Classifying Adolescent Excessive Alcohol Drinkers from fMRI Data

Yong-hun Kim (ykim9), Cindy Liu (cliu15), Joseph Noh (jnoh2)

Abstract—Excessive alcohol drinking impacts the structural development of brain in adolescents, but its impact on the functional activity or connectivity of the brain has not yet been explored. Our goal is to design a classification model to predict if a subject is a heavy drinker based on their resting-state fMRI data (stored as blood oxygen-level dependent (BOLD) signals). Logistic regression of pre-processed data was used as a baseline for CNN/RNN-based models and SVMs. Surprisingly, using derived features and including demographics with logistic regression yielded far better results than applying the simple, processed data to complex models. Code: https://github.com/jnoh4/CS229ToShare or bit.ly/CS229Code

Index Terms—machine learning, computer vision, fMRI, alcoholism

1 Introduction

Existing studies have shown that excessive alcohol drinking can impact the normal structural development of brain anatomy during adolescence [1] [2]. However, it remains unclear whether adolescent binge drinking can impact the functional activity or connectivity of the brain. A preliminary attempt has been made on the NCANDA (National Consortium on Alcohol and Neurodevelopment in Adolescence) dataset to analyze functional networks of hundreds of adolescent participants based on statistical hypothesis testing. However, such analytical models tend to overfit the population, thereby producing false-positive findings [3].

In this project, our goal is to design a classification model to predict if a subject is a heavy drinker based on their resting-state fMRI data (stored as blood oxygen-level dependent (BOLD) signals). After pre-processing, the inputs to our models are parcellated fMRI data as BOLD signals, as well as patient demographic information (age, sex, scanner type). We then used each model (logistic regression, SVM, deep learning) to output a predicted classification of the patient as a heavy drinker versus non-heavy drinker.

2 Related Work

As more and more neuroimaging databases become publicly available, machine learning models are becoming increasingly useful in functional neuroimaging classification. Over the past decade, there have been several attempts to leverage machine learning on fMRI data to classify neurodegenerative diseases or different tasks. These fMRI classifications are often compared to traditional manual classification methods using clinical behavioral data, such as the DSM-IV criteria for psychological disorders.

2.1 SVM and Linear Classifiers

The earliest experiments we found mostly relied on support vector machines (SVMs) or linear classifiers, which achieved accuracies between 69-92%. Chanel et al. used SVMs for classifying austistic spectrum disorder (ASD) from both task-based and resting-based fMRI [4]. Wang et al. performed regression on fMRI scans for depression classification [5]. Yoon et al. used LDA on task-based fMRI scans for schizophrenia classification [6]. Yadav et al. compared the performance of SVMs, Naive Bayes, and neural networks for classifying Alzheimer's disease (AD) from resting-state fMRI, noting that a CNN LeNet5-based model achieved 96.5% accuracy [7]. These experiments show promise for using machine learning models towards fMRI classification, but SVM and linear classifier performance is often still limited by current neurobiological knowledge of these diseases.

2.2 Recurrent Neural Networks

Recent and current techniques for classifying on fMRI data seem to take more advantage of recurrent neural networks (RNNs), which naturally lends itself to time-series data. Chen and Hu developed a RNN-based model that was able to identify individuals by their fMRI "functional brain fingerprints" with up to 94% accuracy [8], demonstrating the ability of RNNs to distinguish between differing brain activation patterns. Dvornek et al. utilized an RNN-based model for autism spectrum disorder classification on resting-state fMRI, achieving 70% accuracy [9].

2.3 Alcohol Abuse and Our Problem

For alcohol abuse classification, little work has been done with machine learning models on fMRI data, with most analyses being done statistically or using basic regression models [10].

These papers vary on their feature selection and validation methods. Based on the results presented by these works, a mixed CNN+RNN-based approach seemed promising due to the nature of our data, which consists of resting-state fMRI only. As a result, we chose to build a

CNN+RNN-based model to classify heavy drinkers, as well as compare the performance against other neural network-based models as well as regression and SVM-based models.

