Pattern Recognition CW1

# Abstract

# Introduction

From security purposes such as criminal identification by law enforcement[[1]](#footnote-1) to entertainment such as automatic face image tagging on social media, machine based face recognition is powerful as a non-invasive tool and has a plethora of applications. Since the first Automatic Face Recognition system developed by Kanade in 1973, technology in face recognition has shifted from simply detecting and determining relationships among individual facial features of subjects to linear projection appearance-based methods[[2]](#footnote-2) such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

To develop a real-time system, it is essential that the recognition algorithm is accurate, fast and simple. It is by these metrics of accuracy, efficiency and simplicity that this paper seeks to implement and evaluate both PCA and LDA methods in face recognition.

Chapter II elaborates on the pre-processing of the given image data in face.mat, and while Chapter III details the maximising the efficiency and simplicity of PCA as a feature extraction method, Chapter IV expounds on improving the accuracy of PCA for face recognition. (Add in Chapter V and VI later)

# Data Pre-Processing

To start off, the given face.mat contains a matrix X (2576x520) and a vector l (1x520). The columns of X represent each of the 520 face samples, with each sample being a 56 pixels by 46 pixels image. The elements of l then show that there are 52 individuals (or classes), each with 10 sample images taken with varying backgrounds, clothing and angles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Class\_1\_(1)*** | ***Class\_1\_(2)*** | ***Class\_1\_(3)*** | ***Class\_1\_(4)*** | ***Class\_1\_(5)*** |
| C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_1.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_2.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_3.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_4.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_5.jpg |
| ***Class\_1\_(6)*** | ***Class\_1\_(7)*** | ***Class\_1\_(8)*** | ***Class\_1\_(9)*** | ***Class\_1\_(10)*** |
| C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_6.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_7.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_8.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_9.jpg | C:\Users\Benjamin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\class_1_10.jpg |

To evaluate the effectiveness of subsequent face recognition methods, the provided face data has to be first split into a training set which is used to train the algorithm and a testing set which is used to measure the recognition accuracy of the classifier.

The ratio in which both sets are partitioned will influence classifier performance, which can be separated into metrics of training error and testing error. Training error is obtained when training data is run through the learned model again and testing error is obtained when unseen data is run through the same learned model.

* Training with too few data points – unable to predict future values well
* Training with too many data points – overfitting
* Show training vs test error with increasing training sets
* Decide on split
* Training an algorithm with few data points will easily have lower training error since it is likely to find curves that touch exactly those points of data
* But test error will be high since with little training data, it will be unable to predict future values well
* No cross validation since the data sets are too small, creating a cross validation set
* We can do a X-fold split of data and re-run tests

# Eigenfaces

To design a robust face recognition algorithm, it is important to encode as much relevant information in a face image as efficiently as possible

PCA involves determining a set of eigenvectors to form a subspace

http://www.vision.jhu.edu/teaching/vision08/Handouts/case\_study\_pca1.pdf

http://maths.dur.ac.uk/users/kasper.peeters/pdf/face\_recognition/PCA/TurkPentland91CVPReigenfaces.pdf

https://pdfs.semanticscholar.org/76a7/fc9d87736c8383576865cf50403e53e74848.pdf

# Application of Eigenfaces

# References

https://pdfs.semanticscholar.org/76a7/fc9d87736c8383576865cf50403e53e74848.pdf

1. https://www.fbi.gov/news/testimony/law-enforcements-use-of-facial-recognition-technology [↑](#footnote-ref-1)
2. https://cdn.intechopen.com/pdfs-wm/51031.pdf [↑](#footnote-ref-2)