Project 2 – Numerical Methods For the implementation of the project we solved the six requirements as follows::

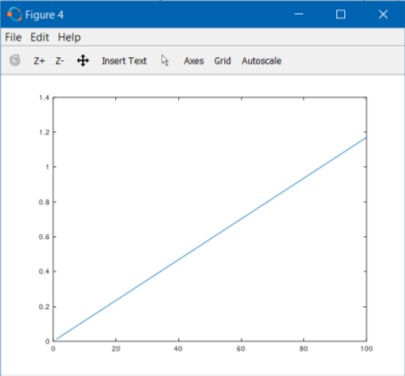
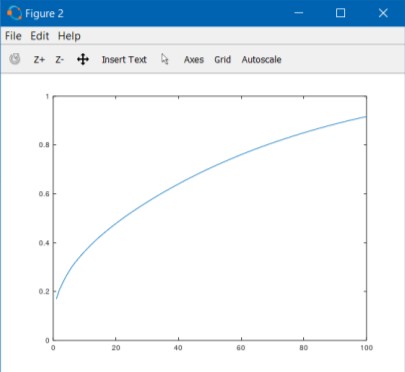
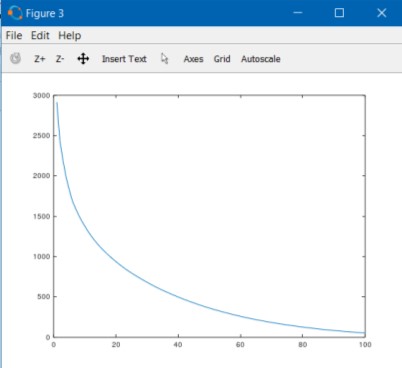
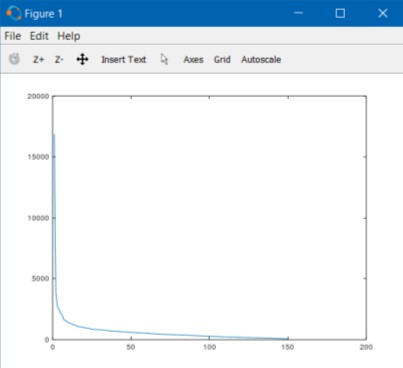
* Task 1:
* For solving this task, we read the image and obtained the U, S, and V matrices and then formed the A\_k matrix.
* Task 2:
  + For solving this task we generated four graphs as follows:
    1. For the first graph we graphed all the singular values of the matrix A in descending order, using the decomposition obtained at the first task, so using the diagonal of the matrix S, depending on the “k” value received as a function parameter.
    2. For the second graph we found out what was on the Oy axis according to the formula given and on the Ox axis I represented the interval from 1 to k.
    3. For the third graph we determined the approximation error for the matrix A according to the given formula, values that will be placed on the Oy axis, and expressed it according to the interval 1: k.
    4. For the fourth graph we determined the compression rate of the data according to the given formula, values that will be placed on the Oy axis, and we expressed it according to the interval 1: k.

(For generating graphics:

