International Journal of Psychosocial Rehabilitation, Vol. 23, Issue 01, 2019
ISSN: 1475-7192

Computer Aided System for Autism Spectrum Disorder Using Deep Learning Methods

K. Sairam, J. Naren, Dr.G. Vithya and S. Srivathsan

Abstract--- *The aim of this study is to apply machine learning algorithms to identify autism spectrum disorder (ASD) patients from brain imaging dataset, based only on brain activation patterns. ASD is a brain-based disorder normally characterized by repetitive and social behaviors but in the present study imaging data from a world- wide multisite database known as ABIDE (Autism Brain Imaging Data Exchange) is used for classification. A deep learning method that combines supervised and unsupervised machine learning method has been employed to do the process. Input is based on the respective neural patterns of functional connectivity using resting state functional magnetic resonance imaging (rs- FMRI) present in pre- processed ABIDE dataset from which associativity matrix is calculated between different regions of the brain which show an anti-correlation of brain function between anterior and posterior areas of the brain. Extracted features are then subjected to the pr e-training stage along with phenotypic information. Finally, the pre-trained weights are given as input to a Convolutional neural network and classifies as ASD or control type.*

Keywords--- *fMRI, Deep Learning, Resting State, Autism.*

I. INTRODUCTION

Autism Spectrum Disorder is a brain-based disorder characterized by social deficits and repetitive behaviour. According to a recent Centre for Disease Control study, ASD affects one in 68 children in the United States. The main aim of psychiatric Neuroimaging research is to identify objective biomarkers that may inform the diagnosis and treatment of brain- based disorders. Data rich deep learning methods are a promising tool for studying the replicability of patterns of brain function across larger, more heterogeneous datasets. The primary goal of the current study is to classify autism spectrum disorder (ASD) patients and control participants based on their neural patterns of functional connectivity using resting state functional magnetic resonance imaging (rs-fMRI) data. Both supervised and unsupervised machine learning (ML) methods was used together to create a novel deep learning method. This method was applied to a large population sample of brain imaging data, the Autism Imaging Data Exchange I (ABIDE I). The secondary goal of this study is to study the neural patterns associated with ASD that contributed most to the classification. The results are then evaluated in the light of networks of brain regions that differentiate ASD from controls and of previous studies of ASD brain function.

ASD is associated with a range of phenotypes that vary in severity of social, communicative and sensory motor deficits. Currently, diagnostic instruments only assess the characteristic social behaviors and language skills of

K. Sairam, B. Tech Computer Science and Engineering, School of Computing, SASTRA Deemed University, Thanjavur, India. E-mail: susheelkums@gmail.com
J. Naren, Assistant Professor, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India. E-mail: naren.jeeva3@gmail.com
Dr.G. Vithya, Professor, School of Computing, KL University, Vijayawada. E-mail: vithyamtech@gmail.com
S. Srivathsan, B. Tech Information and Communication Technology, School of Computing, SASTRA Deemed University, Thanjavur, India. E-mail: srivats64@gmail.com

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019

autism patients. Yet studies have proved that neuro- scientific research can help bridge the gap between a clear mapping of the complexity of a spectrum of alterations in autism behaviour and their neural patterns [11]. Brain imaging studies have significantly advanced the understanding of the neural underpinnings in brain-based disorders and their associated behavior. Such an example would be studies on ASD and its social and communicative deficits [3][13][12][5].The pattern identification for activation of neural connectivity in ASD and the association of these patterns with neural and psychological components contributes to the understanding in etiology of mental disorders [13][11] of not only ASD but also Schizophrenia etc.

One of the challenges in brain imaging studies of neural disorders is to replicate these studies across larger, more demographically heterogeneous datasets that reflect the heterogeneity of clinical populations. Recently, ML algorithms have been applied to brain imaging data to extract replicable brain function patterns. These algorithms can extract replicable, robust neural patterns from brain imaging data of psychiatric disorder patients [9].

II. PREVIOUS WORKS

Several research works related to autism spectrum disorder are described in this section. The segment briefly explains pertinent research on diagnosis based on fMRI data using machine learning and deep learning methods.

