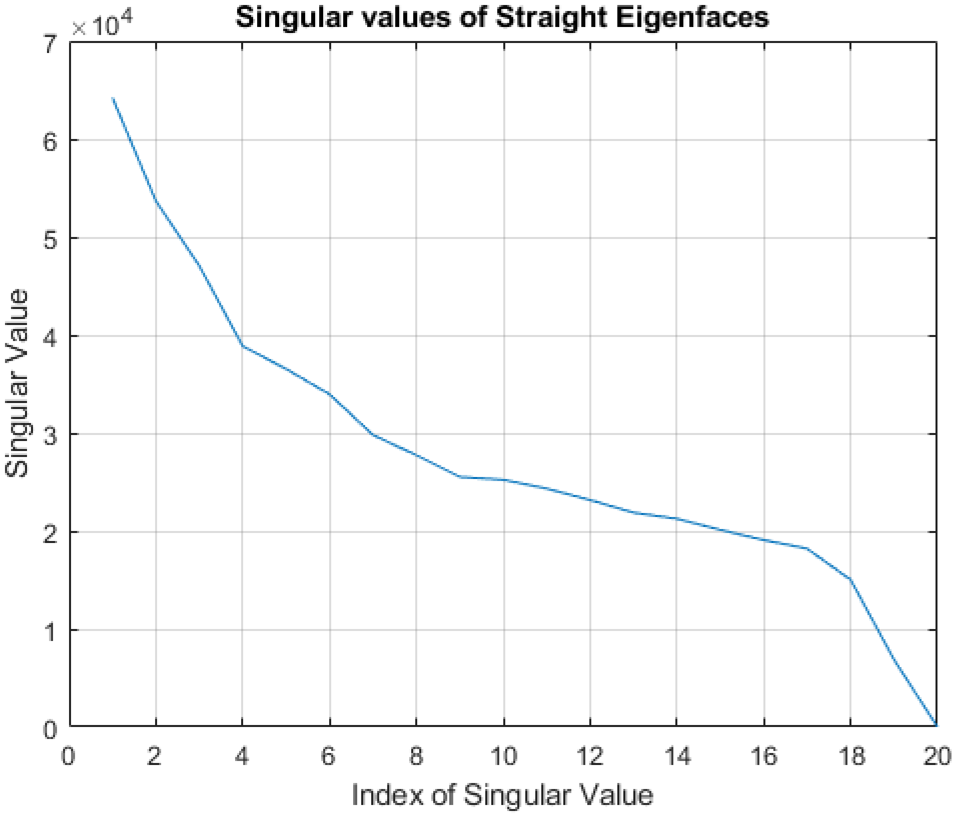
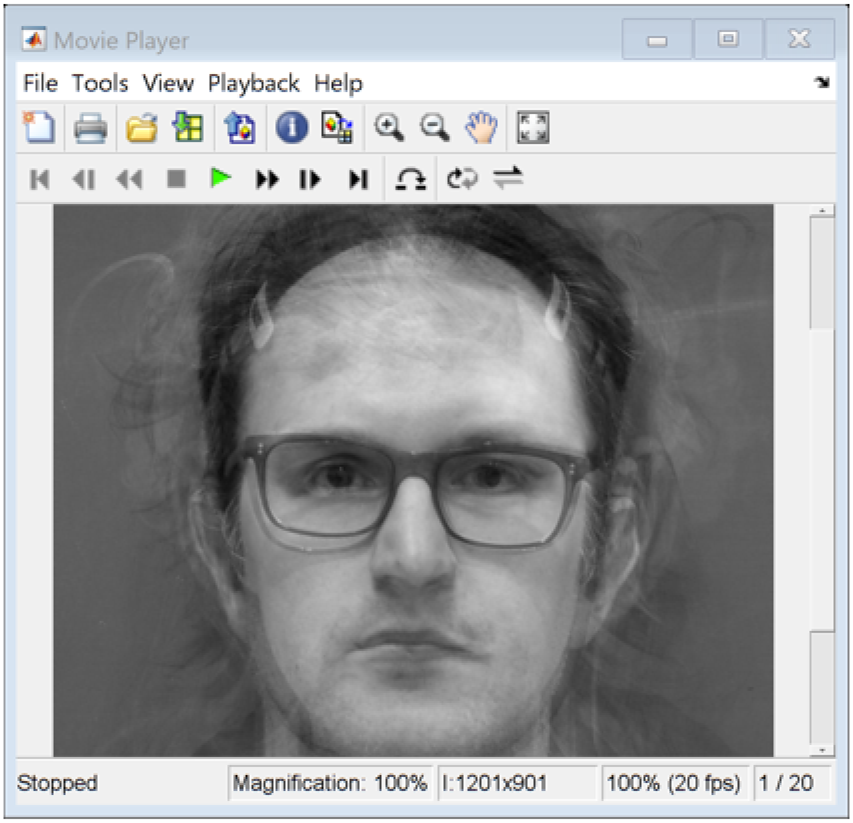
```
close all;
clear;
clc;
% Max Cusick, Tomas Collado, Claudia Markel, Natalia Klim

% I found the method for reading in images here, lets be sure to cite it in
 the report: https://www.mathworks.com/matlabcentral/answers/396955-read-all-
images-in-directory
% Read in all images
imagesR = dir('dataResting\*.jpg');

%Initial values
n = length(imagesR);           % Number of files found
p = 1201*901;                  % Number of pixels per image
A = ones(p, n);                % Initial matrix the size of each image as a
 column by the number of images
numBasis = 8;                 % Number of basis used to reconstruct images

%Loop to read in all images to matrix A
for i = 1:n
    current_image = imagesR(i).name;                            % Get name
 current image
    current_image = imread(['dataResting\' current_image]);     % Retrieve the
 current image from folder
    img = im2gray(current_image);                               % Convert
 image to grayscale
    imgCol = img(:);                                            % Convert
 image to column vector
    A(:,i) = imgCol;                                            % Put image
 column into matrix A
end

%Find mean of A
mR = mean(A,2);

%Calculate eigenvectors and eigenvalues
[U,S,V] = svd(A - mR,'econ');

%Create plot of singular values
figure
plot(diag(S));
hold on;
title('Singular values of Straight Eigenfaces');
xlabel('Index of Singular Value');
ylabel('Singular Value');
grid on;
hold off;

%Reconstruct original images
