Code also on Github at:

https://github.com/kkintnermeyer6/AMATH584HW

1. Given $A = U\Sigma V^*$, $A^* = V\Sigma^* U^*$. Then $AA^* = U\Sigma V^* V\Sigma^* U^*$. Since $V^*V = I$ this reduces to $AA^* = U\Sigma\Sigma^* U^*$. Note that $U^* = U^{-1}$. Since Σ is diagonal $\Sigma\Sigma^*$ is also diagonal with $\sigma_{ii}\bar{\sigma_{ii}}$ on the diagonals which equals $|\sigma_{ii}|^2$. We also note here that (AA^*) has real eigenvalues (property of Hermetian Matrix) and is positive semi-definite (property of any H^*H). As a result, we know that all the eigenvalues of (AA^*) are all positive and real or zero. We then consider the eigendecomposition of $(AA^*) = V\Lambda V^{-1}$, then $V = U$, $V^{-1} = U^*$ and $\Lambda = \Sigma\Sigma^*$. In this case we can pick any σ such that $|\sigma_{ii}|^2 = \lambda$. Out of convention we pick σ to be positive and real. So the singular values must be the square roots of the eigenvalues of AA^*. A similar analysis can be applied to $A^*A = V\Sigma^*\Sigma V^*$ to obtain the same result.

2. Given $A^* = A$. We know that eigenvalues of A are real (property of Hermetian matrix), thus the eigenvalues of A^*A which are the same as the eigenvalues of AA^* will all be real, and furthermore must be positive as the eigenvalues of A^2 are equal to the squares of the eigenvalues of A. We showed in 1. that the singular values are the positive square roots of the eigenvalues of AA^*. Since singular values are always taken to be positive, this means that $sqrt(\lambda^2) = \sigma$. However, a negative eigenvalue can still yield a positive singular value since squaring the negative value will give a positive value and then taking the positive square root of that value will yield a positive singular value. This leads to two possible cases. Either $\sigma = \lambda$ or $\sigma = -\lambda$. So the best we can say is that $|\sigma| = |\lambda|$.

3. Note that $|det(A)| = |det(U)| * |det(\Sigma)| * |det(V^*)|$. We are given that $|det(U)|$ is 1 and we also know that $|det(V^*)| = 1$ due to the properties of a unitary matrix. Thus $|det(A)| = |det(\Sigma)|$. But since Σ is diagonal, its determinant is just the multiplication of all the diagonal terms, which are the singular values. Since singular values are always chosen to be non-negative, $|det(A)| = \Pi_{j=1}^{m}\sigma_j$

