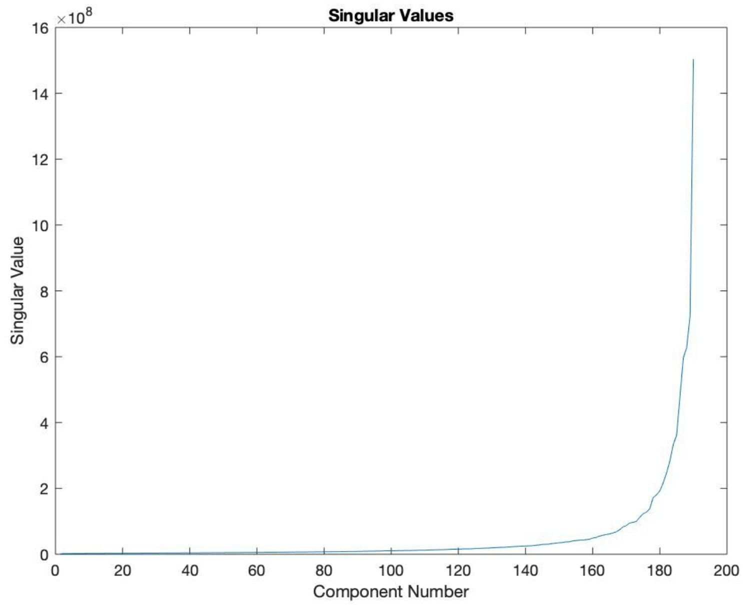
Face Recognition Using PCA

a.



The above plot represents the singular values of $A^T A$, where A is defined as a matrix whose columns each represent one of 190 faces. The principal components should be seleced according to the corresponding highest singular values.

b. Given image 50a, the MSE was computed and plotted for each principal component count from 1 to 190. As the number of components increased, the MSE went down. The MSE starts high at 2000 but eventually achieves near perfect accuracy as all components are used to project the original image. The below subplots show a progressively clearer and more defined image relative to PC number. All faces are neutral as expected.

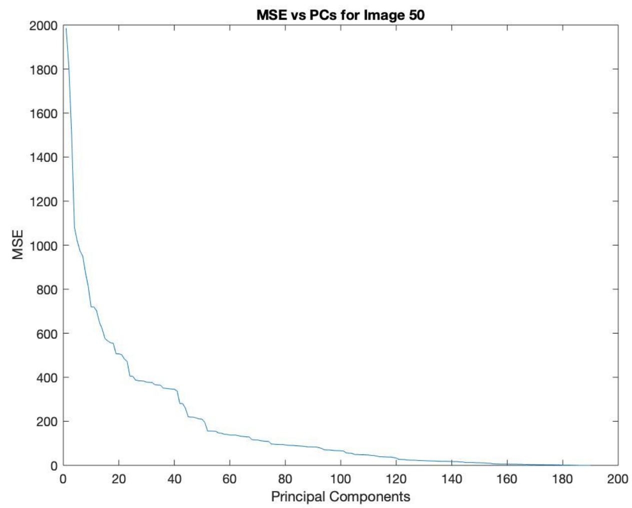
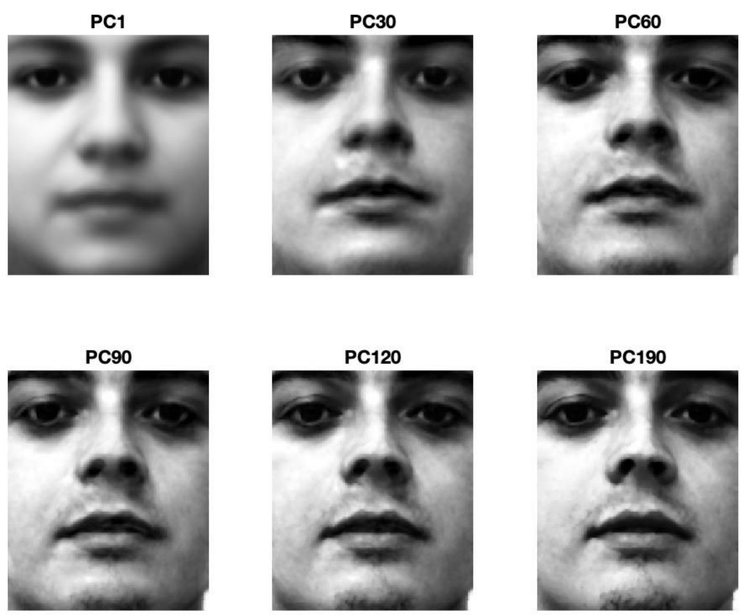
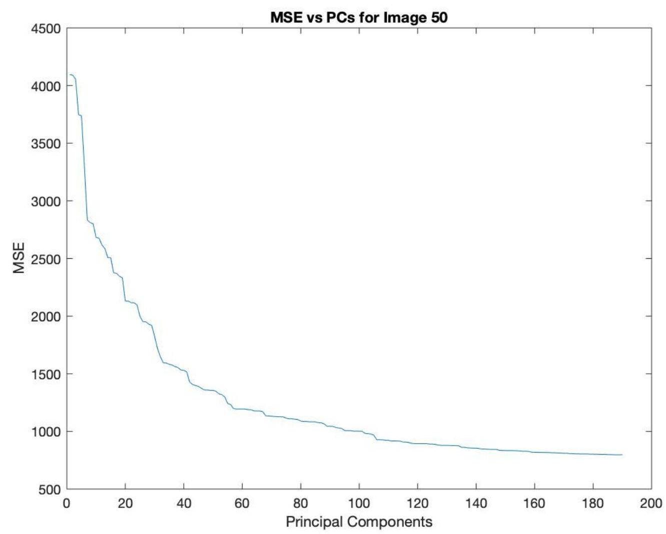


