Harmful Brain Activity Classification

1. Overview

Problem Definition

The goal of this competition is to detect and classify seizures and other types of harmful brain activity by developing a model trained on electroencephalography (EEG) signals recorded from critically ill hospital patients.

Competition webpage

https://www.kaggle.com/competitions/hms-harmful-brain-activity-classification/overview

Metric

The metric is *Kullback–Leibler divergence*.

Training data

train.csv

Metadata for the train set. The expert annotators reviewed 50-second-long EEG samples plus matched spectrograms covering a 10-minute window centered at the same time, and labeled the central 10 seconds. Many of these samples overlapped and have been consolidated.

train_eegs/*.parquet

EEG data from one or more overlapping samples. The column names are the names of the individual electrode locations for EEG leads, with one exception. The EKG column is for an electrocardiogram lead that records data from the heart. All of the EEG data (for both train and test) was collected at a frequency of 200 samples per second.

train_spectrograms/*.parquet

Spectrograms assembled from EEG data. The column names indicate the frequency in hertz and the recording regions of the EEG electrodes. The latter are abbreviated as LL = left lateral; RL = right lateral; LP = left parasagittal; RP = right parasagittal.

Strategy for solving the problem

The main strategy is to ensemble various solutions as the Kullback-Leibler divergence metric penalizes incorrect predictions while being relatively forgiving towards uncertain predictions.

2. Training Data

train.csv

Notebook

`EDA_target.ipynb`

Columns of interest

- `eeg_id` - A unique identifier for the entire EEG recording.
- `eeg_label_offset_seconds` - The time between the beginning of the consolidated EEG and this subsample.
- `spectrogram_id` - A unique identifier for the entire EEG recording.
- `spectogram_label_offset_seconds` - The time between the beginning of the consolidated spectrogram and this subsample.
- `patient_id` - An ID for the patient who donated the data.
- `expert_consensus` - The consensus annotator label. Provided for convenience only.
- `[seizure/lpd/gpd/lrda/grda/other]_vote` – Target columns, the count of annotator votes for a given brain activity class.

Rows

- 106,800 rows – marked events,
- 17,089 events that are at least 50 seconds apart,
- 17,089 unique `eeg_id` (coincide with distinct events),
- 1,950 unique `patient_id`.

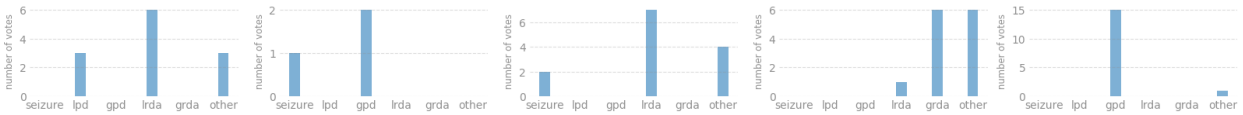
Notes

1. Work only with distinct events:
    a. Overlapping events are given with 2, 4, 6, 8, 10 seconds shifts from the distinct event and provide no new information;
    b. Reduce training time.
2. Grouped train/validation split on `patient_id` to prevent data leakage.

Target

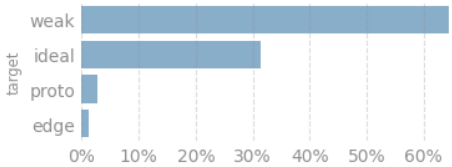
6 columns with a number of votes for a medical condition from a varying number of experts.

Examples of target distributions:

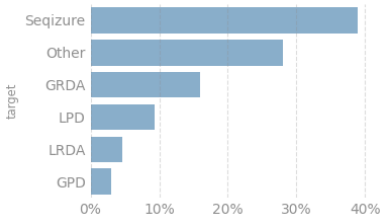


Segments where there are high levels of agreement here are called “idealized” patterns. Cases where ~1/2 of experts give a label as “other” and ~1/2 give one of the remaining five labels are called “proto patterns”. Cases where experts are approximately split between 2 of the 5 named patterns are called “edge cases”. The rest (including cases with less than 3 annotators) are considered “weak”.

