Final Project: Classification of different stages of Alzheimer's disease

Md Asadullah Turja[*1] MTURJA@CS.UNC.EDU
[1] *Department of Computer Science, University of North Carolina at Chapel Hill.*

1. Project Description

Alzheimer's disease (AD) is a progressive and irreversible disorder and is the main cause of dementia. Abnormal brain morphology is considered to be one of the potential biomarkers of AD. In this project, we are give a dataset with 2 structural features of the brain – Grey Matter Volume (GMV) and Cortical Thickness (CT) for a set of subjects. The goal is to classify different stages of Alzheimer's (MCI and AD) from healthy subjects (NC) using a machine learning / deep learning models.

Usually a typical classifier performs well in classifying a diseased brain from a healthy subject brain at the late stages of AD. However, Mild Cognitive Impairment (MCI) patients — a transition state between aging and AD — are harder to classify due to subtle structural changes in cortical areas. In this project, we have explored 3 different deep learning approaches to identify these subtle structural changes from different atlases. More specifically, we have trained a Multi-layer Perceptron (MLP) using several atlases as well as an ensemble of atlases. However, the network was overfitting for the higher resolution atlases as well as the ensemble of atlases (since it increases the feature size significantly). In fact the performance of the classifier seems to drop if we use more than 400 features. This hinders our ability to use an ensemble of atlases at different scales which has the potential to increase the performance and robustness of the classifier. One way to solve this problem would be to increase the data size which is not possible since we have a fixed dataset. Another way is to train a model which can utilize data from several atlases using fewer parameters compared to MLP. In this regard, we trained a multi-modal MLP (MM-MLP) which embeds different atlases to a lower dimensional latent space using separate trainable modules and then concatenates them to generate an intermediate feature vector with fewer dimensions compared to combined size of the ensemble of atlases.

The experiments show that the MLP performs best (0.6091 Macro F1 score) on the validation set when trained with *MIST-444 Template* (MIST-444) whereas *Schaefer2018-200Parcels-7Networks* (Schaefer-200-7) is a close second. However, the performance degraded by 0.3% when trained with features concatenated from both of these atlases (ensemble of MIST-444 and Schaefer2018-200Parcels-7Networks) due to overfitting. The result changes when training the MM-MLP with this same ensemble of atlases. It shows 0.8% improvement in Macro F1 score over original MLP trained with MIST-444.

Finally, we have also explored an ensemble of 3 classifiers (1 for each of AD/NC, NC/MCI, MCI/AD pairs) instead of a single 3 way classifier and then combined the results based on the confidence of these classifiers. The rationale behind this is that since MCI class is quite heterogeneous sharing characteristics with both NC and AD classes, it might degrade the performance of the classifier when trained with all 3 classes together. However, our result shows that the ensemble classifier performs worse than the 3-way classifier (2-3% less Macro F1 score).

For our final prediction on the test data, we have used the MM-MLP model trained with the ensemble of MIST-444 and Schaefer-200-7 since it is the best performing model on our validation set.

2. Model Description

We have explored the following 3 different model architectures.

2.1. Multi-layer Perceptron (MLP)

MLP is a feed-forward neural network architecture with 3 hidden layers of size 256. The input to this model is the concatenated GMV and CT features from either a single atlas (Figure 1(a)) or a multiple atlases (Figure 1(b)). In case of multiple atlases, the GVM and CT features from all the templates are concatenated together.

2.2. Multi-modal Multi-layer Perceptron (MM-MLP)

Here we train a multi-modal model (Figure 1(c)) to overcome the overfitting issues while training with an ensemble of atlases. This model first embeds the GMV/CT features from each of the template separately to a latent low dimensional space (32 units) using an embedding module). It then concatenates these latent feature vectors from each of these atlases and feed it to the classification module. The embedding module is a 3 layer multi-layer perceptron with hidden layer size 32 and the classification module architecture is the same as the MLP in section 2.1.

2.3. Ensemble MLP

In this model (Figure 1(e)), we compute results from 3 classifiers each for classifying a pair $\in \{$(NC, AD), (NC, MCI), (MCI, AD)$\}$. To combine the results, we first pass a subject through all 3 classifiers which gives 3 sets of scores $\{(p_{pair}, 1 - p_{pair})\}$. We then choose a classifier based on the entropy value e of the scores where $e = -p_{pair} \times log(p_{pair}) - (1 - p_{pair}) \times log(1 - p_{pair})$ which indicates the confidence of the classifier (lower entropy means higher confidence). Finally, the classification decision is based on the outcome of the chosen classifier.

