TRANSFORMING THE BRAIN: A Hybrid CNN-Transformer Approach for Diagnosing Harmful Brain Activity

Amit Erraguntla, Aviral Vaidya, and Sean Kim

Introduction:

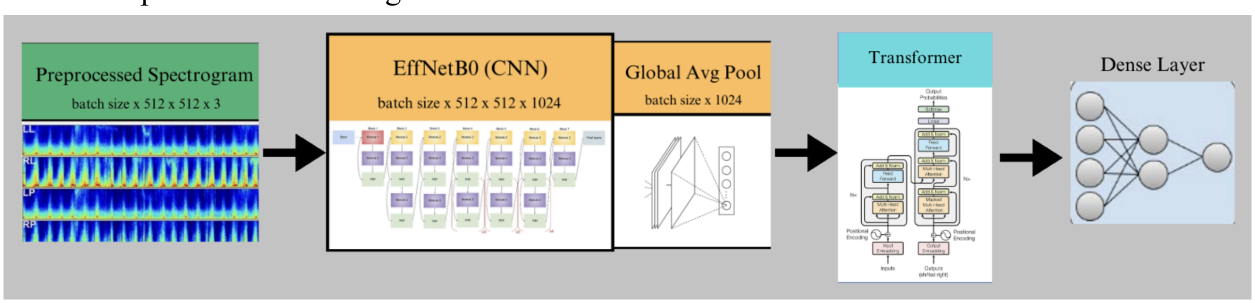
An electroencephalogram (EEG) is a crucial tool for providing insights into a brain by using time series electrical activity at different nodes (see fig. 1).

Our goal was the following: Create a model that can output a probability distribution for various classes of harmful brain activity: SZ (seizure), LPD (lateralized periodic discharge), GPD (generalized periodic discharge), LRDA (lateralized rhythmic delta activity), GRDA (generalized rhythmic delta activity), and OTHER.

This is important because manual review of EEG recordings is time-intensive and expensive. In addition, results are subjective and can vary depending on the reviewer. We hope to improve EEG pattern classification accuracy and allow doctors and researchers to detect harmful brain activity in a faster manner.

Methodology & Preprocessing + Architecture:

We came up with the following model architecture:



Our data is publicly available from Harvard Medical School (HMS), in the form of 50 second EEG and spectrogram data.

For each input, we have two spectrograms, the spectrogram provided in the HMS dataset and a spectrogram we generate ourselves. We use the raw EEG waveform time-series data and apply a wavelet + Fast Fourier transform before generating the spectrogram using the librosa library. Finally, We log transform and normalize both spectrograms before concatenating them together to pass into our model.

After our preprocessing, the rest of our model architecture is as follows:

The preprocessed and concatenated input spectrograms pass through EfficientNetB0, a CNN architecture initialized with pre-trained weights from the ImageNet dataset. This creates 1024 channels for each input which would contain important features of the spectrogram. This goes through a Global Average 2D Pooling layer which turns each channel into a single value. Lastly, This 1024-length sequence goes through a transformer and then finally a fully connected layer with softmax activation.

Our transformer uses multi headed attention in order to capture the relationships between the channels created by the CNN, with added dropout layers to prevent the model from overfitting.

The output is a probability distribution of the six categories: SZ, LPD, GPD, LRDA, GRDA, and OTHER. We use KL Divergence as our loss function to measure the difference between our predicted and actual probability distributions.

Results:

Below is a comparison of our model's performance compared to others:

Model	KL Divergence
Fourier Transform + ViT	0.318432
ViT Pre-Trained Vision Transformer	0.457420
CNN + LSTM + Multimodal Input	0.500874
CNN + Transformer	**0.550406**
EEG + Spectrogram + 2D CNN	0.610243
CNN + LSTM with only EEG data	1.101587
1D CNN (Time Series & Image)	1.106557

Our model achieves comparable performance to most other approaches, outperforming LSTMs and CNN based approaches, but underperforming against multimodal and vision transformer approaches.

For further context, on the left are two of our model's predictions compared to the ideal output. Notice the similarity in the distributions.

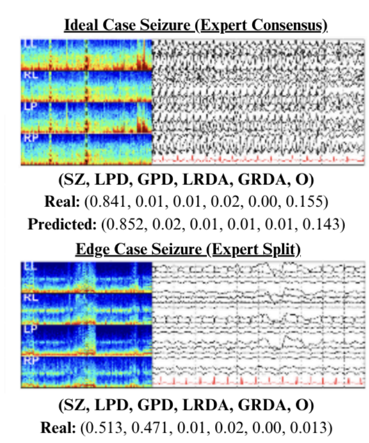


Fig 6: Real and predicted outputs for various inputs.

Challenges:

While designing our project, we ran into several challenges.

Firstly, Our dataset was large and our model architecture was very complex with many trainable parameters leading to long training times. In order to stop running out of RAM and CPU power, we loaded data and trained the model in batches, saving the weights in between.

In addition, the EEG data was difficult to work with and we had trouble with the feature extraction we had initially proposed. While we were unable to extract features for a multimodal approach, we performed heavy data augmentation and created multiple spectrograms for each input.

Reflection:

Overall, we felt like our project was a success. While we did not meet our stretch goal of improving performance compared to all other models on our task, our accuracy was comparable to most advanced approaches, and actually outperformed some of them.

Initially, we had planned to design a multimodal model for our task, but ended up not doing so due to the unexpected complexity of our dataset. The feature extraction we had planned to design for the other modality would have required much more time, since our EEG data was extremely hard to deal with. Preprocessing itself was a lot of work. However, if we had more time, we would have certainly tried to implement some sort of additional feature extraction from our EEGs/Spectrogram comparable to the features doctors look at when they analyze spectrograms and EEGs manually. In addition, with more time we would look into how to implement additional metrics to judge the accuracy of our model, such as measuring Type I and Type II errors, which are difficult to judge when we output a distribution. This would help us gain additional insight into our performance. A possible idea we had was instead of outputting a distribution, we could simply take our greatest vote and compare it to the actual data and judge our errors based on this.

Our biggest takeaway from this project is that real datasets are HUGE. Compared to what we deal with in class, when people build models, they need to figure out ways of simply training the models such that their machine can handle it. We had to learn how to load and train our data in batches for this. In addition, data is far from perfect. Most datasets will be messy and hard to deal with, which is something we certainly experienced during this project. Preprocessing, cleaning, and augmentation can be a huge portion of the work.

Github Repository:
https://github.com/asaiie/multimodal-seizure-detection

Citations:

Data obtained from:
https://www.kaggle.com/competitions/hms-harmful-brain-activity-classification/data

Spectrogram generation methodology:
https://www.kaggle.com/competitions/hms-harmful-brain-activity-classification/discussion/469760

Dataloader segmentation:
https://www.kaggle.com/code/awsaf49/hms-hbac-kerascv-starter-notebook