IMeigen = U(:, [1:numBasis])*S(1:numBasis, 1:numBasis)*V(:, [1:numBasis])';  %
 apply the selected basis to the images
```

```
IM = IMeigen + mR;                                                          %
 add the mean back to the images
IM_2D = reshape(IM, 1201, 901, []);                                         %
 reshape the images

%Show all the Eigenfaces
figure
implay(IM_2D/255)




% Read in all Smiling images
imagesS = dir('dataSmiling\*.jpg');

%Initial values
n = length(imagesS);           % Number of files found
p = 1201*901;                  % Number of pixels per image
A = ones(p, n);                % Initial matrix the size of each image as a
 column by the number of images
numBasis = 8;                 % Number of basis used to reconstruct images

%Loop to read in all images to matrix A
for i = 1:n
    current_image = imagesS(i).name;                            % Get name
 current image
    current_image = imread(['dataSmiling\' current_image]);     % Retrieve the
 current image from folder
    img = im2gray(current_image);                               % Convert
 image to grayscale
    imgCol = img(:);                                            % Convert
 image to column vector
    A(:,i) = imgCol;                                            % Put image
 column into matrix A
end

%Find mean of A
mS = mean(A,2);

%Calculate eigenvectors and eigenvalues
[U,S,V] = svd(A - mS,'econ');

%Plot V value for first Eigenface
figure
plot(V(1,:));
hold on;
title('V values of First Reconstructed Smiling Eigenface');
grid on;
hold off;

%Create plot of singular values
figure
plot(diag(S));
hold on;
```