Plis et al. [9] used structural T1-weighted images and deep learning in order to classify patients with schizophrenia against healthy controls, using data from four different neural sites; the authors also classified patients with Huntington disease from healthy controls. [7] Trained a Deep Belief Network with 3 depths (50 hidden units in the first layer, 50 in the second layer, and 100 in the final layer). They achieved 90% classification accuracy using features extracted from three DBMs in comparison to 68% classification accuracy using raw data classified with a Support Vector Machine.

The t-Distributed Stochastic Neighbor embedding (t-SNE) [16] method was used to reduce the resultant data to a 2-dimensional version; on which results showed a linearly separable projection into patients and control.

In another approach, cross-validation was utilized to further train the machine learning model. For the left-out subjects in the cross-validation fold, the actual value of each connection was subtracted from the estimated values obtained from the autism model and from the control model. The average of subtraction across all 7266 ROIs was computed, and average values of ROIs were summed. Positive values were classified as ASD and negative values, as controls [9] obtained as high as 60.

III. PROPOSED ARCHITECTURE

Proposed architecture consists of steps ranging from the data transformation process to the final classification of ASD and control type patients. The initial step is data transformation process and using it, a correlation analysis for different regions of brain is done followed by pre-training and classification stage. The features are passed as input to stacked de-noising auto encoder for pre-training stage. The stacked model consist of two auto-encoder structure, the first one has an input and output layer of 19900 fully connected to a bottle neck of 1000 neurons from hidden layer. The second auto encoder maps 1000 inputs from the output of the previous auto encoder to outputs through a hidden layer of 600 units.

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019

The revised encoder weights from the output of auto encoder are given as input to a CNN having the configuration: 19900-1000- 600-2. In other words, it has two hidden layers of 1000 and 600 each and an output layer of 2 units which classifies the ASD and control patients.

IV. METHODOLOGY AND APPROACH

Dataset

The ABIDE data was previously used by [6] to classify autism patients from control subjects based on neural connectivity measurements. This study reproduced an approach reported in [6] with modifications that included data sets multiple sites. Blood Oxygen level imaging signal from non-overlapping, grey matter ROIs (SPM8 mask grey.nii) formed by Voxels separated by 5 mm were computed for the 964 subjects used in the dataset. Voxels that were Euclidean close to a specific ROI's seed voxel were included in this ROI [9]and was used to compute a connectivity matrix of size 7266 × 7266, by calculating the pair wise correlation between each ROI. The ABIDE dataset consists of data collected from 505 ASD individuals and 530 matched control patients (typical controls, TC). This data was collected at 17 different imaging sites which includes rs-fMRI images, T1 structural brain images and phenotypic information for each patient.

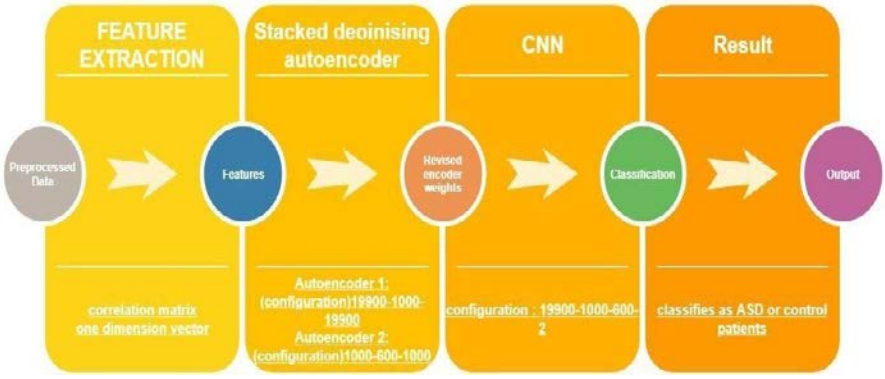

Fig. 1.4.1: Proposed architecture to classify Autism spectrum disorder

V. FEATURE EXTRACTION

To identify essential features from the dataset, functional connectivity was used to classify subjects as ASD and TC. Functional connectivity measures to an extent, an index of the strength of co-activation of brain regions based on the rs-fMRI brain imaging data. Each cell in the connectivity matrix contains a Pearson correlation coefficient. The coefficient is an index of the correlation between two areas of the brain, and it ranges from 1 to −1: values close to 1 indicate that the time series are highly correlated; values close to −1 indicate that time series are anti-correlated.