3 Dataset and Features

3.1 Data

Our original dataset consists of fMRI scans from 715 adolescents/young adults from the NCANDA database. The scans measure the BOLD signal from each brain region per second (over T = 269 timesteps, 2.2 seconds/frame). Preliminary visualization of the data, such as time-series plots of individual patients (Fig. 1) show that most BOLD signals were within the same general range of values.

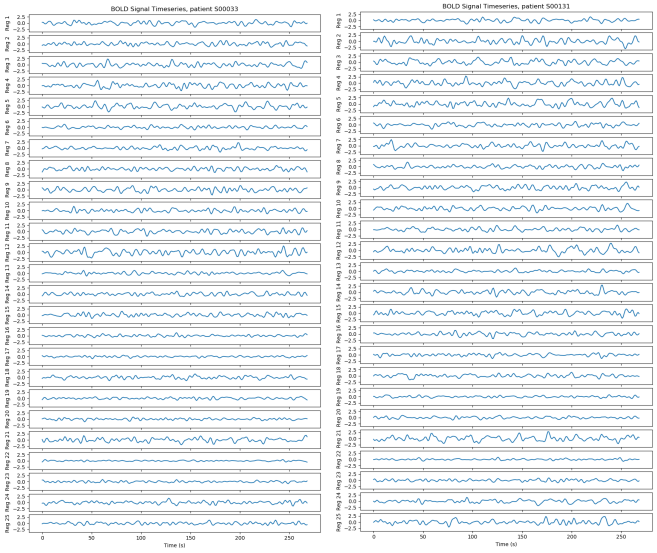

Fig. 1. Example time-series plots from two patients with ICA parcellation. a) female non-heavy drinker b) male heavy drinker.

Most BOLD signal histograms (Fig. 2) were approximately normal; however, we did see some patient BOLD signal distributions that were not normal or skewed, suggesting differences between brain regions.

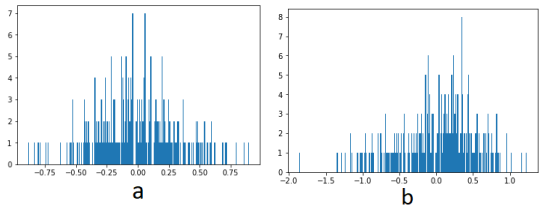

Fig. 2. BOLD Signal Histograms. a) Normal b) Skewed

3.2 Pre-processing and Derived Features

Fundamentally, BOLD signals were normalized by Z-score to reduce variability between patients. Moreover, there was significant class imbalance within the dataset (122 (17%) heavy drinkers out of 715). After setting class balance (50/50), our dataset size was reduced to 244 patients.

Because we were limited by the size of our dataset, it was imperative for us to make smart use of our data. Because neural networks have a high tendency to overfit, we needed a proper dev set in addition to train/test. We took a random sample of 5% of our dataset (13 taken, 231 leftover) and set it aside as our dev set, which will not be used during k-fold cross validation of the test/training data. We decided on 5% of the dataset because we wanted to keep this set size small but still enough to not be overly skewed towards one class (based on the cumulative probability of the binomial distribution, likelihood of getting 3 or less of a single class is .17). For 10-fold validation, the remaining 231 samples were split 90/10, resulting in 208 training and 23 test samples.

Train	Test	Dev	**Total**
208	23	13	244

fMRI scans were split into regions in one of two ways: using Independent Component Analysis (ICA) into N = 25 brain regions, or using Craddock parcellation [11] into N = 100 brain regions.

Parcellation Method	Num regions (N)
Independent Component Analysis (ICA)	25
Craddock Parcellation[3]	100

Fig. 3. Craddock Parcellation examples [11]

The derived features we used for logistic regression, reduced the time series data for each brain region into a single point. For each brain region in each patient, the entire time series was reduced to its signal range (max signal - min signal). In the code, this process was boiled down to:

$$x_{derived}(N) = \max(x_N) - \min(x_N)$$

Other features included demographic information (sex, age, and scanner type).

