Speech and Machine Learning for Neurodegenerative Disease Analysis

Yi-Ting Tsai

Apr. 07, 2022

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Overview

- **Background**
- **Datasets**
- **Method 1: Speech analysis with GMM-UBM**
  - Approach and results
- **Method 2: Speaker Recognition Based neural network model – x-vector**
  - Approach and results
- **Method 3: Accent recognition based neural network model – CNN and x-vector**
  - Approach 1: Not finetuned + Leave-one-out
  - Approach 2: Siamese network finetuned + 10-fold

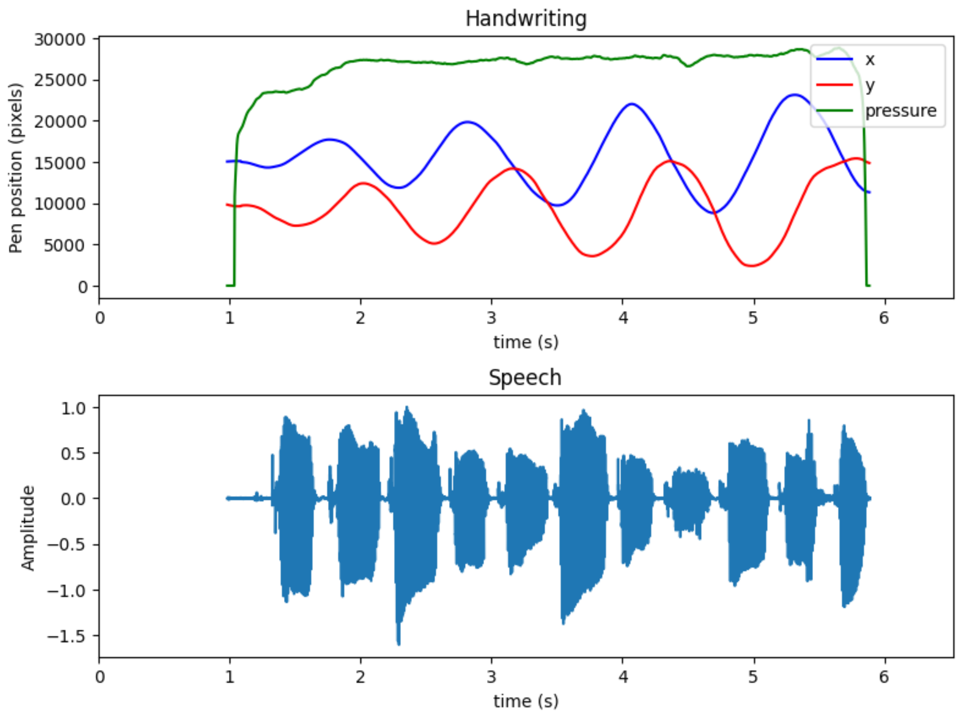
Background – Parkinson’s (PD)

- Past machine learning approaches
  - Detection of detection of dysphonia and hypokinetic dysarthria.
  - Numerous studies have used Gaussian Mixture Model (GMM)-Universal Background Model to classify PD from healthy controls – 80% +- accuracy on common Spanish corpus
  - Text-Dependent Utterance (TDU), Diadochokinetic (DDK), monologues and sustained vowel
- Deep learning approaches (also for Alzheimer's - AD)
  - Classification based on LSTM (TDU,DDK), CNN (sustained vowels)
  - Speaker recognition based models to extract speaker embeddings then do classification
- Challenges
  - Data size
  - Very different performance across different corpus and speech tasks

Datasets

- **NLS dataset** recorded along with

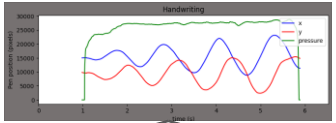
  handwriting (11 kHz), eyetracking (24 kHz)
  - DDK: /Pa-ta-ka/
  - Text-dependent: rainbowpassage
  - Text-independent: wordcolor, cookiethief
- **Neurovoz dataset** used to validate our methods
  - DDK: /Pa-ta-ka/
  - Text-dependent: reading a passage



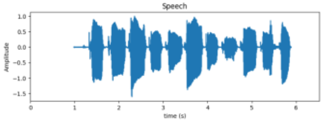
Johns Hopkins Whiting School of Engineering

Method 1: Speech analysis with GMM-UBM

Handwriting signals



Speech signals



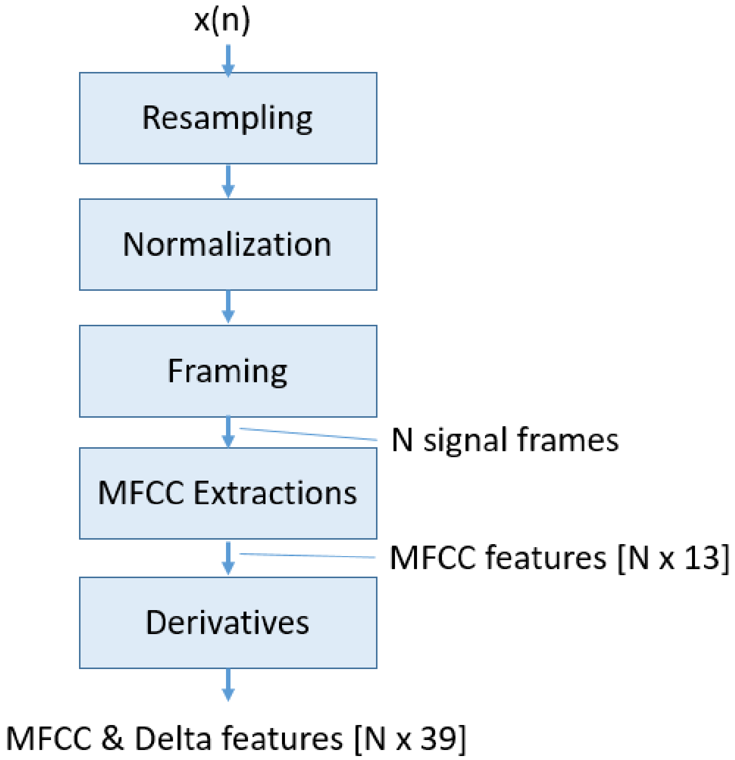
- Handwriting signals → Feature extraction using Matlab digital biomarker library → Classifier - XGBoost
- Handwriting signals → Feature extraction using TSFresh → Classifier - XGBoost
- Speech signals → Feature extraction using MFCCs & Deltas → Classifier - Gaussian Mixture Models (GMM-UBMs)

All classifiers → Fusion model (Logistic Regression) → Classification: PD, ND, CTRL

