Classification of Alzheimer's Disease using Patients' MRI and Related Features

Malavika Bindhi (mbindhi), Kevin Chavez (kechavez), Tita Ristanto (ristanto)

I. Introduction

ACCORDING to a World Health Organization survey from 2017, Alzheimer's Disease is a disease that is affecting an estimated 47 million people worldwide and there are nearly 10 million new cases every year [3]. Clinical diagnosis of Alzheimer's Disease is challenging especially in its early stages. With the aid of classification tools, we are aiming at improving diagnosis efforts. This paper explores some of the methods that can be used to classify patients with Alzheimer's Disease based on MRI and demographic features.

Our study focussed on using MRI biomarkers, demographic data, and cognitive tests gathered from subjects to train Machine Learning models and classify patients as either having AD or not. MRI biomarkers were largely obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [1] database while some MRI biomarkers were processed using the FreeSurfer [4] software suite. The basic pipeline of the study was to first obtain demographic, cognitive test, and MRI image data from as many patients as possible in the database. Next, if the MRI image had not been preprocessed for MRI biomarkers, we processed it using FreeSurfer and gathered volumetric data. Once the data was tabulated and the models trained, we used a separate test set as input to our algorithm. We used Deep Neural Network, SVM, and Decision Tree (and Ensemble Method) models to output a predicted label of AD or Normal (Not AD).

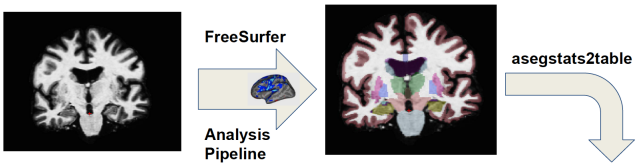
II. Related Work

Prior work in the area of Alzheimer Disease classification suggests that several brain volumetric features provide insight to AD classification [7] [8] . MRI Image segmentation provides a useful biomarker especially when looking at volumetric features in regions like the hippocampus. More recently, AD classification has been attempted with machine learning techniques although this is typically done by manual inspection in a clinical setting as in [8] and [9]. In [6] and [10], the common struggle of classifying in large feature spaces (known as the curse of dimensionality) for AD biomarkers is addressed. The two methods propose using either PCA [6] or forgoing dimensionality reduction in favor of regular logistic regression and SVM models [10] to perform AD classification.

Depending on what AD labels a study looks at, the AD classification task can be very difficult. [7] classified AD against Normal Control (NC) to obtain test accuracies of 87% and 85% using SVM and other classifiers. In the same study, only 78.22% and 72.23% test accuracies were obtained when classifying between three different labels — Mild Cognitive Impairment (MCI), AD, and NC. Another study, [11], classified between Alzheimers, Mild Alzheimer's and Huntington's disease comparing perofmances of Fuzzy Neural Networks (FNNs), ANNs, and SVM models. The accuracies for each model, respectively, were 95.5%, 92.8%, and 89.8%. Our work focuses purely on distinguishing AD ("Dementia" using our terminology) against less severe to normal patients which were labeled together as normal ("NL").

III. Dataset and Features

Alzheimer's Disease Neuroimaging Initiative (ADNI) provide datasets that can be used for various Alzheimer's Disease related studies[1]. The dataset provided contain a set of data that was generated specifically for the TADPOLE Grand Challenge [13]. This particular dataset was used in this project. The dataset consists of various features that can be classified as: demographic, cognitive tests, and biomarkers obtained from MRI and PET scans. The biomarkers are numerical values of the brain volumes identified using the FreeSurfer software. Figure 1 shows a sample data point extracted from the FreeSurfer Analysis Pipeline.



Measure:volume	Left-Lateral-Ventricle	Left-Thalamus-Proper	Left-Putamen	Left-Hippocampus
ADNI_429	17055	7176.8	5730.8	3767.5
Measure:volume	**Left-Lateral-Ventricle**	**Left-Thalamus-Proper**	**Left-Putamen**	**Left-Hippocampus**
ADNI_429	1219.7	691.5	330.7	5372

Fig. 1. Using Freesurfer Processing as part of our pipeline to extract volumetric biomarkers. First a .nii raw format MRI file is processed with recon-all command for each subject. A subcortical segmentation of the brain then labels different volumes regions such as the right-hippocampal region. Lastly, the data is tabulated with the asegstats2table.