4 Methods

To classify heavy drinkers versus non-heavy drinkers, we implemented a series of models based on related experiments found in the literature. For our baseline, we utilized logistic regression. For our main exploration of the project, we experimented with deep learning models. We briefly attempted support vector machines as a foil for deep learning.

4.1 Logistic Regression

Logistic regression, a common binary classification algorithm, utilizes the sigmoid function (also known as the logistic function). Incorporated with linear prediction parameters, θ and the features of x, the classification prediction is given by the following probability distribution:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

whose log likelihood is given by the following:

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})$$

Due to our low number of features (25 when using derived features from ICA parcellated data), we used Newton's method for convergence. Newton's method requires the Hessian of the loss with respect to the features to be calculated, which is impractical for high dimensional data. For fewer features, Newton's method has the benefit of converging quickly, which also allows us to use batch gradient ascent. Newton's method update rule is given by the following:

$$\theta := \theta - H^{-1}\nabla_\theta l(\theta)$$

4.2 Support Vector Machines

Support vector machines (SVMs) map a given set of features to a higher dimensional space so that nonlinear classifications can be made. As opposed to logistic regression, which minimizes functional margin defined by the following equation:

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b)$$

support vector machines seek to minimize the geometric margin, which is defined by the following equation:

$$\gamma^{(i)} = y^{(i)}\frac{w^T x^{(i)} + b}{||w||}$$

In doing so, the convergence of the algorithm takes the norm of the parameters into account, and essentially, the parameters become invariant to random, meaningless scaling. This is important in allowing the parameter change to be small enough for the algorithm to converge appropriately.

As various SVMs have been used on fMRI classification models in the literature, we ran 4 SVMs with different kernels: 1) Linear, 2) Polynomial (degree 2), 3) Sigmoid, and 4) Radial Basis Function (RBF).

4.3 Deep Learning

Finally, we chose to primarily use deep learning in our most promising model, as we saw an analogy between our data, image-processing and natural language processing.

4.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are used frequently in image-processing to recognize patterns that can be anywhere throughout the picture [12]. Moreover, images have RGB "channels", which are taken into account in CNNs. Finally, the results from the channels our summed and outputted into "filters" (each with its own matrix) that theoretically detect a particular, overall "feature" in the image [12].

In our case, the fMRI data is a time-series data for different brain regions. We are trying to recognize patterns of brain activity at any time point throughout the time series along multiple "channels" (brain regions).

We tried using 1-D convolution (each region as an input channel, time series for convolution). Formally, the equation for computing 1-D convolution looks like the following:

$$o[m,i] = b[m] + \sum_{di,n} w[m,n,di] * x[n, i+di]$$

where o represents the output of the convolution, m represents the m^{th} filter output in the convolution, i represents the axis for time series, b represents the bias term in the convolution, w represents the weight matrix associated with a given output filter, x represents the input data and n represents the n^{th} input channel (brain region in this case). (Note: convolution above assumes stride = 1)

4.3.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are used frequently in natural language processing [13], because sentences carry information through words and their potential relations to each other–both short and far. The same applies to our fMRI data. The potential relations *between* convolutions may carry information that is important for the classification of heavy drinkers versus non-heavy drinkers using fMRI data. More specifically, we use a type of RNN called Long Short-Term Memory (LSTM) [14]. An RNN normally functions by storing "hidden states" that essentially "stores" information as the model processes data. These hidden states are updated directly with each input that comes into the RNN. However, in LSTM, additional "weighting" steps occur. A newly calculated hidden state is weighted against the hidden state calculated in the previous iteration and creates a new hidden state value. This new hidden state value is then weighted to account for its relevance for the model's predictive purposes (See Fig. 4 for a visualization of an LSTM model next to its set of key equations) [14].

