IMPROVING AUTISM SPECTRUM DISORDER MRI CLASSIFICATION WITH GENERATIVE ADVERSARIAL NETWORKS

Chelsea Alvarado
cxa6ky@virginia.edu

Dylan Howe
dsh7pd@virginia.edu

Ellen Yu
eyy8k@virginia.edu

May 6, 2022

ABSTRACT

Autism Spectrum Disorder(ASD) encompasses a wide range of neurodevelopmental disorders that primarily affect social skills, communication, and other related behaviors. The road to diagnosis, for many families, is a long and expensive route made up of a battery of psychological tests and doctors appointments. The increased use of deep learning methods to diagnose via medical imaging provides the opportunity to decrease both the financial costs and time to diagnosis for families seeking an answer. Through the use of MRI imaging from the Autism Brain Imaging Data Exchange (ABIDE) we applied two different deep learning approaches, Dense Feedforward Neural Networks (DFNN) and a 3D - Convolutional Neural Networks (3D-CNN). A simple Generative Adversarial Network (GAN) was used to create new data in order to train predictive models. The predictive models that performed best varied depending on the addition of GAN data. The DFNN model performed best on data without added GAN data while the 3D-CNN performed the best overall when combined with GAN created data.

1 Introduction

1.1 Background

Autism spectrum disorder (ASD) is a social and developmental disability that 1 in 44 American children are diagnosed with by the age of 8 [2]. The process of diagnosis is complex and long since it involves developmental monitoring and screening over months and potentially years [3]. Given the complexity of ASD symptoms and obtaining a confident diagnosis, children may wait until adolescences or even adulthood to receive a diagnosis depending on the presentation of symptoms. Consequently, children who do not receive a timely diagnosis fail to receive services and supports that could help them reach their full potential. According to Mellema (2022),

> "Autism spectrum disorder (ASD) is currently diagnosed through a time-consuming evaluation of behavioral tests by expert clinicians specializing in neurodevelopmental disorders. This diagnosis can be challenging due to several factors including the heterogeneity of the spectrum disorder, the uncertainty in the administration and interpretation of behavioral tests, and neurobiological and phenotypical differences that vary only slightly compared to typically developing controls."

The need for robust, accurate, and timely diagnosis methods is very evident. Deep Learning methods provide an opportunity to help bridge the gap. Methods such as CNNs are powerful and already produce some of the best results for brain image scanning.

1.2 Motivation

The motivation behind our project is reducing the time to diagnosis for individuals who may have ASD. The diagnosis of ASD is a long and financially burdensome process for families due to the number of psychological tests and other diagnostic appointments that are required before a final diagnosis can be made. Having the ability to accurately and quickly diagnose an individual would address some of the barriers to seeking diagnosis and treatment. Additionally,

it would help identify individuals who perhaps do not show the "standard" symptom profile, such as is the case with females [7]

1.3 Literature Survey

Advances in neuroimaging, machine learning, and deep learning have allowed new diagnostic methods to be possible for use in ASD diagnosis. The literature on neuroimaging diagnostics has heavily trended toward application of deep learning methods including Autoencoders, Deep Neural Networks, and Dense Feedforward Neural Networks [9] [14]. Traditional machine learning approaches such as Support Vector Machines have also been researched and shown to have similar performance to deep learning models [10]. Further research findings involving deep learning models have shown that when comparing different linear, nonlinear, and deep learning models, the deep learning models such as Dense Feedforward Neural Networks(Dense FNN), Long short-term memory Recurrent Neural Networks (LTSM RNN), and BrainNet Convolutional Neural Networks (BrainNet CNN) outperform other model types on the task of diagnosing Autism Spectrum Disorder via brain MRI's [9].

The models that the researchers across the literature had created and benchmarked were primarily trained on variations of the ABIDE I and ABIDE II ASD brain functional magnetic resonance imaging(fMRI) data set, a data set that contains images of over one thousand participants including controls. The data set is publicly available, which has led to its prominent use across ASD neuroimaging literature [1].

