AMATH 584 Homework 2

Molly Wieringa

October 21, 2020

Abstract

We have been asked to perform and interpret a singular value decomposition on a series of cropped and uncropped images. We find that good image reconstruction can be achieved with rank 100 in the face space (100 eigenmodes) for cropped images and with rank 500 in the face space for uncropped images. This implies that reconstruction that take care to align and crop images before hand are more efficient.

1 Yale Faces B

Having been provided with two datasets, the first a set of cropped and aligned photographs of human faces, and the second a set of corresponding uncropped images, we performed a singular value decomposition (please see "Matlab Code" section below for reference).

The decomposition produces 3 matrices:

- **U**, the "left singular vectors" or principle directions. Essentially, **U** holds the dominant patterns, or "eigenfaces" of the matrix **A** (Figures 1 and 2).
- Σ, a diagonal matrix with the singular values of the the decomposition ordered largest to smallest along the diagonal; we can think of these values as the weights of each eigenface in **U**, i.e. the relative importance of each eigenface.
- **V**, the "right singular vectors" which communicate between **U** and Σ, telling us how the weighted eigenfaces project on to **A**.

In order to ascertain the dominant modes for these photographs, we plotted the singular value spectra for both the covariance matrices of both the cropped images and their uncropped counterparts, isolating the singular values by decomposing $cov(\mathbf{A})$ and taking $diag(\Sigma)$ (Figure 3).

We also reconstructed images using a number of truncated rank values, to test the quality of our reconstructions based on the number of modes used in the reconstruction process. We find that decent reconstructions are possible for the cropped images when at least 100 modes are included; more are necessary for equal quality of the uncropped image reconstructions (Figure 4).

1.1 Results: Cropped vs Uncropped

The singular values for both the cropped image set decays much more slowly than the uncropped set, though the singular values for the cropped images are in general much lower initially (Figure 3). Relatedly, the cropped images produce higher quality reconstructions using fewer eigenmodes than the unaligned and uncropped images.

2 Theorems

- The nonzero singular values of A are the square roots of the nonzero eigenvalue of AA^* or A^*A

We can write AA^* as $(U\Sigma V^*)(U\Sigma V^*)^*$ and A^*A as $(U\Sigma V^*)^*(U\Sigma V^*)$. Taking the appropriate transposes results in:

(1) $AA^* = (U\Sigma V^*)(V\Sigma^* U^*)$
(2) $A^*A = (V\Sigma^* U^*)(U\Sigma V^*)$

The inside products, V^*V and U^*U, are equal to the identity, leaving:

(3) $AA^* = (U\Sigma\Sigma^* U^*)$
(4) $A^*A = (V\Sigma^*\Sigma V^*)$

Because Σ is a diagonal matrix (by definition of the SVD), $\Sigma\Sigma^*$ and $\Sigma^*\Sigma$ are simply Σ^2. Multiplying both sides of equations (3) and (4) by U, we see that:

(5) $AA^*U = U\Sigma^2$

Recognizing that this is a form of the eigenvalue problem $AX = X\Lambda$, we see that $\Sigma^2 = \Lambda$, or $\Sigma = \sqrt{\Lambda}$. QED.

- If $A = A^*$, then the singular values are the absolute values of the eigenvalues of A.

  If $A = A^*$, then A is a Hermitian matrix, and as such, all its eigenvalues are real (proved in the last homework).

  We perform an eigenvalue decomposition of A, such that

  (1) $A = U\Lambda U^*$

  If we rewrite the matrix Λ as $|\Lambda|P$, where P represents a diagonal matrix whose entries are equal to 1 or -1 (depending upon the value of the corresponding entry of Λ, we then have:

  (2) $A = U|\Lambda|PU^*$

  Noting that PU^* is a unitary matrix in its own right, we can recognize equation (2) as a singular value decomposition of the matrix A. As such, the diagonal values of $|\Lambda|$ are the singular values of the SVD, and also the absolute values of the eigenvalues of A. QED.

- Given that the determinant of a matrix U is unity, show $|det(A)| = \Pi_{j=1}^{m}\sigma_j$.