Out of 17089 distinct events:



Distribution of expert consensus among idealized cases with at least 3 annotators:



Raw EEG signals

Notebooks

`EEG display typical data by label.ipynb` – display

- raw EEG data,
- ICA,
- PCA,

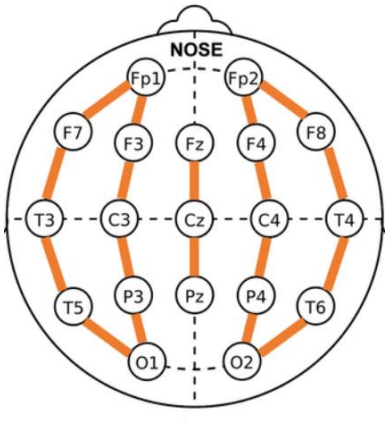
`EEG display montages.ipynb` – display

- longitudinal montage,
- regional average montage.

`EEG bad data.ipynb` – examples of bad raw EEG recordings

Overview

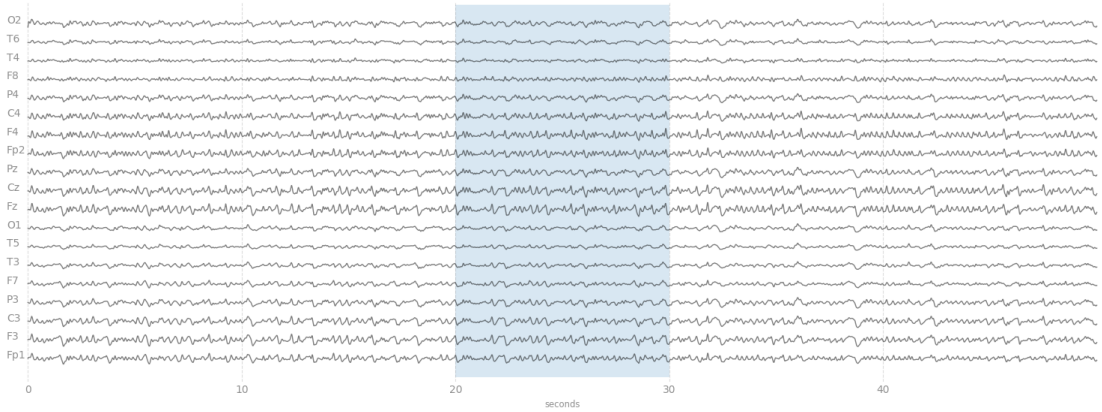
Raw signals (19 synchronized channels) from electrodes that are arranged over the head of a patient as follows:



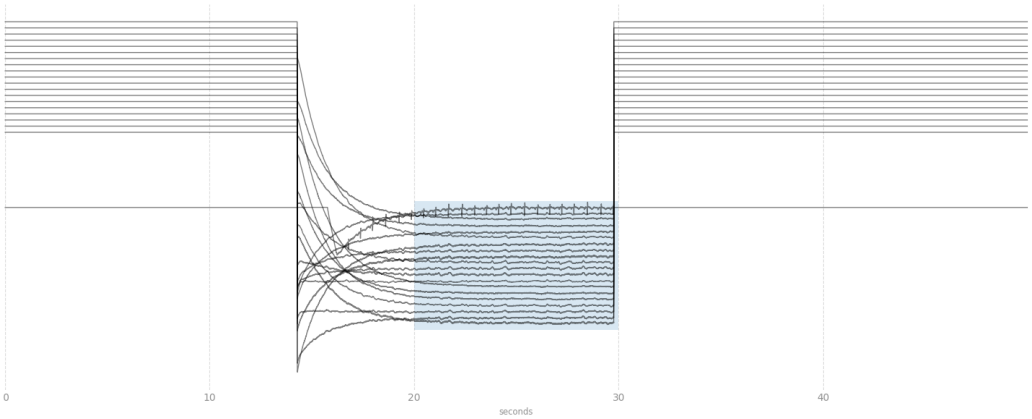
It is important to note that all information of the raw EEG data is contained inside the 0.1-100 Hz frequency range, and often frequencies above 40 Hz are filtered out.