1.4 Hypothesis

Our intention for this project is to build upon and improve the classification results found in our literature review. Our proposed approach is to address the lack of available training data and unbalanced nature of the data set (gender diversity) by using a GAN to generate new examples of MRI's from individuals diagnosed with ASD. We hypothesize that when keeping the model architecture the same, models that are trained on the GAN-augmented data set will outperform our models that were trained on the regular ABIDE data set.

2 Materials and Methods

2.1 Data

The Autism Brain Imaging Data Exchange I (ABIDE) is a collection of functional MRI's and phenotype/co-variate data collected from 1112 individuals across 17 international sites. Of the 1112 samples, 539 are cases for individuals diagnosed with ASD and the remaining 573 are controls that do not have any ASD diagnosis. The data set has been anonymized in compliance with HIPAA guidelines and the fMRI scans have been pre-processed to remove noise and clarify the signal through one of four different preprocessing pipelines [1]. Additional co-variate data is available such as co-morbidities and other psychological testing results.

The data selected for training consisted of both male and female subjects who were between the ages of 3-18. The data provided by ABIDE is provided by multiple study sites. In order to try and address any inconsistencies between site results we ensured that data from any study site must be available for both male and females in order to be included in our final dataset. Our final data distribution consisted of 22 female cases, 61 female controls, 124 male cases, and 184 male controls for a total sample size of 391. All data was preprocessed by ABIDE and our group specifically chose to use data preprocessed using the Connectome Computation System (CCS) pipeline [13].

2.2 Methods & Experiments

The workflow for this project was fairly simple. Since our data was already preprocessed we were able to skip that but still performed data preparation and exploration. This was followed by initial model testing, training, and evaluation on our base data set. Once our evaluation was complete and scores recorded the focus shifted towards designing, testing, and producing a GAN. The data produced by the GAN was then fed into the model architectures established during initial model training. The last step was evaluation on the holdout data on the newly trained models. A workflow diagram can be found in figure 1.

2.2.1 Baseline Model

Our baseline model is a 3D Convolutional Neural Network used by Zunair et al. to evaluate the effect that different preprocessing techniques had on 3D image classification in CT scans [15]. The architecture was selected for our project

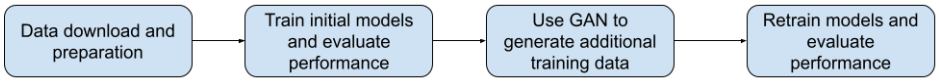

Figure 1: End-to-End Pipeline

6 Hasib Zunair, Aimon Rahman, Nabeel Mohammed, and Joseph Paul Cohen

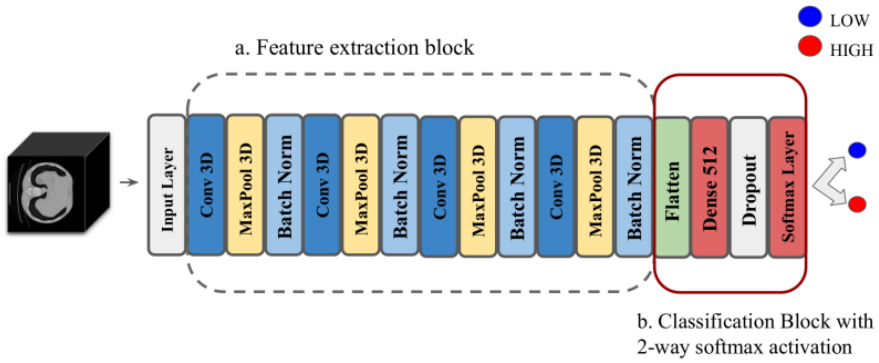

Figure 2: Baseline 3D-CNN Architecture[15]

because it has a proven track record of classifying 3D brain scans, and was likely to adapt well to our data. The architecture consists of:

- Input layer that takes an input shape of (61,73,61,1)
- A normalization layer (Added by our group to improve training performance)
- Two sets of:
  - 3D convolutional layer with 64 filters, kernel size = 3, RELU activation, and "same" padding
  - MaxPool3D layer with pool size = 2
  - Batch Normalization layer
- Two sets of:
  - 3D convolutional layer with 128 filters, kernel size = 3, RELU activation, and "same" padding
  - MaxPool3D layer with pool size = 2
  - Batch Normalization layer
- Global Average Pooling 3D layer
- Dense Layer with 256 neurons and RELU activation
- Dropout layer that drops 30
- Output layer with sigmoid activation

The baseline model was able to obtain over 80% accuracy during initial testing but only 50% recall when classifying positive (ASD diagnosis) cases. This is likely a result of the imbalanced nature of the training dataset before augmentation.

