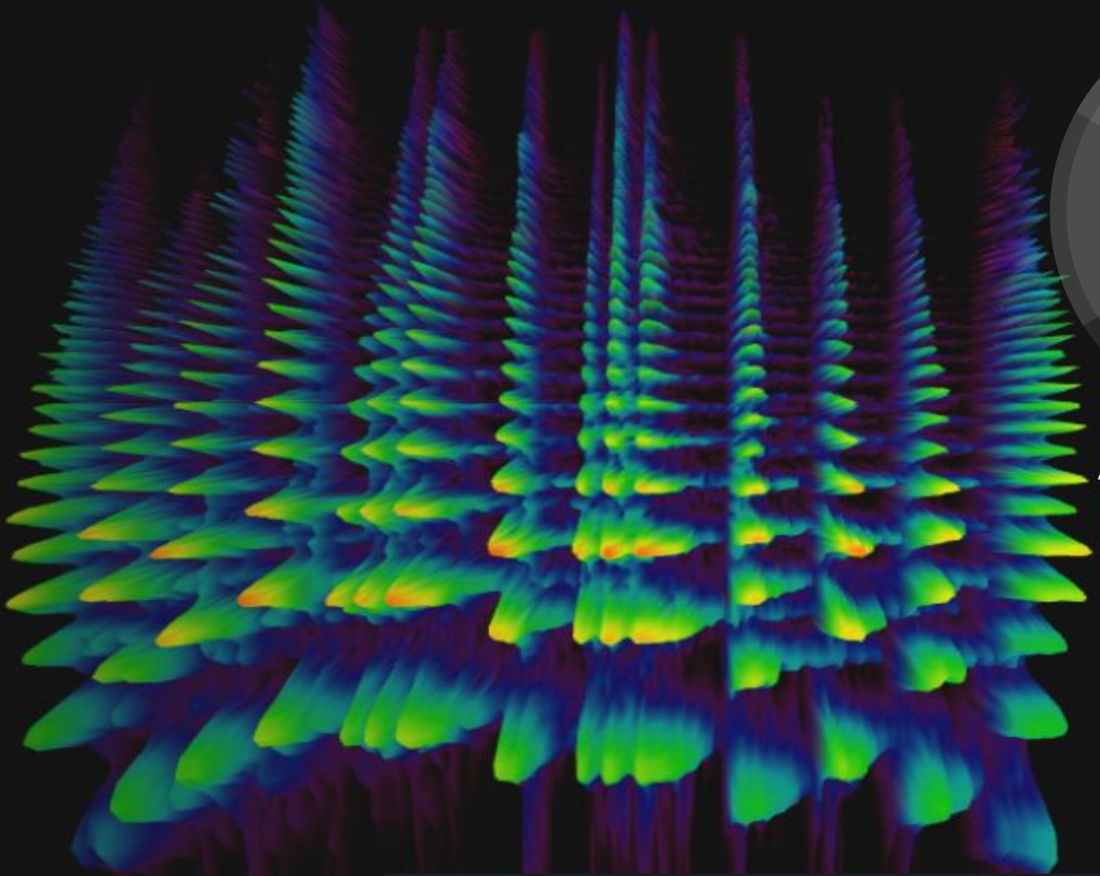


Erdős
Deep Learning
Project
Summer
2024

A Vocal-Cue Interpreter for Minimally Verbal Individuals

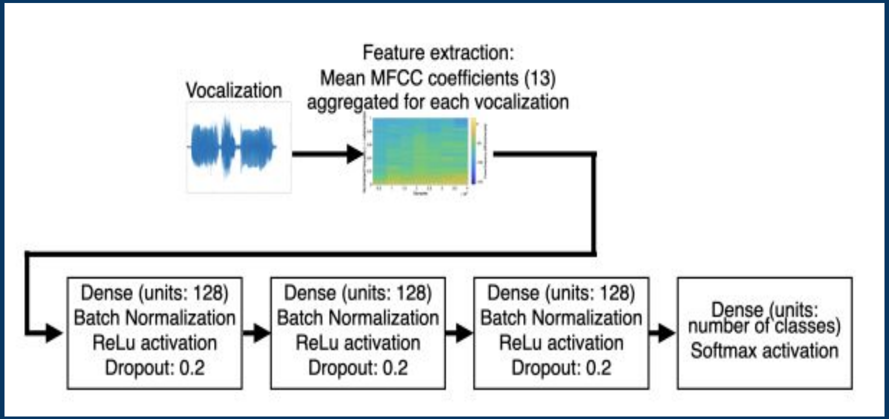
The Team: Julian Rosen, Alessandro Malusà, Monalisa Dutta, Rahul Krishna, Atharva Patil & Sarasi Jayasekara

