Investigating Classification of Autism Spectrum Disorder using rs-fMRIs using Auto-Encoders and Deep Learning

Daniel Helm
University of Oklahoma
daniel.helm@ou.edu

Abstract

In this paper I investigate the application of convolutional networks on fMRI data in an attempt to classify subjects with Autism Spectrum Disorder. Specifically, I utilize stacked convolutional auto-encoders for feature selection and pre-trained image classification networks. I was unable to achieve results better matching state-of-the-art and mostly never get results better than chance accuracy in classification. The results presented here provide an overview of my experiments and should provide a good starting point for other researchers investigating similar problems.

1 Introduction

The ability to study the activity of the brain is crucially important to advances neuroscience, cognitive psychology and many related fields. As the technology to gather information on brain activity advances, new tools and processes are required to analyze and understand the immense amount of data becoming available to researchers.

This project explores using deep learning as one of those tools. More specifically, I'm interested in using convolutional nets and stacked convolutional autoencoders to discover features which can be fed to a classifier. To test these methods, I'm attempting to classify, using fMRI imagery, whether or not a subject has Autism Spectrum Disorder.

1.1 Autism Spectrum Disorder

Autism Spectrum Disorder (ASD) is a label for a group of developmental disorders that cover a range or "spectrum" of symptoms. ASD can range from mild to severe and often affects an individual's social interactions, motor skills, language abilities and interests. As of 2012, the Centers for Disease Control and Prevention have found about 1 in 68 children are affected by ASD. Diagnosis typically occurs by age two and is done by a behaviour-based screening test performed by early childhood health care providers. ASD research is a complicated field, and the disorder often manifests differently across ages and genders.

1.2 Resting State Functional Magnetic Resonance Imagery

Resting State Functional Magnetic Resonance Imagery (rs-fMRI) is a technique used for imaging brain activity. fMRIs are used to measure and map neural activity, but does so indirectly, as the technology is actually measuring oxygen levels in blood as it flows through the brain. This technology enables researchers to study the brain without any intrusive interventions like surgery or the ingesting of radioactive particles. "Resting State" describes the activity of the patient when this specific type of fMRI is being performed. Instead of being prompted to do some activity, physical or mental, the subject is made to remain in a resting state.

The data is best described as four dimensional. The image is volumetric as it records the three spatial dimensions of the subject's brain as well as a temporal dimension, as the data shows brain activity of the volume changing through time. Much as two-dimensional images are divided into a distinct array of pixels across a plane, this volumetric imagery is represented as a three-dimensional array of "voxels" at different intensities across time.

Although powerful, fMRI has limitations. When considering fMRI's resolution, we need to keep in mind two distinct aspects: spatial resolution and temporal resolution. Neither approach the resolution required to fully capture the size of neurons or the speed of neural activity; however, the spatial resolution is high enough to image distinct areas in the brain and researchers have had much success in using the data at its current fidelity.

1.3 ABIDE I Preprocessed Data Set

Due to factors like the cost of the machinery and the time taken to perform the procedure, fMRI data is relatively scarce. In recent years though, initiatives in neuroscience to make sets of data more widely available have given greater access to researchers who would otherwise be without immediate access to subjects, machinery or expertise.

One such initiative is the Preprocessed Connectomes Project (PCP), which released the Autism Brain Imaging Data Exchange (ABIDE). A collaboration of 16 international imaging sites, the resulting data set covers 1112 individuals, 539 suffering from ASD and 573 typical controls. The data is available in pre-processed forms to allow researchers to access standardized forms of the imagery that

have undergone various processing the fix noise, deformation and other errors that occur during imaging.

These fMRIs slices the volume of the brain into a 61x61x73 grid recorded over a variable amount of time slices, typically 100 to 200 depending on the imaging site and individual. The age, gender and handedness of the subjects also varies but is released for each subject.

The data set also provides "statistical derivatives" of each fMRI scan. These are different processes used to extrapolate specific information for the data across time. The result is a volumetric summary of different features like regional homogeneity or connectivity across the hemispheres of the brain.