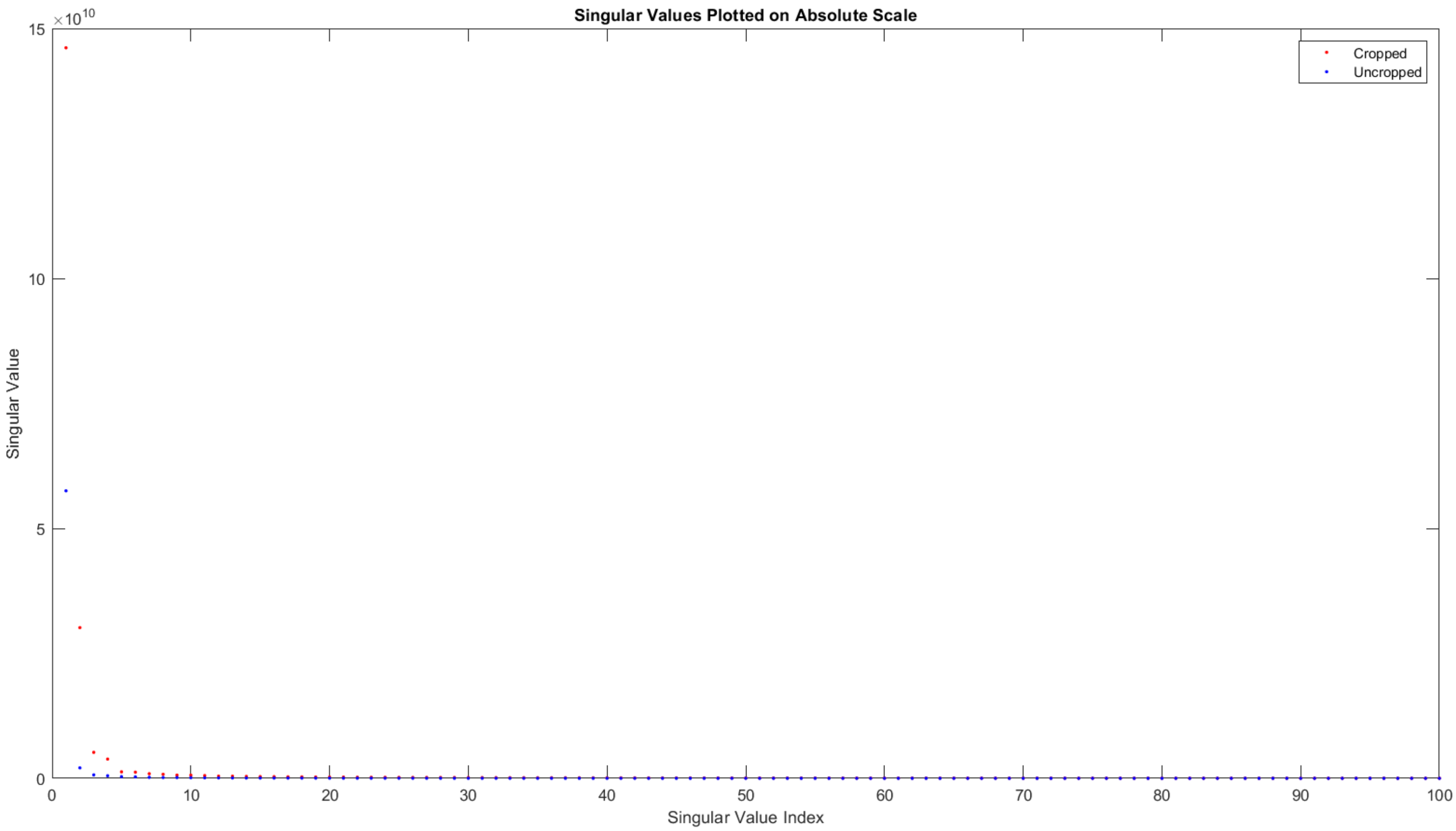
Yale Faces Analysis:

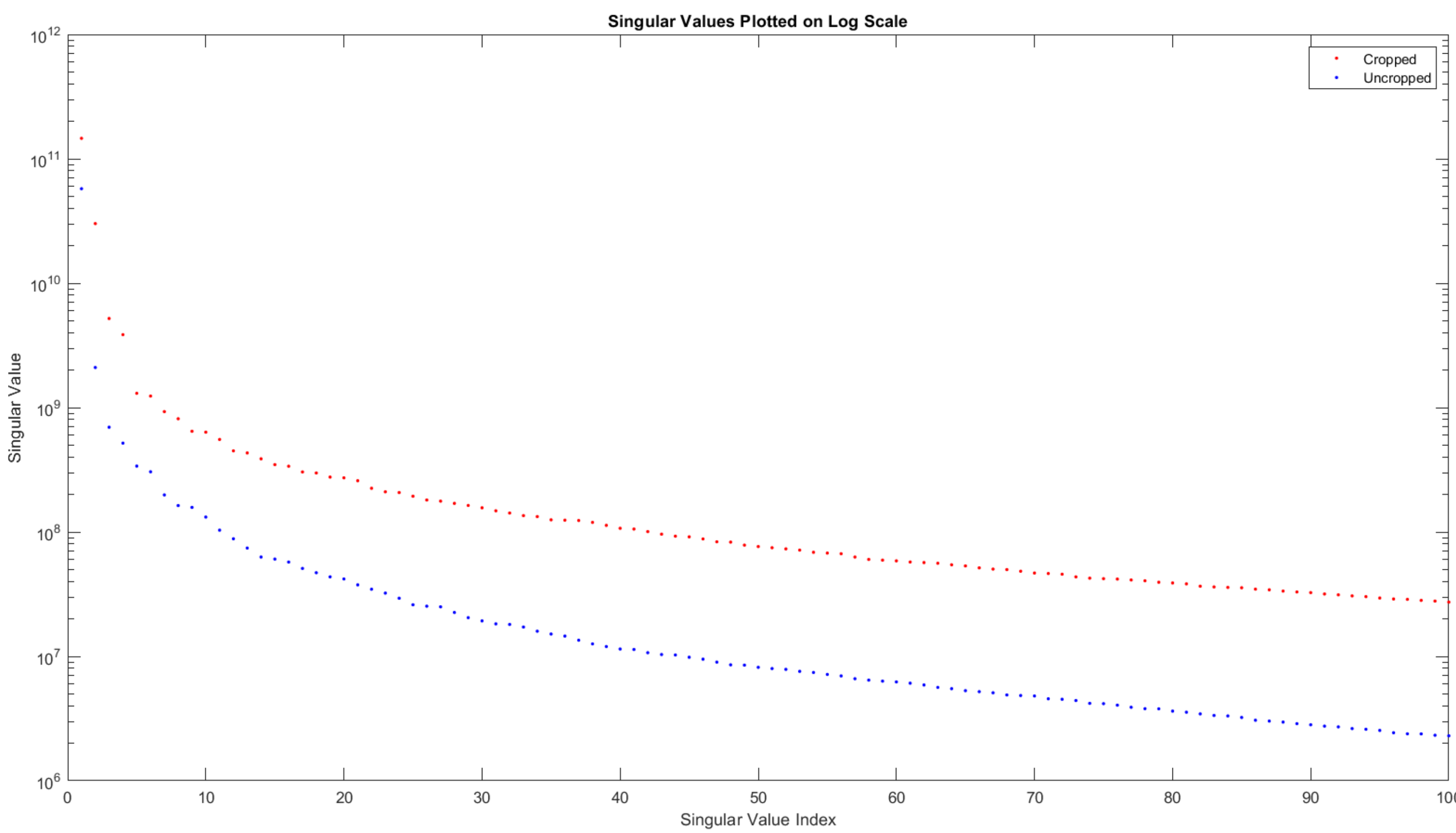
Initial comments and answers to questions in homework, more commentary further below