[The image is from MusicLab]

Motivation

Our work is building upon the results from a 2021 paper titled "Transfer Learning with Real-World Nonverbal Vocalizations from Minimally Speaking Individuals"

Their best model:



[Image from the Original Paper]

ReCanVo Dataset

The data was collected by the authors of the original paper, with 8 minimally verbal individuals. The audio was recorded in long sessions, that then were broken into clips and labeled.

The audio samples that were collected, were then labeled by family members, or caretakers of the participant.

Labels: Happy, Dysregulated, Hungry, ...

Our Approach

- We wanted to train a model for an individual at a time
- We focused on participants 01 and 05
- For each participant,
    - We dropped labels that had fewer than 30 data points
    - Training / validation data split was done with one session being held out as the validation set
- We experimented with adding extra layers of background noise to the files in training data

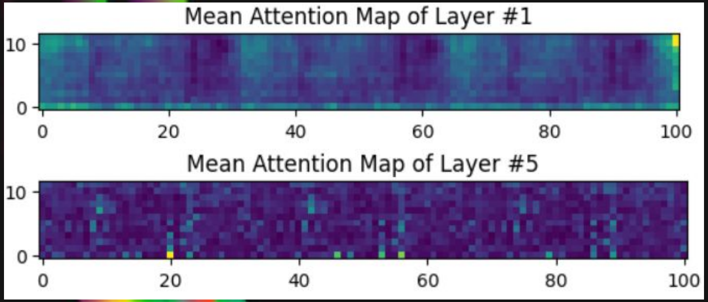
Pipeline

.WAV files → Model features → Label prediction

Pretrained audio models:
- HuBERT (CNN+attention)
- AST (attention)

Direct processing:
- Mel spectrograms

Classifiers:
- Convolutional NN
- Fully connected NN
- Traditional ML classifiers (e.g. XGB)



More on Feature Extraction

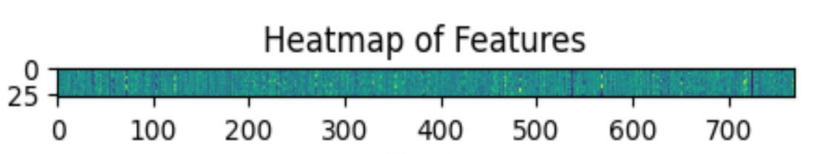
HuBERT

Architecture:

- CNN Encoder + Attention Layers

Data preprocessing:

- Features extracted from HuBERT are a list of 12 tensors. We choose the first tensor among them, and average over the time dimension.



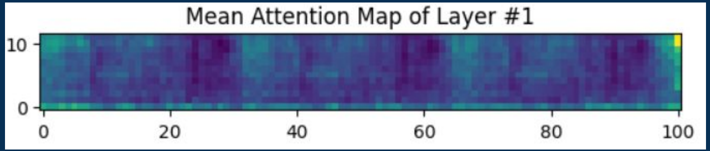
AST

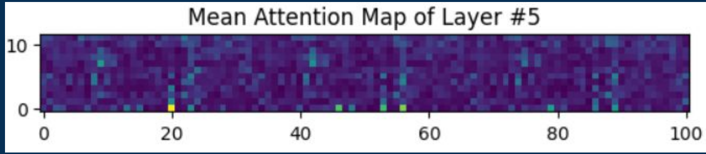
Architecture:

- Purely Attention Layers

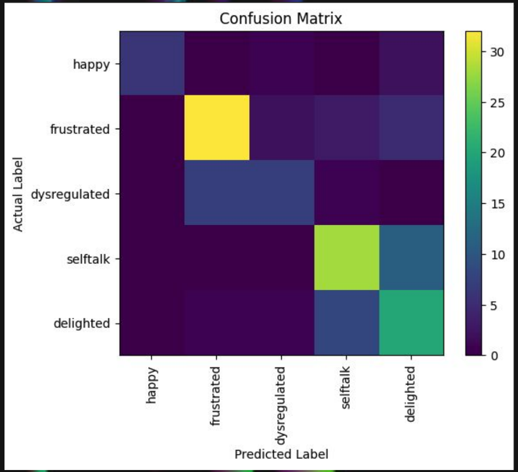
Data preprocessing: We used the output of several different layers including

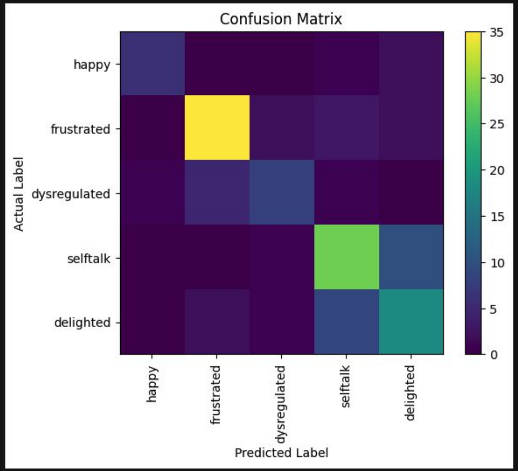
- The initial layer (with entries averaged over one of the dimensions)
- The 1st and 5th Attention Layers





Combatting Overfitting





Outcome is often only a slight change in confusion matrix, but every bit helps!

Classifier has many parameters relative to size of dataset.

Typical problem, typical techniques:

- Early stopping
- Penalizing weights
  - Ridge (L2)
- Dropout
  - On each training epoch, select nodes randomly to omit from network.

More specific overfitting issues as well.

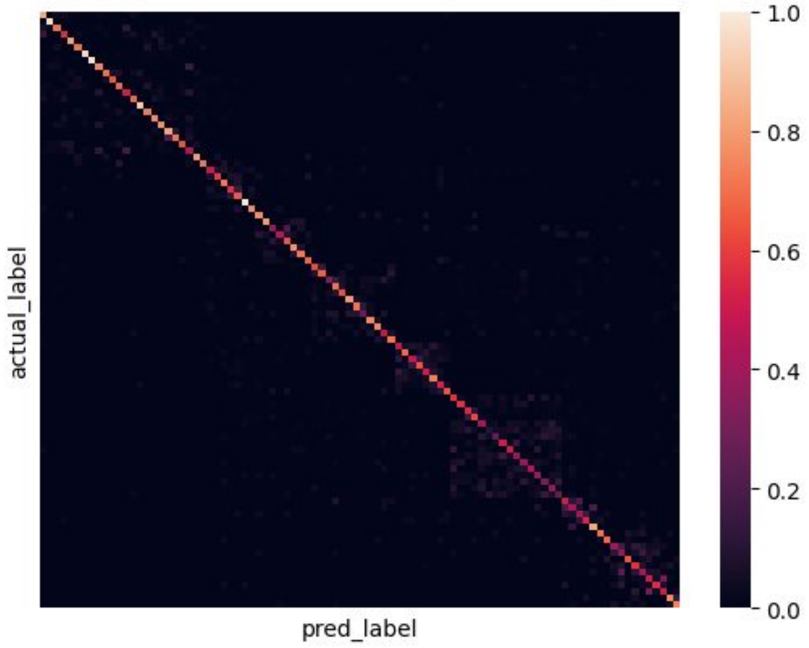
Unintended Session Learning

	No weight	Session weight	Session and label weight
accuracy	0.610	0.642	0.520
balanced_accuracy	0.501	0.520	0.520
unweighted_f1	0.516	0.540	0.486
UAR	0.501	0.520	0.520
logloss	1.075	1.037	1.251

