Linear Discriminant Analysis for Recognising Differences

Burcu Erarslanoglu (5948355) and Joris van de Weg (5174821)
{b.erarslanoglu,j.j.vandeweg}@student.tudelft.nl

Advanced Digital Image Processing
Delft University of Technology

Abstract. Linear Discriminant Analysis (LDA) presents a novel way to classify and predict group membership. When the focus is specifically on neuroimaging data, the classification between pathological conditions from healthy brain scans becomes possible. Due to the large size of imaging data for example with MRI, a dimensionality reduction with Principal Component Analysis (PCA) is often necessary. This study applies the PCA-LDA algorithm to Magnetic Resonance Imaging (MRI) scans of the brain for Alzheimer's disease classification, utilizing the Open Access Series of Imaging Studies (OASIS) dataset which has 4 types of data. Through the PCA-LDA algorithm, the aim is to have a one-dimensional representation of data for classification. Performance evaluation is conducted using k-fold cross-validation. The classification error is found in terms of training and testing errors for a range of principal components r which are used in the dimensionality reduction with the PCA part of the algorithm. By doing so, it is possible to comment on the presence of an optimal r^*. The framework demonstrates potential applications to other neurodegenerative diseases, enhancing early diagnosis and treatment planning.

1 Introduction

Linear Discriminant Analysis (LDA) is a method used to effectively classify and predict group membership. It is a method that is used in image processing applications as well as many other domains. One important application of LDA is in neuroimaging, where scientists distinguish pathologically different conditions. It aims to find differences related to various diseases and disorders from healthy brain scans. The proposed framework in this study is for Alzheimer's disease using Magnetic Resonance Imaging (MRI) scans; however, it could potentially be applied to many other comparative studies such as schizophrenia disorder [1]. This versatile nature of the LDA approach is important because it allows for a broader applicability of LDA in medical diagnostics; therefore, making it a valuable tool for identifying specific pathological changes across the linear combinations of predictors that best separate two or more classes of objects or events [2].

For this analysis, a cohort containing both patients and controls is analyzed. However, when dealing with high-dimensional data, which is almost always the

case for MRI, cohort size is typically small compared to the imaging data. This makes discriminating the data an ill-posed problem [1]. Therefore, Principal Component Analysis (PCA) is first performed on high-dimensional data to achieve dimensionality reduction. In this assignment, the PCA-LDA algorithm is used in MATLAB to classify the pathological changes that occur in the brain due to Alzheimer's disease, providing insights that are critical for early diagnosis and treatment planning [3].

2 Motivation

Alzheimer's disease causes progressive degeneration of brain cells, and it is the main cause of dementia where the diseased person has a significant decline in thinking and capacity for independent daily functioning. Patients experience symptoms of memory loss, change of personality and other cognitive impairments [3]. Early diagnosis of Alzheimer's through diagnostic aids is crucial as they enable healthcare professionals to mitigate the progression of the mentioned symptoms, that are experienced with Alzheimer's, with certain drugs and therapies [3]. Classification methods like LDA play a vital role in enhancing the accuracy and timeliness of Alzheimer's disease diagnosis. With this assignment, the PCA-LDA algorithm is applied in order to achieve Alzheimer's disease classification using MRI scans. This framework can potentially be applied to other neurodegenerative diseases and different imaging modalities.

3 Methodology

This section begins with an explanation of the OASIS dataset. Then the procedures and models used in the PCA-LDA algorithm and classification error computed with the cross-validation are discussed step by step.

3.1 Explanation of the Dataset

The Open Access Series of Imaging Studies (OASIS) dataset used in this assignment consists of high-resolution T1-weighted MRI scans [4] where statistics like the person's age, gender, Total Intracranial Volume (eTIV), and many more can be accessed. The dataset has 4 different types of data, their details are presented in table 1 below.

Table 1: Data types and their explanation from the OASIS dataset

Type	Explanation
OASIS A	Fully-sampled (100%) data with original values.
OASIS B	Sub-sampled (50%) data with original values.
OASIS C	Fully-sampled (100%) data with regressed values of age and eTIV .
OASIS D	Sub-sampled (50%) data with regressed values values of age and eTIV .