The upper triangle values were removed for use of the values in the correlation matrix as features. These values are redundant since they repeat the values in the lower triangle. Main diagonal of the matrix also was not considered, since it represents a feature correlating to itself.

Next, flattening the remaining values to retrieve a one- dimensional vector of features, with the purpose of using it for classification of subjects. The resultant number of features is 19900.

```
In [7]: dset[:100]
Out[7]:
array([-3.52649055e-02,  5.91196939e-02,  9.42654088e-02, -2.24396974e-01,
        1.52111590e-01,  5.28965235e-01,  1.30803928e-01, -3.38883139e-02,
        2.22128868e-01, -2.20051222e-02, -1.68445721e-01, -8.73610452e-02,
        8.82157619e-05,  1.45775929e-01,  4.35941696e-01,  1.31169438e-01,
        1.97372556e-01,  6.16760373e-01,  3.75993192e-01,  4.85035658e-01,
        3.65870185e-02, -9.86837372e-02,  2.84572780e-01,  3.46645355e-01,
        5.94085991e-01,  8.24268162e-02,  4.68472540e-02,  6.57934785e-01,
        9.88771468e-02, -5.00655062e-02,  1.40451133e-01, -3.24021131e-02,
        8.83430064e-01,  4.69738066e-01,  4.09579635e-01,  1.74068641e-02,
       -5.09538352e-02, -6.50372580e-02,  1.29410282e-01,  7.40173310e-02,
        7.13478982e-01,  3.80458146e-01,  3.10660064e-01,  1.29838765e-01,
        5.60217619e-01,  1.49416447e-01,  2.60009408e-01, -2.26991437e-02,
        2.69324128e-02,  9.25805718e-02, -1.11908630e-01,  1.50631964e-01,
       -1.22818481e-02,  1.77667767e-01,  9.07807872e-02,  1.92695037e-02,
        4.32944804e-01, -2.36015860e-02,  2.53631473e-01, -1.25489533e-01,
       -2.20053166e-01,  1.38053030e-01,  2.51848549e-01, -2.96582486e-02,
        1.10588642e-02,  6.72832549e-01,  2.50180483e-01,  3.16578329e-01,
        5.58229648e-02,  6.24593757e-02,  3.85323875e-02, -2.14185625e-01,
        3.75141740e-01,  1.97014377e-01,  1.50060013e-01,  1.64488833e-02,
        8.09327185e-01,  7.21142769e-01, -7.17811659e-02,  4.10412312e-01,
       -2.59993017e-01,  7.78864324e-02, -1.33946419e-01, -2.83072412e-01,
        1.03260808e-01,  2.53776759e-01, -4.49759439e-02, -8.70992020e-02,
        5.32853544e-01,  8.22135925e-01,  7.12666750e-01,  9.04630795e-02,
       -3.89052695e-03, -1.23122334e-01, -5.31029589e-02,  3.93886119e-01,
        2.82349676e-01,  2.07395121e-01, -6.87756687e-02,  3.15217555e-01])
```

Fig.1.4.2: Features extracted from pre-processed dataset

Pre-Training Stage

The De-noising auto encoders are employed to train the neural network model for superior generalization. De-noising auto- encoders reconstruct the input based on a corrupted version of the input [14]. Some positions of the vector derived from a functional connectivity matrix are set to zero before training the model.

As a first measure, two stacked de-noising auto-encoders were used for the unsupervised pre-training stage. The best optimization for validation set using mean squared error was achieved. The input and output layers have 19,900 features fully connected to a narrowed hidden layer of 1000 neurons. The probability of data corruption for the first auto encoder is set from the binomial distribution: n =1, p = 0.8 i.e 20%. The second auto encoder maps 1000 inputs from the output of the previous auto encoder to outputs through a hidden layer of 600 units. The second auto encoder corruption module is parameterized to corrupt a feature with a probability of 30% (for the binomial distribution: n= 1, p = 0.7).

Classification Stage

The encoders' weights and biases from auto-encoders are sent to a Convolutional Neural Network with the configuration: 19,900-1,000-600-2. That is, the CNN assumes an input space of 19,900 features and an output space of 2 numbers. Between the input and output layers, the network has two hidden layers with 1000 and 600 units.

