GROUP 37

AGE PREDICTION USING SCALP ELECTROENCEPHALOGRAPHY AND DEEP LEARNING

Dung Truong, Yayu Lin, Yijia Yan
University of California San Diego

ABSTRACT

The success of deep learning in computer vision has inspired the scientific community to explore new analysis methods. Within the field of neuroscience, specifically in electrophysiological neuroimaging, researchers are starting to explore leveraging deep learning to make predictions on their data without extensive feature engineering. One of the success is from Putten et al. (2018) which created a Convolution Neural Network to predict sex from scalp EEG with probability of more than 80%. In our project, we investigated the robustness of this and other deep learning models as applied on 24-channel EEG data from a large corpus of 1,574 participants to classify between child and adolescent. Our results suggests that deep convolutional neural networks can indeed learn discriminative features from raw EEG data to make prediction, something that traditional machine learning approaches failed to achieve.

1. INTRODUCTION

Identifying brain developmental states is always an important issue especially for young individuals. Traditional brain development evaluation methods include developmental and behavioral screening, and brain imaging evaluation. It would be very helpful to develop an accessible method to show the brain development stage based on the neuroactivity in real-time. Within neuroimaging techniques, electroencephalography (EEG) is an non-invasive and relatively low-cost method to monitor the real-time brain activity with very high temporal resolution. Therefore, the goal of this project is to predict age groups using raw EEG with deep learning to investigate the brain developmental level of children and adolescents.

Conventionally, EEG analysis and machining learning with EEG input greatly rely on feature extraction including spectral analysis and component analysis. On the other hand, a significant advantage of Deep Learning is the ability to make predictions on the data without extensive feature engineering. This project aims to train multiple deep learning and traditional machine learning models on a raw EEG dataset of more than 2000 subjects made publicly available by the Child Mind Institute Healthy Brain Network project to classify different age groups ranging from 5 to 22 years old. The input to our algorithm is a 24-channels x 256 time points (2 seconds x 128 Hz) raw EEG data. We then use a deep convolution network, 1D convolution network, EEGNet, LSTM, convLSTM, and traditional machine learning models to output a predicted age group of child (5-11 yo) or adolescent (12-22 yo).

2. RELATED WORK

To our knowledge, there are only a handful of previous work on EEG-based age prediction using machine learning are found. Within those publications, most of them are applying machine learning models instead of deep learning networks on feature-extracted EEG data. Only one paper was utilizing a deep neural network on raw EEG from sleeping patients to predict the brain age, achieving a mean absolute error (MAE) of 4.604 and a Pearson's r value of 0.933 [9]. We were not able to access the full paper which was published on 03-May-2021, after we submitted our first project proposal. For the machine learning application on EEG-based age prediction studies, Vandenbosch et al. (2019)[11] applied three machine learning models to predict brain maturational level in children and adolescents with an input of 12-channel EEG power spectra from 2667 subject and reached 95.2% accuracy using relevance vector machine; Zoubi et al. (2018)[12] applied five machine learning regression model to predict age using extracted EEG features from particular channels and frequency bands as the input with the stack-ensemble model achieving highest score of R^2 = 0.34(0.06), MAE = 6.87(0.69) and RMSE = 8.46(0.59) in years.

There a rising interest in applying deep learning on EEG dataset includes exploring the potential of utilizing deep learning to directly capture features using input of a large amount of raw EEG data without subjective interpretations. Roy et al. (2019) [1] reviewed 154 papers that apply deep learning to EEG and found that around 40% of studies used convolutional neural networks (CNNs), while 13% used recurrent neural networks (RNNs), and around a half of the studies used raw or preprocessed EEG time series as the in-

put. Our project is inspired by the work of Van Putten et al. (2018) [2], where they trained a Convolutional Neural Network (CNN) to predict biological sex on scalp resting state EEG. They achieved greater than 80% accuracy and we decided to use their model as one of the approach for age group prediction.

