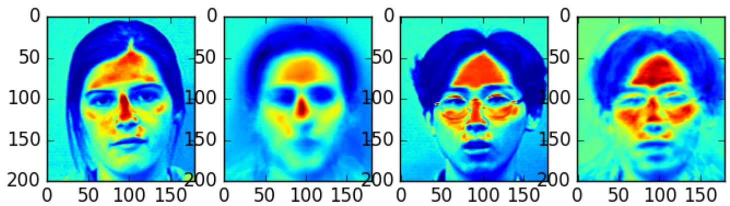
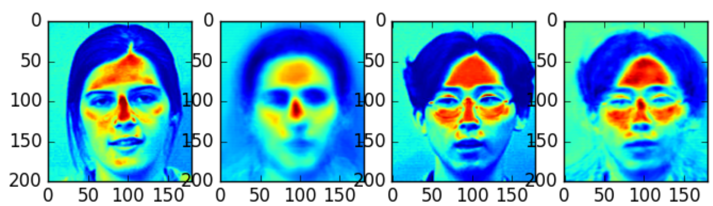
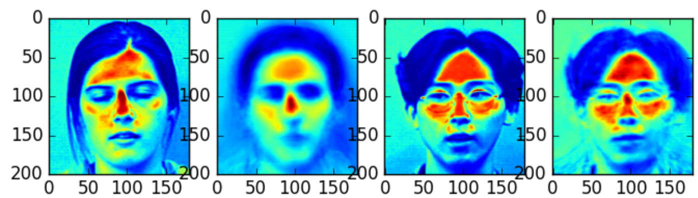


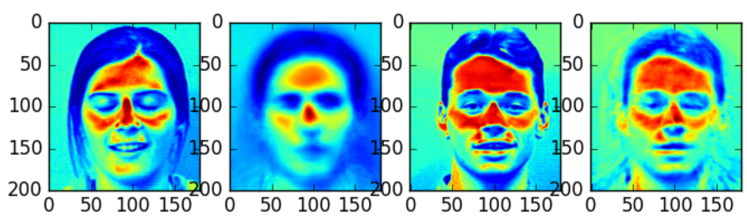
Figure 8. Scatter plots for Subject 2

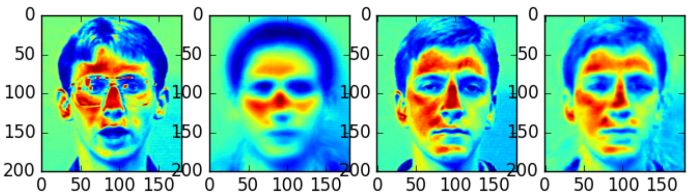


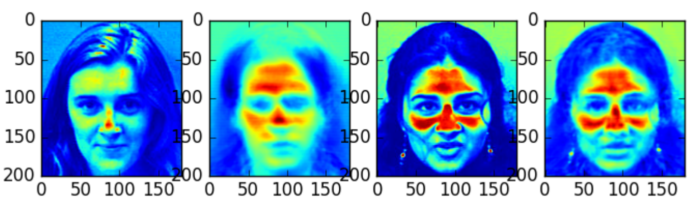

0
50
100
150
200
0 50 100 150

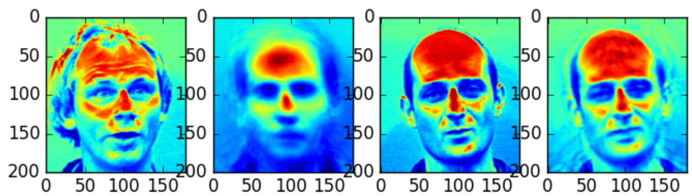

0
50
100
150
200
0 50 100 150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150


0
50
100
150
200
0
50
100
150



When looking at the errors cases of the 1-NN classifier, the matches make sense when comparing the reconstructed images of the test image to the reconstructed image of the best match. This is because all the error case test faces are very different from the rest of the data set in that they have unique expressions, head tilts, mouths open, eyes closed or teeth showing. These test images will then result in very inaccurate reconstructed images as can be seen above. Each of the test faces has a reconstructed face that is instead very straight, centred and blurry. The respective matches of each test image make sense when you look at the general structure of the face, the reconstructed images and the red parts of the face. From the results above one can see that the best match is the one that is closest to the reconstructed image both in structure and dominant facial features, which are in red. For example, the test image just above has a balding man with a large forehead, so his forehead is the discriminating feature, which matches with another bald man. The same conclusion can be made with the other faces as the red parts of the face seem to match with the test image to the training image best match. In these cases, it seems that the reconstructed test image is a better match to the reconstructed training image, as opposed to the original images. Increasing the subspace dimensions would improve the recognition performance, because by limiting ourselves to a subspace dimension that doesn't account for most the variance means we are only some features to discern identity. If we allow for more subspace dimensions, we will be able to allow for more variation in the data and allow more features to contribute to a person's identity.

If the faces only took up 50% of the images and the backgrounds were cluttered, there would be a lot more variance in the images and would create more difficulty in discerning the face from the background. This would result in a PCA subspace that its largest eigenvectors would be highlighting regions in the background as opposed to discriminating features of the face. The average image would probably still be similar as the background would all just blend together. But it would be require more data to get the pattern and to filter out the noise in the background. The challenge here is that you are trying to train the system to recognize faces, but with all the variance in the background from image to image, this causes more uncertainty in the model. So potentially more data would improve this, but I would think it would still result in poor performance because of how uncertain the backgrounds are.