The CNN maps adjusted weights based on the auto encoder weights; so, its supervised training stage is called fine- tuning. The aim of this step is to adjust the weights of the layers to output the expected classes as per our classification and minimize prediction error on the supervised task by the CNN model. The output layer contains two output units: each unit represents the probability of an input to be from an ASD or a TC subject.

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019

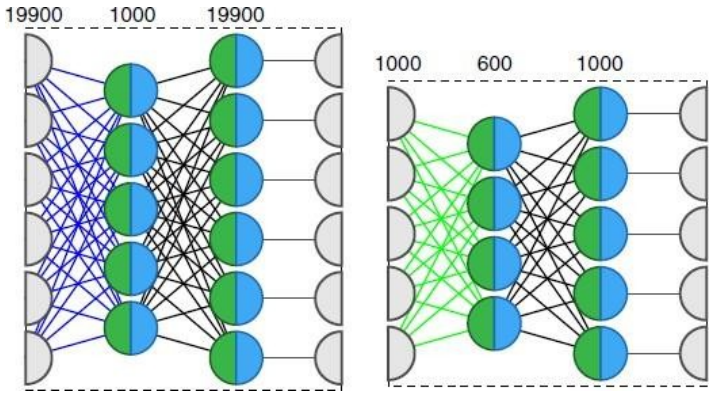

Fig. 1.4.3: Two auto encoder structure

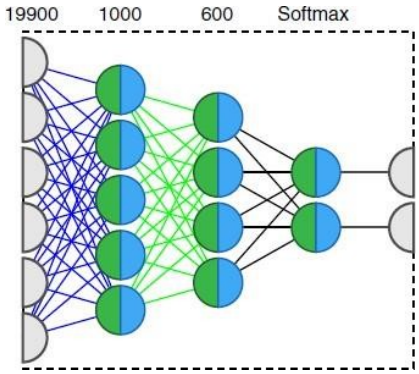

Fig. 1.4.4.: CNN structure

VI. RESULTS AND DISCUSSION

The deep neural network achieved an average classification accuracy of70% (sensitivity 74%, specificity 63%) from the cross-validation folds, and accuracy between 66% to 71% in individual folds. Based on the literature studied, this is the highest classification accuracy achieved so far. The Support Vector Machine method achieved mean accuracy of 65% (from 62% to 72%, sensitivity68%, specificity 62%); while the Random Forest classifier achieved mean accuracy of 63% (sensitivity 69%, specificity 58%). These results show that the deep learning algorithm classified ASD and typical participants above chance in the multi-site ABIDE data. Also, the algorithm outperformed other supervised methods used for comparison. Table 7.1 shows the comparison of three different methods. Figures 7.1 and 7.2 shows the pre-processed data and the features of the model respectively. Figure 7.3 and 7.4 shows the training stage in the model of auto encoder and CNN respectively. Figure 7.5 signifies the performance metrics of the model classification.

Table 1.4.5 Comparison of DNN, RF and SVM on ABIDE

Method	Accuracy	Sensitivity	Specifi city
SVM	0.65	0.68	0.62
RF	0.63	0.69	0.58
DNN	0.70	0.74	0.63

