CNN models for Eye State Classification using EEG with Temporal Ordering

Femi William*
The University of Alabama in Huntsville.
Huntsville, Alabama, USA
fww0001@uah.edu

Feng Zhu
The University of Alabama in Huntsville.
Huntsville, Alabama, USA
fz0001@uah.edu

ABSTRACT

Most studies on eye states (open or closed) detection apply machine learning techniques on subject dependent eye state datasets, but subject independent data with large physiological variation between individuals has not been well explored. Temporal ordering information is important to predict eye state because EEG is a time sequence dataset. In this research, we keep the temporal ordering of the data in place. We create multiple CNN network models and select optimal filters and depth. Our CNN feature models are effective on both subject dependent and subject independent eye state EEG classifications. We got the best subject dependent result with 4 layers of CNNs with an accuracy rate of 96.51% on dataset I and 100% on dataset II. For subject's independent studies, we got the best classification accuracy of 80.47% on dataset I and we got 90.15% on dataset II.

CCS CONCEPTS

• Applied computing → Bioinformatics; **• Computing methodologies → Machine learning algorithms**.

KEYWORDS

EEG, Eye State, CNN, Models subject independent, deep learning

ACM Reference Format:
Femi William and Feng Zhu. 2021. CNN models for Eye State Classification using EEG with Temporal Ordering. In *12th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '21), August 1–4, 2021, Gainesville, FL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3459930.3471160

1 INTRODUCTION

Eye state, closed or open, at any moment may be classified using EEG (Electroencephalography). EEG is the measurement of voltage fluctuations of the brain from the scalp. Classification using EEG signals comes with some challenges because electromyogram(muscular activity) often interfere with EEG being investigated [21]. But this challenge is less common with eyes-closed and eyes-open resting conditions EEG. The applications of the human eye state classification by EEG include: epileptic seizure detection [11], infant sleep-waking state identification [3], computer games, biomedical systems, smart home device controlling, internet of things and detection of car driving drowsiness[23].

Current eye state classifications include subject dependent and subject independent analyses. Subject dependent classification uses a subject's own data to train the model. Whereas subject independent classification does not have such requirement. While EEG data from subject dependent study faces contamination challenges due to noise and muscular activities, subject independent EEG dataset do have large variation from different subject's physiology, scalp shape, thickness together with contamination from noise and muscular activities.

To address these challenges, researchers have applied preprocessing, feature extraction, and state-of-arts algorithms for classification. But deep learning models have not been well established to be successful due to limited data. In [14], Rosler et al, collected 14 channel eye state EEG data [13] from a single subject and demonstrated the performance analyses of 42 classifiers. They obtained 97.3% classification accuracy. In [20], Wang et al. applied time series classification approach to the corpus in [13] with a high error rate of 27%. These studies were based on subject dependent eye state classification. No other study that we know has considered subject independent EEG prediction.

One potential problem with EEG classification is that existing algorithms do not use temporal ordering. Temporal ordering is the chronological arrangement of events in time. Because EEG is a time sequence data, it will be helpful to keep temporal ordering while analyzing eye states.

We applied convolutional neural networks (CNNs), and tuned their hyperparameters. Specifically, we applied various hyperparameters(filters) from a 10×10 table of filter combination to generate multiple two dimensional CNN models for analyses. Then we chose the optimal hyper-parameters and depth based on classification accuracy. On subject dependent analysis, we got 96.51% on dataset I and 100% on dataset II. Subject's independent studies from one EEG device used by all subjects in dataset II showed best classification accuracy of 90.15%. But on analyzing dataset I from CNN networks trained with dataset II we got 80.47% as the highest classification result.

This paper is organized as follows. The next section discusses background and related works. Section III focuses on data acquisition and methodologies. Section IV is the presentation of results

BCB '21, August 1–4, 2021, Gainesville, FL, USA

ACM ISBN 978-1-4503-8450-6/21/08…$15.00
https://doi.org/10.1145/3459930.3471160

and discussion. Then, section V explores areas of future work. The last section concludes our paper.

2 BACKGROUND AND RELATED WORKS

Over the last few decades, researchers have investigated eye state classification algorithms and how they can be applied to brain computer interface. The following discussion focuses on eye state differentiation, subject dependent and subject independent EEG studies.

2.1 Studies on eye states differentiation

Eye state data may be analyzed by frequency analysis and ERPs. In [1], the differences between eyes-closed and eyes-open resting conditions were investigated. The difference exists in cortical processing of visual input, resulting into different level of activation between eyes-closed and eye-open conditions. Boytsova et al (2010) [2] theorized that regardless of environmental lights, eye state transition could be associated with visible changes in the EEG, a sign of the brain's activity change in response to visual stimuli. They investigated the effect of darkness over resting eye state. Under conditions of complete darkness, the two eye states ('opened' and 'closed') significantly differed in their EEG spectral power and coherence in the delta, theta, low alpha upper alpha, low beta, upper beta and gamma frequency bands. Multiple subjects resting eye state EEG signals was studied by Torkamani-Azar et al [18], where cumulative vigilance score, response time were predicted from band-power ratios of EEG signals. They found a close relationship between rest and task-related networks.

