Assessing COVID-19 Status from Crowdsourced Cough Recordings

Filip Miscevic

Caroline Malin-Mayor

Zitong Li

Zixuan Pan

UNIVERSITY OF TORONTO

Agenda

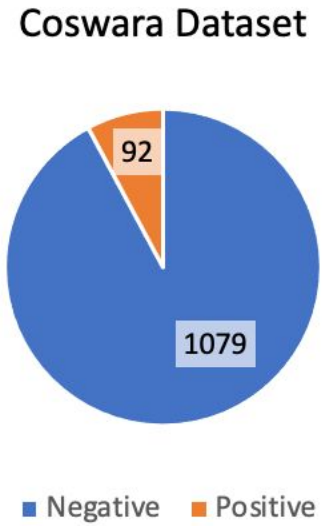
- Background/Motivation
- Datasets
- Introduction to Audio
- Methods
- Results

Background/Motivation

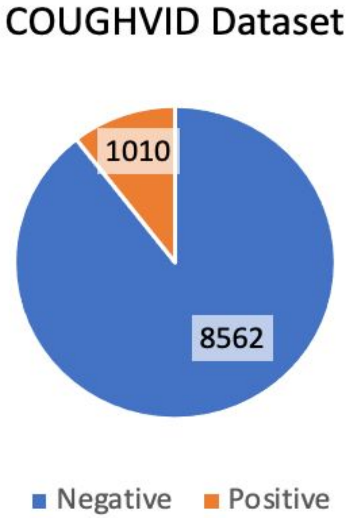
- Covid-19 represents a major novel disease burden on the worldwide healthcare system
- Obtaining Covid-19 status for a huge population is difficult and expensive
- Covid-19 is hard to identify because it shares symptoms with other diseases and carriers can be asymptomatic
- Solution: use machine learning to detect COVID-19 from cough audio

Covid Detection from Cough Audio

Previous work on a small dataset achieved >95% accuracy [2] [3]



Efforts are underway to collect and release larger datasets [5]

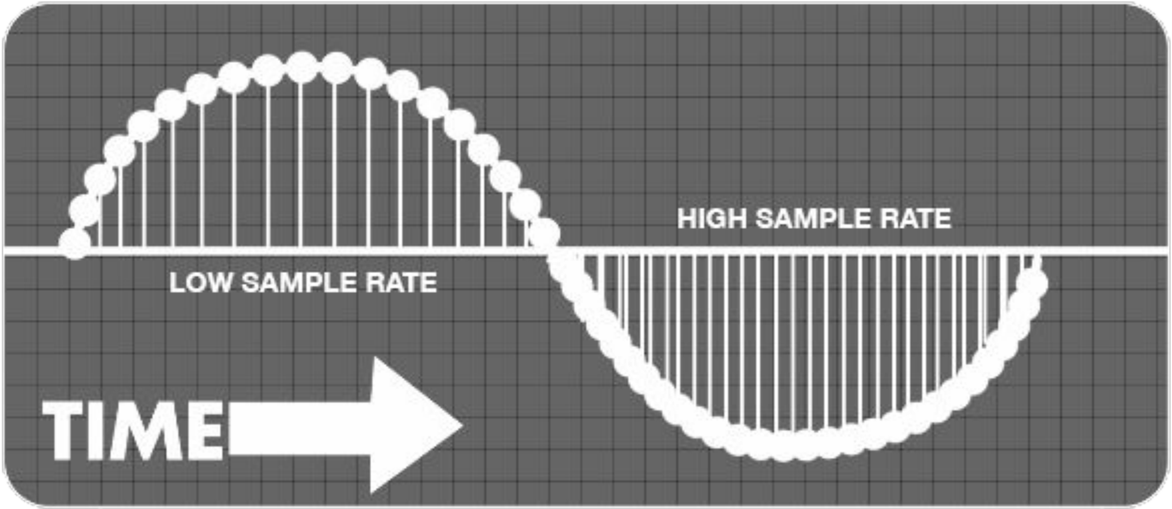


Challenges

- Cough recordings are not uniform
  - Different lengths, number of coughs, volume level, microphones
  - Background noise
- Most ML algorithms are not designed to work on audio - need feature extraction
- Relatively small and unbalanced datasets
- Self-reported labels
  - No other medical information/records