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
1.709955	-18.6006	-30.545	-12.2025	-0.16932	-11.8606	2.3353	-19.7176	-40.5061	-22.377	-26.1811	16.87008	-18.4382	-5.21786	0.365958	-1.4537	-53.3147	24.22675	-39.1632	-20.0575
-2.28681	-4.70978	5.26153	-6.90272	-4.28225	13.641	-3.54152	4.094979	-41.5014	-68.0956	-6.36487	-56.9411	3.702557	-5.91819	11.91257	-3.9129	0.423985	-22.9357	16.45637	-27.4503
0.634842	18.33627	-7.92686	-1.05804	15.34821	2.168905	-13.378	9.76074	16.44909	15.72471	-11.6923	8.206583	-21.9846	26.3872	11.19927	2.111359	-13.6577	2.397193	-5.75734	-0.73743
-0.05911	8.277509	2.853271	4.474924	7.681545	-3.78407	-23.6425	2.239799	-0.33281	15.21194	6.575445	-13.9504	-8.59438	0.08619	-7.36176	-5.90594	-38.9878	9.118493	5.258573	-10.1765
-0.31664	-6.84576	-17.5777	9.782219	-1.73539	7.931989	-30.6702	14.27662	23.62996	-1.21842	-10.9486	-3.54056	-9.48019	-2.63482	-2.9672	4.555738	-24.6396	-34.236	-6.64036	2.100058
-3.31731	-35.5047	-10.5287	-6.70903	-15.1937	-11.3512	-32.9624	-8.88179	-27.1279	-0.16812	-10.2983	-41.2678	-2.99228	-36.1742	-11.9911	1.171583	14.32489	0.693116	11.6833	-8.26069
1.953711	2.786171	11.14501	-10.0442	-6.86833	-3.93484	24.76491	-11.6529	63.74718	31.69774	-4.09407	11.89527	5.055577	-2.88614	-0.32691	-0.09396	13.6564	-2.6665	-5.39171	-3.74889
0.726157	-6.19079	3.202301	17.41125	-15.7342	24.65051	20.87327	23.12186	37.87867	24.99199	5.034314	-8.4529	24.30359	2.220273	-5.27644	-13.3778	31.18535	-20.1258	-5.50246	1.262952
0.74906	7.974062	-1.90935	-7.04161	6.895888	25.76349	24.93183	9.131217	0.642805	10.59608	-3.69671	30.81489	10.40862	-12.066	10.68004	-10.4667	30.52708	-1.95848	13.77592	28.82636
-3.05963	9.445894	30.12907	-9.10465	1.694819	1.677804	-6.29626	8.019389	-45.5162	3.653378	-4.03526	40.43656	11.68354	-5.20673	-5.50373	-7.76903	17.93671	31.12083	18.6698	10.99422
-0.14638	18.58356	19.83436	4.260438	3.924886	-5.54119	34.5068	-2.1023	-15.0293	4.80876	13.34202	14.91815	26.36947	-1.78959	0.144552	5.309275	35.61894	22.61428	10.71967	15.45772
1.839419	-12.6183	-6.90268	5.94906	-15.6933	-19.502	35.94517	-19.691	-17.4187	-4.53531	10.72461	21.82423	12.27536	10.48523	7.088424	6.179369	33.58747	11.37949	-21.974	13.57289
-0.58168	9.505393	-3.82863	11.40067	22.19231	-25.2485	36.20964	-14.2495	83.37837	36.58025	33.10332	9.22769	5.529482	40.93516	-1.61523	12.84474	20.89587	-2.70443	-2.76553	38.19116
0.105309	17.2795	-16.937	3.929923	14.53738	-23.2229	36.71094	-15.1327	10.31622	22.48864	24.05118	2.829299	7.377701	40.24808	1.773639	16.08552	13.94823	-4.3181	-25.2449	7.457117
3.0496	9.495319	14.47809	7.567431	3.889641	5.570867	0.348868	-2.78	64.61328	22.63506	23.56242	23.44977	2.891674	14.71724	7.97934	2.297283	10.30994	-30.0173	6.755099	4.589144
0.945131	28.46524	5.835218	-3.25221	-6.56801	-4.65669	-14.7636	12.97088	-13.9771	-8.28291	0.98883	-13.7552	-23.556	34.00985	14.16691	-2.02006	24.82577	-28.7926	15.46975	-3.23779
-0.08531	7.675781	-15.9977	-2.15877	-7.30208	-6.11462	4.363894	24.16764	8.026005	5.166481	-0.03372	-6.47224	7.269156	-7.48209	-13.9194	3.785232	6.862478	13.52227	-26.1036	0.966241
-1.24193	17.87069	-4.43221	1.005166	-8.85001	13.25725	8.685931	17.12717	30.98819	8.286637	-12.1296	-2.11232	3.483467	4.595281	-18.1476	-3.92026	-12.1834	14.16738	4.977575	-31.0391
1.215114	21.4347	5.269134	7.526977	-6.22305	8.882257	-2.64875	-32.3796	6.418147	-1.36839	-10.6287	17.33626	-7.27719	27.88941	-15.1013	-7.7746	-18.1633	11.42594	-0.70755	-14.701
-1.70373	-8.52928	-7.56481	-8.65332	-5.60624	-11.3168	-16.5811	-30.373	-57.3829	-48.9244	-18.5055	42.6702	-14.0799	9.2956	-7.3306	-23.9597	-28.8411	16.29986	-5.35359	-9.15352
-1.55993	-21.6936	12.71532	-18.1234	5.271849	1.851727	-29.0151	-13.1794	-74.8066	-23.9768	1.264224	27.78787	-6.10971	-32.0511	9.72522	-8.10339	-30.2622	-8.13213	24.34532	3.243613
-5.22734	-56.9116	2.411755	-22.4882	12.05608	15.22674	-27.9433	-7.52732	23.34416	11.997	-41.8101	14.03259	7.182266	-94.3907	2.271193	-23.2096	-44.1183	13.43763	3.593622	4.781541
-1.51065	-22.6196	23.43245	23.22948	12.82089	22.0653	-11.619	8.465143	-14.3528	-20.7173	-1.42535	-13.4839	17.88041	-47.3903	4.626741	-13.6953	21.77588	-9.59386	13.73511	52.87836
4.228799	-14.0293	10.43434	5.19073	15.1516	17.94431	-27.4309	27.06053	13.97457	-28.1047	15.67757	-8.93761	9.398231	-41.2481	16.65878	9.959646	-14.001	2.277862	13.99552	26.93122