One relevant statistic for this assignment is the Clinical Dementia Rating (CDR). CDR assigns ratings on a 0-5 point scale (0 = absent; 0.5 = questionable; 1 = present, but mild; 2 = moderate; 3 = severe; 4 = profound; 5 = terminal) [5]. In the OASIS dataset, there are 182 MRI scans belonging to 60 possible Alzheimer's patients with Mild Cognitive Impairment (MCI) and 122 healthy controls, indicated with CDRs of 0.5 and 0, respectively. The data is preprocessed to allow tissue segmentation into gray matter and white matter probability maps. Segmentations are aligned to the template using non-linear warping and smoothened using a 4 mm FWHM Gaussian kernel. The resulting volumes for the fully-sampled dataset contain 121×145×121 voxels for every subject [4]. Figure 1 shows a transverse and coronal representation of the OASIS A data for a random subject.

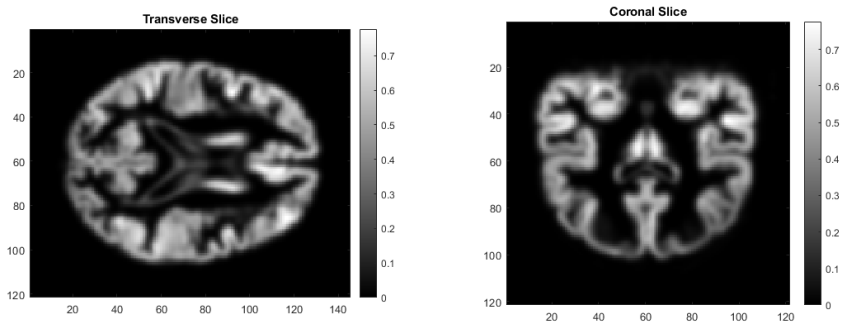


Fig. 1: OASIS A data of subject 50 in the transverse and coronal slices for an MRI scan of the brain

3.2 Analysis Procedures and Models

There are two classes in the Alzheimer's MRI data, which will be distinguished using the PCA-LDA algorithm. The number of patients and controls are formally defined with m_p and m_c, respectively and according to the OASIS dataset, the numbers are given with $m_p = 60$ and $m_c = 122$. The cohort size containing all the samples is denoted with m.

3.2.1 Reshaping

Before PCA can be implemented, the original voxel data of size $n_x \times n_y \times n_z$ has to be reshaped as a 1D vector [1]. This vectorized voxel data belongs to one sample out of the m samples. Horizontally stacking this vectorized data for every sample, results in a matrix denoted as $\mathbf{X}$ of size $(n_x \cdot n_y \cdot n_z) \times m$. Each column vector of size $n_x \cdot n_y \cdot n_z = n$ can be represented with x_i where $i = 1, ..., m$, with $m = 182$. In most discrimination problems, the cohort size "m" is much smaller than the total number of voxels "n". These are called *small sample size problems*

[2] where $m \ll n$. This is no different in this problem where the OASIS dataset is used. Working in such a space makes the problem highly ill-posed and therefore difficult to solve. The PCA-LDA algorithm solves this problem by converting the n-dimensional space to a single dimension space as follows

$$\mathbb{R}^n \xrightarrow{\text{PCA}} \mathbb{R}^r \xrightarrow{\text{LDA}} \mathbb{R}. \tag{1}$$

As shown in eq. (1), the data is first reduced from n dimensions to r dimensions using PCA, and then further reduced to a single dimension using LDA. As a result, two-step dimensionality reduction is achieved.

3.2.2 Principal Component Analysis

PCA algorithm works by finding the principal components, which are the eigenvectors of the covariance matrix of the data that correspond to the largest eigenvalues. By doing so, dimensionality reduction of the data can be achieved while preserving as much information as possible. The covariance matrix, often referred to as the total scatter matrix, [1] is conventionally written as

$$\mathbf{S}_t = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{m}\sum_{i=1}^{m}\mathbf{A}_i\mathbf{A}_i^T = \frac{1}{m}(\mathbf{U}\mathbf{U}^T)_{n\times n}. \tag{2}$$