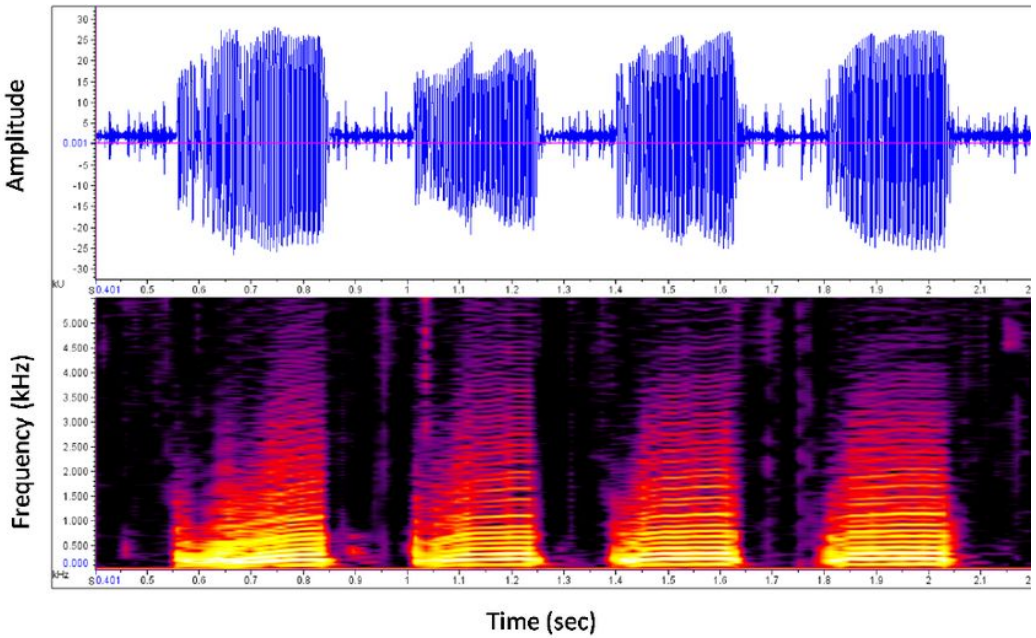
Intro to Audio Signals for ML

- **Sample**: single integer reflecting the amplitude at time point t
- **Sample rate:** number of samples per second



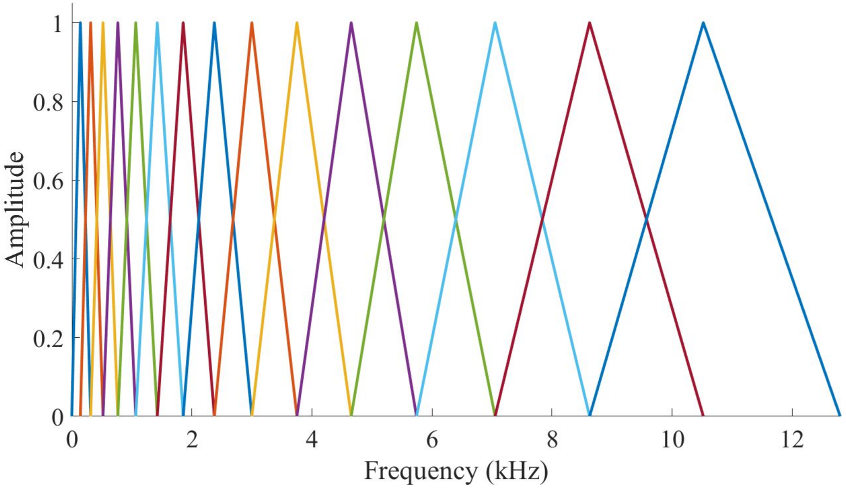
Intro to Audio Signals for ML

- An audio clip represents **time series data** (samples) **of the amplitude,** but another representation is the **frequency domain** (spectrogram)



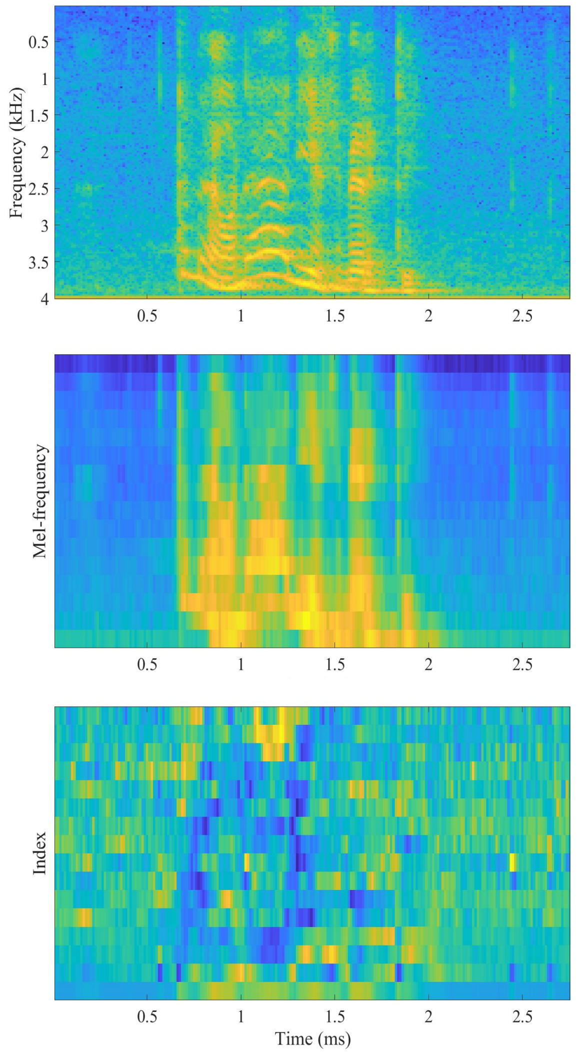
Intro to Audio Signals for ML

- For machine learning on audio signals, common to apply **mel-filterbanks** to simulate the audio frequencies that the human ear is sensitive to
- Each triangle is a single **mel-filter**



Intro to Audio Signals for ML

- Compared with a spectrogram, a **mel-spectrogram** loses fine-grained information while retaining gross information important for speech recognition
- A discrete cosine transform is applied to decorrelate the mel-spectrogram to obtain **mel-cepstral coefficients**