3. Training

3.1. Data Normalization

The only feature engineering that is done is the feature standardization so that each of the features have zero mean and the resultant distribution has unit standard deviation.

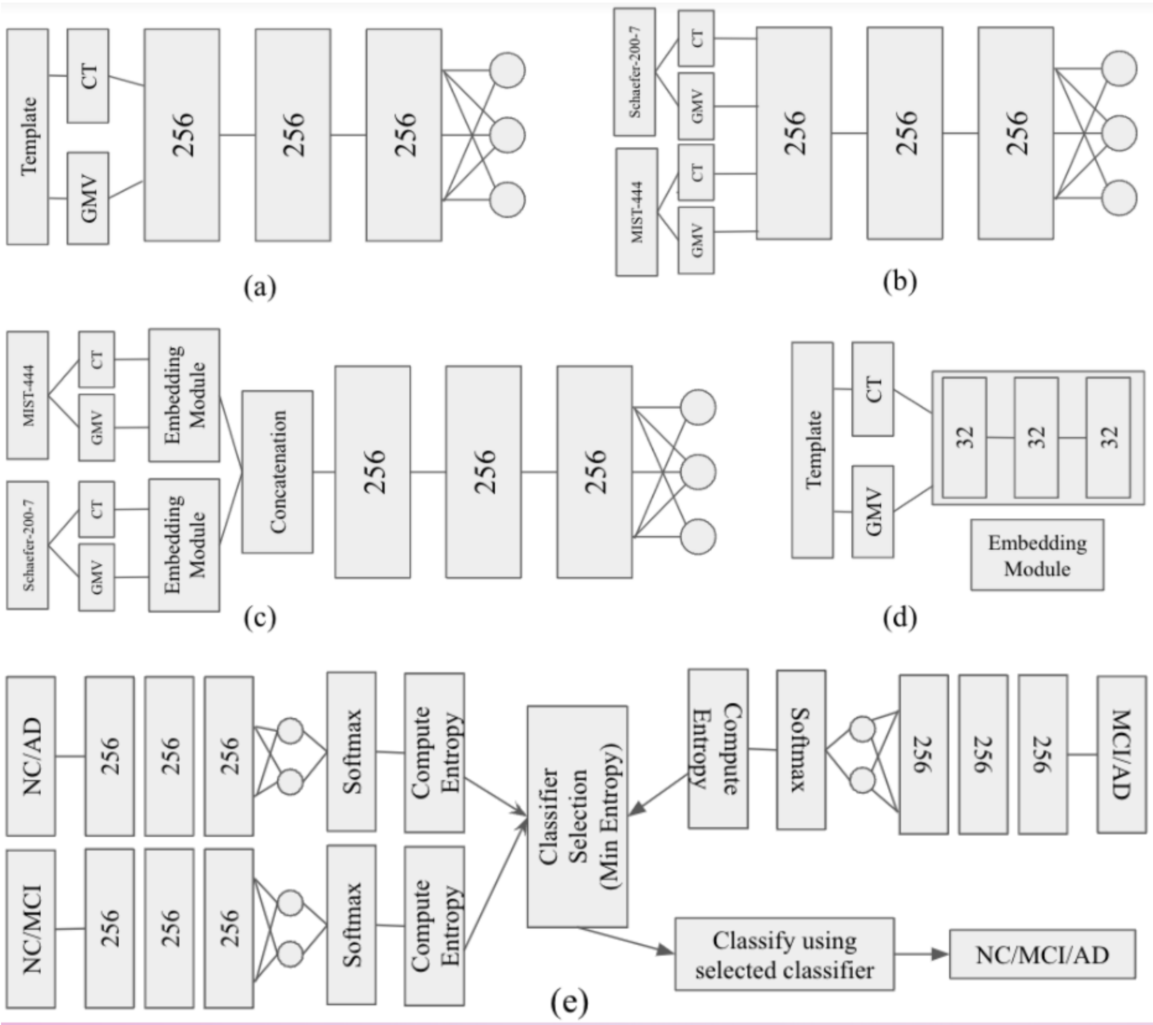


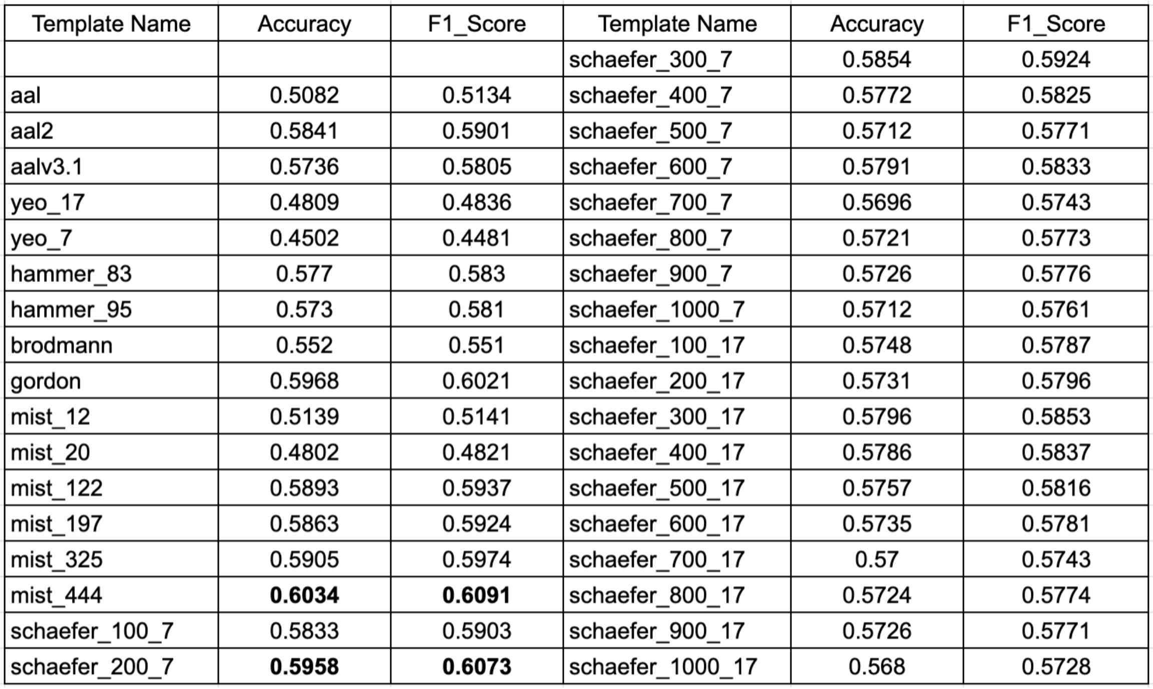
Figure 1: (a) Multi-layer perceptron (MLP) trained with single atlas, (b) MLP trained with concatenated features from multiple atlas, (c) Multi-modal MLP (MM-MLP) trained with low dimensional embedding from multiple atlases, (d) Embedding module for MM-MLP, and (e) Ensemble classifier with 3 separate MLP classifier for each pair of classes.

3.2. Hyper-parameters

All the models are trained by minimizing cross-entropy loss using Adam optimizer with learning rate is 1e-4 and batch size 32. Each of the models are trained for 10000 epochs and the model with the highest macro f1 score in the validation set is selected for evaluation.

4. Results

Our experiments show that the best performing atlas for the MLP model is MIST-444 and Schaefer-200 with macro f1 score of 0.6091 and 0.6073 respectively using both GMV and CT features (Figure 2). We then trained the MLP model on the concatenated features from these two atlases. However, the result degraded to 0.6062 which is lower than both of the single atlases. This can happen because 1. overfitting due to higher number of features, 2. different atlases can be treated as different modality and the MLP model can't utilize the differential information directly from the concatenated features. In this regard, we used MM-MLP which embeds each of the atlases into a lower dimensional latent space using separate MLP modules. This resulted in a superior performance of 0.6171 (Figure 3) which is the highest macro f1 score we got across all our experiments. This model is eventually used to generate the prediction results in the test data. On the other hand, the ensemble models seem to perform worst compared to all other models.



