|  |
| --- |
| **Ans 1:** MATLAB Code: clear;clc; |

load('teapots.mat')

teapot\_data = teapotImages;

m = mean(teapot\_data);

X = teapot\_data - m;

C = cov(X);

[V, D] = eig(C);

[d, ind] = sort(diag(D),'descend');

d = d(1:3,:);

v = V(:,ind(1:3));

c = X\*v;

X\_hat = m+c\*v';

%10 images

for i = 11:20

figure(i);

colormap gray;

subplot(1,2,1);

imagesc(reshape(teapot\_data(i,:),38,50));

title('Before Recon');

axis image;

subplot(1,2,2)

imagesc(reshape(X\_hat(i,:),38,50));

title('After Recon');

axis image;

end

norm(teapot\_data-X\_hat)

--

clear;clc;

load('teapots.mat')

X = teapotImages;

[coefficient\_of\_3, score3] = pca(X,'Algorithm','eig','Rows','all','NumComponents',3);

Xhat3 = mean(X)+score3\*coefficient\_of\_3';

[coefficient\_of\_6, score6] = pca(X,'Algorithm','eig','Rows','all','NumComponents',6);

Xhat6 = mean(X)+score6\*coefficient\_of\_6';

[coefficient\_of\_32, score32] = pca(X,'Algorithm','eig','Rows','all','NumComponents',32);

Xhat32 = mean(X)+score32\*coefficient\_of\_32';

figure(1);

colormap gray;

subplot(2,2,1);

imagesc(reshape(data(10,:),38,50));

title('Before');

axis image;

subplot(2,2,2)

imagesc(reshape(Xhat3(10,:),38,50));

title('TOP3');

axis image;

subplot(2,2,3)

imagesc(reshape(Xhat6(10,:),38,50));

title('TOP6');

axis image;

subplot(2,2,4)

imagesc(reshape(Xhat32(10,:),38,50));

title('TOP32');

axis image;

--

Python code for determining Eigenvalues:  
import numpy as np

import matplotlib.pyplot as plt

from numpy.linalg import eig

from scipy.io import loadmat

if \_\_name\_\_=='\_\_main\_\_':

    data = loadmat('teapots.mat')

    X = data['teapotImages']

    u = np.mean(X, axis=0).reshape(1,1900)

    x = X - np.repeat(u,100,axis=0)

    C = x.T.dot(x)/(x.shape[0]-1)

    D, V = eig(C)

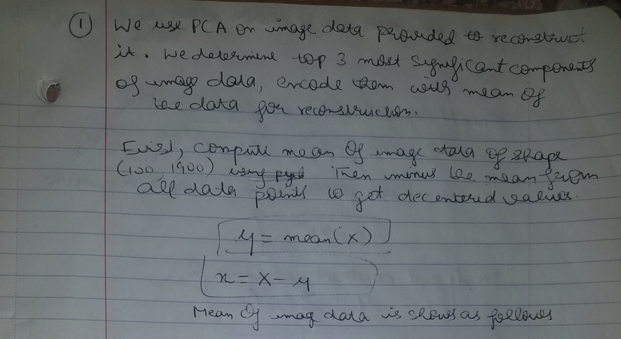
    print(x.shape)

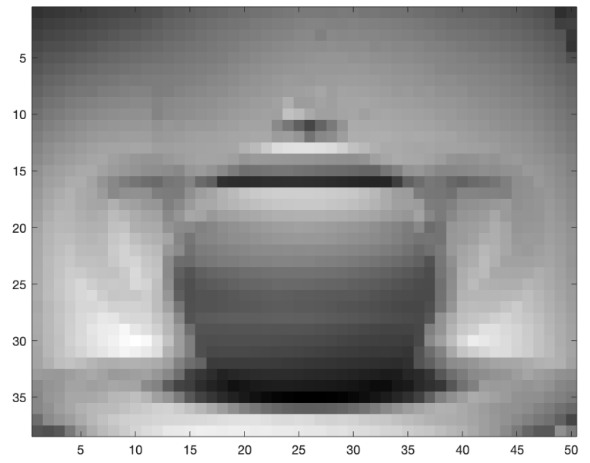
    print(C)