2. The interpretation of the U matrix is the dominant modes of the imageset, The Sigma Matrix represents the dominance of each one of those modes across the total imageset, the V matrix corresponds to how each of the faces projects onto each mode.

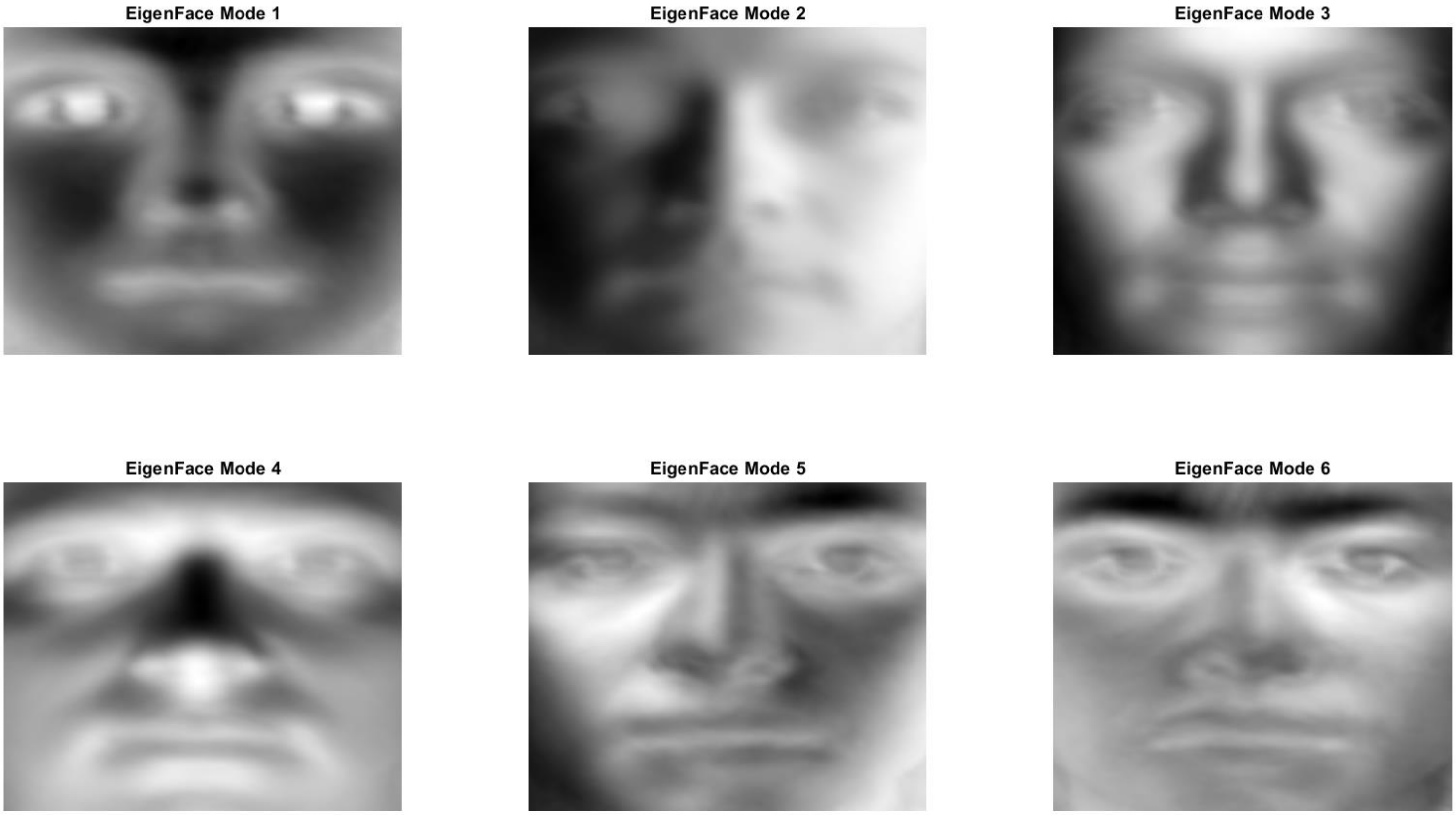
3. Answered via commentary throughout the images below.

