ECE269 - Linear Algebra

Professor: Dr. Piya Pal (UCSD)

Project 2: Face Reconstruction using Principal Component Analysis

Completed by: Sam Cowin

December 17, 2020

Motivation and Project Setup

The main objective of this project was to take a topic from Linear Algebra and apply it to a useful and interesting problem to showcase the power of the material learned in lecture. One such powerful method learned was Singular Value Decomposition, and this project extends this idea. Principal Component Analysis is a method in maximizing the variance of a dataset by mapping the data into a lower dimensional space. How SVD comes into play, is that the SVD decomposition (and subsequent eigendecomposition) is used to find the "most" useful components in the data to describe the entire set. Dimensionality reduction, as this is often used for and called, is common practice in machine learning and PCA is one such popular technique that is used often. Further detail into how exactly this is implemented will be covered throughout the report.

For the setup of the project, we were instructed to read "Face Recognition Using Eigenfaces" by M. A Turk and A. P. Pentland. This paper takes PCA and implements an unsupervised facial recognition and reconstruction algorithm. We were to take the approach they developed and implement the following experiments:

- Compute the principal components (PCs) using first 190 individuals' neutral expres- sion image. Plot the singular values of the data matrix and justify your choice of principal components.

- Reconstruct one of 190 individuals' neutral expression image using different number of PCs. As you vary the number of PCs, plot the mean squared error (MSE) of reconstruction versus the number of principal components to show the accuracy of reconstruction. Comment on your result.

- Reconstruct one of 190 individuals' smiling expression image using different num- ber of PCs. Again, plot the MSE of reconstruction versus the number of principal components and comment on your result.

- Reconstruct one of the other 10 individuals' neutral expression image using different number of PCs. Again, plot the MSE of reconstruction versus the number of principal components and comment on your result.

- Use any other non-human image (e.g., car image, resize and crop to the same size), and try to reconstruct it using all the PCs. Comment on your results.
- Rotate one of 190 individuals' neutral expression image with different degrees and try to reconstruct it using all PCs. Comment on your results.

To do this, we were given 200 smiling and 200 neutral expressions of people face images. These were manually cropped and registered. These images were then manually sorted into a smiling and neutral folder respectively for easy integration into Python. One image was also copied into a rotate folder to be obscured by the final experiment. For the last experiment, a car image was taken and included in a non-human folder. Loading these images and method calls for PCA/reconstruction are all in the main.py file. At this point, the project was setup, and the experiments were completed.

Experiment 1: Compute PCs and Singular Values

The first step in the project is to compute the principal components of the 190 neutral expression images. After loading in all of the images, we convert them to float and scale them by 255.0. This will ease the computational burden and the 255.0 is chosen as this is the pixel range for images. These are then converted to gray scale to have consistency across all of the experiments. To setup the eigenfaces as described in the paper, we then convert the N by N dimensional images into N^2 by 1 dimensional vectors. The mean is taken across the 190 first images of these vectors pixel by pixel. In order for PCA to be effective, we need to normalize the dimensions of the data. This is since we are tracking the largest variance in each dimension and utilizing those components. To normalize this data, we take this average face vector and subtract it from each of the image vectors. The final initial operation that is done in the paper is they take these normalized vectors, and compute the covariance matrix in the following manner:

$$C = AA^T = \frac{1}{M} \sum \Phi_n * \Phi_n^T$$

The problem with this computation is that this results in an N^2 by N^2 covariance matrix which is going to force the rest of the process to be computationally infeasible. To mitigate this, there is a way of utilizing the opposite covariance matrix of $A^T A$ resulting in an m by m covariance matrix. In this project, m (190) is significantly less than N^2 (31266) causing a large benefit by using this approach.

