UNIVERSITY OF TEXAS ARLINGTON

CSE 6363: Project Final Review

<Mihir Ingole, Eldho Joy, Dannasri Srinivasan>

Classification of Autism Spectrum disorder using computer vision with deep learning and the ABIDE dataset

- Autism spectrum disorder (ASD), a type of neurological disorder, appears in children between 6 and 17 years of age and affects communication skills and social behavior.
- ASD affects social interactions and communication and causes repetitive behaviors in patients
- Functional magnetic resonance imaging (fMRI) is used to study the brain and its structures

ABIDE Dataset – Downloading and Preparation

- Downloaded the data from ABIDE(Autism Brain Imaging and Data Exchange) data release which is a public release and open sharing of neuroimaging data.
- ABIDE dataset is contributed by 17 international imaging sites. It consist of 539 individuals suffering from ASD and 573 typical controls.
- The dataset is composed of structural and resting state functional MRI data along with extensive array of phenotypic information
- phenotypic information is classified based on sex, age, and autism diagnostic observation schedule (ADOS) score for ASD subjects and mean framewise displacement (FD) quality, which is a measure of subject head motion, distributions of sex and average age at different sites for typical control (TC) and ASD patients
- User can specify the desired derivative, pipeline, and noise removal strategy of interest - then the script will download the data from FCP-INDI's S3 bucket

ABIDE Dataset – Downloading and Preparation

- Derivatives ->
  - alff (Amplitude of low frequency fluctuations)
  - degree_binarize/weghted (Degree centrality with binarized weighting/ correlation weighted)
  - func_mean (Mean preprocessed functional image)
  - rois_cc400 (400 ROI parcellation atlas)
  - rois_cc200 (Cameron Craddock's 200 ROI parcellation atlas)
- Pipelines ->
  - Cpac (Configurable Pipeline for the Analysis of Connectomes)
  - ccs (Connectome Computation System)
  - dparsf (Data Processing Assistant for Resting-State fMRI)
  - niak (NeuroImaging Analysis Kit)
- Strategies -
  - filt_global
  - filt_noglobal
  - nofilt_global
  - nofilt_noglobal

ABIDE Dataset – Downloading and Preparation

```
# And download the items
total_num_files = len(s3_paths)
for path_idx, s3_path in enumerate(s3_paths):
    rel_path = s3_path.lstrip(s3_prefix)
    download_file = os.path.join(out_dir, rel_path)
    download_dir = os.path.dirname(download_file)
    if not os.path.exists(download_dir):
        os.makedirs(download_dir)
    try:
        if not os.path.exists(download_file):
            print('Retrieving: {0}'.format(download_file))
            request.urlretrieve(s3_path, download_file)
            print('{0:3f}% percent complete'.format(100*(float(path_idx+1)/total_num_files)))
        else:
            print('File {0} already exists, skipping...'.format(download_file))
    except Exception as exc:
        print('There was a problem downloading {0}.\n Check input arguments and try again.'.format(s3_path))

# Print all done
print('Done!')
```

- Command line code

  python download_abide_preproc.py -a -d c400 -p cpac -s filt_global

  -o /path/to/local/download/dir -gt 2 -lt 30 -x M -t Caltech

Where S3_path will be https://s3.amazonaws.com/fcp-indi/data/Projects/ABIDE_Initiative