2.2.2 GAN

Our first GAN consists of a standard architecture of a flatten layer and 3 dense layers for the discriminator and 3 dense layers and a reshape layer for the generator (Figure 3). For the number of nodes, we used 150 and 100 in the hidden layers (Figure 4), which was inspired by a GAN architecture Géron built for a (28, 28, 1) dataset[4]. Comparatively speaking, 100 and 150 is low for our dataset consisting of (61, 73, 61) dimensions, but we wanted to start with a small architecture and build up if necessary. For our first GAN, we tuned on coding size which is the latent representation worked on by the generator and number of epochs which is the number of training iterations. We evaluated the GAN by looking for convergence of the generator loss, visually inspecting all generated samples, and examining GAN evaluation

metrics, peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM). Our first GAN took 10 minutes to train and we trained one per under-represented class.

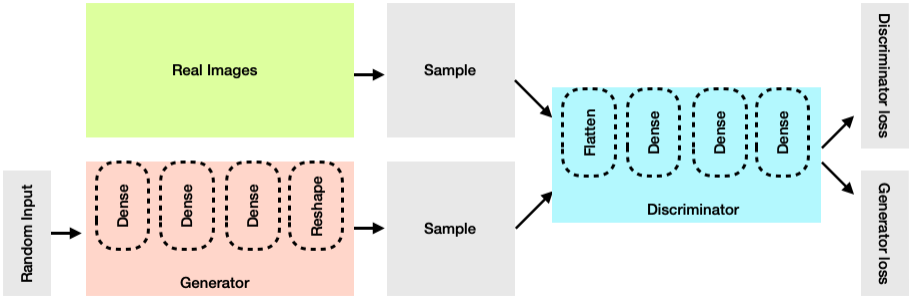

Figure 3: Overview of GAN Architecture

- Discriminator that consists of:
  - Flatten layer that takes an input shape of $(61, 73, 61)$
  - Dense layer with 150 neurons and SELU activation
  - Dense layer with 100 neurons and SELU activation
  - Dense layer with 1 neuron and Sigmoid activation
- Generator that consists of:
  - Dense layer with 100 neurons and SELU activation that has a coding size of 3000
  - Dense layer with 150 neurons and SELU activation
  - Dense layer of 61 * 73 * 61 neurons and Sigmoid activation
  - Reshape layer that has a target shape of $(61, 73, 61)$

Figure 4: Detailed Architecture of First GAN

Our second GAN utilized the same general architecture as our first GAN but scaled the number of nodes in the hidden layers to 5000 and 3000 (Figure 5). We wanted to increase complexity because we noted that the generated images were of lower fidelity according to our GAN assessment metrics and through visual inspection of our samples. For our second GAN, we used the best coding size and number of epoch as determined by our earlier tuning. We inspected convergence, calculated loss of fidelity, and visually examined our generated samples. Our second GAN took 4 hours to train and we stayed consistent with our earlier tactic of training GANs by class.

Since our goal is to improve the classification model for ASD, the ultimate test is whether the use of the generated images increases our AUC and accuracy scores.

- Discriminator that consists of:
  - Flatten layer that takes an input shape of $(61, 73, 61)$
  - Dense layer with 5000 neurons and SELU activation
  - Dense layer with 3000 neurons and SELU activation
  - Dense layer with 1 neuron and Sigmoid activation
- Generator that consists of:
  - Dense layer with 3000 neurons and SELU activation that has a coding size of 3000
  - Dense layer with 5000 neurons and SELU activation
  - Dense layer of 61 * 73 * 61 neurons and Sigmoid activation
  - Reshape layer that has a target shape of $(61, 73, 61)$