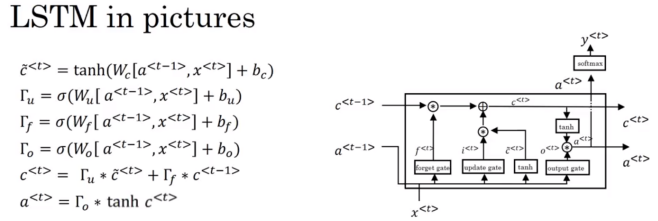

Fig. 4. An example of LSTM visualized, side-by-side with the relevant equations. [14]

4.3.3 Our Deep Learning Models

Our most promising model (Fig. 5), based on existing experiments in the literature and parallels we drew from image processing and natural language processing, consists of a CNN feeding into an RNN followed by a single-layer neural network (to represent the final logistic regression step), outputting the final classification.

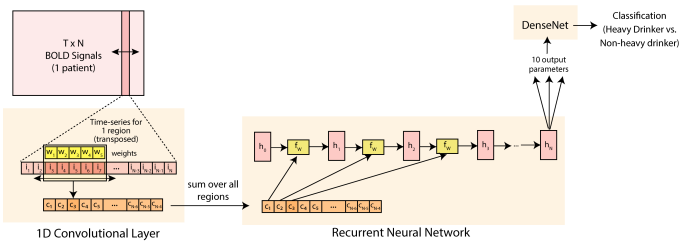

Fig. 5. CNN + RNN + DenseNet-based model

Given the sheer capacity of each deep learning model alone (RRN/CNN) as well as for comparison, we also attempted three more deep learning models: 1) RNN ONLY with a single output, 2) RNN + NN, 3) CNN + Flatten + NN.

5 EXPERIMENTS

In general, for all models, because we used 10-fold cross validation (as described in the data section) to get an accurate measurement of the measurement of our model (small dataset means cross validation is imperative), we had to make sure to set aside a dev set that will *not* be seen during cross-validation, but will be used as a means for adjust for the hyperparameters. This was the best way to properly adjust hyperparameters without being biased by the very data that we eventually train the model on.

5.1 Logistic Regression Hyperparameters

For logistic regression, we either used fMRI data in addition to all demographic information, or with all demographic information excluding age. As expected of Newton's method, the algorithm converged very quickly (1-6 iterations depending on the threshold for convergence; batch gradient descent). The criteria for convergence (c) was measured by the magnitude of change in the parameter

$$c = ||d\theta||_2^2 < \epsilon$$

where the threshold for convergence, ϵ, was determined manually using the dev set. We started with a value of 100 for ϵ and decreased ϵ by a factor of 10 and observed the effect on the average difference in accuracy between the train set and dev set over the k-fold validation (the dev set never enters the train or test set). The average difference in accuracy was invariant to the changes in ϵ (see chart below). In other words, 1 iteration was enough for meaningful convergence of the algorithm and that any additional iterations will not cause significant overfitting. Therefore, we left the value of ϵ at 1e-5, a value used in class.

ϵ	10	1	1e-1	1e-2
Avg train - dev accuracy	.16	.16	.14	.17

5.2 Support Vector Machine Hyperparameters

Using sklearn's package for SVMs [15] and changing the "tolerance" variable to control convergence of the algorithms, we carried out the same analysis as done above for logistic regression. For each SVM (linear, poly2, sigmoid and rbf), we saw that they either underfit or overfit and decided on tol = 10 (see table below). As SVMs were not the central focus of our project, we did not yet look deeper into ways of making SVM more robust for our data, but this certainly will be a future direction for us given the support it's received in literature.

ϵ	100	10	1	1e-1
Avg train - dev accuracy (L)	.12	.12	.59	.59
Avg train - dev accuracy (P2)	.12	.12	.32	.31
Avg train - dev accuracy (S)	.12	.12	.27	.27
Avg train - dev accuracy (RBF)	.12	.12	.35	.36

5.3 Deep Learning Hyperparameters

For the deep learning models, which was implemented through Keras with Theano backend [16], the number of epochs was the best way to control when to stop the training. We found that especially our CNN + RNN + DenseNet model had the tendency to underfit if we didn't run enough epochs, but also overfit if we ran too many. Therefore, we had to establish a good standard for stopping at the correct epoch in order to not underfit or overfit (more important than dropout = 0.8, which didn't have a huge impact). To that end, we established three conditions for exiting iterations. As the model was training, the model was to exit iterating if:

$$1)|TrainAcc - DevAcc| \leq \frac{TrainAcc + DevAcc}{20}$$
$$TrainAcc, DevAcc > THRESHOLD$$

