```
clear;
clc;
close all;
D = './data_fruit';
rng(5);
img = zeros(16,19200);
S = dir(fullfile(D,'*.png')); % pattern to match filenames.
for k = 1:length(S)
    F = fullfile(D,S(k).name);
    I = imread(F);
    img(k,:) = reshape(I,[19200,1]); % initialising a matrix
 containing images as 19200*1 column vectors
end
dn = "done";
mean = sum(img,1)/16; % mean of 16 images
imgs = img-mean;
cv = transpose(imgs)*imgs/15; % covariance of 16 images
[U, S] = eigs(cv,4); % eigen vectors and eigen values
[D, ind] = sort(diag(S), 'descend');
S = S(ind,ind);
U = U(:,ind);
% sorted all eigen values and corresponding eigen vectors in
 descending
% ordet
for i=1:4
    % plotting the first 4 eigen vectors with the one with the highest
    % eigen value on the left ( in the same subplot )
    show_im = reshape(U(:,i).*sqrt(S(i,i))+transpose(mean),[80,80,3]);
    subplot(1,5,i);
    imshow(double(show_im)/255.0);
    title("eig"+i);
end
% displaying the mean image in the leftmost position in the figure
% containing the eigen vector subplots
subplot(1,5,5);
imshow(reshape(double(mean)/255.0,[80,80,3]));
title("mean")
pause(2);
% finding and ploting the first 10 eigen vectors
ten_eigen = eigs(cv,10);
figure;
plot(ten_eigen,'o-');
title('first 10 eigen values');
ylabel('eigen values');
pause(3);
for i=1:16
    figure;
    subplot(2,1,1);
    imshow(reshape(double(img(i,:))/255.0,[80,80,3]));
    % finding the coefficients for all the eigen vectors to estimate
 an
```

```
    % approximate representation of the image as a linear combination
 of the
    % eigen vectors and the mean
    a1 = imgs(i,:)*((U(:,1)));
    a2 = imgs(i,:)*((U(:,2)));
    a3 = imgs(i,:)*((U(:,3)));
    a4 = imgs(i,:)*((U(:,4)));
    subplot(2,1,2);
    % adding the mean to go back to the original co-ordiante frame
 from the
    % mean shifted co-ordinate frame

 imshow(reshape((double(a1*U(:,1)+a2*U(:,2)+a3*U(:,3)+a4*U(:,4)+transpose(mean)))/
[80,80,3]));
    title("image "+i+" and its approximate representation");
    pause(2);
end
pause(2);

for i=1:3
    figure;
    % generating unfiorm random variables in the range (-1,1)
    a1 = randn;
    a2 = randn;
    a3 = randn;
    a4 = randn;
    % normalisinh the co-rfficients by dividing by their sum
    asum = sqrt(a1^2+a2^2+a3^2+a4^2);
    a1 = a1/asum;
    a2 = a2/asum;
    a3 = a3/asum;
    a4 = a4/asum;
    % finding final co-efficients to be multiplied with each eigen
 vectorby
    % mutiplying the random uniform sample values
    % with the square root of the eigen values

 imshow(reshape((double(a1*U(:,1)*sqrt(S(1,1))+a2*U(:,2)*sqrt(S(2,2))+a3*U(:,3)*sq
[80,80,3]));
    title("new fruit "+i);
    pause(2);
end
```


