Deep Learning (DL) methods to diagnose Alzheimer's Disease (AD) using Imaging and Demographics

Dinelka Nanayakkara

May 9 2025

1 Abstract

The primary objective of this project was to predict the incidence of baseline (BL) Alzheimer's Disease (AD) using data from the Alzheimer's Disease Neuroimaging Intiative (ADNI). For prediction, 3D Convolutional Neural Networks (CNN) were used after conducting comprehensive hyperparameter tuning. Using imaging (LogJacobian & tabular) and demographic information, the optimal parameters found achieved a recall of 0.61 for the AD class and an overall weighted F1-score of 0.48. These numbers do not seem too promising, but there is greater scope for fine-tuning these models which might show greater increase in model accuracy.

2 Introduction

Alzheimer's Disease (AD) - one that is neurodegenerative, is the most common cause of dementia. Dementia is currently the seventh leading cause of death, and one of the major causes of disability [1].

Given the critical importance of AD to global public health, novel deeper technologies were used and assessed to see if we could more accurately predict disease status at Baseline. The current literature shows how many other research groups have used Deep Learning for AD prediction, and have achieved high values of accuracy. [3, 5]. Here, we attempt to use multi-modal information combining imaging, tabular imaging, and demographics to look how accurately we could predict BL AD in a given individual.

3 Methods

The dataset that was used for this analysis is part of the Alzheimer's Disease Neuroimaging Intiative (ADNI). MRI features (LogJacobian & tabular) and demographics data were used predict Alzheimer's disease status. Demographics information included as predictors were age, sex, marital status, education (number of years), ethnicity, race, and APOE4 gene count.

Prior to performing any analysis, the first step that was considered was preprocessing and Exploratory Data Analysis (EDA). The different data modalities were combined such that all modalities had the same number of samples. All tabular data (not LogJacobian imaging) was standardized to have mean 0 and unit variance. Only covariates that had more than 96.5% of complete data was

retained for the analysis. Similarly, only samples that had more than 96.5% of complete data was retained for further analysis.

The primary outcome/response variable of this analysis was baseline AD status. The data comprised of 4 categories: AD, control (CN), LMCI (Late Mild Cognitive Impairement), and EMCI (Early Mild Cognitive Impairement). The categories of LMCI and EMCI were combined together to have one category for MCI, thus resulting in 3 outcome categories.

Once all modalities were pre-processed, then 3D Convolution Neural Network (3D-CNN) was used to predict AD at baseline [6]. Although I was initially planning to use a Multimodal Perceptron (MLP) architecture and some pre-trained models as well, this was not possible due to time constraints. The NN architecture that was used here was fairly simple. Given that we did not use pre-trained models, no changes were required to be done to the dimensionality of the imaging data (this is sometimes required to make it possible to use pre-trained models). The NN architecture that was used here comprised of CNN branch to process 3D brain images, and also a fully connected (FC) branch to integrate tabular information (tabular imaging & demographics).

The CNN branch of the architecture comprised of 2 3D convolution layers. The first layer consists of 4 filters with a kernel size of 3 and a stride of 2, followed by a ReLU activation function and a 3D max-pooling layer with a kernel size of 2. The second layer consists of 8 filters with a kernel size of 3, again followed by a ReLU activation function. An adaptive average pooling layer is used to output an 8-dimensional feature vector. This last step was done to reduce the spatial dimension that was brought by the inputted brain images.

This 8-d vector can be interpreted as the features extracted from the inputted brain images. Next, this 8-d feature vector is concatenated with our tabular features (dimension d, which will be introduced in the results section). This results in a combined feature vector size of $8 + d$. This high dimensional feature vector is passed through a fully connected (FC) network that comprises of two linear layers: one hidden layer outputting 32 neurons, followed by ReLU activation, and an output layer with the number of neurons equal to the number of AD classes. The architecture described here is visualized in Figure 1.

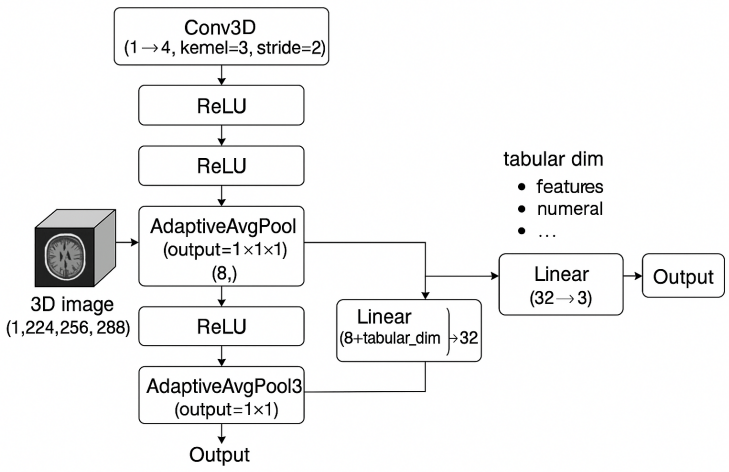

Figure 1: The complete neural network architecture used in this analysis

One important issue that was addressed in the workflow conducted here was class imbalance. One instance within the training process where this issue was addressed was during the choice of the loss function. The loss function used here was a Focal Loss function [4]. Within this loss function, there are 2 parameters that can be tuned: γ and weights α. Hyperparameter tuning within pytorch could be flexibly done via the Optuna framework. [2]. 10 independent trials were conducted to find the optimal set of parameters, and within each trial the