Examples

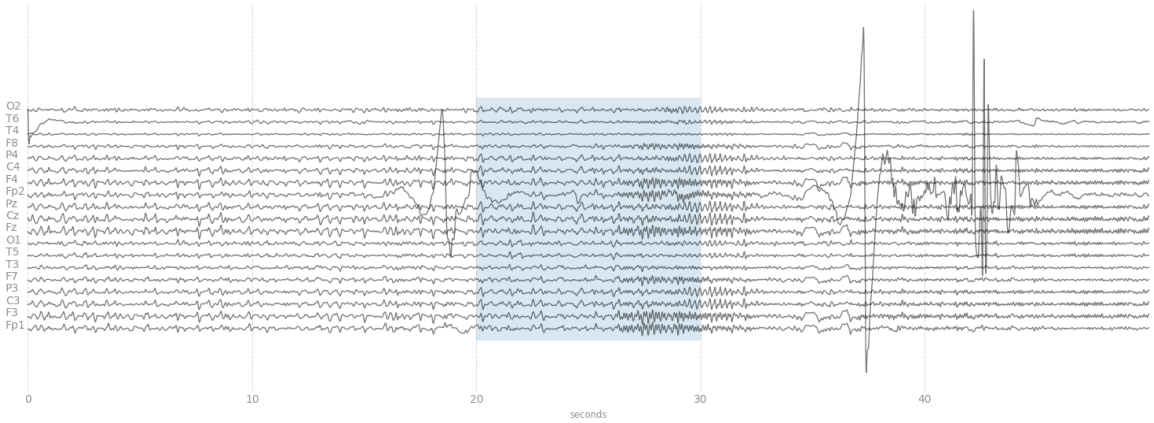
Example of a 50-seconds signal (central 10 seconds are labeled by experts):



There are 17,089 `*.parquet` files. Of them 20 contain data with significant artifacts:



Some signals contain one or more channels with artifacts:

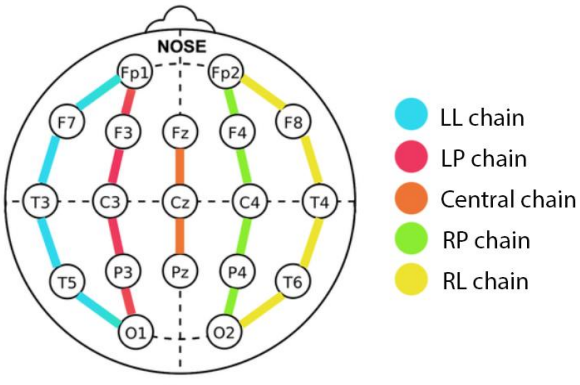


Spectrograms

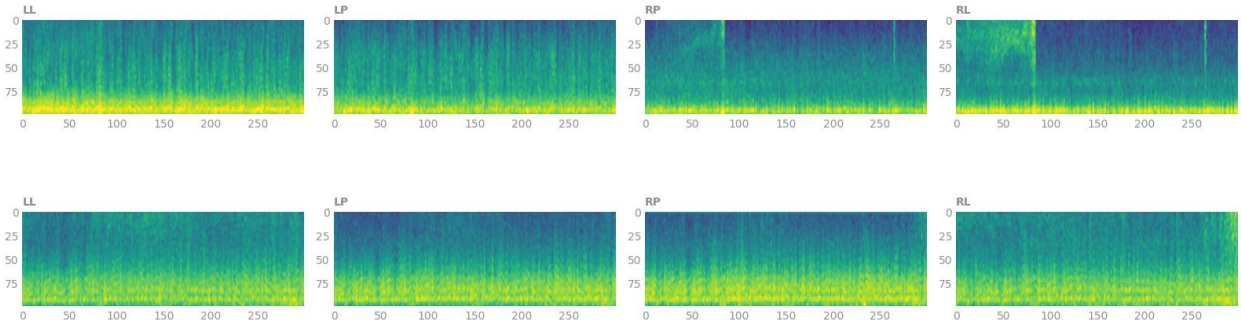
Notebook

`spectrograms.ipynb` – display provided spectrograms.

Provided spectrograms were derived from EEG recordings with an STFT window of 2 seconds. The spectrograms are averaged across longitudinal chains. Recording regions of the EEG electrodes are abbreviated as **LL** = left lateral; **RL** = right lateral; **LP** = left parasagittal; **RP** = right parasagittal.



Examples of provided spectrograms (4 spectrograms per target event):



3. EEG data transformations

Longitudinal montage

Notebook

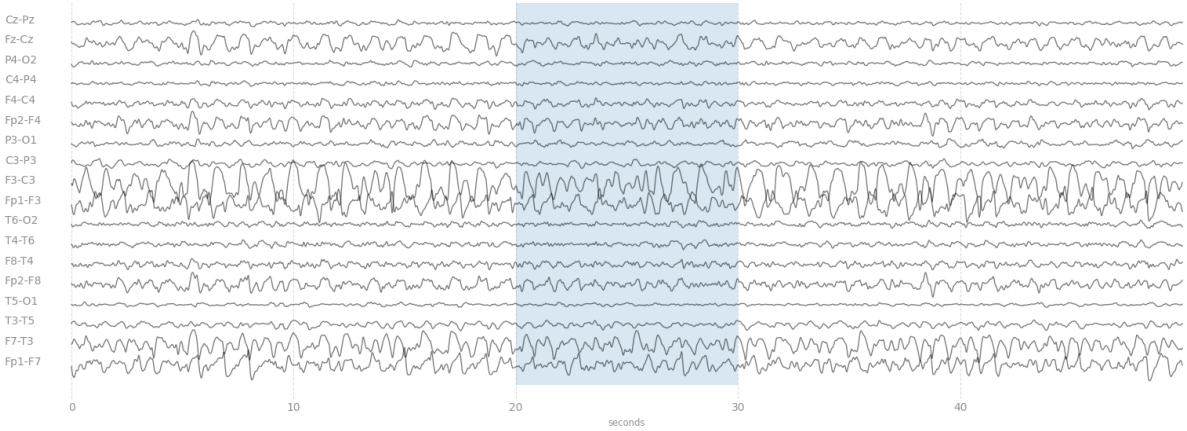
`EEG display montages.ipynb` – examples of longitudinal montage.

Overview

For each of the regional chains the montage consists of pairwise differences of raw signals:

Region	*Electrode signals*	*Montage signals*
Left lateral (LL)	Fp1, F7, T3, T5, O1	Fp1-F7, F7-T3, T3-T5, T5-O1
Right lateral (RL)	Fp2, F8, T4, T6, O2	Fp2-F8, F8-T4, T4-T6, T6-O2
Left parasagittal (LP)	Fp1, F3, C3, P3, O1	Fp1-F3, F3-C3, C3-P3, P3-O1
Right parasagittal (RP)	Fp2, F4, C4, P4, O2	Fp2-F4, F4-C4, C4-P4, P4-O2
Central	Fz, Cz, Pz	Fz-Cz, Cz-Pz