Performance with fully randomized cross validation

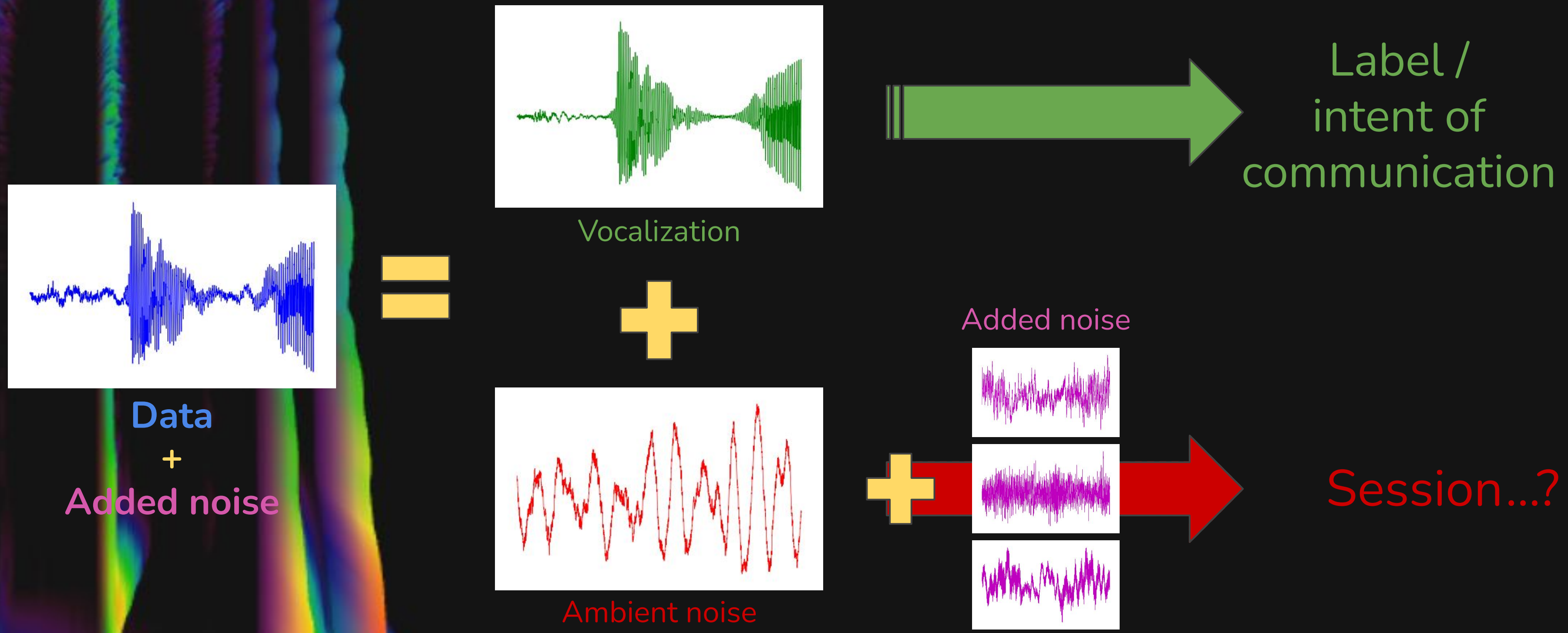
	No weight	Session weight	Session and label weight
accuracy	0.514	0.510	0.449
balanced_accuracy	0.442	0.449	0.449
unweighted_f1	0.452	0.459	0.440
UAR	0.442	0.449	0.449
logloss	1.314	1.259	1.495