2.2 Subject dependent classification

Researches have leveraged on eye state transition to detect and predict eye states. In [14], researchers demonstrated the performance analyses of 42 different classifiers. Among all the 42 trained classifiers, KStar, an instance based learning method was the most promising with an accuracy of 97.30% from k-fold cross-validation. Their eye state corpus is now a benchmark saved at Machine Learning Repository, University of California, Irvine (UCI) [13]. Hamilton et al[5] extended the KStar technique in [14] using an ensemble learning model where Regularized Random Forest(RRF) and KStar on the same dataset applied in [14] yielded a classification accuracy of 97.4 %. Researchers in [16] and [15] independently worked on the corpus in [13] and the authors in [16] applied cross-channel maximum and minimum values to monitor real-time EEG signals in 14 channels. Then, common spatial pattern (CSP) was employed to create discriminating features for classification by logistic regression and artificial neural networks with accuracy of 83.4%. In [15], k-nearest neighbor and multi-layer perceptron neural network models were applied to the same eye state dataset classification. With a k value of 3 they got their best performance of 84.06% accuracy. Model evaluation by k instances of validation dataset alone without a test dataset may be problematic because a useful model should be able to generalize to unknown data [8]. Also, it was observed that these analysis did not keep temporal ordering of the EEG dataset. Because the dataset of [13] consist of one subject alone, these studies are limited to subject dependent classification.

2.3 Subject dependent studies and deep learning

Here we focused our reviews on EEG subject dependent studies where deep learning was fully or partly employed for classification. According to [7], more neural network layers do capture more in-variances of a dataset which could translate to better classification using the right network configurations. Zhang et al [10] evaluated the performance of models with network layers ranging from 2 to 10 on a mental workload EEG classification task. Architectures with seven layers outperformed both shallower (2 and 4 layers) and deeper (10 layers) models in terms of accuracy, precision, F-measure and G-mean. In addition, O'Shea et al [16] compared the performance of a CNN with 6 layers and shallow model like SVM on neonatal seizure classification for 18 subjects. Their results showed that, the deeper network of 6 layers presented better average accuracy of 97.1% in comparison to the shallower model(SVM) with average accuracy of 96.5%. This relationship between network depth and performance has not been shown on eye state EEG. Narejo et al[9] predicted eye state EEG datasets using deep learning architectures namely deep belief network(DBN) and Stacked AutoEncoders. But did mot demonstrate effects in performance upon parameter tuning. Whether temporal ordering of EEG was maintained remained unclear because their report did not mention their validation nor train-test data split strategy. In [17], the EEG signals of 30 normal and 30 depressed persons were analyzed using deep CNN with a focus on each hemisphere of the brain at a time. Applying the data gotten from right hemisphere they got 99.31% average accuracy after ten-fold cross validation, and the left hemisphere they obtained 96.3% from the left. In these studies, researchers decide their kernel(s) combination by previous work or personal discretion or an expert evaluation of the problem's domain to figure out what is an appropriate number of kernel(s) to use on CNN. But this approach of filter selection may be limited to using a singular value without exploring other possibly optimal filter combinations. Also, these are subject dependent studies with less variation compared to subject independent analyses.

2.4 Subject independent classification studies

Here we reviewed subject independent EEG classification studies where a subject's own data is not used to train the model but it is tested on it. In [6] subject-independent emotional labels were classified through adversarial learning with an average accuracy of 75.31% and a standard deviation of 7.33% which was even better compared to other subject independent models. The need to improve performance on subject-independent EEG studies is still wide open due to obvious EEG variations across subjects. In [12] subject independent models were used in demonstrating that such independent classification system can reliably decode two types of imagery namely happy emotional imagery and motor imagery. They recruited 27 healthy subjects for data collection and after analyses they achieved a mean classification accuracy of 75.30%. By using narrow band spectral power of the EEG signals and Support Vector Machine with ANOVA as feature selection mechanism. The authors in [22] classified affective states independent of subjects. While 64.74% classification accuracy was recorded on the 3-class Arousal problem, 62.75% was achieved on valence states classification.

There are three major recurring problems from our reviews which we want to address namely: subject dependent classification without temporal ordering in place, eye sate subject independent EEG classification and exploration of multiple filters for optimal filter combinations. Therefore, we applied CNNs and tuned their hyper parameters to generate 100 CNN models for eye state EEG classification.

3 METHODS AND MATERIALS

In this section, we present the components and strategies employed in our study. We start with a sub section on data descriptions of both datasets. Then we discuss using CNN to classify eye states. The code and downloaded dataset used is found at $https://github.com/wilie247/EEG\_CNN\_Analysis\_Eye\_state$ and the workflow for applying our technique is seen in figure 1.