Method 1: Speech analysis with GMM-UBM

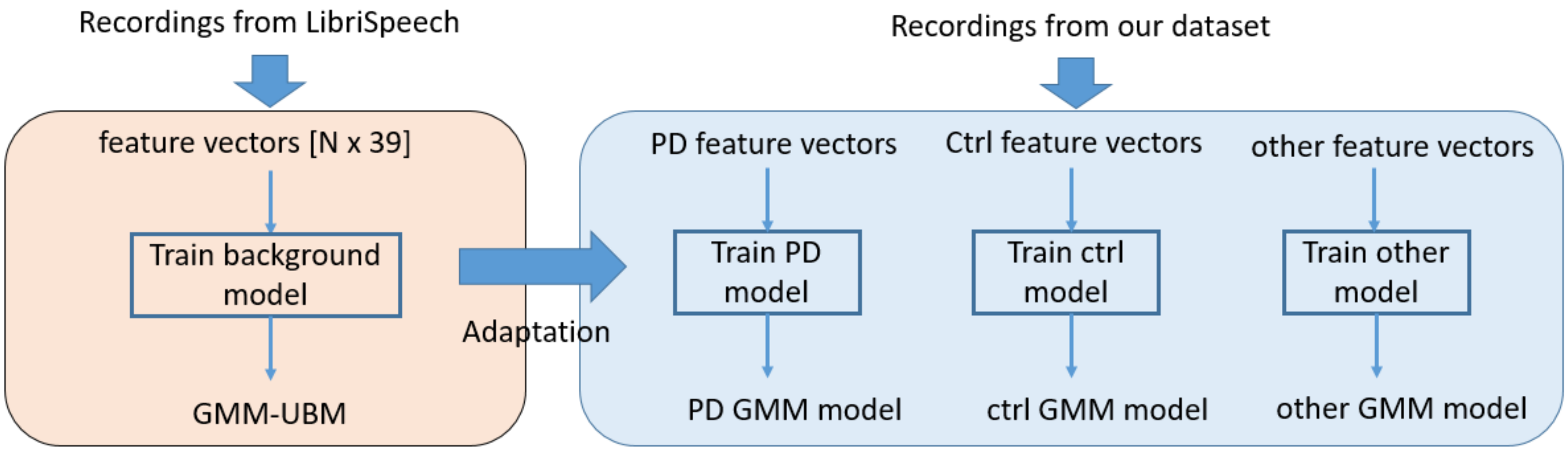
Feature Extraction

- MFCCs and their first and second derivatives Delta and Delta 2 are extracted.
- Extractions are done using the Python Librosa toolbox.
    - Firstly, all signals are resampled to 22050 Hz and normalized.
    - 13 MFCC coefficients, and frame size of 512 with a 50% frame overlap.
- Feature vectors of dimension 39 are extracted.



Method 1: Speech analysis with GMM-UBM

- Data loaded and cleaned, MFCC & Delta features extracted for each speech recording
- Background model (UBM) trained using speech recordings from the LibriSpeech ASR corpus + Neurovoz (total duration of 292.2 mins used)
- GMM-UBM trained for each class using mean-only Maximum A Posteriori (MAP) adaptation of UBM (relevance factor = 16, number of mixtures = 16)



Method 1: Speech analysis with GMM-UBM

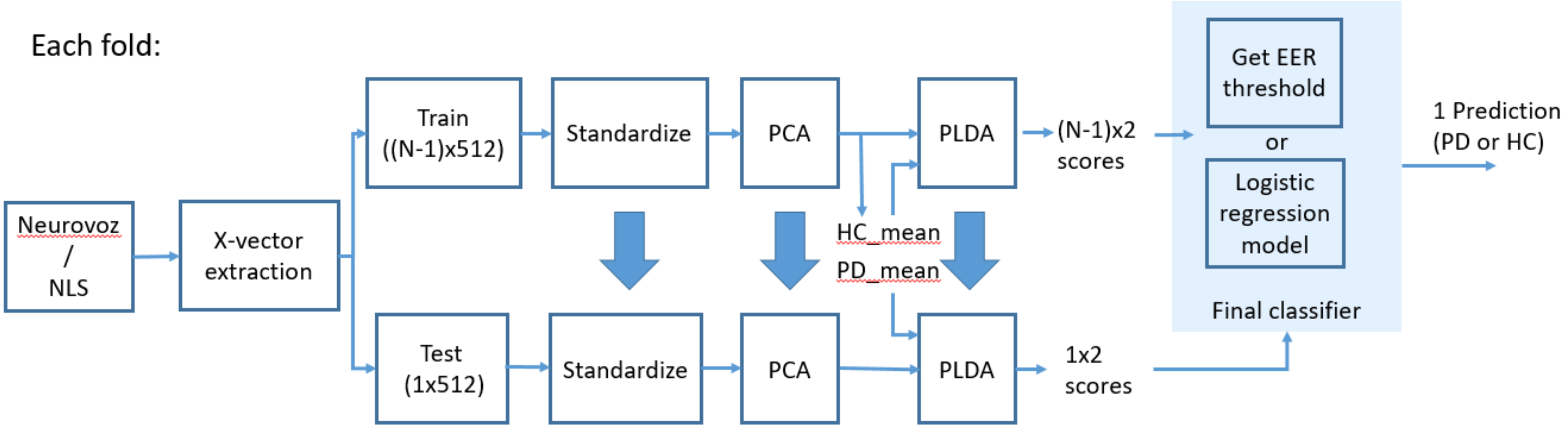
NLS DDK (12 controls, 21 PD, and 21 other at the time) experiment results using leave-one-out cross-validation

		PD vs ND vs CTRL	PD vs ND	PD vs CTRL	PD vs ND + CTRL	PD + ND vs CTRL
Speech only	Accuracy	38.89%	52.38%	39.39%	51.85%	63%
	F1 macro score	0.378	0.519	0.365	0.484	0.518
Handwriting only - TSFresh	Accuracy	40.35%	50%	50%	56.14%	73.68%
	F1 macro score	0.3855	0.4905	0.4667	0.5	0.591
Handwriting only - biomarker	Accuracy	29.82%	36.36%	68.75%	48.29%	75%
	F1 macro score	0.276	0.3418	0.6761	0.3532	0.604
Fusion - TSFresh features + speech	Accuracy	-	-	-	42.60%	74.07%
	F1 macro score	-	-	-	0.378	0.57
Fusion - biomarker features + speech	Accuracy	-	-	-	68.50%	69%
	F1 macro score	-	-	-	0.498	0.497
Fusion - All features + speech	Accuracy	25.93%	-	-	48.15%	72%
	F1 macro score	0.253	-	-	0.378	0.521

Method 1: Speech analysis with GMM-UBM

- GMM log-likelihood ratio classification with Neurovoz (PD 44, Ctrl 42)
  - Accuracy = 74.42%, F1 macro score: 0.7436
  - Past research with GMM-UBM + Rasta-PLP features best reaches 84±7% accuracy
- Our features and methods work much better in previously researched datasets, potential reasons?
  - Language, PD severity, face masks, instructions
  - Multitasking results in inconsistent patterns (eg. some controls speak slower than they should be able to too)
- The same trend exists in Method 2 and Method 3 as well

