



LINEAR ALGEBRA IN DIGITAL HEALTH

MXB201 Group Project

Group 14

Joade Lennox	n10523669
Liam Ferrante	n10467718
Liam Hyland	n10790357
Louis Yanagisawa	n10221221

Contents

Introduction.....	1
PART I.....	1
Methodology	1
Potential issues and roadblocks.....	3
Pseudocode element.....	4
Implementation	5
Findings	5
Recommendations.....	6
PART II.....	7
Methodology	7
Potential issues and roadblocks.....	7
Findings	8
Recommendations.....	12
Final Conclusions.....	13
References.....	14

Introduction

Speed. Accuracy. Reliability. Three fundamentals of which every piece of medical machinery aims to encapsulate. With the massive technological explosion in recent decades, being able to get fast and accurate medical analysis has revolutionised how we interact and diagnose surface, and more importantly, below the surface ailments. Behind every great machine, however, is a complex system of programs and algorithms, crucial to outputting accurate and almost more importantly, readable information. The program presented in this report aims to fill a gap in the market, allowing for precise, convenient, and easy to interpret visualisation of water diffusion in the brain, extracted from raw MRI signal files. As it stands, the program currently outputs fractional anisotropy and diffusion direction maps.

PART I

Methodology

Critical to ensure rapid, effective, and correct diagnosis of brain ailments, the MRI (Magnetic resonance imaging) system uses a complex system of powerful magnets to scan reactions in the brain, with flexibility on the focus of the area of tissue.

The area of focus for this section of the report is fitting the diffusion tensor presented by Jiang, et al. [1] and elaborating its role in a mathematical equation that allows for quick analysis of raw input from an MRI machine.

$$S = S_0 e^{-b \mathbf{g}^T \mathbf{D} \mathbf{g}} \quad [1] \quad (\text{Equation 1})$$

Which to suit this reports' purposes can be simplified to:

$$\log\left(\frac{S}{S_0}\right) = -b \mathbf{g}^T \mathbf{D} \mathbf{g} \quad (\text{Equation 2})$$

Where each element is defined as:

- S_0 represents the ‘natural’ signal that was detected before use of any form of gradient pulses were employed.
- The b is defined as a scalar value in which contains information about strength and timing of the pulses.
- The element g_i denotes the unit vector that hold crucial information pertaining to the direction of the corresponding pulse.
- D in this context is a three-by-three symmetric positive definite matrix, that is, $D \in \mathbb{R}^{3 \times 3}$.
- And finally, S , representative of the transformed signal, output by equation 1 for each individual g_i directional value.

Exploring the diffusion SPD matrix further, it’s six elements are unknown and will be a main element of section one.

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix} \quad [1]$$

Since the matrix D is symmetric, there are only the six unknowns ($D_{xx}, D_{xy}, D_{xz}, D_{yy}, D_{yz}, D_{zz}$) in the equation stated above the matrix D . We can divide both sides by the known signal value of S_0 , with the log of both sides taken to gather the equation that was highlighted above (equation 2):

$$\log\left(\frac{S}{S_0}\right) = -b\mathbf{g}^T \mathbf{D} \mathbf{g}$$

This equation is assumed to hold for each of the directions g_i . Assuming 64 g_i ’s are used, estimating D from the signal data from each voxel results in a least squares problem with 64 equations and the 6 unknowns from D . As stated by Jiang, we are able to present the 6 unknowns as a vector:

$$\tilde{D} = [D_{xx} \ D_{yy} \ D_{zz} \ D_{xy} \ D_{xz} \ D_{yz}]$$

We also note that \mathbf{g} represents the normalised diffusion sensitisation gradient vector:

$$\mathbf{g} = (g_x, g_y, g_z)^T \quad [1]$$

After a natural logarithm transformation was taken on both sides, the vector \mathbf{g} is now written as:

$$\tilde{\mathbf{g}} = [g_x^2 \ g_y^2 \ 2g_x g_y \ 2g_x g_z \ 2g_y g_z]^T$$

From these vectors, Liang shows that we then can have the following:

$$\sum_{i,j=x,y,z} (g_i g_j) D_{ij} = \tilde{\mathbf{g}}^T \cdot \tilde{D} = \ln\left(\frac{\left(\frac{S}{S_0}\right)}{b}\right) = ADC \quad [1]$$

Where ADC is the apparent diffusion constant. Again, as presented by Liang, applying different gradients such as $g_k (k = 1, \dots, K; K \geq 6)$ and obtaining the signals S_k , a system of equations can be generated:

$$\tilde{\mathbf{g}}_k^T \cdot \tilde{D} = \ln\left(\frac{S_k/S_0}{b_k}\right) \quad [1]$$

We can create an overdetermined linear system by allowing A to be a 64×6 coefficient matrix and B to be a 64×1 right-hand side vector. The resultant system takes the form:

$$A\tilde{D} = B$$

- Mean diffusivity mapping

The Mean Diffusivity map is a visualisation of the average diffusion rate of a given sample area covered in the scan. Higher MD values typically correspond to higher fluid content and fluid mobility while lower values usually correspond to less fluid content and mobility.

