Scientific report regarding the compressed storage and retrieval of images using PCA

Abstract

Many topics regarding image compression and representation are discussed throughout this report. Principal component analysis, Discrete cosine transforms and quantisation are topics referred to when considering the application of compression and image representations. Experimentation of truncated basis, image retrieval and so fourth. Principal component analysis (PCA) aims to calculate an ordered linear combination of eigenvectors and eigenvalues. Given this knowledge the experimentation carried out there is a 4 stage process in regard to this specific report. Initially the dataset is used in order to generate the PCA basis for the images. This is the main focus of the report and most if not all of the stages of experimentation involved some aspect of PCA>

Methodology and Experimental results

Eigenspace representation of images.

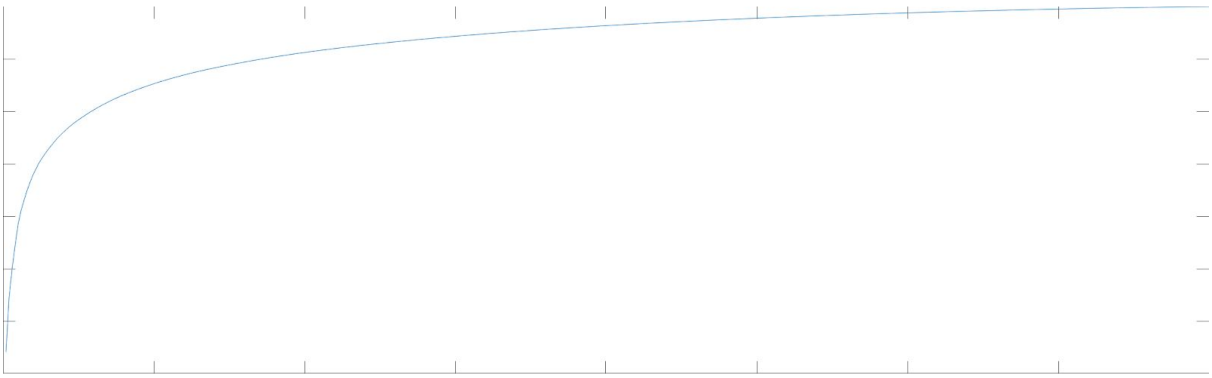
Once the data has been loaded into matlab. Calculating the PCA representation of the images dataset is the first step. This gives us three items regarding information in the eigenspace. V, the set of all basises, represented as eigenvectors. Mu, the average face for the dataset, and L, the set of eigenvalues representative of the orthogonal pair with the eigenvector respectively.

As is seen in the figure on the right, the average face shown on the right is the averaged sum of intensity values in the given dataset. Observationally, we can see that the images are normalized meaning that the positioning of the faces are in line with each other; clearly due to the fact the average values in each corner are extremely low and you can recognise distinct facial features throughout the image.

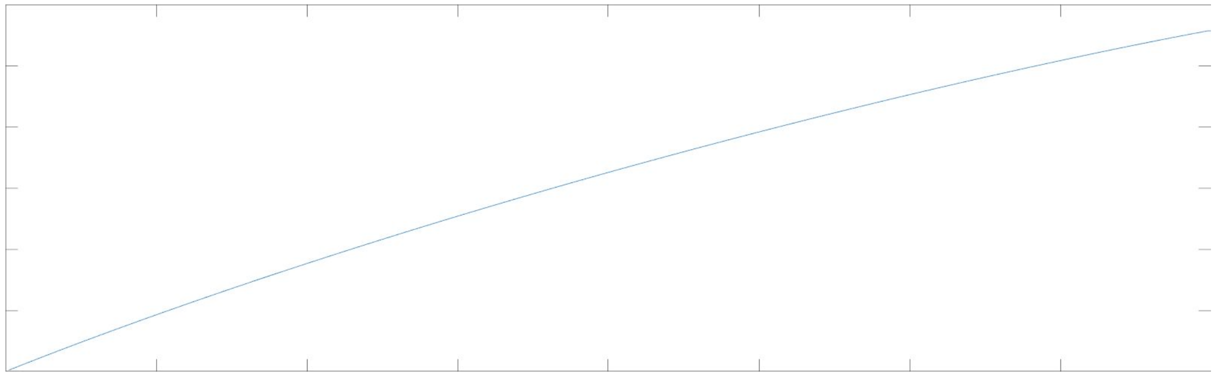
The basis V is representative of a linear combination that is used to generate and retrieve compressed images.

