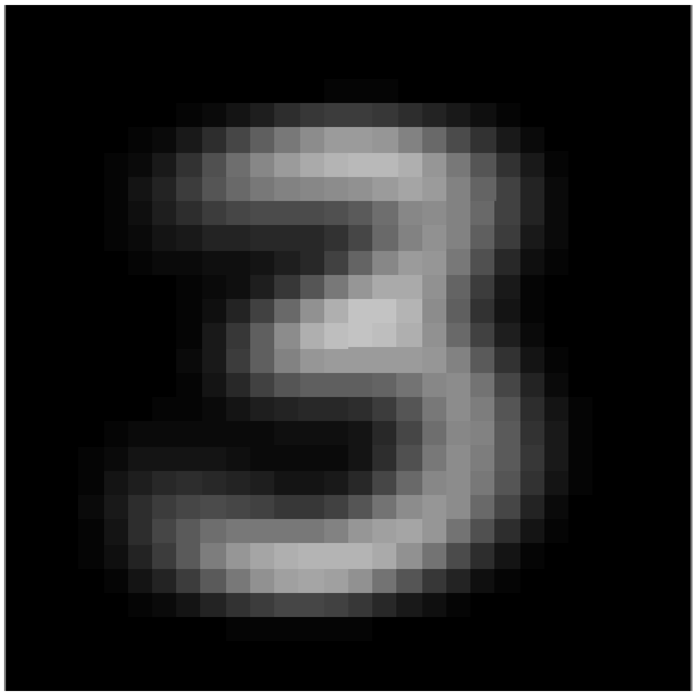
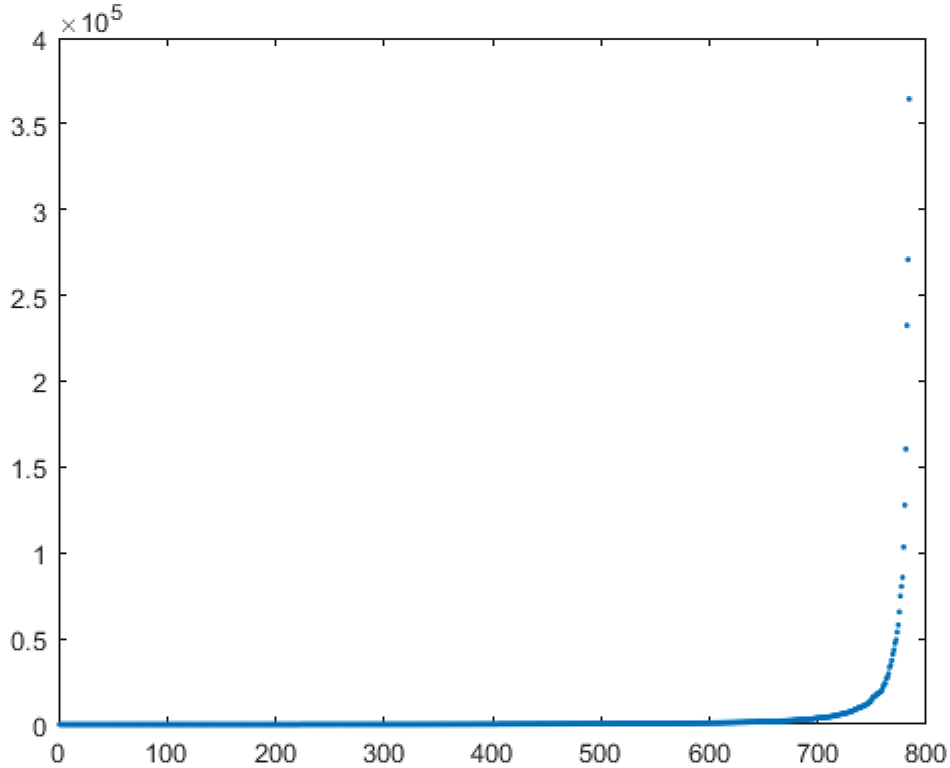
```
%Principal Component Analysis
% The following code utilizes the MNIST databse of handwritten digits,
% which contains a training set of 60,000 numbers intended to train an
% algorithm to recognize handwritten numbers.

load mnist_all.mat
% FIGURE 1
% First, we plot the first four instances of training digit 3
for i=1:4,
    subplot(2,2,i);
    digit = train3(i,:);
    digitImage = reshape(digit,28,28);
    image(rot90(flipud(digitImage),-1));
    colormap(gray(256)), axis square tight off;
    hold on
end;
```

```
T = zeros(1,784);
T(1,:) = mean(train3);

%AVERAGE VALUES
figure('Name', 'Average Values')
digit = T(1,:);
digitImage = reshape(digit,28,28);
image(rot90(flipud(digitImage),-1)),
```

```
colormap(gray(256)), axis square tight off;
hold on


% Next, we form the covariance matrix associated with the training
 data for
% the number 3, and plot the associated eigenvalues to determine how
 many
% principal components to keep
```



```
%EIGENVALUES
figure('Name', 'Eigenvalues')
Q3 = cov(double(train3));
[E3,Lam3] = eig(Q3);
plot(diag(Lam3), '.')

E3 = E3(:,775:784); %I chose to keep 10 principal components

% Next, we mean-enhance the first four training digits (as seen in
 Figure 1) by
% adding T to the corresponding linear combination of these
 eigenvectors
for i=1:4,
    T_enhanced(i,:) = T(1,:) + (double(train3(i,:)) - T(1,:))*E3*E3';
end;
```




```
%ENHANCED
figure('Name', 'Enhanced 3s')
for i=1:4,
    subplot(2,2,i);
    digit = T_enhanced(i,:);
    digitImage = reshape(digit,28,28);
    image(rot90(flipud(digitImage),-1));
    colormap(gray(256)), axis square tight off;
    hold on
end;
```

Published with MATLAB® R2019b