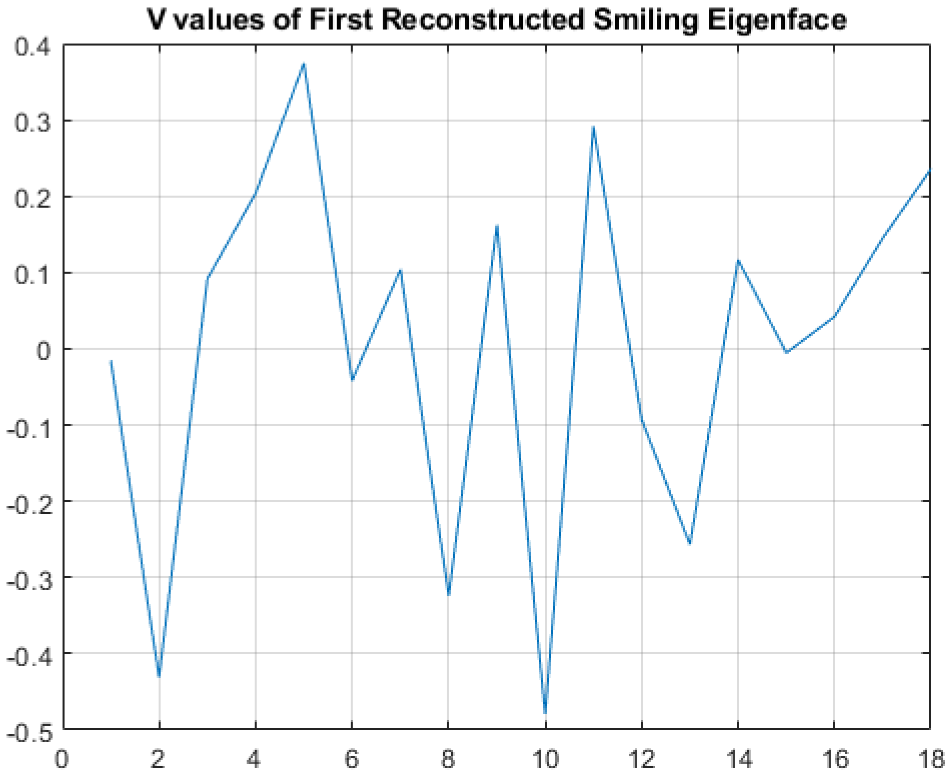
```
title('Singular values of Smiling Eigenfaces');
xlabel('Index of Singular Value');
ylabel('Singular Value');
grid on;
hold off;

%Reconstruct original images
IMeigen = U(:, [1:numBasis])*S(1:numBasis, 1:numBasis)*V(:, [1:numBasis])';  %
 apply the selected basis to the images
IM = IMeigen + mS;                                                           %
 add the mean back to the images
IM_2D = reshape(IM, 1201, 901, []);                                          %
 reshape the images

%Show all the Eigenfaces
figure
implay(IM_2D/255)




% Read in all Frowning images
imagesF = dir('dataFrowning\*.jpg');

%Initial values
n = length(imagesF);           % Number of files found
p = 1201*901;                  % Number of pixels per image
A = ones(p, n);                % Initial matrix the size of each image as a
 column by the number of images
numBasis = 8;                 % Number of basis used to reconstruct images

%Loop to read in all images to matrix A
for i = 1:n
    current_image = imagesF(i).name;                            % Get name
 current image
    current_image = imread(['dataFrowning\' current_image]);    % Retrieve the
 current image from folder
    img = im2gray(current_image);                               % Convert
 image to grayscale
    imgCol = img(:);                                            % Convert
 image to column vector
    A(:,i) = imgCol;                                            % Put image
 column into matrix A
end

%Find mean of A
mF = mean(A,2);

%Calculate eigenvectors and eigenvalues
[U,S,V] = svd(A - mF,'econ');

%Plot V value for first Eigenface
figure
plot(V(1,:));
```