(The difference in accuracy between train and dev sets is fairly small, and they both have reached a threshold value of our interest (0.6; better than random guessing))

$$2)|TrainAcc - DevAcc| \geq \frac{TrainAcc + DevAcc}{20}$$
$$TrainAcc > MAX; TrainAcc > DevAcc$$

(The difference in accuracy between train and dev sets is getting too large, the train accuracy is too high (0.65), and it is greater than the dev accuracy. Suggests that the model is beginning to overfit.)

$$3)TrainAcc > MAX; DevAcc < MIN$$

(The train accuracy is high (>0.65) while the dev accuracy is low (<0.55). Suggests that the model is beginning to overfit).

As for window size, stride and output filter dimensions for CNN; and output dimensions for RNN, we manually judged, based on the rate of training and changes in accuracy differences between the train and dev set, to have window size = 5, stride = 1, output filter dimensions = 5 and output dimensions for RNN = 5 (except for when RNN was the only layer present).

5.4 Performance Evaluation Metrics

To measure the final performance of all our models, we utilized accuracy and F1 score

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Since we balanced our classes, accuracy gave us a good indications of how the models performed. However, F1 score was a way to formally take both recall (what proportion of correct predictions was for heavy drinkers over non-heavy drinkers?) and precision (what proportion of heavy drinker predictions was correct?) into account.

6 RESULTS/DISCUSSION

n = 23	Predicted: NO	Predicted: YES	
Actual: NO	TN = 8	FP = 3	11
Actual: YES	FN = 4	TP = 8	12
	12	11	

Fig. 6. Confusion Matrix for a representative result of a single K-Fold cross validation instance for Logistic Regression (ICA)

Parcellation	Model	Train (N = 208)	Train F1 Score	Test (N = 23)	Test F1 Score
ICA	LR	0.805	0.807	0.727	0.721
	LR-Age	0.654	0.634	0.536	0.496
	C+R+N	0.644	0.661	0.501	0.541
	C+N	0.629	nan	0.524	nan
	R	0.497	0.094	0.493	nan
	R+N	0.584	nan	0.442	nan
	L	0.506	0.672	0.506	0.666
	P(2)	0.506	0.672	0.507	0.668
	S	0.506	0.672	0.506	0.668
	RB	0.506	0.672	0.506	0.668
Craddock	C+R+N	0.607	nan	0.438	nan
	C+N	0.643	0.623	0.541	nan
	R	0.526	0.463	0.523	0.438
	R+N	0.608	0.637	0.533	0.566
	L	0.506	0.672	0.507	0.667
	P(2)	0.506	0.672	0.507	0.669
	S	0.506	0.672	0.506	0.663
	RB	0.506	0.672	0.507	0.667

Fig. 7. Logistic Regression (LR); RNN (R); CNN (C); NN (N); SVM (L)inear, (P)oly 2, (S)igmoid, (RB)F; Yellow = Best model; Blue = Best model - age

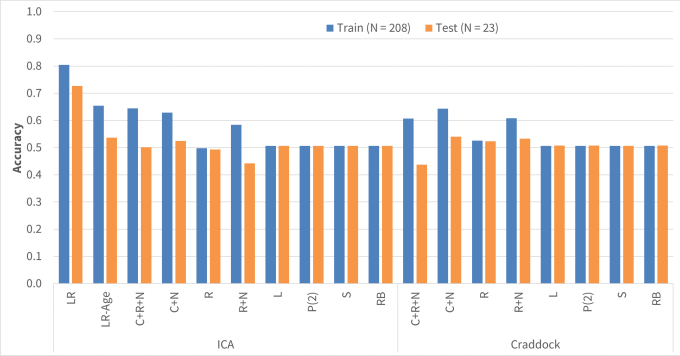

Fig. 8. Performance plotted against various models

Logistic regression with our derived features and demographics performed the best (Fig. 6, 7, 8). Surprisingly, we found that most of the correlation came from age. Upon taking out only age from the demographic information, we immediately lose the success of logistic regression and find that any form of training yields no significant benefits (slight overfitting). However, this finding is helpful in suggesting that we should integrate demographics into our deep learning models.

