(also known as principal component analysis)

Exercises 11.1

- The last method of analyzing data based on eigenvalue/vectors of covariance matrices can be applied to any type of data; let's consider a set of images here
- First, download the 'att_faces.zip' file with a set of face images from CAPS11 material section in Google Classroom and extract it into 'att_faces' folder.
- Second, copy the script 'load_faces.m' in 'CAPS11files.zip', also downloaded from the material section in Google Classroom. Make sure the 'load_faces.m' script and 'att_faces' folder are in the same directory.
- Using this script, load 400 face images (92x112 pixels) to X, a 400x10304(=92x112) matrix, by typing

```
>> load_faces
```

```
>> imshow(reshape(X(100,:),[112,92])/255)
```

'/255' may not be necessary depending on your system

`reshape` reshapes a vector into a matrix of 92x112, which is treated as an image

Continued to the next page…

Exercises 11.1

- Calculate the first 20 eigenvalues of the covariance matrix of X and plot them
  - **Remark**: In this example, each data point is a single image; it resides in 92x112=10304-dimensional space; there are 400 data points (=face images); thus, cov(X) is a 10304x10304 matrix and its computation is very, very time-consuming (don't do this)
  - Hint: Recall the relation between SVD and the eigenvalue problem; use SVD instead of `eig(cov(X))`; to be specific, type below

```
>> [U,W,V]=svds(X-ones(400,1)*mean(X),20);
```

`svds` calculates a specified number of largest singular values and related vectors

  - See that the first few singular values (square root of eigenvalues) are very large and the subsequent singular values are very small
  - Remark: This means that the data reside only in a *low-dimensional subspace* in the 10304-dim data space
- Calculate also the eigenvectors and then display them as images of 92x112 pixels
  - Hint: Eigenvectors have negative elements in general and thus some normalization of brightness necessary; you can display the first eigenvector as a 92x112 image by

```
>> svec=V(:,1);
>> imshow(reshape(svec,[112,92]),[min(svec),max(svec)])
```