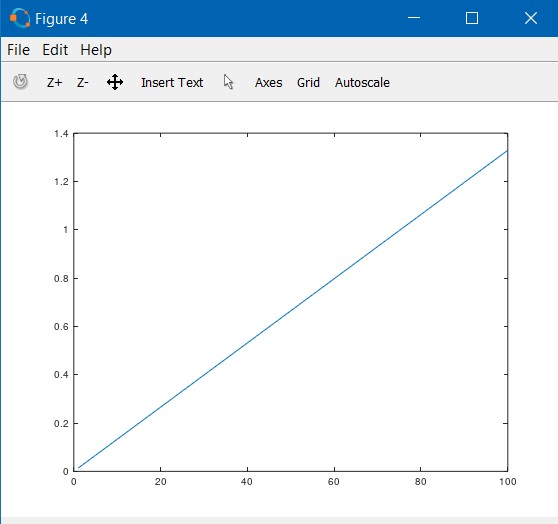
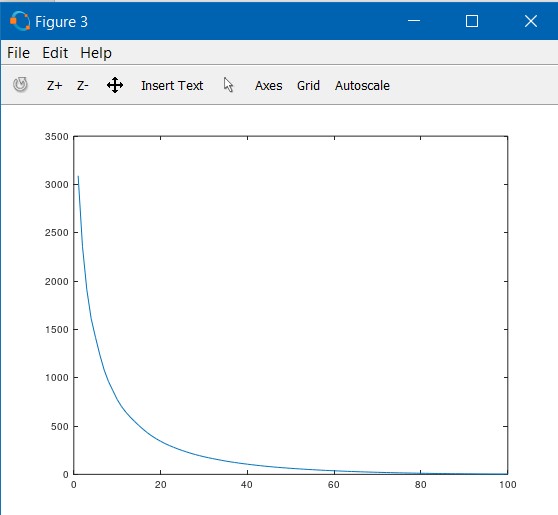
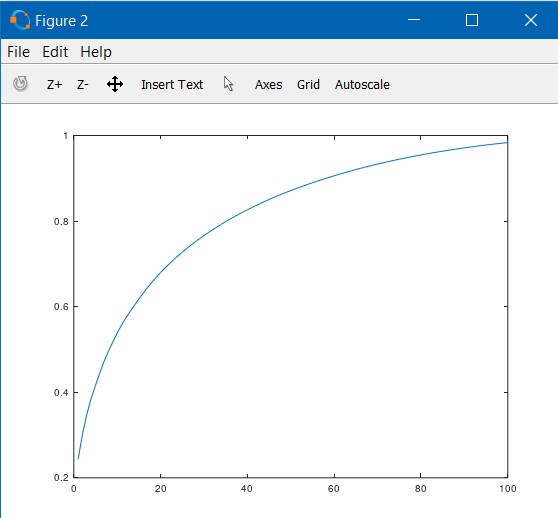
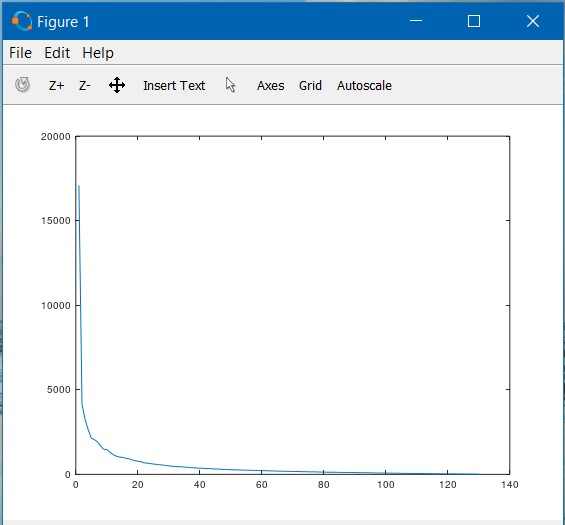
In the case of the first two graphs they are generated quite quickly, but for the third and fourth graphs it takes around 3-4 minutes.)

The graphs for this task are the following:

 For the second image and k = 1:100



 For the third image and k = 1:100



* Task 3 :
  + For solving this task I follow the steps of the algorithm, thus finding the Z matrix, use the svd function to find the U matrix that will use to find the W matrix and thus using all the steps of the algorithm find the A\_k matrix, the S matrix existing already from the decomposition given by svd function.

* Task 4:
  + For solving this task, I follow the steps of the algorithm, similar to task 3, and I find the A\_k matrix

and the S matrix.