1.4 Deep Learning, Image Classification and Transfer Learning

In the past few years, deep learning has excelled in a variety of areas and has made great strides in the difficult task of image classification. Much of the success comes from advancements in convolutional neural net architectures but also important are the large data sets of labeled imagery like CIFAR-10, which are freely available.

A variety of techniques have been developed in the rapidly progressing field of research, and the models are trained to recognize features in the imagery using huge labeled data sets.

Once the weights for different convolutional kernels are learned, the models handle classification tasks for new classification tasks quite well by being fine-tuned on smaller amounts of data. This process is not nearly as computationally expensive as retraining the whole model and is called transfer learning. In effect, the model has already learned to "see" various features and merely need to learn how a new category of images combines those features together.

Transfer learning is a great solution for smaller data sets. The weights of convolutional layers that define these features are tied to the structure of the convolution. In RGB photographs, convolutions happen across two dimensions using three data channels (red, green, blue), but if the organizational structure of the incoming data changes significantly (in dimension or channel), pre-trained weights can no longer be retained and transfer learning can only be attempted if the input data is restructured to meet the original convolutional structure.

1.5 Stacked Convolutional Autoencoders

Stacked Autoencoders provide an effective way to learn distinctive features in data in a semi-supervised manner. It is a supervised method in that we have an expected output for the net which allows the use of the typical backprop methods which are effective in deep neural nets. What differs is that we don't have a classification, regression or any outside information for output of the model in mind, but rather the output is compared against the input. The loss function is thus rating the network's success at reconstructing the input.

This architecture can be implemented using convolutional networks, bringing their benefits which are wonderful in spatially structured data.

In the stacked convolutional autoencoder, there is a structured input, followed by a series of convolutional layers, which are "stacked," leading to a point of minimal representation size that I call the bottleneck of the architecture. This is fed to a series of deconvolutions (sometimes called transpose convolutions), which act to reverse the original convolutions as best as possible. Finally, at the output, the intention is to have a reconstruction of the input, but after being passed through a reduced bottleneck that retains enough information to sufficiently represent the original, much larger data space. Each convolution increases the number of channels going forward, with the bottle neck having the most channels and the inputs and outputs having the fewest.

To make a classifier from an autoencoder, a typical classifier architecture is used, taking its input from the the bottleneck of the autoencoder.

2 Related Work

2.1 Deep Learning for MRI research

A variety of work has been done investigating the effectiveness of using deep learning to analyze MR imagery.

Using deep learning for neuroimaging applications has been used to study intrinsic brain networks under specific task constraints. Using structural MRI data, researchers have succeeded in classification tasks for Huntington Disease using a Deep Belief Network (DBN), which consists of stacked Restricted Boltzman machine networks (RBM) trained layer-wise. They claim "deep learning has a high potential in neuroimaging applications," though admit "we did find it difficult initially to find workable parameter regions, but we hope that other researchers won't have this difficulty starting from the baseline that we provide in this paper." (Plis et al. 2013)

The LeNet-5 architecture (LeCun et al. 1998) has been used to distinguish discriminative features for classification of Alzheimer's Disease with an accuracy of of 96.85% (Sarraf and Tofighi 2016). Their work, however, is unclear in how they use a 2D network with 4D input. They state that the 4D data was "concatenated across z and t axes and then converted to a stack of 2D images" without clarifying what that process entails. In their conclusions, they do claim that "more complicated network architecture encompassing more convolutional neural layers is recommended for future work, and for more complicated problems."

Stacking sparse autoencoders for unsupervised feature learning has also been used for classifying brain states using fMRI data. Done using a 100-voxel experiment, researchers conclude that "multi-layer non-linear feature mapping methods are promising for high dimensional input spaces with a low number of labelled samples by leveraging unlabelled data" (Vural 2014).