eig1
eig2
eig3
eig4
mean

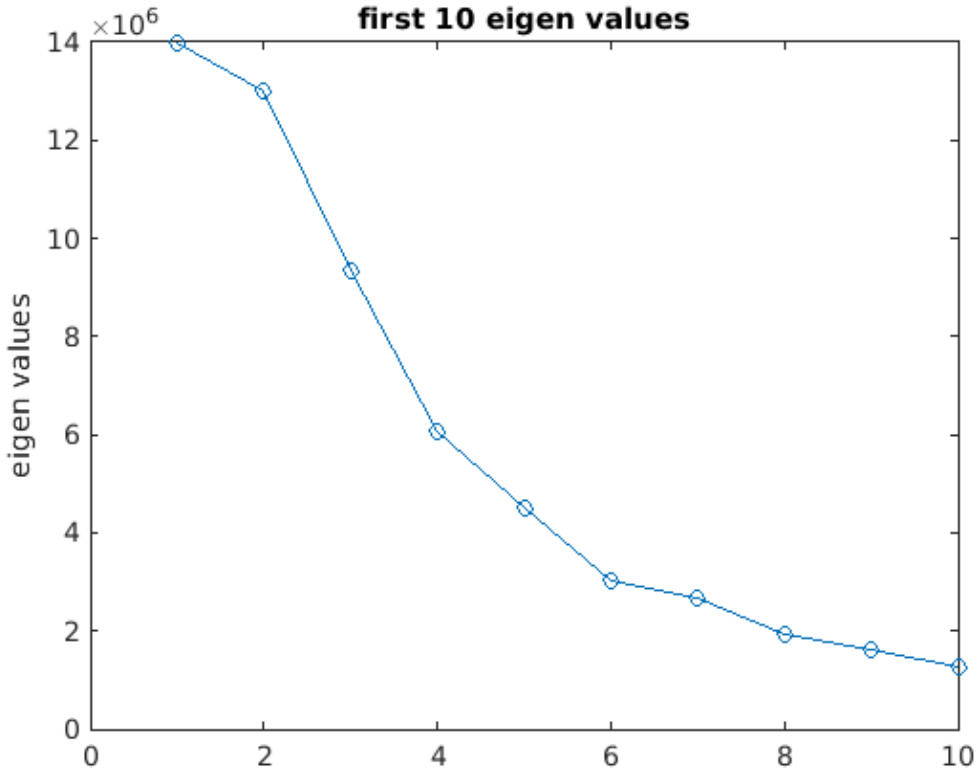

first 10 eigen values
×10^6
eigen values
14
12
10
8
6
4
2
0
0
2
4
6
8
10

image 1 and its approximate representation

image 2 and its approximate representation

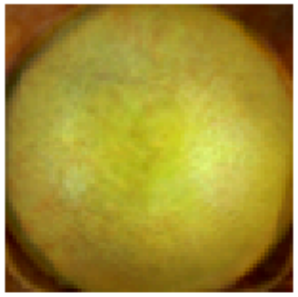
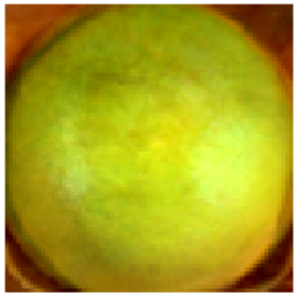


image 3 and its approximate representation



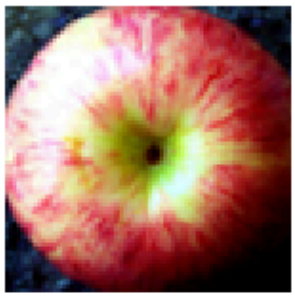


image 4 and its approximate representation

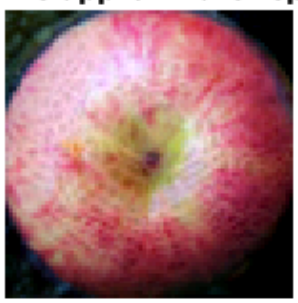


image 5 and its approximate representation

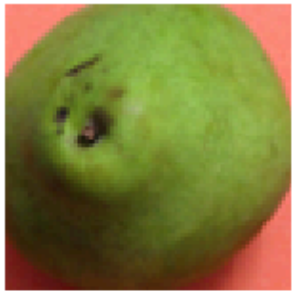


image 6 and its approximate representation

image 7 and its approximate representation

image 8 and its approximate representation

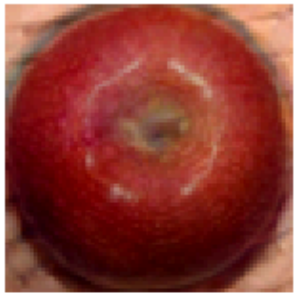


image 9 and its approximate representation

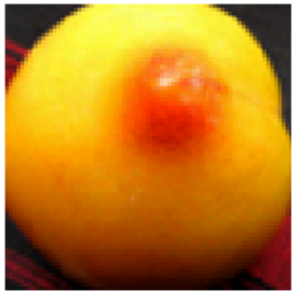
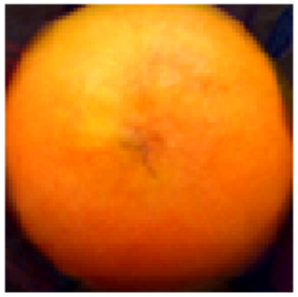


image 10 and its approximate representation

image 11 and its approximate representation



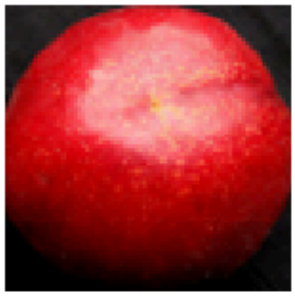


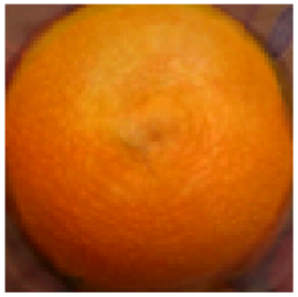
image 12 and its approximate representation

image 13 and its approximate representation

image 14 and its approximate representation

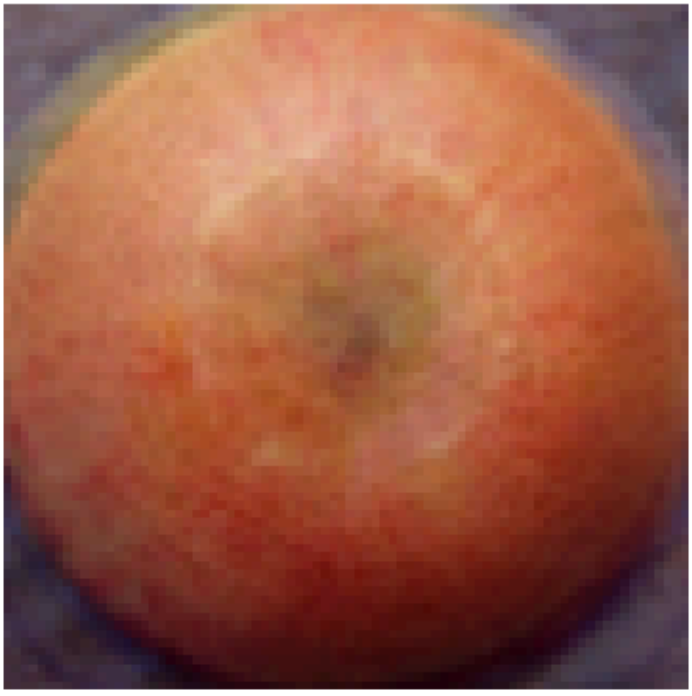
image 15 and its approximate representation

image 16 and its approximate representation



new fruit 1

new fruit 2



Published with MATLAB® R2020b