Extracting the eigenvalues from D, these values can be used in creating the MD map by taking the average eigenvalue for each pixel.

$$MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{e}$$

- Fractional Anisotropy Map (FA)

The Fractional Anisotropy map is a way of visualising the diffusion asymmetry of a voxel, characterized by its eigenvalues (Elster, 2019).

Following this concept, once the collection of \tilde{D} matrices are found, the eigenvalues are easily extracted and input into the following equation,

$$FA = \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}} \quad [5] \quad (\text{Equation 3})$$

The result is a matrix with dimensions of the original image, containing elements ranging from zero to one. Expanding on the definition of the FA values, a perfect example of isotropic diffusion will be categorised as having an FA value of zero, where $\lambda_1 = \lambda_2 = \lambda_3$. As a voxel exhibits progressive diffusion anisotropy, it's respective eigenvalues will become unequal, with the inequality causing equation 3 to tend towards a value of one. The chosen method of visualisation will be a greyscale figure with larger values being represented by progressively brighter shades of white.

- Principal Diffusion Direction Map

One extremely helpful dataset that can be modelled is the Principal Diffusion Direction Map. It expands on the Fractional Anisotropy map and allows for visualisation of the diffusion directions in a three-dimensional space.

Specific colour choices are of no consequence, and for this element will be defined by red, green and blue for the x, y and z axis respectively. Convention, however, dictates that the orientation of the primary eigenvector be responsible for the hue while the FA value controls brightness.

The PDD value for each pixel can be calculated by multiplying its FA value by cosine of the angle the first eigen vector creates with the three axes.

$$\begin{aligned} PDDM_{red} &= FA * \cos(\alpha) \\ PDDM_{green} &= FA * \cos(\beta) \\ PDDM_{blue} &= FA * \cos(\gamma) \end{aligned}$$

Potential issues and roadblocks

- Image corruption

File corruption is an always a possibility issue when working with any form of software, and may be the result of interrupted processing, incorrect formatting, hardware issues, driver, or software clashes or simply a by-product of unrelated OS issues. Since most of these options reside outside the domain

of any analysis software such as the one being developed here, its responsibility of original file preservation falls onto the operator.

Concerns surrounding the software designed in this report being the potential source of file corruption can be mitigated by completely avoiding use of the original file to begin with in processing. A temporary copy could be created, and all processing done from there. While this would certainly preserve any corruption of the raw file, it could be considered overkill and would only be implemented if it became a non-isolated problem.

- *Abnormal amounts of noise*

Baring similarities to a corruption scenario, little can be done in processing (at this level) to clean up the data without additional information such as the result of a systematic error, wherein the ‘random’ noise can be replicated, and raw scans corrected.

- *Dealing with negative numbers*

As such is that the nature of machines is imperfect – issues can (and will) arise when collecting and processing the data irrespective of how meticulous the method is designed may be. In this context, signal strength readings are only as good as the receiver and its associated software. While negative readings are provably infeasible in theory, the reality is that it is a genuine possibility and therefore requires all measurements to undergo a simple processing algorithm to ensure that all values meet the requirements of a natural log transformation. It was decided to take the absolute value of any negative value. It should be noted however, experimentation did occur with making negative values equal to zero, but it was observed to have little effect on the final image.

Pseudocode element

Load values S, S0, b, g and mask

$A = [gx^2, gy^2, gz^2, 2gxgy, 2gxgz, 2gygz]$

Error handling of S and S0

Applying mask to S, fixing negative and zero values because of the log() in the next step

for every row,column,slice (voxel) in S

$$B(\text{row},\text{column},\text{slice}) = \log(S(\text{row},\text{column},\text{slice})/S0(\text{row},\text{column}))/-b$$

$d = A \setminus B$

D = matrix of d

for every row and column (pixel) in S

$$[eVec, eVal] = \text{eig}(D)$$

$$e1, e2, e3 = eVal$$

$$MD = \text{average}(eVal)$$

$$FA = \sqrt{\frac{(e1-e2)^2 + (e2-e3)^2 + (e1-e3)^2}{2*(e1^2 + e2^2 + e3^2)}}$$

unit vectors x, y and z in the 3d plane

v = largest vector in eVec

PDDM_red =FA*angle created by v and x

PDDM_green =FA*angle created by v and y

PDDM_blue =FA*angle created by v and z

display(MD)

display(FA)

display(PDDM)

Implementation

The implementation of the algorithm in MATLAB can be found in the partI.mlx. The figures produced, mean diffusivity map, fractional anisotropy map and principal diffusion direction map can be found below.