model was trained using 10 epochs. The weights for the 3 classes were expected to sum to 1 (although not required for the focal loss function, and I assumed that γ could take on values between 0 and 2 (again, no certain reason as to why this was picked). During training, the optimal set of parameters were picked using a multi-objective function trying to maximize both the (i) recall for the AD class, and (ii) overall weighted F1 score [7]. The optuna module outputs a figure (namely the Pareto front) which helps identify the best trial. Parallel computing on the longleaf cluster was used to make the parameter tuning process efficient.

As much as visualizing results is ideal and helpful, I could not produce these due to time constraints. The train/test split in this analysis was 70/30. The training data was further split to 80/20 between training (56% of the entire data) and validation data (44% of the entire data).

4 Results

Using the methods laid out in the preceding section, the results are laid out here. The ADNI dataset comes from a large study that longitudinally observes subjects over a span of 30 months (ADNI GO). However, this analysis was done only cross-sectionally and not looking at the data longitudinally. Different modalities of the ADNI data comprised of different sample sizes. A summary of the sample sizes are provided in Table 1.

Modality	**Sample size**
Demographics	2430
Tabular Imaging	1336
Imaging (LogJacobian)	1268

Table 1: Sample size specific to each modality

The demographics data comprised of common factors. One factor known in the

literature is the relationship between the presence of the APOE4 gene and AD. A simple visualization of this is presented in Figure 2 (ignore the NA label for diagnosis) to get an idea of what this might look like.

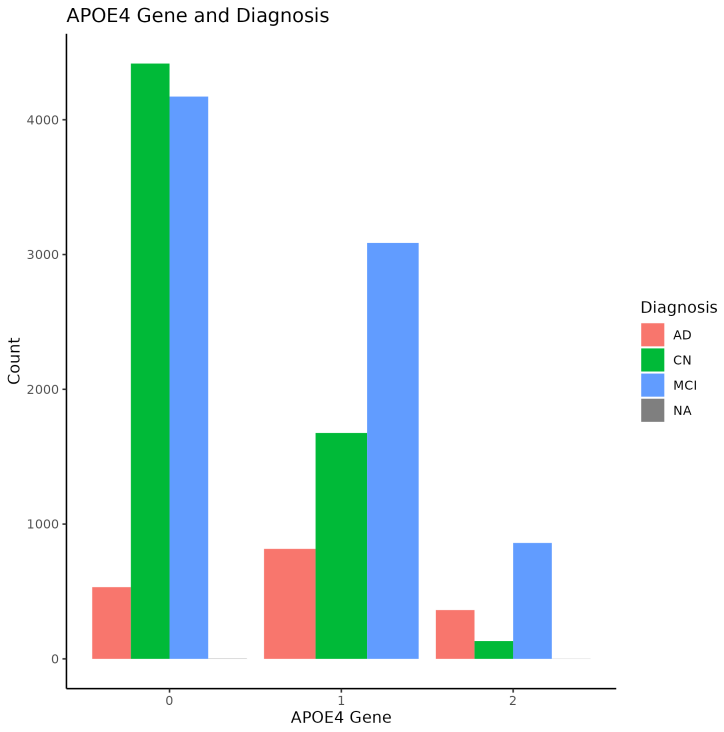

Figure 2: Relationship between the number of APOE4 genotypes and AD presence

The tabular imaging data comprised of 3 datasets: global, Region of Interest (ROI) thickness, and ROI volume. The global file under the tabular features comprised of 6 variables: Pearson Correlation, Brain Volume (BVol), WVol (white area volume), GVol (grey area volume), sum of the brain thickness, and the mean cortex thickness. The global dataset can be viewed as sample specific summary statistics of the brain. In addition to this, we also have information on a deeper level of the brain. We have such information for brain thickness, and brain cortex volume (in voxels). Brain thickness data comprised of 102

columns corresponding to the thickness at a point/location in the brain, whereas the brain volume comprised of 101 columns corresponding to the volume at a point/location in the brain. Brain volume data was missing the column with column name 4 (probably corresponding to the 4^{th} data point in the brain. There were columns which had missingness $> 96.5\%$ and were removed. From the 203 non-global columns, 31 columns were removed due to missingness. Samples that had missingness $> 96.5\%$ were removed from the analysis.