Fig. 1.4.4: Pre-processed fMRI dataset

```
In [7]: dset[:100]
Out[7]:
array([-3.52649055e-02,  5.91196939e-02,  9.42654088e-02, -2.24396974e-01,
        1.52111590e-01,  5.28965235e-01,  1.30803928e-01, -3.38883139e-02,
        2.22128868e-01, -2.20051222e-02, -1.68445721e-01, -8.73610452e-02,
        8.82157619e-05,  1.45775929e-01,  4.35941696e-01,  1.31169438e-01,
        1.97372556e-01,  6.16760373e-01,  3.75993192e-01,  4.85035658e-01,
        3.65870185e-02, -9.86837372e-02,  2.84572780e-01,  3.46645355e-01,
        5.94085991e-01,  8.24268162e-02,  4.68472540e-02,  6.57934785e-01,
        9.88771468e-02, -5.00655062e-02,  1.40451133e-01, -3.24021131e-02,
        8.83430064e-01,  4.69738066e-01,  4.09579635e-01,  1.74068641e-02,
       -5.09538352e-02, -6.50372580e-02,  1.29410282e-01,  7.40173310e-02,
        7.13478982e-01,  3.80458146e-01,  3.10660064e-01,  1.29838765e-01,
        5.60217619e-01,  1.49416447e-01,  2.60009408e-01, -2.26991437e-02,
        2.69324128e-02,  9.25805718e-02, -1.11908630e-01,  1.50631964e-01,
       -1.22818481e-02,  1.77667767e-01,  9.07807872e-02,  1.92695037e-02,
        4.32944804e-01, -2.36015860e-02,  2.53631473e-01, -1.25489533e-01,
       -2.20053166e-01,  1.38053030e-01,  2.51848549e-01, -2.96582486e-02,
        1.10588642e-02,  6.72832549e-01,  2.50180483e-01,  3.16578329e-01,
        5.58229648e-02,  6.24593757e-02,  3.85323875e-02, -2.14185625e-01,
        3.75141740e-01,  1.97014377e-01,  1.50060013e-01,  1.64488833e-02,
        8.09327185e-01,  7.21142769e-01, -7.17811659e-02,  4.10412312e-01,
       -2.59993017e-01,  7.78864324e-02, -1.33946419e-01, -2.83072412e-01,
        1.03260808e-01,  2.53776759e-01, -4.49759439e-02, -8.70992020e-02,
        5.32853544e-01,  8.22135925e-01,  7.12666750e-01,  9.04630795e-02,
       -3.89052695e-03, -1.23122334e-01, -5.31029589e-02,  3.93886119e-01,
        2.82349676e-01,  2.07395121e-01, -6.87756687e-02,  3.15217555e-01])
```

Fig 1.4.5: Features after Pearson Coefficient Correlation

The features as shown in figure 5.2 are given as input to the deep neural network. The corresponding training of the DNN model is shown in figures 5.3 and 5.4.