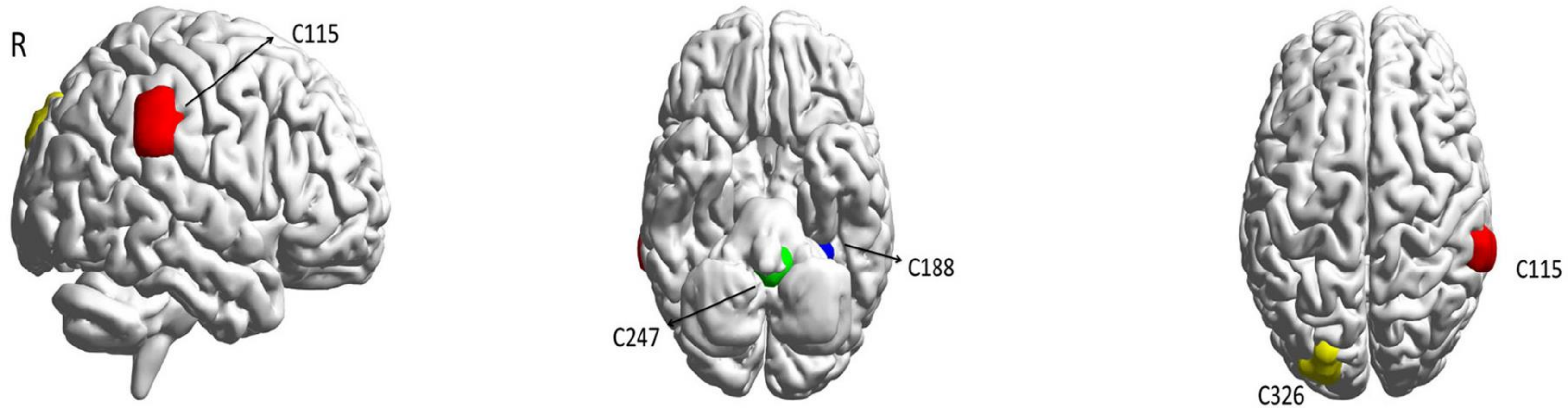
ABIDE Dataset

- We used the CC400 functional parcellation atlas of the brain.
- In this atlas, a brain connectivity matrix is constructed for the average time series of the ROI, partitioned into 400 regions.
- There are many different parameters in MRI imaging, including voxel size, flip angle, TR, TE, and T1

	Voxel size (mm^3)	Flip angle (deg)	TR (ms)	TE (ms)	T1 (ms)
CALTECH	1	10	1,590	2.73	800
CMU	1	8	1,870	2.48	1,100
KKI	1	8	8	3.7	843
LEUVEN	0.98× 0.98×1.2	8	9.6	4.6	885.145
MAX MUN	1	9	1,800	3.06	900
NYU	1.3 × 1.3	7	2,530	3.25	1,100
OHSU	1	10	2,300	3.58	900
OLIN	1	8	2,500	2.74	900
PITT	1.1 × 1.1 × 1.1	7	2,100	3.93	1,000
SBL	1	8	9	3.5	1,000
SDSU	1	45	11.08	4.3	NA
STANFORD	0.86 × 1.5 ×0.86	15	8.4	1.8	NA
TRINITY	1	8	8.5	3.9	1060.17
UCLA	1 ×1× 1.2	9	2,300	2.84	853
UM	1.2 × 1 × 1	15	250	1.8	500
USM	1 ×1× 1.2	9	2,300	2.91	900
YALE	1	9	1,230	1.73	624

ABIDE Dataset

The most important ROIs for ASD classification in the prediction model according to the saliency map. *Red, Blue, Green, and Yellow areas corresponding to (61.9; −36.3; 34.4), (−27.6; −40.2; −17.6), (−2.1; −43.0; −40.7), (−22.5; −85.5; 31.0), respectively*



ABIDE Preprocessing

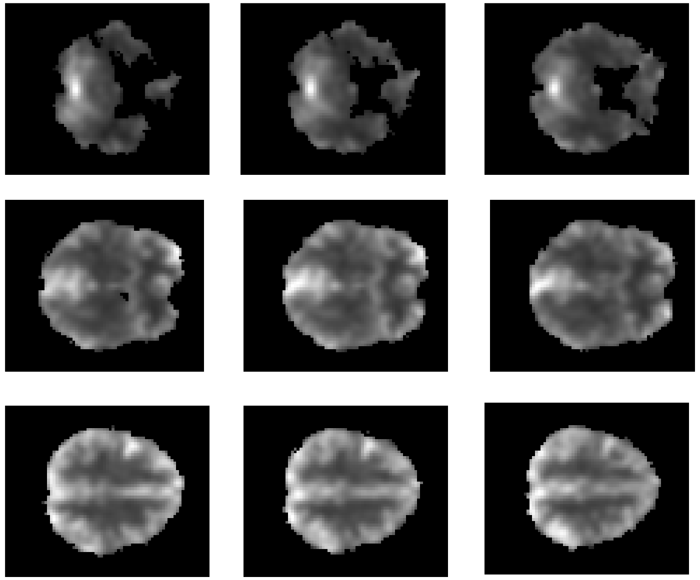
- fMRI data after Band Pass Filtering and Global Signal Regression.