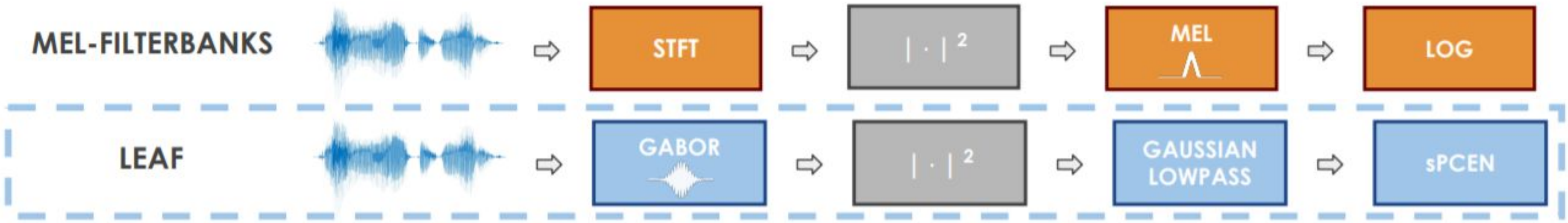
Credit: [Tom Bäckström]

Taking existing approaches and improving them with preprocessing to apply to crowdsourced data

- LEAF - Learnable Audio Frontend [4]
  - Introduce learnable parameters into the modules in Mel-filterbanks (filtering, pooling, compression/normalization)
  - Preserve the structure of Mel-filterbanks and improve the performance of each module within
- Data Augmentation
  - Increase the amount of data by adding copies of slightly modified data samples
  - Helps the model overfitting issue without the need to collect more data
- Self-Supervised Audio Pre-training
  - Leverage information from large audio datasets

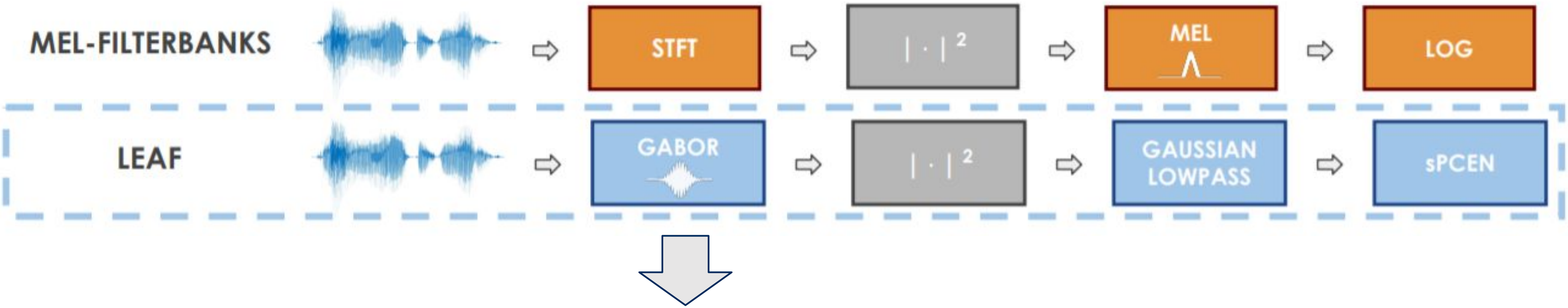
LEAF - **LE**arnable **A**udio **F**rontend [4]

- Learnable modules are introduced to Mel-Filterbanks
  - Frontend parameters are trained **end-to-end** with the model parameters



LEAF - LEarnable Audio Frontend

- Learnable modules are introduced to Mel-Filterbanks
  - Filtering: Gabor 1D-Convolution filter
  - Frame extraction : sliding window VS fixed frame size



$$f_n = |x * \varphi_n|^2 \in \mathbb{R}^T, \quad n = 1, \ldots, N,$$

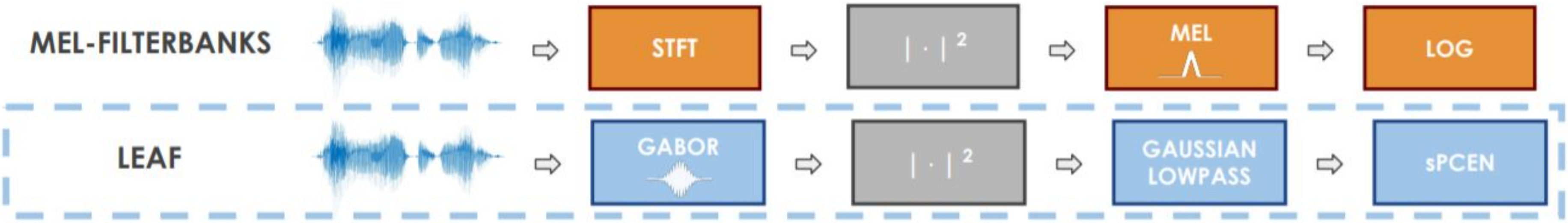
$$\varphi_n(t) = e^{i2\pi\eta_n t}\frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{t^2}{2\sigma_n^2}},$$

$(\eta_n)_{n=1..N}$ Central Frequency

$(\sigma_n)_{n=1..N}$ Width of the filter

LEAF - LEarnable Audio Frontend

- Learnable modules are introduced to Mel-Filterbanks
  - Frontend parameters are trained **end-to-end** with the model parameters
  - Pooling: Downsampling the resolution using convolution with Gaussian lowpass filter

