Project-4 Documentation

Anudeep Reddy

December 10, 2018

1 PCA

1.1 compute_Z() function

Matrix X, boolean variables ’centering’ & ’scaling’ are passed as arguments. If statement is used to perform centering and scaling on matrix X and final output is stored in variable ’Z’ and returned.

1.2 compute_covariance_matrix() function

’Z’ from the above function is an argument to this function. Covariance of ’Z’ is computed using "np.dot()" and stored ’COV’ variable and finally returned.

1.3 find_pcs() function

Covariance matrix computed in above function is passed to get eigen values and vectors. Eigen values are stored in ’eig_vals’ and eigen vectors are stored in ’eig_vects’ numpy variables. Eigen values are sorted and ordered indexes are returned by "np.argsort()" and these indexes are used to sort eigen vectors. And ’eig_vals’ & ’eig_vects’ are returned.

1.4 project_data() function

Arguments Z(centered and scaled matrix), PCS(eigen vectors/ principal components), L(eigen values) are pased. Also, arguments, ’k’ which tells the number of principal components to use, & ’var’ - threshold value which tells to add features until you hit that threshold. An if else statement is executed to select principal components in either one of the method (k or var). And ’Z_star’ is computed with Z and selected number of principal components and finally returned.

1.5 load_data() function

An ’input_dir’ argument, which has the path to the training data (images) is passed to this function. Image names are read first using "os.listdir(input_dir)" and stored in a list variable ’images’. And path+image names are read using "plot.imread()" which gives us images in matrix form which is then converted into float type. Flattened matrix is then flattened and stored in an ’img_arrays’ variable as column. All the images are appended to this ’img_arrays’ variable in column-wise. Finally, ’img_arrays’ is returned.

