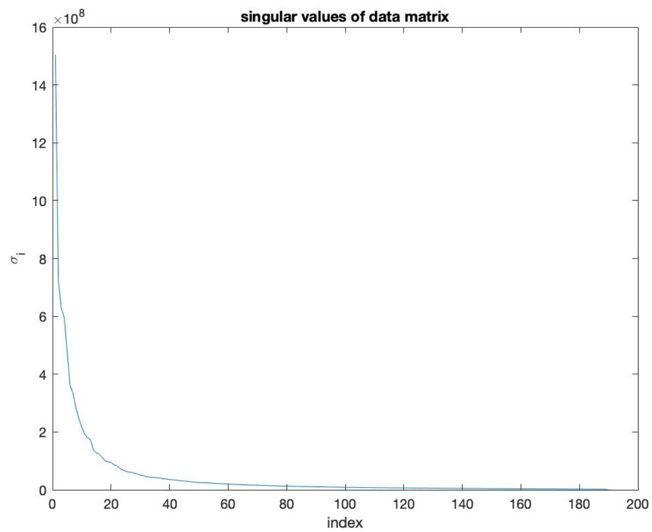
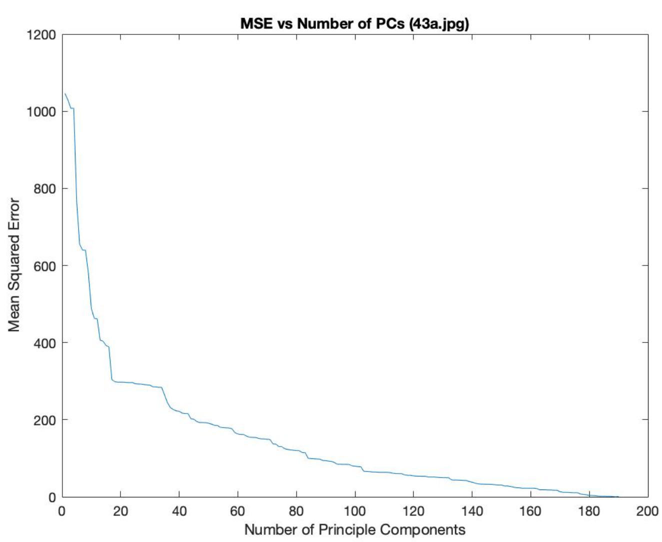
Lucas Lin
A12063068

Mini Project 2: Face Recognition Using Principal Component Analysis

A. The principle components of the data are the normalized columns of the V matrix in the SVD of the data covariance matrix, this also happens to normalized eigenvectors covariance matrix since it is PSD. The reason for using the K largest eigenvalues is to maximize the variance for the least number of components. To determine to number of components to use in practice, we need to consider computation time and accuracy because increasing the number of components increases accuracy but increases computation time. The plot below shows the singular values.

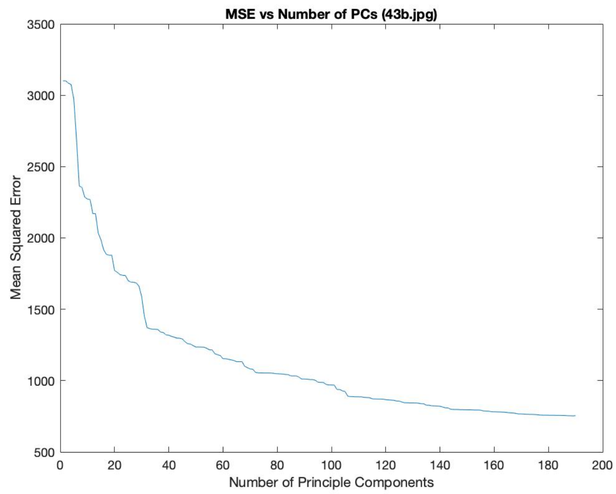


B. The MSE (l2 norm squared divided number of pixels) by decreases as the number of principle components increases. However as mentioned earlier the increased accuracy comes at the cost of greater computation time. This graph can help