The dataset consists of about 1900 features and we use only 245 of these features for training the model. To be specific, we use all the demographic features, three of the cognitive features, and all of the features obtained from UC Berkeley's AV45 analysis (MRI biomarkers). The data is labeled as normal (NL), mild cognitive impairment (MCI), and dementia. Table 1 shows the breakdown of the data used from the ADNI database.

A. Preprocessing of Data

Due to the amount of missing feature values in the original dataset, a preprocessing of the data was required. Firstly,

TABLE I
DATASET DESCRIPTION

Dataset Label	Dementia	MCI	NL	Total Sample Size
Completed Data	1454	3298	2140	7359
After processing	104	324	254	682
MCI to NL	104	0	578	682
MCI to Dementia	428	0	254	682

any sample that has missing values is removed. Also, the features that have text labels are converted to numerical labels. After removing these missing values, there are some very few (less than 10) sample points with additional labels. These are the cases where the labeling was changed in the dataset, for example, from "MCI to Dementia" or "MCI to NL". In such cases, the labels were converted to the correct label i.e. in case of "MCI to Dementia", the sample was relabeled as Dementia. Table 1 also shows the sample sizes for each class after this preprocessing. Since these are very few cases, this change does not reflect severe changes in the dataset. Additionally, since the goal is to classify only Alzheimer's Disease, the labels of MCI and NL are treated as non-demented or normal, thus deriving a dataset with only two classes. Another option, was to classify a patient as cognitively impaired or not. In this case, the labels of MCI and Dementia were treated as a single class while NL was treated as another class. Both of these classification were attempted but the major goal was to classify Alzheimer's Disease.

245 features with 682 sample points cause high variance. Hence, two compression methods were attempted, namely PCA and LDA. Both of these methods were used to compress the number of features to 10. PCA identifies the principal components that has maximum variance and does not depend on the labels. LDA tries to find features that maximize the separation between different classes and hence is label dependent. Both of these were implemented using the sklearn python library [12].

B. Data Split

For training different models and determining the best model, the dataset was split into a train set (80% of the samples) and a test set (20% of the sample). We applied k-fold cross-validation on this training set, with $k = 5$, to obtain new train and dev sets. The new train set and dev set are used in the training of the model while the test set is unused in the training.

IV. METHODS

A. Decision Trees, Random Forest, and AdaBoost

Decision Tree algorithm learns a dataset by making the best split of features in the training set into subsets based on the homogeneity of each new subset. The process goes on until a stopping criterion (e.g. tree size or number of observation per node) is reached. We can control how the decision tree algorithm splits nodes and two popular measures are Gini index and entropy.

$$Gini = 1 - \sum_{i=1}^{C} p_i^2 \text{ and } Entropy = \sum_{i=1}^{C} -p_i log_2(p_i)$$

In this problem, we also used ensemble method called Random Forest. Instead of building a single tree like Decision Tree algorithm, we can build B randomly built independent trees f_b and each of them can give us a prediction. The final prediction $\hat{f}$ is an average of all predictions (for regression) or majority vote (for our case, classification). This can be helpful in some cases, as the last action of making a final prediction can reduce the variance of the model.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

Another method that we tried was AdaBoost. It creates T weak learners h_t (in this case trees) and boosts these weak learners to stronger ones. The way it boosts the weak learners is by increasing the weight of data points whose label is misclassified. We used exponential loss function to update the weights. In the next iteration, the algorithm will try hard to avoid misclassifying these data points as it would greatly increase the penalty. Similar to Random Forest, the final prediction H is computed as a weighted majority vote of the weak learners' prediction.

$$H(y) = sign(\sum_{t=1}^{T} \alpha_t h_t)$$

B. Support Vector Machines

Both linear SVM and a polynomial-kernel SVM models were trained as part of our study. A radial basis function (RBF) kernel was also experimented to an extent, but failed to meet a reasonable criteria and so is not discussed in detail.

SVM Classifiers seek to find a hyperplane separating the labeled data such that the widest margin is created between the labeled data points. One appealing aspect of SVM its ability to construct this hyperplane in high to infinite dimensional spaces. The problems may be formed as a constrained optimization problem given training examples $\{x_1, x_2, ..., x_m\}$ and training labels $\{y_1, y_2, ..., y_m\}$:

$$\min_{\gamma,w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i$$

$$s.t. y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, i = 1, ..., m$$

$$\xi_i \geq 0, i = 1, ..., m$$

where ξ, w, and b are hyperplane parameters and C serves to regularize and control the relative weighting between margin goals.