First, for the new covariance matrix, the eigenvalues and corresponding eigenvectors are computed utilizing inbuilt eigendecomposition libraries. Then recognize from the following:

$$A^T Av = \lambda v \rightarrow AA^T Av = A\lambda v \rightarrow CAv = \lambda Av \text{ (since scalar } \lambda\text{)}$$

Meaning that the two covariance matrices have the same eigenvalues in λ and their eigenvectors differ by multiplying by A. Additionally, this means when we scale the lower dimensional eigenvectors to the higher dimension, they will be the eigenvectors corresponding to the largest m eigenvalues for the higher dimensional space. Finally, after computing the new eigenvectors, they must be normalized to unity to prevent arbitrarily large variance solutions from being seen as the "maximum" variance when it was just based on the scale. This completes computing the 190 principal components and singular values as can be seen in lines 8-57 of PCA_detect.py. Below is the image of the singular values computed showing they are decreasing as to be expected since the first values are the most "useful" and have the most information. There are also only 189 of the 190 components since the we centered these 190 assumed independent components around zero and thus one of the components is dependent on the others, resulting in only 189 linear independent principal components. Lastly, they are computed as the square root of the non-zero eigenvalues of the covariance matrix.

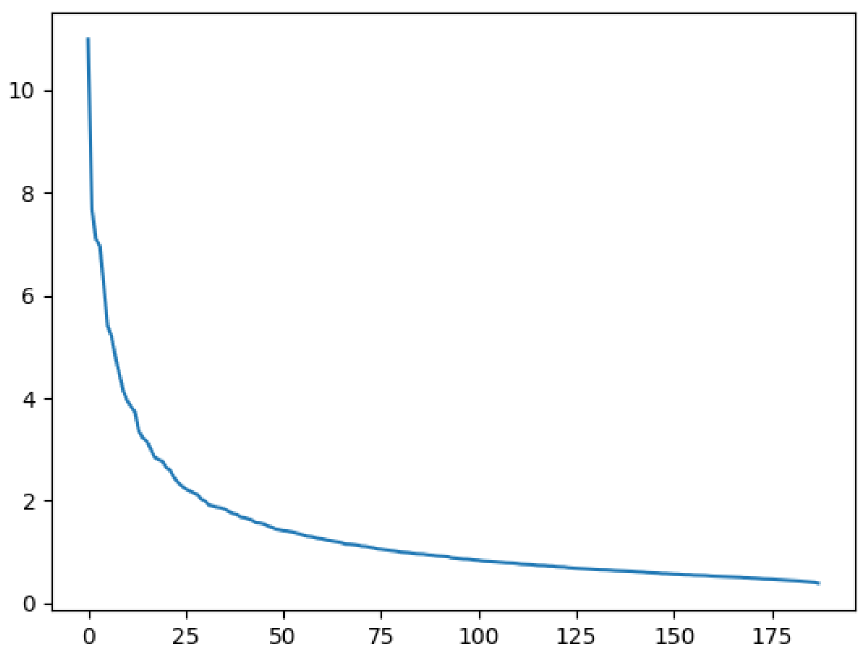


Figure 1: Singular Values of the 189 Principal Components

Experiment 2: Reconstruct Neutral Training Image and Plot MSE vs. Number of Principal Components

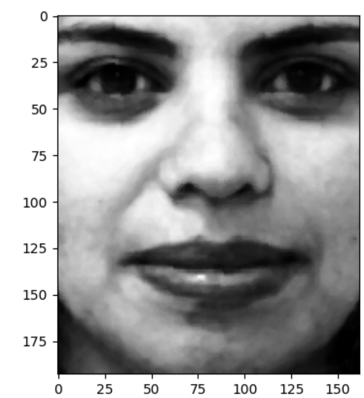
Now that the principal components are all computed, reconstruction can commence. For the first experiment, we are using a training image as the selected option for reconstruction. For PCA, we want to take the k largest principal components, and using these components, reconstruct the image. Intuitively, what we are hoping to accomplish is put together a linear combination of the principal components weighted in such a way that the original image will be reconstructed. Calling these weights w, and using the principal component eigenvectors as u:

$$\hat{\Phi} = \sum_{1}^{k} w_j u_j$$

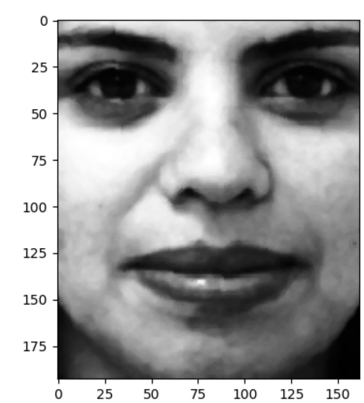
The main challenge, thus, is to compute w. This is done by taking the inner product of the principal component eigenvectors with the image to be reconstructed with the average face removed. Conceptually, we are finding the amount the principal component eigenvectors are contained in the normalized mean image, and then weighting the principal component by this amount. Again, intuitively, using fewer components would naturally make a worse reconstruction as you are dropping information that could be contained in the other principal components. Finally, the average face is added back in and thus the reconstruction is complete. The full formula is below for input image Γ:

$$\hat{\Phi} = \sum_{1}^{k} (u_j^T \Phi) u_j + \text{mean}, \text{ where } \Phi = \Gamma - \text{mean}$$

The looping for the different number of k's and selection between the different experiments is done in lines 73-105 of PCA_detect.py and the reconstruction is done in PCA_detect_utils.py in lines 29-43. One image was reconstructed shown below along with the plot of the MSE vs. the number of Principal Components Selected. As expected, the MSE goes to 0 (not exactly zero due to computation imprecision). This is expected since this face was a part of the training set and thus the 189 principal components will perfectly reconstruct the image. Additionally, the curve for MSE is expected as the first few principal components have the largest information contained within them, and thus the largest drop in MSE is due to them. The elbow in the curve showcases that we likely could stop the reconstruction at 100 principal components to save on computational cost and still receive a decent reconstruction. This is further exemplified as the last few principal components do not really change the MSE to a noticeable degree. For perfect reconstruction, as shown below, all 190 components are needed.



(a) Reconstruction



(b) Original

Figure 2: Neutral Expression Training Image Reconstruction

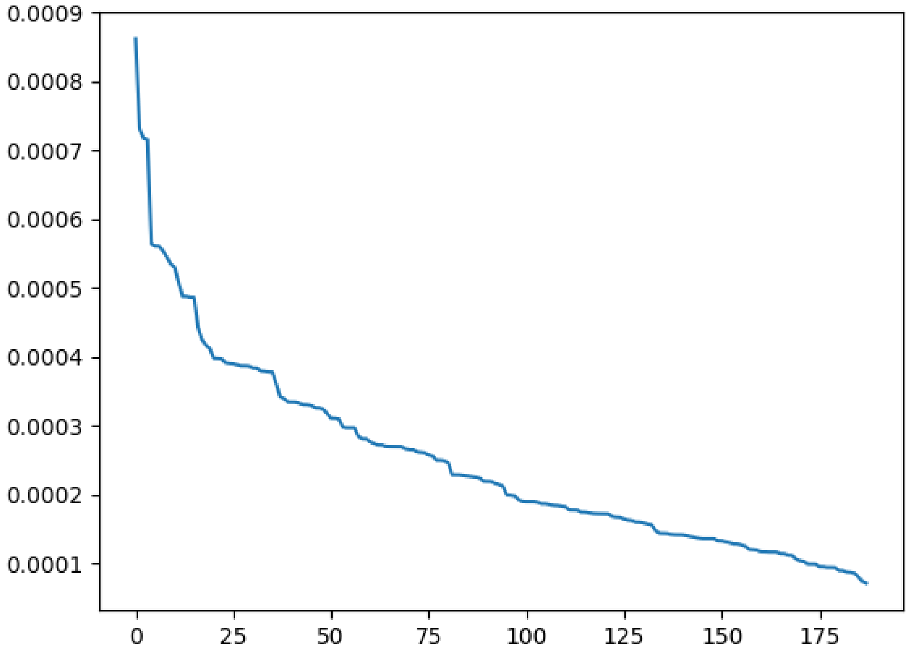
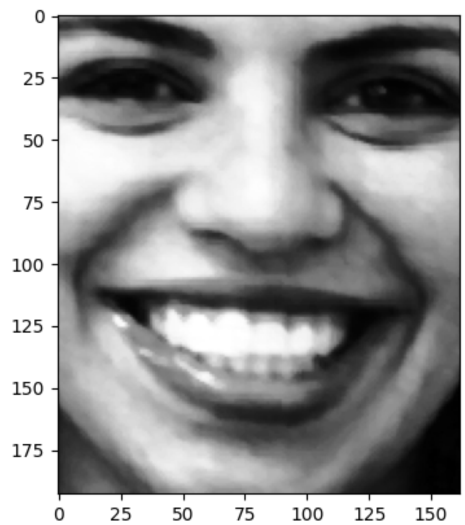


