Dumbledore: A Vision Based Attendance Analyser

Harry Lynas

BSc Computer Science, pm002501@reading.ac.uk

ABSTRACT

This paper presents the proof of concept for a vision based attendance analyser. PCA and the extension WMPCA are two facial verification algorithms that are used in this application to try and determine which students are present or absent in a lecture setting. WMPCA outperforms PCA by better handling environmental issues such as poor illumination, providing the core feature of facial verification needed to implement such a system. Tests are provided that show the algorithms different success rates with different data sets, highlighting how WMPCA is feasible to use for this system. This proof of concept could be developed into a full application due to the testing done and the exemplar audit features provided.

# Introduction

Dumbledore is a proof of concept for a vision based attendance analyser. The concept is that a webcam can be placed in a lecture setting and then automatically detect which students are present or absent in real time. The software would then be able to report to a central database or through email.

To achieve this initially PCA was implemented as a baseline facial verification algorithm. To improve performance and help deal with lecture setting issues such as poor lighting, WMPCA was then implemented at a software level. These two core algorithms provide the backbone for the application.

There is a growing demand for vision based systems and technology has advanced rapidly within the last decade[1]. This article describes the two selected algorithms and how they performed when tested.

# Principle Component Analysis (PCA)

PCA is a mathematically process that is used during the eigenface approach to face verification. Each eigenface represents a distance from the mean face in a set of images. This method of face verification is often referred to as the eigenface approach[2].

## Pre-processing

Before PCA can be performed the input images need to be pre-processed. Pre-processing involves converting the image to greyscale and resizing it to a standard size, and then extracting the pixel data into a one dimensional normalised array. A one dimensional array is created from the two dimensional array of pixels by concatenating on each row of pixels so that it becomes one long array. The values of each pixel are normalised between 0 and 1 to make calculations easier. Without normalisation the RGB (red, green, blue) value of the pixel is used which is often near the maximum value of an integer. If the value is near the maximum value of an integer then when calculations are performed with these values integer overflow errors start to occur resulting in data corruption and invalid results.

## Eigenface

Once the data has been pre-processed PCA can be performed on it and the eigenfaces retrieved. The mean is subtracted from each image, then the covariance matrix is calculated from the image data with mean subtracted. The eigenvalues and eigenvectors can be calculated from the resulting covariance matrix, and then the principle components are retrieved by sorting the eigenvalues descending and retrieving the amount wanted. Not all of the eigenvectors are required because the ones with a low eigenvalue describe little change. Figure 1shows some example normalised eigenfaces.

245929

Figure 1. Example Normalised Eigenfaces

Once the eigenfaces have been determined an unknown image can be projected onto the eigenspace to determine the unknown images weights and thus calculate the Euclidean distance. The image with the lowest distance to the unknown image is most likely to be the same person if under a set threshold, if it is too great a distance and exceeds the threshold then the unknown image can be assumed to not be in the database. If the Euclidean distance is zero then the image is an exact match on every pixel.

# Weighted Modular Principle Component Analysis (WMPCA)

WMPCA[3] is an extension of PCA where each image of a face can be split into regions and then PCA performed on each region. The same pre-processing stage is done as with PCA, however the regions are also created during this stage by splitting the image horizontally.

Whether the results convey a match with an unknown image is determined by a weighted sum of errors from each region. The unknown image is also split into the same number of regions. This algorithm performs with a higher verification success rate than PCA due to being able to match on key regions such as the mouth or eyes.

WMPCA has been implemented at a software level rather than at a hardware level which the original algorithm paper achieves. The WMPCA algorithm runs significantly faster than the PCA algorithm due to the nature of how each region can be processed on a separate thread. Each thread runs concurrently on modern CPU’s with the ability to multithread and thus more of the systems resources can be taken advantage off.

# Optimisations

To achieve a greater success rate in both PCA and WMPCA the first three eigenfaces are discarded. This is because the first three eigenfaces primarily describe changes in illumination and brightness which are not desired. This is because illumination and brightness creates the greatest distance from the mean face due to it having the most effect on the image. It is much more desirable to have the eigenfaces that contain information about the facial features.

An alternative solution that could have been implemented to deal with the illumination and brightness problem would have been to make sure there are images of each individual under different lighting conditions in the training set[4]. This however requires far more data to then be collected and processed and thus was deemed not to be a viable solution.

