Using Deep Learning to Classify Saccade Direction from Brain Activity

Ard Kastrati*
akastrati@ethz.ch
Department of Electrical Engineering, ETH Zurich
Zurich, Switzerland

Martyna Beata Płomecka*
martyna.plomecka@uzh.ch
Department of Psychology, University of Zurich
Zurich, Switzerland

Roger Wattenhofer
Department of Electrical Engineering, ETH Zurich
Zurich, Switzerland

Nicolas Langer
Department of Psychology, University of Zurich
Zurich, Switzerland

ABSTRACT

We present first insights into our project that aims to develop an Electroencephalography (EEG) based Eye-Tracker. Our approach is tested and validated on a large dataset of simultaneously recorded EEG and infrared video-based Eye-Tracking, serving as ground truth. We compared several state-of-the-art neural network architectures for time series classification: InceptionTime, EEGNet, and investigated other architectures such as convolutional neural networks (CNN) with Xception modules and Pyramidal CNN. We prepared and tested these architectures with our rich dataset and obtained a remarkable accuracy of the left/right saccades direction classification (94.8 %) for the InceptionTime network, after hyperparameter tuning.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

neural networks, gaze detection, simultaneous Electroencephalography and Eye-tracking, time-series classification

ACM Reference Format:
Ard Kastrati, Martyna Beata Płomecka, Roger Wattenhofer, and Nicolas Langer. 2021. Using Deep Learning to Classify Saccade Direction from Brain Activity. In *ETRA '21: 2021 Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers), May 25–27, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3448018.3458014

1 INTRODUCTION

During everyday life, we direct our eyes to attend and extract new information from our visual environment [Grant and Spivey 2003]. Therefore, gaze information is a widely used behavioral measure in cognitive science and psychology to study attentional focus, cognitive control, or decision making [Cohen et al. 2007]. Recently, researchers have shown an increased interest in estimating gaze direction from brain activity. One of the most successful attempts to solve this problem was the Predictive Eye Estimation Regression [LaConte et al. 2007], an imaging-based method that uses machine learning algorithms to estimate the direction of gaze in the functional Magnetic Resonance Imaging (fMRI) time series based on voxel-wise data from the eyes [Son et al. 2020]. This indicates a great potential for neuroscience labs that cannot afford an eye-tracking system or don't have the expertise in analyzing eye-tracking data.

However, fMRI data acquisition is costly and does not provide temporal resolution at the level that cognition takes place. In contrast, EEG is a safe and cost-friendly method that measures the electrical activity of the brain directly and enables measurement in clinical settings (e.g. long-term recordings in a hospital bed).

Machine learning techniques allow extracting information from EEG recordings of brain activity and play a crucial role in several significant EEG-based research and application areas[Roy et al. 2019a]. In particular, deep learning allows computational models to learn representations of data with multiple abstraction levels and, therefore, to use all of the information that the dataset has to offer [Vahid et al. 2020]. Although deep learning for EEG time series classification proved to be successful and widely used [Roy et al. 2019b], it still lags behind image recognition in terms of experimental studies and architectural designs. In this project, as a first step towards the classification of gaze position, we hypothesize that the saccade's direction (left and right) can be restored by using the combination of EEG and deep learning. This, in turn, could enrich already acquired and available EEG studies with additional information about saccades direction, making re-analyses of these datasets possible with the aim to validate the results and address novel research questions that were previously inapplicable (e.g. age-related brain activity differences confounded by the fact that older subjects move their eyes more often than young subjects [Płomecka et al. 2020]).

Moreover, recovering eye-gaze information from EEG data can potentially improve many EEG-based assistive technologies, like limb prosthesis or mobile robots build on non-invasive brain-computer interface (BCI), providing additional rich measure and helping the existing ones for better performance [Kapralov et al. 2019]. We believe that this study offers some important insights into more advanced assistive technologies, such as hybrid approaches that combine ET with EEG [Millán et al. 2010].

There has been some research carried out on the classic supervised machine learning techniques; for example, [Bulling et al.