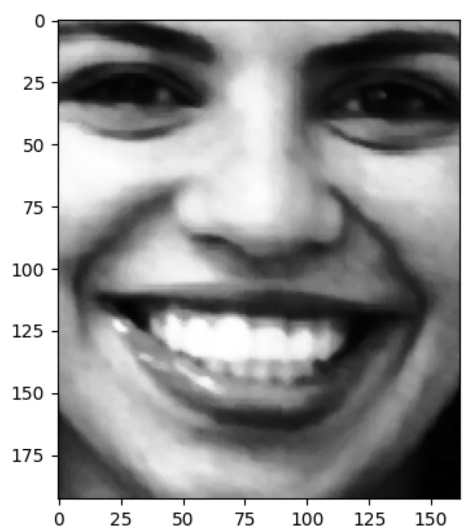
Figure 3: Neutral Expression Training Image Reconstruction: MSE vs. Number of Principal Components

Experiment 3: Reconstruct Smiling Image and Plot MSE vs. Number of Principal Components

This reconstruction follows identically to the Experiment 2 method, except that the image to reconstruct was chosen from the smiling folder of images. The results for the reconstruction and the MSE versus the number of principal components are shown below. The reconstruction looks perfect to the human eye, however examination of the MSE curve shows that the final MSE is slightly higher than that when reconstructing the neutral expression training image. This is expected as this image was not incorporated in the training data, and thus will not have perfect results. Since the face closely resembles that of the neutral expression training image, the reconstruction is still very strong. Again, there is an elbow in the MSE curve and the first principal components cause the largest change in MSE. As already mentioned, this does not converge to zero, and the initial MSE for 1 PC is higher than that seen by the training image reconstruction to further showcase this experiment will not be optimal.



(a) Reconstruction



(b) Original

Figure 4: Smiling Expression Image Reconstruction

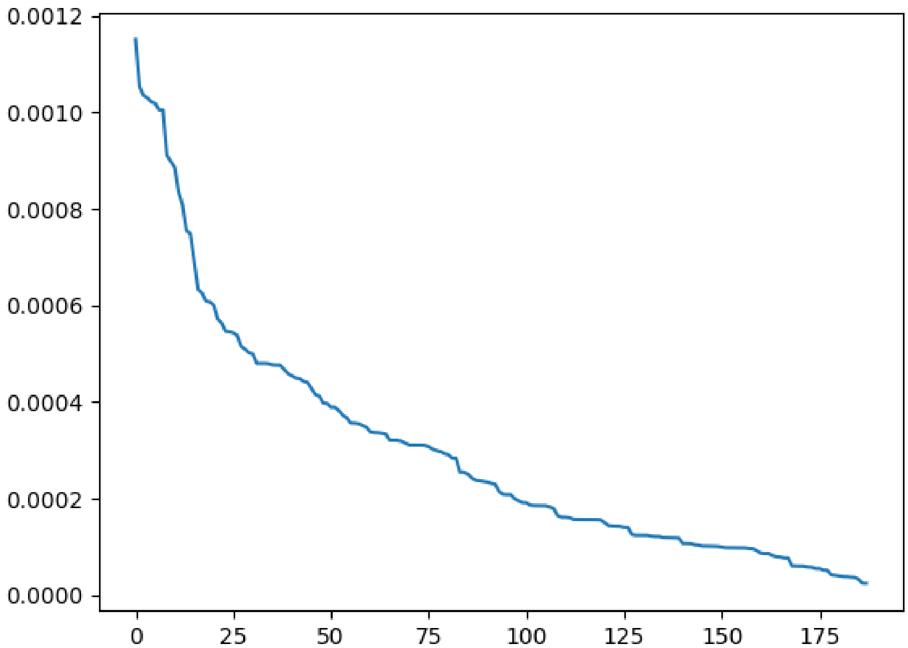
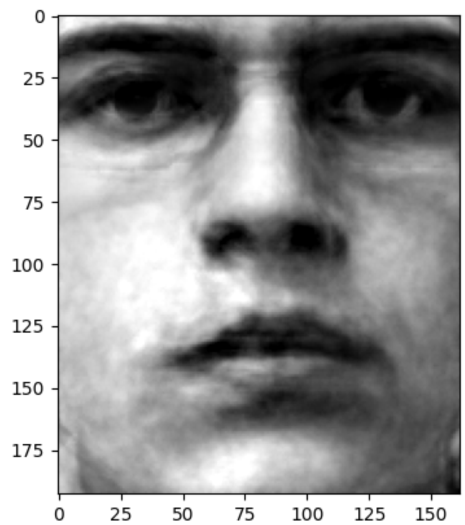