  Taking the SVD of matrix A, we can write:

  (1) $|det(A)| = |det(U\Sigma V^*)| = |det(U)det(\Sigma)det(V^*)|$

  Since U and V^* are both unitary matrices (by definition of the SVD), their determinants are equal to 1. We're therefore left with:

  (2) $|det(A)| = |det(\Sigma)|$

Because Σ is a diagonal matrix, its determinant is equal to the product of the values along its diagonal:

(3) $|det(A)| = |det(\Sigma)| = \Pi_{j=1}^{m} \sigma_j$. QED.

3 Figures

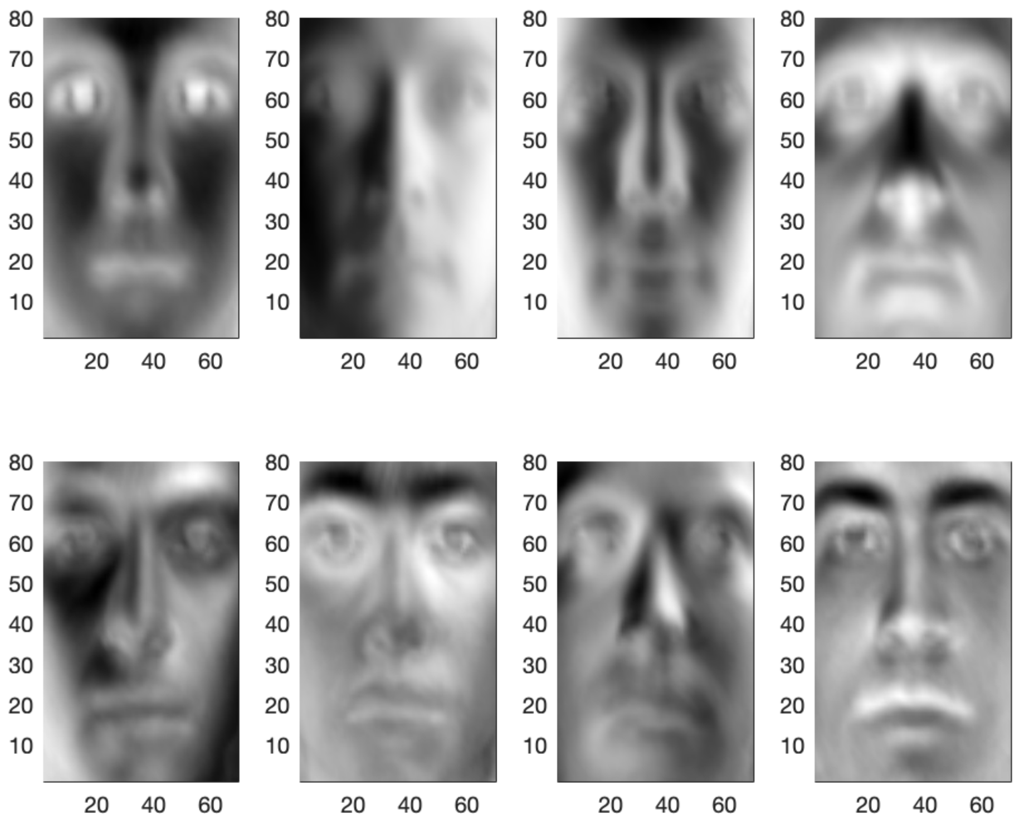


Figure 1: First eight principal directions of the cropped image set.