*Both authors contributed equally to this research.

ETRA '21 Short Papers, May 25–27, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8345-5/21/05.
https://doi.org/10.1145/3448018.3458014

2010] classified different directions and lengths of saccades with a mean precision of 76.1 %. However, to the best of our knowledge, there were no attempts to employ deep learning to classify saccade direction (left vs right) from EEG brain activity.

A principal challenge in developing an EEG based Eye-Tracker is obtaining rich enough and precisely annotated data. This barrier we already have overcome with our available data set containing recordings from 364 healthy participants. For the proposed project, we explored the prosaccade task (a part of the antisaccade task [Antoniades et al. 2013]), where participants were asked to look left or right, based on the direction of the target displayed on the screen. This paper's results serve as a motivation for our future work. We currently record the new dataset that allows us to extend the classification of saccades direction beyond left and right.

2 MODELS AND METHODS

This study make use of an existing simultaneously recorded EEG and infrared video-based eye-tracking dataset. The dataset comprises 364 participants between 19 and 83 years of age with normal or corrected to normal vision. Written informed consent was given by all participants before the experiment. None of the participants communicated neurological impairments (e.g. recent concussions, Parkinson disease, dementia, epilepsy) or any psychiatric diagnosis at the date of measurement. Furthermore, no medication that could affect the ET/EEG signal was taken.

2.1 Data Acquisition

For all recorded data, an identical acquisition setup has been used. The data were collected with an 128-Channel EEG system and simultaneous eye-tracking recordings with an infrared video-based Eye-Tracker. The experimental setup reflects the current state-of-the-art, with an electrically shielded recording room.

2.1.1 Eye-Tracking Acquisition. Eye-tracking data were used to provide a ground truth information for the analysis of the prosaccade task. Besides the trigger onset, the direction and length of the saccades were used for the reduction of errors (see Subsection 2.3). Eye-tracking was recorded mostly on the left eye (in three cases on the right eye) with an infrared video-based eye-tracker (Eye Link 1000 Plus, SR Research, http://www.srresearch.com), which is accurate down to 0.15° (typical 0.25°-0.50°) and has a sampling rate of 500Hz. The calibration was done by a 9-point-grid before every block of the prosaccade task. During the calibration, the participants were asked to look at nine targets appearing on the borders of the screen, the edges of the screen, and the middle of the screen. After the calibration, a validation was performed with the same protocol to assess whether the average error of all points was kept below 1°. If not, the eye-tracking was adjusted, and the calibration redone. For the final sample used in this manuscript, the average error of all participants was equal to 0.56° (calculated as the average of the error of all points).

The algorithm provided by EyeLink 1000 Plus was used to identify saccades, fixations and blinks. The acceleration threshold was set as 8000° per second, the velocity threshold as 30° per second and the deflection threshold as 0.1°. The resulting data set included gaze locations in a XY-coordinate system with pixels as units.

2.1.2 Electroencephalography Acquisition. High-density EEG data was recorded at a sampling rate of 500 Hz, recording reference "Cz", using a 128-channel EEG Geodesic Hydrocel system (Electrical Geodesics, Eugene, Oregon).

2.2 The Pro- and Antisaccade Task

The pro- and antisaccade paradigm was based on the internationally standardized protocol for antisaccade testing developed by Antoniades et al [Antoniades et al. 2013]. Each task started with a central fixation square. The participants were asked to focus on the center of the screen for a randomized time-period between 1 and 3.5 seconds. Subsequently, the cue (i.e. dot) appeared in a horizontal line on the left or the right hand-side of the central fixation square. In the subtask "prosaccade", the participants were asked to focus their gaze on the cue as fast as possible. In the subtask "antisaccade" the participants were instructed to perform a saccade in the opposite side of the cue. In both tasks, the cue was shown for 1 second. As soon as the dot disappeared, the participants shifted their focus back to the center of the screen. The horizontal traverse of the eye formed a visual angle of 8° left or right of center. The appearance of the dots in the left- and the right-hand square was randomized. The subtasks were presented in blocks in the following order: 1. Prosaccade, 2. Antisaccade, 3. Antisaccade, 4. Antisaccade, 5. Prosaccade. Each prosaccade block consisted of 60 trials (30 trials per visual hemifield) and each anti-saccade block of 40 trials (20 trials per visual hemifield). There was a one-minute break between each block. The dataset used in this study only makes use of the prosaccade trials.