Method 2: Speaker Recognition Based neural network model – x-vector

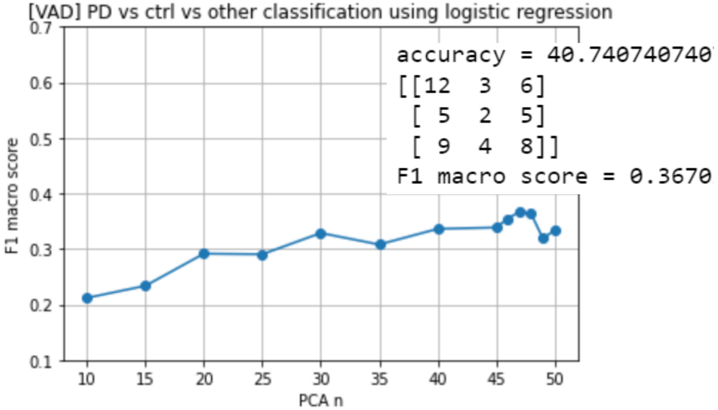


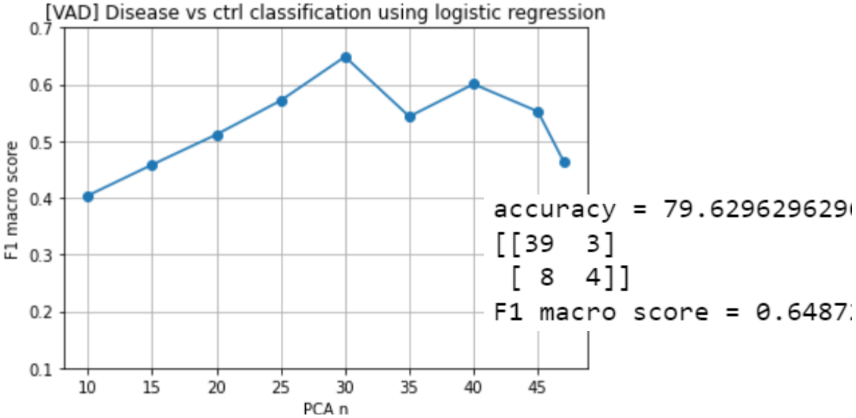
- Leave-one-out cross-validation
- X-vector: deep neural network (DNN) embeddings for speaker recognition, extracted using a pre-trained x-vector network (3.2% error rate to recognize thousands of speakers)
- PCA and PLDA: feature dimension reduction, then group transformed speaker embeddings of the same class together

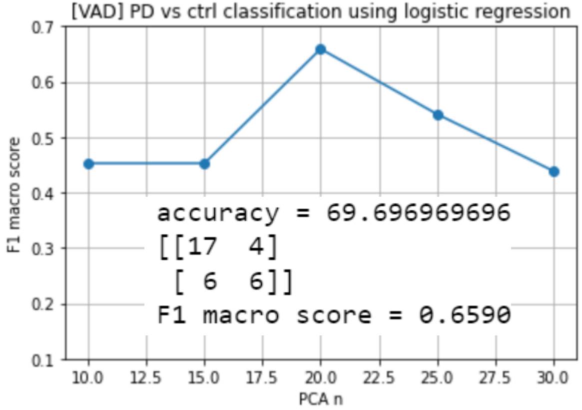
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

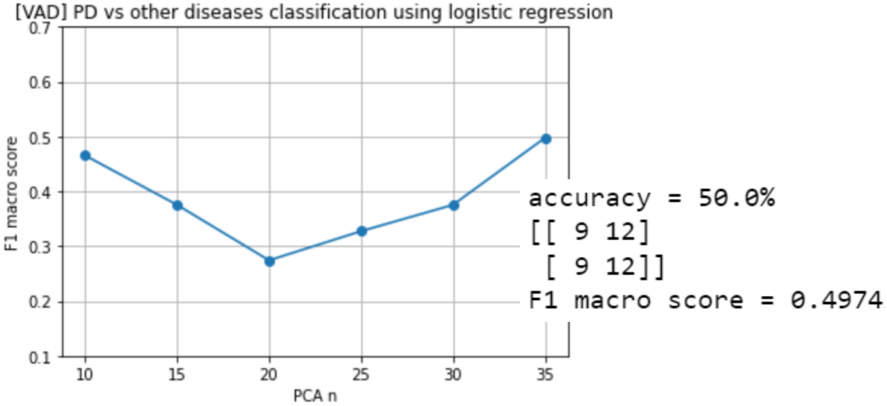
Method 2: Speaker Recognition Based neural network model – x-vector

NLS pataka results: 54 [PD-21, CTRL-12, AD-4, other-17]