These basis vectors represented by these images can be seen to show the variation in the data, the manner in which PCA represents data dictates that the vectors with the most variation are in the beginning, due to the ordered nature of the data. This can be seen in the images as there is much more clutter and less distinguishable features to be visually observed, take for example, the first and last image are quite drastically different, due to the exponential decrease in variation.

The above plot below shows the cumulative eigen energy given the number of eigenvalues used. The amount of these values used in order to achieve an accurate representation of the original image (85%+). To represent 85% of the data is 2/8ths of the eigenvalues available. This can be inferred to then reveal the energy compaction property of PCA.

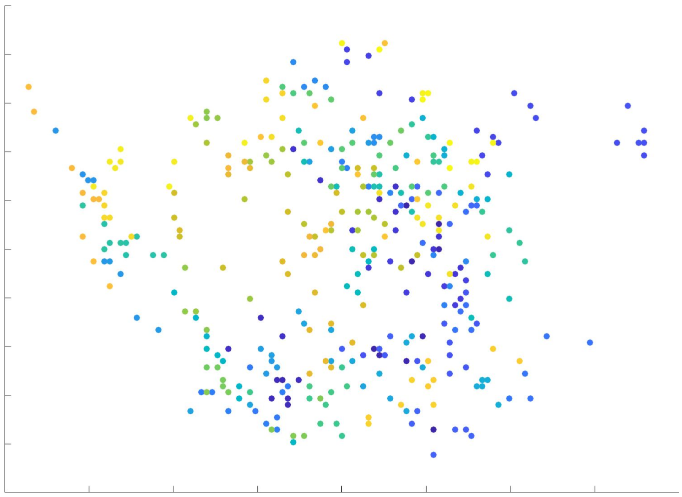


Repeating the same experiment, the data given initially to produce the PCA coefficients and related information is done with a randomly generated set of images rather. The energy compaction plot is given below again.

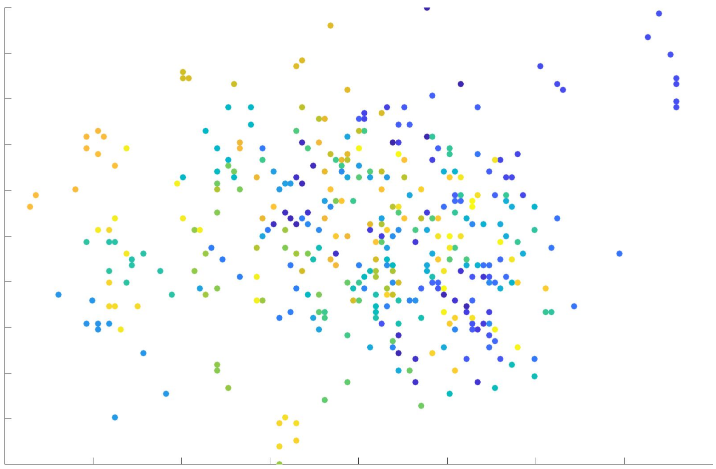


It is observed that this data has a much larger inherit variation between images. This means attempting to reconstruct images given training data produced using PCA is much less effective as for an accurate representation for the data a much higher number of basis vectors are required, significantly decreasing the compaction energy for this specific random dataset.

Furthermore, we experimentally observe the ith vs the jth coefficient in every basis. Plotting them against each other and ensuring that each subject is specified by a different colour. To the right is every 2nd and every 3rd value in the basis plotted. It is clear that the values of each image are grouped together, the colours indicate that as such.

