Single Value Decomposition Applied to Image Processing

Hongyu Cao

Department of Mathematics

University of Washington

**ABSTRACT**

Singular Value Decomposition (SVD) is very powerful for processing data. Since every image consists of lots of pixels and each pixel has a color which could be expressed as a number, SVD is also an efficient way to process different types of image. This report provides a basic introduction of SVD, PCA and some specific examples of application of analyzing Yalefaces.

**I. Introduction**

All of the images are in the folder called yalefaces and we have cropped and uncropped images. I Loaded all of the images with ending ‘pgm’ in ‘yalefaces\_croped’ and transferred the type of each pixel into double so we can get a matrix for each image. Then reshaped each image matrix into a single column and combined them together into one single matrix that has a dimension (I skipped the images in the last file since my Matlab can’t read them). Calculated the mean for each row and we can get an average face (show in next section). Subtracted mean from all the data because we don’t want redundant data which depicted human’s face common feature. After that, we can do a SVD of this matrix and get U, S, V. U is a basis of the faces, S is the stretch and compression and V is linear combination. SVD also realized PCA by changing the basis of these faces and there’s no redundant data. Also each image can be also reconstructed by the new basis and in the next section I will show some plots of modes.

**II. Theoretical Background**

A *singular value decomposition* (SVD) is a factorization of a matrix into a number of constitutive components all of which have a specific meaning in applications [1]. The SVD is essentially a transformation that stretches/compresses and rotates a given set of vectors [1]. It always exists for all matrices so it’s a very stable way to decompose matrix. An matrix A can be decomposed as the following:

(2.1)

U is a matrix with orthonormal column vectors and means rotation. V is a unitary matrix and also means rotation. is the singular value and is a diagonal matrix and its diagonal from top left to bottom right is . It means stretch/compress. The following pictures provides a specific example:

**III. Algorithm Implementation and Development**

Matlab already has SVD command so there’s no new algorithm that I need to build.

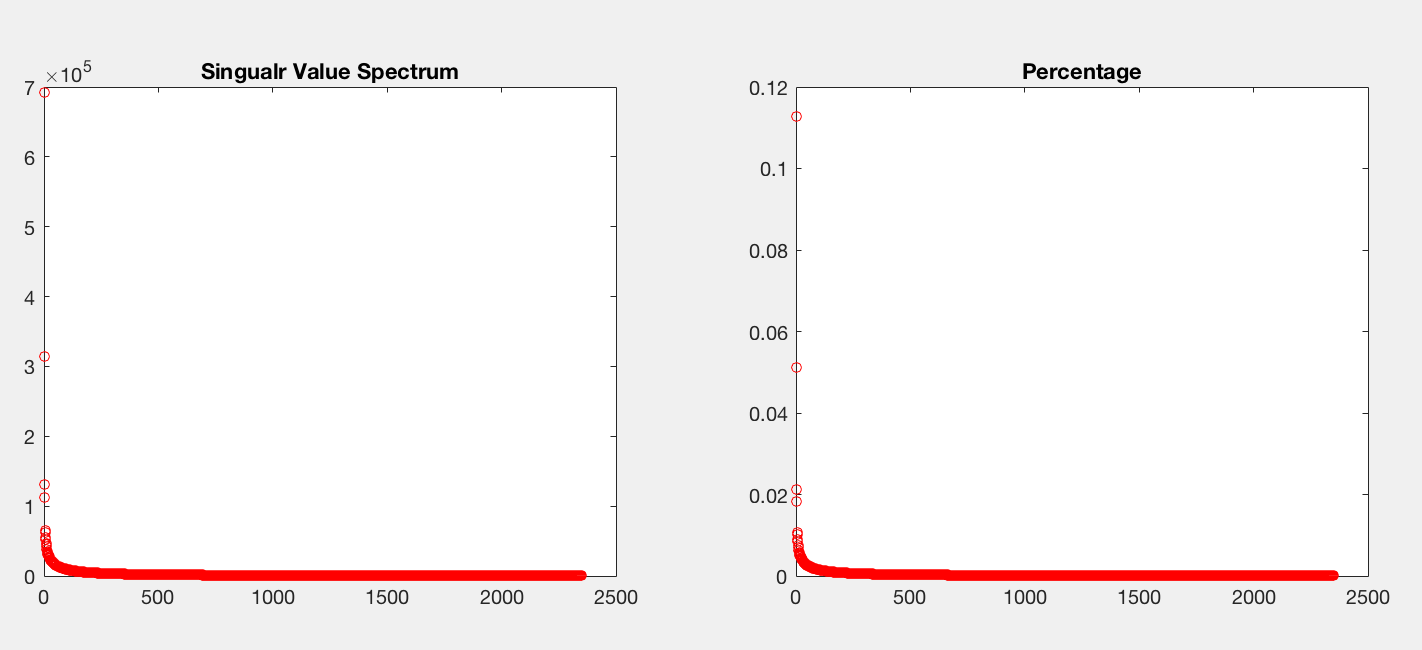
[U, S, V] = SVD(C);