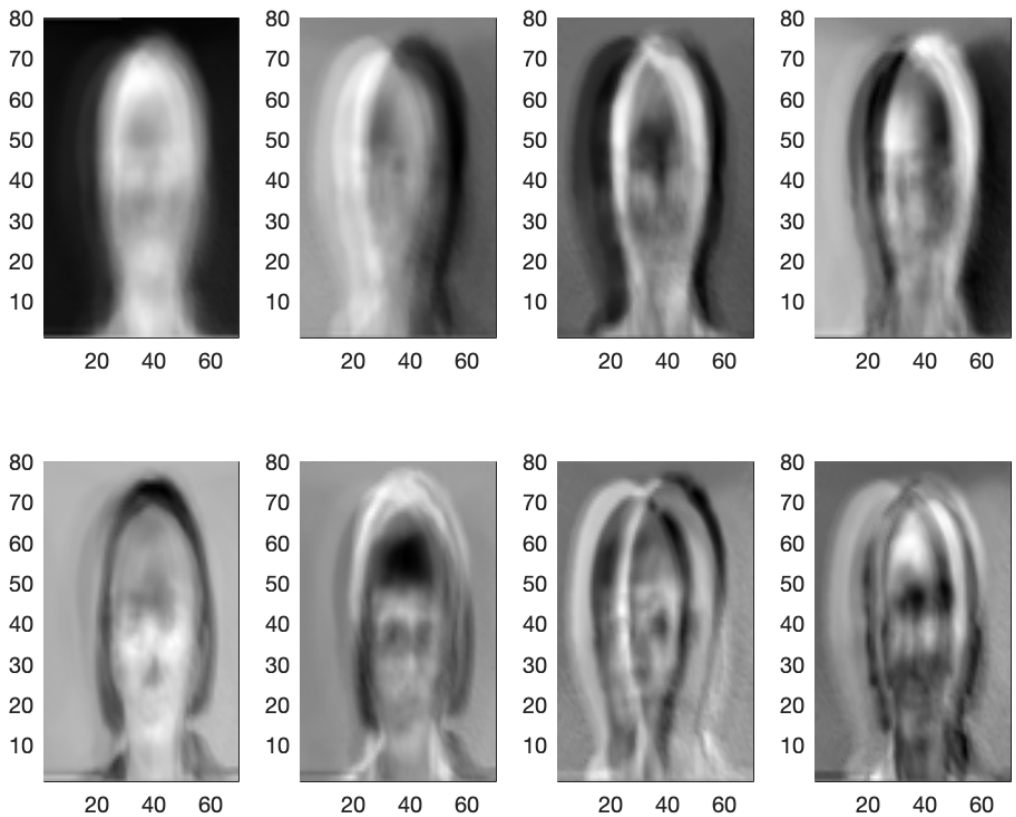


Figure 2: First eight principal directions of the uncropped image set.