Findings

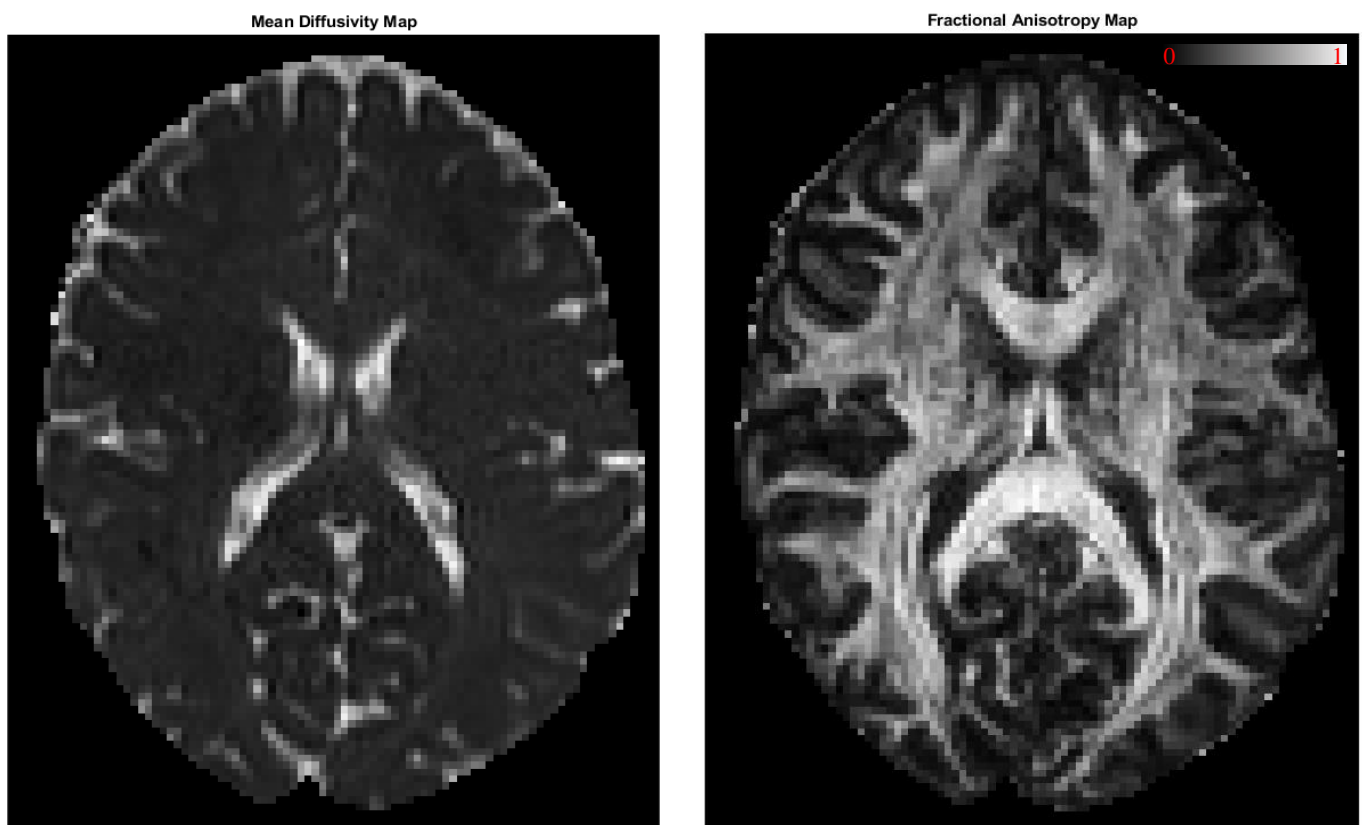


Figure 1. Shown here are the Mean Diffusivity map (left) coupled with the Fractional Anisotropy map (right).

From the MD map it is shown that the middle part of the brain and the edges have the highest mean diffusivity value indicating it contains the largest fluid content and fluid mobility.