In an overview of 2014 studies on classification of fMRI data using deep learning, (Suhaimi, Htike, and Rashid 2006) finds "[stacked autoencoder deep learning] is reliable specifically for fMRI datasets due to very few labelled data in neuroimaging field." One of the papers it uses to reach this conclusion is a study on classifying ADHD which actually uses a Deep Belief Network (Kuang and He 2014). The fea-

tures fed to the DBN are reduced through a preprocessing techniques of FFT and wavelet transforms on 48 volumetric areas according to the "brodmann template," with features from each area fed to a distinct DBN whose outputs are fed into another DBN.

The ongoing research that is most closely comparable to the work I'm pursuing in this paper is that of Vigneshwaran et al. Using the ABIDE dataset, they have pursured various methods for classifying ASD subjects using deep learning. In (Vigneshwaran et al. 2015), they specifically take a whole brain approach, exploring the possibilities of rejecting analyses using Region of Interests, such as in the ADHD study above. To do this, they make use of a specific statistical derivative called Regional Homogeneity (ReHo) because it "evaluates the similarity between the time series of a given voxel and a specific number of its nearest neighbors." Interestingly though, the researchers then use a standardized template for Automated Anatomical Labeling to select voxels in 116 ROI to limit their feature space before using a Chi-square statistical analysis to determine their final input features. Using a Projection Based Learning Meta-cognitive Radial Basis Function Network Classifier, they were able achieve impressive results that I will use as a comparison for my own results.

3 Problem Definition and Algorithm

3.1 Task Definition

My task is to build a classifier that takes as its input the fMRI data of a subject and output whether the subject has ASD. This is a compelling problem because ASD is not typically diagnosed from observations of the brain, and the data could reveal interesting information about the brain function of ASD individuals. If successful, this could open an avenue for machine learning to be applied to various other problems and diagnoses where medicine suspects differing patterns of brain activity but has not yet been able to pin down what those difference are and to do so from a whole brain perspective, without any a priori assumptions about the differing patterns of brain activity.

My hypothesis was that using these methods I'd be able to achieve a classification accuracy above chance, which given that the class distribution of my dataset was roughly equal, was 50%. I used classification accuracy as my metric of success. Beyond that, I was hoping to match the success rates seen in (Vigneshwaran et al. 2015). I have not achieved this ambitious task, so the remainder of this paper documents the architecture design decisions for my various experiments. I hope these observations will be a good starting point for other researchers.

3.2 Algorithm Definition

The experiments detailed below can be seen as occurring in two realms: a) model architecture and b) data selection and representation in that architecture. For both of the model categories, I took advantage of the statistical derivatives provided by ABIDE. Given a volume of 61x61x73 across 100 time steps (27 million data points) for each subject, using the raw data in its original form is infeasible.

In addition to this, in the majority of my experiments, I've chosen to limit my training data to males ages 18 and older, in order to simplify the learning problem. This accounts for the differences in how ASD manifests differently in developed brains and by gender. This also helped maintain decent batch sizes for when larger models pushed the 8GB memory limit of my GPU. In all experiments I used TensorFlow.

For each architecture, each statistical derivation was scaled to a $[0, 1]$ range across the entire training data set. The multipliers used for scaling the training set were then applied to the test set.

Transfer Learning For transfer learning, I used the 19-layer deep convolutional network architecture from (Simonyan and Zisserman 2014), pre-trained on ImageNet. This trained model is used for 2D data with 3 channels, which is not ideal for volumetric data, but 3D convolutional architectures are in their infancy and trained models are not widely available.

The challenge here is how to represent the statistical derivations, which are volumetric in nature, as 2 dimensional, 3 channel images.

For my first experiment with this model, I manipulated a statistical derivative volume by cropping it to a bounding box that kept nearly all non-zero imaging data. From there I flattened the array and removed 516 data points from each end for the purpose of reshaping it to a 3 channel, 2-D array with equal length sides. The data maintained a structured form, but not one coherent to a human as seen in Figure 1.