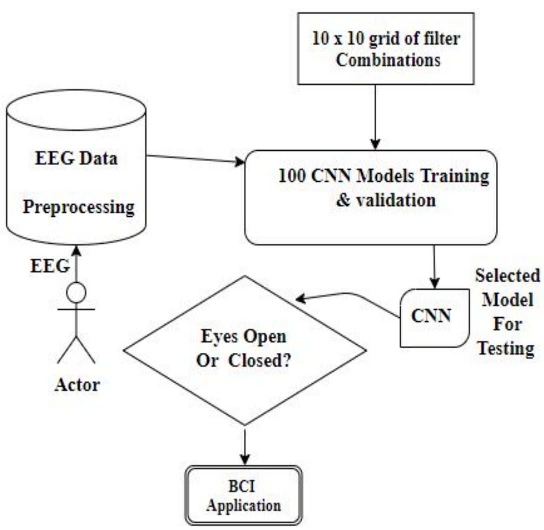

Figure 1: Work flow of our methods and materials

3.1 Data Description

There are two online EEG Eye state datasets we use: dataset I and dataset II. Dataset I is a singular subject EEG thus it is not enough for subject independent analyses.

Dataset I

Dataset I [13] was provided by Oliver Roesler from Cooperative State University, Stuttgart, Germany. It is a one subject's continuous EEG measurement with the Emotiv EPOC EEG headset. The duration of the measurement was 117 seconds. The corpus consists of 14,977 instances with 15 attributes each (14 attributes representing the values of the electrodes and the eye states). The instances are stored in the corpus in chronological order and thus preserves temporal dependencies. 8,255 (55.12%) instances of the corpus correspond to the eye open and 6,722 (44.88%) instances to the eye closed state. The eye state was detected via a camera during the EEG measurements and added later manually to the file after analyzing the video frames. The 14 channel are AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, in that order and figure 2 shows their placement according to the 10-20 International Electrode Placement System.

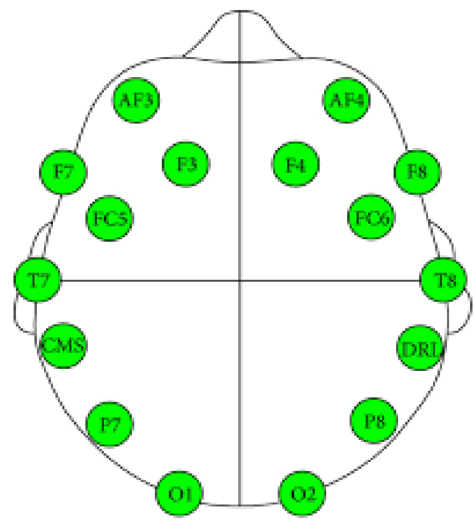

Figure 2: 14 electrodes placement with 2 reference electrodes (CMS and DRL)

Dataset II

Dataset II from [18] was a 10 subjects' eyes-open and eyes-close dataset. Each subject's dataset contains 2.5 minutes of eyes-open (EO) and 2.5 minutes of eyes-closed (EC) resting-state EEG. The EEG activity was collected at 2,048 Hz via 64 Ag/AgCl active electrodes mounted according to the 10-10 International Electrode Placement System. The raw dataset provided was down-sampled to 256 Hz from the original sampling rate of 2,048 Hz. Experiments were conducted in the early afternoon to induce drowsiness in the already idled brain networks.

3.2 Data Pre-processing

This stage involves data cleaning, data reduction, data integration, data transformation and data arrangement to make it fit for our algorithm. This is crucial because it ensures reliable and accurate results together with trading off between efficiency and time-complexity [4]. For preprocessing and analysis, we used a 64-bit window machine, Intel(R) Core(TM) i7 CPU 860 @ 2.80Hz (8 CPUs) and 16Gig RAM.

Dataset I

We extracted the labels from the 15th column of the dataset and normalized the data by z-score. To make the data suitable for the CNN algorithm, we formed a 4D matrix version of the data, whereby 1st dimension indicate row(s), second is 14 of channels data, third indicate channels(3 for rbg image, 1 for EEG), and the fourth is trial index. Thereafter we split the data to training(60%) and testing(40%) sets with temporal ordering in place. The training sets was further split-ted by Kfold cross validation where K is 10 to show what happens when temporal ordering is missing. Therefore 60% of the entire dataset was used for training while 40% of the dataset was used for testing. Moreover, a two-dimension dataset was formed to represent a trial by epoching the data to 100 milliseconds window per trial. Therefore, the 117 seconds long datasets was split-ted to epochs of 13×14 dataset each to represent a singular trial. At transition points the majority labels between eye open and closed serve as the label otherwise each trial represent eye open or eye close.

Dataset II

We extracted the same channels data on Dataset I (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) from the 64 Channels on Dataset II. We normalized the data by z-score. To make the data suitable for a CNN algorithm, we formed a 4D matrix version of the data, whereby 1st dimension indicate row(s), second is 14 of channels data, third indicate channels (3 for rbg image, 1 for EEG), and the fourth is trial index. Then we arranged the data with temporal ordering in place as 1-60% in chronological order of the data-points used for training the models and remaining 40% used for testing with temporal ordering in place. This dataset has no inflection point as the eye opened and closed signals were already separated to two different files by [18]. Two-dimension dataset was formed to represent a trial by epoching the data to 100 milliseconds window per trial. Therefore, for all subjects, each of the 2.5 minutes long datasets was split-ted to epochs of 25×14 dataset each to represent a singular trial.