Performance with session holdout cross validation



Confusion matrix of the session classifier

Adding Noise

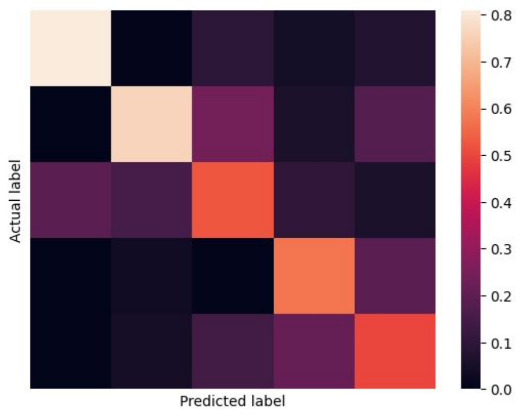


Added noise from DEMAND dataset

Adding Noise

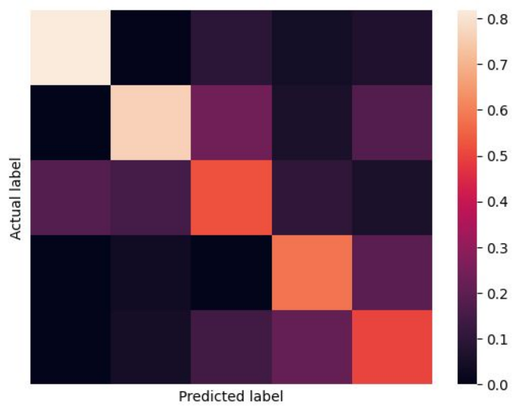
Base model – no added noise

	No weight	Session weight	Session and label weight
accuracy	0.613	0.644	0.527
balanced_accuracy	0.510	0.527	0.527
unweighted_f1	0.524	0.548	0.496
UAR	0.510	0.527	0.527
logloss	1.063	1.025	1.235



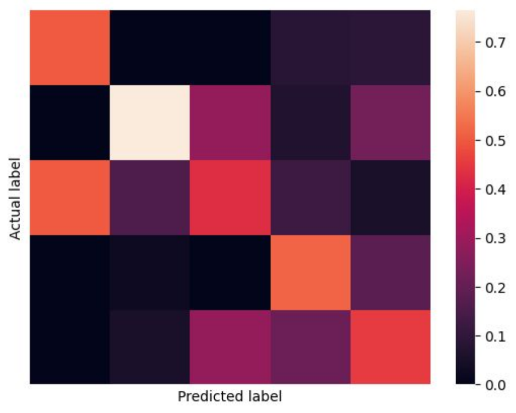
One added noise, randomly selected from the entire set

	No weight	Session weight	Session and label weight
accuracy	0.614	0.645	0.530
balanced_accuracy	0.514	0.530	0.530
unweighted_f1	0.529	0.551	0.499
UAR	0.514	0.530	0.530
logloss	1.063	1.025	1.234



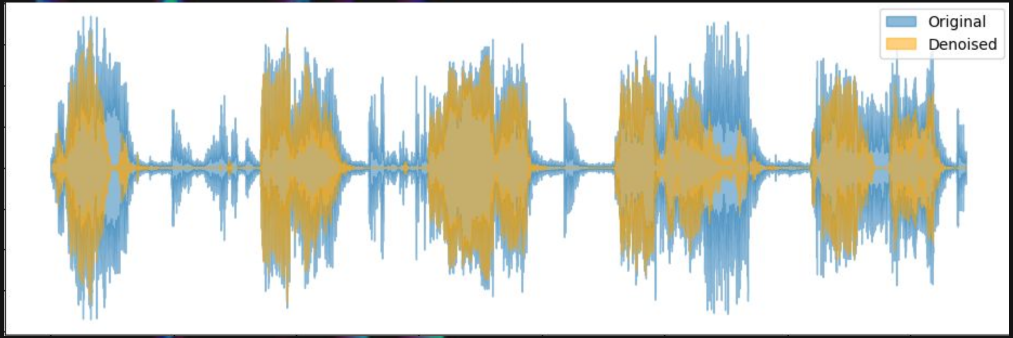
Random number of added noises, only from the class “DLIVING”