4. Answered via commentary throughout the images below.





I note here that the rate of decay of the singular values seems to be somewhat similar for both the cropped and uncropped images, which is a slightly unexpected result. However, we do see that the cropped images have a much more dominating first mode than the uncropped images. What I believe is happening is that the rank space of the faces is similar between the cropped and uncropped images, but the amount of energy captured in the modes is distributed differently between the two regimes. In the case of cropped images, the first mode seems to contain a huge portion of the total energy, whereas in the uncropped images, the energy is a little bit more spread out. The first six modes for the cropped faces are shown below.



Shown below is the reconstruction of all the cropped faces from the first “r” eigenface modes. The rank space for the cropped images appears to be somewhere 100-150. With 100 modes capturing the majority of the facial structure and the next 50 capturing very specific details such as zits on a face.

Rank 25 approx.

Full Image

Rank 25 approx.

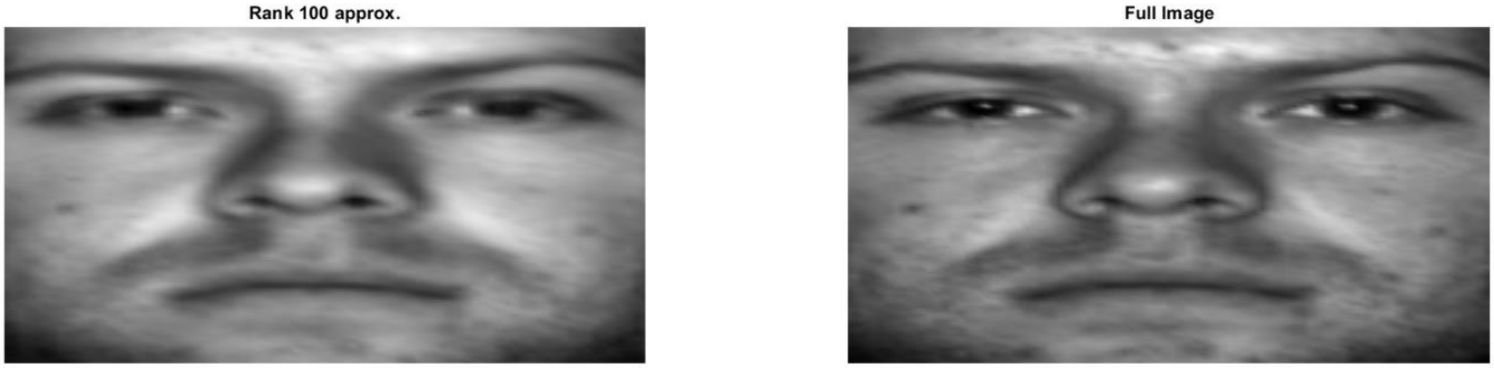
Full Image

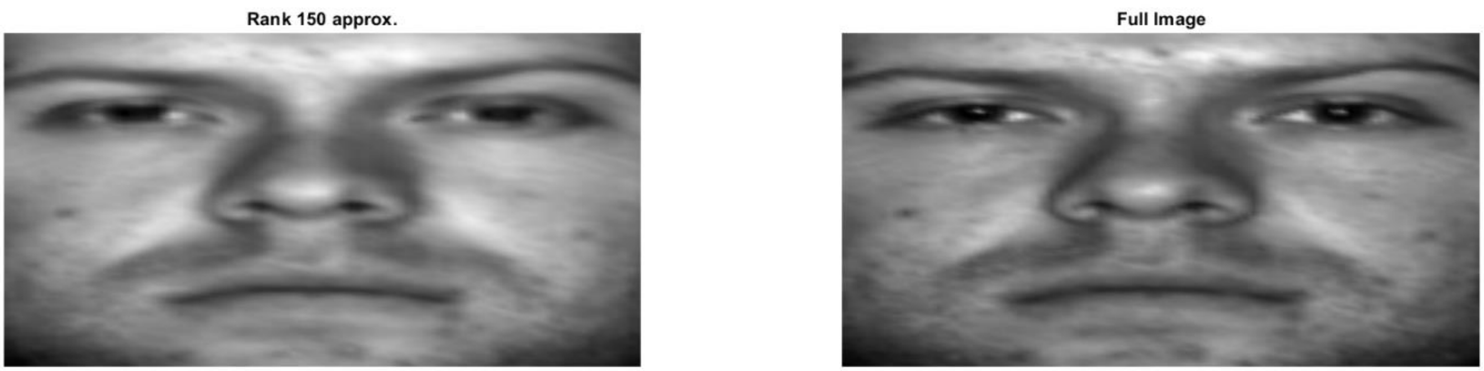
Rank 50 approx.

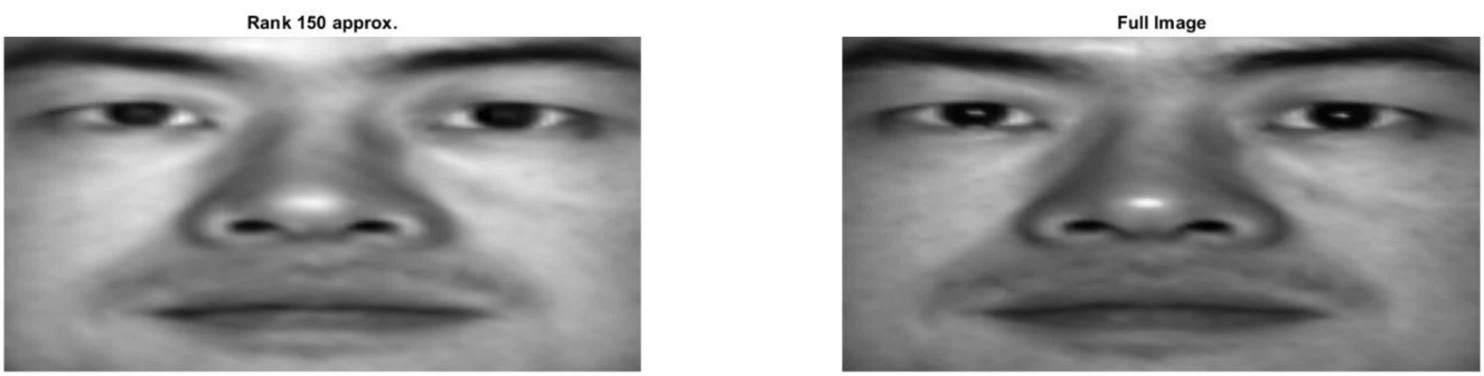
Full Image

Rank 50 approx.

Full Image







The first six modes of the uncropped images are shown below. Notice how much less clear these modes are than for the cropped images. For the cropped images, these tended to be recognizable facial structures, for the uncropped images, these tend to be more a blend of faces at different angles.