Log-Jacobian data was available for 1268 samples. Each file corresponding to a sample comprised of a numpy array of dimension 216 x 256 x 291. These were converted to Pytorch tensors prior to passing it through to the 3D CNN model. No file was observed with all zero cells, nor were these sparse data files. This confirmed that we did not have any missingess in our imaging data. Post removal of samples due to missingess and to contain samples that only had imaging data, we ended with 764 that were to be used in the analysis.

Parameter tuning was conducted using Optuna due to the class imbalance that was seen in the data. Our 764 samples were distributed as follows:

Class	**Sample size**
AD	91
MCI	373
CN	300

Table 2: AD class distribution to identify imbalance

This class imbalance suggests that we should overweight the AD class to account for lower proportion in the AD class. The results from the hyperparameter tuning from Optuna can be visualized using the Pareto front presented in Figure 3.

We can see from Figure 3, that the best trial is the point that has the darkest red shade. The combination of recall (AD class) and the weighted overall F1-

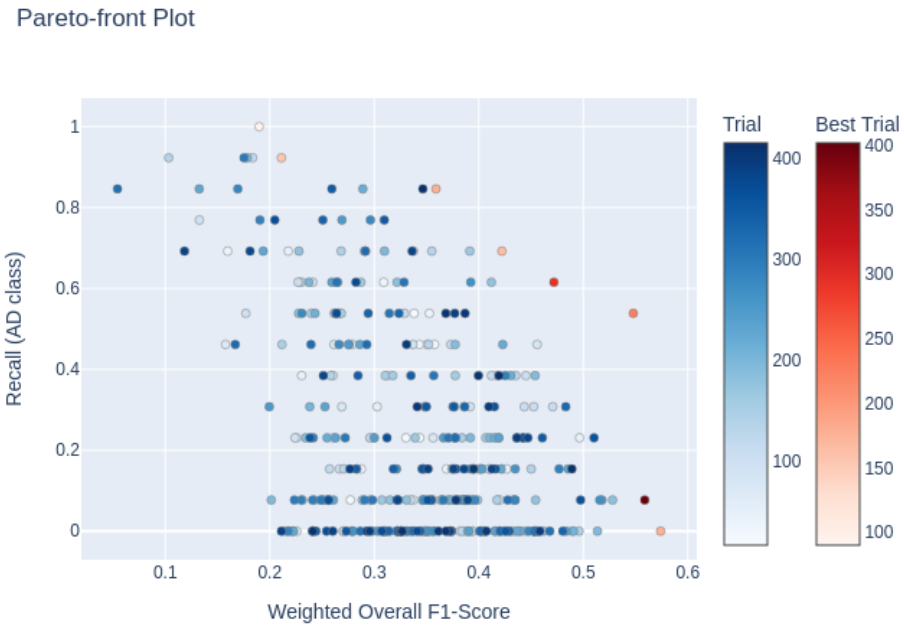

Figure 3: Pareto front after hyperparameter tuning

score for this trial was 0.61 and 0.48 respectively. The combination of γ and α weights for this trial were $\gamma = 1.024$ and $\alpha = (0.27, 0.10, 0.63)$ respectively on the validation set.

Due to time constraints, I could not test this model on the test data (30%).

5 Discussion

A 3D CNN architecture was modified and used to predict BL AD using tabular imaging, LogJacobian imaging, and demographic data. After tuning our focal loss function using Optuna, we obtained a recall of 61% for the AD class and also an overall weighted F1 score of 48%. These numbers do not look too promising,

but comprehensive model tuning was not conducted due to time constraints. This model was not running this well prior to accounting for class imbalance. Prior to using a focal loss function, I used cross-entropy instead. Cross-entropy did not yield results as good as what is presented here. Moreover, the NN architecture itself could be improved by adding more layers, neurons, and fine tuning the stride and kernel size hyperparameters.

In addition to a 3D CNN, further work could comprise of using a Multi-Layer Perceptron architecture, a fusion model, and transformers to address this problem. The availability of more time on this analysis would have provided space to using these methods in this analysis itself. If this current work seems interesting and if you believe we could take it forward with more time in hand, it would be a great privilege for me to be a part of it and contribute.

References

[1] 2023 alzheimer's disease facts and figures. *The Journal of the Alzheimer's Association*, 19:1598–1695, 2023.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[3] Wolk DA Fan Y Li H, Habes M. A deep learning model for early prediction of alzheimer's disease dementia based on hippocampal magnetic resonance imaging data. *Alzheimers Dement.*, 15, 2019.

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[5] Masurkar A.V. Rusinek H. et al. Liu, S. Generalizable deep learning model for early alzheimer's disease detection from structural mris. *Sci Rep*, 12, 2022.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.