The above optimization for SVM can also be solved, more generally, for higher order feature mapping using the Kernel trick. In our SVM model we also used a seventh-order ($d = 7$) polynomial kernel to fit more complex set of features involved in classifying for AD:

$$K(x, z) = (x^T z + c)^d$$

Once the SVM model is trained and the hyperparameters are solved for, classification is done by checking what regions (different sides of the margins) are designated for each label. Given a new sample (x_t, y_t), the point will lie in region on one side of the margin and the proper label is chosen.

C. Neural Network

A neural network model was also used to train the data. The network comprised of 2 hidden layers. The first layer consisted of 3 neurons and the second layer consisted of 2 neurons. The sigmoid function was used as the activation function for the hidden layers as well as the output layer. Cross entropy function is used as the loss function. Cross entropy loss is defined as:

$$L(y, \hat{y}) = -yln(\hat{y}) - (1-y)log(1-\hat{y})$$

where y is the true label, $\hat{y}$ is the predicted label. The sigmoid function is defined as:

$$g(z) = \frac{1}{1+e^{-z}}$$

The neural network algorithm finds the output by applying weights to the input features for each example and then finding an output using the activation functions from the previous layer. This continues until the last layer is reached which is the process of forward propagation. Each of the weights acts to identify the importance of the input features of the previous layer. In backpropagation, the weights and biases are updated in each iteration of training by finding the gradient of the loss with respect to the weights and biases. The neural networks converges to identify the weights needed to be applied to each layer to minimize the loss function i.e. correctly classify the input.

For this project, the neural network mentioned above was implemented using the MLPClassifier from the scikit-learn Python library [12]. We used trial and error to identify some of the parameters required for the classifier. An LBFGS solver was used and maximum number of iterations was set to 1000. A learning parameter that scales inversely was initialized to 0.01. Regularization was also attempted using $\alpha = 0.001$.

V. Results and Discussion

If we use only two labels, namely Dementia and Normal/NL, by converting MCI to NL, the algorithms can easily distinguish between the two. The training, dev, and test accuracy ranged from 0.95-1, 0.87-0.94, and 0.88-0.91, respectively. Below is the top 8 combinations of algorithms that we used:

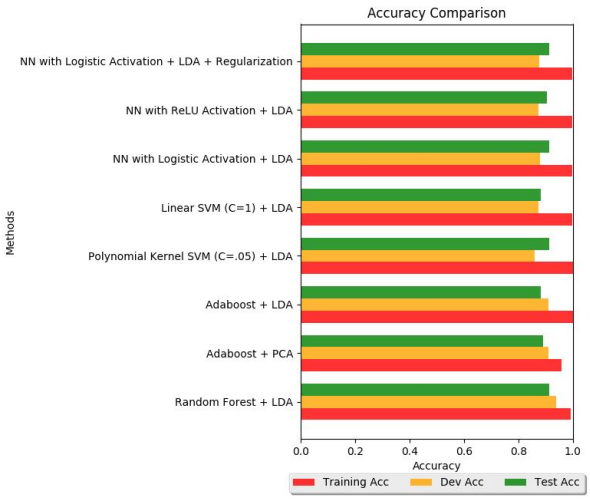


Fig. 2. Accuracy score comparison between classification models

Polynomial Kernel SVM, Random Forest, and Neural Network with Logistic Activation topped the chart with only slight difference. If we converted MCI to Dementia, the test score would be lower at around 0.75. We suspected that in this case, the separation between MCI and NL was not clear so that the algorithm was struggling to find the decision boundary. The degree of cognitive impairment within the label MCI was defined very broadly. A more useful scale would be a numerical one by which the severity of cognitive brain impairment is measured in a fine-grained manner, but this dataset generalized varying degree of impairment by a single MCI label. This might be the cause of the problem.

LDA was proven to be useful in this case in reducing the variance of the model. It reduced the dimension from 245 to 10 features and to some extent increased the test accuracy. PCA also improved the model in this case, although by smaller percentages. There was a large gap between training and test score when PCA or LDA was not included in the model and this was clearly a sign of overfitting.