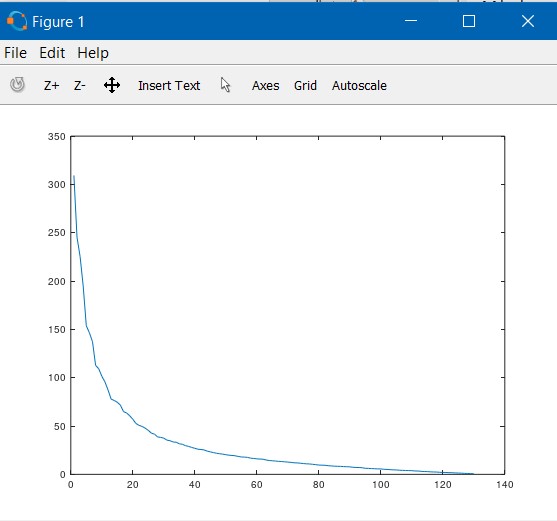
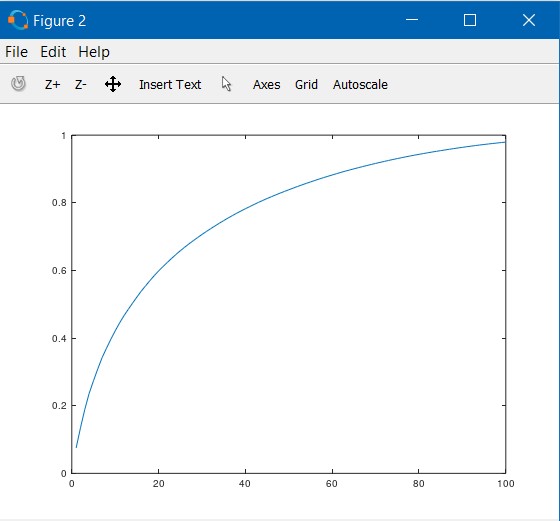
* Task 5:
  + For solving this task we generated four graphs as follows:
    1. For the first graph we graphed all the singular values of the matrix A using the decomposition obtained at the third task, so using the diagonal of the matrix S, depending on the minimum between the values m and n, that is the number of lines and columns of the matrix A\_k.
    2. For the second graph I found out what was on the Oy axis according to the formula given and on the Ox axis I represented the interval from 1 to k.
    3. For the third graph I determined the approximation error for matrix A according to the given formula, values that will be placed on the Oy axis, and expressed it according to the interval 1: k.
    4. For the fourth graph, we determined the data compression rate according to the given formula, values that will be placed on the Oy axis, and we expressed it according to the interval 1: k.

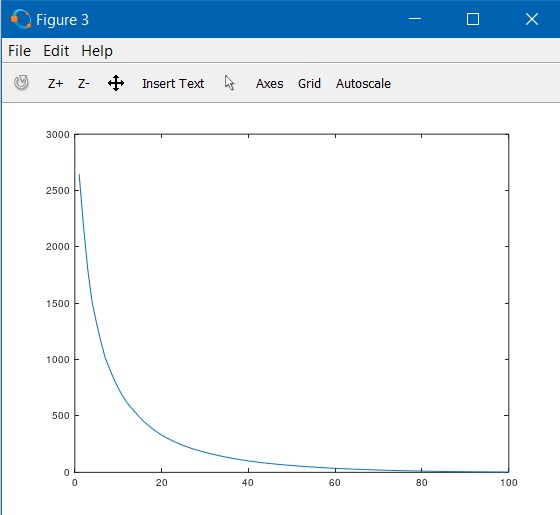
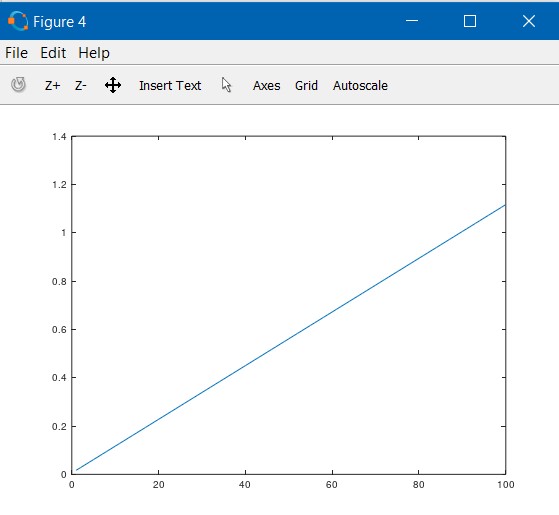
(For generating graphics:

In the case of the first two graphs they are generated quite quickly, but for the third and fourth graphs it takes around 3-4 minutes.)