And we can get U, S and V.

**Sec. IV. Computational Results**

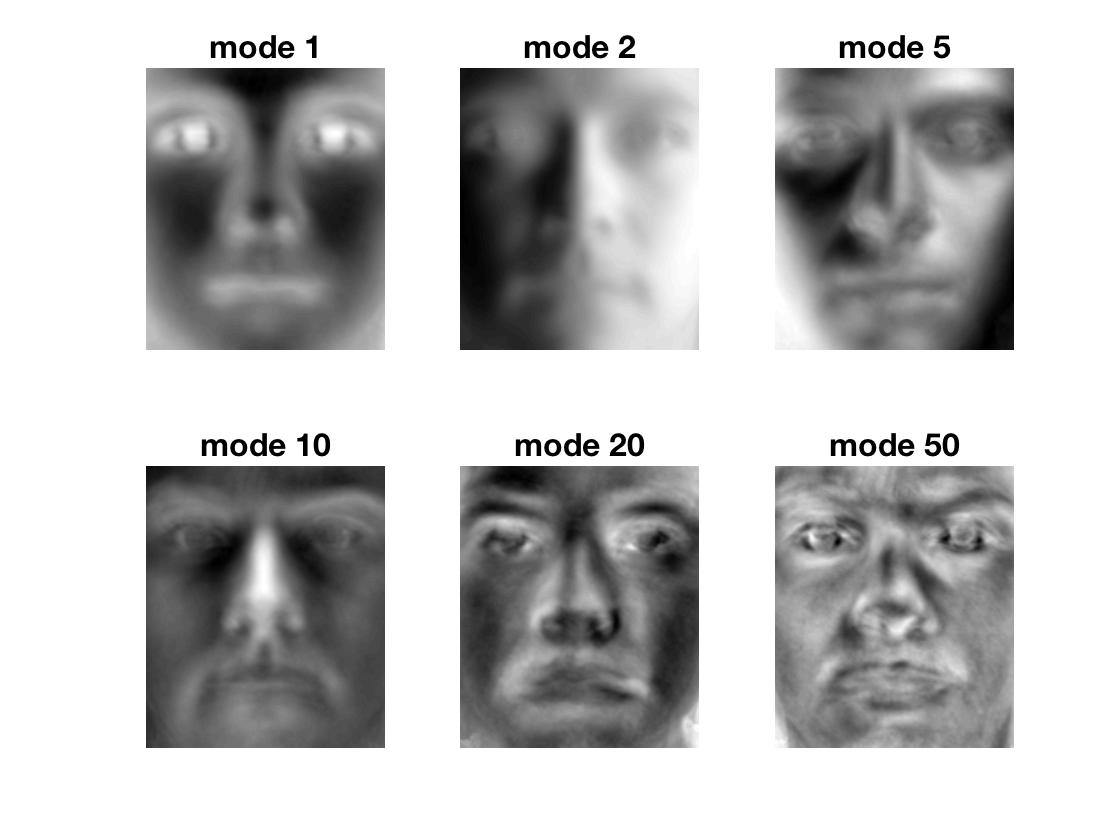
After loading all of images into Matlab and reshaping all of them, we calculate the mean value of each rows and in order to give a better sense for you, I plotted the mean face:



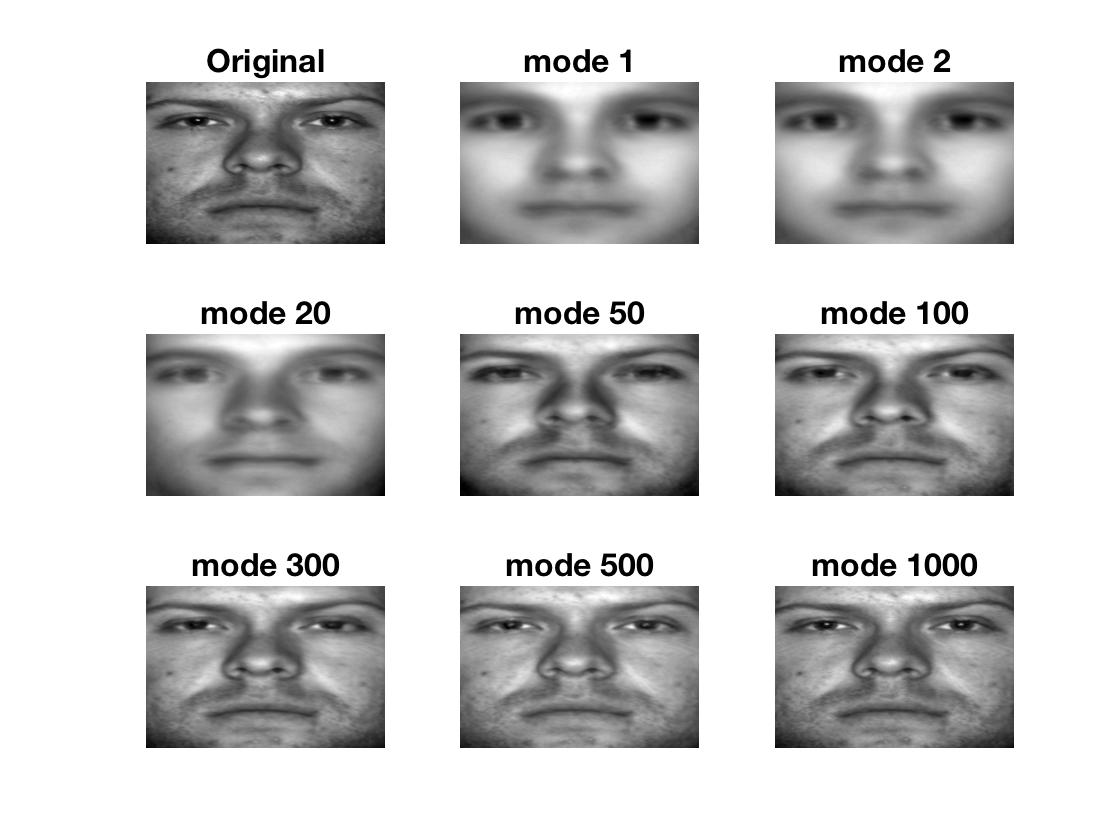
It’s an obvious human face since there’s a lot of similar feature or data, which is redundant and we need to get rid of it, and SVD can realize this purpose. After subtracting the mean from the unprocessed matrix C, I did a SVD of it and let’s first observe the singular value: 

As we can see from the plot, the first few values is extremely high and then decrease very fast and almost near 0.