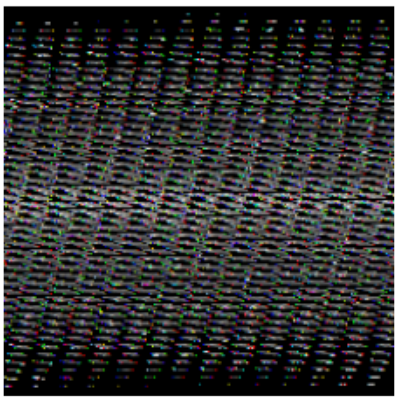

Figure 1: Visual result of naive reshaping of volume for transfer learning

With enough training, the training set accuracy actually started to increase as high as about 60%, however the test set's classification accuracy remained below chance. As the imagery was mostly noise, it seems the net was deep enough to overfit through learning identifying characteristics from individual training images' noise patterns.

My next experiment was using three contiguous 2D slices of the brain volume, each placed in a color channel. Here, training accuracy was as low as 76% and as high as 92%

after just 300 epochs, depending on learning and dropout rates. Testing accuracy remained below 50%.

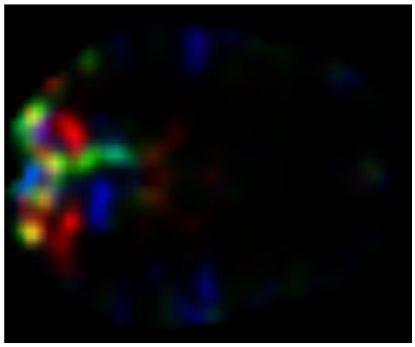

Figure 2: Visual result of placing 3 contiguous slices as distinct color channels using the lcfd derivative

Believing that this method indiscriminately isolated just a portion of the data, I experimented with a method that utilized one derivative volume per a color channel. For each derivative, I would take the maximum value per voxel column. This can be understood as looking down on the volume and choosing a value only from the slice which had the maximum value, as seen in Figure 3. The results were the same as above. I also tried using the mean column value to no effect.

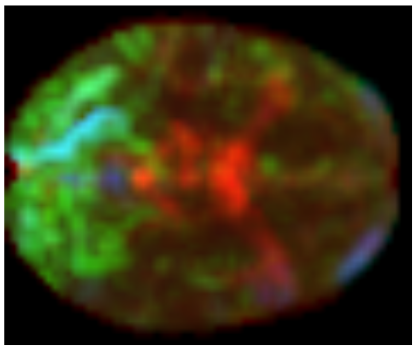

Figure 3: Visual result of using a top-down max to determine pixel value

I began to worry that although I was summarizing the derivatives in a specific way, I was flattening the space such that the projection for 3 dimensions into 2 was losing a considerable amount of location information. I was inspired by the results of (Vigneshwaran et al. 2015) from using a single statistical derivative and some neuroimagery overviews that illustrated 3D location by showing projections from the x, y and z dimension. So, for my final experiments I tried using the max projection method above on a single derivative, but from the perspective of the x, y and z axis. Each were assigned to a different color channel, resulting in an image like that in Figure 4. These results were the most promising and will be further addressed below.

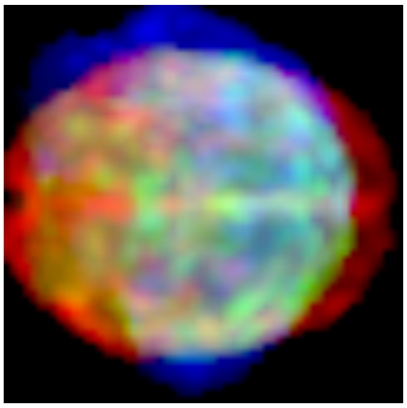

Figure 4: Visual result of using projecting across each axis per color channel

Stacked Convolutional Autoencoder The other architecture I experimented with was stacked convolutional autoencoders.

In my autoencoders, I set the input data as a volume with four channels, each channel being a different statistical derivative. This, in theory, allows kernel filters to find regional features across the various summary types and to develop complex, non-linear functions relying on interactions between the statistical derivatives.