For both datasets, each ith trial has a dimension $t \times c$, where t represents the time points and c is the number of channels per trial. A single trial is classified 'open' or 'closed'. The challenge is to find a hypothesis that maps a single trial data matrix to a scalar (binary) value y_i that can be used to determine the class of this singular trial event.

$$\begin{aligned} X_i &\in R_{t*c} \\ y_i &= f(X_i) \end{aligned} \tag{1}$$

3.3 Theoretical framework of applied CNN

A deep network is a hierarchical model where each layer applies a linear transformation followed by a non-linearity to the preceding layer. Specifically, let $X \in R^{n \times D}$ be the input data where each row of X is a multi-dimensional data point, D is EEG data with $d \times c$ samples per trial and n is the number of training examples. Let $W^{\mathrm{k}} \in R^{d_{k-1} \times d_k}$ be a matrix representing a linear transformation applied to the output of layer $k-1, X_{k-1} \in R^{N \times d_{k-1}}$, to obtain a d_k-dimensional representation $X_{k-1}W^{\mathrm{k}} \in R^{N \times d_k}$ at layer k. Each column of W^{k} represent a convolution with some filter or the application of a linear classifier (as in fully connected networks). Let $\psi_k : R \rightarrow R$ a non-linear activation function known as a rectified linear unit (ReLU) $\psi_k(x) = max\{0, x\}^2$. This non-linearity is applied to each entry of $X_{k-1}W^{\mathrm{k}}$ to generate the k$^{\mathrm{th}}$ layer of a neural network as $X_k = \psi_k(X_{k-1}W^{\mathrm{k}})$ [19].

$$\begin{aligned} \phi(X, W^1, ..., W^K) = \psi_k(\psi_{k-1}(.. \\ ..\psi_2(\psi_1(XW^1)W^2)..W^{K-1})W^K) \end{aligned} \tag{2}$$

3.4 Analyses on Dataset I

According to Table II analyses were done on individual layers from 1 to 4. Each layer with the same amount of models all generated from table I. The run time for classification is <0.1sec but training took from 4minutes to 1hour with respect to the number of layers.

One layer CNN models

Analyses were carried out on a set of one layer 100 CNN models generated from parameters on Table II and a grid of 10×10 filter combinations as seen in table I. Models used ReLU activation function to add non linearity, and a softmax for classification layer. The tabulated results showing the three best performing models, their filter sizes and quantity, testing accuracy, sensitivity and specificity are shown in table III and figure 3 presents the performance based on the grid.

Table 1: Kernel size and number grid

	F_{c1}	F_{c2}	F_{c3}	F_{c4}	F_{c5}	F_{c6}	F_{c7}	F_{c8}	F_{c9}	F_{c10}
F_1	1, 1	1, 2	1, 3	1, 4	1, 5	1, 6	1, 7	1, 8	1, 9	1, 10
F_2	2, 1	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2, 10
F_3	3, 1	3, 2	3, 3	3, 4	3, 5	3, 6	3, 7	3, 8	3, 9	3, 10
F_4	4, 1	4, 2	4, 3	4, 4	4, 5	4, 6	4, 7	4, 8	4, 9	4, 10
F_5	5, 1	5, 2	5, 3	5, 4	5, 5	5, 6	5, 7	5, 8	5, 9	5, 10
F_6	6, 1	6, 2	6, 3	6, 4	6, 5	6, 6	6, 7	6, 8	6, 9	6, 10
F_7	7, 1	7, 2	7, 3	7, 4	7, 5	7, 6	7, 7	7, 8	7, 9	7, 10
F_8	8, 1	8, 2	8, 3	8, 4	8, 5	8, 6	8, 7	8, 8	8, 9	8, 10
F_9	9, 1	9, 2	9, 3	9, 4	9, 5	9, 6	9, 7	9, 8	9, 9	9, 10
F_{10}	10, 1	10, 2	10, 3	10, 4	10, 5	10, 6	10, 7	10, 8	10, 9	10, 10

Table 2: Models hyperparameters

Stride	Padding	MaxEpochs	Filter size	Filter count	Layers
1	same	50	[1 -10]	[1 - 10]	[1-4]

Table 3: One Layer CNN Model Performance

Model	Filters(s, c)	Val.(%)	Test(%)	Sens.	Spec.
1	4, 4	100	93.45	0.94	0.93
2	5, 5	100	91.92	0.89	0.99
3	5, 10	99.71	91.49	0.94	0.87