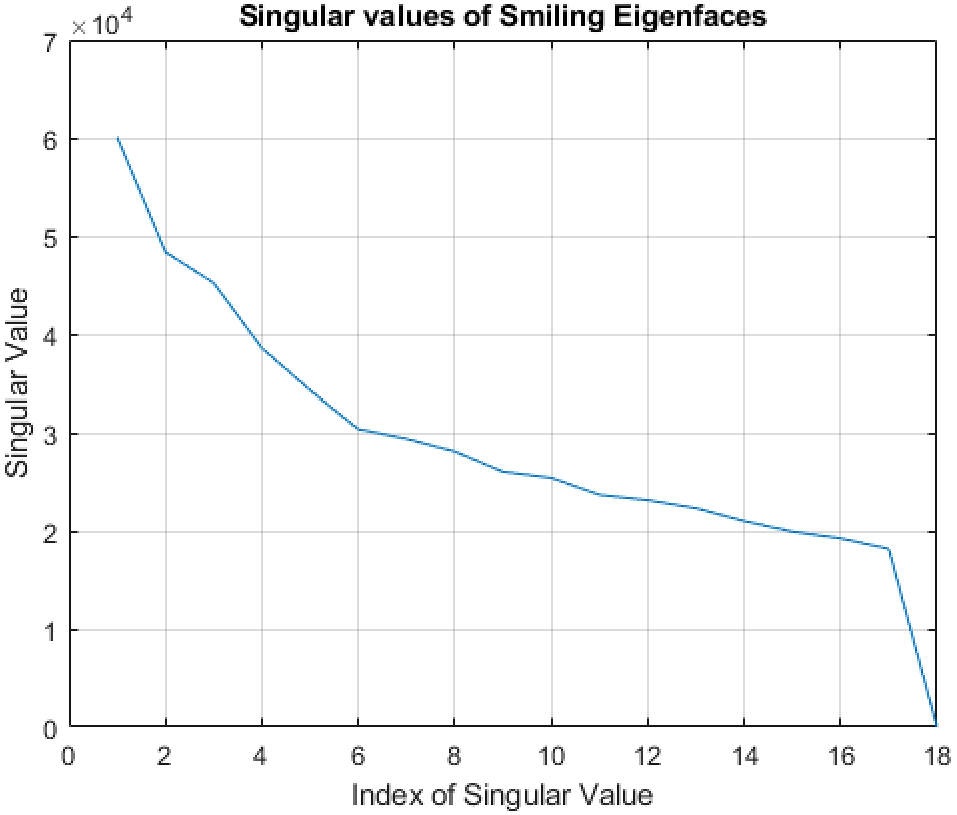
```
hold on;
title('V values of First Reconstructed Frowning Eigenface');
grid on;
hold off;

%Create plot of singular values
figure
plot(diag(S));
hold on;
title('Singular values of Frowning Eigenfaces');
xlabel('Index of Singular Value');
ylabel('Singular Value');
grid on;
hold off;

%Reconstruct original images
IMeigen = U(:, [1:numBasis])*S(1:numBasis, 1:numBasis)*V(:, [1:numBasis])';  %
 apply the selected basis to the images
IM = IMeigen + mF;                                                            %
 add the mean back to the images
IM_2D = reshape(IM, 1201, 901, []);                                           %
 reshape the images

%Show all the Eigenfaces
figure
implay(IM_2D/255)
```