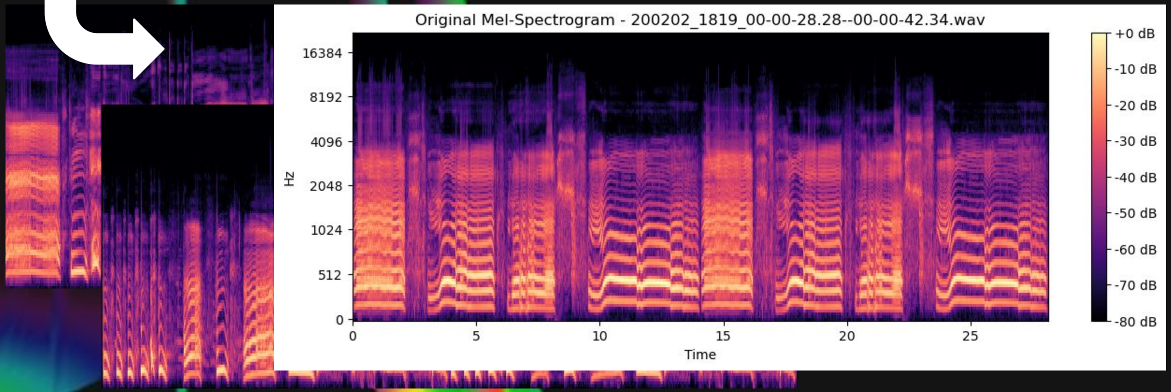
	No weight	Session weight	Session and label weight
accuracy	0.561	0.602	0.435
balanced_accuracy	0.411	0.435	0.435
unweighted_f1	0.386	0.409	0.346
UAR	0.411	0.435	0.435
logloss	1.139	1.101	1.334