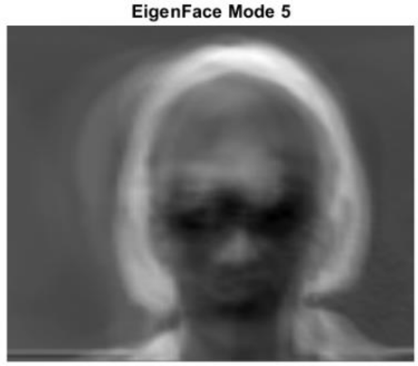
EigenFace Mode 1

EigenFace Mode 2

EigenFace Mode 3

EigenFace Mode 4

EigenFace Mode 5



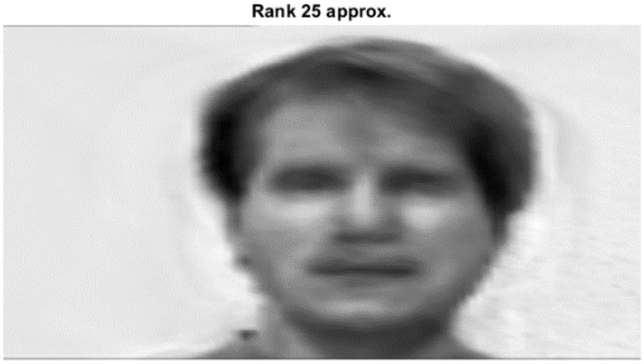
EigenFace Mode 6

Shown below is the reconstruction of all the uncropped faces from the first “r” eigenface modes. The rank space for the cropped images appears to be much more closer to 150 than for the cropped images. Interestingly enough, the background coloring clearly falls somewhere within modes 100-150.

Rank 25 approx.

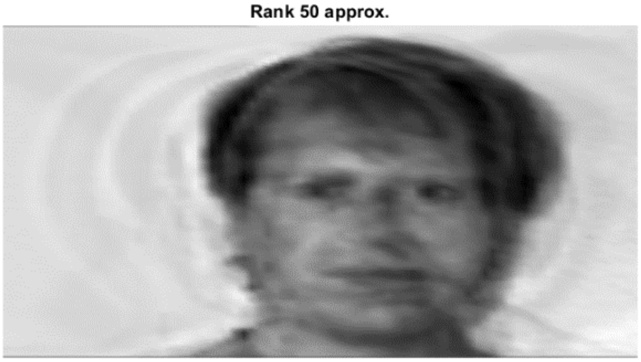
Full Image

Rank 25 approx.



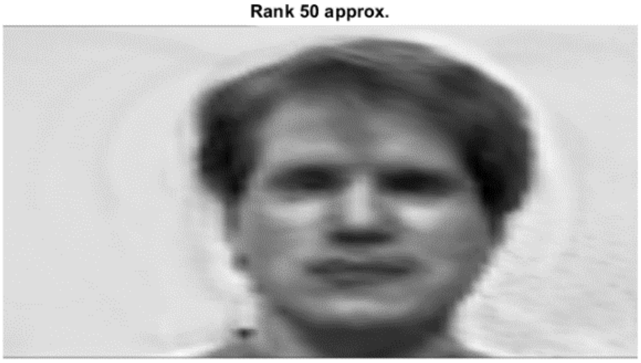
Full Image

Rank 50 approx.



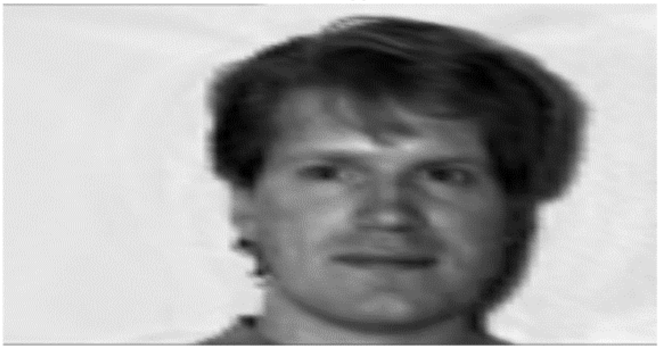
Full Image

Rank 50 approx.