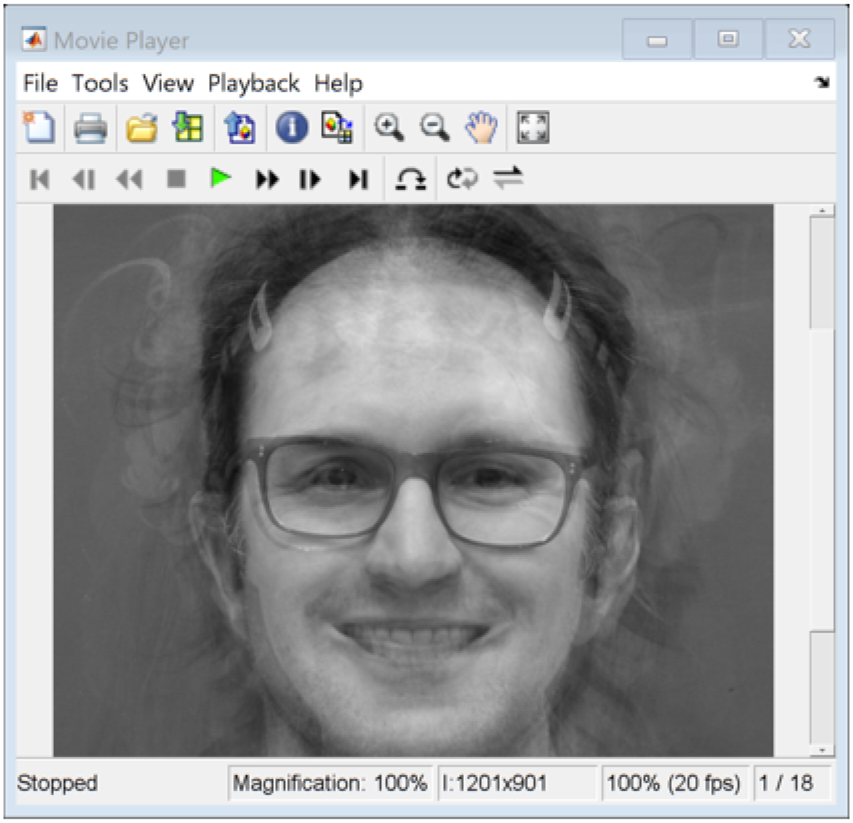




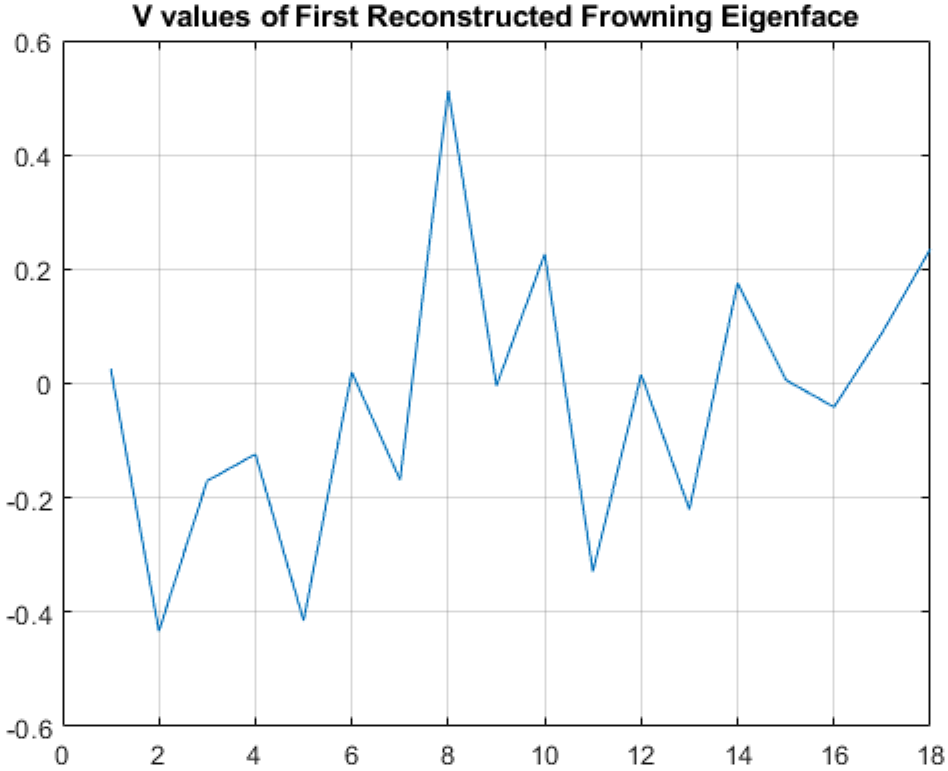
Stopped | Magnification: 100% | I:1201x901 | 100% (20 fps) | 1 / 20