3. DATASET AND FEATURES

3.1. EEG recordings

The dataset, collected and made publicly available by the Child Mind Institute Healthy Brain Network project, contains resting EEG data from more than a thousand juvenile (5-22 years) participants. [3] We were able to obtain 1574 subject data from the published dataset. High-density EEG data were recorded in a sound-shielded room at a sampling rate of 500 Hz with a bandpass of 0.1 to 100 Hz, using a 128-channel EEG geodesic hydrogel system by Electrical Geodesics Inc. (EGI) [3]. The data are publicly available for download at http://fcon_1000.projects.nitrc.org/indi/cmi_healthy_brain_network. We only considered the resting data files. These were 6 minutes in length and were composed of successive 20-s to 40-s periods of eyes open and eyes closed rest respectively.

3.2. Raw data preprocessing

Although DL may be applied to raw EEG data without any preprocessing [1], we minimally preprocessed the data following the practice in [2] using EEGLAB v2021 [4] running on MATLAB 2020b. We used only eye-closed data segments ($\approx$ 170s per subject), ignoring the first and last 3 seconds of each eye-closed period (resulting in five periods of 34 seconds). We removed the mean baseline for each data epoch from each channel, down-sampled the data to 128 Hz, and subsequently band-pass filtered the data between 0.25–25 Hz (FIR filter of order 6601; 0.125 Hz and 25.125 Hz cutoff frequencies (-6 dB); zero phase, non causal). Data were re-referenced to the averaged mastoids and cleaned using Artifact Subspace Reconstruction EEGLAB plug-in cleanrawdata (v2.3) [5], an automated method that removes artifact-dominated channels and portions of data (parameters used were 5 for FlatLineCriterion, 0.7 for ChannelCriterion, and 4 for LineNoiseCriterion). Removed channels were then interpolated using 3-D spline interpolation (EEGLAB interp.m function). No bad portions of data were removed. While our recordings have 128 channels data, the comparison study [2] used only 24 channels. From the 128 available channels, we thus selected, by visual inspection of the overlaid channel maps, the 24 closest channels to the montage used in [2]. Finally, we segmented eye-closed data periods into non-overlapping 2-s windows: each preprocessed 2-s epoch was used as a sample for our final dataset. Each subject provided about 81 2-s samples (mean 80.8 ± 3.32). Each sample in our dataset thus had dimension 24x256 (24 channels and 2(s) x 128(Hz) time points). The 128-Hz down sampling and 2-s window length were identical to those used in [2]. No bad epochs were removed. No further preprocessing was performed for learning from the raw data.

Following [2], we split our dataset into training, validation, and test sets in size ratio 60:30:10. Each segment received a binary label, indicating a child (0) or a adolescent (1). This gave 71,300 samples (885 participants; 61% child) for training, 39,868 samples (492 subjects; 63% child) for validation, and 16,006 samples (197 subjects; 55% child) for testing.

4. METHODS

We explored different Deep Learning models that have previously been shown to work well with either raw EEG data or time-series data. We also performed a baseline test using traditional machine learning models. Summaries of these models are detailed in sections below. For a detailed view of the architecture of each model we experimented, see the Appendix.

4.1. Putten's model

We attempted to reconstruct the network architecture from the original paper [2] but faced challenges as the details necessary to replicate the network were in some cases missing and in others inconsistent. Here we describe our best effort at replicating the model. Each of the first 4 CNN layers is followed by a max pooling layer then a dropout layer using a 25% dropout rate. The output of each convolutional and fully connected (FC) layer (except the last) was transformed by a Rectified linear unit (ReLU). The classification layer is a 2-unit FC layer with a softmax activation, resulting in a probability p for child or adolescent ($p < 0.5$ for child and $p \geq 0.5$ for adolescent). The number of the network trainable parameters was 12,713,934.

4.2. Conv1D

One-dimensional CNN has been applied successfully to time-series data for the task of human activity recognition using accelerometer and gyroscopic data [6]. 1D convolution convolves signal only on one dimension, in our case the time dimension. The filter size matched with the input size along the height dimension, in our case 24 channels. We thus developed a model Conv1D that includes two 1D CNN layers, followed by a dropout layer for regularization, then a pooling layer. Drop out was added to address the common issue of overfitting in Deep Learning.