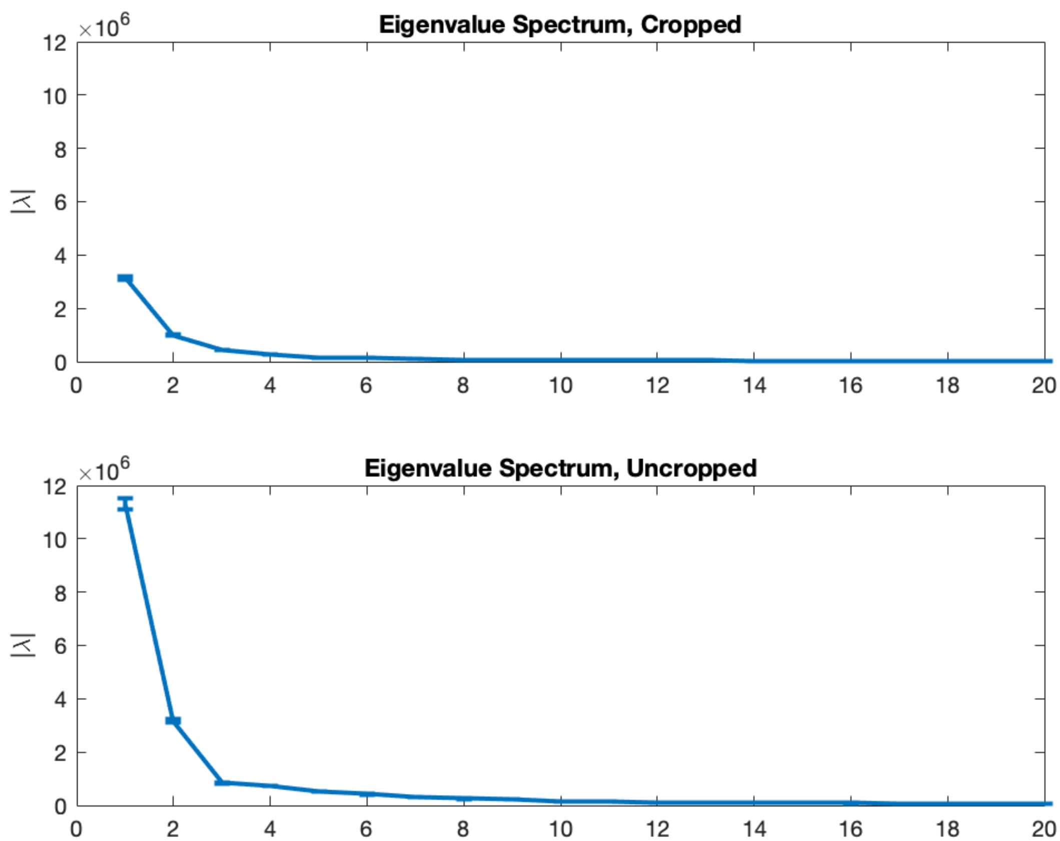


Figure 3: Top: eigenvalue spectrum of cropped images; bottom: eigenvalue spectrum of uncropped images.

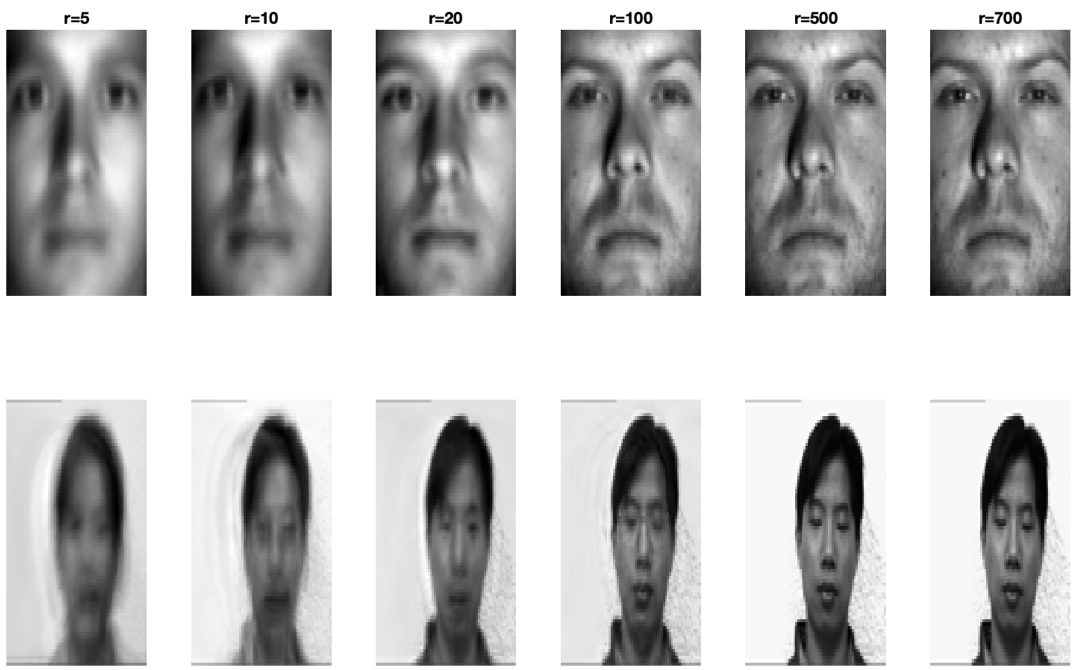
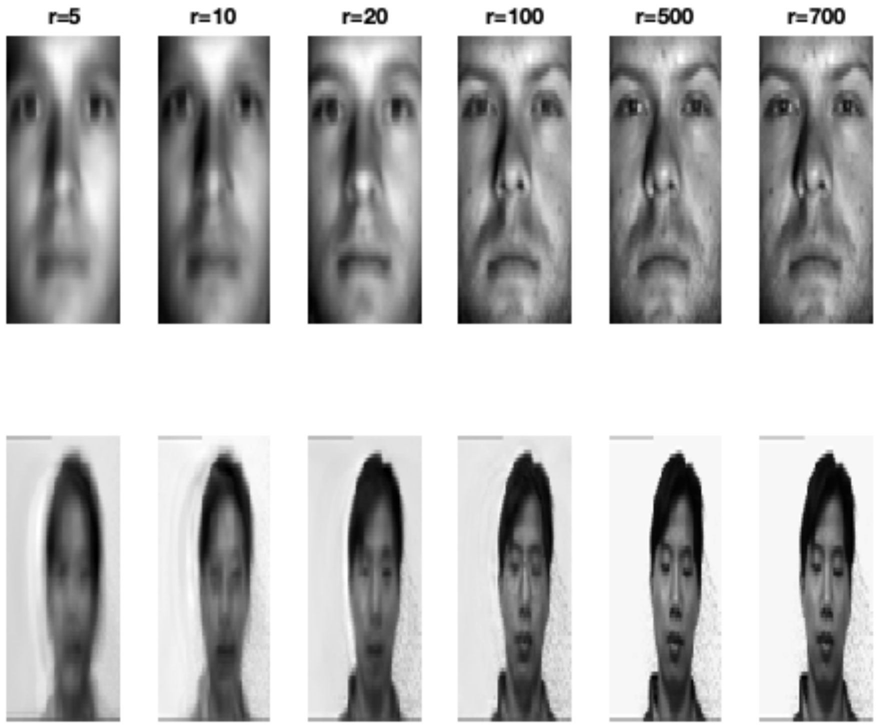