I ran quite a few experiments here, at times systematically, other times shooting in the dark, in an effort to identify the effects of different parameters. I experimented with altering *(with final preferences in bold)*:

- Activation functions: Sigmoid, ReLU, **ELU**
- Weight Initializations: Constant, **(He et al. 2015)'s** $\sqrt{2/n_i}$
- Learning Rate: Constants, **Exponential Decay**
- Architecture Form: Very Deep & Narrow, **Shallow & Wide**
- Convolution Aspects: Kernel Size, Stepping Size, Bottleneck Size
- Pooling: Max Pool, 2x2x2 kernel convolutional layer stepping by two, **Integrating pooling into previous layer via stepping size** > 1
- Layer Training: **All at Once**, Greedy Layer-wise
- Data Processing: Subtracting the mean across the dataset for each channel and scaling, **only scaling**

The culmination of these choices has resulted in architectures that have best reduced the cost function in fewer epochs. All fail once their bottleneck is fed as the input to a fully connected network using a softmax classifier trained using a cross-entropy loss function. As seen in Figure 5, the test set never achieves better than chance accuracy.

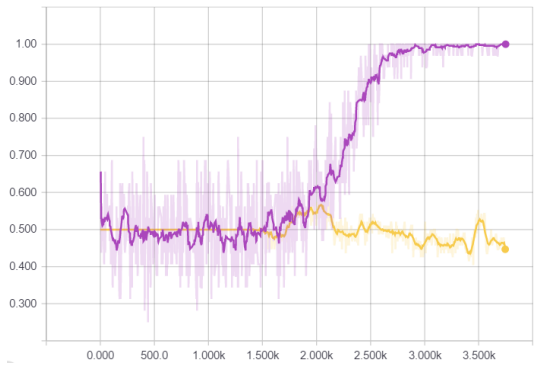

Figure 5: Classification Accuracy for Training (magenta) vs Test Sets (yellow) after 3700 epochs. Darker lines are smoothed.

Of note: Several choices above are not what I'd consider "best practices," but rather what I could achieve given the hardware limitations of my GPU and the memory-hungry nature of 3D convolution and TensorFlow. Max Pooling, for example, though having no learned features to consume memory, is implemented in such a way in Tensorflow that the full size output of the previous layer is written to memory, causing each pooling layer to result in ballooning memory usage. Given some researchers' movement away from max pooling layers (Springenberg et al. 2014), I chose to use the memory instead for wider and deeper architectures.

My final autoencoder had 4 convolutional layers and 4 deconvolutional layer, each with a stride of 2 and a kernel size of 7x7x7. The channels for each layer were 32, 64, 128, and 256.

In the following section I will explore why classification failure is likely due to the autoencoding process, and what I've done to try to solve the problem.

4 Experimental Evaluation

I will use this section to discuss what I think the problems are and some of my attempts to fix them. In the only experiment to get results, I will compare them to those of (Vigneshwaran et al. 2015).

4.1 Transfer Learning

My most successful experiment in transfer learning achieved results averaging just below 60% accuracy by 300 epochs. I used the alff statistical derivation in this experiment using a maximum value projection across each axis for each color channel, while considering only male subjects 18 and older. (Vigneshwaran et al. 2015) were able to achieve 79.4% accuracy given the same gender and age constraints using the ReHo derivatives (my model achieved only chance results with ReHo). They also achieved over 68% accuracy for adolescent males, while my results hovered just above chance.

Note: These results were achieved just before completing this paper, so they represent only a single run. I was unable to do multiple runs to determine if the model would be successful given differing test/training sets.