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019

```
Exp=aal_whole_0, Model=ae1, Iter=  677, Cost=0.489004 0.436058 0.446818
Exp=aal_whole_0, Model=ae1, Iter=  678, Cost=0.493583 0.438228 0.447696
Exp=aal_whole_0, Model=ae1, Iter=  679, Cost=0.491020 0.436679 0.445428
Exp=aal_whole_0, Model=ae1, Iter=  680, Cost=0.490052 0.436721 0.446378
Exp=aal_whole_0, Model=ae1, Iter=  681, Cost=0.493064 0.433962 0.447571
Exp=aal_whole_0, Model=ae1, Iter=  682, Cost=0.493019 0.435562 0.447427
Exp=aal_whole_0, Model=ae1, Iter=  683, Cost=0.494723 0.437414 0.446922
Exp=aal_whole_0, Model=ae1, Iter=  684, Cost=0.494478 0.435126 0.445054
Exp=aal_whole_0, Model=ae1, Iter=  685, Cost=0.494090 0.433462 0.445231
Exp=aal_whole_0, Model=ae1, Iter=  686, Cost=0.494159 0.434846 0.447408
Exp=aal_whole_0, Model=ae1, Iter=  687, Cost=0.493354 0.436662 0.446680
Exp=aal_whole_0, Model=ae1, Iter=  688, Cost=0.489486 0.437092 0.444635
Exp=aal_whole_0, Model=ae1, Iter=  689, Cost=0.491161 0.436101 0.444982
Exp=aal_whole_0, Model=ae1, Iter=  690, Cost=0.490854 0.434224 0.447672
Exp=aal_whole_0, Model=ae1, Iter=  691, Cost=0.492472 0.435842 0.446264
Exp=aal_whole_0, Model=ae1, Iter=  692, Cost=0.489645 0.436208 0.447637
Exp=aal_whole_0, Model=ae1, Iter=  693, Cost=0.490782 0.435754 0.446149
Exp=aal_whole_0, Model=ae1, Iter=  694, Cost=0.492936 0.434605 0.444925
```

Fig 1.4.6: Training Model of Auto Encoder for pre-training

```
Exp=aal_whole_3, Model=ae2, Iter=  975, Cost=0.577750 0.576768 0.576473
Exp=aal_whole_3, Model=ae2, Iter=  976, Cost=0.577748 0.576729 0.576389
Exp=aal_whole_3, Model=ae2, Iter=  977, Cost=0.577593 0.576687 0.576348 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  978, Cost=0.577573 0.576711 0.576405
Exp=aal_whole_3, Model=ae2, Iter=  979, Cost=0.577637 0.576667 0.576381 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  980, Cost=0.577657 0.576654 0.576374 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  981, Cost=0.577594 0.576593 0.576392 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  982, Cost=0.577637 0.576562 0.576352 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  983, Cost=0.577903 0.576634 0.576358
Exp=aal_whole_3, Model=ae2, Iter=  984, Cost=0.577603 0.576578 0.576310
Exp=aal_whole_3, Model=ae2, Iter=  985, Cost=0.577492 0.576522 0.576264 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  986, Cost=0.577508 0.576545 0.576307
Exp=aal_whole_3, Model=ae2, Iter=  987, Cost=0.577557 0.576543 0.576235
Exp=aal_whole_3, Model=ae2, Iter=  988, Cost=0.577199 0.576522 0.576212 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  989, Cost=0.577711 0.576471 0.576223 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  990, Cost=0.577306 0.576498 0.576142
Exp=aal_whole_3, Model=ae2, Iter=  991, Cost=0.577296 0.576513 0.576220
Exp=aal_whole_3, Model=ae2, Iter=  992, Cost=0.577759 0.576477 0.576150
Exp=aal_whole_3, Model=ae2, Iter=  993, Cost=0.577481 0.576410 0.576208 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  994, Cost=0.577553 0.576427 0.576145
Exp=aal_whole_3, Model=ae2, Iter=  995, Cost=0.577260 0.576424 0.576092
Exp=aal_whole_3, Model=ae2, Iter=  996, Cost=0.577284 0.576381 0.576091 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  997, Cost=0.577306 0.576402 0.576139
Exp=aal_whole_3, Model=ae2, Iter=  998, Cost=0.577460 0.576367 0.576054 Saving better model
Exp=aal_whole_3, Model=ae2, Iter=  999, Cost=0.577168 0.576344 0.576065 Saving better model
```

Fig 1.4.7: Training Model of CNN for Classification

After training the model, testing against the test set and the performance metrics are shown in figure 7.5.

	index	Exp	Accuracy	Precision	Recall	F-score	Sensivity	Specificity
0	0	aal_whole	0.647036	0.649956	0.680138	0.663894	0.680138	0.612316

Fig.1.4.8: Performance metrics of the model classification

The 65% mean accuracy on the trained model in the present study improves the current state of the art technique. Previous studies so far suggests that supervised methods are far effective at classifying high-dimensional spaces in smaller population samples; deep neural networks allows to represent more complex functions, especially when used with auto encoders. The networks thus effectively reduce the dimensionality of problems having a very large feature space.

