# Healthcare: Alzheimer’s Disease Stage Diagnosis, Applying 3D Convolution Neural Networks to MRI Brain Scans

**Ben Moore**
Department of Aeronautics & Astronautics
Stanford University
benmoore@stanford.edu

**Mohammadhossein Shafinia**
Department of MS&E
Stanford University
shafinia@stanford.edu

**Utkarsh Tandon**
Department of Computer Science
Stanford University
utandon@stanford.edu

## Abstract

Alzheimer’s disease is an incurable, progressive neurological brain disorder and is the 6th leading cause of death in the USA [1]. Magnetic resonance imaging (MRI) is a technique used to diagnose Alzheimer’s disease in patients. In this paper we apply a number of varying neural network architectures, auto encoding and machine learning techniques to predict the condition of patients based on MRI scans. We produce multiple models with comparable performance to previous researches in the area based on ADNI dataset.

*Index Terms*— MRI, Alzheimer’s Disease, Deep Learning, Convolutional Network, Residual Network, Auto-encoder

## 1 Introduction

Earlier detection of Alzheimer’s disease is essential in order for patients to receive proper and timely treatment, to prevent further brain tissue damage. Magnetic resonance imaging (MRI) is one of the best tools to detect brain abnormalities associated with Alzheimer’s disease. Detection of Alzheimer’s disease using MRI scans is a complex and subtle problem due to the small variations between Alzheimer’s disease MRI data and standard healthy MRI data [2]. Highly skilled and experienced doctors are currently required to make a decision on the health status of a patient which could be costly. The ultimate goal of the project is to build a network that can outperform the human capability of diagnosis of Alzheimers disease using MRI scans.

The input to the 2D network and 3D network is a slice of the 3D MRI scan and the full 3D MRI scan respectively. The output of the network is a prediction assigning a patient to one of the following categories; Healthy Control (HC), Mild Cognitive Impairment (MCI), and Alzheimers Disease (AD). We used an SVM as a baseline model and implemented a number of different networks spanning a range of techniques; transfer learning with various architectures, auto-encoding, 2D and 3D convolutional neural network. Our code repository is available online[1].

## 2 Related work

To gain an initial understanding of the problem the following paper [3] provides a detailed overview into the groups of deep learning techniques including patches and voxel based implementations.

2D MRI deep learning investigation on the OASIS database [4], used 2D patches and developed deep neural network architecture to classify substages of Alzheimers disease [2]. Another 2D MRI deep learning investigation, this time on the ADNI database [5], which applies a combination of convolutions and pooling layers to predict binary classes [6].

A 3D MRI deep learning study on the ADNI database, demonstrates high accuracy results, but can be improved by the use of an autoencoder included in the architecture design [7]. Another promising 3D MRI study on CAD-Dementia, but this time with a more adaptable 2-stage architecture [8]. This paper was able to conduct feature extraction really well by introducing multiple layers of 3D convolutions and pooling. [9] proposes a residual and plain 3D CNN architecture to avoid feature extraction which requires multipe processing steps and achieves comparable results ( 80% accuracy) to previous approaches requiring preprocessing.

[1] https://github.com/benkmoore/cs230_MRI

,

[10] proposes a 3D CNN with the use of an autoencoder, in order to learn relevant features of the scan using the CAD-Dementia dataset [11] and demonstrates superior performance compared to conventional classifiers. [7] uses a pretrained one layer over complete autoencoder with weight and sparsity constraints. The pretrained autoencoder is used as the first layer in a three layer network, consisting of a maxpooling layer and then flattening the output to be connected to a dense layer. The output is a vector of AD, MCI and HC conditions.

## 3 Dataset and Features

In this study we used the ADNI dataset [5] to train and validate our 2D and 3D convolutional neural network classifiers. ADNI is currently the largest publicly available dataset for Alzheimer's Disease, with 600 unique subjects accounting for roughly 2300 scans in total. Many subjects are scanned multiple times over a 3 year span in an attempt to highlight longitudinal changes. Subjects are classified as either AD (Alzheimer's Disease), MCI (Mild Cognitive Impairment), or HC (Healthy Control). The raw T1-weighted anatomical scans were acquired with a 3D MPRAGE sequence [5]. Although this data is considered "processed" by ADNI, there is very little cleaning actually performed in the raw form. Many subjects have varying dimensions, voxel intensities, and acquisition parameters due to being scanned by different machines and at different institutions [5]. Previous studies have chosen to resample this raw data to the same shape and normalize voxel intensities, and although this leads to visually comparable samples, voxel-level information is most likely compromised. Many past studies have chosen to use almost all 2300 scans as if they are independent samples, when in actuality many samples are from the same subject - thus leading to a considerable data leakage problem (train subjects appearing in test set) [8].