2.3 Dataset Preprocessing

Infrared eye tracking data was preprocessed with the EyeLink 1000 algorithm to identify saccades, gaze fixations and blinks. Saccades onsets were detected by the velocity and acceleration of the eye movements using the standard acceleration (8000° per sec2), velocity (30° per sec) and deflection (0.1°) thresholds. Fixations were defined as time periods without saccades and eye blinks are regarded as a special case of a fixation, where the pupil diameter is either zero or outside a dynamically computed valid pupil.

EEG data was preprocessed with the openly available preprocessing toolbox "Automagic" for MATLAB [Pedroni et al. 2019] including bad channel identification and interpolation, 0.5 Hz high-pass filtering, artifact (as muscular noise, heart signals and sweating) correction with automated independent component analysis.

After preprocessing, the EEG and eye-tracking data were synchronized using "EYE EEG extension" [Ehinger and Dimigen 2019] to accurately enable extracting EEG data time-locked to the onsets of fixations and saccades from the infrared Eye-Tracker. The preprocessed sample consisted of 40 000 one-second trials locked to the stimulus onset. This sample was screened again and 3778 trials were later excluded due to: occurrences of eye blinks between the cue presentation and the saccade, saccade onset earlier than 100 or later than 800 ms after cue presentation, no saccades onset after the cue presentation, or an error trial. An error was defined as a saccade away from the target stimulus. If 70 % or more trials were rejected, the whole subject was excluded from the analysis (7

subjects were excluded and thus only the data from the remaining 357 participants is used in our models).

2.4 Classification of Gaze Direction

2.4.1 Training and Validation Dataset. The dataset consisted of 36222 training and validation samples collected from the 357 participants, from which 80% of the data was used for training and 20% for validation. We studied the performance for cross-subject classification. Classification results are reported for two sets of analyses. In the first method, the data was shuffled between all the participants and then divided into two groups of sizes: 80% for training and 20% for validation. In the second approach, we chose 287 participants for the training set and the remaining 70 participants for the validation set. Note that in the first approach, in expectation, the model is trained with 80% and validated with 20% of the samples of *each* participant. On the other hand, the second approach enforces a fully cross-subject analysis since the validation data is from entirely new participants. Each sample has the size (500, 128): 500 measurements for every 2ms (a time window of 1 second in total) and 128 electrodes. The electrodes' placement is irrelevant for all used models since, in each filter, convolution is applied to all electrodes. The primary motivation for the one-second window's length is based on the experiment's design. Stimulus is always presented for one second, and the participant is supposed to perform a saccade in that time window. Additionally, the exact description of the experiment and the length of the stimulus presentation can be found in [Płomecka et al. 2020]. The data set (left and right) was equally balanced, and each sample was mapped either to 0 (for looking left) or 1 (for looking right).

2.4.2 Baseline Classifiers. We approached the left-right classification problem initially by using several standard classifiers implemented in the scikit-learn library. We report the results of the following baseline classifiers: Random Forest, Nearest Neighbors, Linear SVM and Naive Bayes. Other standard classifiers (such as non-linear SVM, AdaBoost, etc.) provided by scikit-learn are not reported since they don't scale, and with our big dataset, it is infeasible to train such models.