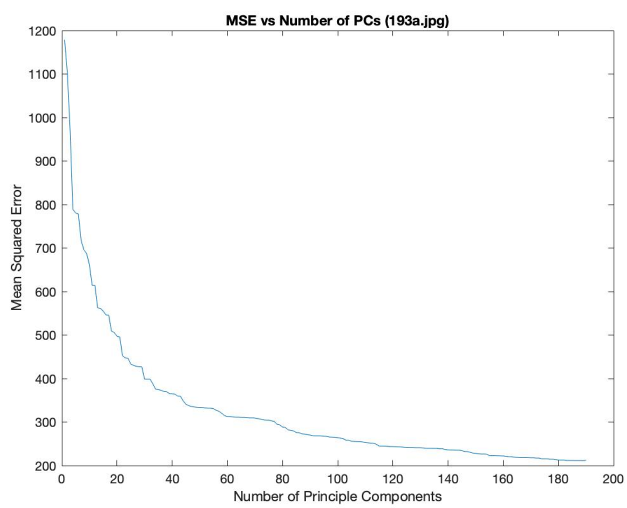


determine how many components to choose by looking at the elbows. For example, there is a steep accuracy increase until 17 PCs where the MSE starts to decrease more gradually. Notice that the MSE is almost zero when all PCs are used (not exactly zero from rounding errors?). This is because the image is in the subspace of the set of all eigenvectors.

C. I decided to use the same face as in part b (face 43). While the error still decreased with more principle components, the error does not reach zero as in the neutral case. This is because the smiling face is not in the range space of the covariance matrix. Below is a reconstruction with all of the components.