Example



Reference

https://www.learningeeg.com/montages-and-technical-components

Spectrograms

Notebook

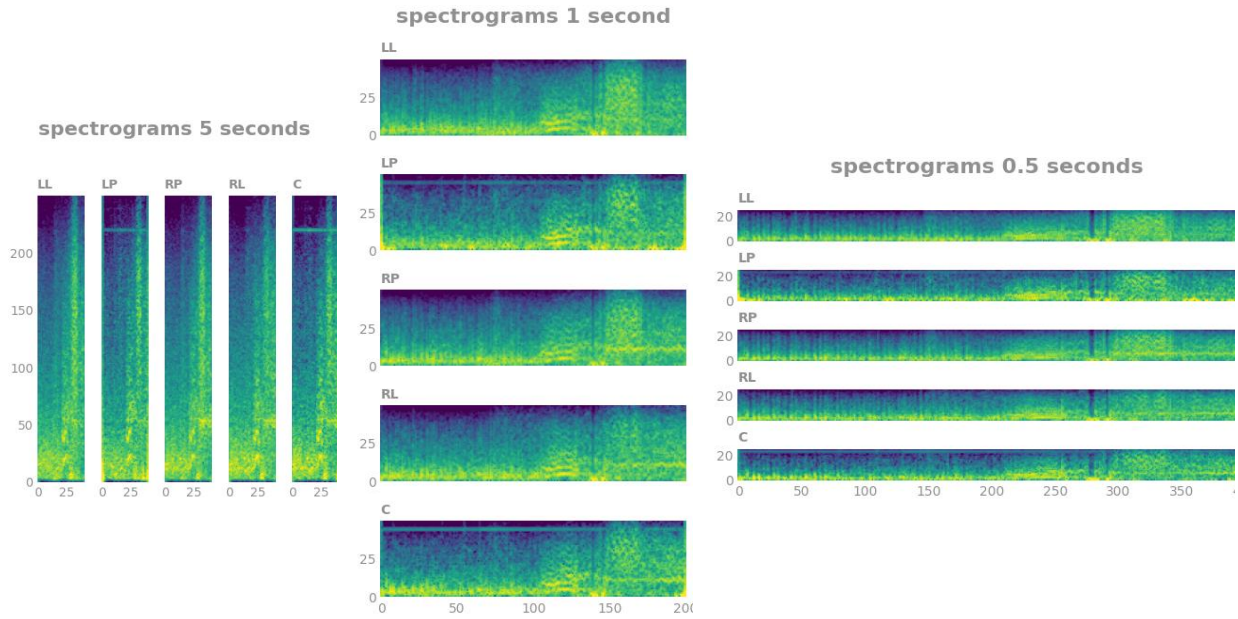
`spectrograms from montages.ipynb` – examples of spectrograms from longitudinal montage with different STFT windows.

Overview

One typical method for EEG signal analysis involves generating spectrograms from the raw EEG data. Spectrograms are produced from longitudinal montage signals and then averaged across each chain.

Example

Depending on the STFT window size the spectrograms of the EEG data contain different frequency and temporal information:



4. Solutions

Notebook

`solutions.ipynb` – detailed summary of out-of-fold predictions.

Overview

The data is split into 5 folds using `StratifiedGroupKFold` method from the sklearn library, with grouping performed over `patient_id` feature and stratification over `expert_consensus` feature. Each model is trained on 4 folds and valildated on the 5th fold, then the solution is verified against a hidden test set.

All solutions were implemented using PyTorch, with Adam optimizer and KLDivLoss.

EfficientNet-B0 on provided spectrograms

Data

Class `Spectrogram_4channels_Dataset` from `src/spectrogram_dataset.py`.

All four provided spectrograms (width 300, height 100) were combined into a single image 300x400 pixels, and then duplicated to create a 3-channel image compatible with EfficientNet architecture.

Model

Class `EfficientnetWrapper` from `src/efficientnet_wrapper_model.py`.

Efficientnet_b0 base model with a fully-connected classification layer.