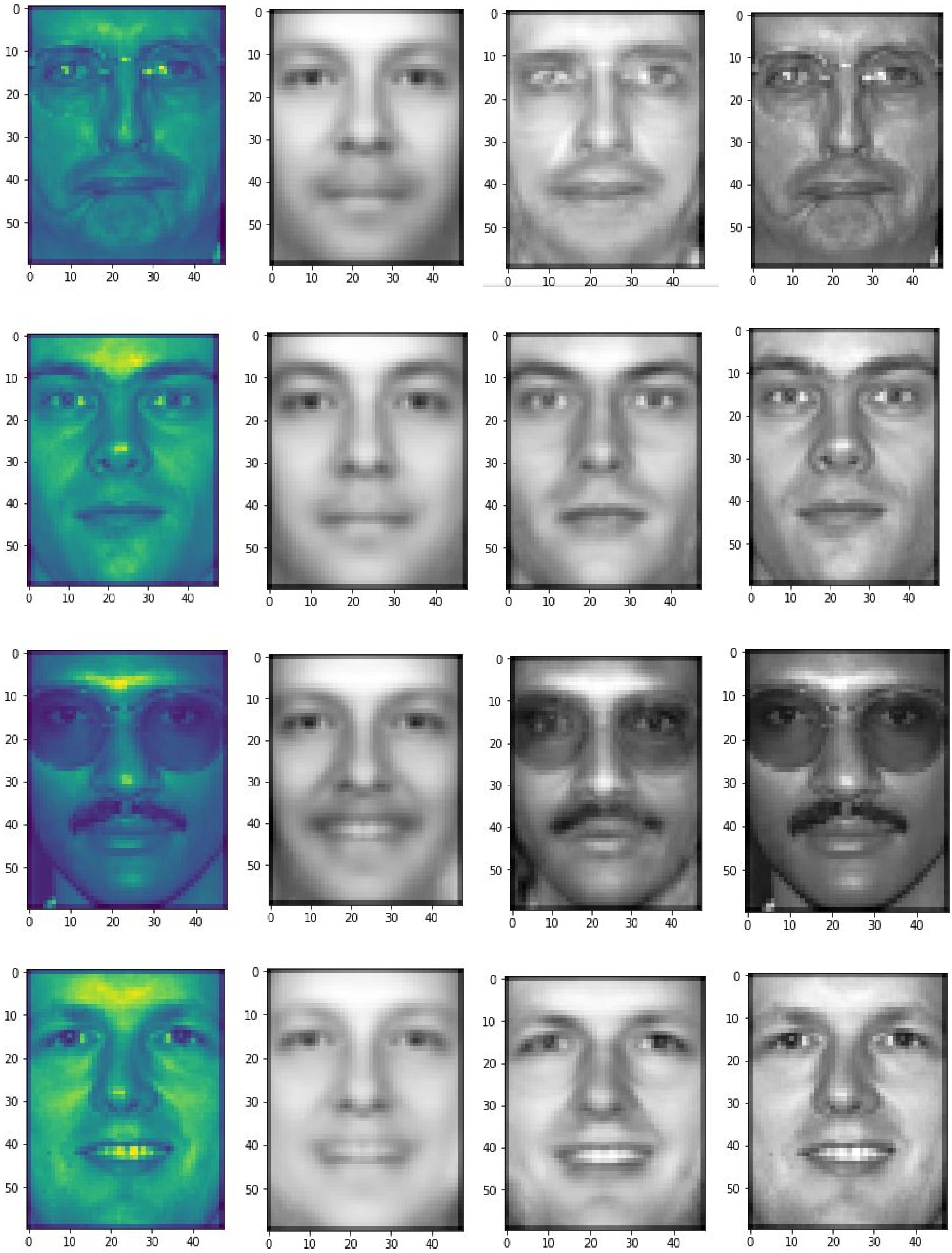
1.6 compress_images() function

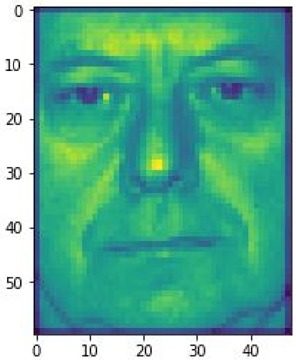
Transformed matrix and a ’k’ value is passed as an argument. Using ’PCA’ method above eigen values and vecotors are computed. And Z_star is also computed with the given ’k’ value and Z. ’Z_star’ is then compressed with the ’k’ number of principal components and the resultant is stored in ’x_compress’ variable.

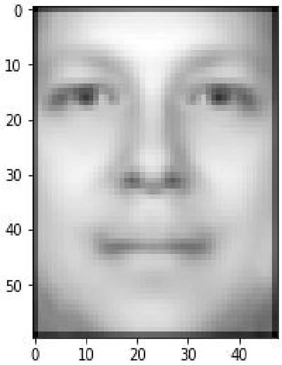
An if statement is used to check if an ’Output’ folder exists, if not will create folder with that name. And ’x_compress’ variable which consists images in flattened form is converted back into its original (60 x 48) matrix is then saved using plot.imsave() in .png format in the output folder.

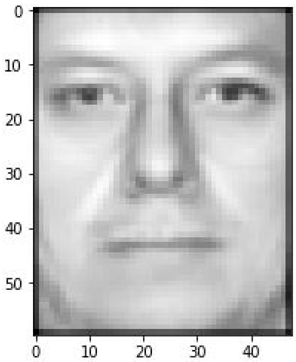
1.7 Original and transormed pictures

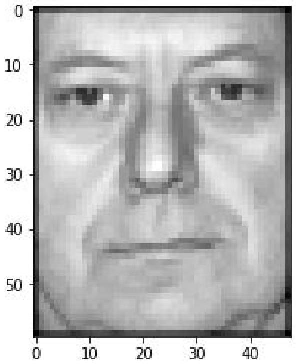
Image-1: original, l. Image 2: k=10, Image-3: k=100, Image-4: k=500




0
10
20
30
40
50
0
10
20
30
40


0
10
20
30
40
50
0
10
20
30
40


0
10
20
30
40
50
0
10
20
30
40


0
10
20
30
40
50
0
10
20
30
40