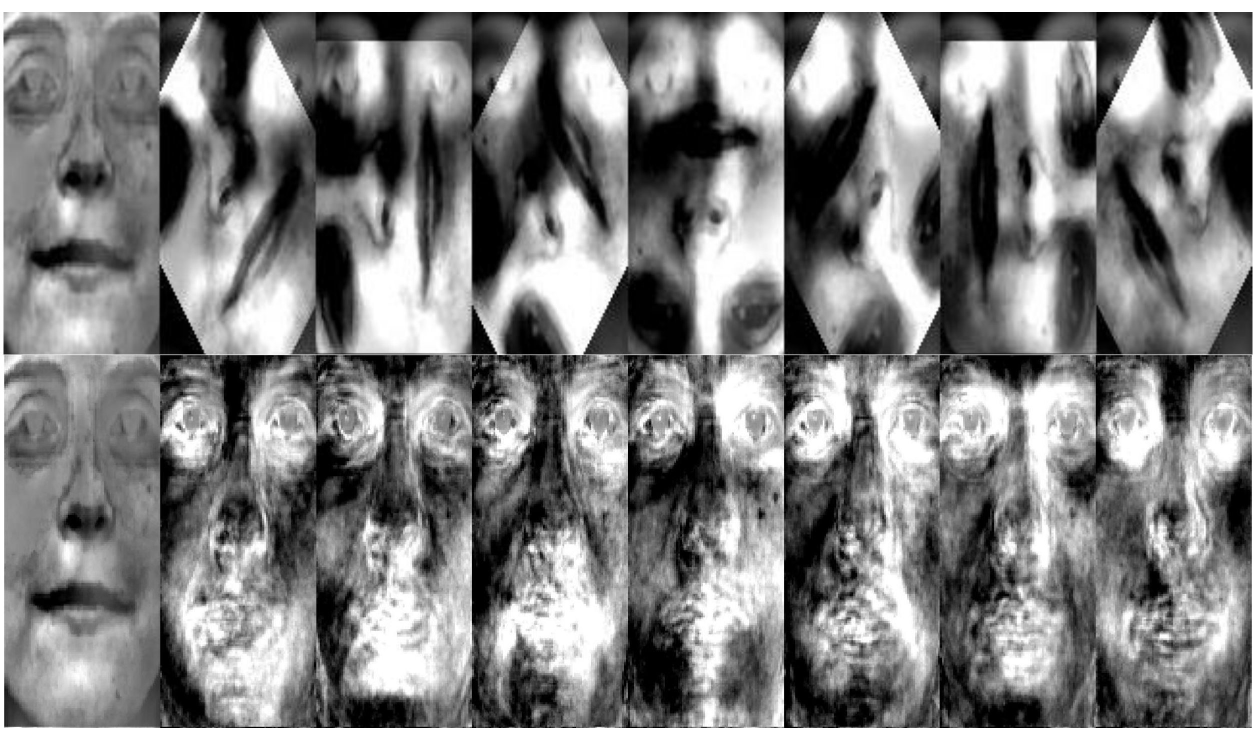


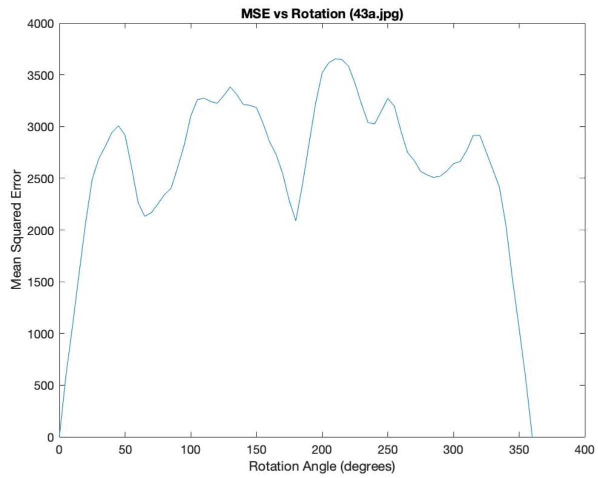
D. I used neutral face 193. This produced similar results to the previous section, but with lower MSE values. The results suggest that covariance is more representative of neutral expressions rather than a person's face.



E. I tried to reconstruct a picture of a slice of pizza using the eigenfaces. The MSE for this was 2882. The reconstruction was not very good, but you see the similarities between the images such as the darkness from the tray, darker shade near the crust, and triangular shape of the slice.

F. Face rotations also don't work that well. The upper pictures show the image of the target rotated face, Phi (after mean-centering), and the lower pictures show the reconstruction. Notice that the rotated images are 'mean centered' using the vertical face. This most likely caused all of the reconstructed rotations to be heavily dominated by the upright face. But if you look closely you can notice the darkness where the eyes are and shadows of the nose and mouth. A plot of MSE vs rotation is also provided. We can see that the 180 degree rotation was a local minima as well as around 60 and 285. The plot is roughly symmetric and error increases dramatically away from vertical.





