**Project 3: Principal Component Analysis and Eigenfaces for Face Recognition**

**CS479: Pattern Recognition**

**Terence Henriod**

**April 28, 2014**

**Introduction**

In this project the Principle Component Analysis (PCA) technique was used to demonstrate a dimensionality reduction technique as applied to face recognition.

The data used was a set of 48x60 grayscale images provided by the instructor, originally from the FERET database. 1204 images were used for “training” and 1196 images were used for testing the application developed by the student. The FERET database provides a diverse selection of faces in terms of gender and ethnicity.

The techniques and data described above were used to develop and test an application that found the mean (average) face, various “eigenfaces,” various matching or mismatching occurrences, as well as rejection and acceptance rates in face comparisons.

**Technical Discussion**

*Principal Component Analysis*

Principal Component Analysis seeks to reduce the dimensionality of the data by finding a way to represent the data in a space that still preserves most of the information, yet requires far fewer dimensions to do this.

PCA accomplishes this by finding the unit-length eigenvectors of a data set, and then keeping a subset of the eigenvectors corresponding to a subset of the largest eigenvalues. This a linear combination of the subset of eigenvectors will preserve a large amount of the data, usually with far fewer vectors.

*Computation of Eigenvalues and Eigenvectors*

*Computation Reduction Techniques*

*Determining Similarity*

*Maximum Likelihood Estimation*

It should be mentioned that the result of the Maximum Likelihood Estimations for the mean and covariance parameters of the Gaussian distribution were used for parameter estimation. Maximum Likelihood Estimation simply uses Calculus methods to find the maximum of a likelihood of a given parameter for a given set of data. In short, the well-defined estimates for the Gaussian distribution’s parameters are the mean and covariance that are always presented when discussing such parameters:

The mean µ is estimated by:

|  |  |
| --- | --- |
|  | [1] |

The covariance matrix Σ is estimated by:

|  |  |
| --- | --- |
|  | [2] |

*Multivariate Gaussian Distribution*

It is only fitting that the probability density function (pdf) of the Multivariate Gaussian Distribution is listed:

|  |  |
| --- | --- |
|  | [3] |

This pdf (or variations of it) is used in all parts of the project. In short, we have assumed that all data follows the Gaussian (Normal) Distribution, and use it to compute the likelihood that certain test objects actually belong to a given class.

*Color Spaces*

As it turns out, different color spaces can be used to better encode/represent information. For example, the RGB color space tends to encode a lot of redundant information. The Blue component contains both information that is redundant to Red and Green, but it also contains brightness information, which is unimportant for identifying skin. The 3-dimensional RGB space can be mapped to the 2-dimensional *r*, *g* space by the following:

|  |  |
| --- | --- |
|  | [4] |

|  |  |
| --- | --- |
|  | [5] |

Similarly, the YCrCb space can be used, but since the Y component contains irrelevant luminance information, this 3-dimensonal space can converted from RGB to YCbCr space and also be collapsed to a 2-dimensional space by neglecting the transformation of RGB to Y:

|  |  |
| --- | --- |
|  | [6] |

|  |  |
| --- | --- |
|  | [7] |

These dimensionality reductions both ease the burden of computations as well as get rid of possibly confounding excess information.

*Thresholds and ROC Curves*

In this project, when classifying pixels, arbitrary likelihood thresholds are systematically used. This means that many arbitrary likelihood values were chosen, and if a pixel has a likelihood of belonging to the class of skin pixels lower than the threshold, it was classified as a non-skin pixel; alternately, a pixel was classified as skin only if it had a likelihood (as determined by the Gaussian pdf and estimated parameters) greater than the given threshold.

By computing the rates of false acceptance and false rejection, a Receiver Operating Characteristic curve was generated, which is useful in helping to estimate the ideal threshold to be used for classification.

It should be noted that the term ROC curve is usually applied to tracking the discriminability between multiple classes, but in our case we only test against one class so the curve should be referred to as an *Operating Characteristic curve*.

**Results**

It should be noted that when it comes to images, that if one is expecting a light colored image, it is possible that extreme white values were turned black in the conversion from data to image due to the thresholding of .pgm formats, but the numerical data should be intact in the program/file’s data stores. Images are used to allow for rapid human understanding and not for actual analysis purposes. (This mostly refers to the “eigenfaces”, as the numerical data was stored in text files, not images, for robustness).

