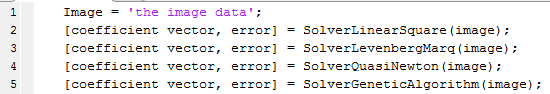
**Read Me**

This section will explain how to run the different sections of source code. In order to run any of the code you will need Matlab R2016B. You will also need the optimisation toolbox installed, the statistics and machine learning toolbox and the image processing toolbox.

To create the library just run ZernikePreCompute(); in Matlab. The 'dim' value can be altered to create a dictionary of different sized images. It is normally set to 256 as this is the size of the images that are currently being used.



All optimisation algorithms are run the same way. They take in one input of an image and return the error and the coefficient vector. The image has to be of the correct size to map to the Zernike library and it has to of the form of a 2D array.



The coefficient vector returned will be 105 digits long and can be used to recreate the image with ZernikeCalcWithPoly(coefficient vector);



For KNN the user inputs a value of k, numcoeffs, and the data being for training into the KNN training method chosen. The SVM only uses the numcoeff Input. Eval is the evaluation matrix holding the information on how the classifier performed. The evaluation matrix holds the confusion matrix o the results of the classifier and the testing error.