Appendix

```
%% import training data
tic
Gamma = zeros(31266,190);

for i = 1:100
   s = "./frontalimages_spatiallynormalized_cropped_equalized_part1/"+int2str(i)
+"a.jpg";
   Gamma(:,i) = reshape(imread(s),[31266,1]);
end
for i = 101:190
   s = "./frontalimages_spatiallynormalized_cropped_equalized_part2/"+int2str(i)
+"a.jpg";
   Gamma(:,i) = reshape(imread(s),[31266,1]);
end
%%
Psi = sum(Gamma,2)/190;
A = Gamma - Psi;
C = A*A';
% [V,D] = eig(A'*A); % A'A has same eigenvalues as C
[U,S,V] = svd(A'*A);
V = normc(A*V); % eigenvectors of C are A*V
%% part a) plotting eigenvalues
fa = figure;
plot(1:190,diag(S));
ylabel('\sigma_i');
xlabel('index')
title('singular values of data matrix')

%% part b) determining number of components
% PCs are eigenvectors with the largest eigenvalues
s = "./frontalimages_spatiallynormalized_cropped_equalized_part1/43a.jpg";
Phi = double(reshape(imread(s),[31266,1]))-Psi;
error_n = zeros(190,1);
for i=1:190 % i number of PCs
```

```
    eigfaces = V(:,1:i);
    Phi_hat = eigfaces*(eigfaces'*Phi); % project onto eigenvectors
    error_n(i) = norm(Phi_hat-Phi);
end
fb = figure;
plot(1:190,error_n/31266)
title('MSE vs Number of PCs (43a.jpg)');
ylabel('Mean Squared Error')
xlabel('Number of Principle Components')

figure
imshow(reshape(uint8(Phi_hat + Psi),[193,162]))
%% part c)
s = "./frontalimages_spatiallynormalized_cropped_equalized_part1/43b.jpg";
smile = double(reshape(imread(s),[31266,1]));
Phi_s = smile - Psi;
error_s = zeros(190,1);
for i=1:190 % i number of PCs
    eigfaces = V(:,1:i);
    Phi_hat = eigfaces*(eigfaces'*Phi_s); % project onto eigenvectors
    error_s(i) = norm(Phi_hat-Phi_s);
end
fc = figure;
plot(1:190,error_s/31266)
title('MSE vs Number of PCs (43b.jpg)');
ylabel('Mean Squared Error')
xlabel('Number of Principle Components')

figure
imshow(reshape(uint8(Phi_hat + Psi),[193,162]))
%% part d)
s = "./frontalimages_spatiallynormalized_cropped_equalized_part2/193a.jpg";
newface = double(reshape(imread(s),[31266,1]));
Phi_nf = newface - Psi;
error_nf = zeros(190,1);
for i=1:190 % i number of PCs
    eigfaces = V(:,1:i);
    Phi_hat = eigfaces*(eigfaces'*Phi_nf); % project onto eigenvectors
    error_nf(i) = norm(Phi_hat-Phi_nf);
end
fd = figure;
plot(1:190,error_nf/31266)
title('MSE vs Number of PCs (193a.jpg)');
ylabel('Mean Squared Error')
xlabel('Number of Principle Components')
```

```
figure
imshow(reshape(uint8(Phi_hat + Psi),[193,162]))
%% part e)
s = "./pizza.jpeg";
pizza = rgb2gray(imread(s));
pizza = imresize(pizza, [193,162]);
Phi_p = double(reshape(pizza, [31266,1])) - Psi;
Phi_hat = V*(V'*Phi_p); % project onto eigenvectors
error_p = norm(Phi_hat-Phi_p)/31266;
pizza_reconst = reshape(uint8(Phi_hat + Psi),[193,162]);

fe = figure;
imshow([pizza pizza_reconst])
%% part f)
s = "./frontalimages_spatiallynormalized_cropped_equalized_part1/43a.jpg";
face = imread(s);
figure
imshow(face);
error_r = zeros(73,1);
count = 1;

ff2 = figure;
for i = 0:72
   face_r = imrotate(face,i*5,'bilinear','crop');
   phi_r = double(reshape(face_r,[31266,1]))-Psi;
   phi_hat = V*(V'*phi_r); % project onto eigenvectors
   error_r(i+1) = norm(phi_hat-phi_r);
   if isequal(mod(i*5,45),0) && ne(360,i*5)
      subplot(2,8,count)
      imshow(int8(reshape(phi_r,[193,162])))
      subplot(2,8,count+8)
      imshow(int8(reshape(phi_hat,[193,162])))
      count = count + 1;
   end
end

ff1 = figure;
plot(0:5:360,error_r/31266)
title('MSE vs Rotation (43a.jpg)');
ylabel('Mean Squared Error')
xlabel('Rotation Angle (degrees)')

%%
toc
```