Alzheimer's disease diagnosis with MRI

Ping-Hsuan Hsieh
Department of Computer Science
National TsingHua University
Hsinchu, Taiwan
hanson109000109@gapp.nthu.edu.tw

Chia-Yi Lin
Department of Computer Science
National TsingHua University
Hsinchu, Taiwan
rogerline0527@gmail.com

Hsin-Han Chang
Department of Computer Science
National TsingHua University
Hsinchu, Taiwan
changhsinhan319@gmail.com

Shao-Tang Sung
Department of Computer Science
National TsingHua University
Hsinchu, Taiwan
7sunny78910@gmail.com

Chi-Fang Chou
Department of Computer Science
National TsingHua University
Hsinchu, Taiwan
chifangsunny@gmail.com

Hsi-Chen Wang
Department of Management and Technology
National TsingHua University
Hsinchu, Taiwan
jessicawang91920@gmail.com

***Abstract*—We have developed a project focused on diagnosing Alzheimer's disease using MRI scans. This project encompasses two main aspects. Firstly, we aim to determine the stages of Alzheimer's with the highest possible accuracy through MRI images. Secondly, we strive to predict the stages of Alzheimer's as early as possible, utilizing both images and other relevant features. We have achieved these objectives by employing various machine learning models that we developed in-house. Our methods do have improved performance outcomes in this field.**

I. Introduction

Alzheimer's disease, also known as AD, is a progressive neurodegenerative disorder. Symptoms like severe memory loss and cognitive impairments highlight the importance of early diagnosis, particularly in today's aging society.

MRI is a promising tool, and its diagnostic accuracy can be further enhanced through machine learning. This is why we are focusing on developing an MRI-based model to diagnose Alzheimer's disease, exploring techniques for improved early detection.

In addition to accurately examining the stages of AD through MRI, we have introduced some innovative ideas. We propose that predicting the disease's progression by incorporating specific features could significantly aid in its diagnosis, thereby benefiting patients. Therefore, our primary goal has shifted to determining whether MRI can stage Alzheimer's disease more accurately and predict its progression. This has led us to a dual strategy: classifying patient groups and predicting disease progression.

II. Method

A. Data Collection and Preprocessing

In this project, we opted to utilize two datasets for our model training, which we refer to as Data A and Data B, for simplicity.

1) Data A:

Data A, sourced from our mentor, comprises a collection of 3184 MRI images preprocessed using MPR, gradwarp, and B1 correction. The image dimensions are (96, 96, 96), which is a moderate size suitable for our model training. However, midway through our project, we decided to incorporate diagnostic conversion information as a feature, and Data A lacked this data. Consequently, we sought another dataset to fulfill our objective.

Despite this, Data A has proven effective as input for our model. Firstly, the quantity of 3184 images is sufficient. Additionally, the image quality is commendable, and the consistent preprocessing method enhances the uniformity of our input data.

To augment our model, we incorporated additional features to enrich the dataset. We integrated information obtained from the same data source, including the diagnoses and symptom checklists completed by patients prior to their MRI exams. Features such as experiencing coughing, vomiting, or dizziness on the exam day were added as supplementary inputs to our model. To merge these features with our original Data A, we cross-referenced patient IDs and examination dates from both data collections. If a match was found, we included the corresponding data in our dataset.

2) Data B:

We obtained Data B from the Image & Data Archive (IDA) managed by the Laboratory of Neuro Imaging (LONI) at the USC Mark and Mary Stevens Neuroimaging and Informatics Institute. In this dataset, our primary focus was on the diagnostic conversion of patients. The selected data originated from ADNI2 and ADNI GO (ADNI stands for Alzheimer's Disease Neuroimaging Initiative), featuring a DXCHANGE attribute with nine

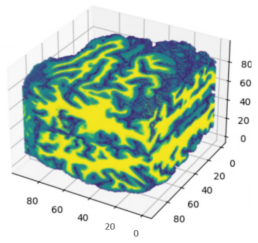

Fig. 1. Preprocessed MRI image from Data A.