Figure 5: Detailed Architecture of Second GAN

2.2.3 DFNN Models

The main approach taken by the group was to first model the DFNN architectures reported to obtain the best performance by Mellema 2022. The complex dense network and highest performing dense network architectures shown in Figure 6 were the base models for training our predictive models. Our final three models after model testing were:

- Highest performing dense network architecture
- Highest performing dense network architecture with modifications
- Complex dense network architecture

Simple dense network	Complex dense network	Highest performing dense network
L2 Regularization: 2.3e−4	L2 Regularization: 2.3e−4	L2 Regularization: 1.1e−4
Dense: 16 neurons	Dense: 128 neurons	Dense: 64 neurons
Dropout: 53% removed	Dropout: 18% removed	Dropout: 13% removed
Dense: 16 neurons	Dense: 128 neurons	Dense: 64 neurons
Decision Layer: 1 neuron	Dropout: 18% removed	Decision Layer: 1 neuron
	Dense: 64 neurons	
	Dropout: 18% removed	
	Dense: 42 neurons	
	Decision Layer: 1 neuron	

Figure 6: DFNN Architectures [9]

Access to all the parameters used by the original authors, such as activation functions, optimizers, or learning rates were not available for us to use when training our models. In order to best address these gaps we chose to use RELU as the activation in the dense layers, a sigmoid activation for the output layer, and Adam as the optimizer with its default learning rate but paired with a learning rate scheduler. Since the primary performance metric used by Mellema 2022 was AUROC we used that as our main performance metric as well.

In order to address instability of metrics during training of the DFNN models we first chose to increase our batch size to 32, which did help to slightly stabilize scores between epochs. Due to memory limitations and TensorFlow errors that arose on Rivanna, we were not able to train batch sizes over 16 despite being able to do larger batch size earlier in prior to the start of May.

Our best performing model of all the models tested ended up having the architecture shown in figure 7.

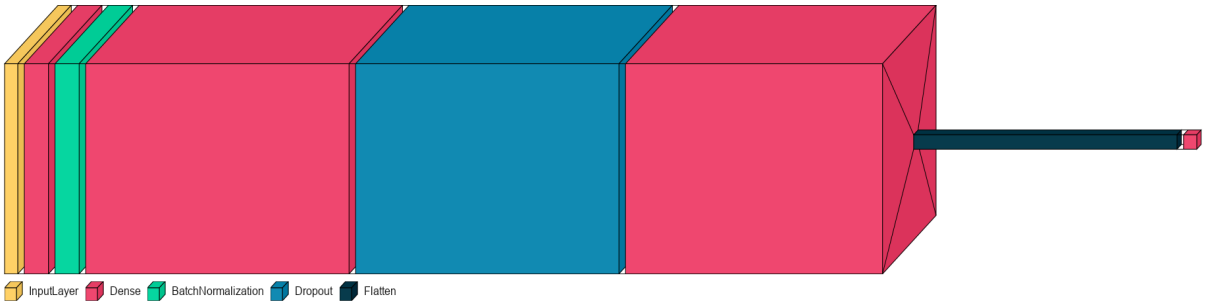

Figure 7: Best performing DFNN Architecture

The modified architecture was developed through testing primarily the addition of batch normalization layers throughout the base architectures. Additional dense layers were tested but they generally did not increase metrics by much, if at all. More specifically, the modifications in the highest performing dense network in figure 6 were the addition of a batch normalization layer after the firs dense layer and increasing the number of neurons in the second dense layer from 64 to 128. Our goal was to keep the architectures minimal simply due to the amount of hyper-parameters there are to tune and shown performance in a simpler architecture by Mellema 2020's benchmark scores. Additional optimizers, such as RMSProp, were also tested in conjunction with different architectures but the Adam optimizer always performed best on DFNN model architectures.

2.2.4 Model Deployment

The group deployed the trained 3D-CNN model to Google's Cloud AI Platform using the SavedModel format from TensorFlow. The deployed model is available on the cloud for prediction services, which would enable imaging labs to submit MRI data directly to the predictive model for near-instant diagnosis. The deployed model is set up with automatic scaling, so the computing resources available will automatically increase or decrease to accommodate changes in traffic. In a real-world setting, the pre-processing steps that were performed on our training data set would need to be built in to the model pipeline so that it could receive raw MRI's as input, but for now the group can successfully submit MRI's to the hosted model and generate predictions. The model is also able to accommodate new versions, so if future retrained versions of the model achieve better performance, the model can be updated without disrupting service.