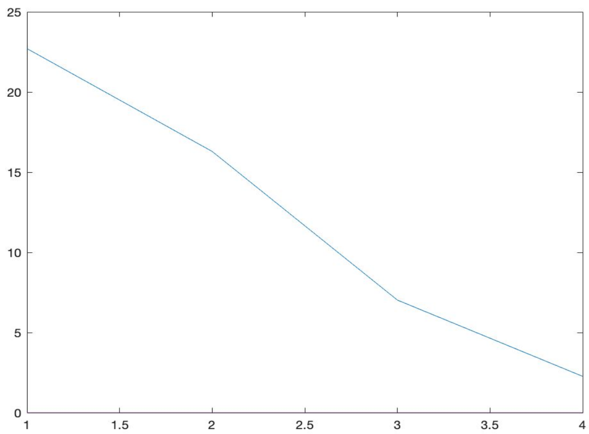


In this plot on the right, every 2nd and every 5th value is used instead of every 3rd value, it is observed that this plot has more variation between colours(subjects), the grouped together values are at a greater euclidean distance and this therefore directly implies a greater mean difference between values.



Experimentation with a truncated basis.

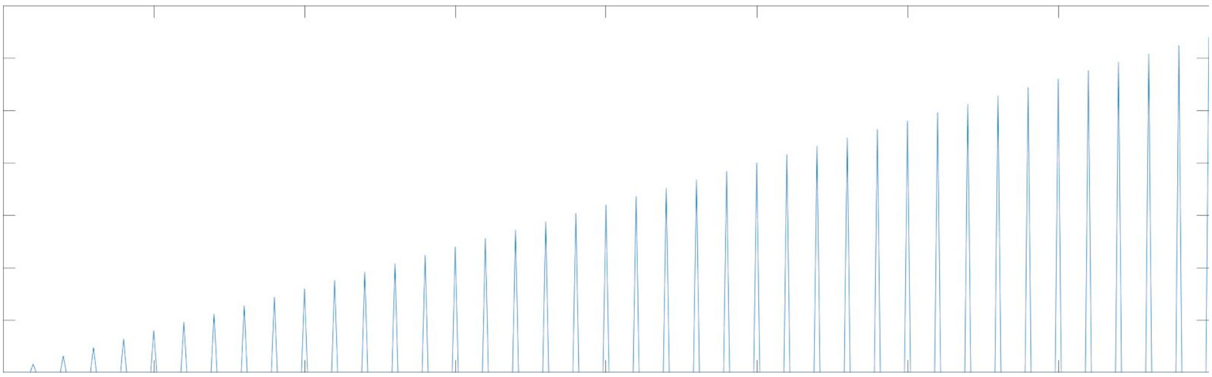
The image shown above is the concatenation of 4 images in which different levels of truncation of the basis vectors V would produce when an image is projected and then unprojected. The first image is the reconstruction given a basis of only the first 50 values



Experimentation with the truncated basis vector was done in observational terms, however to further these inferences the use of error the mean square error calculation gives us an intrinsic difference between two images in this case. The resultant error value based off of the truncation level is shown above.

Applying quantisation.

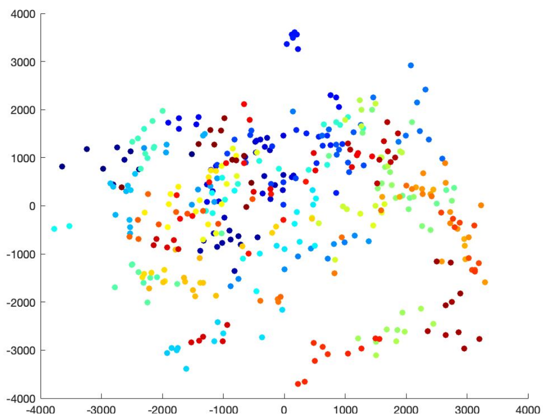
The quantisation of PCA coefficients is extremely straightforward. Coeffs = round(Coeffs*1000), Coeffs = Coeffs ./ 1000; This essentially takes the values in the matrix, this being a float, typically 64 bits long depending on the variation between data. In the case of these images this value is around 140 (+/-15). Rounding this value as an integer removes multiple significant figures, this has a negligible effect on image clarity however the difference in file size is drastic due to the conversion between floating point and binary integer.