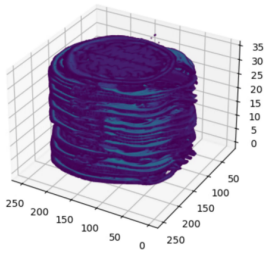

Fig. 2. Preprocessed MRI image from Data B.

different stages of conversion, such as stable to MCI, MCI to AD, and AD back to MCI, etc.

Handling Data B posed several challenges. Initially, downloading the data from the website consumed almost an entire weekend due to its vast size and the presence of a considerable amount of unwanted information. Consequently, we had to filter and select the relevant data, using a process similar to that applied for Data A. This resulted in only 2705 images deemed suitable for our model training.

Another challenge emerged from the dataset's original format, which was DICOM (.dcm), whereas we intended to use images in Nifti (.nii) format for our model. Thus, we conducted the conversion of all images to meet our model's requirements.

Despite its utility, Data B presents some drawbacks. The dataset size of 2705 images falls short of our ideal target, and the variance in the preprocessing methods applied to the images introduces considerable variability. Consequently, we need to carefully account for these factors in our model design.

B. Model Creation and Evaluation

Based on data A and data B, we have created two distinct types of models to fit each dataset respectively. These are referred to as model A and model B for simplicity in the discussion that follows.

1) Model A:

For model A, our objective is to use data A to develop a model that can predict the patient's condition from an MRI image. The possible states are divided into three categories: CN (Cognitively Normal), MCI (Mild Cognitive Impairment), and AD (Alzheimer's Disease).

a) Basic 3D CNN:

For this model, we are attempting to build a standard 3D CNN model to predict the states. Initially, we utilize a two-layer model. Each layer comprises a convolution layer followed by a pooling layer, with ReLU as the activation function. After these two layers, we apply flatten and dense functions to attempt to generate the probabilities. We employ softmax as the activation function for this stage. Additionally, Adam is used as the optimizer, and categorical crossentropy is selected as the loss function.

Ultimately, we achieved an accuracy of nearly 58%, but we encountered some unexpected results. We have investigated these anomalies and detailed our findings below.

Initially, when we built and tested our model for the first time, the accuracy surprisingly exceeded 80%. However, we discovered that this high accuracy was due to an oversight in our data handling: we hadn't separated the data from different patients properly. As a result, some of the data was being used both in training and testing simultaneously. This overlap falsely inflated our model's performance. After we corrected the data division method, the accuracy significantly decreased.

Secondly, we encountered a limitation with our RAM capacity, which was insufficient to store the large dataset. This was primarily due to the substantial size of each MRI, which is up to 96 * 96 * 96 in dimensions. The limited amount of data that our system could handle led to easy overfitting of our model. To address these challenges and enhance accuracy, we realized the need to explore alternative solutions. Consequently, this led us to develop the next model.

b) Mix 3D CNN:

In this model, our primary objectives are to increase accuracy and address the challenge posed by the large size of MRI data.

The structure of the 3D CNN model remains the same as the basic one. However, this time we divided the data into two parts. Our approach involved building two separate models to fit each part and then combining the probabilities generated by these two models. By dividing the data initially, we could delete the part 1 data when building the model with part 2 data, thus ensuring that our RAM was sufficient. Initially, we anticipated

that combining the probabilities at the beginning would significantly enhance performance. Unfortunately, the accuracy only saw a modest increase, reaching nearly 61%.

c) ResNet50

ResNet50 is a convolutional neural network with a depth of 50 layers. It uses "residual blocks" to solve the problem of gradient vanishing in deep networks, where these blocks enable the direct backward propagation of gradients to earlier layers through skip connections. ResNet50 is commonly used for image recognition tasks and demonstrates high efficiency in performance. It is also frequently used as a pre-trained model for transfer learning. However, in our tests, ResNet50 did not perform well, possibly due to the unsuitability of MRI data types or the model being too small in size. Consequently, we did not adopt the architecture of ResNet50.