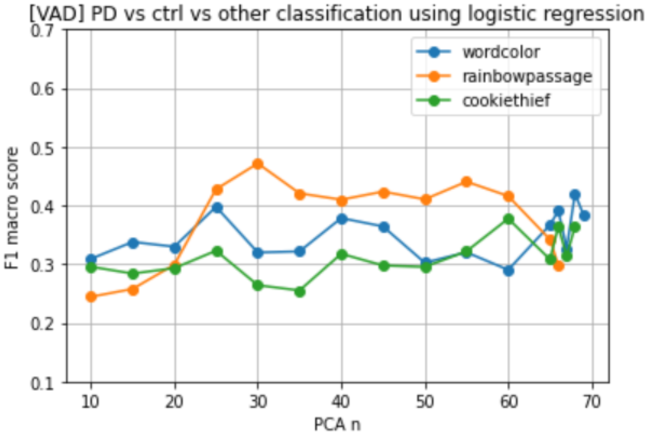


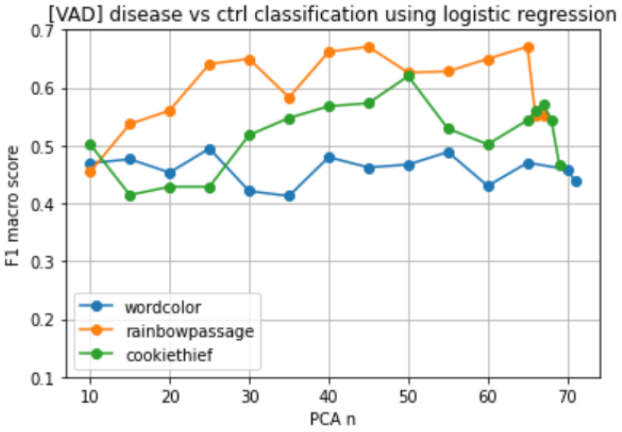


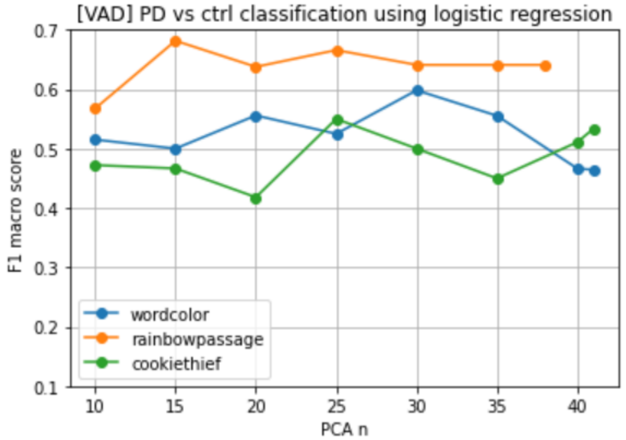


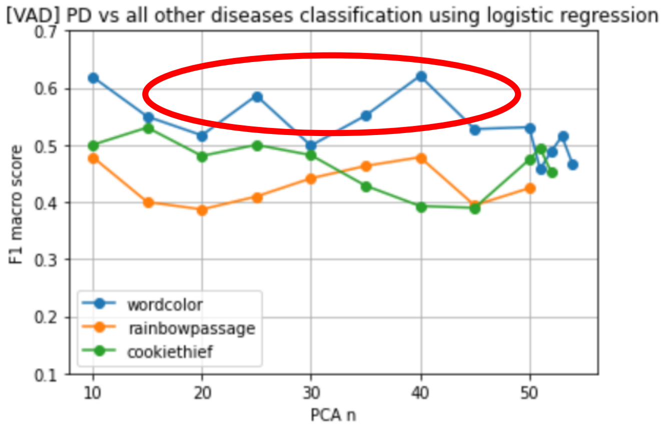


Method 2: Speaker Recognition Based neural network model – x-vector









NLS wordcolor:
75 [PD-28, CTRL-17, AD-4, other-26]

NLS rainbowpassage:
71 [PD-25, CTRL-17, AD-4, other-25]

NLS cookiethief:
73 [PD-28, CTRL-17, AD-4, other-24]

Method 2: Speaker Recognition Based neural network model – x-vector

```
accuracy = 42.666666666
[[12  7  9]
 [ 4  7  6]
 [ 9  8 13]]
F1 macro score = 0.4200
```

```
accuracy = 47.887323943
[[ 9  4 12]
 [ 0  8  9]
 [ 6  6 17]]
F1 macro score = 0.4715
```

```
accuracy = 36.986301369
[[12 10  6]
 [ 5  9  3]
 [ 9 13  6]]
F1 macro score = 0.3636
```

```
accuracy = 60.0%
[[15 13]
 [ 5 12]]
F1 macro score = 0.5982
```

```
accuracy = 69.047619047
[[18  7]
 [ 6 11]]
F1 macro score = 0.6816
```

```
accuracy = 53.333333333
[[12 16]
 [ 5 12]]
F1 macro score = 0.5333
```

```
accuracy = 58.666666666
[[38 20]
 [11  6]]
F1 macro score = 0.4946
```

```
accuracy = 71.830985915
[[39 15]
 [ 5 12]]
F1 macro score = 0.6706
```

```
accuracy = 71.232876712
[[44 12]
 [ 9  8]]
F1 macro score = 0.6198
```

NLS wordcolor:
75 [PD-28, CTRL-17, AD-4, other-26]

NLS rainbowpassage:
71 [PD-25, CTRL-17, AD-4, other-25]

NLS cookiethief:
73 [PD-28, CTRL-17, AD-4, other-24]

```
accuracy = 62.068965517
[[18 10]
 [12 18]]
F1 macro score = 0.6206
```

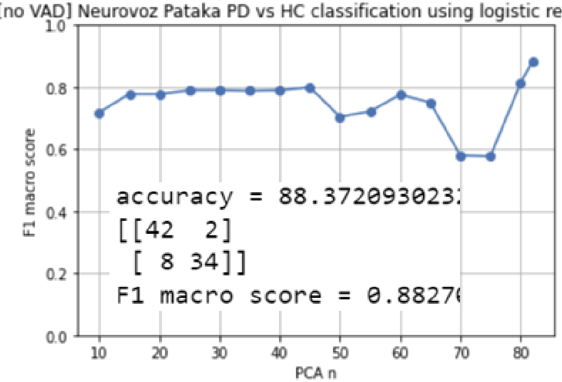
```
accuracy = 48.148148148
[[11 14]
 [14 15]]
F1 macro score = 0.4786
```

```
accuracy = 53.571428571
[[12 16]
 [10 18]]
F1 macro score = 0.5303
```

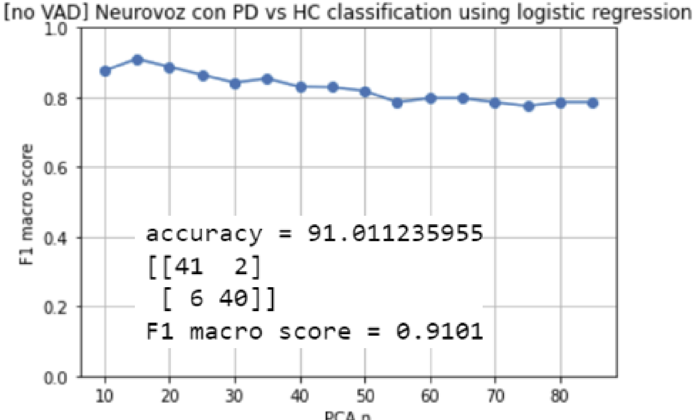
Method 2: Speaker Recognition Based neural network model – x-vector

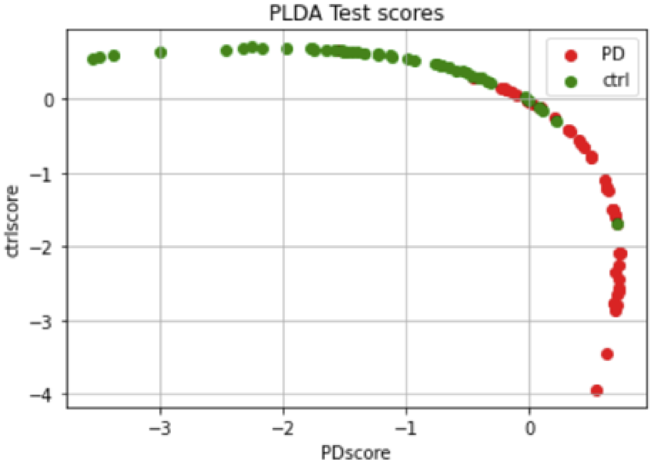
Test scores plotting & Neurovoz with limited data comparison

Neurovoz pataka results: 86 [PD-44, CTRL-42]



Neurovoz con results: 89 [PD-43, CTRL-46]