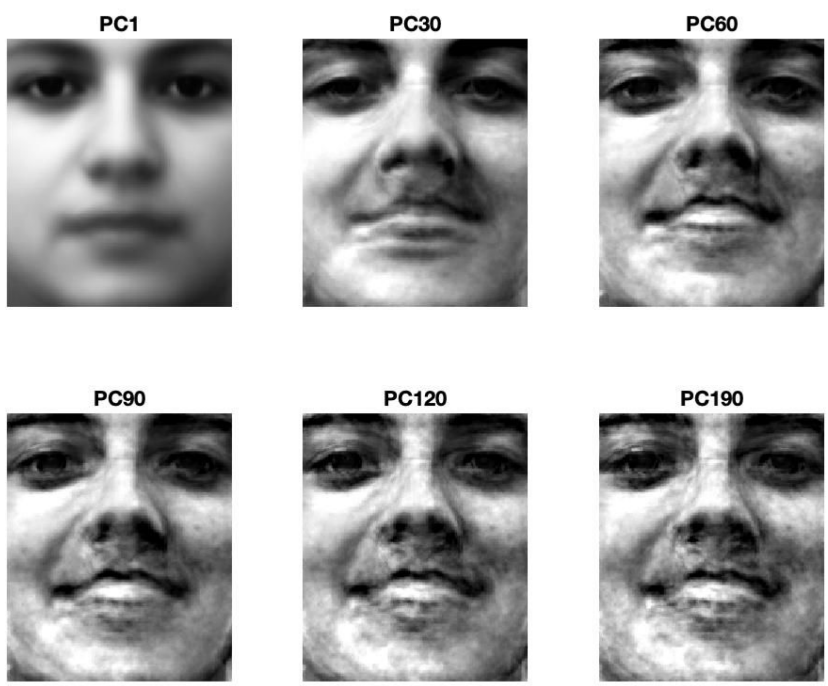
image 50a



c. Given image 50b, the MSE vs principal component plots came out much higher in error. The curve shows the MSE decreasing from 4000 to 1000, compared to the neutral version whose MSE went from 2000 to 0. This is because the principal components were computed from neutral faces, not smiling ones. Upon varying the number of PC’s, the projection almost but doesn’t quite show a smiling person with full teeth, since the training set was based off of neutral images.

image 50b





d. Given image 195a, the MSE plot showed better accuracy than the smiling image in part c, but worse accuracy than the neutral images in part a. That is, the MSE goes from 3300 to 750. This was too be expected since projecting neutral image are more accurate than smiling images. However, it worse than projecting neutral images projections that are actually from the same training set (first 190).