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Fc1	88.87	88.43	84.93	56.11	63.32	64.6	83.2	87.77	79.9	64.63
Fc2	84.28	73.14	61.14	57.64	59.61	55.9	72.5	55.02	74.5	60.48
Fc3	70.09	88.21	56.99	63.54	61.14	71	57.6	60.26	77.7	63.54
Fc4	75.98	82.31	73.8	93.45	64.63	85.6	57.6	74.89	56.3	66.16
Fc5	76.2	57.42	67.9	62.01	91.92	57.2	68.6	56.77	64.9	76.42
Fc6	73.8	70.96	61.79	61.14	72.93	61.8	64	75.76	67.3	72.27
Fc7	83.41	77.07	79.04	71.62	76.86	76.4	86.5	63.54	87.3	71.83
Fc8	81.22	64.85	80.13	67.9	70.74	73.4	75.8	83.63	75.6	61.35
Fc9	73.14	85.59	87.56	63.97	69.21	63.5	55.7	88.43	77.3	89.52
Fc10	85.59	62.66	77.07	82.97	91.49	74.2	66.4	73.14	66.4	68.34

Figure 3: One layer Model Classification accuracies from all filter combination on dataset I test set

Two-layer CNN models

Two-layers and hyperparameters as shown in Tables I and II are applied to generate CNN models. The 100 CNN models generated used ReLU activation function to add non linearity, and a softmax for classification layer. The results showing the filter sizes, filter quantity, accuracy, sensitivity and specificity on test dataset is shown in table IV and figure 4 presents the entire model performance based on the grid where all model were generated.

Table 4: Two Layer CNN Models Performance

Model	Filters(s, c)	Val.(%)	Test(%)	Sens.	Spec.
1	2, 6	99.86	92.36	0.89	1.0
2	8, 4	99.85	91.7	0.89	1.0
3	8, 3	99.86	91.05	0.89	0.96

Three and Four Layers of CNN Models

Three and four layers networks with filter sizes and number combinations are shown in Table I, and hyperparameters as shown in Table II were used to configure CNNs for analyses. The 100 CNN models generated used ReLU activation function to add non linearity to the model, and a softmax for classification layer. Other network configuration is shown in Table II. The results showing the filter size and quantity for each model against the recorded

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Fc1	36.03	53.93	79.69	61.35	83.41	59.61	58.3	61.57	75.11	55.68
Fc2	36.03	61.35	61.14	77.07	55.46	68.12	55.9	63.54	67.25	61.79
Fc3	86.68	61.35	59.17	92.36	60.7	63.1	62.66	91.05	55.68	67.47
Fc4	65.28	60.92	56.99	58.52	67.03	58.52	53.06	91.7	80.57	78.82
Fc5	61.14	61.35	61.35	58.3	66.16	58.3	63.32	57.21	62.66	69.65
Fc6	61.35	92.36	84.06	81	57.64	74.89	60.48	60.48	87.12	60.92
Fc7	65.94	76.42	80.35	60.04	55.68	58.73	55.46	77.73	60.04	66.38
Fc8	71.83	75.76	60.26	61.57	71.18	62.45	64.85	60.7	55.02	67.9
Fc9	81.22	61.14	69.43	62.88	65.28	65.28	63.1	63.76	64.85	60.7
Fc10	61.35	63.76	72.05	73.8	82.31	68.78	63.76	66.81	55.68	55.9

Figure 4: Two layer Model Classification accuracies from all filter combination on dataset I test set

accuracy, sensitivity and specificity from test dataset is shown in Tables V and VI with respect to layers. And figures 5 and 6 present the entire model performance based on the grid from where models were generated.

Table 5: Three Layer CNN Models Performance

Model	Filters(s, c)	Val.(%)	Test(%)	Sens.	Spec.
1	4, 10	99.86	94.11	0.96	0.91
2	1, 5	99.85	92.14	0.89	1.0
3	1, 1	82.57	91.92	0.89	0.99

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Fc1	91.92	84.5	60.7	56.55	61.35	58.95	55.9	59.2	58.3	58.08
Fc2	62.01	72.71	59.61	68.78	60.26	56.55	71.62	58.1	59.83	73.36
Fc3	36.03	55.02	84.06	61.14	61.14	60.04	72.27	56.6	64.19	60.92
Fc4	66.38	90.61	61.35	68.56	56.99	63.1	60.48	88.4	84.72	65.5
Fc5	92.14	68.56	61.14	89.08	57.42	75.98	70.31	60.7	60.48	89.08
Fc6	61.14	72.93	61.14	69.21	55.24	65.28	72.71	62.5	57.42	54.8
Fc7	61.35	61.14	80.79	62.01	71.18	80.35	67.47	62	63.1	66.38
Fc8	64.41	90.83	59.17	55.02	61.79	63.97	63.76	59.2	58.73	58.52
Fc9	74.45	71.83	87.77	61.13	61.35	61.79	59.83	62.9	60.26	58.52
Fc10	76.42	88.43	84.5	94.11	60.7	60.26	61.14	61.4	59.61	69.87

Figure 5: Three layer Model Classification accuracies from all filter combination on dataset I test set

Table 6: Four Layer CNN Models Performance

Model	Filters(s, c)	Val.(%)	Test(%)	Sens.	Spec.
1	3, 10	99.85	96.51	0.95	0.99
2	5, 4	100	92.36	0.89	1.0
3	8, 3	99.71	87.77	0.96	0.77