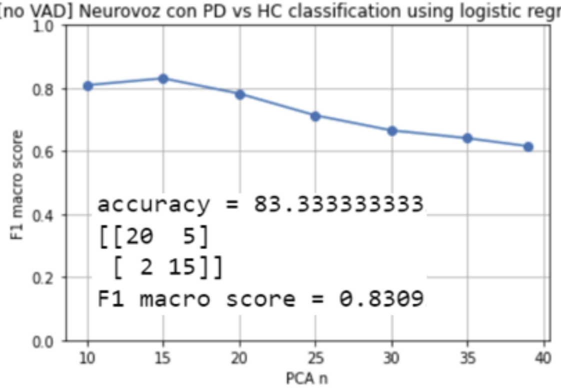


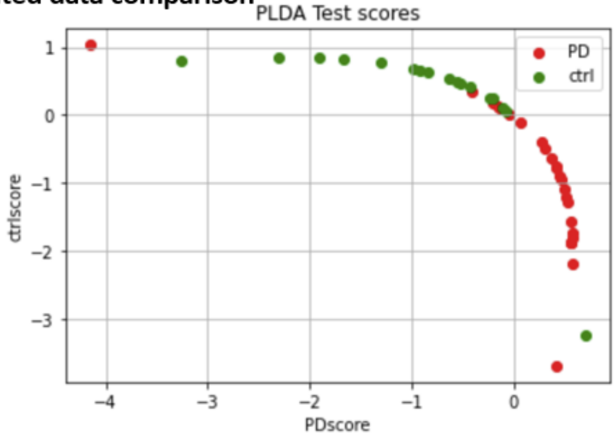
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Method 2: Speaker Recognition Based neural network model – x-vector

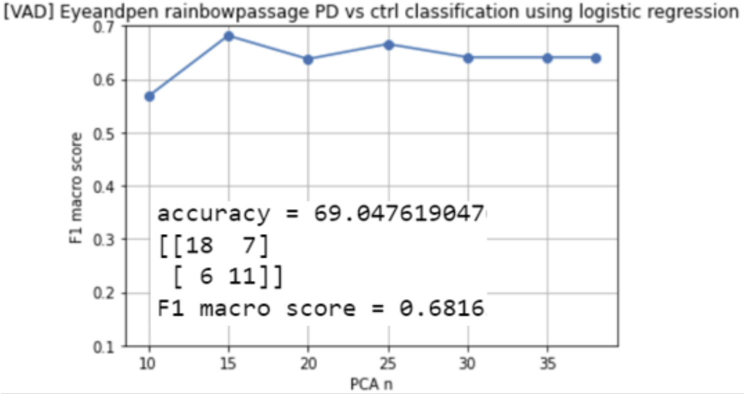
Test scores plotting & Neurovoz with limited data comparison

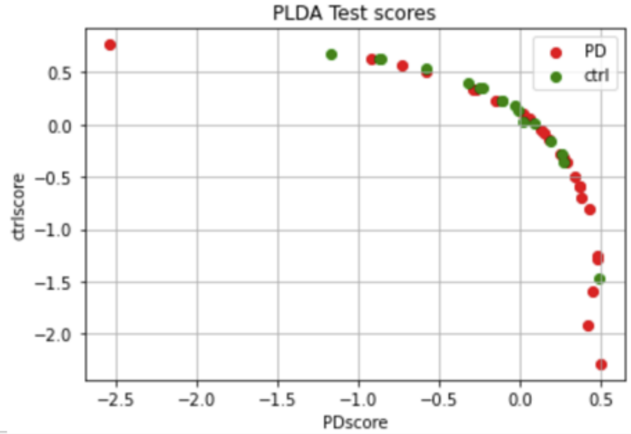
Neurovoz con results with limited data: 42 [PD-25, CTRL-17]





NLS rainbowpassage results: 71 [PD-25, CTRL-17, AD-4, other-25]



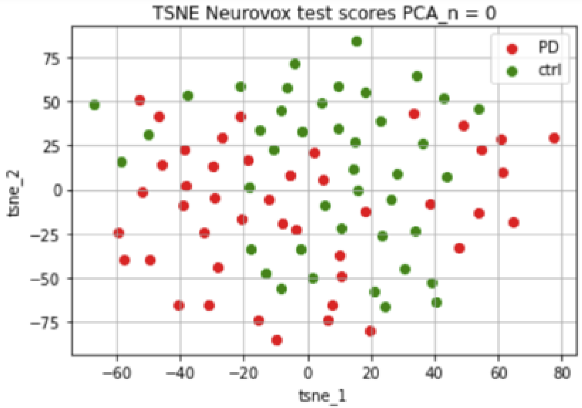


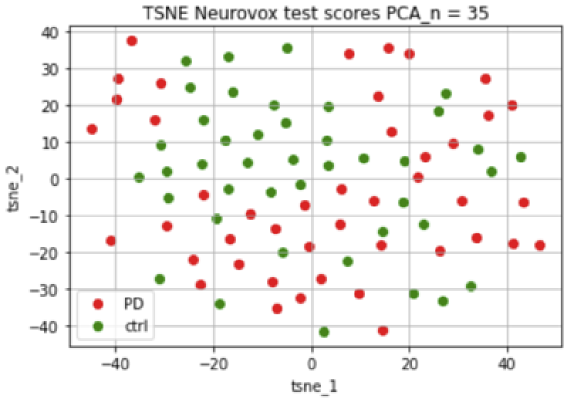
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Method 2: Speaker Recognition Based neural network model – x-vector

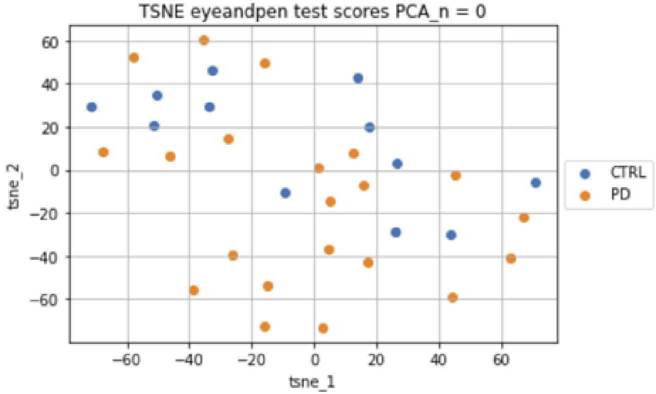
TSNE experiments

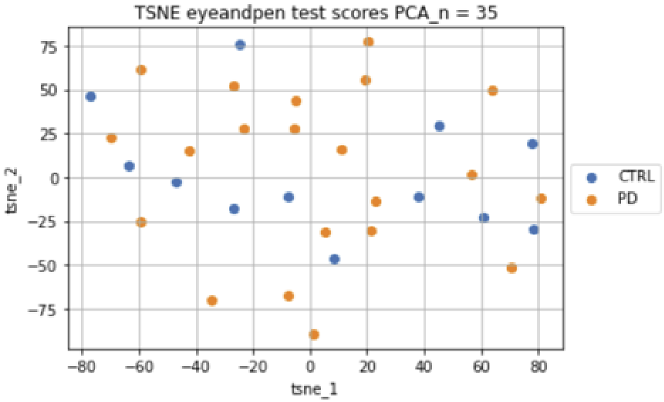
Neurovoz pataka results: 86 [PD-44, CTRL-42]





NLS pataka results: 54 [PD-21, CTRL-12, ALZ-4, other-17]