Then I plot the images of different ranks:

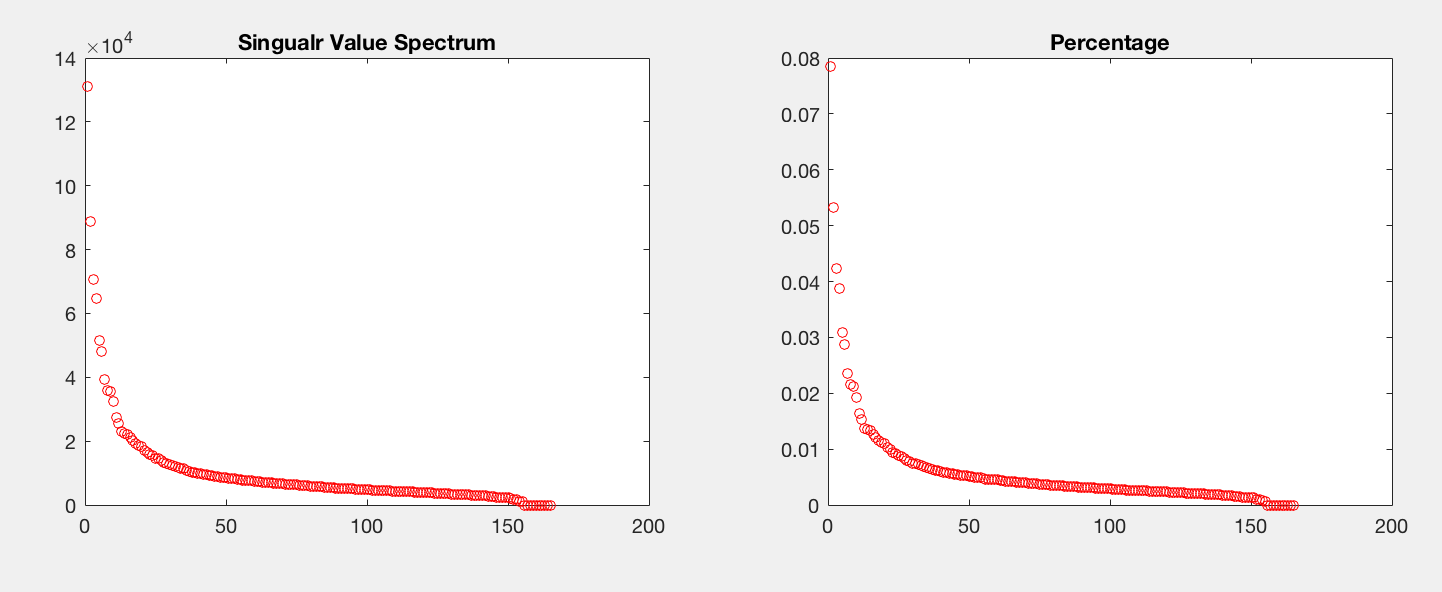
****

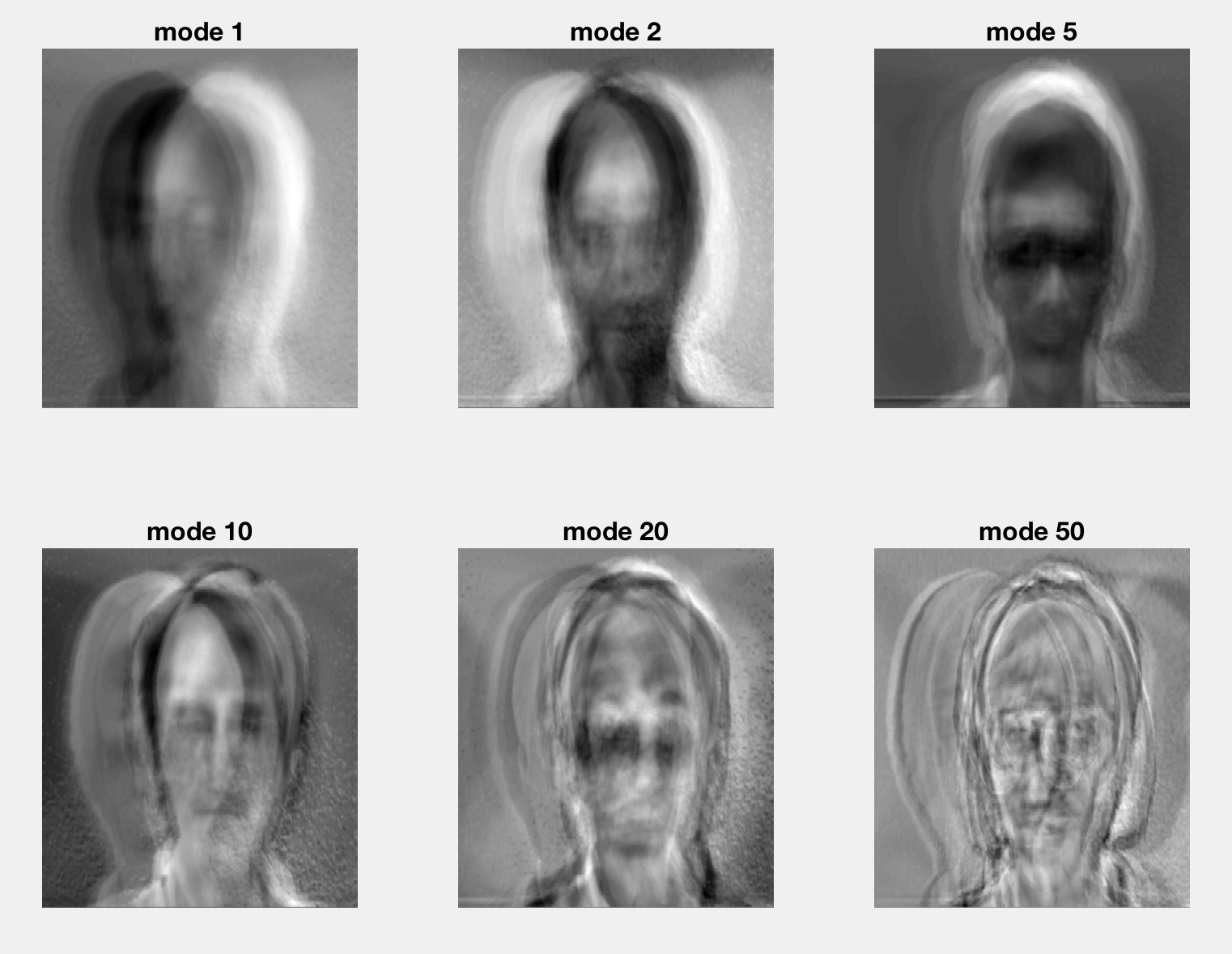
Since by SVD we just changed the basis of the original matrix so we can reconstruct the images by the new