Figure 4: Photograph Reconstructions, based on truncated rank.



Published with MATLAB® R2020b

Contents

AMATH 584 Homework 2: Yale Faces B

```
clear; clc; close all
```

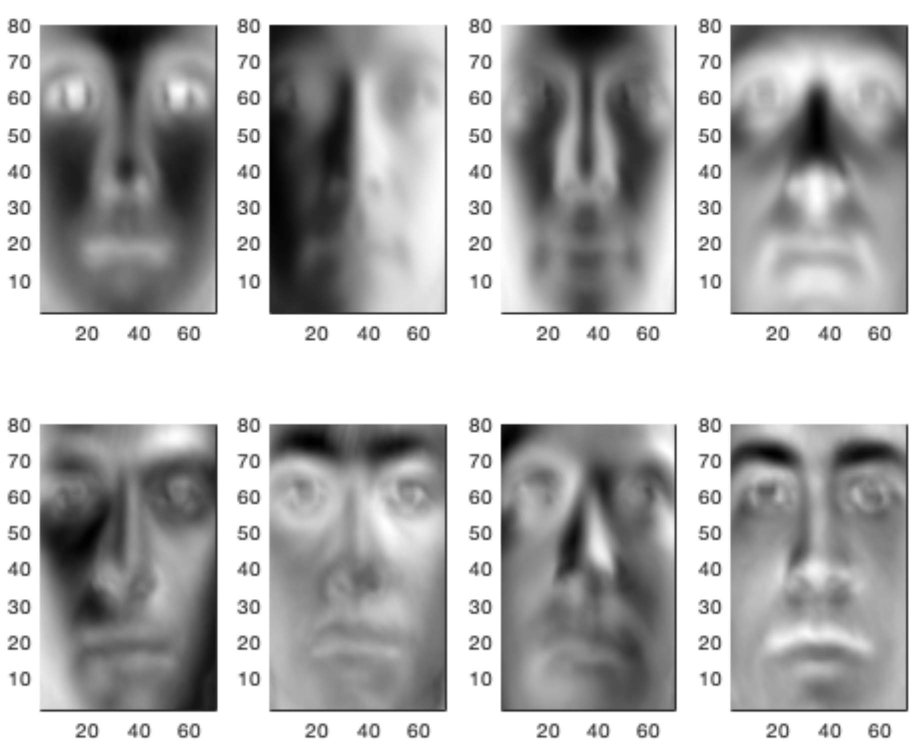
Load data

```
% Make big, emtpy data matrices
A_cropped = [];
A_uncropped = [];

% Set up image dimensions
dim1 = 80;
dim2 = 70;

% Begin file reading iteration for the cropped images
count = 1;
for i = 1:39
    path = ['CroppedYale/yaleB' num2str(i, '%02.f')];
    P = path;
    addpath(genpath(P));
    dict = dir(P);

    for j = 1:numel(dict)
        file = dict(j).name;
        if length(file) > 3
            image = imread(file);
            data = imresize(double(im2gray(imread(file))), [dim1,dim2]);
            A_cropped(:,count) = data(:)';
            count = count + 1;
        end
    end
end

% Begin the file reading iteration for the uncropped images

%expressions = ['centerlight','glasses','happy','leftlight','noglasses',...
%    'normal','rightlight','sad','sleepy','surprised','wink'];

count = 1;
for i = 1:15
    path = 'yalefaces_uncropped/yalefaces';
    P = path;
    addpath(genpath(P));
    dict = dir(P);
```

```
    for j=1:numel(dict)
        file = dict(j).name;
        if length(file) > 3
            image = imread(file);
            data = imresize(double(im2gray(imread(file))), [dim1,dim2]);
            A_uncropped(:,count) = data(:)';
            count = count + 1;
        end
    end

end
```