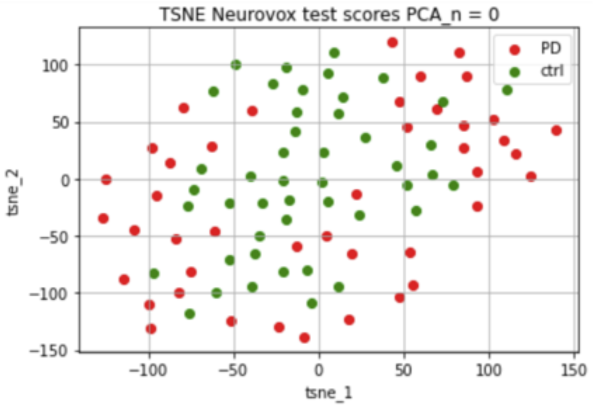


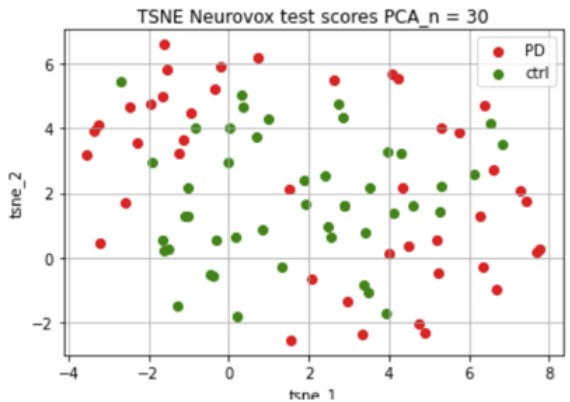
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Method 2: Speaker Recognition Based neural network model – x-vector

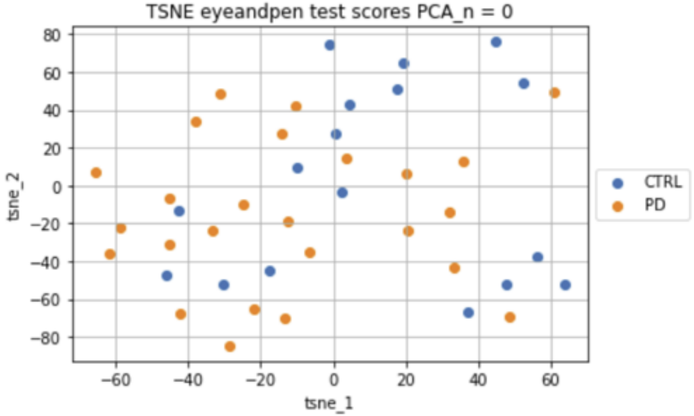
TSNE experiments

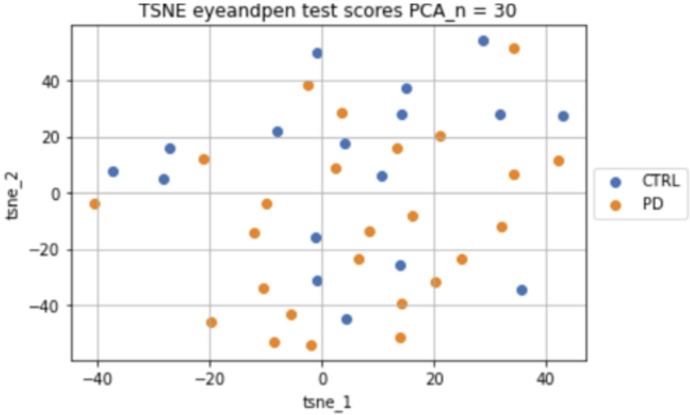
Neurovoz con results: 89 [PD-43, CTRL-46]





NLS rainbowpassage results: 71 [PD-25, CTRL-17, AD-4, other-25]



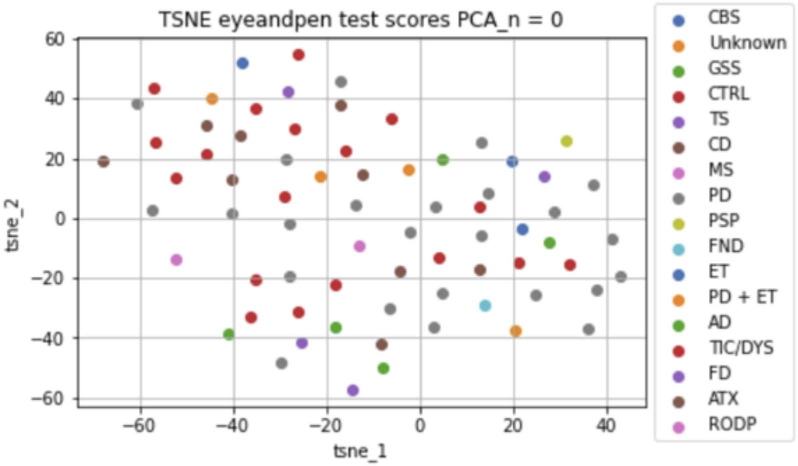


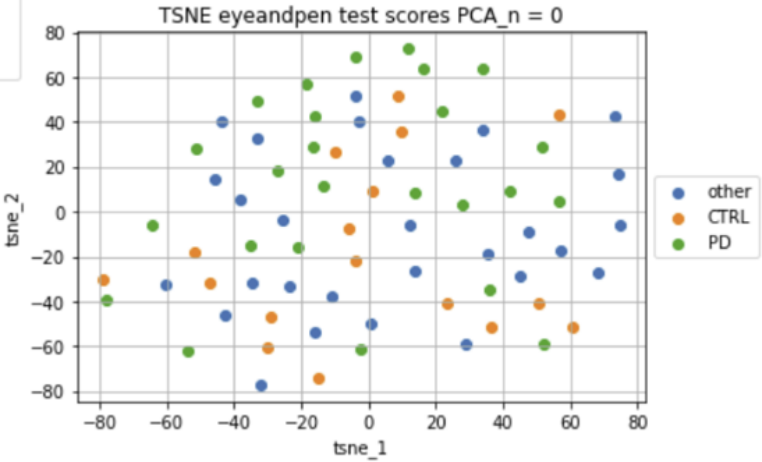
Johns Hopkins Whiting School of Engineering

Method 2: Speaker Recognition Based neural network model – x-vector

TSNE experiments

NLS rainbowpassage results: 71 [PD-25, CTRL-17, AD-4, other-25]





Method 2: Speaker Recognition Based neural network model – x-vector

x-vector features (Nx512) + retrained fully connected layer classifier (MLP classifier)

Neurovoz pataka results: 86 [PD-44, CTRL-42]

Hidden layer size	PD vs ctrl (F1 macro score)
(100,)	0.8138
(512,)	0.8371
(100,100)	0.8022
(512,512)	**0.8372**
Best with PLDA+logis	**0.8827**

Neurovoz con results: 89 [PD-43, CTRL-46]