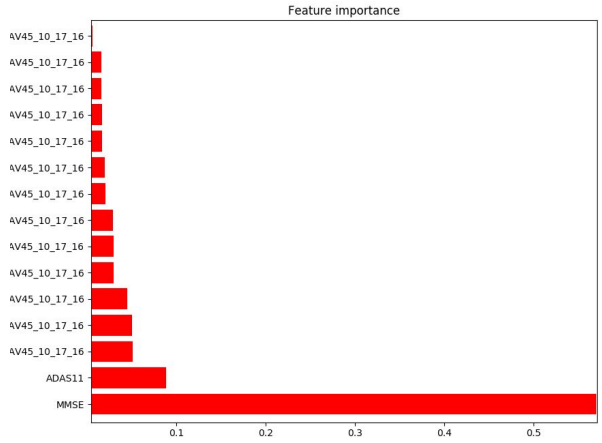


Fig. 3. Feature Importance Score

Feature importance analysis was performed using tree based algorithm and it turns out that two of three cognitive test fea-

tures (MMSE and ADAS11) are two most important features in the model, followed by MRI features. We can also see it from the tree visualization (in decision tree section), where MMSE becomes the primary split. One of our objective was to build a good model with minimal human input. So far, we successfully achieved high score accuracy without test like CDR.

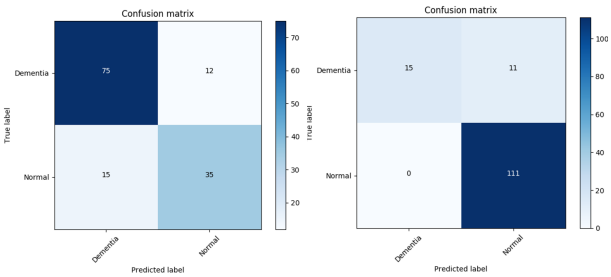


Fig. 5. Confusion Matrix for Random Forest Model. Left: "MCI to Dementia". Right "MCI to NL"

A. Decision Trees, Random Forest, and AdaBoost

A decision tree model was initially generated using Gini criterion as a measure of impurity. Other measures of impurity, such as cross-entropy and misclassification, were also used, but generally had worse performance. Using Decision Tree visualization below, we can intuitively see how the algorithm processes MRI and cognitive features to make a prediction whether one is demented or not. However, the initial/baseline model generally did not perform well with training and test set accuracy of 96% and 70%, respectively. The tree grew excessively large and over-fitted the training data. Limiting the branches and pruning method were done, but ensemble methods were better in terms of test accuracy.

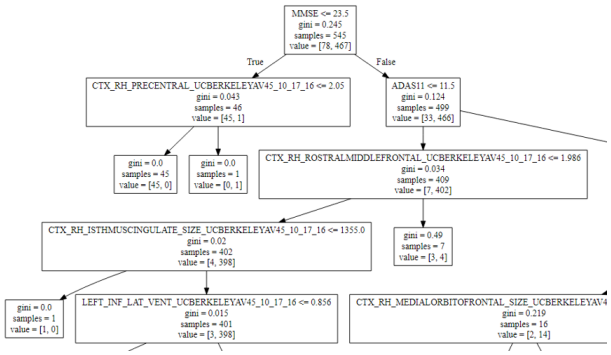


Fig. 4. Decision Tree Visualization. The entire tree is too large to be shown on this report, but this image shows top feature splitting in which we are interested.

Random Forest was able to improve the test accuracy score. We tried several different number of trees and maximum depth of the trees. When we used 80 independent trees and limited maximum depth to 5, the test accuracy went beyond 0.9. Each of those 80 simpler trees with lower bias gave a prediction result and when it took majority vote, the variance decreased and the model was better.

For comparison against the rest of the tree methods, we also used AdaBoost. Intuitively, as a boosting algorithm, AdaBoost would increase the variance of the model as it boosted the weak learners to strong ones. The result varied; in some cases AdaBoost performed better than Decision Tree and in some others, it didnt. In general, Random Forest gave us the best result.

Random Forest came out as the best algorithm among these methods with highest test accuracy in both "MCI to NL" and "MCI to Dementia" cases. In the first case, Random Forest had 0.912 test accuracy compared to 0.81 using improved Decision Tree. In the latter case, Random Forest outperformed other tree based algorithms, achieving 0.8 test accuracy. High accuracy in the first case might partly be an effect of imbalance proportion between two labels, in which there are only 26 "Dementia" cases out of total 136 data points in the training set. We need to feed more "Dementia" labeled data points into our model to increase our confidence with the result. It is more fair to evaluate this case using F1 score shown in table II as it takes into account the proportion of the two labels.