2.4.3 DNN Architectures. We compared five different state-of-the-art DNN architectures: EEGnet [Lawhern et al. 2018], Inception Time Network (IncTime) [Fawaz et al. 2020a], traditional Convolutional Neural Network (CNN), Convolutional Pyramidal Neural Network (PyrCNN) [Krizhevsky et al. 2017], and a network with an Xception module (Xception) [Chollet 2017]. All models were trained in an ensemble composed of 5 identical architectures. For the final classification, we used the major votes out of 5 classifications from each ensemble. During the training of the CNN, PyrCNN, IncTime and Xception, we tuned the following parameters:

- *The kernel size of each convolution*
- *The depth of the architecture*
- *The bottleneck size*
- *The number of filters used in each layer.*

For the EEGnet architecture, we tuned:

- *the dropout fraction*
- *the length of temporal convolution*
- *the number of temporal filters*
- *the number spatial filters*
- *the number of pointwise filters.*

All models were trained using one GPU per run (NVIDIA GeForce GTX 1080 Ti), in Tensorflow, using the Keras API [Abadi et al. 2016].

- *Standard Convolutional Neural Network*
  This module is the simplest one which consists of a straightforward convolutional step. In this architecture we didn't use any residual connections.
- *Pyramidal CNN* [Krizhevsky et al. 2017]
  The traditional CNN model requires attention to reduce number of learnable parameters, with no meaningful reduction in performance. Thus we implemented pyramid structure typical for biological neurons.
- *EEGnet* [Lawhern et al. 2018]
  EEGnet performs two convolutional steps in sequence. First, a 2D convolutional filters outputting feature maps containing the EEG signal at different band-pass frequencies. The next step is a depthwise convolution to learn a spatial filter.
- *InceptionTime* [Fawaz et al. 2020a]
  InceptionTime Network [Fawaz et al. 2020b] is a novel deep learning ensemble for time series classification that applies multiple filters simultaneously to an input time series. The InceptionTime module includes filters of varying lengths, that allow the network to automatically extract relevant features from time series.
- *Xception Network* [Chollet 2017]
  This CNN model is built on the backbone of the Xception [Chollet 2017], that was designed by replacing the Inception modules with a depthwise separable convolution. The basic idea behind the Xception network was adopting multiple one-dimensional filters with different kernel sizes to extract features. Simultaneously, the resulting feature maps was concatenated to construct the output features. The use of the depthwise separable convolutions, significantly mitigated the required number of parameters in the network.

3 RESULTS

In the following, we present our results for the baselines and the deep neural networks. For every model, we tuned their hyperparameters: for the baselines, we used cross-validation, and for the deep neural networks, we used Keras Tuner. During hyperparameter optimization, we have observed that different hyperparameters choices achieve similar performance. For brevity, we report the accuracy of model ensembles only on the best obtained architectures for all described models. Other hyperparameters choices are also possible, and the difference in performance among the best ten hyperparameter choices is less than 1% for each model. Some models have considerably different architectures from other models, and thus they are subject to some limitations in their comparability, which we will elaborate in the next section.

3.1 Baseline Results

In Table 1 we report our results of the baselines: Random Forest, Nearest Neighbors, Linear SVM and Naive Bayes implemented in scikit-learn library. We tuned the reported hyperparameters by using cross-validation.

Table 1: Classification accuracy of the baselines, their running time (in seconds) and the tuned hyperparameters by using cross-validation.

Architecture	Max Acc (%)	Runtime	Hyperparameters
Random Forest	65.99	883	max depth: 5, min samples leaf: 0.1, min samples split: 0.1, nr. of estimators: 1000
Linear SVM	70.57	18550	tolerance: 1e-3, regularization term: 0.1, max iter: 12000
Nearest Neighbor	54.12	3658	nr. of neighbors: 100, leaf size: 100
Naive Bayes	54.22	57	—

Table 2: Classification accuracy (Max Acc (%)) for the ensembles of different architectures, number of trainable parameters of model (# of par), and running time of ensemble, each consisting of 5 models (Runtime,in seconds).