4.3. EEGNet

EEGNet is a deep learning model that has been shown to perform well across various Brain Computer Interface (BCI)

tasks [7]. The model comprises of variations of convolutional layer types, including Depthwise convolution [10] and Separable convolution, which is a Depthwise Convolution followed by Pointwise Convolution [10]. The Depthwise convolution apply a 2-d depth filter at each depth level of the input sample. Separable convolution is then a Depthwise convolutional operation with the height and width dimension separated. The argument made for this model was that the model was able to learned meaningful representations from raw EEG signal and performed well across BCI tasks while having very small number of parameters (compared to most other deep learning model). In fact, our implementation of EEGNet for our dataset only resulted into 766 trainable parameters.

4.4. LSTM

Long-short Term Memory (LSTM) is a branch of the Recurrent Neural Network (RNN) architecture that features backward connections between LSTM cells in addition to the common feed-forward connections. This characteristic makes LSTM networks adapt at learning sequences of data of arbitrary length, hence its name. LSTM is widely used in time-series forecasting, which fits the nature of our task when the EEG data is essentially a 24-channel time-series signal.

Furthermore, we look into ConvLSTM, a variant of the LSTM base model with predecessing convolutional layers to extract features from raw input before feeding into the LSTM layers. Our implementation of ConvLSTM model features 4 Conv1D layers to condense the input features on each channel on the time axis, and one LSTM layer to learn the sequence of features. This model has 563,714 parameters in total, with 563,330 trainable.

4.5. Traditional Machine Learning

To form a baseline for performance comparison, we explore some of the traditional machine learning models. For each of these models, we used their default implementation in SciKit-Learn [8] with no further hyperparameter tuning. As in the case with our deep learning models, no hyperparameter tuning was performed in order to place emphasis on comparing different classifiers architectures. For our experiments we implemented and measured the performance of the following traditional ML models:

- AdaBoost
- Decision Tree
- Random Forest

5. EXPERIMENTS

5.1. Experiment setup

All of our experiments were run on a computing cluster equipped with a single NVIDIA V100 SMX2 GPU (32GB). Our software environment consists Python 3.7.10, PyTorch 1.3.1, Tensorflow 1.4.1 with Keras and SciKit [**?**] libraries. The validation data were used to assess the models' performance during training and inform stopping rules. Where applicable, we apply a AdaMax optimizer with default configuration (learning rate $= 0.002, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$) so that our models are optimized for efficiency and still comparable on the same standard. Additionally, we set the training batch size to 256 to train each of our model for at least 50 epochs before evaluating their prediction results.

5.2. Evaluation metrics

Per-sample prediction accuracy was reported for all models (Table 1). We obtained more detailed performance results for our best performing models, Putten and Conv1D. Following [2], we also obtained a final performance estimate for the test dataset by taking the mean gender probability pave of the first 40 2-s samples for each subject; if probability > 0.5, the subject was classified as 1 (adolescent) or as 0 (child) otherwise. We refer to this as per-subject performance. We also computed precision, recall, and F1-score using formula below. Table 2 below shows detailed performance results for these two models.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP, FP, and FN stand for True Positive (# of correctly classified adolescences), False Positive (# of child wrongly classified as adolescences), and False Negative (# of adolescences wrongly classified as child).

5.3. Experiment results

We first look at the binary classification accuracy on the testing set of the various models we tested on. The Putten model achieved the highest testing accuracy of 74.37%, followed closely by the Conv1D model with 71.45% accuracy. Among the traditional machine learning methods serving as baseline models, the Random Forest model exhibited optimal performance with 65.72% accuracy, with simple Decision Tree model ranking low at 60.88% accuracy. Notice that both the EEGNet and ConvLSTM models showed sub par performance at 55.24%, which is curiously is the exact proportion of one class in the test set. The actual results agreed with our concern that both models indeed generated all predictions on the same class. With further investigation, we found that those models experienced severe under-fitting during training. Specifically, their loss curve only decreased during the initial epoch then has remain unchanged at a high level no matter how many additional epochs trained. The root cause of

Models	Test Accuracy
Putten	**74.37%**
Conv1D	71.45%
EEGNet	*55.24%*
ConvLSTM	*55.24%*
AdaBoost	64.41%
Decision Tree	60.88%
Random Forest	65.72%