This total scatter matrix in eq. (2), presents the mean-centered data which can be written with $\mathbf{A}_i = (x_i - \bar{x})$ which is then extended to the $\mathbf{U}$ matrix [6] by incorporating the summation sign. The eigenvalues and eigenvectors of $\mathbf{S}_t$ are found with the eigendecomposition

$$\mathbf{S}_t\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{3}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues and $\mathbf{V}$ is the matrix of eigenvectors. However, due to the problem being highly ill-posed as discussed before, computing $\mathbf{S}_t$ with $\mathbf{U}\mathbf{U}^T$ is extremely hard. Therefore, instead of deriving the eigendecomposition for $(\mathbf{U}\mathbf{U}^T)_{n\times n}$, it is derived for $(\mathbf{U}^T\mathbf{U})_{m\times m}$. This will result in the same eigenvalue matrix $\mathbf{\Lambda}$ but with a different eigenvector matrix $\mathbf{V}$. Therefore, the eigendecomposition problem is written as

$$\mathbf{S}_t\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \quad \rightarrow \quad \frac{1}{m}(\mathbf{U}^T\mathbf{U})_{m\times m}\mathbf{V}_{m\times m} = \mathbf{V}_{m\times m}\mathbf{\Lambda}_{m\times m}. \tag{4}$$

The next step is to sort the eigenvalues and eigenvectors in a descending order as $\mathbf{\Lambda}^{\text{sorted}}$ and $\mathbf{V}^{\text{sorted}}$ respectively. By doing so, it is possible to select the principal components that capture the most variance because the eigenvectors corresponding to the largest eigenvalues capture the most variance. Later, the matrix $\mathbf{V}^{\text{sorted}}_{m\times m}$ is reduced to $\mathbf{V}^{\text{sorted}}_{m\times r}$ by selecting the top r principal components. To achieve the mapping between $\mathbb{R}^n$ and $\mathbb{R}^r$ as stated in eq. (1), the following computation first takes place

$$\mathbf{P}_{n\times r} = \mathbf{U}_{n\times m}\mathbf{V}^{\text{sorted}}_{m\times r}. \tag{5}$$

Next, the projection of x_i onto $\mathbf{P}$ results in vectors of length r with the computation

$$x_i' = \mathbf{P}^T(x_i - \bar{x}) = \mathbf{P}^T\mathbf{A}_i. \tag{6}$$

This concludes the PCA part of the algorithm since x_i' is now in the space $\mathbb{R}^r$.

3.2.3 Linear Discriminant Analysis

LDA is performed on the PCA transformed data x_i' from eq. (6) to find the direction that best separates the two classes of patients and controls. The goal is to end up in a one-dimensional space of $\mathbb{R}$ from eq. (1). From the CDR rating, the numbers of patients and controls are available with m_p and m_c, respectively. By using these, the class means could be found as $\bar{x}_p$ and $\bar{x}_c$. The mean vectors of length r which are denoted with x_i' from eq. (6) is denoted with $\bar{x}'$. These are incorporated when writing the class-within scatter matrix [7],

$$\mathbf{S}_w = \frac{1}{m}\sum_{j=\{p,c\}}\sum_{i=1}^{m_j}(x_{ij} - \bar{x}_j)(x_{ij} - \bar{x}_j)^T \tag{7}$$

and class-between scatter matrix [7],

$$\mathbf{S}_b = \frac{1}{m}\sum_{j=\{p,c\}} m_j(x_j' - \bar{x}')(x_j' - \bar{x}')^T. \tag{8}$$

Then, the generalized eigendecomposition problem is solved for $q = \mathbf{S}_w^{-1}\mathbf{S}_b$ with

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \quad \rightarrow \quad q\mathbf{V} = \mathbf{V}\mathbf{\Lambda}. \tag{9}$$

As done for the PCA case, the eigenvalues and eigenvectors of the LDA matrices in eq. (9) are sorted in a descending order. The eigenvector that corresponds to the largest eigenvalue is taken as the first discriminant direction, in other words, the LDA direction which can be denoted with q'. Next, it is possible to compute

$$a = \mathbf{P}q' \tag{10}$$

and following the computation of a, with the projection of x_i onto a, a one-dimensional scalar value for each x_i^* is found as

$$x_i^* = a^T(x_i - \bar{x}) = a^T\mathbf{A}_i. \tag{11}$$

This concludes the LDA part of the algorithm since x_i^* is now in the space $\mathbb{R}$.