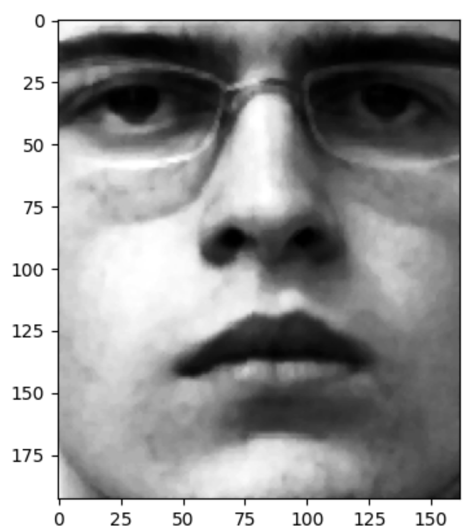
Figure 5: Smiling Expression Image Reconstruction: MSE vs. Number of Principal Components

Experiment 4: Reconstruct Neutral Image and Plot MSE vs. Number of Principal Components

Again, this reconstruction method follows identically to the previous two experiments except that a different image is selected for reconstruction. Different than the first experiment, this neutral image to be used was not in the training data. Additionally, there was a face image that included glasses, which was not common to the faces in the training data, so this image was selected to see how the algorithm would handle this obscurity. The reconstruction results and the MSE plot are included below. To the human eye, this reconstruction is certainly worse than the other two experiments so far. The glasses are not reconstructed well, and the face itself is also not as smooth as the original image. This is also reflected in the MSE plot, as this MSE does not converge close to 0 as the smiling reconstruction did. interestingly, the initial MSE was on par with the other experiments, but the point of convergence for the reconstruction was much worse. The same elbow shape is maintained in the MSE plot.