From the FA map it is shown that most of the brain has ununiform diffusion with most of the brain having white lines, indicating that the diffusion direction is unequal in all direction.

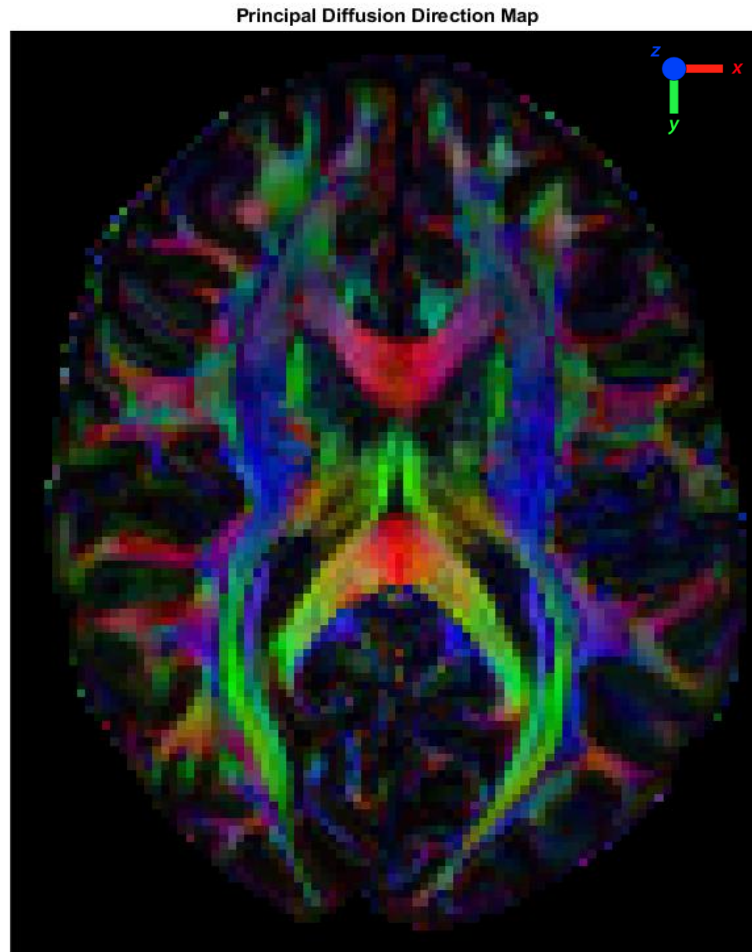


Figure 2. The Principal Diffusion Direction map, useful for determining the predominant diffusion direction in a three-dimensional space, with axis direction represented by colour. Red for left and right, Green up and down and blue for in and out of the page.

From the PDD map it is hard to tell what direction the diffusion is happening, but it does indicate that diffusion is occurring in all directions in the scan. The PDD map closely follows the FA map just with colour which makes sense as instead of mapping the uniform distribution of the scan its mapping the actual direction of the diffusion. With red indicating x axis diffusion, green indicating y axis diffusion and blue indicating z axis diffusion, it is safe to assume that diffusion is not occurring in any one direction over the other with diffusion in all directions present throughout the map.

Unfortunately, a larger than anticipated quantity of noise was observed surrounding the perimeter of the brain slice. It is entirely possible that this is a problem with the supplied mask, and which step of processing it was implemented in.

Recommendations

The current process works on a single MRI with 64 slices but can be extended to work with multiple slices over multiple patients. To facilitate these changes, the code can be modified so that there are more values or have the values of a higher dimensions to contain more information to accommodate

for the increase in patient numbers, and can also feature an increased number of loops to process and iterate all this information.

Some ideal modifications would be to reduce the errors in the data. There is a concerning amount of lost data as the negative values need to be dealt with somehow, as the results can interfere with the data - especially due to a log function transformation when calculating B, since the log of a negative number produces complex value and the log of 0 is infinity. Although measures were taken, such as rounding them to zero or taking the positive value, this process does not infer what the original values might have been.