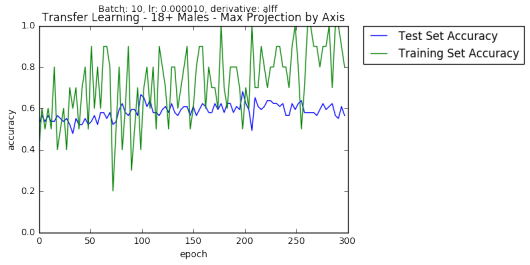

Figure 6: Transfer Learning results using Axis Projection Method on alff derivative

I suspect that even if 2D transfer learning can be a viable solution for a problem as complex as fMRI data, there is not currently enough data being retained for the classifier to be trained on. That is, for my first few experiments, there may not be any patterned distinctions left in the input data between ASD and typical subjects for the model to identify after slicing a single derivative to just three layers. For the experiments using max projections, there is more distinguishing information retained, but the features are likely not chosen with the discrimination as previous research which did not require features to maintain a spacial structure.

4.2 Stacked Convolutional Autoencoders

Before building the classifier part of the network I was very optimistic about my autoencoder architecture. Real learning was occurring, and I was able to greatly reduce the SSE cost function between the inputs and the output by implementing an exponential decay rate. I was also able to decrease the number of epochs required for convergence by implementing He et al's Weight Initializations (He et al. 2015) along with ELU activation functions. What follows is my attempt at now diagnosing why classification is failing.

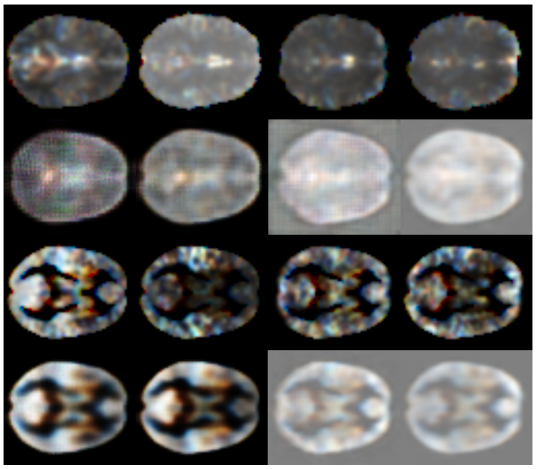

Figure 7: Rows 1 & 3 are inputs, Rows 2 & 4 their reconstructions. Columns from Left to Right: Sigmoid Shallow: 5000 epochs, Sigmoid Shallow: 8000 epochs, ELU Shallow, ELU Deep

Mean Reconstruction Analyzing the imagery of the reconstructed outputs, I could see that the reconstructions across subjects were distinct but still quite similar. I figured that my models were learning a mean of the data set in order to quickly reduce the loss across batches of images. The model was doing this as a way of "hedging its bets" against differences in the dataset it hadn't yet learned to reproduce. Generalization is good, but only to a point. The problem is that the models never learned to reproduce those differences.

This technique of minimizing the loss function was successful for the model because of the consistency of a highly structured input. In preprocessing, the volumes have been transformed to have nearly identical shapes in the volume space.

My initial attempt at fixing this was by first calculating the training set mean and subtracting it from each subject before input. The result was that reconstructed images were almost entirely black. The models were simply learning the *new* mean of the dataset and classification was still failing.

I suspected the models were learning the mean, but needed to confirm it as the reconstructed outputs weren't identical.

Dot Test At this point I realized I needed to confirm that my model could solve a simpler problem. I devised an experiment where I would place a clear marker into all subjects with ASD and omit this marker from typical subjects. The result was a "Dot Test" where I generated a 3x3x3 cube in each channel for ASD subjects in a consistent position in the volume. I trained the model on this altered dataset and still got poor results. Reviewing the reconstructed output images, the reason was obvious: the model was in fact learning the mean as the dot was appearing in typical subjects.