Template Name	Accuracy	F1_Score	Template Name	Accuracy	F1_Score
			schaefer_300_7	0.5854	0.5924
aal	0.5082	0.5134	schaefer_400_7	0.5772	0.5825
aal2	0.5841	0.5901	schaefer_500_7	0.5712	0.5771
aalv3.1	0.5736	0.5805	schaefer_600_7	0.5791	0.5833
yeo_17	0.4809	0.4836	schaefer_700_7	0.5696	0.5743
yeo_7	0.4502	0.4481	schaefer_800_7	0.5721	0.5773
hammer_83	0.577	0.583	schaefer_900_7	0.5726	0.5776
hammer_95	0.573	0.581	schaefer_1000_7	0.5712	0.5761
brodmann	0.552	0.551	schaefer_100_17	0.5748	0.5787
gordon	0.5968	0.6021	schaefer_200_17	0.5731	0.5796
mist_12	0.5139	0.5141	schaefer_300_17	0.5796	0.5853
mist_20	0.4802	0.4821	schaefer_400_17	0.5786	0.5837
mist_122	0.5893	0.5937	schaefer_500_17	0.5757	0.5816
mist_197	0.5863	0.5924	schaefer_600_17	0.5735	0.5781
mist_325	0.5905	0.5974	schaefer_700_17	0.57	0.5743
mist_444	**0.6034**	**0.6091**	schaefer_800_17	0.5724	0.5774
schaefer_100_7	0.5833	0.5903	schaefer_900_17	0.5726	0.5771
schaefer_200_7	**0.5958**	**0.6073**	schaefer_1000_17	0.568	0.5728

Figure 2: Results of MLP across several atlases

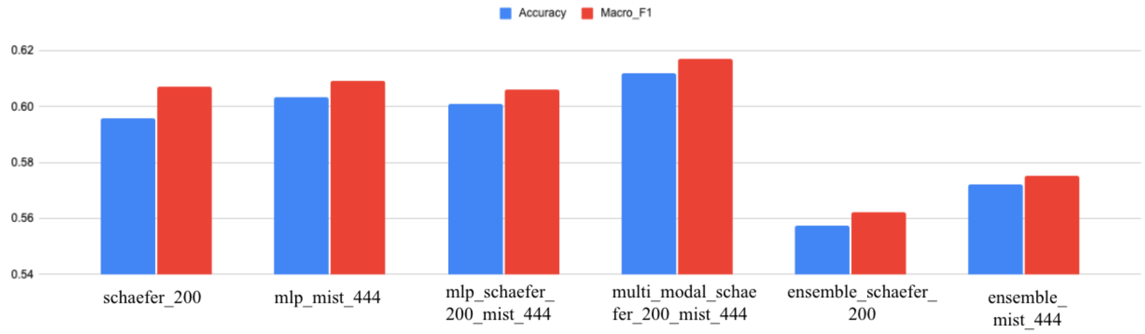


Figure 3: Comparison of the best performing atlases across different models.

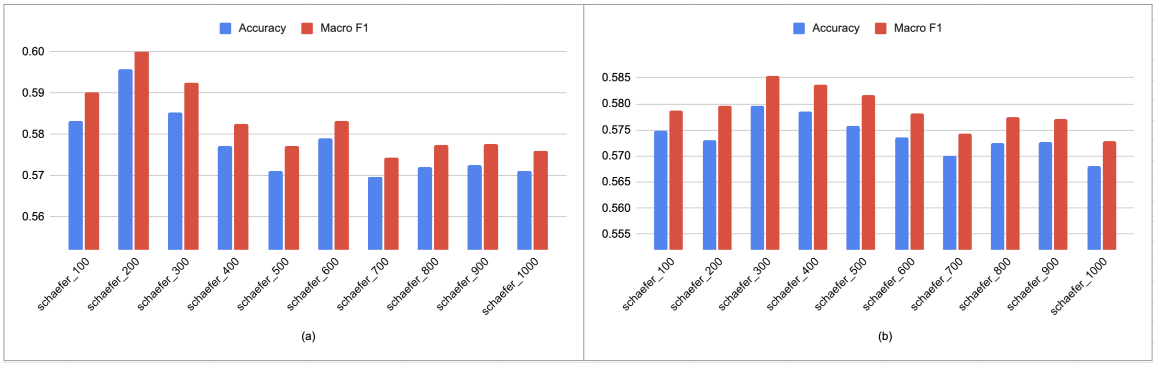


Figure 4: Comparison of the effect of ROI number for (a) Schaefer altas network 7, and (b) Schaefer altas network 17.