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
1.288359	-22.4637	-4.98757	-6.03065	10.37785	1.099065	-0.34758	-3.25409	-8.92825	2.448179	5.587133	-0.57148
1.530402	-30.9086	7.205127	-2.87108	6.870298	5.38872	2.895479	-0.20633	-12.0516	-2.59444	6.907773	-0.11248
-4.30062	-24.7335	15.9552	-1.26083	1.010922	6.43349	-0.90197	1.319136	-8.90797	-1.27667	3.478904	1.800717
-9.19304	-11.1652	12.95361	-3.17527	-0.73286	2.194801	-11.6234	0.641052	-2.35353	6.654646	-4.00238	2.892987
-4.41202	-4.31407	1.934714	-6.96939	3.418279	-3.82601	-21.2113	-2.38271	2.406439	13.12303	-12.6136	1.506189
7.861188	-7.40124	-5.24468	-8.87516	8.91448	-6.06465	-20.1839	-8.01453	3.204292	10.9138	-18.8281	-1.14904

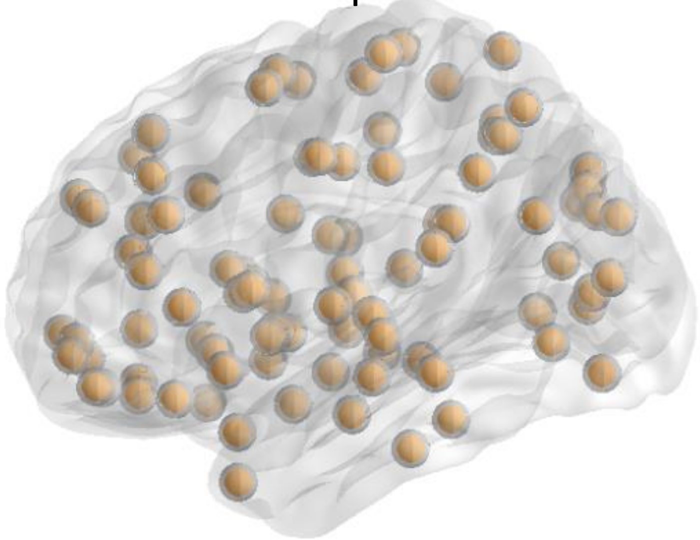
```
abide_loaded[0].shape
```

```
(196, 392)
```

ABIDE Dataset



CC400 functional parcellation atlas



- After the preprocessing, we obtained 871 quality MRI images with phenotypic information
- The Pearson correlation coefficient (ranges from -1 to 1) - index between two areas of the brain regions, with 1 representing high correlation between the two areas of the brain and vice versa.
- 392 * 392 matrix is found in the CC400 functional parcellation atlas for each subject.

Creating Correlation Matrices using PCC

Pearson Correlation Coefficient

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

```
correlationMartices[0].shape
```

```
(392, 392)
```

```
correlationMartices[0]
```

```
array([[ 1.        , -0.18410496, -0.16604868, ...,  0.10782889,
         0.13257605,  0.08509509],
       [-0.18410496,  1.        ,  0.34760289, ..., -0.10660081,
         0.0901855 , -0.23478802],
       [-0.16604868,  0.34760289,  1.        , ..., -0.29897388,
         0.02292012, -0.04428938],
       ...,
       [ 0.10782889, -0.10660081, -0.29897388, ...,  1.        ,
         0.25247047,  0.33777588],
       [ 0.13257605,  0.0901855 ,  0.02292012, ...,  0.25247047,
         1.        ,  0.20485758],
       [ 0.08509509, -0.23478802, -0.04428938, ...,  0.33777588,
         0.20485758,  1.        ]])
```