Table 1. Testing accuracy of trained models

Metrics	Putten	Conv1D
Per-segment Accuracy	**74.37%**	71.45%
Per-segment Precision	75.24%	**78.29%**
Per-segment Recall	**63.72%**	50.5%
Per-segment F1	**69%**	61.39%
Per-subject Accuracy	**80.71%**	75.13%
Per-subject Precision	84.72%	**89.8%**
Per-subject Recall	**69.32%**	50%
Per-subject F1	**76.25%**	64.23%

Table 2. Various metrics on the performance of Putten and Conv1D models

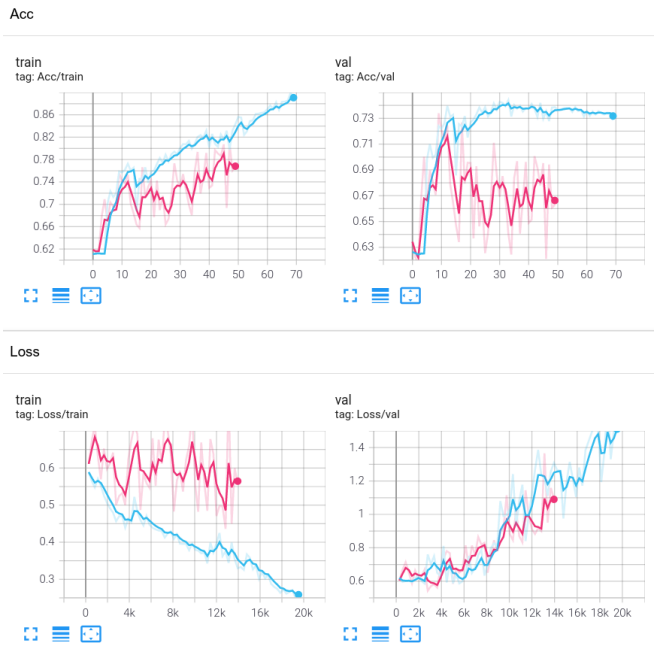

Fig. 1. Accuracy and loss curves during model training (Putten in pink, Conv1D in cyan)

this issue remains curious and should be explored for future researches.

We further analyze the learning curves of the best performing Putten and Conv1D models as seen in Fig. 1. We notice a common pattern of decreasing training loss against increasing validation loss, with validation accuracy quickly converging at a level considerably lower than the training accuracy. Said characteristics indicates clearly of significant over-fitting of the training set. We also notice that in the long run, the converged Conv1D model outperforms the Putten model in both accuracy and stability. One common solution is early stopping, in which training is stopped when validation and training loss start to diverge. Thus we saved intermediary models during training and chose the version with best validation accuracy to evaluate on test set at the end.

From the results of Table 2, with high precision and low recall, we can see that Conv1D is conservative in classifying a positive example (adolescent). On the other hand, Putten model strike a better balance between precision and recall, as reflected clearly by the F1 score as compared to Conv1D. We suspect that this is due to Conv1D being shallower and thus adding more layers might allow this model to perform better.

6. CONCLUSION

Our project contributes a new insight of applying deep learning models on raw EEG dataset without feature extractions to predict the brain age group of child and adolescent. We focused on several deep learning approach including a deep convolution neural network inspired by an CNN for image recognition, an 1D convolution network, EEGNet, and convLSTM. We also applied traditional machine learning models include decision tree and AdaBoost to compare the performance. Within all models we tested, the deep convolution network has the best performance with 74.37% test accuracy, followed by Conv1D model with 71.45% test accuracy.

We suspected Putten model to perform well as it strikes a balance between the model complexity and regularization. The model has 6 convolutional layer, making it deep, while having dropout in between each of those conv layers, making it less prone to overfitting right away. Meanwhile, for LSTM models, the complexity with low regularization could be the reason why the model immediately overfitted and stopped learning. The EEGNet model, although sound in theory, has too few parameters which could be why it's underperforming. We are intrigued by the ability of Conv1D to perform relatively well, even when it's not as deep as Putten model. Overall, that LSTM models underperfom comparing to CNN models for time-series data is intriguing. This goes against the common assumption that LSTM is always better fit for sequential data.