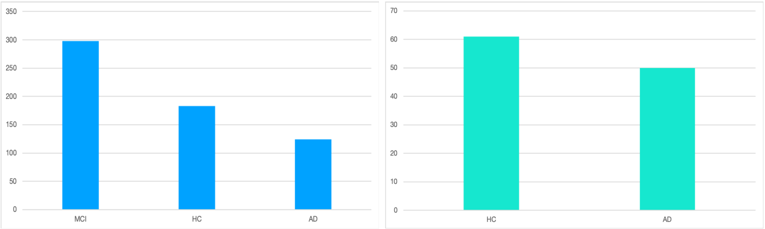


Figure 1: Histogram of ADNI datasets distribution, Raw (Left), Preprocessed (Right)

We took a different approach - choosing only the 600 unique subjects (last visit selected) and attempted to resample the raw data to a common size (192 x 192 x 160) with normalized voxel intensities. After many experiments with all of our 2D and 3D CNN approaches we realized that the data would need extensive preprocessing to be viable for a subject level CNN - especially given the low number of patients. Only a subset of these patients were viable for our preprocessing pipeline - scans labeled as "spatially normalized and N3 corrected." This included 50 AD subjects and 61 HC subjects all of the size (110 x 110 x 110). We performed a series of preprocessing steps that included skull stripping to remove non-brain voxels, voxel intensity normalization, and realignment to standard space coordinates. For our 3D classifier we had a train-validation-test split of 60%, 20%, 20%. For our 2D classifier we were able to use Keras' ImageDataGenerator to conduct data augmentation for our transfer learning pipeline.

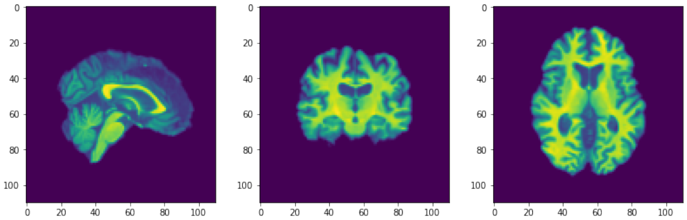


Figure 2: Preprocessed MRI images from sagittal, coronal, and axial planes

## 4 Methods

### 4.1 SVM

As an initial baseline model we implemented a SVM and used flattened single MRI slices for each patient as input to the model. We experimented with weighted class balancing and a number of various kernels and orders of the kernels (polynomial); such as linear, sigmoid, polynomial and radial basis function.

### 4.2 3D Convolutional Neural Networks

Since MRIs are inherently 3D volumetric information, a model that uses all 3 dimensions as spatial information is the logical choice for extracting disease trends. We implemented two ground-up 3D convolutional Neural Networks, VoxCNN and 3DResNet.

#### 4.2.1 3D Baseline

We initially began with a baseline 3D CNN model that used three 8 filter (3 x 3 x 3) convolution layers with (2 x 2) max pools after each convolution layer. This connected to two fully connected layers (200 and 500) and a 2 class softmax. We used cross-entropy loss and an Adam optimizer with default parameters.

#### 4.2.2 VoxCNN

The VoxCNN model takes its inspiration from the 3D VGGNet model which is commonly used for 3D volumes. It's been found to work well on MRI subject level tasks for this reason - commonly given the name VoxCNN [9]. Figure 3 shows the architecture for this deep model where a series of convolutional layers are followed by a max pool, eventually ending with a series of dense FC layers and a 2-class softmax. The input size is $110 \times 110 \times 110$ of a 3D MRI voxel density, and the output is a 2-class which is the probability of the patient to be AD or HC. The chosen learning rate was $27 * 10^{-6}$, a parameter the original paper found useful through grid search. We selected a batch size of 10 to ensure that each batch contained a spread of both classes. The optimizer was an Adam optimizer with cross-entropy loss. We made sure the class weights were balanced during training by weighting subclasses according to their split.