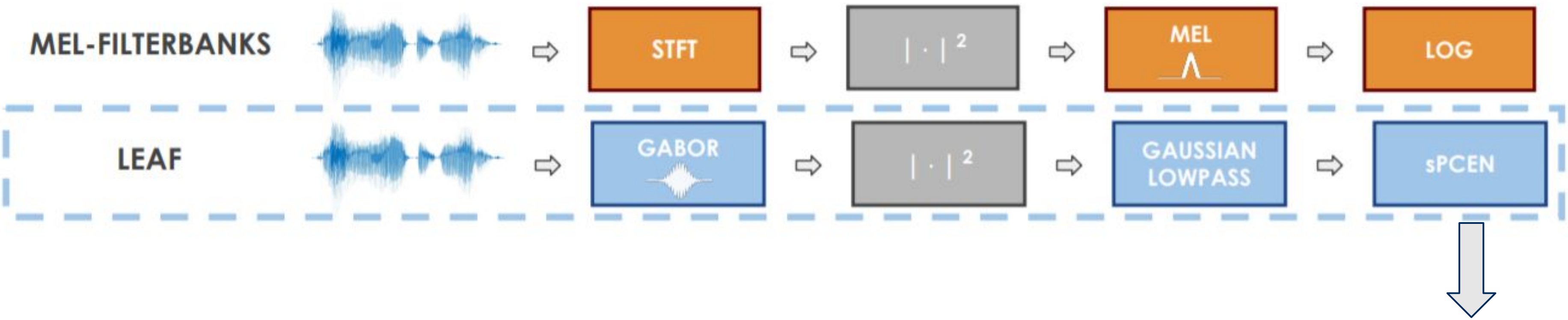


$$\phi_n(t) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{t^2}{2\sigma_n^2}},$$

$(\sigma_n)_{n=1..N}$ Filter width

LEAF - LEarnable Audio Frontend

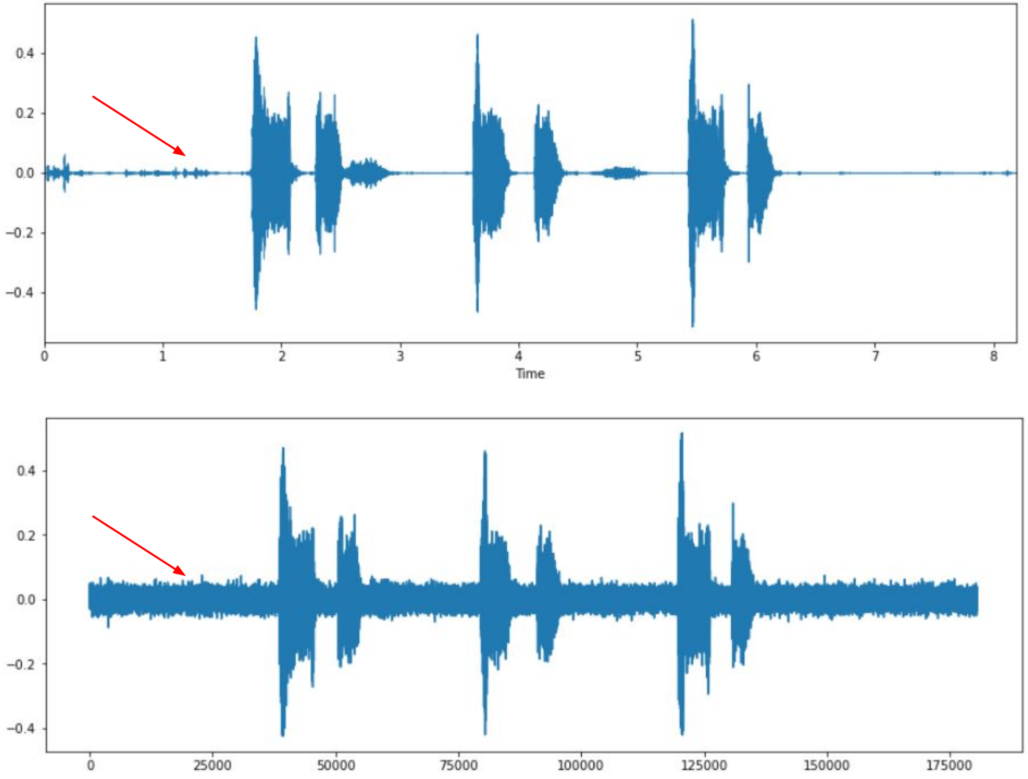
- Learnable modules are introduced to Mel-Filterbanks
  - Frontend parameters are trained **end-to-end** with the model parameters
  - Normalization/Compression: compressing each channel with moving sum



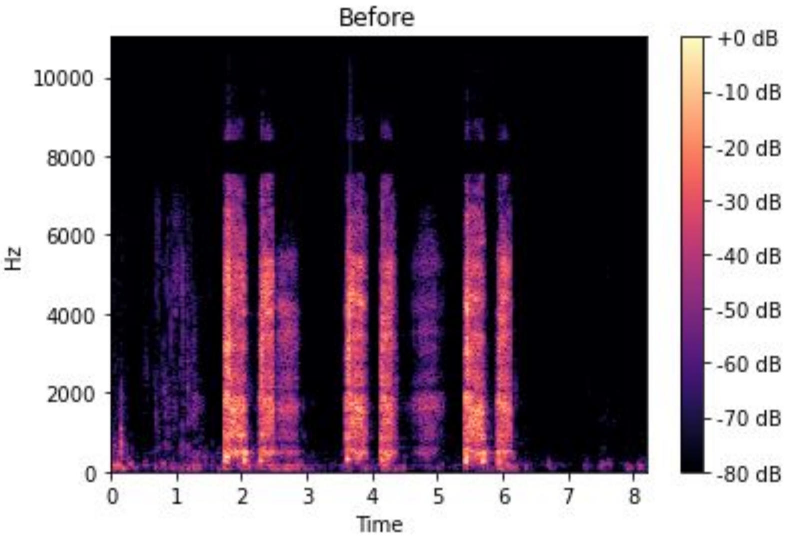
$$\mathrm{PCEN}(\mathcal{F}(t,n)) = \left(\frac{\mathcal{F}(t,n)}{(\varepsilon + \mathcal{M}(t,n))^{\alpha_n}} + \delta_n \right)^{r_n} - \delta_n^{r_n},$$