(a) Reconstruction



(b) Original

Figure 6: Neutral Expression Image Reconstruction

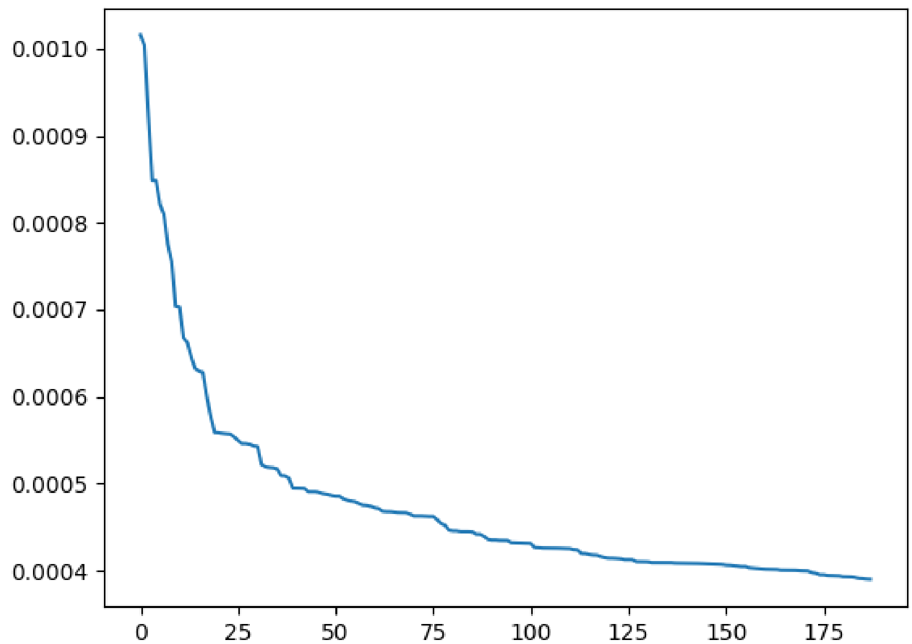
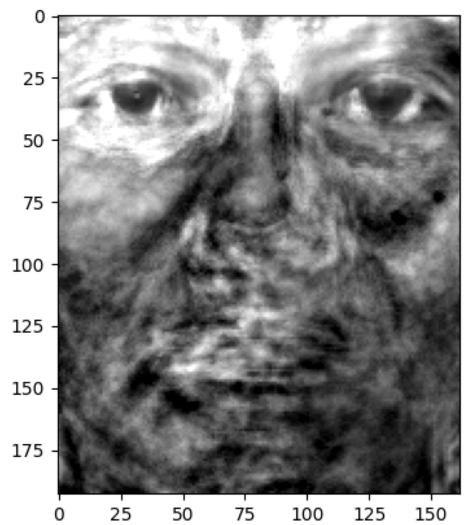


Figure 7: Neutral Expression Image Reconstruction: MSE vs. Number of Principal Components

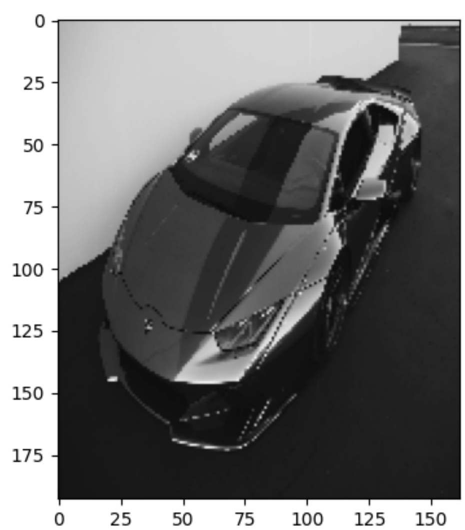
Experiment 5: Reconstruct Non-human Image

For this reconstruction, we were tasked with finding a non-human image and attempting to reconstruct this with all of the principal components. What this means is we are seeing if there is any viable reconstruction capability for images that are not faces centered in the middle of the frame. A stock car photo was taken for this experiment. Since this photo was of different size compared with the training data, it was resized to fit the same dimensions, and its image vector was computed from there. Since MSE was not a requirement for this experiment, the reconstruction follows the same method already described except only the reconstruction for all PCs is found.

The reconstruction comparison is below. It is evident that the reconstruction is very poor. There is nothing that resembles a car in the reconstruction. This makes sense since the only data we had to attempt to reconstruct the image from was faces, and the liklihood that these faces project well into an image of a car is low. What was done unexpectedly well was that the colors (mainly background) of the car image was handled correctly. This is evident in the reconstruction as the top left of the face that was created is lighter compared with the rest of the face that was reconstructed. This follows the same pattern as the image since there is a light colored wall in the top left with the dark car and floor as the rest of the image. In this sense, utilizing this training data may be a viable option for some kind of segmentation with this algorithm, but for pure reconstruction of the non-human image this is not a useful technique.