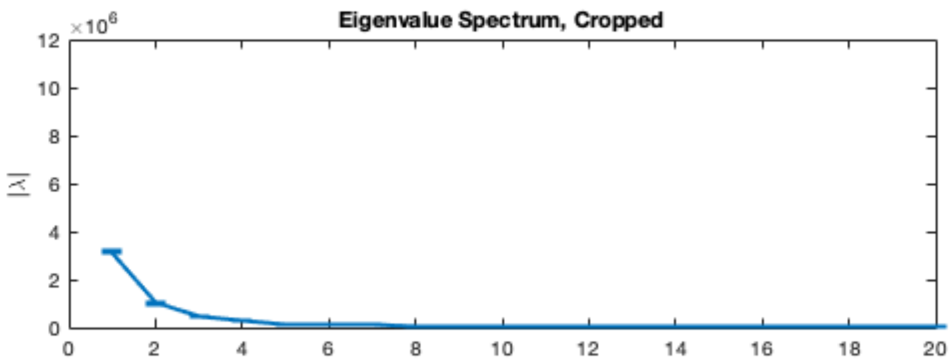
Perfrom a singular value decomposition on the cropped images

```
[U_c,S_c,V_st_c] = svd(A_cropped);

% U represents
% S represents
% V_st represents

% Plot the first few columns of U
figure(1);
for i = 1:8
    col = U_c(:, i);
    new_image = reshape(col, [dim1, dim2]);
    subplot(2,4,i);
    pcolor(flip(new_image)), shading interp, colormap gray
end
% suptitle('Eigenfaces, Cropped')
```



PCA analysis- cropped

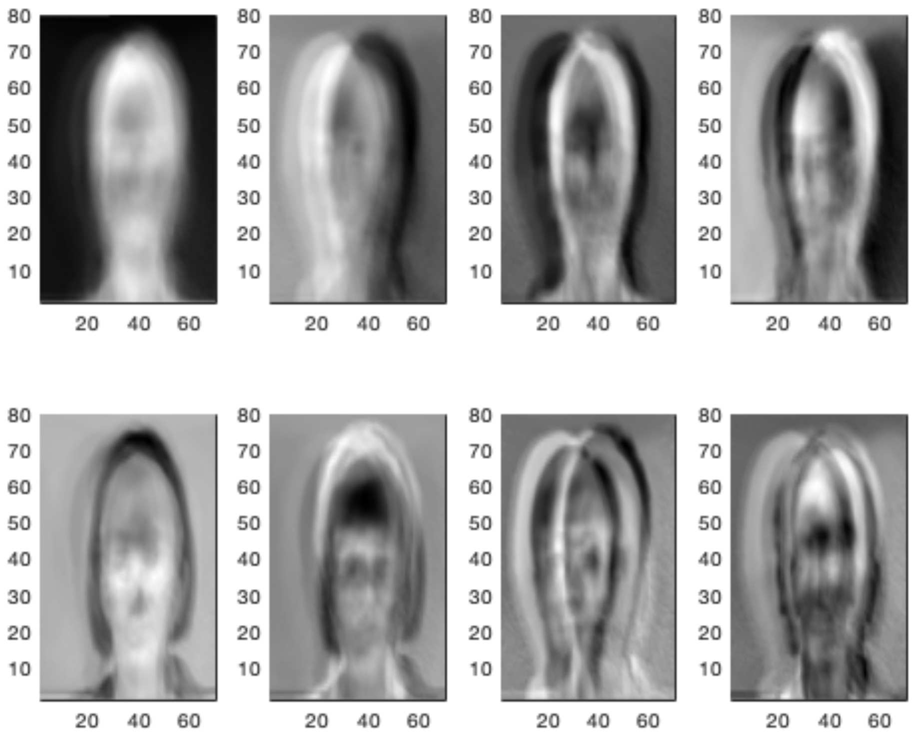
```
[~,~,Spca] = pca(A_cropped);
eigvals_c = Spca;
%eigvals_c = svd(C_c);

%N_st = trace(C_c)^2/trace(C_c.^2);

% Plot the eigenvalue spectrum
N_st = dim1*dim2;
err_c = eigvals_c*sqrt((2/N_st));

figure(2);
subplot(2,1,1);
errorbar(eigvals_c(1:20), err_c(1:20), 'linewi', 2)
ylim([0 12E6]);
ylabel('|\lambda|')
title('Eigenvalue Spectrum, Cropped')
% subplot(2,1,2);
% semilogy(eigvals_c(1:20), 'linewi', 2)
% ylim([0 10E7]);
% ylabel('|\lambda|')
```