Generally, SVMs were not our major focus. Purely based on data we obtained while adjusting for the threshold hyperparameter, we saw that all SVMs either underfit or overfit. To prevent overfitting, we were able to adjust our threshold, but we may need to explore other hyperparameters/kernels relevant to SVMs.

The bulk of our work went into attempting deep-learning models. As we initially experimented with different hyperparameters, one of the first things we noticed was the tendency to underfit or overfit. We had to make sure to have a dev set we could test on such that we could determine window size, output filter dimension, and output dimension for RNN. We also attempted using regularization (to no avail) as well as drop-out. Our most important use of the dev set was to know when to drop after an epoch. To ensure the model trained enough, but not too much, we reasoned that the best place to stop was when both the train and dev set accuracies were fairly close and above a certain threshold, or when the two accuracies were getting too far apart. In doing so, we were able to avoid overfitting. However, all-in-all, we saw that we could not yield any significant improvements in accuracy/F1 score.

7 Conclusion / Future work

Overall, although we were able to control many of the hyperparameters to prevent under/overfitting, and adjust the number of layers, we found that deep learning models did not perform very well. This is either because our overall architecture was fundamentally flawed, inclusion of demographics is crucial, or we were simply lacking in data to be able to make meaningful distinctions (as indicated by aggressive guessing of a single class). As for SVMs, they were not our primary focus for developing our models, so we weren't able to see their full potential on the data. Nonetheless, our most preliminary work suggests they will have similar issues as our deep learning models. Although we do have our most "successful" model (baseline - logistic regression using derived features), we also found that removing age as a factor reduces its efficacy, suggesting that even logistic regression with our derived features was not any more successful than our other models.

Moving forward, there are multiple things we would like to try for our future directions. As mentioned above, we simply may not have had enough data to be able to pick up sensitive features. To circumvent this issue, we can use transfer learning, which is commonly used for model development on medical images due to relatively modest sample sizes. This involves applying learned parameters from other large datasets, ideally those that combine images and sequence data as our fMRI dataset does, and training our own final few layers to use features detected from larger data, but predicting for or own.

Another important direction is to incorporate demographics as features into our deep learning models. Although we were limited by the fact that all of us had just learned to use Keras and didn't have the time to incorporate multiple data inputs, we saw how important demographics can be. Moreover, we have not normalized for an natural, biological changes in brain function as an individual ages. Including demographics may potentially drastically change our models' performances.

Another idea would be to look into different parcellation methods for pre-processing the data (Craddock with more regions, etc.), as currently there is no consensus on the best parcellation method of fMRI data.

Lastly, we can look into making our SVM models more robust for our classification problem. For this project, we kept most of the default scikit-learn parameters (except for tolerance) as exploring SVMs was not the main focus of our project. However, fine tuning additional parameters (as in poly, sigmoid, rbf kernels) or considering additional/custom kernels may help with the issue we had of either extremely underfitting or extremely overfitting.

8 CONTRIBUTIONS

All team members contributed equally to writing this project report. Joseph Noh researched models and built the frame-work to use through Keras/Theanos. He was primarily responsible for pioneering different methods/models. Yong-hun Kim tested code for different model combinations and hyperparameters, and recorded the resulting data. Cindy Liu generated a large number of images/tables for data and result visualizations, implemented the SVM models, and created the model diagram.

We would like to acknowledge Qingyu Zhao for providing the pre-processed data and serving as a mentor for the project, giving us guidance on our set-up and answering our questions about the data. We would also like to thank TAs Atharva Parulekar and Raphael Townshend for providing guidance and advice during project office hours, as well as Professors Andrew Ng and Ron Dror for teaching CS229 this quarter.