d) EnhancedFeatureNet

"EnhancedFeatureNet" is designed to extract deeper features by repeatedly performing convolution. This approach allows for a more intricate and in-depth analysis of the input data, which can potentially enhance the model's performance in specific tasks. In our observations, EnhancedFeatureNet indeed demonstrated superior performance. Consequently, we based our final model, model A, on this approach. Here is the structure of our final modelA:

- **Input Convolution Layer**: Initially, the data passes through two convolutional layers. The first layer expands single-channel input to 256 channels, and the second layer reduces these 256 channels to 64. Each layer is followed by batch normalization to accelerate training and stabilize the learning process, and ends with an ELU (Exponential Linear Unit) activation function.
- **Block-A**: This is a convolutional block that includes a convolutional layer, batch normalization, ELU activation, and max pooling. The purpose of this block is to further extract features and reduce data dimensions.
- **Repeated Block-A**: Block-A is repeated five times, meaning the data undergoes the same feature extraction and dimensionality reduction process five times. This helps the model learn deeper feature representations.
- **Calculating flattened size**: After passing through the repeated Block-A, the dimensions of the data are reduced. In order to feed the convoluted feature map into the dense layers, it's necessary to convert it into a one-dimensional vector. This requires calculating the flattened size of the feature map.
- **Block-B**: A series of dense layers used for final classification of the convolution-extracted features. These layers gradually reduce the feature dimensions from the flattened size to 50, then to 10, and finally to 3. The ReLU activation function is used in the dense layers, with a Dropout layer before the final layer to reduce overfitting. Lastly, the Softmax function is used for multi-class classification.

The model ultimately achieved an **accuracy of 0.631** and an **F1 score of 0.553** in three-category classification tests.

2) Model B:

For model B, our aim is to use data B to develop a model that, given an MRI image and certain features, can predict whether the patient's condition will worsen.

To achieve this goal, we experimented with two different models for Model B, which include

a) Hybrid 3D CNN:

We implemented our initial concept for this type of model.

Because our goal was to use MRI images and additional features to predict the probability of disease progression, our initial approach involved combining two separate models. The first model, a CNN, was dedicated to processing the MRI images, while the second, a traditional Neural Network, focused on analyzing the additional features. After creating these two models, we concatenated them and added several Neural Network layers to the combined model. This enabled us to generate the final output.

This approach was both intuitive and easy to implement; however, we encountered significant performance issues. The model achieved only about 50% accuracy across training, validation, and testing phases. Intriguingly, the accuracy appeared to be unrelated to the hyperparameters. Instead, it correlated with how we partitioned the data into training, validation, and testing sets. We experimented with different random seeds for dividing the data and observed a consistent pattern: if the seed remained the same, the final accuracy across all datasets stayed constant regardless of any adjustments to our hyperparameters.

Additionally, we observed another peculiar phenomenon: although the loss decreased with each iteration, the accuracy improved significantly only during the first two or three iterations and then plateaued for the remainder of the training process. This pattern is puzzling, and we are still trying to understand the underlying reasons for this occurrence and how to address it.

b) 3D CNN + SVM:

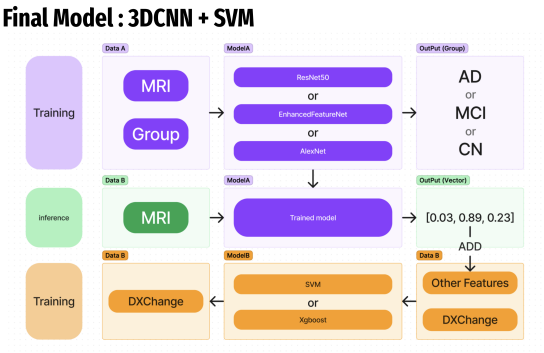

Fig. 3. Structure of 3DCNN + SVM

By using DataA, we successfully trained ModelA, which demonstrated excellent capabilities in distinguishing Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Normal Control (CN). In order to further develop a model capable of predicting disease progression, we plan to utilize DataB, which contains temporal information. Our strategy involves first using the pre-trained ModelA to generate feature vectors from the MRI files in DataB. Subsequently, we will combine these feature vectors extracted from the MRI files with other features present in DataB to train a Support Vector Machine (SVM) model. We expect that by integrating the unique features from DataB, we can further enhance the model's performance.