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Fc1	36.03	36.03	59.61	77.29	86.46	58.73	52.62	80.79	61.14	60.04
Fc2	36.03	61.79	87.34	68.12	57.64	60.7	80.57	53.06	62.23	63.32
Fc3	61.57	76.2	61.14	55.24	69.43	57.64	77.07	87.77	62.88	73.36
Fc4	86.03	61.57	67.47	59.39	92.36	62.88	69.21	72.93	61.57	60.04
Fc5	36.03	80.57	59.17	56.11	59.39	60.7	60.7	55.68	65.94	57.21
Fc6	61.14	87.34	64.63	74.45	70.96	60.26	60.04	64.85	59.17	57.42
Fc7	61.57	61.14	83.84	75.55	56.55	62.66	58.95	67.03	67.25	61.57
Fc8	81.66	85.81	54.15	72.49	65.72	78.82	69.21	55.9	62.66	60.04
Fc9	65.28	61.14	61.57	60.7	61.14	57.21	52.84	62.45	57.64	61.14
Fc10	61.14	63.54	96.51	61.14	61.35	63.1	72.27	59.61	62.01	60.04

Figure 6: Four layer Model Classification accuracies from all filter combination on dataset I test set

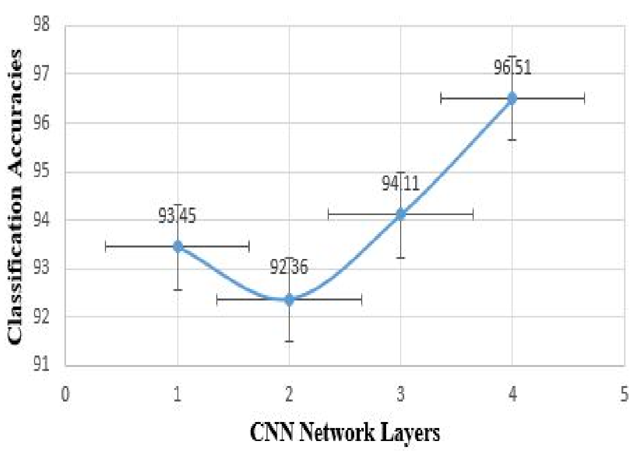

Figure 7: Classification accuracy against CNN network layers

3.5 Subject dependent analyses on Dataset II

The dataset from each subject was divided into training and test set for analysis. The network configuration applied to these datasets is described in Table VII. The filter combinations as seen in table I was applied for 100 CNN models generation. Four layers network performed best on dataset I, thus, only four-layer CNN was configured on dataset II. The top three models based on accuracy, sensitivity and specificity for the 10 subjects are described in tables VIII.

We noticed there were multiple models with similar results, and whenever we encountered such results, we selected models based on least filter size, or least filter counts. We chose the least filter combination size because they took less time to train. The run time for classification is <0.1sec but training took about 2hour for each subject.

Table 7: Models hyper-parameters

Stride	Padding	MaxEpochs	Filter size	Filter count	Layers
1	same	50	[1 - 10]	[1 - 10]	4

3.6 Subject independent analyses on Dataset II

Due to the large neural variability associated with subject independent studies, models would require large training set to enhance prediction. The data sets were applied using 80% and 20% train-test data split, the dataset from Subjects 1 to 8 was used to train 100 CNN models generated from the configuration in table VII and filter combination in table I. These models were applied to test the eye state signal datasets from Subjects 9 and 10 which were previously unseen by the models.

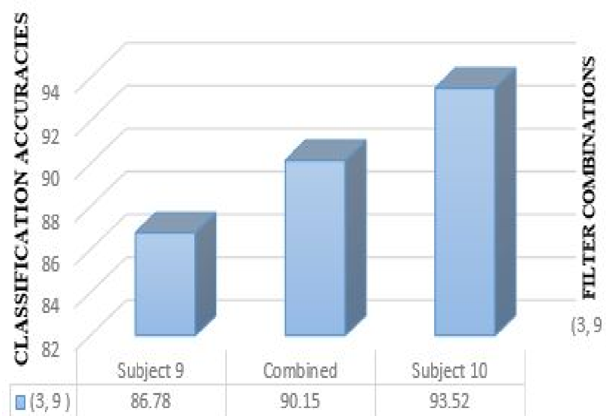

Figure 8: Topmost model performance from independent analysis

4 RESULTS AND DISCUSSION

The grid of Table I and hyperparameters from Table II provided the parameters from generating 100 CNN models. Upon analyses the best is chosen. We observed that within filter size ranges from 1 - 3, optimal models are seen when a training dataset is large. But with small training datasets, it mostly requires filter size ranges 3-10 to find optimal CNN model(s). Therefore, it will be best to have much dataset so it takes less time to search for an optimal filter(s) combination. With subject dependent analyses on dataset II, we found 100% performance on filter size from 1-3 as seen in table 8.