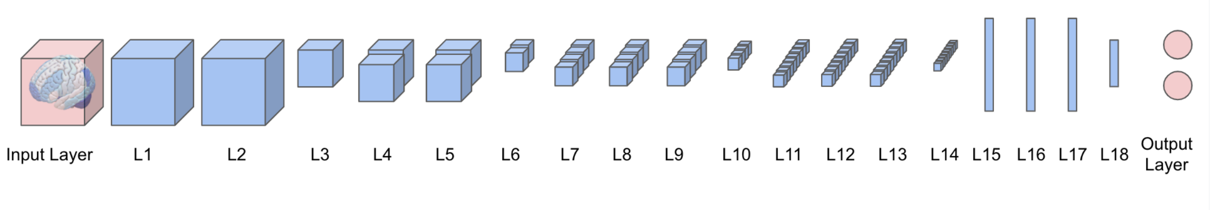


Figure 3: VoxCNN network structure, $L_{1,2}$:8 filters, $L_{4,5}$:16 filters, $L_{7,8,9}$:32 filters, $L_{11,12,13}$:64 filters all $3 \times 3 \times 3$ 3DConvNet, $L_{3,6,10,14}$:$2 \times 2 \times 2$ MaxPool filters, $L_{15,18}$:Dense Layers, $L_1$6:BatchNorm, $L_1$7: DropOut

#### 4.2.3 3D ResNet

In the 3D ResNet we use a deep residual network with the same inputs and outputs as VoxCNN but with an additional 27 hidden layers. The ResNet architecture includes three 3D convolutional layers at first followed by 6 residual networks each including a batch normalization and finally two 3D convolutional networks. The architecture of the residual network is taken from VoxNet architecture presented in [12]. At the end of the network there is a fully connected hidden layer. Also, all the parameters in the ResNet architecture are trainable.

### 4.3 Autoencoder

The Autoencoder input size is the same as the ResNet, and outputs are three states diagnostic. After normalizing image data, a 3D Convolutional Neural Network (3D-CNN) is used in two stages. The first stage is pre-training a sparse overcomplete autoencoder, to extract underlying local features of the image. Inputs of each autoencoder are a $5 \times 5 \times 5$ voxel patch, The number of hidden units is 150 and we add sparsity constraints on the hidden unit in our cost function. Semi-stochastic gradient descent is used for training the autoencoder. With the help of this autoencoder, it is possible to have a layer of 150 feature maps in the first layer of the neural net. The second layer is a pooling layer that would choose one voxel from each $5 \times 5 \times 5$ patch. The benefit of this layer is to reduce the number of parameters to avoid overfitting and reduce memory and computation costs.

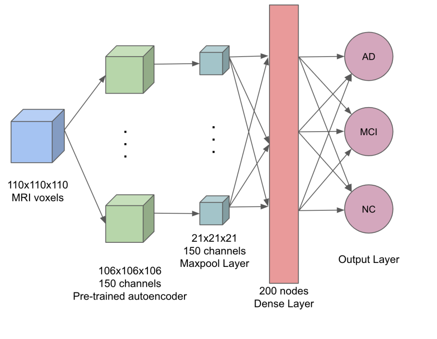


Figure 4: Autoencoder network structure

The second stage is training a fully connected 3 layer neural net that using the pre-trained CNN as the input feeding a maxpooling layer. The fully connected layer has 200 neurons. The cost function that we use in this stage is cross-entropy function and we do not optimize the first stage parameters to reduce the computational complexity in this stage. In [7] they used 800 neurons in fully connected layer; however as we had memory issues we decreased it to 200.

### 4.4 Transfer Learning

Two pretrained network architectures, Alexnet [13] and ResNet [14] were used to apply transfer learning to this problem. The following parameters were tuned using a random selection over a set range for ten epochs over the dataset; learning rate (log range), weight regularization (log range), batch size and number of frozen (non trainable) layers. We experimentedd with two loss functions, categorical cross entropy and Kullback–Leibler divergence. The Kullback-Leibler divergence is a natural distance function which measures the difference between the true probability distribution and the predicted probability distribution. We varied the slices inputted to the model through the RGB channels of the network by repeating a single layer three times or selecting three different layers.