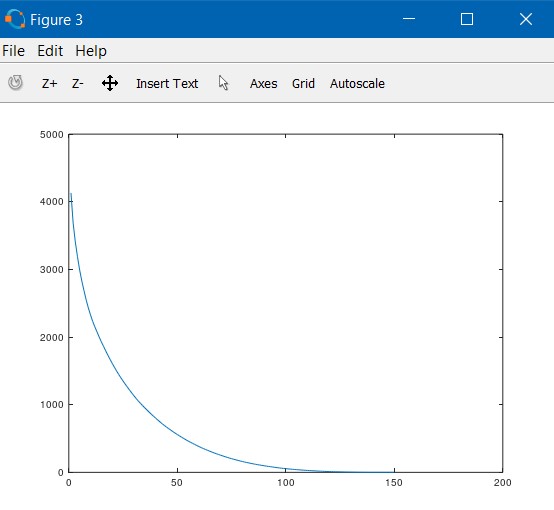
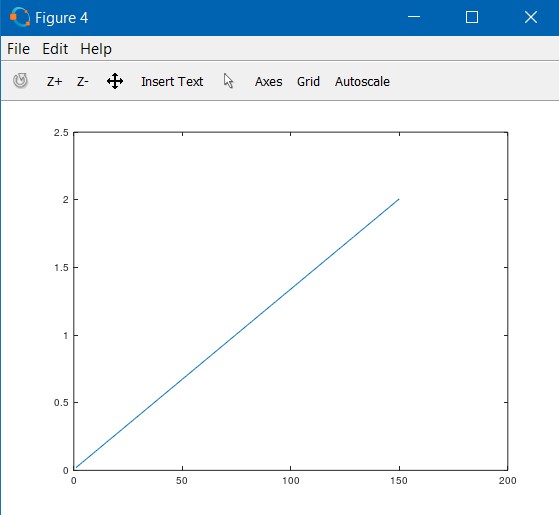
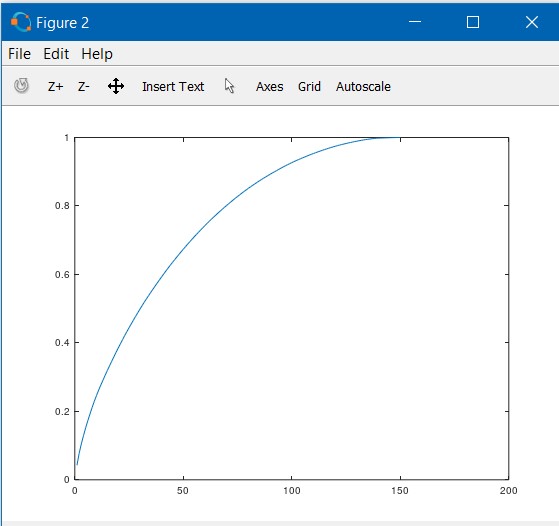
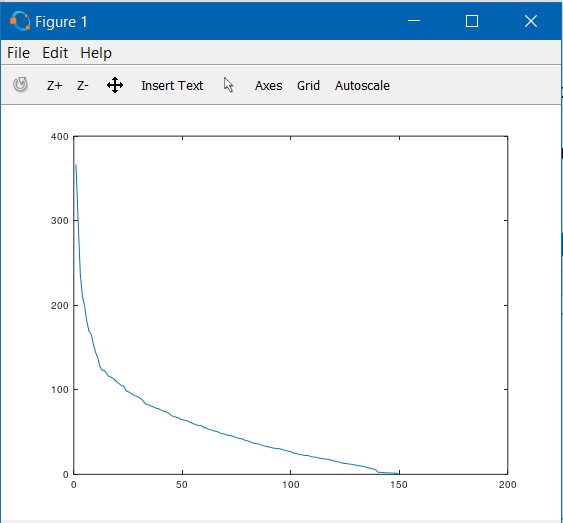
The graphs for this task are the following:

 For the third image and k = 1:100

 For the fourth image and k = 1:150



* Task 6 :
* For solving this task I first followed the steps of the algorithm, after which I used the "mean" function to find the average on lines, first transposing the T matrix, the lines becoming columns, the mean function making the average and then transposing the result again so that the "m" vector has either column. After that, I use the eig function to find the eigenvalues greater than 1 by iterating through the "valp" diagonal matrix and if the eigenvalue is greater than 1, the corresponding vector in the "vect" matrix is attached to the "V" matrix, finally calculating according to the formulas "eigenfaces" and “Pr\_img” matrices.
  + - After that in the face\_recognition function is repeating the first and third steps as in the eigenface\_core function, then we will determine the minimum distance between the column vector resulting in the face\_recognition function and each column of the pr\_img matrix resulting from the first function, using the norm (norm function) and updating each time the column index in the pr\_img matrix.