3.3 Assessment Metrics

The PCA-LDA algorithm maps the high dimensional input data for every subject using the found computational vector a into a scalar x_i^*. To properly evaluate

the effectiveness of this dimensionality reduction technique, it is necessary to assess the performance of the vector a on new, unseen data rather than the data used to generate a. This ensures that the algorithm's ability to generalize is tested appropriately. To achieve this, k-fold cross-validation will be used to split the data into a train and test set. This is a procedure of separating the data into k number of groups, where every group will be used once as the test set. Using cross-validation will result in a more accurate estimate of the algorithm's generalization ability by training and testing on different subsets of the data, thus preventing overfitting. For every fold, the computation vector a will thus be derived from its training set, which will be referred to as a_{train} and is computed similarly to eq. (10) as

$$a_{\text{train}} = P_{\text{train}} q'_{\text{train}}. \tag{12}$$

The x_i^*s are then computed for the train and test set data as

$$\begin{aligned} x^*_{i,\text{train}} &= a^T_{\text{train}} \mathbf{A}_{i,\text{train}}, \\ x^*_{i,\text{test}} &= a^T_{\text{train}} \mathbf{A}_{i,\text{test}}. \end{aligned} \tag{13}$$

For classifying the subjects, the nearest mean classifier is sufficient due to the algorithm's simple 1-dimensional output, which refers to one feature. The classification of $x^*_{i,\text{train}}$ is conducted by finding the closest class-mean, based on the labelled $x^*_{i,\text{train}}$ data.

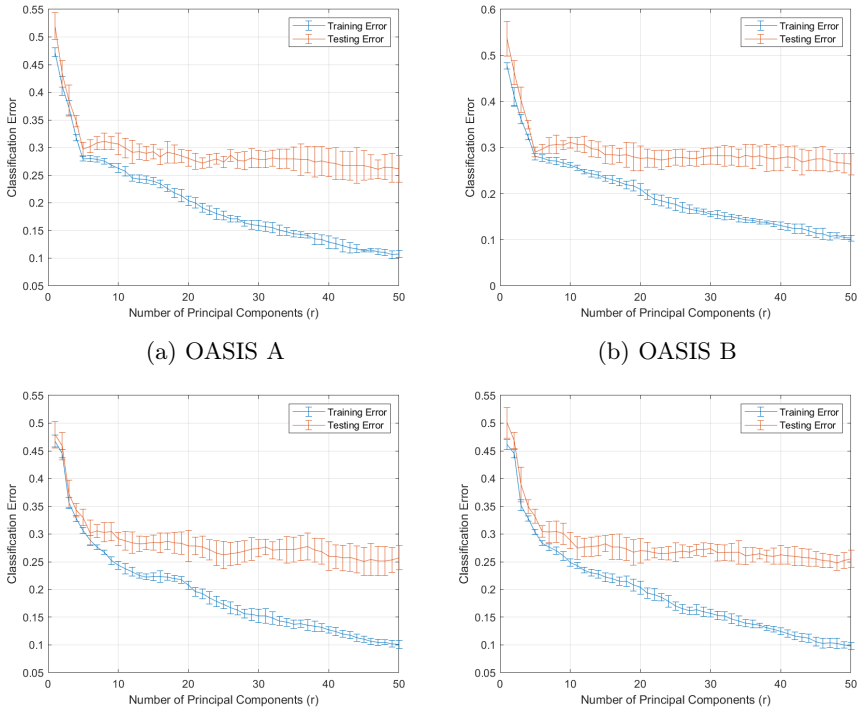
Finally, with these classifications, the performance for every fold can be measured. To ensure robust generalized results, this whole procedure is repeated l times. For every run, based on the classifications, the accuracy and standard deviation are stored for evaluation. The performance can be assessed based on the classification error, but also on the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These metrics can help in the evaluation of the algorithm's performance from multiple perspectives, ensuring a balanced assessment of its strengths and weaknesses. These results will be presented in Section 4, which are generated using variable values of $k = 5$ and $l = 10$.

For gaining insights into the behaviour of the performance when changing r, this repetitive process is carried out for a range of different r values. Which aims to determine the 'optimal' parameter r^* for the PCA process. This total assessment is done for every dataset to compare the effects of sub-sampling and regressing the effects of age and eTIV.

Additionally, because the data types OASIS A, B, C, and D are different in terms of image sizes, the generation of the data for evaluation is expected to be different. Therefore, the total time for the code to complete l runs is stored for every data type for comparison purposes.