TABLE II
RANDOM FOREST MODEL METRICS

	Accuracy	Recall	Precision	F1 score
MCI to NL	0.91	0.58	1	0.73
MCI to Dementia	0.8	0.86	0.83	0.85

In "MCI to NL" case, Random Forest did a good job in terms of precision. Given that the prediction is Dementia, there is a high chance Random Forest correctly predicts this. But it has a bad recall; about half of the actual Dementia cases were predicted as normal. In "MCI to Dementia" case, all metric scores were performing quite well at around 0.8.

B. Support Vector Machines

For both linear and polynomial kernel SVM classifiers, the C regularization parameter was manually sweeped to find an optimal value. Initially, before more data was gathered, overfitting was a concern especially when using a higher order polynomial kernel. When we upgraded from ≈ 300 data points to 682 data points, we realized that the need for regularization was not as important. Therefore, only two values, $C = 1.0$ and $C = .05$, were reported.

Although SVM does well in high dimensional feature spaces, we find better performance when dimensionality reduction was used. Out of the PCA and LDA approaches that we explored, LDA produced the better results seen in figure 2.

In both the "MCI to NL and "MCI to Dementia cases, the best accuracies and F1 scores were obtained using a Polynomial ($d = 7$) Kernel SVM Classifier with $C = .05$ model and second best results with a Linear SVM Classifier

with $C = 1.0$. For both cases of MCI labeling, the polynomial kernel SVM classifier performed better than linear SVM classifier with. In the first case, polynomial SVM obtained . In the second case, polynomial SVM obtained .76 compared to .71 for linear SVM.

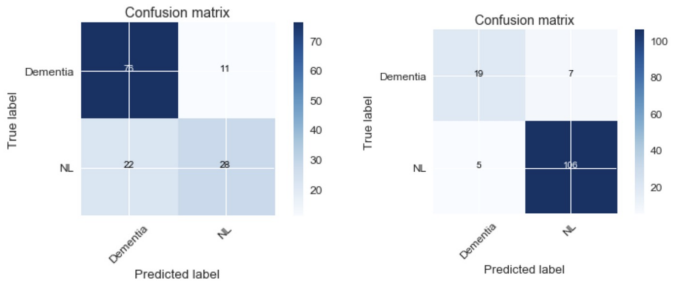


Fig. 6. Confusion Matrix for Polynomial Kernel SVM Model. Left: "MCI to Dementia". Right "MCI to NL"

From Table III, the F1 score for the polynomial Kernel in the case of "MCI to NL " is .76. This is more indicative of how classification performs with when taking into account the fewer labels of Dementia contained in the data set. In the case of "MCI to Dementia" the reverse trend occurred. Here, the polynomial SVM classifier had accuracy and F1 scores at .76 and .82 which is are more comparable.

TABLE III
SVM POLYNOMIAL KERNEL MODEL METRICS

	Accuracy	Recall	Precision	F1 score
MCI to NL	0.91	0.73	0.79	0.76
MCI to Dementia	0.76	0.87	0.78	0.82

C. Neural Network

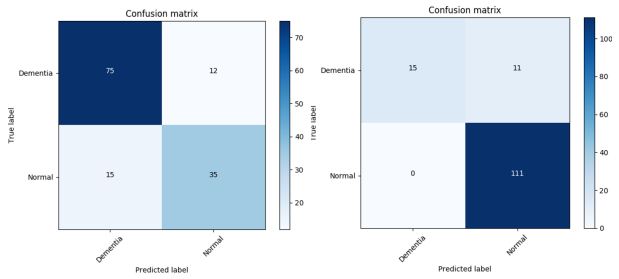


Fig. 7. Confusion Matrix for Neural Network Model. Left: "MCI to NL". Right: "MCI to Dementia"

For the neural network model, an attempt was made to use the logistic function and ReLU function. The learning rate was initialized to 0.01 and was inversely scaled. This was determined after some trial and error with the model. Training was performed using these two activation functions and different network sizes. Also, regularization was attempted. Since we had too many features originally, the model was over-fitting to the training set. Thus we had to reduce the feature dimension. After compressing the features using LDA, the performance of all the models improved and were very similar to each other. PCA did not succeed in obtaining the same result. This was because LDA tries to find features that has maximum separation between classes while PCA does not make use of class labels.