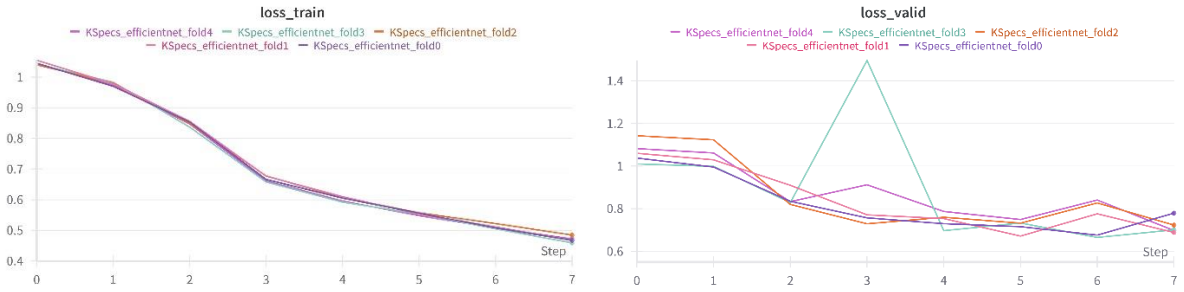
Training

During training the learning rate was fixed at 10^{-3}, and for the first 2 epochs only the custom classifier linear layer weights were updated. After the first 2 epochs all layers of the model were finetuned.

The best scores for each fold are:

FOLD	VALIDATION SCORE	HIDDEN TEST SCORE
0	0.676	0.60
1	0.672	0.62
2	0.724	0.55
3	**0.665**	**0.52**
4	0.697	0.60

Training progress for each fold:



Notes

1. The efficientnet_b0 model overfit on the training dataset without freezing all the weights except the last classification layer: the best validation scores after training were ~0.7, but scores on the hidden test set were ~0.8.
2. Creating a more complex classification head worsened the validation scores.
3. Using versions b1, b2 worsened validation scores.

WaveNet on longitudinal montage

Data

Class `EEG_Dif_Dataset` from `src/eeg_dif_dataset.py`.

Eight channels from raw EEG data are assembled into a longitudinal montage. Then, a low-pass filter is applied, and the signal is downsampled by a factor of 5.

Model

Class `SequentialWaveNet` from `src/wavenet.py`.

Suggested in notebook https://www.kaggle.com/code/cdeotte/wavenet-starter-lb-0-52/notebook.

WaveNet architecture is applied to longitudinal montage signals, averaged for each chain. Classifier is applied to concatenated features from each chain.

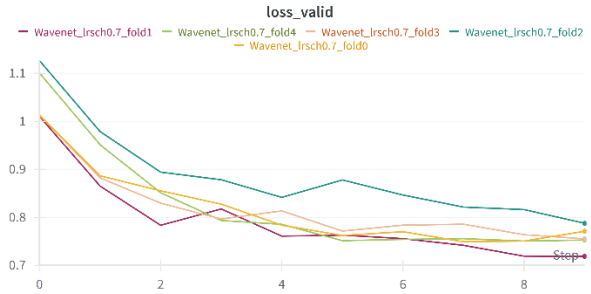
Training

Exponential learning rate scheduler was very effective for the learning process. Starting learning rate is 10^{-3}, reduced for subsequent epochs with gamma=0.7.

The best scores for each fold are:

FOLD	VALIDATION SCORE	HIDDEN TEST SCORE
0	0.750	0.55
1	**0.719**	**0.52**
2	0.789	0.55
3	0.755	0.55
4	0.751	0.55

Training progress for each fold:



Notes

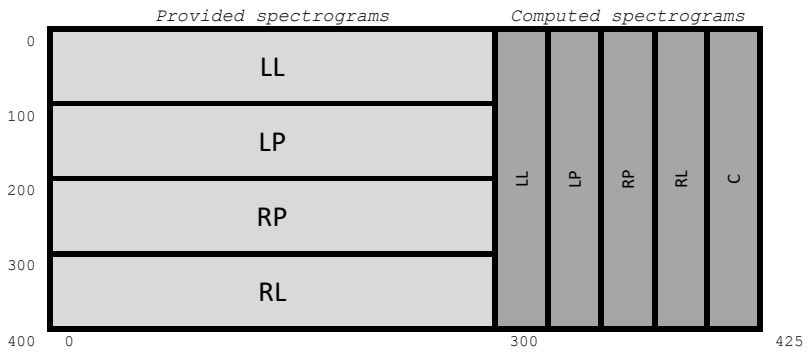
1. Without learning rate scheduler, with learning rate fixed at 10^{-3}, validation score stalled at ~0.8.
2. Fixing the learning rate at values of 10^{-4} or 10^{-5} resulted in validation scores greater than 0.9.

EfficientNet-B0 on combined spectrograms

Data

Class `SpectrogramsCombinedDataset` from `src/spectrograms_combined_dataset.py`.

The four provided spectrograms (width 300, height 100) and five spectrograms obtained from longitudinal montage with STFT window 0.5 second (width 400, height 25) were combined into a single image 425x400 pixels, and then duplicated to create a 3-channel image compatible with EfficientNet architecture:



Model

Class `EfficientnetWrapper` from `src/efficientnet_wrapper_model.py`.

Efficientnet_b0 base model with a fully-connected classification layer.

Training

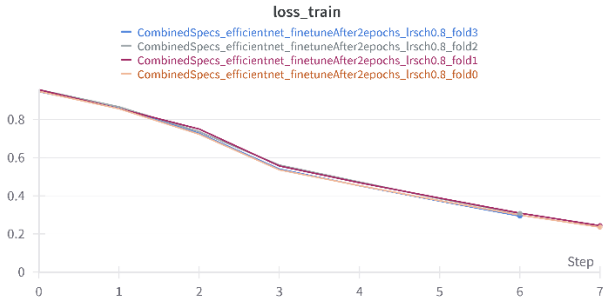
Exponential learning rate scheduler with gamma=0.8 was used. Starting learning rate is 10^{-3}.

For the first 2 epochs only the custom classifier linear layer weights were updated. After the first 2 epochs all layers of the model were finetuned.

The best scores for each fold are:

FOLD	VALIDATION SCORE	HIDDEN TEST SCORE
0	0.657	0.57
1	**0.584**	**0.50**
2	0.631	0.51
3	0.604	0.55
4	0.708	–

Training progress for each fold:



Note

STFT window of length 0.5 seconds for spectrograms was selected based on experiments with CNN architecture (see section 6). Better temporal resolution proved to be more informative than better frequency resolution.

Ensemble

Averaging the predictions of the best trained models from all 3 solutions has shown good results. The best configuration turned out to be:

	Models trained on folds	
WaveNet on longitudinal montage	1, 2	**0.37** *score on the hidden test set*
EfficientNet-B0 on provided spectrograms	2, 3	
EfficientNet-B0 on combined spectrograms	1, 2, 3	

5. Approaches that didn't work

Raw EEG features

WaveNet was unable to extract features for successful validation – while the training score improved with each epoch, validation score was very high.

Regional averages

WaveNet on 5 regional averages converged to validation scores >0.9.

An approach using spectrograms of regional pairwise coherence (see reference) failed due to constraints on the length of input data, 50 seconds were not enough to produce the proposed spectrograms.

Reference

Chapter 2 of the thesis: https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1966&context=theses

EfficientNet-B0 on computed spectrograms from longitudinal montage

The model overfit on the images of computed spectrograms, validation score was very high.

The size of the resulting images was 200x250 for STFTs of 5 seconds and 1 second, and 125x400 for STFT window of 0.5 seconds, which is closer to the size of the images (224x224) the model was trained on. But the sizes were smaller than 300x400 from the provided spectrograms where the model proved to be efficient.