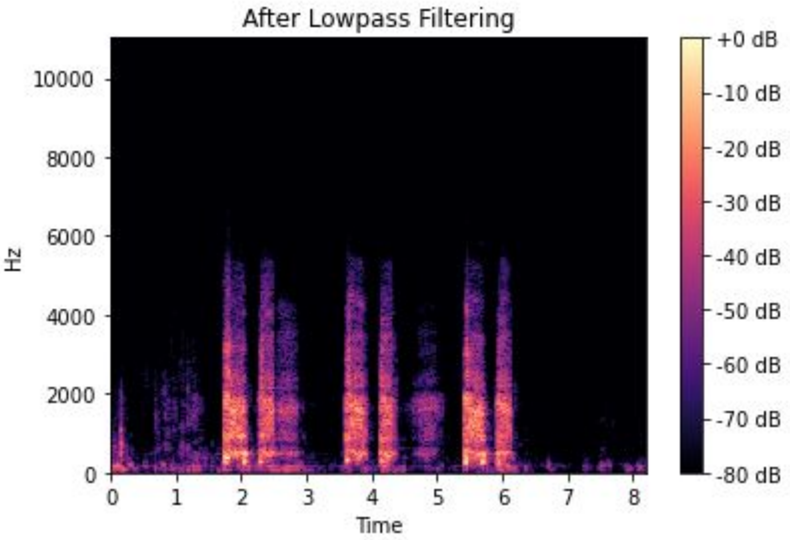
$$\mathcal{M}(t,n) = (1-s)\mathcal{M}(t-1,n) + s\mathcal{F}(t,n),$$