basis. Here’s the images with different ranks of reconstructing the first face:



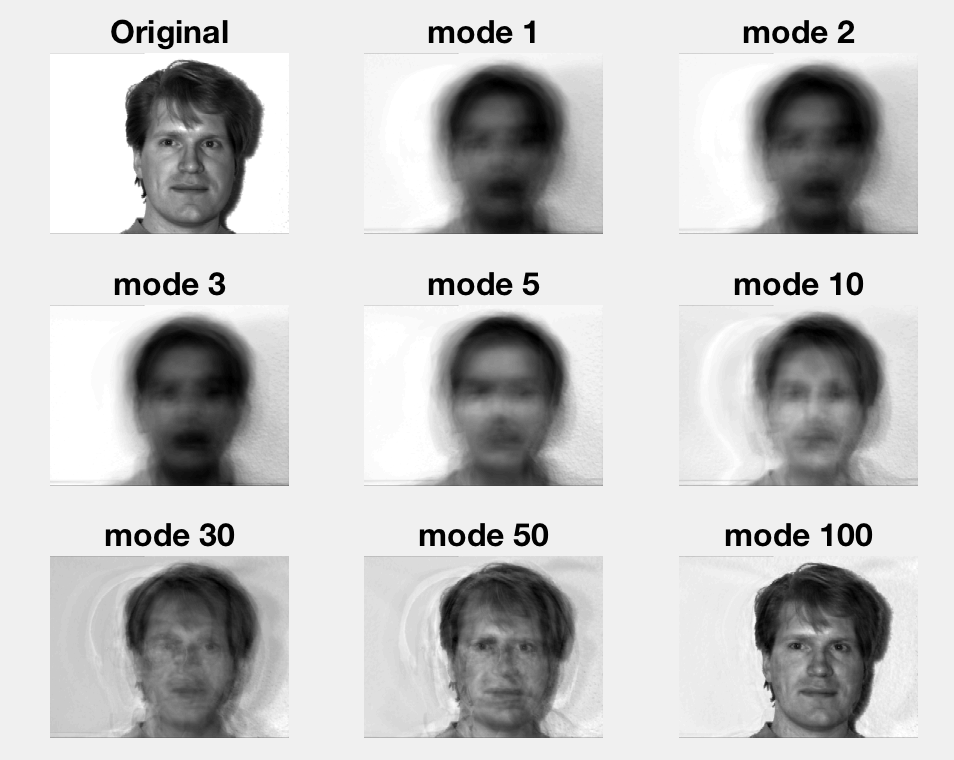
The higher rank I used, the more clear and similar to the original picture. When rank is over 100, it’s good enough.

