4. (25 points) Principal Component Analysis (PCA).

Read all instructions before you start.

Download the dataset comprising images of handwritten digits in `http://yann.lecun.com/exdb/mnist`; this has been downloaded in the folder "data" and stored as "mnist.mat". Use the entire training set of 60000 examples.

Each image is stored as a matrix (28×28) of numbers. You can visualize these images (or matrices) in Matlab using the functions imagesc() or imshow(). Use the Matlab command "axis equal" to use the same units on each axis of the image.

For the following computations, make sure to convert (cast) the integer data type to a floating-point type. For this question, you cannot use the functions mean(), cov(), and pca() in Matlab.

For every digit, from 0 to 9, compute:

(i) the mean μ (3 points),

(ii) the covariance matrix C (5 points), and

(ii) the principal mode of variation determined by the eigenvector v_1 and the corresponding eigenvalue λ_1 (where λ_1 is the largest of all eigenvalues) of the covariance matrix C (7 points).

Note: Before computing the mean and covariance matrix, convert each 28×28 pixel image matrix to a $28^2 \times 1$ vector by concatenating its columns. To visualize the $28^2 \times 1$ mean vector, convert it back to a matrix and then visualize it using imagesc(). Use the reshape() function to change matrices to vectors and vice versa. The covariance matrix will be of size $28^2 \times 28^2$.

• (5 points) For each digit, sort the 28^2 eigenvalues of the covariance matrix and plot them as a graph. Comment and justify what you observe. How many "principal" / significant modes of variation (i.e., number of "large" eigenvalues) do you find, for each digit ? Are the significant modes of variation equal to 28^2 or far less ? Why ?

• (5 points) For each digit, show the 3 images side by side: (i) $\mu - \sqrt{\lambda_1} v_1$, (ii) μ, and (iii) $\mu + \sqrt{\lambda_1} v_1$, to show the principal mode of variation of the digits around their mean. Comment and justify what you observe. For a certain digit, say 1, what does the principal mode of variation tell you about how people write that digit ?

5. (10 points) Principal Component Analysis (PCA) for Dimensionality Reduction.

   Read all instructions before you start.

   Download the dataset comprising images of handwritten digits in `http://yann.lecun.com/exdb/mnist`; this has been downloaded in the folder "data" and stored as "mnist.mat".

   As of now, for each digit, each 28×28 pixel image is represented using 28^2 coordinate values in the Euclidean space of dimension 28^2. Suppose you decide to re-represent the images using only 84 coordinates (instead of $28^2 = 784$) in a 84-dimensional basis for some 84-dimensional hyperplane within the original Euclidean space, such that the chosen 84-dimensional hyperplane maximizes the total dispersion of the original data (for the chosen digit) within the hyperplane.

   • (5 points) Write a function to compute those 84 coordinates, for each of the ten digits (0–9).

   • (5 points) Give an algorithm for regenerating / reconstructing the image using those 84 coordinates (and the knowledge of the designed 84-dimensional basis). For each of the ten digits (0–9), pick an image, and show the original and the reconstructed images side by side.

6. (25 points) Principal Component Analysis (PCA) for Another Image Dataset

   Read all instructions before you start.

   Consider the dataset provided within the folder “data_fruit”

   For this question, you cannot use the functions mean(), cov(), and pca() in Matlab.

   Each datum is an image of size 80×80 pixels with 3 color channels red (R), green (G), and blue (B), i.e., a $80 \times 80 \times 3$ array. For PCA, each image should be resized to a vector of length 19200. For visualization, reshape each vector back to a RGB image of size 80×80 pixels using the function reshape(), followed by a shift and rescaling of the values into the range $[0, 1]$, followed by displaying the matrix using the function image().

   • (9 points: 1 + 4 + 4) Similar to the analysis done in previous question, find the mean μ, the covariance matrix C, and the top 4 principle **eigenvectors** of C. Display the **mean** and the **eigenvectors** as images (side by side, in the same figure); you can use the function subplot(). Find the top 10 **eigenvalues**, sort them, and plot their values on a graph. Use the function eig**s**() for efficient computation.

   • (8 points: 4 + 4) For each fruit image in the dataset, finds its **closest** representation as a linear combination of the top 4 eigenvectors added to the mean. Use the measure of closeness as the Frobenius norm of the difference. Describe the algorithm used to produce this closest representation in mathematical terms and describe the logic behind your algorithm. Display the original fruit image and its closest representation, as images (side by side, in the same figure).

   • (9 points: 3 + 3 + 3) Using all of the top 4 eigenvectors and the mean image, **sample** random images to generate new images of "fruit". Describe the underlying algorithm clearly in words and including suitable mathematical notation. Display three such images that are distinct from any image in the given the dataset, but are representative of the dataset and can be considered as that of a new / generated fruit.