Augmentation - Gaussian Noise

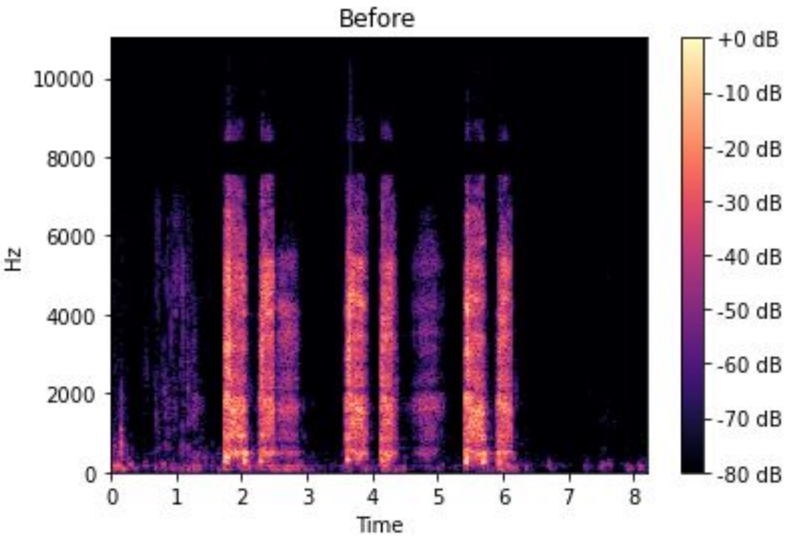


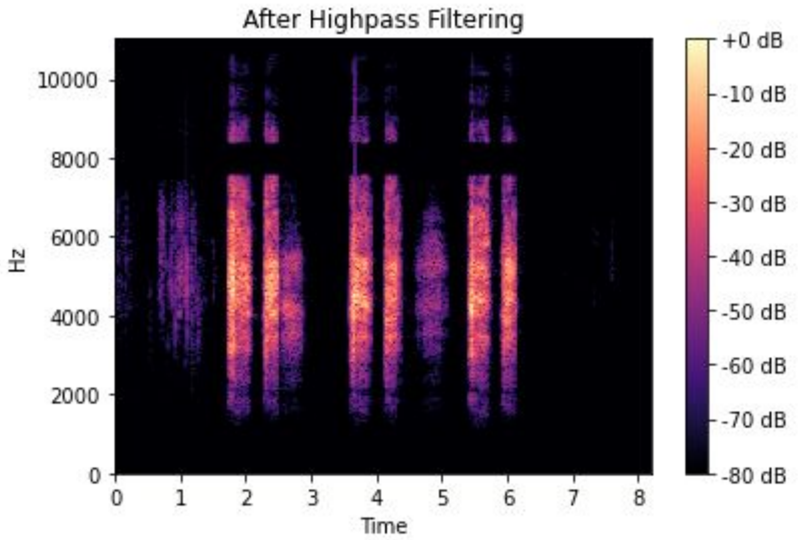
Augmentation - Low Pass Filtering



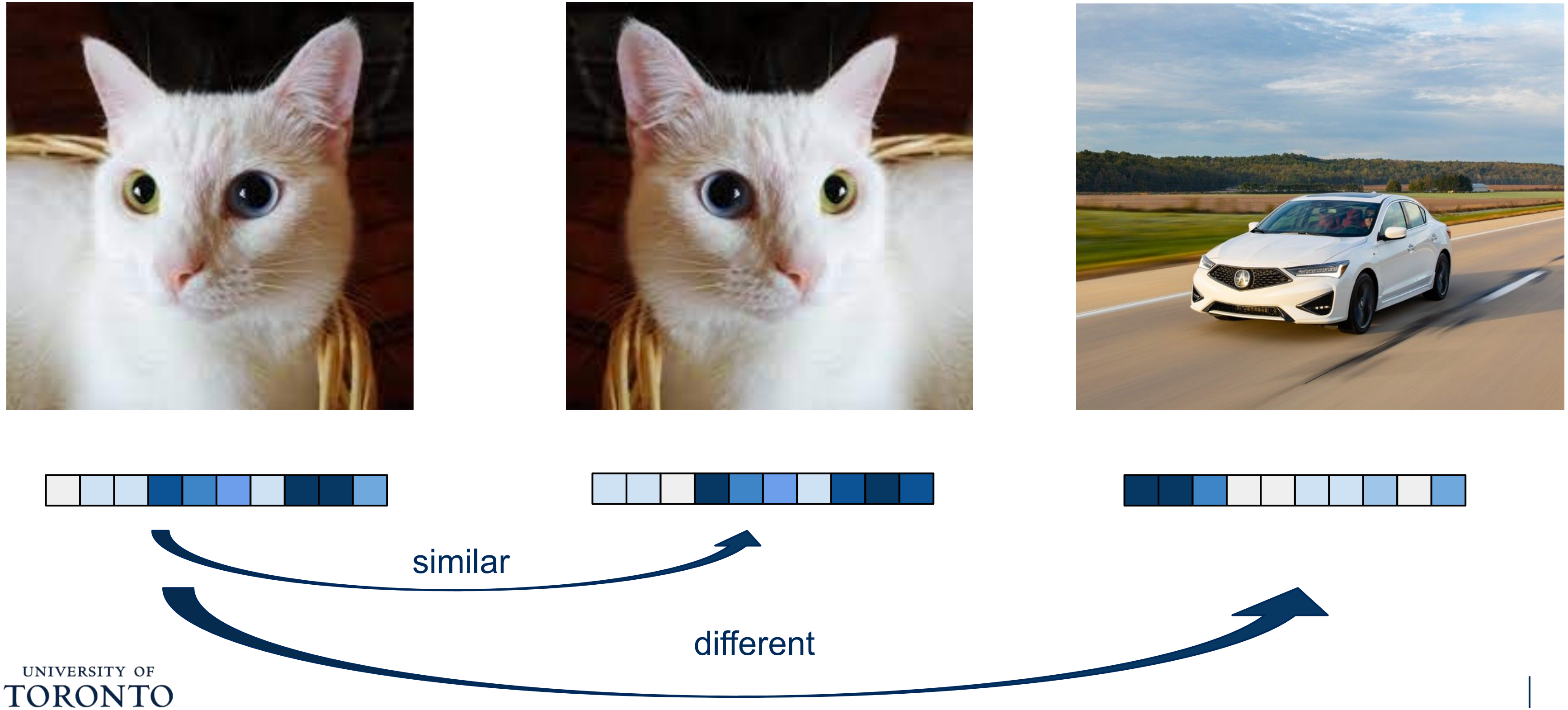


Augmentation - High Pass Filtering



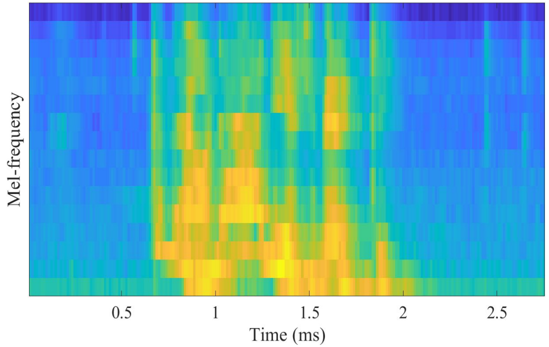


Self-Supervision



Bootstrap Your Own Latent - Audio (BYOLA) [1]

Input: Mel-Spectrogram



Augmentations:

- Random mixing with other samples - simulate background noise
- Crop and resize - approximate pitch shifting and time stretching

Pre-Trained BYOL-A Model

- Trained on Audio Set dataset - 2 million 10-second audio clips from Youtube videos labeled with 600+ classes

- Using a simple linear classifier on top of the representations, BYOL-A achieved state-of-the-art performance on many downstream tasks