Table 8: CNN Models Performance on Subjects 1 - 10

Subjects	Model	Filters(s, c)	Test(%)	Sens.	Spec.
Subject 1	1	1, 5	100	1.0	1.0
	2	2, 2	100	1.0	1.0
	3	1, 10	98.94	0.98	1.0
Subject 2	1	1, 3	100	1.0	1.0
	2	2, 2	100	1.0	1.0
	3	2, 4	99.19	1.0	0.98
Subject 3	1	1, 2	100	1.0	1.0
	2	2, 1	100	1.0	1.0
	3	2, 2	100	1.0	1.0
Subject 4	1	1, 1	100	1.0	1.0
	2	1, 3	100	1.0	1.0
	3	1, 4	100	1.0	1.0
Subject 5	1	1, 2	100	1.0	1.0
	2	1, 4	100	1.0	1.0
	3	2, 2	100	1.0	1.0
Subject 6	1	1, 2	100	1.0	1.0
	2	1, 3	100	1.0	1.0
	3	2, 2	100	1.0	1.0
Subject 7	1	1, 2	100	1.0	1.0
	2	1, 3	100	1.0	1.0
	3	1, 4	100	1.0	1.0
Subject 8	1	1, 3	100	1.0	1.0
	2	2, 1	100	1.0	1.0
	3	2, 2	100	1.0	1.0
Subject 9	1	1, 3	100	1.0	1.0
	2	2, 1	100	1.0	1.0
	3	2, 3	99.84	.99	1.0
Subject 10	1	1, 2	100	1.0	1.0
	2	1, 4	100	1.0	1.0
	3	2, 2	100	1.0	1.0

We deduced that at least the filter size or quantity tend to be odd number to obtain the best performing models. Thus, odd numbers may be a good candidate filter size or filter combination on future EEG studies. With respect to subject independent studies, when Subject 9 dataset was tested on models trained using subject 1-8 dataset, the model from filter combinations of (3, 9) produced the best result of 86.78% classification accuracy. And the same model on Subject 10 produced the best result of 93.52% classification accuracy. As seen in figure 8, the overall performance on all testing dataset combined showed 90.15% classification accuracy.

On EEG device and subject independent analyses, Dataset II was used to train 25 CNN models generated from hyper-parameters described in table VII. And Dataset I from another EEG device and subject was applied for testing, the result is seen in figure 9 with the highest accuracy of 80.47% at filter combination of (3, 4).

	F1	F2	F3	F4	F5
Fc1	31.39	69.95	72.79	52.42	54.93
Fc2	31.39	45.41	34.89	56.43	43.91
Fc3	31.39	64.44	54.76	34.06	68.28
Fc4	68.61	38.56	80.47	48.25	51.25
Fc5	67.45	63.44	61.77	37.56	58.6

Figure 9: Classification accuracies from Dataset I from 25 models trained by Dataset II

The performance analyses on dataset I showed improvement as the number of CNN layers increases. This is shown in figure 7 where the highest classification accuracy on the test dataset from 100 CNN models is plotted against the number of CNN layers. There was a slight degradation of performance between model with layer 1 and layer 2. This may suggest that with an optimal choice of filter combinations, increasing the CNN layer from 1 to 2 layers may not impact performance for eye signal EEG analysis. But there was improvement upon increasing the layers from layer 1 to 3, 2 to 3 and 3 to 4.

Thus, we got the best results from the highest network depth from 1 to 4 with 96.51% from test set with temporal ordering. Which is consistent with the hypothesis from [24] and [10] where they surmised that network depth can be proportional to performance.

With regard to dataset II analyses, we did not use the entire 64 channels from the original dataset, rather we extracted just 14 channels corresponding to the available channels from dataset I. Upon analyses, we got prediction accuracy of 100% from four layers CNN models on subject dependent analyses. Thus, the entire electrode channels for an EEG study may not be required to efficiently perform classification.

We noted that dataset II is more balanced and more in quantity than dataset I because dataset I has temporal length of 117 seconds for both tasks (eye open and eye closed) from a device having 128 samples per seconds. Whereas, dataset II was acquired for 2.5 minutes per task for each subject using a device with 256 sampling rate. This may be the reason why the results from dataset II is better than the results from dataset I because deep learning does perform best with much data.

In comparison with previous studies, the study in [14] got classification accuracy of 97.30% from their KStar algorithm and study[5] got 97.4% by applying Regularized Random Forest RRF and KStar to the same dataset[13]. Although, these seem to be good results but only validation set was used for performance evaluation (missing test set). We got classification accuracy of 96.51% but this is achieved on the test set. And on subject independent studies, we tested thesame dataset[13] on networks trained using all subjects from dataset II and we got 80.47% classification accuracy. Thus, we did both subject dependent and independent studies while keeping temporal ordering in place.

5 FUTURE WORK

The datasets applied to this study may not have involved much complex activities of the brain because the task for acquisition involved eye open and eye close states. Our future work will apply these analyses to more complex EEG activities in future studies. An ensemble classifier may be formed from selected high performing CNN models to do prediction in the future. Filter combinations are read only hyper-parameters on CNN features learning but an automatic selection of optimal combination before training is still open for research. The application of our grid-like filter combinations on short term memory detection is what we hope to do in the future.