For future work, we would like to apply different visualization techniques on the trained CNN models to see what discriminative features from the raw EEG that the models have learned. For scientific endeavors, explainability often triumph predictions and being able to generate scientific hypohtheses automatically from data is the ideal for scientific discovery in the age of big data.

7. CONTRIBUTIONS

Within our group each member dedicated their effort to the team project and made fair contributions.

Dung Truong: Data preparation and preprocessing, models training and evaluation, paper write-up.

Yayu Lin: Data preparation, literature review, paper write-up.

Yijia Yan: Model research, design, implementation and evaluation, report write-up and formatting.

8. REFERENCES

[1] Roy, Y., et al., Deep learning-based electroencephalography analysis: a systematic review. Journal of Neural Engineering, 2019. 16 051001.

[2] Van Putten, M. J. A. M., Olbrich, S. and Arns, M., Predicting sex from brain rhythms with deep learning. Scientific Reports, 2018. 8.

[3] Alexander, L., Escalera, J., Ai, L. et al., An open resource for transdiagnostic research in pediatric mental health and learning disorders. Scientific Data, 2017. 4, 170181.

[4] Delorme, A. and S. Makeig, EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of neuroscience methods, 2004. 134(1): pp. 9-21.

[5] Mullen, T. R., Kothe, C. A. E., Chi, Y. M., Ojeda, A., Kerth, T., Makeig, S., et al., Real-time neuroimaging and cognitive monitoring using wearable dry EEG. IEEE Trans. Bio-Med. Eng, 2015. 62(11), 2553–2567.

[6] https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/

[7] Vernon J Lawhern et al 2018 J. Neural Eng. 15 056013

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[9] Yoav Nygate, Sam Rusk, Chris Fernandez, Nick Glattard, Jessica Arguelles, Jiaxiao Shi, Dennis Hwang, Nathaniel Watson, 543 EEG-Based Deep Neural Network Model for Brain Age Prediction and Its Association with Patient Health Conditions, Sleep, Volume 44, May 2021, Page A214, https://doi.org/10.1093/sleep/zsab072.541

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," CoRR, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357

[11] Vandenbosch, M. M. L. J. Z., van 't Ent, D., Boomsma, D. I., Anokhin, A. P., & Smit, D. J. A. (2019). EEG-based age-prediction models as stable and heritable indicators of brain maturational level in children and adolescents. Human Brain Mapping, 40(6), 1919–1926. https://doi.org/10.1002/hbm.24501

[12] Zoubi, O. Al, Wong, C. K., Kuplicki, R. T., Yeh, H. wen, Mayeli, A., Refai, H., ... Bodurka, J. (2018). Predicting age from brain EEG signals-a machine learning approach. Frontiers in Aging Neuroscience, 10(JUL), 1–12. https://doi.org/10.3389/fnagi.2018.00184

9. REPLY TO REVIEWS

9.1. Review by Group 17

Group 37 presents an interesting topic on the application of machine learning to predict age using scalp electroencephalography data. They have used the healthy brain network project's data to train, validate and test their model. They train their model to classify it as a child or adolescent. They have used the EEGLAB toolbox from Matlab to prepare the data for training.

1. They have implemented the models described in their literature survey thoroughly.
2. There is a comparison between different implemented models (Conv1D, EEGNet, Decision Tree, etc)

Question:

1. How the 24 channels data are being handled in 1D convolution and EEGNet?
2. For Putten et al. implementation are the authors using only 10 epochs to get rid of overfitting? As in the learning curve, we see final validation accuracy is around 67% for Putten et al. but for the test data the accuracy is 74%. Have authors investigated this discrepancy?
3. As this is a classification problem, have the authors tried to look into the confusion matrix

9.1.1. Reply

1. 1D convolution convolves signal only on one dimension, in our case the time dimension. The filter size matched with the input size along the height dimension, in our case 24 channels. For EEGNet, the first layer is 2D convolution as commonly known. The Depthwise convolution apply a 2-d depth filter at each depth level of the input sample. Separable convolution is then a Depthwise convolutional operation with the height and width dimension separated.
2. As can be seen from training vs. validation plot Putten et al. model overfitted very quickly, thus continuing training after 10 epochs would only exacerbate the problem. We picked our final model based on best validation accuracy, which was also around 74%.