PART II

Methodology

Important for facial recognition software, eigenfaces play a critical role in training software to identify shapes and features associated with the human face. The eigenface method is an approach that attempts to create a generalised ‘mean face’ from a collection of human faces (with $r \times c$ pixels). For each face, some may be diseased, and some not. An approach to creating an ideal system is to let the whole database of faces be a single matrix. Let a matrix be $A \in \mathbb{R}^{rc \times N}$. Every column of this matrix A represents a patient’s greyscale $r \times c$ image that was reformed into a tall column vector within \mathbb{R}^{rc} . The source of faces database used for this investigation is the publicly available Extended Yale Face Database [2] [3], of which have been pre-processed for convenience [4]. The database contains 1000 images of which all have been utilized. Initially, a mean face was taken from the 1000 faces sourced. This was simply done by taking the average of all the column contained in A . Next, mean centring was applied. Again, quite a simple process was used where the mean face was subtracted from the matrix A , leaving behind faces that have some quite distinct traits, like a moustache or larger nose. For clarity, we let describe A using a reduced singular value decomposition:

$$A = U \Sigma V^T$$

Where $U \in \mathbb{R}^{rc \times N}$, $V \in \mathbb{R}^{N \times N}$ and $\Sigma \in \mathbb{R}^{N \times N}$. We also let σ_i from smallest to largest be the diagonal of the matrix Σ . The left singular vectors, $u_i \in \mathbb{R}^{rc}$, are known as the eigenfaces. The leading ones show the most prominent differences between A and the mean face. Sometimes the singular values will decay rapidly which requires a different approach to reconstruction. A linear combination of the first v eigenfaces, $v \ll N$, means that every face in the set would be represented by a v -dimensional coordinate vector. Or in other words, its projection onto the “eigenface space.”

Presenting an image of a face through its projection onto the v -dimensional requires the truncated singular value decomposition, or truncated SVD, to be calculated. It is known that U is rectangular and V and Σ are square. The above used form of $A = U \Sigma V^T$ can be translated into the following form using MATLAB notation:

$$A(:, i) = U(:, 1:v) \Sigma(1:v, 1:v) V(i, 1:v)^T$$

We can let a vector, $x_i = \Sigma(1:v, 1:v) V(i, 1:v)^T$, to then get:

$$A(:, i) = U(:, 1:v) x_i$$

Where x_i is the coordinate vector of the individual column of $A(:, i)$ in terms of the left singular vectors $U(:, 1:v)$.

Potential issues and roadblocks

- Lighting

As highlighted by Kuang-Chih (2005, pp.684), incorrect lighting or lack thereof can significantly inhibit a software’s ability to detect and extrapolate the required data from an image. However, a

complete set of images using a single face with lighting from multiple angles can help overcome this limitation, allowing for leniency with the proportion and angle of light. Conveniently, the database being employed contains a variety of fixed pose faces with light sources projected from multiple horizontal and vertical angles.

- *Unusual or unique facial features*

Something that could affect the final ‘mean’ face are outliers to generic facial structure. Individuals with prominent scars, birth defects or facial hair will influence the outcome with a much more significant factor than those with more generic features. Along a similar vein, to reduce the amount of unnecessary noise, glasses should be removed prior to an image being taken. After a summary analysis of the Yale database, it was concluded that no significant outlier or correction needed to be made.

- *Non-centred images*

To analyse effectively the images must contain, in an ideal situation, a forward facing and centred profile, with the nose in common location between images. Any deviation from this standard will introduce a blur effect in the final image.

Findings

By representing each image (of a face) as a single column of data, the analysis will have to work with one thousand *very* tall matrices. Averaging the values across these one thousand columns and creating the image that this mean column represents will produce the ‘Mean Face’. It represents the average face of the face data and as such, will have very generic facial features.



Figure 3. The Mean face as per calculated from the Yale database.

Returning to the one thousand column matrix that represents the one thousand faces, if this mean column created above, is subtracted from each of these columns, we will be left with a matrix of one thousand columns (referred to as the mean-centred matrix) that represents exactly how each face is different from the mean face, this is known as mean centring.

The most economical way to describe the mean-centred matrix using limited information is by using the reduced singular value decomposition:

$$A = U\Sigma V^T$$

Where $U \in \mathbb{R}^{rc \times N}$ where $r = 192$ (vertical resolution of an image of a face), $c = 168$ (horizontal resolution of an image of a face), $N = 1000$ (number of faces we are examining), $V \in \mathbb{R}^{N \times N}$ and $\Sigma \in \mathbb{R}^{N \times N}$, with the singular values σ_i on the diagonal of Σ ordered from largest to smallest.

The left singular vectors $u_i \in \mathbb{R}^{r_c}$ are eigenfaces and the leading eigenvectors represent the features that most strongly capture the differences between individuals' faces in the data set and the mean face (due to the ordering of the singular values σ_i in Σ). This is typically referred to as Principal Component Analysis. Illustrate below are the first twenty eigenfaces for this dataset.