*Preliminary Results*



Figure 1: The mean (average) face from the set fa\_H.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| (a) | (b) | (c) | (d) | (e) |
|  |  |  |  |  |
|  |  |  |  |  |
| (f) | (g) | (h) | (i) | (j) |

Figure 2 (a – j): The “eigen-faces” corresponding to the 10 largest eigenvalues/eigenvectors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| (a) | (b) | (c) | (d) | (e) |
|  |  |  |  |  |
|  |  |  |  |  |
| (f) | (g) | (h) | (i) | (j) |

Figure 3 (a – j): The “eigen-faces” corresponding to the 10 smallest eigenvalues/eigenvectors.

*Matching Exercise Results*

It appeared that individuals with unique features (unique facial hair, scars, glasses) were more easily identified by the system. This is of course, intuitive, but should be noted. Where possible, matching images that featured images of the same individual were used.

|  |  |
| --- | --- |
| Test Image | (Correctly) Matched Image |
|  |  |
| 00001\_930831\_fb\_a.pgm | 00001\_930831\_fa\_a.pgm |
|  |  |
| 00733\_941201\_fb.pgm | 00733\_941201\_fa.pgm |
|  |  |
| 00793\_941205\_fb.pgm | 00964\_960627\_fa.pgm |

Figure 4: 3 query images that were “correctly” matched using 80% of the information and their corresponding gallery images. Note that the 3rd image was matched to a relatively similar face (chubby, dark skinned) as a “correct match” but it was not actually matched properly to the individual.

|  |  |
| --- | --- |
| Test Image | (Incorrectly) Matched Image |
|  |  |
|  |  |
|  |  |

Figure 5: 3 query images that were mismatched using 80% of the information and their corresponding gallery images. Note that the set of mismatched images did not vary with amount of information kept.

|  |  |
| --- | --- |
| Test Image | (Correctly) Matched Image |
|  |  |
| 00001\_930831\_fb\_a.pgm | 00001\_930831\_fa\_a.pgm |
|  |  |
| 00733\_941201\_fb.pgm | 00733\_941201\_fa.pgm |
|  |  |
| 00793\_941205\_fb.pgm | 00964\_960627\_fa.pgm |

Figure 6: 3 query images that were appropriately matched using 90% of the information and their corresponding gallery images. One must acknowledge that the results ended up with the same “correct” identifications as before.

|  |  |
| --- | --- |
| Test Image | (Incorrectly) Matched Image |
|  |  |
|  |  |
|  |  |

Figure 7: 3 query images that were mismatched using 90% of the information and their corresponding gallery images

|  |  |
| --- | --- |
| Test Image | (Correctly) Matched Image |
|  |  |
| 00001\_930831\_fb\_a.pgm | 00001\_930831\_fa\_a.pgm |
|  |  |
| 00733\_941201\_fb.pgm | 00733\_941201\_fa.pgm |
|  |  |
| 00793\_941205\_fb.pgm | 00964\_960627\_fa.pgm |

Figure 8: 3 query images that were appropriately matched using 95% of the information and their corresponding gallery images. One must acknowledge that the results ended up with the same “correct” identifications as before.

|  |  |
| --- | --- |
| Test Image | (Incorrectly) Matched Image |
|  |  |
|  |  |
|  |  |

Figure 9: 3 query images that were mismatched using 95% of the information and their corresponding gallery images

Figure 10: The CMC curve for the results. It is doubtful that my program functioned properly, which would explain the strange behaviors of the curve.

Note that there was little difference between the classification results, since only a small amount of information was added with each experiment, and not much of the information is actually necessary to identify with an individual. In fact, many faces have a lot in common, which is why the mean face (Figure 1) is easily recognized as a face.

It should be noted that the distance threshold did need to be varied significantly for the different amounts of information kept. This is likely because when less is measured (the lower amount of information kept), there will be less separation between the data than when there is more to differentiate the data.

*Receiver Operating Characteristic Exercise*

This curve looks odd due to the low functionality of the program used. Its results are not particularly meaningful without serious debugging of the program.

**Appendix**

*Program Listings*

The student would like to thank the makers of the Eigen linear algebra C++ library for their open-source contribution to the world. Hours were saved in coding and debugging thanks to their contribution. Eigen can be found at <http://eigen.tuxfamily.org/>.

The project was coded in C++. This included the use of the C++ STL, particularly the vector template container.

Project\_3\_driver.cpp: The driver program for the assignment. This includes all routines used to accomplish the assignment.

*Complete program listings are attached to this document.*