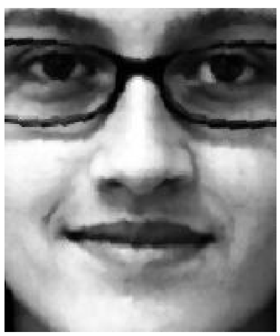
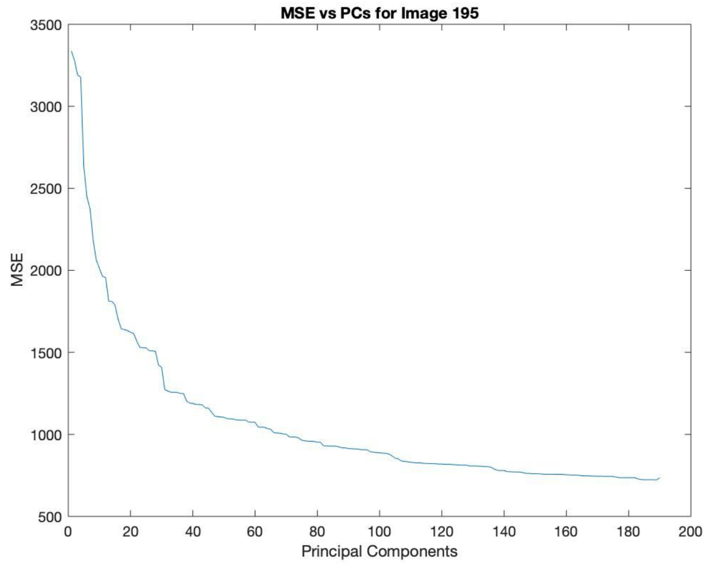
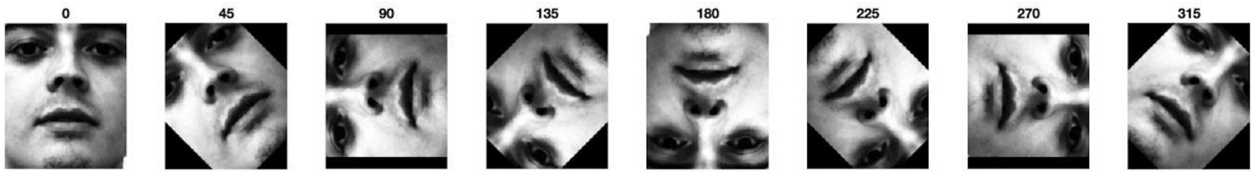


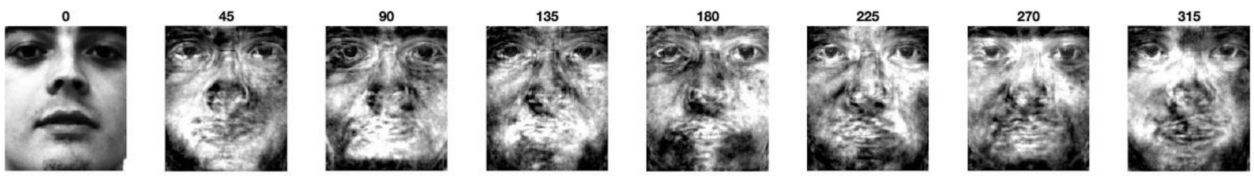
image 195a



e. The projection of a yin-yang symbol onto the eigenspace with maximum principal components is shown below. One can see the projection clearly mirroring the two white and black halves down the vertical center. The projection still came out to be a neutral face, which was expected.

f. The subplots below show the projections of rotated images at 45 degree increments using all 190 PC's. All the images come out to be neutral-faced people, but distorted according to the orientation. This can especially be seen with the location of the eyes in the unprojected images, which appear at relatively the same location in the corresponding projected image.