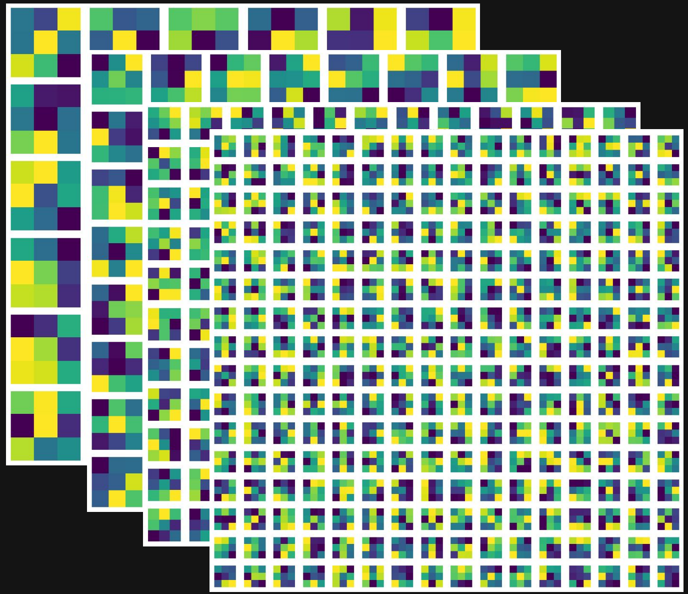


Added noise from DEMAND dataset

“Noise” Cancellation

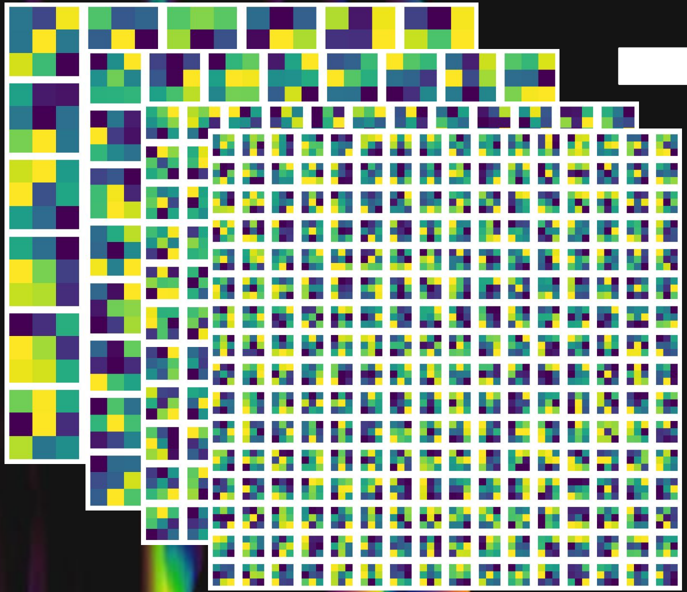


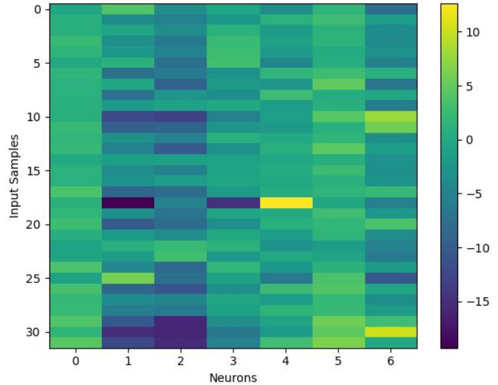


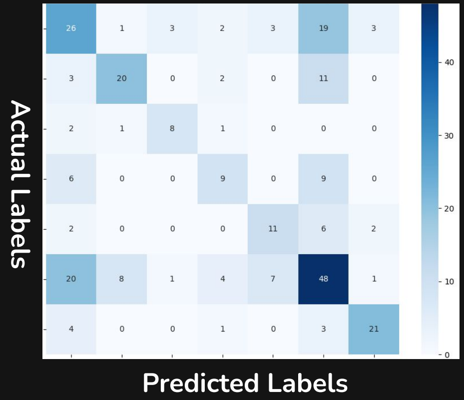


Noise-Cancellation encoder-decoder model: [denoiser]

“Noise” Cancellation





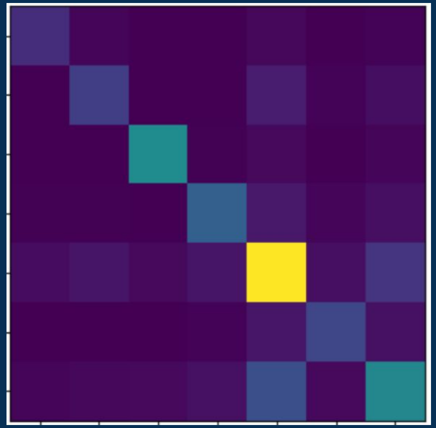


F1 score P01	Mel Spec only	Hubert+FC
Raw	0.54	0.77
De-noised	0.45	0.73

Results for Participant 01

Feature Extractor	Classifier	F1 Score
HuBERT	1 dense layer (with penalty)	0.793
HuBERT	2 dense layers	0.793
HuBERT	XGBoost	0.762
AST	XGBoost	0.707
AST	1 dense layer	0.698
Mel Spectrograms	4 CNN layers	0.535

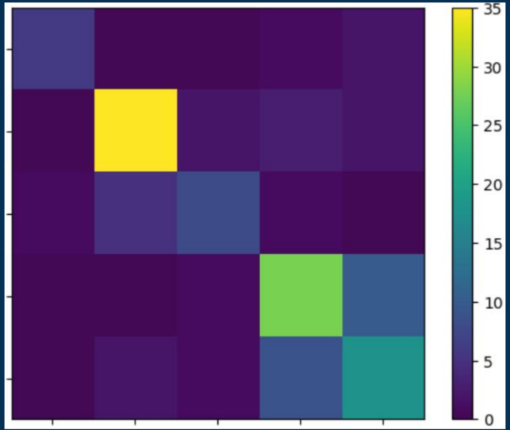
Best Performing Model:
HuBERT + 1 dense layer
Confusion Matrix on the Test Set



Results for Participant 05

Feature Extractor	Classifier	F1 Score
HuBERT	2 dense layers	0.627
HuBERT	1 dense layer (with penalty)	0.619
HuBERT	XGBoost	0.603
AST	1 dense layer	0.548
Mel Spectrograms	4 CNN layers	0.472

Best Performing Model:
HuBERT + 2 dense layers
Confusion Matrix on the Test Set



Conclusions

And Observations

- On the test sets for participants 01 and 05 respectively, the best performing model displayed F1 scores of 0.712 and 0.582, both of which are improvements on the original team's results that had inspired us.
- HuBERT + a few extra layers fine tuned, worked best for the participants we considered.

Further Directions

- Combine our "noise engineering" methods with more architectures
- Attempt classification by broader label classes, e.g., by sentiment (positive vs. negative) and energy level (high vs. low).
- Build a model that can be generally trained and then be fine tuned for each individual

Special Thanks to:

Roman Holowinsky – Director of the Erdös Institute
Kristy Johnson – For Advice
Lindsay Warrenburg – Lead Instructor, DL
Marcos Ortiz – Lead TA, DL

On behalf of the Our Team - Julian, Ale, Rahul, Atharva, Monalisa, and Sarasi.