3 Results

3.1 Baseline and DFNN Model Results

After retraining the baseline 3D-CNN model on the augmented dataset, we saw a very slight improvement in accuracy, from 80.44% to 86.99%. However, we did see a significant increase in AUROC, which improved from 80.03% in the original model to 90.51% in the model trained on the augmented dataset. It is worth noting that the accuracy for the original model is a poor indicator of model performance, since the original training dataset was skewed and the model was over-predicting the majority class.

All three of our DFNN models were able to match and slightly exceed the AUROCs reported by Mellema both with and without GAN-produced data. We also selected the highest-performing DFNN architecture and repeated the same process, retraining it on the dataset with the newly produced GAN samples. Our highest DFNN AUROC score was approximately 0.83 which was obtained on testing data that did not include GAN-produced data while using the highest performing dense network architecture provided in Figure 6, with modifications. The accuracy of the DFNN model decreased slightly from 83.05% to 82.12%, while its AUROC increased slightly from 79.73% to 81.69%.

Table 1: Model Performance Comparison

Model	Accuracy	AUROC
6 height*Baseline CNN Model - Original Dataset*	86.44%	80.03%
Baseline CNN Model - Augmented Dataset	86.49%	90.51%
Best DFFN Model - Original Dataset	83.05%	82.13%
Best DFFN Model - Augmented Dataset	79.73%	81.69%

4 Conclusion

4.1 Discussion

The implications and possible applications of the results achieved by our models are not clear. While the models did fairly well on metrics and outperformed some of the reported results in Mellema 2022 we do not believe there is a clean cut answer as to whether it would be possible to solely rely on the results of any MRI classification model for ASD. ASD is truly a very complex disorder and the brain is still not well understood. The conclusion that can be made from the results of our models and training is that GANs could play a major role in refining state-of-the-art models. The data produced by a good GAN architecture would address problems such as difficulty in obtaining MRI brain images and diagnostic inequalities such a gender differences.

Our initial hypothesis was correct in the most simple interpretation.The AUROC scores of the majority of our models exceeded those reported in Mellema 2022's bench marking. However, it is important to note that the data we used was only a portion of what was used in the benchmark study. The group also acknowledges the lack of robust evaluation metrics on the data produced by our GAN model. Given limitations on time and resources the group was not able to develop the GAN as much as we would have liked to. Therefore, results from our models evaluated after the addition of GAN produced data should not be used as benchmarks but intermediary steps to guide further model training on bothe the predictive models and GAN model.

4.2 Further Research

With more time, we would like to explore RefineGAN and 3D-GAN-superresolution which have been the subject of papers focused on increasing the fidelity of GAN generated MRI images. Specifically, RefineGAN outperformed DAGAN, ReconGAN, and KIGAN in MRI reconstruction [8] and 3D-GAN-superresolution improved upon SRGAN to generate high resolution MRI images [12]. We would also like to explore DCGAN and AEGAN which have been applied on other 3D datasets such as smallNORB[5]and StyleGAN which has achieved impressive results on facial feature generation[6]. Last but not least, to simply our workflow, we would like to consider Conditional GAN through which we can control the class we generate[11].

In terms of methodology and experimental design, we would like to first and foremost use KerasTuner to scale our hyperparameter tuning. We would also like to code additional image fidelity metrics such as normalized mean square error (NMSE), Visual Information Fidelity (VIF) and Frechet Inception Distance (FID) score. Although we did not come across this in our literature review, we would like to take a second look and search for opportunities to conduct transfer learning. This is ideal for our situation where we have low sample size and storage and memory limitations. If transfer learning is not a possibility, we would like to explore methods to solve our storage and memory limitation through external hosting and distributing computing. That way we can load additional ABIDE data (up to 2000 samples), augment our data through techniques such as rotating, adding Gaussian noise, and sharpening and blurring. We theorize these methods should help us along our way of improving our GAN.