Do the same step for uncropped images we can get result as the following:





We can see the eigenfaces are not aligned because it’s uncropped.



But we can still reconstruct the images.

**Sec. V. Summary and Conclusions**

From all the results above, SVD can remove the redundancy, which is human face’s common feature, of all the images by converting its basis. So we can reconstruct the original by those new basis. Also the more ranks we use, the more precise reconstruction of image we get.

The difference between cropped and uncropped images is uncropped eigenfaces are not aligned to the other.

**Appendix A MATLAB functions used and brief implementation explanation**

SVD – Singular Value Decomposition into U, S, V

reshape – convert a matrix into different sizes

pcolor – transfer matrix with pixels value into an image

**Appendix B MATLAB codes**

The following is for cropped images:

close all; clear all; clc

C=[];

%% Load images

list = dir('\*\*/\*.pgm');

l = length(list);

for i=1:l

strcat(list(i).folder,'/',list(i).name);

A=double(imread(strcat(list(i).folder,'/',list(i).name)));

B=reshape(A,[],1);

C=[C B];

end

%% subtract mean

[m,n]=size(C);

Original = C;

mn=mean(C,2);

C=C-repmat(mn,1,n);

%% SVD

[U,S,V]=svd(C,'econ');

%% plot energy

figure(1)

subplot(2,2,1), plot(diag(S),'ro','Linewidth', [0.5],'MarkerSize', 5),title('Singualr Value Spectrum')

subplot(2,2,2), plot(diag(S)/sum(diag(S)),'ro','Linewidth',[0.5],'MarkerSize', 5), title('Percentage')

%% show eigenfaces

figure(2)

a = 0;

for i = [1 2 5 10 20 50]

a = a+1;

ei = U(:,i);

subplot (2,3, a), pcolor(flipud(reshape(ei, 192,168))), shading interp, colormap(gray),axis off,title (['\fontsize{16}mode ' num2str(i)]);

end

%% Reconstruct images

A=Original( :,1);

A=reshape(A, 192,168);

figure(3)

subplot(3,3,1),pcolor(flipud(A)), shading interp, colormap(gray),title ('\fontsize{16}Original'), axis off

a = 2;

for r=[1 2 20 50 100 300 500 1000]

re =U(:,1:r)\*S(1:r,1:r)\*V(1,1:r).';

subplot(3,3,a), a = a + 1;

re = reshape(re, 192, 168);

pcolor(flipud(re)), shading interp, colormap(gray), title (['\fontsize{16}mode ' num2str(r)]), axis off

end

The only different between the codes of cropped and uncropped is the size of each image and in reconstruction we need to add the mean back.

References:

[1] Data Driven Modeling and Scientific Computation Methods for Complex System and Big Data, Nathan Kutz