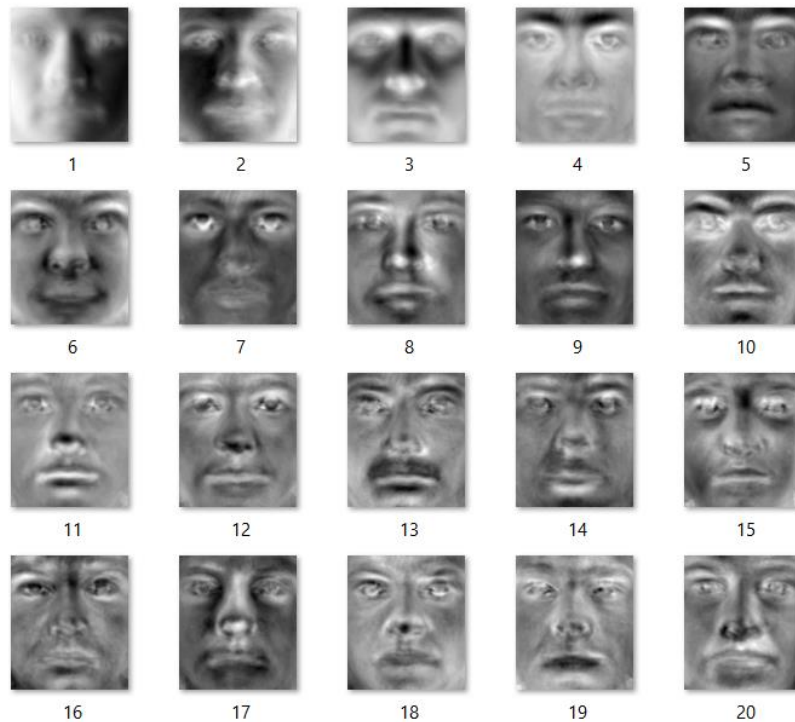


Figure 4. The first 20 leading Eigenfaces.

These leading eigenfaces represent the twenty most significant features that assist in distinguishing between faces (at a mathematical level).

However, if the singular values decay rapidly, it might be possible to reconstruct the original faces using only a relatively small number ($v \ll 1000$) of leading eigenfaces and linear combinations between them (truncated singular value decomposition). The reconstructions would be identified by v -dimensional coordinate vectors with the projections of these coordinate vectors onto the 'eigenface space'.

Observing the first few singular values:

```
1.4981
1.3162
0.8023
0.5178
0.4554
0.4080
0.3869
0.3534
0.2989
0.2900
⋮
```

Figure 1 First few singular values

Evidently, these singular values do appear to decay quickly. But how many leading eigenfaces will be required to sufficiently represent the data whilst creating reconstructions of much smaller data size? On observing the relative decay in magnitude for the singular values instead, it produces:

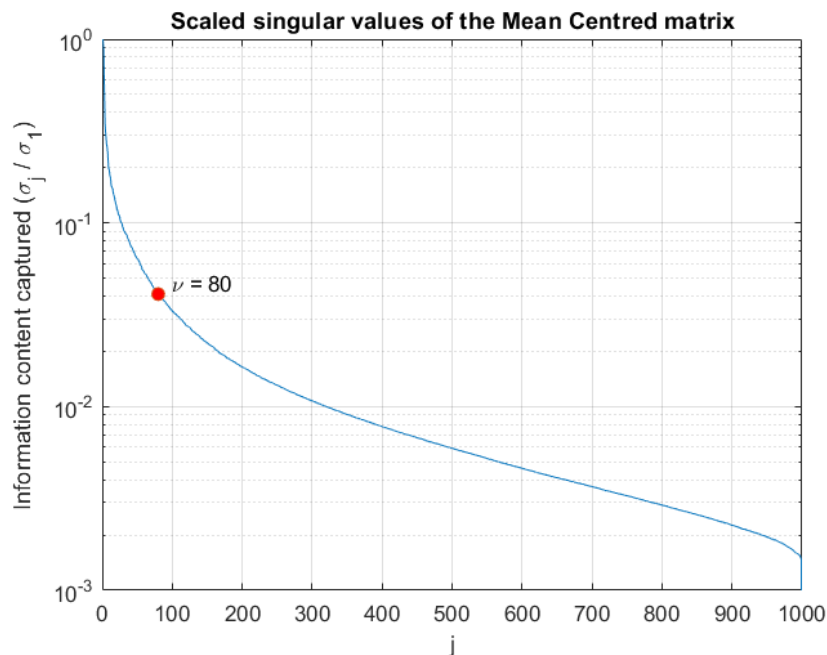











Figure 5. The scaled singular values of the mean centred matrix.