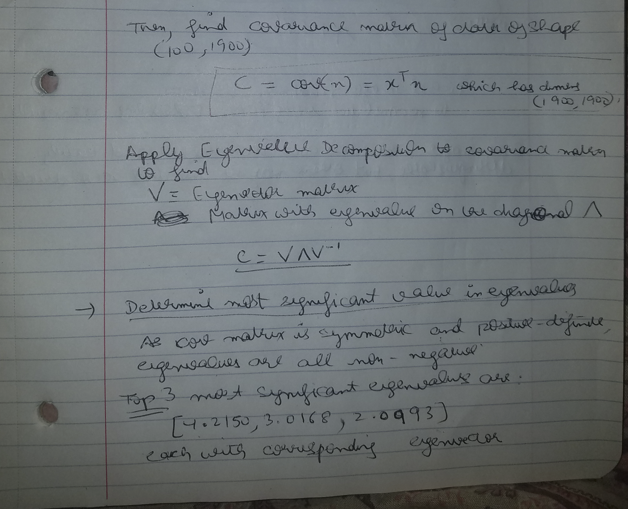
    print(D[:3])

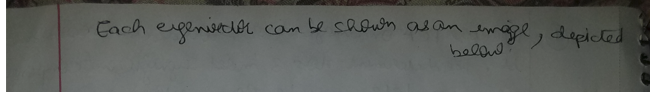
    print(V[:,:3])

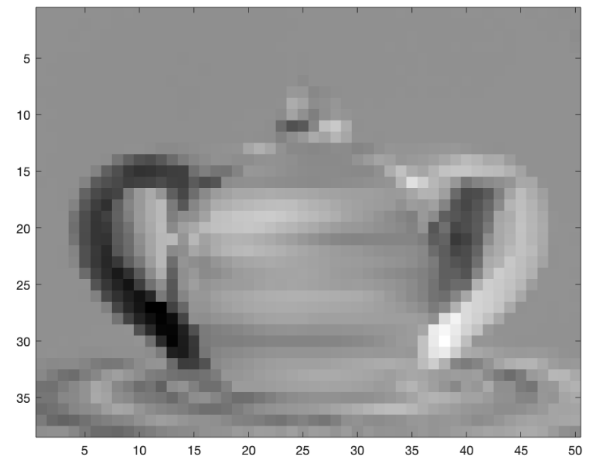
-----

Writeup:  