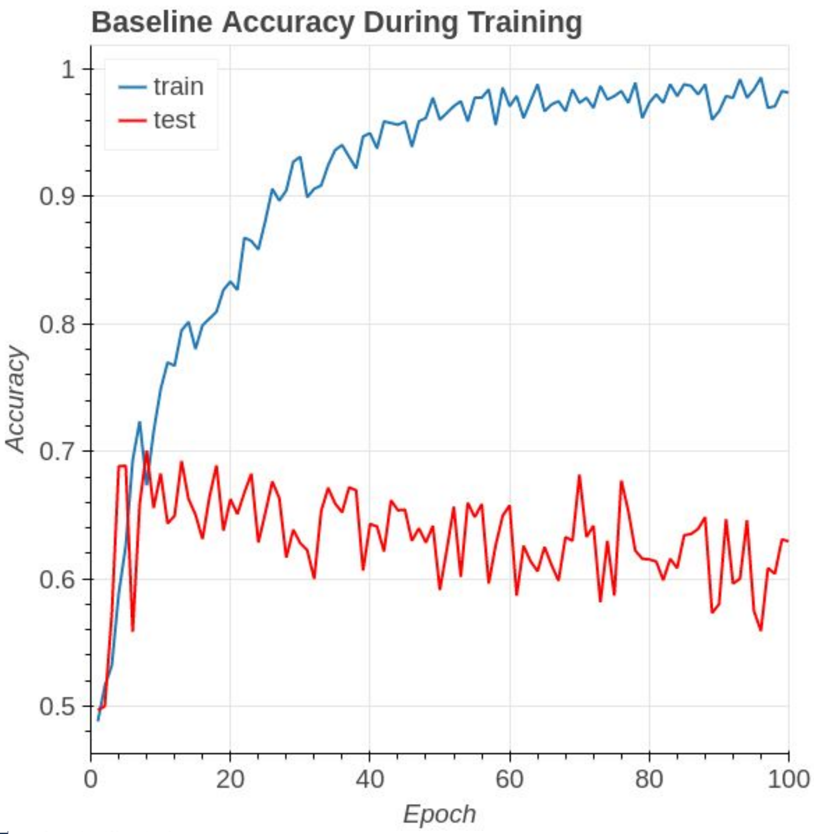
- Our goal: Apply the pre-trained BYOL-A model to our cough data, and train a simple linear classifier on top of the representations

Initial Results

Results Reported in Previous work [3]

Classifier	Performance			
	Spec	**Sens**	**ACC**	**AUC**
Resnet50	*98%*	*93%*	*95.33%*	*0.976*
Resnet50	98%	93%	95.01%	0.963

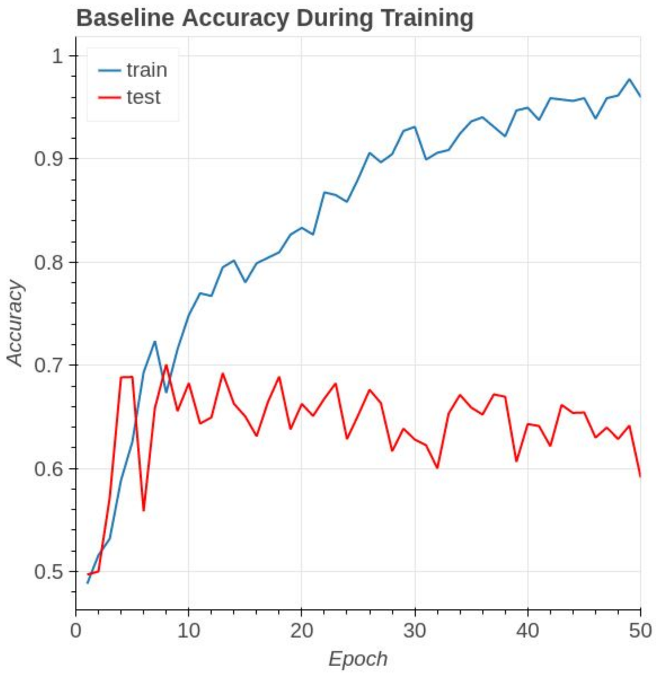
Coswara Baseline: Resnet18 with engineered feature extraction



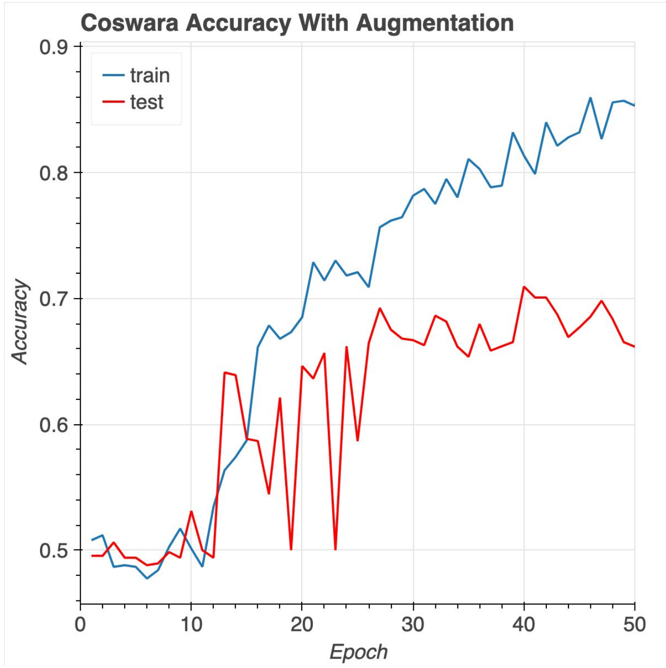
Confusion Matrix at best epoch (8)