This illustration vividly demonstrates the behaviour of the decay in the singular values. Additionally, it can be observed that 80 ($\nu = 80$) leading eigenfaces would describe about 1.7 orders of magnitude of information content of the Mean Centred matrix.

What exactly does it look like to have different amounts of information content retained? Below compares the first 3 faces of the data, each represented by linear combinations of: 1, 5, 10, 20, 40, 60, 80 and 100 leading eigenfaces.

No. Leading Eigenfaces	Face 1	Face 2	Face 3
1			
5			
10			

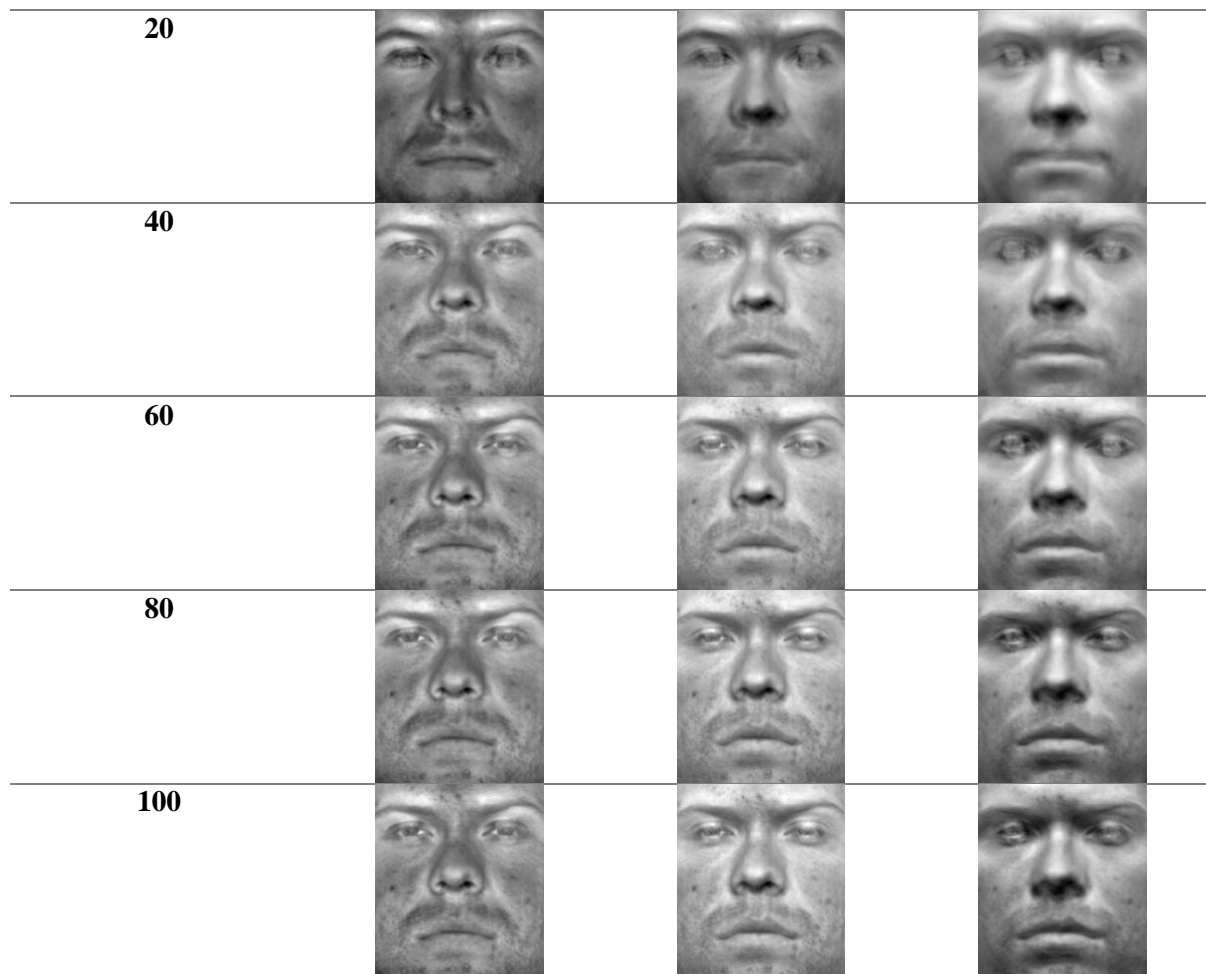


Figure 6. Linear combinations of the first three eigenfaces at 1, 5 10, 20,40, 60, 80 and 100 leading eigenfaces.