Finally, when combining the MRI file vectors with features such as DXMDES, Group, SITEID, Sex, Age, and others, the predictive performance for disease progression is as follows:

* **Accuracy**: Around 80%.
* **F1 score**: Around 0.74.

III. Issues and Conclusion

A. Conclusion

We discovered that having critical features is more effective in predicting the probability of aggravation.

B. Patient-based and Image-based

In our approach, we conducted a comparative analysis between patient-based and image-based model. The patient-based model considers diverse view of disease progression. In contrast, the image-based model focuses solely on MRI image data. Using image-based model may not get the true performance because one subject data may be allocated to two set, meaning that it would validate itself, leading to accuracy drop.

C. Overfit

After training a significant number of models, we observed that all of them inevitably tended to overfit eventually. We believe this issue arises because our models are too 'small' or not sufficiently complex to effectively handle the training of 3D images.

We attempted to mitigate this overfitting issue by incorporating a 'dropout' layer into our models. However, the effectiveness of this approach appeared to be minimal.

D. Hardware Limitation

Due to the substantial RAM consumption by 3D graphs during both I/O processes and training, we were unable to use all the data simultaneously for training our models. Consequently, we had to select only a portion of the data for this purpose.

E. Final Performance for Model A

- **Accuracy**: Around 63.1%.
- **F1 score**: 0.553

F. Final Performance for Model B

- **Accuracy**: Around 80%.
- **F1 score**: Around 0.74.

G. Future work

- **Upgrade Hardware**
  Given that hardware limitations have been a bottleneck in both data I/O and model training processes, we are considering two potential solutions for the future: either upgrading our hardware or subscribing to Google Colab Pro+.

- **Enlarge Model**
  Because 3D graphs contain a vast amount of information, a larger model is necessary to utilize this data more effectively. We hope that after upgrading our hardware, we will be able to create and train a more substantial model in our next endeavor.

- **Increase Clean Data Amount**
  In Data B, there is a wide variety of MRI images. Some of these images have undergone skull removal, while others have not. Going forward, our aim is to gather more data pertaining to a single type of MRI image. This approach will enable us to train Model B with more uniform images, potentially enhancing the model's effectiveness.

IV. Data and Code Availability

Here is the GitHub link to our project, where we have provided all of our source codes and detailed information about our data: https://github.com/MR600hans/ML_3DCNN

V. Author Contribution Statements

P.H(20%): Team leading, Model A (ResNet50, Enhanced-FeatureNet), Model B (CNN + SVM), report writing.

C.Y(17%): study design, build model B (Hybrid), writing

H.H(13.5%): study design, assisting build model, writing

S.T(18%): study design, build model A, discover the problem of model A, writing.

C.F(16.5%): study design, data collection, data preprocessing, final presentation, writing.

H.C(15%): study design, data collection, data preprocessing, writing

References

[1] Hasib Zunair. "3D image classification from CT scans." keras.io. Accessed: Sep. 23, 2020. [Online.] Available: https://keras.io/examples/vision/3D_image_classification/

[2] LIN Weiming, GAO Qinquan, DU Min. "Convolutional neural network based method for diagnosis of Alzheimer's disease." joca.cn. Accessed: Dec., 2017. [Online.] Available: http://www.xml-data.org/JSJYY/2017-12-3504.htm

[3] Michael W. Weiner. "Recent publications from the Alzheimer's Disease Neuroimaging Initiative: Reviewing progress toward improved AD clinical trials." ncbi.nlm.nih.gov. Accessed: Oct. 29, 2019. [Online.] Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6818723/

[4] Karteek Popuri. "Using machine learning to quantify structural MRI neurodegeneration patterns of Alzheimer's disease into dementia score: Independent validation on 8,834 images from ADNI, AIBL, OASIS, and MIRIAD databases." ncbi.nlm.nih.gov. Accessed: Oct. 1, 2020. [Online.] Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7469784/