Network Architecture

- symmetric matrix which contains connectomes or functional connectivity information shows the correlation between the mean values of the time series obtained from an ROI (region of interest)
- Each cell in the matrix contains a Pearson correlation coefficient, and each row is the representation of the ROI.
- The functional connectivity matrices between pairs of ROI are fed as input to convolutional layers.
- Final -> 1 fully connected hidden layer and each linear layer followed by a tanh activation function. The parallel filters with dimensions from 1×392 to 7×392 act on rows representing the brain regions.

Network Architecture

- So we have taken into account 400 filters of length 1 and width 392–400 filters of length 7 and width 392.
- The sizes of the weights are equal to the representation matrix in the convolutional neural network.
- The hidden layer followed by max-pooling is used to reduce the number of features and avoid the overfitting problem The whole result set obtained is passed to the multilayer perceptron (MLP) to complete the classification.
- the output node is concated and fully connected to a dense layer, which is subsequently used for classification.

Input Layer
392 * 392
Conv 2D
392 * 400
Conv 2D
391 * 400
Conv 2D
390 * 400
Conv 2D
389 * 400
Conv 2D
388 * 400
Conv 2D
387 * 400
Conv 2D
386 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Max Pooling
1 * 400
Concat
7 * 400
Dropout
Keep rate = 0.25
FC
2800 * 1
Softmax
2 * 1
Input Layer
Conv 2D
Max Polling
Concat
Dropout
FC
Softmax

Model Implementation

```
model.summary()
```

```
Model: "model"

__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 392, 392, 1  0           []
                                )]

 conv2d (Conv2D)                (None, 392, 1, 400)  157200      ['input_1[0][0]']

 conv2d_1 (Conv2D)              (None, 391, 1, 400)  314000      ['input_1[0][0]']

 conv2d_2 (Conv2D)              (None, 390, 1, 400)  470800      ['input_1[0][0]']

 conv2d_3 (Conv2D)              (None, 389, 1, 400)  627600      ['input_1[0][0]']

 conv2d_4 (Conv2D)              (None, 388, 1, 400)  784400      ['input_1[0][0]']

 conv2d_5 (Conv2D)              (None, 387, 1, 400)  941200      ['input_1[0][0]']

 conv2d_6 (Conv2D)              (None, 386, 1, 400)  1098000     ['input_1[0][0]']
```

Model Implementation and Differences

```
max_pooling2d (MaxPooling2D)    (None, 1, 1, 400)    0          ['activation[0][0]']

max_pooling2d_1 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_1[0][0]']

max_pooling2d_2 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_2[0][0]']

max_pooling2d_3 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_3[0][0]']

max_pooling2d_4 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_4[0][0]']

max_pooling2d_5 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_5[0][0]']

max_pooling2d_6 (MaxPooling2D)  (None, 1, 1, 400)    0          ['activation_6[0][0]']

concatenate (Concatenate)       (None, 7, 1, 400)    0          ['max_pooling2d[0][0]',
                                                                 'max_pooling2d_1[0][0]',
                                                                 'max_pooling2d_2[0][0]',
                                                                 'max_pooling2d_3[0][0]',
                                                                 'max_pooling2d_4[0][0]',
                                                                 'max_pooling2d_5[0][0]',
                                                                 'max_pooling2d_6[0][0]']

dropout (Dropout)               (None, 7, 1, 400)    0          ['concatenate[0][0]']

flatten (Flatten)               (None, 2800)         0          ['dropout[0][0]']

dense (Dense)                   (None, 400)          1120400    ['flatten[0][0]']

dense_1 (Dense)                 (None, 2)            802        ['dense[0][0]']
```

```
=====================================
Total params: 5,514,402
Trainable params: 5,514,402
Non-trainable params: 0
_____________________________________
```