3. They have been included in the report with discussion.

9.2. Review by Group 22

This talk explains a supervised deep learning solution to predict age groups using raw EEG data. Predicting age based on EEG could help in understanding the brain development in children and provide methods to diagnose developmental disorders. The speakers use the "Healthy Brain Network Project" dataset to train their network to predict between two ground truth age classes. The dataset and preprocessing steps were well explained and backed by prior work. They take inspiration from a paper that uses EEG data and DL to predict human biological sex and repurpose it to solve the age classification problem. The model is compared with multiple baselines and an accuracy of 80.71% is achieved.

Strengths of the Project:

1. The goal was clearly stated and explained. The motivation for deep learning was also explained well.
2. The dataset seems to have been well explored for this project.
3. The comparison with multiple baseline models set the expectation of performance for this problem well. It also helped highlight the improvement brought by the proposed solution

Things to improve:

1. The validation loss curve seems to steadily increase. A case of overfitting would be when it decreases first and then increases. What could be the reason for the steady decrease, and how can it be overcome?
2. Majority voting was used with the Putten model to improve the classification accuracy, why was the same not done with the baseline models? In specific, Conv1D seems to perform similarly to vanilla Putten model, so it might end up performing similarly when voting is used (comparable to 80.71%)
3. Why is accuracy a good metric for this problem? Considering accuracy might usually be a weak metric if there is class imbalance.
4. Since default hyperparameters were used, was this determined to be the best choice? If not, can tuning be performed?

9.2.1. Reply

1. Validation loss for Putten et al. decreased in the first four epochs then started to increase, showing overfitting. We recognize that it's difficult to observe that from the plot. We decided to early stopped the model and chose to evaluate the model at best validation accuracy. Future tuning can be done to overcome overfitting including adding more dropout and weight decay.
2. We performed majority voting on Putten model since its test accuracy per sample is higher from Conv1D. We wouldn't expect Conv1D to outperform Putten model if it's already inferior using per-segment accuracy.
3. Those metrics have been included in the report with discussion.
4. We did not perform any hyperparameter tuning as the initial focus was on trying out different neural architecture. Further hyperparameter tuning can be absolutely performed.

9.3. Review by Group 30

The project was based on implementing a supervised deep learning solution to classify between two age groups of subjects from their EEG recordings. The project members use the "Healthy Brain Network Project" for training their network. The project members provide solutions implemented in the existing literature and do a good job of explaining the data distribution and preprocessing pipelines. They train a deep learning model based on a previous paper that used the model for classifying between the biological sex of the subjects from their EEG recordings. Using this method, they achieve an accuracy of 80They compare this model with the baselines from simpler models, to highlight the improvement in performance achieved by their model.

Some questions/possible improvements:

1. In the presentation, we see the training and validation loss curves. The validation loss curve seems to increase consistently over all the epochs, while the training loss curve decreases. We also see a saturation of accuracy values at that level. This may point to overfitting of the model. How would you avoid the same?
2. Were the default hyperparameters optimal or would tuning help further improve the model's performance?
3. Was Majority voting implemented in all the models? If not, why so? If yes, did it increase the performance of the other models as well, specifically the Conv1D model.

9.3.1. Reply

1. Validation loss for Putten et al. decreased in the first four epochs then started to increase, showing overfitting. We recognize that it's difficult to observe that from the plot. We decided to early stopped the model and chose to evaluate the model at best validation accuracy. Future tuning can be done to overcome overfitting including adding more dropout and weight decay.

2. We did not perform any hyperparameter tuning as the initial focus was on trying out different neural architecture. Further hyperparameter tuning can be absolutely performed.

3. We performed majority voting on Putten model since its test accuracy per sample is higher from Conv1D. We wouldn't expect Conv1D to outperform Putten model if it's already inferior using per-segment accuracy.