Architecture	Max Acc (%)	# of par	Runtime
CNN	91.1	315,681	6830
PyrCNN	93.8	320,513	3575
EEGnet	89.8	186,689	10866
Xception	90.5	158,441	17177
IncTime	94.9	393,377	11235

Table 3: Overview of the hyperparameters of CNN, PyrCNN, Xception and IncTime. DEPTH - depth of the network, BATCH - batch normalization size, KERNEL - Kernel size, FILTER - Filter size, BN - bottleneck size

Hyperparameters	CNN	PyrCNN	Xception	IncTime
RES	TRUE	FALSE	TRUE	TRUE
DEPTH	12	6	18	12
BATCH	64	64	54	64
KERNEL	64	16	40	64
FILTER	16	16	64	16
BN	-	-	-	16

3.2 DNN Results

The summary of performance of used models is presented in Table 2. We compared the highest obtained accuracy of ensemble, the number of parameters of the models and running time on cluster. All reported models achieved maximum accuracy greater than 89.8%.

As shown in the Table 2, the InceptionTime Network architecture outperformed other solutions with almost 95% accuracy and became the best architecture for our rich dataset.

CNN, PyrCNN, Xception and IncTime share the same general structure: they only differ in the way how the convolution module in each step of the depth is implemented. Thus they have the same hyperparameters. The hyperparameters presented in Table 3 are tuned by using Keras Tuner. Additionally, we also tuned the EEGnet architecture, obtaining accuracy of 89.8% with the following hyperparameters: dropout fraction was set to 0.5, length of temporal convolution set to 64, number of temporal filters set to 32, number of spatial filters to learn within each temporal convolution was set to 8, and the number of pointwise filters to 512.

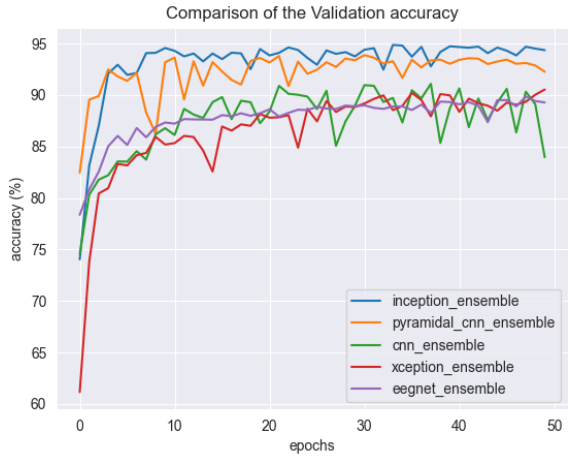
The comparison of the validation accuracy and validation binary cross-entropy loss of tested ensembles for both approaches of cross-subject classifications are presented in Figure 1 and Figure 2.

4 CONCLUSIONS

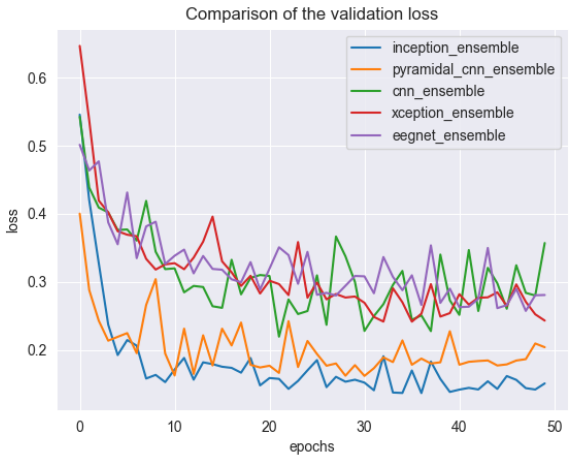
In this study, we proposed the binary classification of gaze direction (left/right), as the first step of the larger project, aiming to predict the general gaze position. To the best of our knowledge, the prosaccade task was not tested before on state-of-the-art neural networks for time series classification. Moreover, there were no attempts to classify saccades directions from the brain activity using deep learning. The findings of this study suggest that InceptionTime is the best architecture suitable for the prosaccade task (classifying left or right). Comparing this result with the best results of some standard classifiers, where Random Forest achieves a maximum accuracy of 65.99% and a Linear SVM of 70.57% shows that deep neural networks are very suited for classification of gaze direction and perform with a remarkable accuracy. The bad performance of standard classifiers is also due to the fact that we used in our experiments for each classifier 500×128 features for each sample. The accuracy in baseline classifiers could be improved by first applying a feature extraction algorithm. However, this typically requires understanding of EEG data and for some tasks can be very difficult. On the other hand, deep neural networks are able to learn these features end-to-end which makes them preferable.