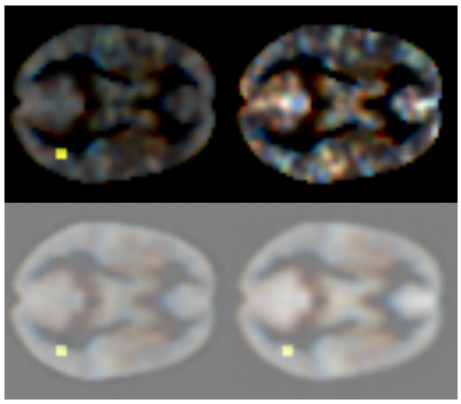

Figure 8: Top Row: Inputs, Bottom Row: Reconstructed Outputs

My inclination at this point was to do layer-wise training of the auto-encoder. I trained a network of only the outer layers for 700 epochs, then inserted the next pair of convolution deconvolution layers while deactivating weight updating for the already trained layers. I repeated this until all layers were trained. I was still unable to pass the "dot test."

Unfortunately, this is as far as I've gotten in solving this problem. My intuitions on how to continue are explained below.

5 Future Work

There is plenty more work to be done in this area, especially with respect to 3D convolutional networks and auto-encoders.

For autoencoders, I think the next step would be implementing a denoising auto-encoder or exploring techniques from DBNs and RBMs. I know a craftsperson never blames their tools, but I believe a huge limitation of my approach currently is the size to which these networks are restrained. I was only able to create about 4 layers, none of which had a stride of 1, before running out of memory on my GPU and CPU training was infeasible. The encoding portion of my auto-encoder was shallower and narrower than state-of-the-art image classification networks and given the 3D nature of the problem, it likely needed to be at least a large to accomidate 3D features.

In moving forward, I'd try to implement the architecture used by Facebook's C3D project for classifying videos. They've achieved impressive classification results using 3x3x3 kernels, max pooling layers, and many more convolutional channels and layers than my system would support. It too would be a good candidate for transfer learning in the future, as they're released a trained model for sports video classification. I hope that other researchers might have better luck than I did in compiling their custom fork of Caffe.

Another area where research on fMRI is inherently limited for deep learning is in sample sizes. Clinical trials are small and expensive, and although data sets like ABIDE help to aggregate many trials, as mentioned above, the total is still small in comparison to the feature space. In future works, I would hope that a generalized auto-encoder could be built from not only the ABIDE dataset, but additional data sets encompassing various diagnoses. A preprocessed data set already exists for ADHD, and as the International Neuroimaging Data-Sharing Initiative (INDI) continues to grow, we can expect to see more open neuroimagery data to explore. If a more generalized brain model could be represented in a much smaller feature space (and even standardized like many image classification models are), researchers could use transfer learning to help investigate smaller-scale trials in new areas of research quite painlessly.

6 Conclusion

In this paper I've provided an overview of several, mostly unsuccessful attempts at building a classifier for ASD from fMRI data using deep learning techniques. I have found that 2D convolutional networks show promise but in my implementations still need refinement. My implementations of 3D convolutional auto-encoder networks have not been successful, but research suggests that larger networks show promise and hopefully future investigations in this area will be more fruitful.

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on im-

agenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.

Kuang, D., and He, L. 2014. Classification on adhd with deep learning. In *Cloud Computing and Big Data (CCBD), 2014 International Conference on*, 27–32.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Plis, S. M.; Hjelm, D. R.; Salakhutdinov, R.; and Calhoun, V. D. 2013. Deep learning for neuroimaging: a validation study. *CoRR* abs/1312.5847.

Sarraf, S., and Tofighi, G. 2016. Deep learning-based pipeline to recognize alzheimer's disease using fmri data. *bioRxiv*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. A. 2014. Striving for simplicity: The all convolutional net. *CoRR* abs/1412.6806.

Suhaimi, N. F. M.; Htike, Z. Z.; and Rashid, N. K. A. M. 2006. Studies on classification of fmri data using deep learning approach.

Vigneshwaran, S.; Mahanand, B.; Suresh, S.; and Sundararajan, N. 2015. Using regional homogeneity from functional mri for diagnosis of asd among males. In *2015 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Vural, O. F. L. O. F. T. Y. 2014. Deep learning for brain decoding. *2014 IEEE International Conference on Image Processing (ICIP)*.