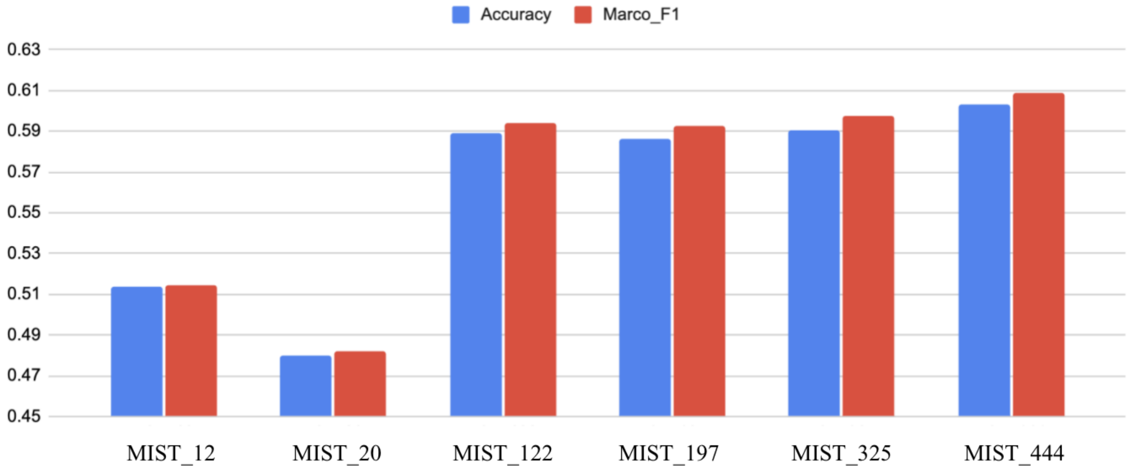


Figure 5: Comparison of the effect of ROI number for MIST atlases.

5. Discussion

Effect of the number of ROIs in atlases: The atlases in the dataset are generated by parcellating the brain cortical surface into a set of region of interests (ROIs) where the voxels in each of the ROI are structurally or functionally homogeneous. The structural measurements in our dataset are an average across all the voxels inside an ROI. This way the atlases provide a way to reduce the dimensionality of the measurements as well as noise from the data. However, if the ROI's are too large (i.e. fewer number of ROI's), this can result in over-smoothing and information loss. On the other hand, if the ROI's are too small (i.e. higher number of ROI's), training a deep learning model can lead to overfitting specially if we don't have a big enough dataset. In this project, we have investigated the optimal number of ROI's for the Schaefer atlases and MIST atlases. We found that for the Schaefer atlases the model performance peaks at 200 ROI's. If we use more than 500 ROI accuracy and f1 score degrades more than 2% (Figure 4). However, for the MIST atlases, the best model performance was achieved using MIST-444 template which has the highest number of ROI across MIST atlases (Figure 5).

Integration of multi-scale features: Atlases with low number of ROI's provide more clean and smooth data with less noise although they contain less information compared to an atlas with high number of ROI. The higher ROI atlases although noisy and prone to overfitting may contain important information for diagnosis. Fusing relevant information across different scale is a challenging and important problem to solve. In our work, we used MM-MLP to merge embeddings from atlases at different scales. MM-MLP works best with 2-3 atlases (for example, schaefer-200, schaefer-300, schaefer-400) with macro f1 score around 0.6 and seems to degrade if we add more scales.

Ensemble Classifier: In our experiments, the ensemble classifier in section 2.3 performs significantly worse compared to the single atlas MLP and MM-MLP models. This is probably due to the sub-optimal performance of the classifiers since they are not trained jointly. A better way would have been to train a common backbone network to learn a latent embedding with all the classes and then fork multiple heads for each of the classification pairs. This would be equivalent to multi-task learning with each task being a classification pair (See: MTL).

6. Important links

Github link: https://github.com/mturja-vf-ic-bd/bios772_final_project.git