code:

```
%% Part a: eigen decomp of A
clear all;
close all;
CurDir = '/Users/alberttan/Documents/ECE 269/Project/training';
filesA = dir(strcat(CurDir,'/*a.jpg'));
neutralImages = [];

for i  = 1:length(filesA)
   anImage = imread(strcat('Project/training/',filesA(i).name));
   anImage = reshape(anImage, [31266 1]);
   neutralImages = [neutralImages anImage];
end
neutralImages=double(neutralImages);

filesB = dir(strcat(theCurDir,'/*b.jpg'));
smilingImages = [];

for i  = 1:length(filesB)
   anImage = imread(strcat('Project/training/',filesB(i).name));
   anImage = reshape(anImage, [31266 1]);
   smilingImages = [smilingImages anImage];
end
smilingImages=double(smilingImages);



theCurDir = '/Users/alberttan/Documents/ECE 269/Project/testing';
filesC = dir(strcat(theCurDir,'/*a.jpg'));
theNeutralImagesTest = [];

for i  = 1:length(filesC)
   anImage = imread(strcat('Project/testing/',filesC(i).name));
   anImage = reshape(anImage, [31266 1]);
   theNeutralImagesTest = [theNeutralImagesTest anImage];
end
theNeutralImagesTest=double(theNeutralImagesTest);




M = mean(theNeutralImages,2);
```

```
A = double(theNeutralImages)-M;
[V,D]=eig(A'*A);
diagD=diag(D);
singularValues=svd(A'*A);
x=1:1:190;
y=singularValues;
plot(x,sort(y));
title('Singular Values');
xlabel('Component Number');
ylabel('Singular Value');

PCs=normc(A*V);

flag=imread('yinyanh.jpg');
greyFlag=rgb2gray(flag);
imshow(greyFlag)
figure
flagMatrix=uint8(imresize(greyFlag,[193,162]));
imshow(flagMatrix);
flagFinal=double(imresize(flagMatrix,[31266,1]))-M;

%% part b, one image

%compute lowest for arbitrary k


errorArray=zeros(1,190);
imageArray=zeros(31266,6);
phi= flagFinal-M; %independent of k
indexSelection=[1,30,60,90,120,190];


matrix=reshape(phi,[193,162]);
count=1;

for i=1:190
    arr = 191-i:190;
    eigVecMatrix = PCs(:,arr);
    phi_hat_i = eigVecMatrix*(eigVecMatrix'*phi);

    if ismember(i,indexSelection)
        imageArray(:,count)=phi_hat_i;
        count=count+1;
```

```
    end


    error_i=immse(phi,phi_hat_i);
    errorArray(i)=error_i;
end
%% plot error part a.2

errorx= 1:1:190;
figure;
plot(errorx,errorArray);
title('MSE vs PCs for Image 195');
xlabel('Principal Components');
ylabel('MSE');

%% part f rotation
figure
for i=0:7
    rotation=i*45
    phi= theNeutralImages(:,136); %stays the same no matter k
    J = imrotate(reshape(phi,[193,162]),rotation,'bilinear','crop');
    phi_rotated=reshape(J,[31266,1])-M;

    phi_hat_rotated_i = PCs*(PCs'*phi_rotated);
    subplot(2,8,i+1)
    imshow(uint8(J));
    title(rotation);

    subplot(2,8,i+9)
    displayIm=uint8(phi_hat_rotated_i+M);
    displayImMatrix = reshape(displayIm,[193,162]);
    imshow((displayImMatrix));
    title(rotation);
end


%% display

displayIm = PCs*(PCs'*flagFinal)+M;
displayImMatrix = reshape(displayIm,[193,162]);
imshow(uint8(displayImMatrix));

%% display multiple of image 100
```

```
figure
for i=1:6
    subplot(2,3,i)
    displayIm=uint8(imageArray(:,i)+M);
    displayImMatrix = reshape(displayIm,[193,162]);
    imshow((displayImMatrix));
    title(['PC', num2str(indexSelection(i))]);
end
```