Perform a singular value decomposition on the uncropped images

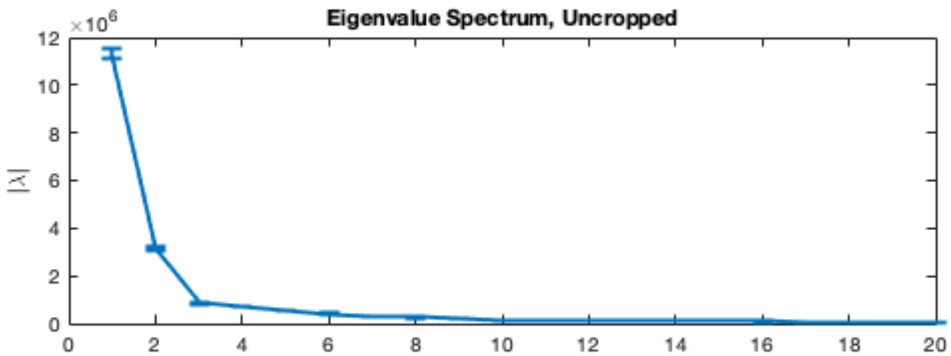
```
[U_uc,S_uc,V_st_uc] = svd(A_uncropped);

% U represents
% S represents
% V_st represents

% Plot the first few columns of U
figure(3);
```

```
for i = 1:8
    col = U_uc(:, i);
    new_image = reshape(col, [dim1, dim2]);
    subplot(2,4,i);
    pcolor(flip(new_image)), shading interp, colormap gray
end
%suptitle('Eigenfaces, Uncropped')
```



PCA analysis- uncropped

```
% Calculated SVD on the covariance matrix

[~,~,Spcau] = pca(A_uncropped);
eigvals_uc = Spcau;

N_st = dim1*dim2;
%N_st = trace(C_uc)^2/trace(C_uc.^2);

% Plot the eigenvalue spectrum
err_uc = eigvals_uc*sqrt((2/N_st));

figure(2);
subplot(2,1,2);
errorbar(eigvals_uc(1:20), err_uc(1:20), 'linewi', 2)
ylim([0 12E6])
ylabel('|\lambda|')
title('Eigenvalue Spectrum, Uncropped')
% subplot(2,1,2);
% semilogy(eigvals_uc(1:20), 'linewi', 2)
% ylim([0 10E7]);
% ylabel('|\lambda|')
```



Rank-based Reconstructions- uncropped

```
figure();
count = 1;
r = [5 10 20 100 500 700];
for ri= 1:length(r) % Truncation value
    face = 42;
     Xapprox_c  = U_c(:,1:r(ri)) * S_c(1:r(ri),1:r(ri)) * V_st_c(:,1:r(ri))';
     Xapprox_uc = U_uc(:,1:r(ri))*S_uc(1:r(ri),1:r(ri))*V_st_uc(:,1:r(ri))'; % Approx. image
    face_c = reshape(Xapprox_c(:,face), [dim1, dim2]);
    face_uc = reshape(Xapprox_uc(:,face),[dim1,dim2]);
    subplot(2,length(r),count)
    imagesc(face_c), axis off, colormap gray;
    title(['r=',num2str(r(ri),'%d') ]);
    subplot(2,length(r), count+length(r))
    imagesc(face_uc), axis off, colormap gray;
    count = count + 1;
end
```