Hidden layer size	PD vs ctrl (F1 macro score)
(100,)	**0.8647**
(512,)	0.8532
(100,100)	0.8410
(512,512)	0.8191
Best with PLDA+logis	**0.9101**

NLS pataka results: 54 [PD-21, CTRL-12, ALZ-4, other-17]

Hidden layer size	PD vs ctrl (F1 macro score)	PD vs other (F1 macro score)
(100,)	0.3863	0.3318
(512,)	0.3717	0.3272
(100,100)	0.3863	0.3567
(512,512)	0.4063	0.3318
Best with PLDA+logis	**0.6590**	**0.4974**

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Method 2: Speaker Recognition Based neural network model – x-vector

NLS rainbowpassage results: 71 [PD-25, CTRL-17, AD-4, other-25]

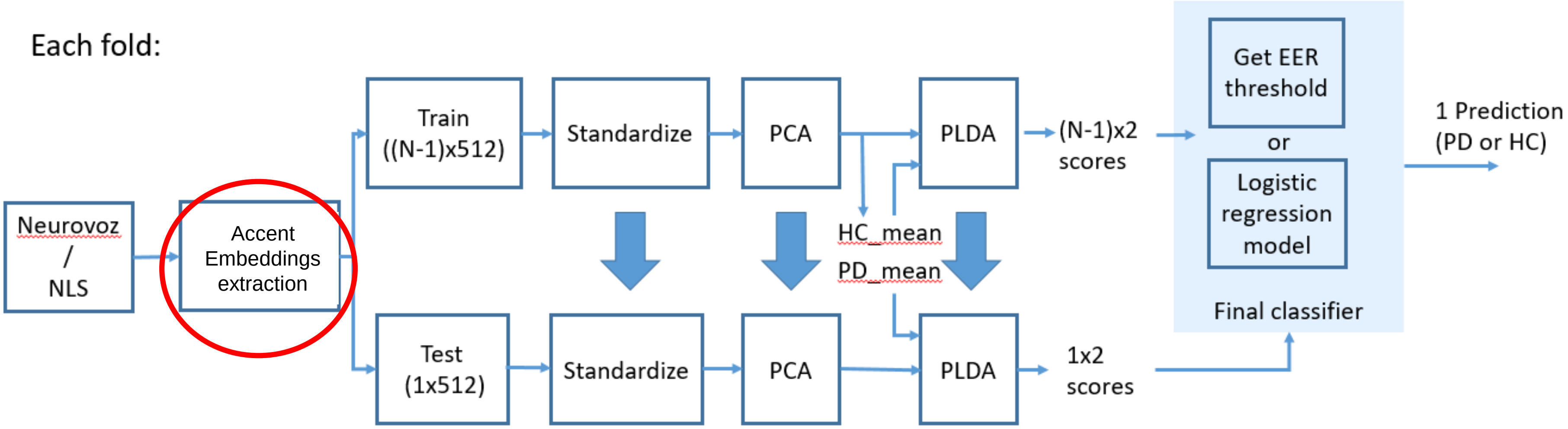
Hidden layer size	PD vs ctrl (F1 macro score)	PD vs other (F1 macro score)
(100,)	0.6541	0.4041
(512,)	0.5654	0.3784
(100,100)	**0.6816**	0.4319
(512,512)	0.6326	0.4319
Best with PLDA+logis	**0.6816**	**0.4786**

NLS wordcolor results: 75 [PD-28, CTRL-17, AD-4, other-26]

Hidden layer size	PD vs ctrl (F1 macro score)	PD vs other (F1 macro score)
(100,)	**0.6309**	0.4821
(512,)	0.6120	0.5149
(100,100)	0.5558	0.4821
(512,512)	0.5817	0.4821
Best with PLDA+logis	0.5982	**0.6206**

Method 3: Accent recognition based neural network model – CNN and x-vector

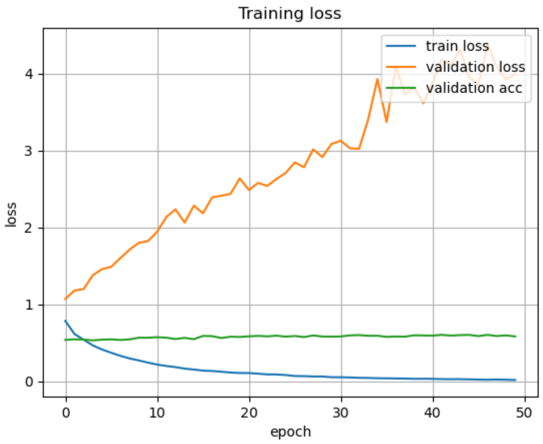
Approach 1: Not finetuned + Leave-one-out



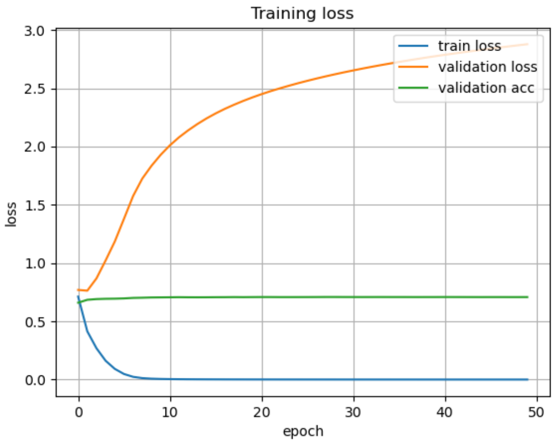
Same thing as Method 2, but instead of using an x-vector network pre-trained to recognize speakers to extract speech features from our data sets, we pre-train an accent recognition model

Accent recognition model training

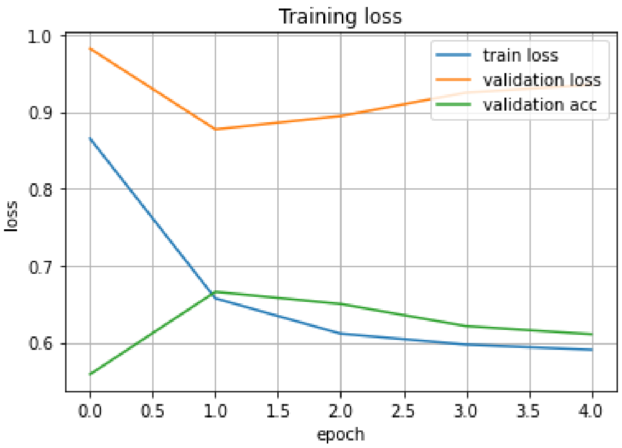
- Dataset: VCTK (~15 hrs of speech from 22 English speakers, 31 American speakers, and 19 Scottish speakers), 3-class classification



CNN: AlexNet
Best val acc: 0.545827



CNN: DenseNet161
Best val acc: 0.684695



X-vector network
Best val acc: 0.665885

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Method 3: Accent recognition based neural network model – CNN and x-vector