## 5 Experiments/Results/Discussion

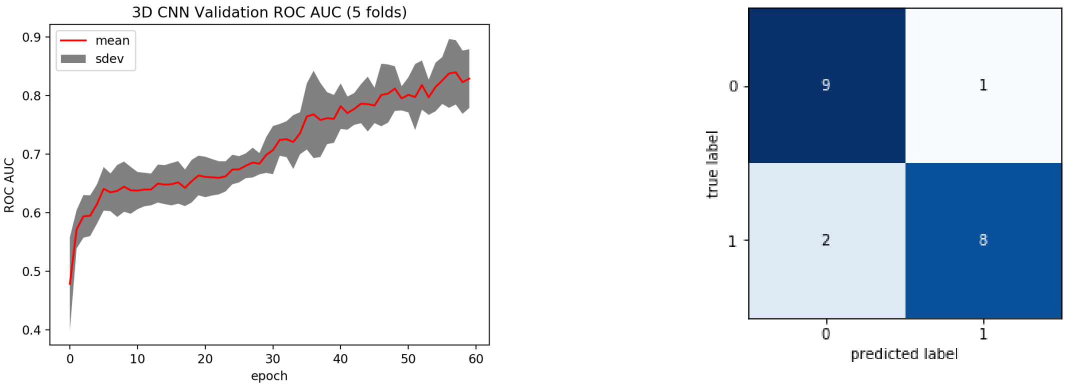
### 5.1 SVM

The SVM model struggled to differentiate between the two categories regardless of the kernel or weighted classes and produced accuracies of ~55% for both the ADNI and preprocessed ADNI datasets. After conducting error analysis it was clear to see from the confusion matrix produced by the predictions of the SVM that the model was simply predicting HC for every example. Initially we thought this might have been due to the imbalanced dataset but this problem also reappeared when balancing the classes by weight and when using the balanced ADNI preprocessed dataset. Based on the literature review it is clear that feature selection plays a critical role in this problem [7] and in this case may be a reason why the SVM performs poorly, as no features were hand designed in this case.

### 5.2 3D Convolutional Neural Networks

VoxCNN: In order to get an appropriate measure of accuracy we performed 5 folds of cross validation over our dataset. We set aside 20 subjects for testing accuracy and trained over 100 epochs for each fold, where each fold is a different split of the train-validation data. Our validation loss took a while to begin converge (after around 20 epochs). The validation accuracy also had a tendency to jump around due to the small dataset size. We then calculated the ROC AUC for train and validation on end of every epoch. These scores are then averaged and the standard deviation displayed in Figure 5a. The fold that performed best (0.86 Validation AUC) was run on our test set and produced the confusion matrix seen in Figure 5b. See poster for train/val loss and accuracy plots.

ResNet: The main characteristic of the 3D ResNet network is that it converged too fast on training data. After just five iterations we got 100 percent accuracy on the data. However, validation set accuracy was incredibly volatile. See section 5.5 for comparisons.



(a) Validation ROC

(b) 3D CNN Confusion Matrix

Figure 5: VoxCNN averaged training ROC and confusion matrix

### 5.3 Autoencoder

The autoencoder training results were satisfactory on validation sets, meaning it would compress the data efficiently as the input for the first layer. However, despite [7] achieving results of ~80% accuracy for final results, we didn't achieve comparable results from this structure. One possible reason is that although the number of parameters is fairly high with more than 8 million parameters, the number of layers is too small to be able to detect such a complicated structure. One other reason ould be that we decreased the number of fully connected layer neurons by a factor of four because of memory issues as the input size of our data is about three times bigger than what it is in [7].

### 5.4 Transfer Learning

Based on our hyperparameter search we found that the optimal parameters were a learning rate of $10^{-5}$, weight regularization of $0.01$, batchsize 1 and zero frozen layers. The smaller batch enabled more stochastic weight updates during training, the learning rate of $10^{-5}$ was small enough to ensure that the gradients did not explode and the weight regularization ensured that the model did not overfit. The single right most slice as shown in figure 2 produced the highest accuracy and F1 score compared to the other single or all layer inputs.

Initially we found that both 2D models overfit to the training data and the validation accuracies remained approximately constant. In order to address the high variance problem on the three category dataset we added dropout after a number of activations and also experimented with regularized the weights of each layer. The models then struggled to converge for both the train and validation sets so we searched randomly over a log range for the learning rate which ultimately produced a model which converged. The final ResNet model achieved ~86% accuracy and an F1 score of 0.82 on the training set.

At one point based on error analysis of the predictions it was clear that the model was repeatedly predicting MCI and was rarely predicting the other two categories. Therefore we removed the MCI samples and only inputted the AD and HC examples, as we believed that the MCI data may have been introducing noise to the dataset. The model however still struggled to differentiate between two categories.