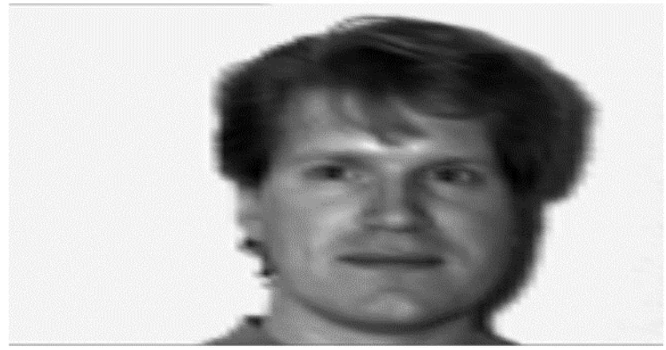


Full Image

Rank 100 approx.



Full Image



Rank 100 approx.

Full Image

Rank 150 approx.

Full Image

Rank 150 approx.

Full Image

```
clear all; close all; clc;

%% Cropped Images
---------------------------------------------------------------------
% mainfolder = 'C:
\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces_cropped\CroppedYale';
% DTest = dir(mainfolder);
% DTest(1).name;
%z_pic = imread("C:
\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces_cropped\CroppedYale\yaleB01\ya
leB01_P00A+000E+00.pgm","pgm");
z_pic = imresize(imread("C:
\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces_cropped\CroppedYale\yaleB01\ya
leB01_P00A+000E+00.pgm","pgm"),[120 80]);
%imshow(ans)


% Reading in images
count = 0;
%AllImages = zeros(size(z_pic,1),size(z_pic,2),2432);
%ImagesColumns = zeros(size(z_pic,1)*size(z_pic,2),1);

%Run once to read in the data and save to a .mat file to avoid slowdowns
%when rerunning code.
% for j = 1:39
%     if j == 14
%         continue
%     end
%     temp = "";
%     if j < 10
%         temp = "0";
%     end
%     folder = "yaleB" + temp;
%     folder = folder + j;
%     currentfolder = 'C:
\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces_cropped\CroppedYale\' +
folder;
%     D = dir(currentfolder);
%     for k = 3:66
%         strName = D(k).name;
%         count = count+1;
%         %str_trim = strName(1:strfind(strName,'.')-1);
%         currentfile = currentfolder + "\" + strName;
%         A = imresize(double(imread(currentfile,"pgm")),[120 80]);
%         AllImages(:,:,count) = A;
%         %C = reshape(imresize(double(imread(currentfile,"pgm")),[120 80]),size
(z_pic,1)*size(z_pic,2),1);
%         %ImagesColumns = [ImagesColumns,C];
%
```

```
%     end
% end
%save AllImages to AllImagesStacked.mat


%Run once to reshape data into matrix and save to .mat file to avoid
%slowdowns when running code.
% load("AllImagesStacked");
% z_A = AllImages(:,:,1);
% C_ColumnPic = zeros(size(z_A,1)*size(z_A,2),2432);
% for j = 1:2432
%     A = AllImages(:,:,j);
%     A = reshape(A,size(A,1)*size(A,2),1);
%     C_ColumnPic(:,j) = A;
% end


% load("PicsInColumns");
% Corr_Matrix = C_ColumnPic*C_ColumnPic';


% load("CorrelationMatrix")
% [U,S,V] = svd(Corr_Matrix,'econ');
% [v_eigs,d_eigs] = eigs(Corr_Matrix,20,'lm');

% Avg Faces calculation
% load("PicsInColumns");
% AvgFaces = zeros(size(C_ColumnPic,1),size(C_ColumnPic,2)/64);
% for j = 1:38
%     temp = zeros(size(C_ColumnPic,1),1);
%     for k = 0:63
%         temp = temp + C_ColumnPic(:,j+k);
%     end
%     temp = temp*(1/64);
%     AvgFaces(:,j) = temp;
% end

% Uncomment here ------------------------------------
% Pulling just one copy of each face
load("PicsInColumns");
SingleFaces = zeros(size(C_ColumnPic,1),size(C_ColumnPic,2)/64);
for j = 1:38
    SingleFaces(:,j) = C_ColumnPic(:,(j-1)*64+1);
end


load("eigenDecomp")
load("SVD")
S_prime = diag(S);
```

```
S_prime = S_prime(1:100,1);
f1 = figure;
plot(S_prime,'r.');
f2 = figure;
semilogy(S_prime,'r.');

numEigenfaces = 150;
U_firstTwenty = U(:,1:numEigenfaces);
U_firstSix = U(:,1:6);
f3 = figure;

for k = 1:size(U_firstSix,2)
    U_small = U_firstSix(:,k);
    U_small=reshape(U_small,size(z_pic,1),size(z_pic,2));
    subplot(2,3,k)
    pcolor(flipud(U_small))
    shading interp
    colormap gray
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("EigenFace Mode "+ string(k))