5 Member Contribution

The members in our group are Chelsea Alvarado, Dylan Howe, and Ellen Yu. All three group members split the work evenly, especially regarding to model testing where we each tackled at least one model architecture. Each team member took responsibility for all work related to their model architecture which includes, code, testing, and any writing in the final report and presentation. Chelsea has contributed to the initial literature search, data preprocessing and organization, and model testing by working on the DFNN models from Mellema [9]. Dylan took charge on being able to access and download data model testing our baseline 3D-CNN model, and model deployment. Ellen found additional literature, located data and applied for access in addition to working on the GAN model. Each member is responsible for writing up on their models they have worked on and tested. This not an exhaustive list of each member's tasks but it covers the main tasks taken on by each member.

References

[1] *Overview* (2013). ABIDE Preprocessed. Retrieved March 12, 2022, from **http://preprocessed-connectomes-project.org/abide/**

[2] Centers for Disease Control and Prevention. (2022, March 2). Data & statistics on autism spectrum disorder. *Centers for Disease Control and Prevention*. Retrieved from **https://www.cdc.gov/ncbddd/autism/data.html**

[3] Centers for Disease Control and Prevention. (2022, March 2). Screening and Diagnosis of Autism Spectrum Disorder. *Centers for Disease Control and Prevention*. Retrieved from **https://www.cdc.gov/ncbddd/autism/screening.html**

[4] Géron, A. (2022). Chapter 17. Representation Learning and Generative Learning Using Autoencoders and GANs. In *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd Edition). O'Reilly Media, Inc. **https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ch17.html**

[5] Harsh. (2020, June 26). Applying Generative Adversarial Network to Generate Novel 3D Images. **https://medium.com/analytics-vidhya/applying-generative-adversarial-network-to-generate-novel-3d-images-ba70e1176dac**

[6] Karras, T., Laine, S., &Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv* **https://doi.org/10.48550/arxiv.1812.04948**

[7] Lai, M.C., Lombardo, M.V., Ruigrok, A.N., Chakrabarti, B., Wheelwright, S.J., Auyeung, B., Allison, C., & Baron-Cohen, S. (2012). Cognition in males and females with autism: Similarities and differences. *PLoS ONE, 7*(10). **https://doi.org/10.1371/journal.pone.0047198**

[8] Lv, J., Zhu, J., &Yang, G. (2020). Which GAN? A comparative study of generative adversarial network-based fast MRI reconstruction. *Phil. Trans. R. Soc. A*379: 20200203. **https://doi.org/10.1098/rsta.2020.0203**

[9] Mellema, C.J., Nguyen, K.P., Treacher, A. & Montillo, A. (2022). Reproducible neuroimaging features for diagnosis of autism spectrum disorder with machine learning. *Scientific Reports, 12*, 3057 (2022). **https://doi.org/10.1038/s41598-022-06459-2**

[10] Mostafa, S., Yin, W., and Wu, F.X. (2019). Diagnosis of autism spectrum disorder based on Eigenvalues of brain networks. *IEEE Access 7*, 128474–128486. **https://doi.org/10.1109/ACCESS.2019.2940198.**

[11] Paul, S. (2021, July 13). Keras Documentation: Conditional GAN. In *Keras*. **https://keras.io/examples/generative/conditional_gan/**

[12] Sánchez, I., & Vilaplana, V. (2018). Brain MRI super-resolution using 3D generative adversarial networks. *CoRR* abs/1812.11440 **https://doi.org/10.1098/rsta.2020.0203**

[13] Xu, T., Yang, Z. Jiang, L., Xing, Xiu, X.X., & Zuo, X.N. (2015) A Connectome Computation System for discovery science of brain *Science Bulletin, 60*(1), 86-95. **https://doi.org/10.1007/s11434-014-0698-3**

[14] Yin, W., Mostafa, S., & Wu, F. (2021). Diagnosis of Autism Spectrum Disorder Based on Functional Brain Networks with Deep Learning. *Journal of Computational Biology, 28*(2), 146-165. **https://doi.org/10.1089/cmb.2020.0252**

[15] Zunair, H. (2020, September 23). Keras Documentation: 3D image classification from CT scans. In *Keras*. **https://keras.io/examples/vision/3D_image_classification/**