It is also worth noting that we obtained a remarkable accuracy (93.8 %) for the standard Convolutional Neural Network with a typical pyramidal shape, where the number of the channels increases but the number of the spatial and time dimensions decreases for each layer of the neural network. Finally, we can see that the difference in validation accuracy and loss between the first approach (shuffling of the data) in Figure 1 and second approach (validation set contains only participants' data that is not used for training) in Figure 2 is negligble and is always less then 1%. This confirms that our models are participant-agnostic.

The generalizability of these results is subject to certain limitations. We used hyperparameter optimization methods such as Keras Tuner and cross-validation to see the best achieved accuracy. As we mentioned in the previous section, some models considered in this paper have considerably different architectures and use different methods. Therefore, they are not easily comparable. Although the reported deep neural network models have hundreds of thousands of parameters, they still differ in their size and training time. For instance, InceptionTime achieves the highest accuracy but has more

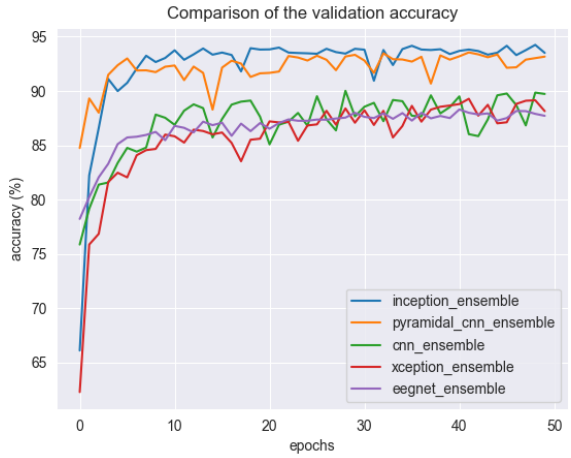


(a) Comparison of the validation accuracy of tested ensembles

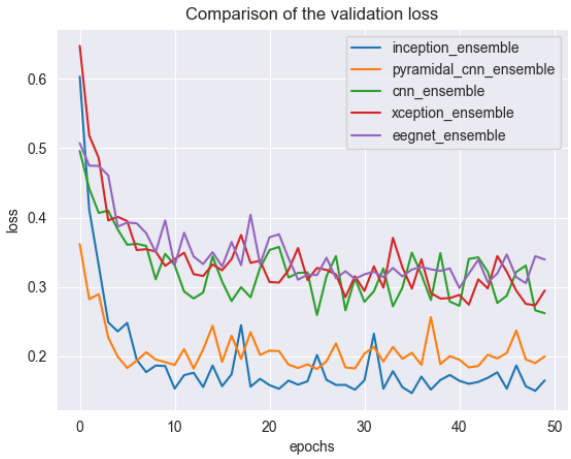


(b) Comparison of the validation loss of tested ensembles

Figure 1: Approach 1: Comparison of validation accuracy and validation loss where the data was shuffled and then divided into two groups: 80% for training and 20% for validation.



(a) Comparison of the validation accuracy of tested ensembles



(b) Comparison of the validation loss of tested ensembles

Figure 2: Approach 2: Comparison of validation accuracy and validation loss where we chose 287 participants for the training set and the other remaining participants for the validation set.

parameters and is more resource expensive. We can also observe in the baseline models that Linear SVM performs better than other models. However, it needs several hours of training time until it converges. EEGNet also has a different architecture with fewer parameters, but it has a higher training time due to the large filters used in each convolution. While our project uses a conveniently big dataset, more research is required to determine the dataset's influence and architecture size on model performance. Although achieving exceptionally high accuracy, reaching almost always 90% for the prosaccade task classification, it is necessary to test these architectures with other datasets and generalize them for more saccade directions and locations on the screen.