(a) Reconstruction



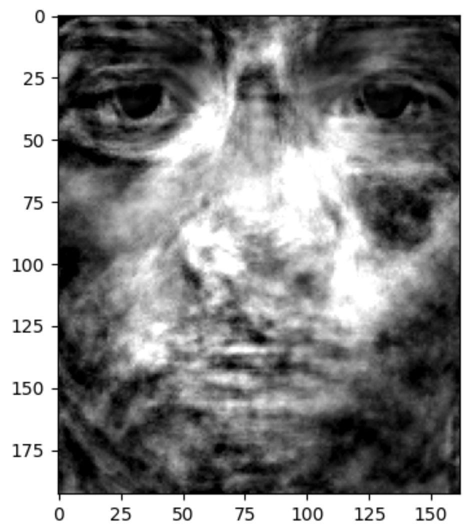
(b) Original

Figure 8: Non-human Image Reconstruction

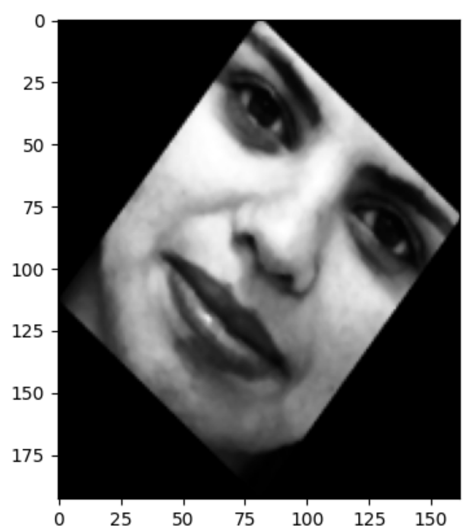
Experiment 6: Reconstruct Rotated Neutral Image

The final experiment that was completed was that of reconstructing a rotated image of different degrees with all of the principal components. The difficulty in this experiment was how to map the rotated images into the same size as the training data in order to run the algorithm as we have done for the other experiments. The method that was chosen was to take one of the images from the training data (the same image from experiment a) and rotate/resize the image in such a way to fit into the frame of the original image. For right rotations (90, 190, etc.) this meant simply rotating and stretching the image for the new dimensions. For angles between these right rotations, this will introduce dead space between the shrunken rotated image and the frame of the original image.

This can be seen in the reconstructions included below. For the 45 degree rotation, similar to the previous experiment, the deadspace is segmented colorwise in the reconstruction. For the reconstructed image portion (along with the other rotations), two faces can be seen with one on the angle and the other in the original position of the face. This makes sense as all of the data was for images that were right side up, so it will partially fill in for that angle as well as the rotated face. These leaves almost a ghost-like effect in the reconstructed image as seen below. This would likely be mitigated if rotations were incorporated into the training data. Along these lines, further improvements in the reconstruction would be made by adding images to the training data with variations in lighting, size, accessories (glasses, piercings, etc.), hair color, etc. MSE was not asked for this portion of the experiment either, however, the MSE results would be poor as the reconstruction is objectively not good. Results for 45 degree rotation, 90 degree rotation, and 180 degree rotation are included below.