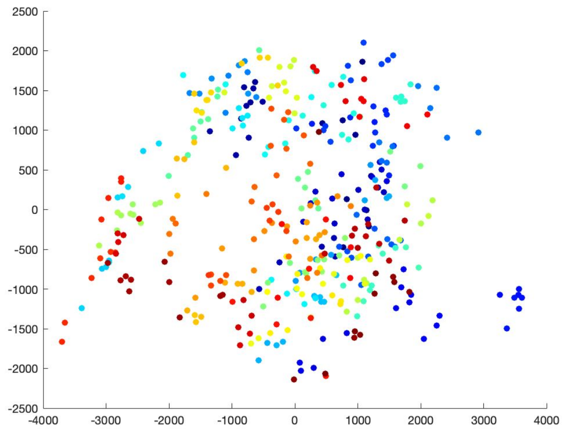
Compression ratio of change in truncated basis(without quantisation) for every 10th integer up to 400.

Image retrieval.

The retrieval of images is a procedure often burdened with computational strain. In this experiment we create a training set of feature vectors which give a series of vectors describing a particular image. On the right there are two plots, the first plot indicating the same process to display the data as the data used in the eigenspace visualisation. In this specific case the first choice is a comparator between the 1st and 2nd values.



The secondary plot shows the data given the 2nd and 3rd values. Just as observed prior to this, these data plots indicate the variation between these three data points. However in this case, the plot is also appended onto by the suspect vector. This allows us to observe that suspect feature vectors when plotted in a euclidean plane, group together



due to their similarity. It is this grouping together that is used in order to detect the subject being used at that specific point.

Image no.25 search

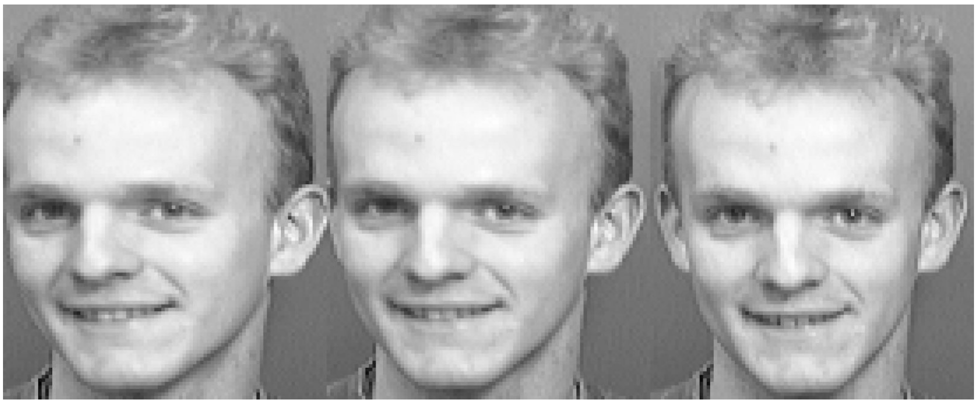


Image no.45 search

Image no.3 search

Image no.5 search

The above images indicate the two closest feature vectors to that of the suspect. These vectors are concatenated after unprojection with the suspect image. In this case you can see the retrieval of images is successful.

As a measure of accuracy, it can be seen that when the process of training data given every image minus the suspect image, if it is to be checked and confirmed that the result is true we can give the percentage of true positive to false positives, as there are no possible false or true negatives as the image is taken from the same dataset in this case. With k=3 using knn, 395/400 or 98.75% of the returned values were true positives, and the rest false positives. With just k=1, the accuracy was given to be 390/400 or 97.5%. This indicates that changing implementing knn algorithm for image retrieval improves accuracy of subject detection by 1.25%.