Approach 1: Not finetuned + Leave-one-out

Experiment Results (best-F1 macro score across different PCA_n)

	Neurovoz pataka	Neurovoz con	NLS pataka	NLS rainbowpassage
X-vector (speaker)	0.8827	0.9101	0.6590	0.6816
X-vector (accent)	0.6860	0.7525	0.6192	0.6198
DenseNet161 (accent)	0.8250	0.8650	0.6330	Needs to retrain

Neurovoz pataka: **86 [PD-44, CTRL-42]**

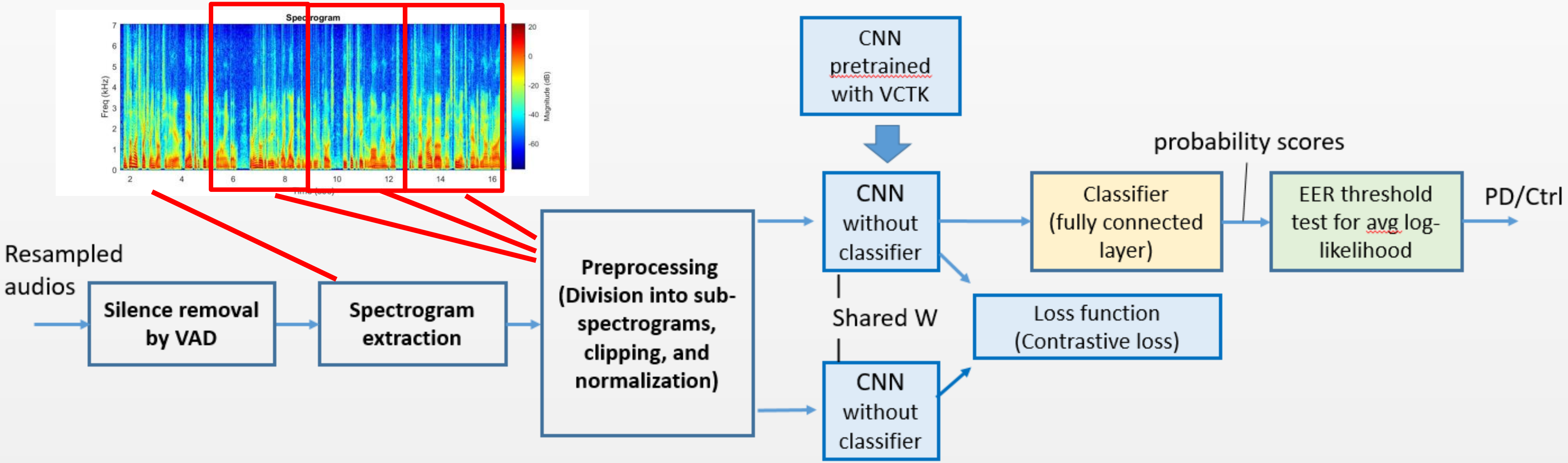
Neurovoz con : **89 [PD-43, CTRL-46]**

NLS pataka: **33 [PD-21, CTRL-12]**

NLS rainbowpassage results: **50 [PD-25, CTRL-25]**

Method 3: Accent recognition based neural network model – CNN and x-vector

Approach 2: Siamese network finetuned + 10-fold



- **Goal**: Use Siamese Network to increase "training samples" so that we can fine-tune the pre-trained network with our NLS data
  - performance not good: 10-fold cross-validation reduces the number of training speakers compared to leave-one-out cross-validation
  - To be optimized

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Method 3: Accent recognition based neural network model – CNN and x-vector

Approach 2: Siamese network finetuned + 10-fold

	Neurovoz pataka	NLS rainbowpassage
AlexNet	accuracy = 54.651162790 [[22 20] [19 25]] F1 macro score = 0.5459	accuracy = 62.0% [[19 6] [13 12]] F1 macro score = 0.6124
Accent + AlexNet fine-tuned	accuracy = 59.302325581 [[24 18] [17 27]] F1 macro score = 0.5925	accuracy = 54.0% [[21 4] [19 6]] F1 macro score = 0.4945
DenseNet	accuracy = 61.627906976 [[27 15] [18 26]] F1 macro score = 0.6162	Retrain in progress
Accent + DenseNet fine-tuned	accuracy = 67.441860465 [[30 12] [16 28]] F1 macro score = 0.6742	accuracy = 60.0% [[12 13] [7 18]] F1 macro score = 0.5941

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

References

- L. Berus, S. Klancnik, M. Brezocnik, and M. Ficko. Classifying parkinson’s disease based on acoustic measures using artificial neural networks. Sensors, 19:16, 12 2018.
- L. Moro-Velazquez, J. A. Gomez-Garcia, J. D. Arias- Londo˜no, N. Dehak, and J. I. Godino-Llorente. Advances in parkinson’s disease detection and assessment using voice and speech: A review of the articulatory and phonatory aspects. Biomedical Signal Processing and Control, 66:102418, 2021.
- Tomas Arias-Vergara, Juan Camilo Vásquez-Correa, Juan Rafael Orozco-Arroyave, Jesús Francisco Vargas-Bonilla and Elmar Nöth, "Parkinson’s disease progression assessment from speech using gmm-ubm", Interspeech, pp. 1933-1937, 2016.
- Laureano Moro-Velazquez, Jorge Andres Gomez-Garcia, Juan Ignacio Godino-Llorente, Jesus Villalba, Juan Rafael Orozco-Arroyave and Najim Dehak, "Analysis of speaker recognition methodologies and the influence of kinetic changes to automatically detect parkinson’s disease", Applied Soft Computing, vol. 62, pp. 649-666, 2018.
- LibriSpeech: an ASR corpus based on public domain audio books", Vassil Panayotov, Guoguo Chen, Daniel Povey and Sanjeev Khudanpur, ICASSP 2015
- Douglas A. Reynolds, Thomas F. Quatieri, Robert B. Dunn, Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, Volume 10, Issues 1–3, 2000, Pages 19-41,
- Palanisamy, Kamalesh & Singhania, Dipika & Yao, Angela. (2020). Rethinking CNN Models for Audio Classification.
- Yuni Zeng, Hua Mao, Dezhong Peng, and Zhang Yi. 2019. Spectrogram based multi-task audio classification. Multimedia Tools Appl. 78, 3 (February 2019)
- L. Moro-Velazquez, J. Villalba and N. Dehak, "Using X-Vectors to Automatically Detect Parkinson’s Disease from Speech," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 1155-1159
- Yamagishi, Junichi and Veaux, Christophe and MacDonald, Kirsten. CSTR VCTK Corpus: English Multispeaker Corpus for {CSTR} Voice Cloning Toolkit (version 0.92). University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2019.
- Laureano Moro-Velazquez, Jorge Andres Gomez-Garcia, Juan Ignacio Godino-Llorente, and Najim Dehak, “A forced gaussians based methodology for the differential evaluation of parkinson’s disease by means of speech processing,” Biomedical Signal Processing and Control, vol. 48, pp. 205–220, 2019.