TABLE IV
NN MODEL METRICS

	Accuracy	Recall	Precision	F1 score
MCI to NL	0.91	0.73	0.79	0.76
MCI to Dementia	0.75	0.86	0.81	0.83

The best testing accuracy were obtained using the logistic function without a regularization. This model had 2 hidden layers with 3 neurons in the first layer and 2 neurons in the second layer. All the different neural network models tend to perform better when using "MCI to NL" labeling. We suspect there is limited separation between MCI and NL class features while there is sufficient separation of these two classes from the Dementia features. Table IV shows the precision, recall, F1 score, and accuracy for each of these two cases.

VI. CONCLUSION/FUTURE WORK

There were two cases that we considered in this problem: "MCI to NL" and "MCI to Dementia". In all cases, Polynomial Kernel SVM, Random Forest, and Neural Network with Logistic Activation performed similarly. When considering "MCI to NL" the models performed well in terms of accuracy with a maximum score of .91. The maximum F1 score for this case, was .76 which we suspect is due to an imbalance in the number of Dementia labels (15% based off Table 1). Conversely, for the "MCI to Dementia" case, the models had a better F1 score with a maximum of .85 while the accuracy maximum was .8. In the second case, the number of Dementia data points were more comparable to NL points (at 63% Dementia data points). We suspect this is the case since the data labels are more balanced for "MCI to Dementia". In our study, our concern was with classifying subjects with AD against non-AD. The separation between MCI and Dementia is more clear than MCI and NL, so when combining MCI and NL together helps the algorithm figure out the decision boundary with less misclassification error.

If we had more time, we would perform more robust classification using 3 labels: Alzheimer/Dementia, Mild Cognitive Impairment, and Normal. Sample size is also something that can be improved; we need a dataset with complete sets of features (less missing data). We would also like to approach the problem starting from raw MRI images so that we may train a different Deep Learning model such as Constitutional Neural Networks (CNNs) that would perhaps finding stronger features as biomarkers rather than pure volumetric data.

VII. CONTRIBUTIONS

All decisions regarding the method and dataset selection was performed together by all team members. We divided up the task of training each model such that each person worked on one model. We shared the results of analysis with each other and then came together to make the poster and the final report.

REFERENCES

[1] "ADNI — Alzheimer's Disease Neuroimaging Initiative", Adni.loni.usc.edu, 2017. [Online]. Available: http://adni.loni.usc.edu.

[2] "Machine Learning classification of MRI features of Alzheimer's disease and mild cognitive impairment subjects to reduce the sample size in clinical trials - IEEE Conference Publication", Ieeexplore.ieee.org, 2017. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6091962.

[3] "Dementia", World Health Organization, 2017. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs362/en/.

[4] "FreeSurferWiki - Free Surfer Wiki", Surfer.nmr.mgh.harvard.edu, 2017. [Online]. Available: https://surfer.nmr.mgh.harvard.edu/fswiki.

[5] "OASIS", Oasis-brains.org, 2017. [Online]. Available: http://www.oasis-brains.org/app/template/Index.vm.

[6] Ramesh Kumar Lama, Jeonghwan Gwak, Jeong-Seon Park, and Sang-Woong Lee, Diagnosis of Alzheimers Disease Based on Structural MRI Images Using a Regularized Extreme Learning Machine and PCA Features, Journal of Healthcare Engineering, vol. 2017, Article ID 5485080, 11 pages, 2017. doi:10.1155/2017/5485080

[7] O. Ben Ahmed, J. Benois-Pineau, M. Allard, C. Ben Amar and G. Catheline, "Classification of Alzheimers disease subjects from MRI using hippocampal visual features", 2017. https://link.springer.com/article/10.1007/s11042-014-2123-y

[8] De Santi, Susan, et al. "Hippocampal formation glucose metabolism and volume losses in MCI and AD." Neurobiology of aging 22.4 (2001): 529-539.

[9] Johnson, Keith A., et al. "Brain imaging in Alzheimer disease." Cold Spring Harbor perspectives in medicine 2.4 (2012): a006213.

[10] Casanova, Ramon, et al. "Classification of structural MRI images in Alzheimer's disease from the perspective of ill-posed problems." PloS one 7.10 (2012): e44877.

[11] Geetha, C., and D. Pugazhenthi. "Classification of alzheimer's disease subjects from MRI using fuzzy neural network with feature extraction using discrete wavelet transform." Biomedical Research (2017): 1-1.

[12] Pedregosa, F., Varoquaux, G., Gramfort, A. & Michel, V. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[13] TADPOLE - Home. [Online]. Available: https://tadpole.grand-challenge.org/. [Accessed: 15-Dec-2017].