6 CONCLUSION

In this paper, we proposed a grid-like tool of filter combinations for CNN model generation. We analyze both subject dependent and subject independent eye state EEG classifications. There is not a magic number to filter size or quantity to create efficient CNN models for eye state classification across subjects. Rather, with a little search across a finite space, optimal filter(s) and network depth can be discovered for use. On subject dependent analysis, we got 96.51% on dataset I and 100% on dataset II. Subject's independent studies from one EEG device used by all subjects in dataset II showed best classification accuracy of 90.15%. When analyzing dataset I from CNN networks trained with dataset II, we got 80.47% as the highest classification result.

REFERENCES

[1] Robert J Barry, Adam R Clarke, Stuart J Johnstone, Christopher A Magee, and Jacqueline A Rushby. 2007. EEG differences between eyes-closed and eyes-open resting conditions. *Clinical neurophysiology* 118, 12 (2007), 2765–2773.

[2] Yu A Boytsova and SG Danko. 2010. EEG differences between resting states with eyes open and closed in darkness. *Human physiology* 36, 3 (2010), 367–369.

[3] PA Estévez, CM Held, CA Holzmann, CA Perez, JP Pérez, J Heiss, M Garrido, and P Peirano. 2002. Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. *Medical and biological engineering and computing* 40, 1 (2002), 105–113.

[4] Salvador García, Julián Luengo, and Francisco Herrera. 2015. *Data preprocessing in data mining*. Vol. 72. Springer.

[5] Cameron R Hamilton, Shervin Shahryari, and Khaled M Rasheed. 2015. Eye state prediction from EEG data using boosted rotational forests. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 429–432.

[6] Sunhee Hwang, Minsong Ki, Kibeom Hong, and Hyeran Byun. 2020. Subject-independent EEG-based emotion recognition using adversarial learning. In *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 1–4.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[8] John BO Mitchell. 2014. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4, 5 (2014), 468–481.

[9] Sanam Narejo, Eros Pasero, and Farzana Kulsoom. 2016. EEG based eye state classification using deep belief network and stacked autoencoder. *International Journal of Electrical and Computer Engineering (IJECE)* 6, 6 (2016), 3131–3141.

[10] Alison O'Shea, Gordon Lightbody, Geraldine Boylan, and Andriy Temko. 2017. Neonatal seizure detection using convolutional neural networks. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[11] Kemal Polat and Salih Güneş. 2007. Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform. *Appl. Math. Comput.* 187, 2 (2007), 1017–1026.

[12] Andreas M Ray, Ranganatha Sitaram, Mohit Rana, Emanuele Pasqualotto, Korhan Buyukturkoglu, Cuntai Guan, Kai-Keng Ang, Cristián Tejos, Francisco Zamorano, Francisco Aboitiz, et al. 2015. A subject-independent pattern-based brain-computer interface. *Frontiers in behavioral neuroscience* 9 (2015), 269.

[13] Oliver Roesler. 2013. EEG eye State Dataset. *Baden-Wuerttemberg Cooperative State University (DHBW), Stuttgart* (2013).

[14] Oliver Rösler and David Suendermann. 2013. A first step towards eye state prediction using eeg. *Proc. of the AIHLS* (2013).

[15] Kadir Sabancı and Murat Köklü. 2015. The classification of eye state by using kNN and MLP classification models according to the EEG signals. (2015).

[16] Abolfazl Saghafi, Chris P Tsokos, and Hamidreza Farhidzadeh. 2017. Common spatial pattern method for real-time eye state identification by using electroencephalogram signals. *IET Signal Processing* 11, 8 (2017), 936–941.

[17] P Sandheep, S Vineeth, Meljo Poulose, and DP Subha. 2019. Performance analysis of deep learning CNN in classification of depression EEG signals. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 1339–1344.

[18] Mastaneh Torkamani-Azar, Sumeyra Demir Kanik, Serap Aydin, and Mujdat Cetin. 2020. Prediction of reaction time and vigilance variability from spatio-spectral features of resting-state EEG in a long sustained attention task. *IEEE journal of biomedical and health informatics* 24, 9 (2020), 2550–2558.

[19] Rene Vidal, Joan Bruna, Raja Giryes, and Stefano Soatto. 2017. Mathematics of deep learning. *arXiv preprint arXiv:1712.04741* (2017).

[20] Ting Wang, Sheng-Uei Guan, Ka Lok Man, and TO Ting. 2014. EEG eye state identification using incremental attribute learning with time-series classification. *Mathematical Problems in Engineering* 2014 (2014).

[21] Femi William, Ramazan Aygun, and Feng Zhu. 2020. ERP Template Matching for EEG Single Trial Classification. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2876–2883.

[22] Haiyan Xu and Konstantinos N Plataniotis. 2015. Subject independent affective states classification using EEG signals. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 1312–1316.

[23] Mervyn VM Yeo, Xiaoping Li, Kaiquan Shen, and Einar PV Wilder-Smith. 2009. Can SVM be used for automatic EEG detection of drowsiness during car driving? *Safety Science* 47, 1 (2009), 115–124.

[24] Jianhua Zhang, Sunan Li, and Rubin Wang. 2017. Pattern recognition of momentary mental workload based on multi-channel electrophysiological data and ensemble convolutional neural networks. *Frontiers in neuroscience* 11 (2017), 310.