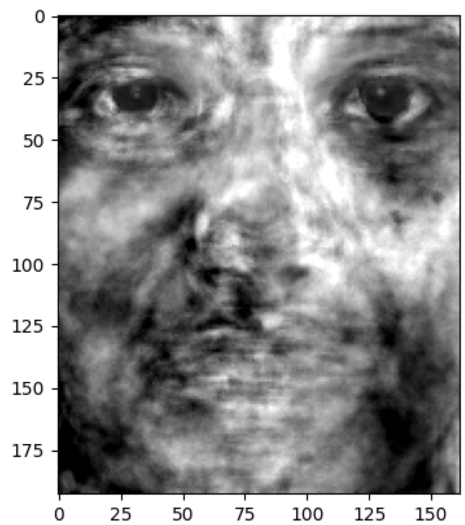


(a) Reconstruction

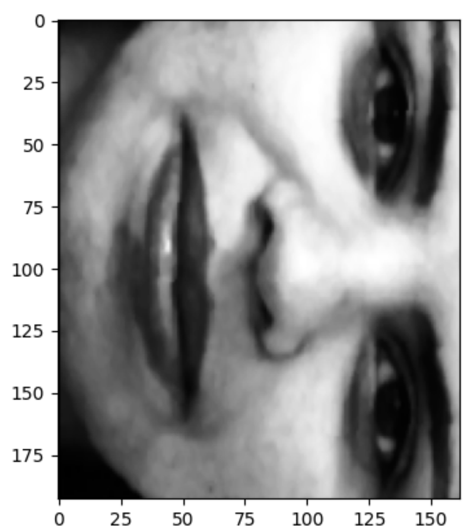


(b) Original

Figure 9: 45 Degree Rotation Neutral Training Image Reconstruction

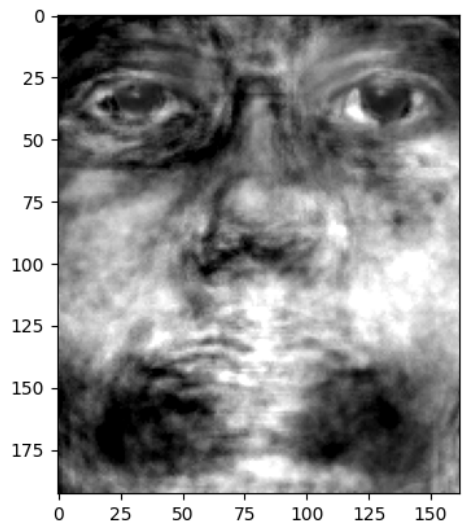


(a) Reconstruction

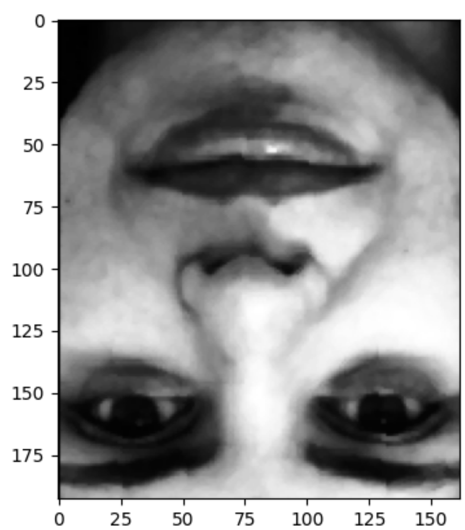


(b) Original

Figure 10: 90 Degree Rotation Neutral Training Image Reconstruction



(a) Reconstruction



(b) Original

Figure 11: 180 Degree Rotation Neutral Training Image Reconstruction

Conclusion

The goal of this project was to utilize class material in Singular Value Decomposition and Principal Component Analysis in order to implement unsupervised facial reconstruction of various images. Throughout the experiments suggested in the project, we discovered that this reconstruction works well for images that are centered in the frame, have specific lighting, and are of only faces. This was exhibited in perfect reconstruction of training data, and good reconstruction of non-training data face images under certain restrictions.

Additionally, we discovered that augmentations in the images in rotations, non-human images, etc. cause the results for image reconstruction to be unusable. What these augmentation results do enable, however, is possibly some color segmentation for backgrounds or objects, but this would need to be proven further. The facial reconstruction of augmented images would be further improved by introducing less inflexible images in the training data.

Overall, using a straightforward and computationally efficient approach put forward by Turk and Pentland, unsupervised facial reconstruction was achieved. This in and of itself demonstrates the utility in the material that was learned in class, and shows that non-mechanical understanding of these Linear Algebra techniques will enable someone to succeed in machine learning, computer vision, and many other applications.

References and Links

In addition to class material, the following were used to implement the project and write the report:

[1]

M. A Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", Proceedings of IEEE CVPR 1991.

[2]

C. D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM 2000

[3]

Links provided for the training data:
`http://fei.edu.br/~cet/frontalimages_spatiallynormalized_cropped_equalized_part1.zip`
`http://fei.edu.br/~cet/frontalimages_spatiallynormalized_cropped_equalized_part2.zip`

[4]

Free stock image for car found at the following website:
`https://www.pexels.com`