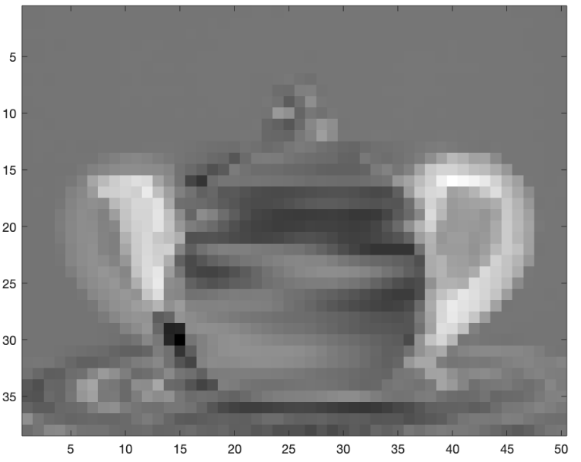


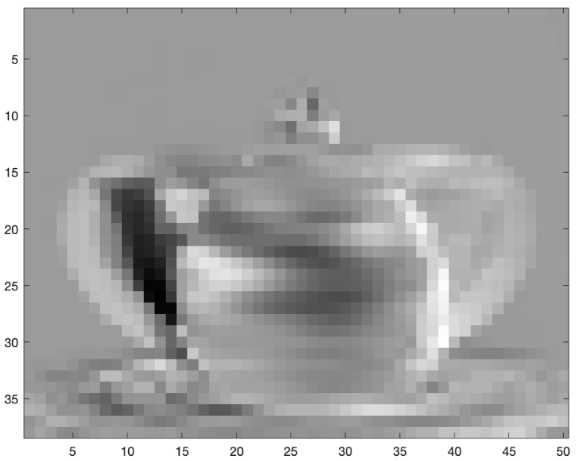


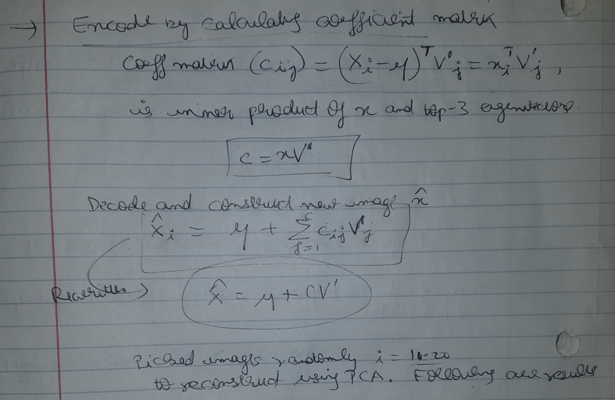


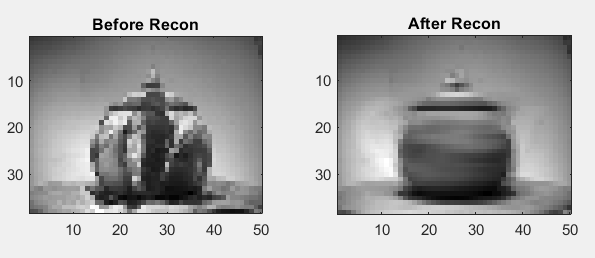
**Eigen Value=4.2150**

**Eigen Value: 3.0168**

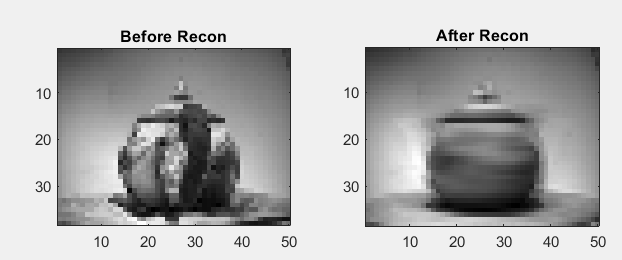


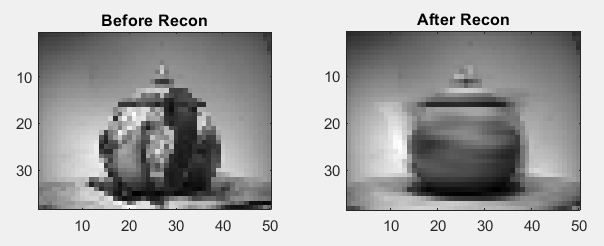
**Eigen Value: 2.0993**  
  


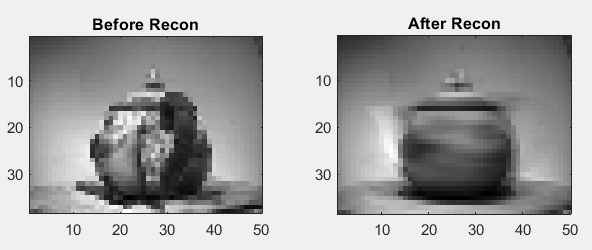


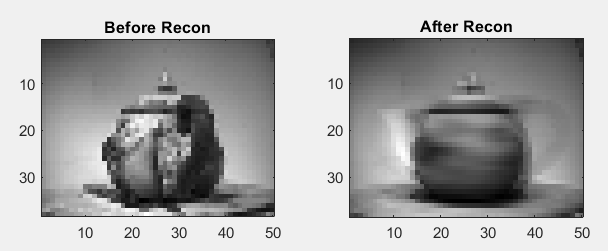


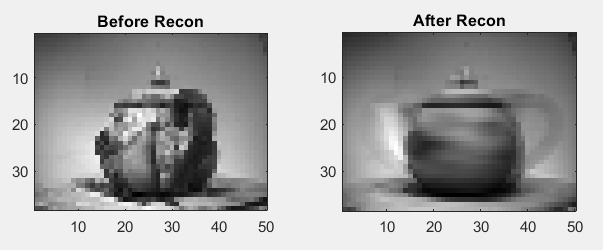


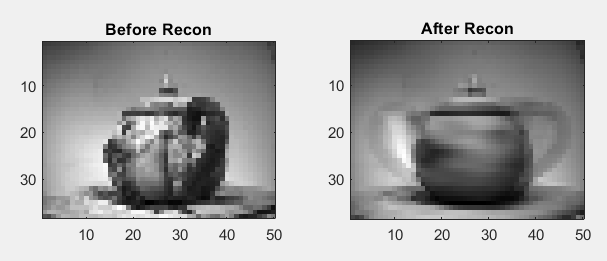


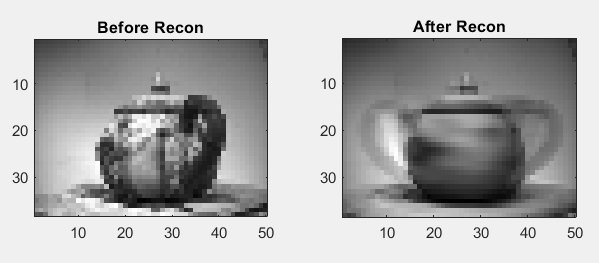


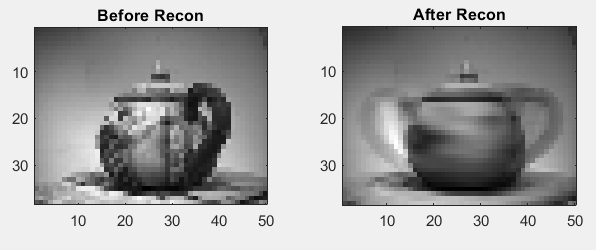










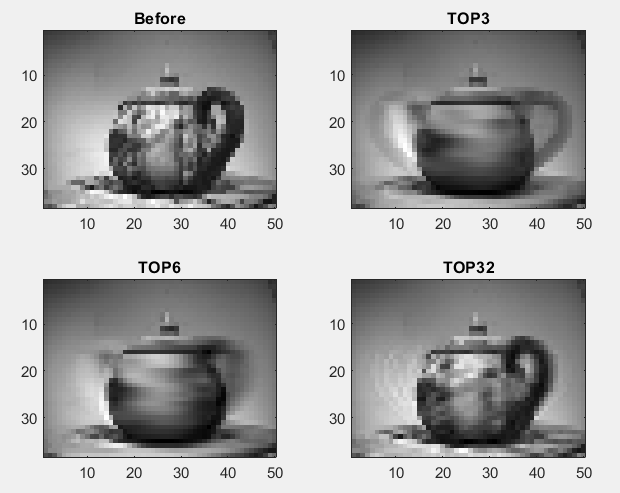


As observed, some images are well reconstructed and distinguishable but many others appear to be not that well reconstructed. Only top 3 components are sharp/aggressive. As per Eigenvalue decomposition, 32 values are more than 0.1, and 6 values are higher than 1. Decoding with more components could get a better reconstruction.

Performance of PCA encoder is determined by computing L-2 norm of X and X hat. The outcome number is 13.6262.

By picking top-6 components, norm value goes down to 9.8303; by picking top 32 component, norm reduces to 3.1382.

Following are snips of image data encode/decode with top 3, top 6 and top 32 most significant components.



Encoding more components into reconstruction gives better outcome with detailing. Deciding on how many most significant components we pick comes down to the requirements of the problem, item in question and how main feature is defined.

**Ans 2:**



**Ans 3:**