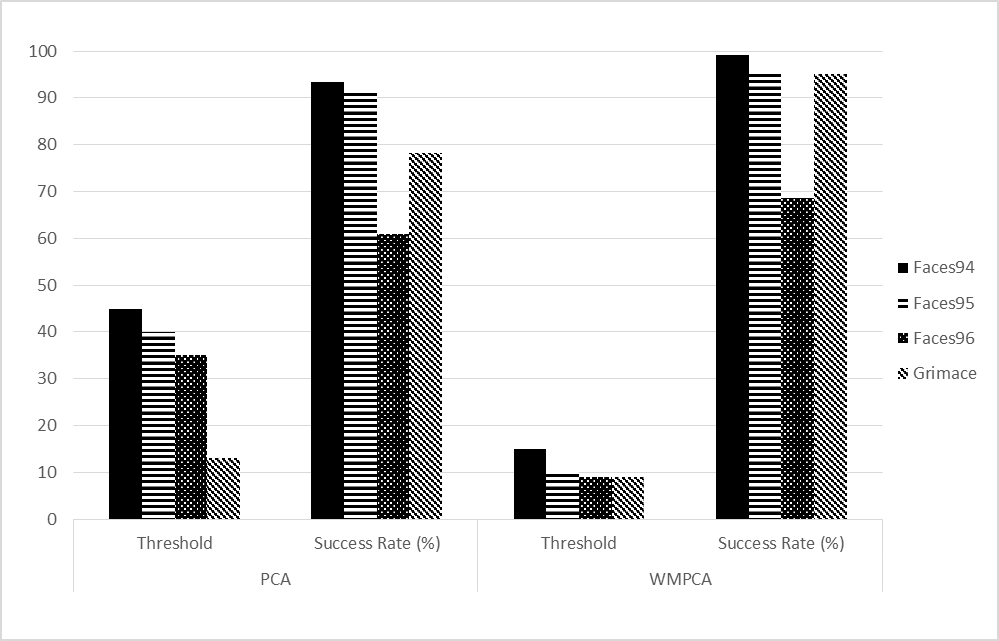


Figure 2. Testing the University of Essex Facial Databases

# Testing

The University of Essex provides several facial image databases[5]. Each database is of an increasing difficulty to learn, starting at ‘faces94’, up to ‘faces96’, then ‘grimace’. 80% of eigenfaces were kept and the standard image size was set to 50x50 pixels. When WMPCA was tested three regions were used. There are 395 individuals making 7900 images in this image set. The results of these tests can be seen in Figure 2.

It can be observed that there was a general trend in that the lower the number of pixels to handle resulted in a lower threshold required, whether this is lower pixels by less images, smaller regions, or less principle components kept. The grimace database was also stated to be the hardest to learn by the University of Essex, however the results obtained here contradict this. It was noted that the grimace database had far more consistent lighting and backgrounds than the faces96 database which is the probable reason behind it being easier than expected. Both algorithms struggled with the faces96 database achieving the lowest success rate. However the faces96 database was also the most similar to a lecture situation where the pose and illumination changes between each subject greatly. This means the results are much more useful as a guideline on how a real world application would perform.

It can also be noted that when analysing the results manually that there are exceptions where images are found to be lower than the threshold set but do not match, which suggests that some of these results will be not be completely accurate, however it gives a good baseline for the performance over a large set of data which is predicted to be hard to learn. When using an automated testing system in facial verification there will always be false matches and false match fails.

The faces96 database was tested with a much lower percentage of principle components kept, as seen in Table 1. The success rate was a lot higher likely because the components that describe small changes in the image are mostly discarded now, and the faces96 database suffers from small pose and illumination changes throughout the database. Subsequently the large amount of images for each subject allows for a good match rate with the information left after discarding a lot of data.

Table 1. Results of Testing Faces96 Database

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Eigenfaces (%)** | **Threshold** | **Success Rate** |
| PCA | 0.05% | 18 | 82% |
| WMPCA | 0.05% | 5 | 84.95% |

# Miscellaneous

To support the face verification system a graphical user interface powered by JavaFX[6] was implemented. This allows for the algorithms to be applied to a live webcam input feed and to add unknown faces to a database as they are found. The live input feed works by acquiring still images from a webcam every small interval and performing the recognition and verification process upon them. This process is performed fast enough to achieve a high amount of frames per second.

The live feed camera can be used to subsequently create databases for a lecture setting, adding each student as they are found and assigning them a name. The system will remember he learnt information within a flat-file database and subsequently know the student names for when reporting. Databases can be loaded on a per lecture basis to improve accuracy and reduce computation time, or a global database for all students can be used. There is also the option to test the algorithms with a set of images for when testing and measuring the algorithms performance.