	Predicted 0	Predicted 1
Actual 0	312	28
Actual 1	48	45

Initial Augmentation Results



Original

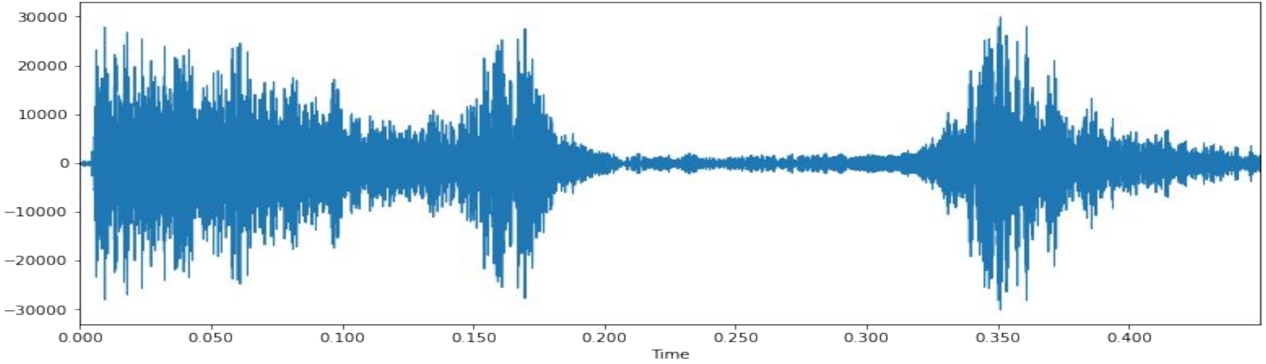


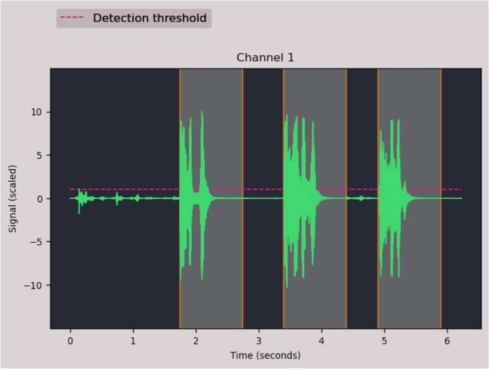
Data Augmentation

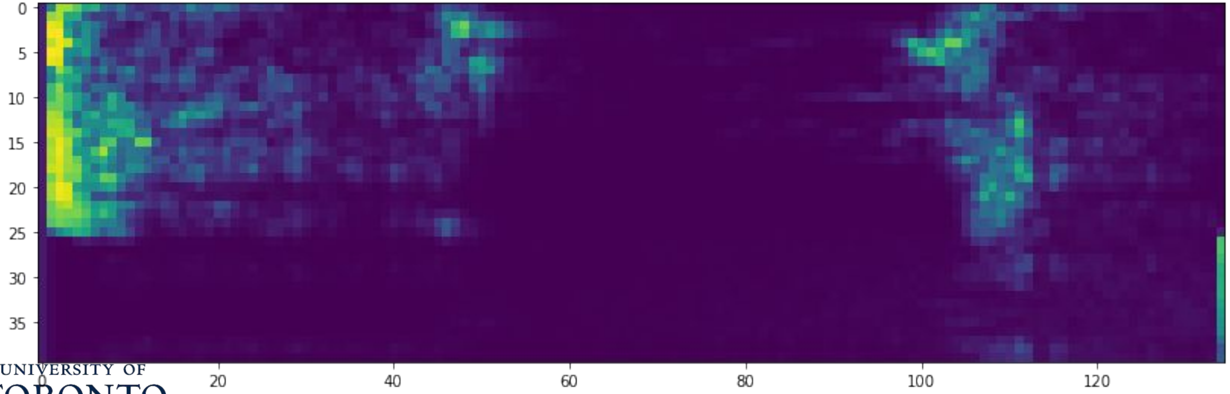
Practical Challenges

- Had to re-create code from paper to get baseline
  - Feature extraction in paper is complicated!
- Baseline model did no better than chance on COUGHVID data
- Baseline model also not achieve expected performance on Coswara

LEAF Implementation







Next Steps

- LEAF results
- BYOL-A self-supervised model results
- SMOTE
  - Further data augmentation by creating synthetic samples for the minority class (from original paper)

References

[1] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation,” *arXiv:2103.06695 [cs, eess]*, Apr. 2021, Accessed: Dec. 06, 2021. [Online]. Available: http://arxiv.org/abs/2103.06695

[2] N. Sharma *et al.*, “Coswara -- A Database of Breathing, Cough, and Voice Sounds for COVID-19 Diagnosis,” *Interspeech 2020*, pp. 4811–4815, Oct. 2020, doi: 10.21437/Interspeech.2020-2768.

[3] M. Pahar, M. Klopper, R. Warren, and T. Niesler, “COVID-19 cough classification using machine learning and global smartphone recordings,” *Comput Biol Med*, vol. 135, p. 104572, Aug. 2021, doi: 10.1016/j.compbiomed.2021.104572.

[4] N. Zeghidour, O. Teboul, F. de C. Quitry, and M. Tagliasacchi, “LEAF: A Learnable Frontend for Audio Classification,” *arXiv:2101.08596 [cs, eess]*, Jan. 2021, Accessed: Dec. 06, 2021. [Online]. Available: http://arxiv.org/abs/2101.08596

[5] L. Orlandic, T. Teijeiro, and D. Atienza, “The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms,” *Sci Data*, vol. 8, no. 1, p. 156, Jun. 2021, doi: 10.1038/s41597-021-00937-4.

Initial “results” - Coughvid struggles

- Data is stored in various format and some samples are missing cough audios
- Train accuracy is high but test accuracy is low - model is overfitted
-