A. MODEL ARCHITECTURES

Layer (type)	Input Shape	Param #	Tr. Param #
Conv2d-1	[1, 1, 24, 256]	1,000	1,000
ReLU-2	[1, 100, 22, 254]	0	0
MaxPool2d-3	[1, 100, 22, 254]	0	0
Dropout-4	[1, 100, 11, 127]	0	0
Conv2d-5	[1, 100, 11, 127]	90,100	90,100
ReLU-6	[1, 100, 9, 125]	0	0
MaxPool2d-7	[1, 100, 9, 125]	0	0
Dropout-8	[1, 100, 4, 62]	0	0
Conv2d-9	[1, 100, 4, 62]	180,300	180,300
ReLU-10	[1, 300, 3, 60]	0	0
MaxPool2d-11	[1, 300, 3, 60]	0	0
Dropout-12	[1, 300, 1, 30]	0	0
Conv2d-13	[1, 300, 1, 30]	630,300	630,300
ReLU-14	[1, 300, 1, 24]	0	0
MaxPool2d-15	[1, 300, 1, 24]	0	0
Dropout-16	[1, 300, 1, 23]	0	0
Conv2d-17	[1, 300, 1, 23]	90,100	90,100
Conv2d-18	[1, 100, 1, 21]	30,100	30,100
Flatten-19	[1, 100, 1, 19]	0	0
Linear-20	[1, 1900]	11,679,744	11,679,744
Linear-21	[1, 6144]	12,290	12,290

Total params: 12,713,934
Trainable params: 12,713,934
Non-trainable params: 0

Fig. 2. Summary of model following [2]

Layer (type)	Output Shape	Param #	Tr. Param #
Conv1d-1	[1, 64, 254]	4,672	4,672
ReLU-2	[1, 64, 254]	0	0
Conv1d-3	[1, 64, 252]	12,352	12,352
ReLU-4	[1, 64, 252]	0	0
Dropout-5	[1, 64, 252]	0	0
MaxPool1d-6	[1, 64, 126]	0	0
Flatten-7	[1, 8064]	0	0
Linear-8	[1, 100]	806,500	806,500
ReLU-9	[1, 100]	0	0
Linear-10	[1, 2]	202	202

Total params: 823,726
Trainable params: 823,726
Non-trainable params: 0

Fig. 3. Summary of Conv1D model

Layer (type)	Output Shape	Param #
conv1d_41 (Conv1D)	(None, 24, 64)	49216
max_pooling1d_25 (MaxPooling	(None, 12, 64)	0
conv1d_42 (Conv1D)	(None, 12, 64)	12352
max_pooling1d_26 (MaxPooling	(None, 6, 64)	0
batch_normalization_7 (Batch	(None, 6, 64)	256
dropout_49 (Dropout)	(None, 6, 64)	0
conv1d_43 (Conv1D)	(None, 6, 128)	24704
max_pooling1d_27 (MaxPooling	(None, 3, 128)	0
conv1d_44 (Conv1D)	(None, 3, 128)	49280
max_pooling1d_28 (MaxPooling	(None, 2, 128)	0
batch_normalization_8 (Batch	(None, 2, 128)	512
dropout_50 (Dropout)	(None, 2, 128)	0
lstm_15 (LSTM)	(None, 256)	394240
dropout_51 (Dropout)	(None, 256)	0
dense_30 (Dense)	(None, 128)	32896
dense_31 (Dense)	(None, 2)	258

Total params: 563,714
Trainable params: 563,330
Non-trainable params: 384

Fig. 4. Summary of ConvLSTM

Layer (type)	Output Shape	Param #	Tr. Param #
Conv2d-1	[1, 4, 28, 197]	260	260
BatchNorm2d-2	[1, 4, 28, 197]	8	8
Conv2d-3	[1, 8, 5, 197]	192	192
BatchNorm2d-4	[1, 8, 5, 197]	16	16
ELU-5	[1, 8, 5, 197]	0	0
AvgPool2d-6	[1, 8, 2, 50]	0	0
Dropout-7	[1, 8, 2, 50]	0	0
SeparableConv2d-8	[1, 8, 4, 37]	192	192
BatchNorm2d-9	[1, 8, 4, 37]	16	16
ELU-10	[1, 8, 4, 37]	0	0
AvgPool2d-11	[1, 8, 1, 5]	0	0
Dropout-12	[1, 8, 1, 5]	0	0
Flatten-13	[1, 40]	0	0
Linear-14	[1, 2]	82	82

Total params: 766
Trainable params: 766
Non-trainable params: 0

Fig. 5. Summary of EEGNet