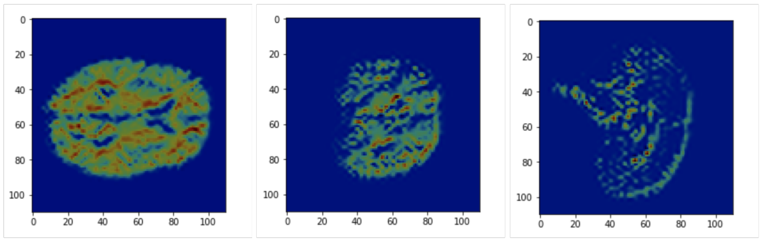


Figure 6: Class activation map of three MRI layers from an AD patient

The class activation map shown in figure 6 visualizes the attention of the model over the input by using the gradients with respect to the second to last layer. This enables the retention of spatial information in the map before it is enters the dense layer. From rightmost image in figure 6 we can see that the model is focusing on matter in the hippocampus region and in the other layers we can see that the model is looking at areas of general white matter decay which are indicative of Alzheimer's disease.

### 5.5 Experiments Comparison

Table 1: Model performance comparison

| | **Accuracy (%)** | **F1 Total** | **F1 HC** | **F1 AD** |
|---|---|---|---|---|
| **VoxCNN** | 85.63 | 0.81 | 0.78 | 0.81 |
| **2D ResNet** | 86.92 | 0.82 | 0.90 | 0.82 |
| **3D ResNet** | 83.33 | 0.74 | 0.88 | 0.74 |
| **3D CNN Baseline** | 69.56 | 0.67 | 0.72 | 0.69 |

## 6 Conclusion/Future Work

We have tried several deep learning algorithms to tackle the problem of AD classification in MRI scans. In 3D CNNs the VoxCNN algorithm had the best results with more than ~82% accuracy and 0.86 validation AUC. The 3D ResNet also had comparable accuracies however it had significant fluctuating in different epoch iterations and converged on train in less than an epoch. The transfer learning approach surprisingly provided us with the best accuracy overall (87%). We believe this is due to the ability to augment our data easily using 2D generators. The autoencoder couldn't perform well as the architecture was minimal under memory constraints - leading to poor encodings. One major challenge was that we had a lack of unique preprocessed data. Many studies have used samples of the same subject throughout the dataset, leading to data leakage. Another challenge that is high dimensionality of the input which makes the number of parameters large. For the future work we plan to manually preprocess and clean more data to feed the network. Also, given larger memory we would want to increase the autoencoder's capacity so that it can properly translate encoded info down the network.

## 7 Contributions

Ben: Transfer learning, SVM, data preparation, saliency map
Mohammadhossein: Autoencoder, 3D ResNet, data exploration, model diagrams
Utkarsh: 2D and 3D CNNs, VoxCNN, data cleaning/preprocessing, acc. plots

## References

[1] Alzheimer's disease facts and figures. `https://www.alz.org/alzheimers-dementia/facts-figures`.

[2] Jyoti Islam and Yanqing Zhang. Brain mri analysis for alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. *Brain Informatics*, 5(2):2, May 2018.

[3] Alexander SelvikvågLundervoldab. An overview of deep learning in medical imaging focusing on mri, Dec 2018.

[4] Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer's disease. `http://www.oasis-brains.org`.

[5] The alzheimer's disease neuroimaging initiative (adni) database. `http://adni.loni.usc.edu/`.

[6] Ashish Gupta, Murat Seçkin Ayhan, and Anthony S. Maida. Natural image bases to represent neuroimaging data. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–987–III–994. JMLR.org, 2013.

[7] Adrien Payan and Giovanni Montana. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks, 2015.

[8] Hosseini-Asl, Ehsan, Keynto, and Ayman. Alzheimer's disease diagnostics by adaptation of 3d convolutional network, Jul 2016.

[9] Sergey Korolev, Amir Safiullin, Mikhail Belyaev, and Yulia Dodonova. Residual and plain convolutional neural networks for 3d brain MRI classification. *CoRR*, abs/1701.06643, 2017.

[10] Ehsan Hosseini-Asl, Georgy L. Gimel'farb, and Ayman El-Baz. Alzheimer's disease diagnostics by a deeply supervised adaptable 3d convolutional network. *CoRR*, abs/1607.00556, 2016.

[11] Cad-dementia: the clinical use of computer-aided diagnosis methods for dementia by performing a large-scale objective validation. `https://caddementia.grand-challenge.org/`.

[12] Lequan Yu Hao Chen, Qi Dou and Pheng-Ann Heng. Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation, 2016.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.