4 Results

The code implemented the PCA-LDA algorithm and tested its accuracy of classification for different values of r by applying k-fold cross-validation for a repetition of l times. The MATLAB code and some additional results can be found at [8]. The results in terms of classification error are shown in Figure 2. The classification performance regarding TP, TN, FP, and FN are shown in Figures 3 and 4 for OASIS A and C data, respectively. Lastly, the time it took to generate these plots for every data type can be found in Table 2. The reader may refer to Table 1 again for an explanation of the data types of OASIS A, B, C, and D.



(a) OASIS A

(b) OASIS B

(c) OASIS C

(d) OASIS D

Fig. 2: Classification error with standard deviation bars for the train and test dataset per principal component

Subfigure 2a shows the training and test error for the OASIS A dataset. It shows that adding more principal components than $r = 5$ does not result in a big decrease in the testing error. The training error is showing a decrease in the classification error for using more principal components, making the gap between the errors bigger. Sub-sampling the data does not cause a worse performance, as can be seen in Subfigure 2b. Subfigures 2c and 2d, show the results for regressing the effects of age and eTIV. It can be seen that it has a small increase in performance when compared to the original dataset.

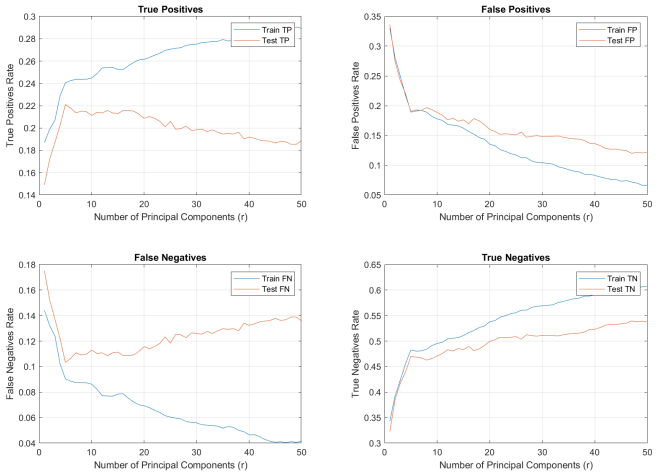


Fig. 3: Performance metrics over a range of principal components for OASIS A

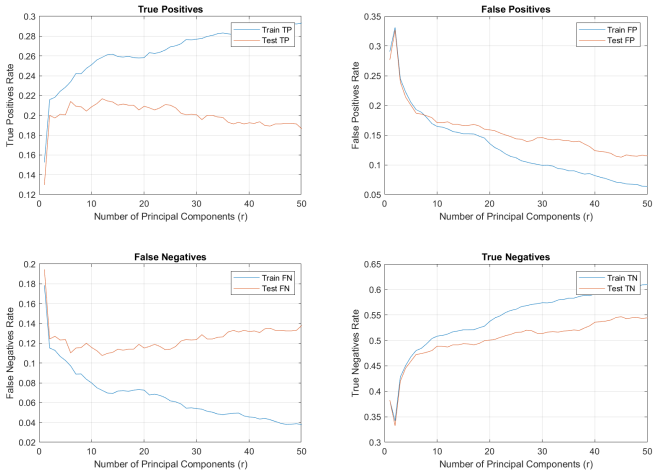


Fig. 4: Performance metrics over a range of principal components for OASIS C

Figures 3 and 4 show the classification performance regarding TP, TN, FP, and FN. When the data types from Table 2 are considered, evaluating the efficiency of the PCA-LDA algorithm for OASIS A and C essentially means evaluating the effect of age and eTIV on the classification performance. When Figures 3 and 4 are compared for their curves, it is seen that the performance seems to be similar mostly with some small variability. Both datasets show a higher r, a higher TN rate, and a lower FP rate. However, the FN and TP rates do not show better results for a r higher than 10. These findings will be discussed more in section 5.

Table 2: Performance metrics for different datasets

Dataset	Size	Elapsed Time [s]	Average Time per Run [s]
OASIS A	121×145×121×182	4932.93	493.29
OASIS B	60×72×60×182	673.71	67.37
OASIS C	121×145×121×182	4817.41	481.74
OASIS D	60×72×60×182	669.51	66.95

Table 2 above presents performance metrics for the different data types in terms of their shape, elapsed time, and average time per run. The fully-sampled OASIS A and OASIS C data types which contain the original data sizes have the longest elapsed time. The sub-sampled OASIS B and D data types which contain the regressed data have a significantly shorter elapsed time when compared to others and achieve the shortest elapsed times overall. During k-fold cross-validation the algorithm is run for $l = 10$ times. Due to this, the average time per run is approximately the total elapsed time divided by 10, providing an estimate of the computational efficiency for each data type.