Another finding of this study is that all models reach a high accuracy (at least 85%) in very few epochs. Although we have used five ensembles to reduce the unstable results for each epoch, we can see that the binary cross-entropy loss still shows instability patterns. Increasing the number of ensembles decreases the significant variance of the results but increases the number of parameters. The binary cross-entropy loss decreases with the number of epochs. Nevertheless, it has a high variance, showing that even with different regularization methods, the models (except for the InceptionTime) have unstable behaviour.

A planned follow-up to this work will be an online public benchmark repository with different types of datasets and implementation

of all discussed models and extending this work for other tasks (not only left vs. right).

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 265–283.

Chrystalina Antoniades, Ulrich Ettinger, Bertrand Gaymard, Iain Gilchrist, Arni Kristjánsson, Christopher Kennard, R John Leigh, Imran Noorani, Pierre Pouget, Nikolaos Smyrnis, et al. 2013. An internationally standardised antisaccade protocol. *Vision research* 84 (2013), 1–5.

Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Tröster. 2010. Eye movement analysis for activity recognition using electrooculography. *IEEE transactions on pattern analysis and machine intelligence* 33, 4 (2010), 741–753.

François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

Michael X Cohen, Christian E Elger, and Charan Ranganath. 2007. Reward expectation modulates feedback-related negativity and EEG spectra. *Neuroimage* 35, 2 (2007), 968–978.

Benedikt V Ehinger and Olaf Dimigen. 2019. Unfold: An integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis. *PeerJ* 7 (2019), e7838.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020a. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020b. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

Elizabeth R Grant and Michael J Spivey. 2003. Eye movements and problem solving: Guiding attention guides thought. *Psychological Science* 14, 5 (2003), 462–466.

Nikolay V Kapralov, Jaroslav V Ekimovskii, and Vyacheslav V Potekhin. 2019. EEG-Based Brain-Computer Interface for Control of Assistive Devices. In *International Conference Cyber-Physical Systems and Control*. Springer, 536–543.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

S LaConte, C Glielmi, K Heberlein, and X Hu. 2007. Verifying visual fixation to improve fMRI with predictive eye estimation regression (PEER). In *Proc. Intl. Soc. Magn. Reson. Med*, Vol. 3438.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. 2018. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering* 15, 5 (2018), 056013.

José del R Millán, Rüdiger Rupp, Gernot Mueller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kubler, Robert Leeb, et al. 2010. Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in neuroscience* 4 (2010), 161.

Andreas Pedroni, Amirreza Bahreini, and Nicolas Langer. 2019. Automagic: Standardized preprocessing of big EEG data. *Neuroimage* 200 (2019), 460–473.

Martyna Beata Płomecka, Zofia Barańczuk-Turska, Christian Pfeiffer, and Nicolas Langer. 2020. Aging Effects and Test–Retest Reliability of Inhibitory Control for Saccadic Eye Movements. *Eneuro* 7, 5 (2020).

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. 2019a. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering* 16, 5 (2019), 051001.

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. 2019b. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering* 16, 5 (2019), 051001.

Jake Son, Lei Ai, Ryan Lim, Ting Xu, Stanley Colcombe, Alexandre Rosa Franco, Jessica Cloud, Stephen LaConte, Jonathan Lisinski, Arno Klein, et al. 2020. Evaluating fMRI-Based Estimation of Eye Gaze during Naturalistic Viewing. *Cerebral Cortex* 30, 3 (2020), 1171–1184.

Amirali Vahid, Moritz Mückschel, Sebastian Stober, Ann-Kathrin Stock, and Christian Beste. 2020. Applying deep learning to single-trial EEG data provides evidence for complementary theories on action control. *Communications biology* 3, 1 (2020), 1–11.