The project is split up into two core projects – the graphical user interface (GUI) and the application program interface (API). During the testing for the data collected and shown in Figure 2 and shown in Table 1, only API functions were called from a temporary testing project as the GUI was not required. The GUI only serves to further improve the proof of concept of how this can be used in a lecture setting and provides useful functionality for managing this situation.

OpenCV[7] is used to detect the pixel locations of any faces on any given still image. From this data the faces can be extracted, pre-processed, and given to the facial verification algorithm of choice.

An exemplar audit feature has been implemented into the API whereby faces are detected in real time over a period of time from a live webcam input feed and then the results are emailed – noting which known people were absent and present, as well as any unknown people. To learn an unknown person the person must be captured and a name assigned to them.

# Conclusions

The tests conducted prove that the success rate with different databases is suitably high enough to be used in a lecture setting. Each face detected can be verified and the results over a period of time sent to admin or a central database. The faces can be redetected and retested at a set interval to verify similar results are retrieved throughout the lecture and to prevent errors from being reported. WMPCA is the better algorithm to use of the two since it seems to cope better with illumination and brightness changes, as expected.

Finding and setting the threshold to use is the most difficult task to accomplish when using this system. A possible future solution would be to have an individual threshold per student, however this would then need to be updated as more students are added to the database.

Using image dimensions of 50x50 pixels per face seems to be more than sufficient to handle accurate face verification. At 50x50 dimensions there are 2500 pixels per face, which quickly becomes computationally expensive when the facial verification algorithms need to run. PCA can thus take a long time to learn a set of faces, however WMPCA is significantly faster due to the splitting of regions. A 50x50 image split into three horizontal regions, as done in the tests in this report, becomes only 50x16 images thus resulting in only 800 pixels per region. If computation time becomes too long then the standard region size could be reduced without having too much of a drastic impact on the accuracy of the facial verification algorithms.

Currently when WMPCA is used the regions are split at equal points on the horizontal axis. However as a future possible improvement that could improve accuracy for facial verification, facial recognition algorithms could be used to detect the exact coordinates of key facial features such as eyes, mouth, and nose, and then these used as regions. This would be feasible as they could always be resized to a standard size if needed without losing too much key information.

Future research into other possible facial verification algorithms can be done if the success rate was wished to be higher, however increasing the accuracy without significantly having an impact on performance will be difficult. The highest performing algorithm to date is GaussianFace[8] which outperforms humans for the first time. However it is too computationally expensive to run in a feasible time frame.

This proof of concept provides the resources required to develop this into a full product. Competitors to this have similar concepts, such as Aurora[9] which specialises in “world leading face recognition systems” and boasts being able to handle attendance. However such a company is entirely closed-source and in a commercial setting. This software provides the basis for the development of a competitive product.

**Acknowledgement:** M. Manjunathiah for project supervision and guidance.

# references

1. W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips, “Face Recognition: A Literature Survey”, National Institute of Standards and Technology, (2003), pp1, [Online] http://www.face-rec.org/interesting-papers/General/zhao00face.pdf
2. M. Turk, A. Pentland, “Eigenfaces for Recognition”, [Online] http://www.face-rec.org/algorithms/PCA/jcn.pdf
3. A. P. Kumar, S. Das, V. Kamakoti, “Face Recognition using Weighted Modular Principle Component Analysis”, [Online] http://www.cse.iitm.ac.in/~vplab/publi\_journal/conference/IP\_028\_239.pdf
4. W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips, “Face Recognition: A Literature Survey”, National Institute of Standards and Technology, (2003), pp42, [Online] http://www.face-rec.org/interesting-papers/General/zhao00face.pdf
5. L. Spacek, “Computer Vision Science Research Projects”, (20/06/2008), [Online] http://cswww.essex.ac.uk/mv/allfaces/index.html
6. Oracle, “JavaFX – The Rich Client Platform”, [Online] http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html
7. OpenCV, “Open Source Computer Vision”, [Online] http://opencv.org/
8. C. Lu, X. Tang, “Surpassing Human-Level Face Verification Performance on LFW with GaussianFace”, [Online] http://luchaochao.me/papers/GaussianFace.pdf
9. Aurora, “Aurora is the UK’s leading provider of face recognition technology”, [Online] http://www.facerec.com/index.html

All web addresses referred to in this paper were verified on 24th March 2015.