Model Hyperparameters

- Activations: tanh and softmax
- Loss: Sparse Categorical Crossentropy
- Optimizer: Adam
- Learning Rate: 0.005
- Batch Size: 32
- Epochs: 300
- Strategy: 10-fold cross-validation

Challenges and workaround (What we did additionally)

```python
#https://debugah.com/%E3%80%90tensorflow%E3%80%91internalerror-failed-copying-input-tensor-21264/

from keras.backend import set_session
from keras.backend import clear_session
from keras.backend import get_session
import gc

# Reset Keras Session
def reset_keras():
    sess = get_session()
    clear_session()
    sess.close()
    sess = get_session()

    try:
        del classifier # this is from global space - change this as you need
    except:
        pass

    print(gc.collect()) # if it does something you should see a number as output

    # use the same config as you used to create the session
    config = tf.compat.v1.ConfigProto()
    config.gpu_options.per_process_gpu_memory_fraction = 1
    config.gpu_options.visible_device_list = "0"
    set_session(tf.compat.v1.Session(config=config))
```

```
---------------------------------------------------------
ResourceExhaustedError                    Traceback (most
~\AppData\Local\Temp\ipykernel_23568\1288803584.py in <mod
     28     labels_test = np.array(labels_test)
     29
---> 30     model.fit(x=data_train,y=labels_train,batch_si
     31     accuracy_metrics.append(model.evaluate(data_te
```

Accuracy and Comparison.

```
25/25 [==============================] - 4s 157ms/step - lo
3/3 [==============================] - 0s 71ms/step - loss:
Final model accuracy:  0.5816092014312744
```

References	Best method	Performance (%)		
		Specificity	Sensitivity	Accuracy
Nielsen et al. (2013)	Multiple bins and leave- one-out classifier	58.00	62.00	60.00
Parisot et al. (2017)	Graph Convolutional Networks (GCN)	–	–	69.50
Dvornek et al. (2017)	LSTM32	–	–	66.80
Parisot et al. (2018)	Graph Convolutional Networks (GCN)	–	–	70.40
Aghdam et al. (2018)	Deep belief Network (DBN)	32.96	84.00	65.56
Xing et al. (2018)	CNN with element-wise filters (CNN-EW)	70.40	66.44	66.88
Kazeminejad and Sotero (2019)	Deep learning and PCA	65.00	67.00	66.00
Sharif and Khan (2019)	Multi-Layer Perceptron (MLP) and Feature Selection	–	–	56.26
Abraham et al. (2017)	SVC-l1 and SVC-l2 Networks	–	–	67.00
Heinsfeld et al. (2018)	SVM	62.00	68.00	65.00
Heinsfeld et al. (2018)	Deep Neural Networks (DNN) and transfer learning	63.00	74.00	70.00
Present study	CNN	61.00	77.00	70.20

Accuracy on best fold:

```
0.5632184147834778
0.6436781883239746
0.5632184147834778
0.5862069129943848
0.4712643623352051
0.6781609058380127
0.5517241358757019
0.6206896305084229
0.5057471394538879
0.6321839094161987
```

References

- Automated Detection of Autism Spectrum Disorder Using a Convolutional Neural Network [Zeinab Sherkatghanad, Mohammadsadegh Akhondzadeh, Soorena Salari, Mariam Zomorodi-Moghadam, Moloud Abdar, U. Rajendra Acharya, Reza Khosrowabadi and Vahid Salari.]
- Identification of autism spectrum disorder using deep learning and the ABIDE dataset [Anibal Sólon Heinsfeld, Alexandre Rosa Franco R. Cameron Craddock, Augusto Buchweitz, Felipe Meneguzzi]
- Multisite functional connectivity MRI classification of autism : ABIDE results [JaredA.Nielsen, Brandon A. Zielinski, P. Thomas Fletcher, Andrew L. Alexander, NicholasLange, ErinD.Bigler, Janet E. Lainhart and Jeffrey S. Anderson]

Thank You!