end

f5 = figure;
ProjectionsValues = SingleFaces'*U_firstTwenty;
RankTwentyApproxImages = U_firstTwenty*ProjectionsValues';
countPlotted = 0;
for k = 1:2
    Approx = reshape(RankTwentyApproxImages(:,k),size(z_pic,1),size(z_pic,2));
    Full = reshape(SingleFaces(:,k),size(z_pic,1),size(z_pic,2));
    countPlotted = countPlotted + 1;
    subplot(2,2,countPlotted)
    pcolor(flipud(Approx))
    shading interp
    colormap gray
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("Rank " + string(numEigenfaces) +" approx.")

    countPlotted = countPlotted + 1;
    subplot(2,2,countPlotted)
    pcolor(flipud(Full))
    shading interp
    colormap gray
    set(gca,'xtick',[])
```

```
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("Full Image")

end


% ----------------------------------------------------


%% Uncropped images ----------------------------------
z_pic = imresize(imread("C:
\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces\subject01.centerlight","gif"),
[120 80]);
%imshow(z_pic)


% Reading in images
count = 0;
AllImages2 = zeros(size(z_pic,1),size(z_pic,2),165);

%Read in images and save in 3-d matrix
D = dir("C:\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces");
for k = 3:167
    strName = D(k).name;
    count = count+1;
    currentfile = "C:\Users\Karl\Documents\Documents\AMATH584\hw2\yalefaces\" +
strName;
    A = imresize(double(imread(currentfile,"gif")),[120 80]);
    AllImages2(:,:,count) = A;
end


z_A = AllImages2(:,:,1);
C_ColumnPic2 = zeros(size(z_A,1)*size(z_A,2),165);
for j = 1:165
    A = AllImages2(:,:,j);
    A = reshape(A,size(A,1)*size(A,2),1);
    C_ColumnPic2(:,j) = A;
end


% Corr_Matrix2 = C_ColumnPic2*C_ColumnPic2';


% [U2,S2,V2] = svd(Corr_Matrix2,'econ');
% [v_eigs2,d_eigs2] = eigs(Corr_Matrix2,40,'lm');
```

```
load("eigenDecomp2")
load("SVD2")
S_prime = diag(S2);
S_prime = S_prime(1:100,1);
figure(f1);
hold on
plot(S_prime,'b.');
title("Singular Values Plotted on Absolute Scale")
xlabel("Singular Value Index")
ylabel("Singular Value")
legend("Cropped","Uncropped")
figure(f2);
hold on
semilogy(S_prime,'b.');
title("Singular Values Plotted on Log Scale")
xlabel("Singular Value Index")
ylabel("Singular Value")
legend("Cropped","Uncropped")

numEigenfaces = 150;
U_firstTwenty = U2(:,1:numEigenfaces);
U_firstSix = U2(:,1:6);
f4 = figure;

for k = 1:size(U_firstSix,2)
    U_small = U_firstSix(:,k);
    U_small=reshape(U_small,size(z_pic,1),size(z_pic,2));
    subplot(2,3,k)
    pcolor(flipud(U_small))
    shading interp
    colormap gray
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("EigenFace Mode "+ string(k))
end

f6 = figure;
ProjectionsValues = C_ColumnPic2'*U_firstTwenty;
RankTwentyApproxImages = U_firstTwenty*ProjectionsValues';
countPlotted = 0;
for k = 1:2
    Approx = reshape(RankTwentyApproxImages(:,k),size(z_pic,1),size(z_pic,2));
    Full = reshape(C_ColumnPic2(:,k),size(z_pic,1),size(z_pic,2));
    countPlotted = countPlotted + 1;
    subplot(2,2,countPlotted)
    pcolor(flipud(Approx))
```

```
    shading interp
    colormap gray
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("Rank " + string(numEigenfaces) +" approx.")

    countPlotted = countPlotted + 1;
    subplot(2,2,countPlotted)
    pcolor(flipud(Full))
    shading interp
    colormap gray
    set(gca,'xtick',[])
    set(gca,'xticklabel',[])
    set(gca,'ytick',[])
    set(gca,'yticklabel',[])
    title("Full Image")

end
```