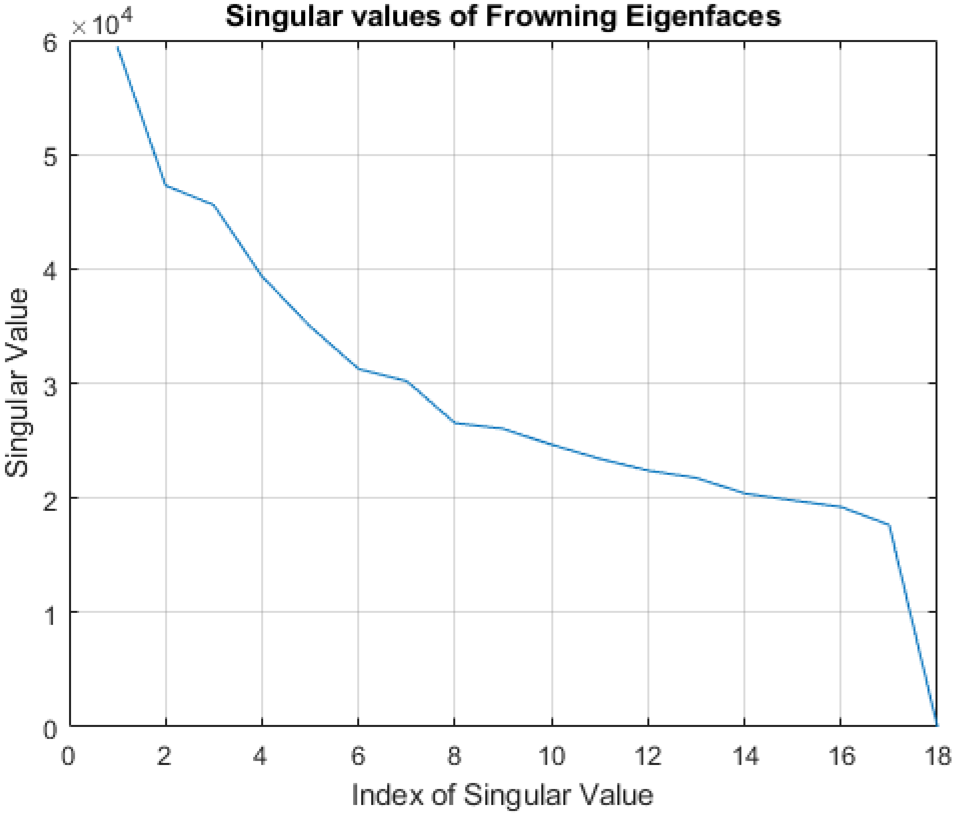

V values of First Reconstructed Smiling Eigenface
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
2
4
6
8
10
12
14
16
18

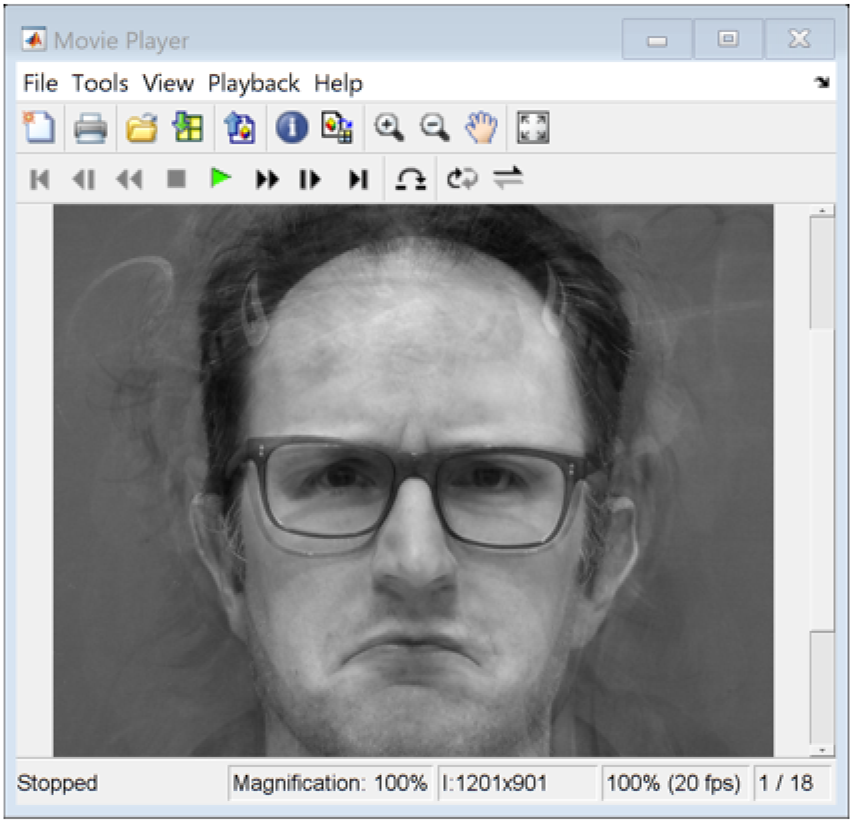

Singular values of Smiling Eigenfaces
$\times 10^4$
7
6
5
4
3
2
1
0
Singular Value
0
2
4
6
8
10
12
14
16
18
Index of Singular Value



Stopped | Magnification: 100% | I:1201x901 | 100% (20 fps) | 1 / 18


V values of First Reconstructed Frowning Eigenface
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
2
4
6
8
10
12
14
16
18


Singular values of Frowning Eigenfaces
× 10^4
6
5
4
3
2
1
0
Singular Value
0
2
4
6
8
10
12
14
16
18
Index of Singular Value



Movie Player
File Tools View Playback Help
Stopped
Magnification: 100%
I:1201x901
100% (20 fps)
1 / 18

Published with MATLAB® R2022b