As expected, increasing the number of eigenfaces used to reconstruct the data, increases the quality of the reconstruction. Further, this demonstrates the acceptable quality of using only 80 leading eigenfaces to reconstruct faces. Changing from using 60 eigenfaces to 80 eigenfaces, there is a noticeable increase in reconstruction quality (especially in face 3), while transitioning from 80 eigenfaces to 100 does not lead to such notable differences.

How does file size compare between reconstructions using 60, 80 and 100 leading eigenfaces?

Table 1. File sizes according to leading eigenfaces

No. Leading Eigenfaces	Face 1	Face 2	Face 3
Original Image	17.0KB	16.8KB	16.9KB
60	15.9KB	15.4KB	14.8KB
80	16.0KB	15.8KB	15.2KB
100	16.1KB	15.9KB	15.4KB

As expected, the reconstructions have a smaller file size than the original image, and the file size increases with the number of leading eigenvalues used.

Conclusively, using 80 leading eigenfaces provides decent reconstruction of the data by capturing about 1.7 orders of magnitude of information content, whilst having a smaller file size than the original.

Recommendations

Modifications

- The MATLAB code could be modified to be clearer and simpler. There exists repeated functionality (especially column to matrix representation and vice versa) that could be condensed into a reusable function extracted into its own file. It should be noted however, almost all of the existing code is not hard-coded, it is *flexible* and will function with different sizes of images and amounts of data.
- Further, a more rigorous method could be implemented for the selection of $\nu = 80$. Rather than selecting this value by inspection of a graph, perhaps a mathematical process could be constructed that finds an optimal value that is still $\ll 1000$, whilst producing reconstructions of acceptable quality.

Improvements

- A short video could have been produced that directly compares the effects of increasing or decreasing the number of leading eigenfaces used for reconstruction. It could animate through multiple reconstructions of a particular face as ν changes thus allowing for a greater understanding of the relationship between ν and quality.
- It would be interesting to see how beginning with images of much greater resolution impacts the significant features identified (Would they be roughly the same? Would there be a new feature only available to high resolution data be found?). Additionally, how much would the file size be reduced in comparison to a detailed image compared to a reconstructed image with varying ν values?
- Using coloured images would also be an intriguing area to investigate. This problem could be solved with a following a similar solution, instead of grayscale images, the images could be separated into 3 datasets each representing the 3 digital primary colours (R, G, B). This would translate the 3-dimensional data back into familiar 2D data.

Adaptation

- Instead of using faces as the data, MRI brain scans could be used. Performing singular value decomposition on a mean centred MRI brain scan data known to be that of a healthy brain, would provide insight into the features that make up a '*healthy brain*' and the same could be done with a '*diseased*' brain. The coordinate vectors of MRI brain scans of new patients could then be compared to these found features of *healthy* and *diseased* brains.
- Computers are extraordinarily fast with computations, including the processing of matrices, unless they are incredibly large (i.e., MRI brain scans). Truncated singular value decomposition allows us to take huge amounts of data and extract the key components, the essence of the data, without unnecessary detail (leading eigen-brains / singular-brains). This can lead to early diagnosis and intervention for many patients.

Final Conclusions

Medical analysis and digital health have become essential in the last few decades. The ability to use advanced technology to aid investigations has enabled for more effective evaluations. With detailed MRI scans capable of reproducing images of the brain and face we are in the best position to give vital aid in digital health. The ability to detect anomalies in both the brain and face will allow professionals to diagnose possible health issues much earlier in their development, greatly impacting an individual's survivability. Overall, it is standout that this technology is absolutely essential in allowing digital health to do wonders for the world.

References

- [1] Jiang, H., Van Zijl, P. C., Kim, J., Pearlson, G. D., & Mori, S. (2006). DtiStudio: resource program for diffusion tensor computation and fibre bundle tracking. *Computer methods and programs in biomedicine*, 81(2), 106-116.
- [2] Lee, K. C., Ho, J., & Kriegman, D. J. (2005). The Extended Yale Face Database B. <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>.
- [3] Georgiades, A., Belhumeur, P., & Kriegman, D. (2001). From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6), 643-660.
- [4] K.C. Lee, J. Ho, & D. Kriegman (2005). Acquiring Linear Subspaces for Face Recognition under Variable Lighting . *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5), 684-698.
- [5] Allen D. Elster (2019). Diffusion Tensor Imaging. <https://mriquestions.com/dti-tensor-imaging.html>.
- [6] Ugurbil, K., Van Essen, D., et al. (2017). Human Connectome Project (Young Adult). <https://www.humanconnectome.org/study/hcp-young-adult>.