5 Discussion

As shown in Figure 2, all data types show very similar results. Therefore it could be argued, in the case of the OASIS dataset, that sub-sampling does not impact the accuracy of classifying the condition of MCI. Furthermore, regressing the effects of age and eTIV show a slight difference in classification error for rs between 1 to 10, but for higher r values it reaches the same accuracies as the original datasets. One can argue that when the age and eTIV information is regressed, the MCI-related changes that occur in the brain can be better understood. Since normal ageing and a lower eTIV as a result of normal ageing factors are regressed, only MCI and Alzheimer-related pathological changes can be addressed. On the other hand, removing some age-related or eTIV-related variance which might be relevant to the disease progression could potentially exclude some disease-specific information. The distributions of age and eTIV, do not show a clear separation between the two groups. Therefore, the age and eTIV information do not provide much variance to affect the results drastically.

When reviewing Figures 3 and 4, it is seen that the increasing number of principal components only benefits the FPs and TNs. On the other hand, the FNs and TPs are showing worse results after $r = 10$. With Alzheimer's, early detection is extremely important for timely intervention and treatment. In that case, ensuring that as few true cases as possible are missed, the FNs need to be minimized. This perspective should suggest that the r^* is 5. Figure 2 shows that the test accuracy does not improve much after r^*. However, FP's rate mustn't be too high, which presents a trade-off due to its rising behaviour. It can lead to unneeded stress for patients, additional medical tests, and potential over-treatment.

For Table 2, the computational time for OASIS A and OASIS C are the highest due to their sizes. The column vectors x_i, explained in section 3.2, for OASIS A has a size n which is approximately 7 times higher than those of OASIS B and D. This caused the difference in computation time which is roughly 7 times, as can be seen in Table 2. The small time difference for OASIS A and C could have been attributed to an inconsistent computational environment.

Lastly, the OASIS dataset could be called unbalanced because it contains 60 possible Alzheimer's patients having MCI and 122 controls. This presents a shortcoming for the PCA element in the algorithm. PCA aims to preserve the variance in the data along the principal components, but it does not consider class labels during this process. This imbalance can cause the principal components to capture variance that is only relevant for the bigger class which is the controls.

6 Conclusion

This paper showed how to effectively reduce the dimensionality of the high-dimensional MRI data through PCA and LDA for Alzheimer's classification. The results presented no significant change in classifying a sub-sampled dataset, which makes it very attractive due to its lower computation time. Results for datasets where the effects of age and eTIV are regressed, also show no difference, which is caused because of the relatively elderly cohort group in the dataset.

For future work, it could be relevant to work with a more balanced dataset. Using a balanced dataset could introduce a PCA dimensionality reduction that preserves the variance in the data better for both classes and therefore potentially reduces the number of FPs and FNs. Additionally, in this paper, there were no other classification methods and algorithms discussed that could serve as a baseline or perform better than the PCA-LDA algorithm.

References

1. M. W. A. Caan, K. A. Vermeer, L. J. van Vliet, C. B. L. M. Majoie, B. D. Peters, G. J. den Heeten, and F. M. Vos, "Shaving diffusion tensor images in discriminant analysis: A study into schizophrenia," *Medical Image Analysis*, vol. 10, pp. 841–849, 2006.
2. K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 1990.
3. Z. Breijyeh and R. Karaman, "Comprehensive review on alzheimer's disease: Causes and treatment," *Molecules*, vol. 25, no. 24, p. 5789, 2020.
4. T. Adel, T. Cohen, M. Caan, and M. Welling, "3d scattering transforms for disease classification in neuroimaging," *NeuroImage: Clinical*, vol. 14, pp. 506–517, 2017.
5. ScienceDirect, "Clinical dementia rating (cdr)." Accessed: 2024-06-02, Link.
6. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
7. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* Wiley, 2nd ed., 2000.
8. Joris van de Weg and Burcu Erarslanoglu, "Github repository," 2024. Link.