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019

VII. CONCLUSION AND FUTURE WORK

In this work, a model is proposed for classification of ASD and control type patients. Through conclusive analysis, deep learning methods that reliably classify big multi-site datasets were found. To account for multiple sites the model should be trained upon multiple additional sources of variance in subjects, scanning procedures and equipment in comparison to single site datasets [15]. Such variation adds noise to the brain imaging data that challenges the ability to draw signatures from the brain. Activation that can classify disease states; yet the achievement of reliable classification accuracy despite such noise generated from different equipment and demographics shows promise for machine learning applications to clinical datasets and for future application of machine learning in the assistance of identification of mental disorders. [9] stated that the overall assessment of classification of ASD using resting-state fMRI data thus far falls short of biomarker standards; such obstacle is not overcome in the present study. Yet, a step in the direction of more reliable results has been taken.

REFERENCES

[1] Abraham, A., et al., 2017. Deriving reproducible biomarkers from multi-site resting-state data: an autism-based example. *NeuroImage* 147, 736–745.

[2] Akshoomoff, N., Corsello, C., Schmidt, H., 2006. The role of the autism diagnostic observation schedule in the assessment of autism spectrum disorders in school and community settings. *Calif. Sch. Psychol*. 11 (1), 7–19.

[3] Altman, D.G., Bland, J.M., 1994. Diagnostic tests 2: predictive values. *BMJ 309* (6947), 102.

[4] Anderson, J.S., et al., 2011. Functional connectivity magnetic resonance imaging classification of autism Brain *134(12)*, 3739-3751.

[5] Craddock, R.C., Holtzheimer, P.E., Hu, X.P., Mayberg, H.S., 2009. Disease state prediction from resting state functional conn ectivity. *Magn. Reson. Med.* 62 (6), 1619–1628.

[6] Craddock, R.C., James, G**.**, 2012. A whole brain fMRI atlas spatial generated via spatially constrained spectral clustering. Human brain 33 (8).

[7] Fox, M.D., 2010. Clinical applications of resting state functional connectivity. Front. Syst. *Neurosci*. 4, 19.

[8] Pereira, F., Mitchell, T., Botvinick, M., 2009. Machine learning classifiers and fMRI: a tutorial overview. *NeuroImage* 45 (1), S199–S209. [9]. Plis, S.M., et al., 2014. Deep learning for neuroimaging: a validation Study. Front. *Neurosci*. 8 (August), 229.

[9] Plitt, M., Barnes, K.A., Martin, A., 2015. Functional connectivity classification of autism identifies highly predictive brain features but falls short of biomarker standards *NeuroImage*: Clinical 7, 359–366.

[10] Just, M.A., Keller, T.A., Kana, R.K., 2013. A theory of autism based on frontal-posterior under connectivity. In: Development and Brain Systems in Autism. *Psycologicy Press*, New York, pp. 35–63.

[11] Kana, R.K., Keller, T. a., Cherkassky, V.L., Minshew, N.J., Just, M.A., 2009. Atypical fronto-posterior synchronization of theory of mind regions in autism during mental state attribution. Soc. Neurosci. 4 (2), 135–152.

[12] Jordan, M.I., Mitchell, T.M., 2015. Machine learning: trends, perspectives, and prospects. 349 (6245).

[13] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising auto encoders. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103.

[14] Nielsen, J.A., et al., 2013. Multisite functional connectivity MRI Classification of autism ABIDE results. *Front. Hum. Neurosci*. 7 (September), 1–12.

[15] Maaten, L.V.D., Hinton, G., van der Maaten, L.H.G., 2008. Visualizing data using t-SNE. J. *Mach. Learn. Res*. 9, 2579–2605.

Received: 16 Nov 2018 | Revised: 25 Dec 2018 | Accepted: 10 Jan 2019