REFERENCES

[1] L. M. Squeglia, J. Jacobus, and S. F. Tapert, "The effect of alcohol use on human adolescent brain structures and systems," *Handbook of Clinical Neurology*, vol. 125, pp. 501–510, 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780444626196000288

[2] S. Hiller-Sturmhfel and H. S. Swartzwelder, "Alcohols effects on the adolescent brainwhat can be learned from animal models." [Online]. Available: https://pubs.niaaa.nih.gov/publications/arh284/213-221.htm

[3] S. H. Park, Y. Zhang, D. Kwon, Q. Zhao, N. M. Zahr, A. Pfefferbaum, E. V. Sullivan, and K. M. Pohl, "Alcohol use effects on adolescent brain development revealed by simultaneously removing confounding factors, identifying morphometric patterns, and classifying individuals," *Scientific Reports*, vol. 8, no. 1, Dec. 2018. [Online]. Available: http://www.nature.com/articles/s41598-018-26627-7

[4] G. Chanel, S. Pichon, L. Conty, S. Berthoz, C. Chevallier, and J. Grzes, "Classification of autistic individuals and controls using cross-task characterization of fMRI activity," *NeuroImage: Clinical*, vol. 10, pp. 78–88, 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2213158215300279

[5] X. Wang, Y. Ren, and W. Zhang, "Depression Disorder Classification of fMRI Data Using Sparse Low-Rank Functional Brain Network and Graph-Based Features." [Online]. Available: https://www.hindawi.com/journals/cmmm/2017/3609821/

[6] J. H. Yoon, D. V. Nguyen, L. M. McVay, P. Deramo, M. J. Minzenberg, J. D. Ragland, T. Niendham, M. Solomon, and C. S. Carter, "Automated classification of fMRI during cognitive control identifies more severely disorganized subjects with schizophrenia - ScienceDirect." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920996412000035

[7] R. Yadav, A. Gautam, and R. B. Mishra, "Classification of alzheimer using fmri data and brain network," *Computer Science Information Technology*, 2018.

[8] S. Chen and X. Hu, "Individual Identification Using the Functional Brain Fingerprint Detected by the Recurrent Neural Network," *Brain Connectivity*, vol. 8, no. 4, pp. 197–204, May 2018. [Online]. Available: http://www.liebertpub.com/doi/10.1089/brain.2017.0561

[9] N. C. Dvornek, P. Ventola, and J. S. Duncan, "Combining phenotypic and resting-state fMRI data for autism classification with recurrent neural networks," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. Washington, DC: IEEE, Apr. 2018, pp. 725–728. [Online]. Available: https://ieeexplore.ieee.org/document/8363676/

[10] M. A. Schuckit, T. L. Smith, M. P. Paulus, S. F. Tapert, A. N. Simmons, N. J. Tolentino, and A. Shafir, "The ability of functional magnetic resonance imaging to predict heavy drinking and alcohol problems 5 years later," *Alcoholism: Clinical and Experimental Research*, vol. 40, no. 1, p. 206213, 2016.

[11] R. C. Craddock, G. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg, "A whole brain fMRI atlas generated via spatially constrained spectral clustering," *Human Brain Mapping*, vol. 33, no. 8, pp. 1914–1928, Aug. 2012. [Online]. Available: http://doi.wiley.com/10.1002/hbm.21333

[12] M. Browne and S. S. Ghidary, "Convolutional neural networks for image processing: An application in robot vision," *Lecture Notes in Computer Science AI 2003: Advances in Artificial Intelligence*, p. 641652, 2003.

[13] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, "A generalized recurrent neural architecture for text classification with multi-task learning," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[14] "Long short term memory (lstm) - recurrent neural networks." [Online]. Available: https://www.coursera.org/lecture/nlp-sequence-models/long-short-term-memory-lstm-KXoay

[15] "sklearn.svm.svc." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[16] "Keras: The python deep learning library." [Online]. Available: https://keras.io/