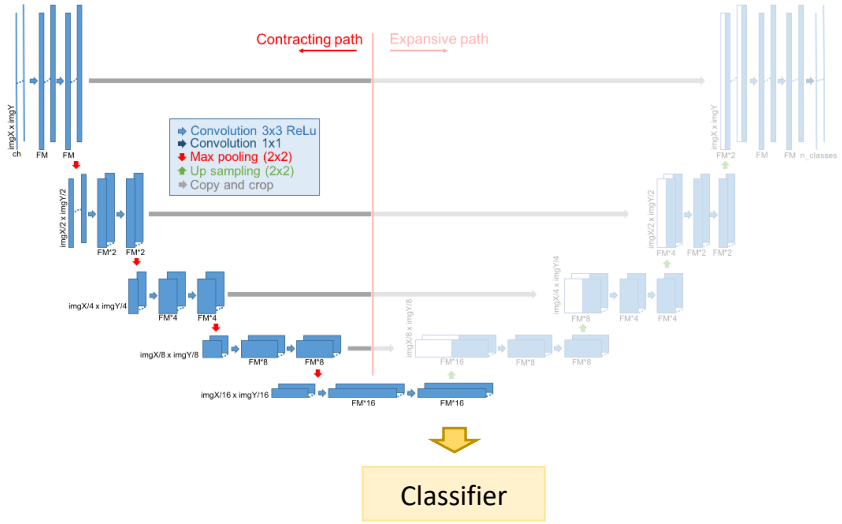
EfficientNet-B1

Version B1 has ~1 million more parameters than B0. A couple of experiments have shown that this increase in the model's complexity didn't improve the validation score.

6. Experiments with state-of-the-art CNN architecture

A state-of-the-art CNN model, the contracting path of the U-Net architecture, was applied to various configurations of spectrogram images.

CNN architecture:



All of the experiments failed to outperform the EfficientNet-B0 model by a significant margin of at least ~0.1, but due to the transparency of the architecture a lot of valuable insights were obtained.

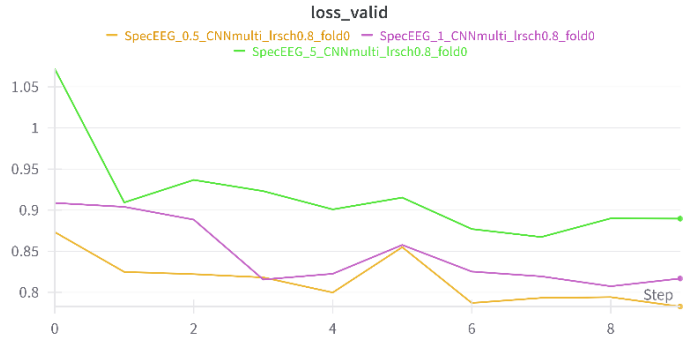
Several series of experiments were performed:

A. Input as a single-channel image, with spectrograms concatenated into a single channel.
B. Input as a multi-channel image, with each spectrogram processed by the model independently and the features from each channel concatenated before being passed to the classification head.
C. Input as a multi-channel image, with each spectrogram as a channel.

Insights

1. **It is important to obtain features from all spectrograms separately,** averaging across all spectrograms decreases the performance*:* approaches A and B have shown similar validation scores, while the approach C has shown worse validation scores.
2. Increase in the number of initial convolutions from 16 to 32 improves the performance, but it exponentially increases the number of parameters. CNN with 64 initial convolutions overfits.

3. Experiments on spectrograms computed from longitudinal montage have shown that **a better temporal resolution is more important than a high frequency resolution**: spectrograms with STFT 0.5 seconds outperformed spectrograms with STFT 1 second, and spectrograms with STFT 1 second outperformed spectrograms with STFT 5 seconds.



4. Ways of normalizing the spectrograms didn't affect the score.
5. Reducing learning rate made the validation curves smoother but it stopped improving at a worse level.
6. Making the classifier more complex – a sequence of fully-connected layers with ReLU activations instead of a single fully-connected layer – didn't improve the validation scores.