19/11/24

OPERA

→ Began reading in the github documents for the OPERA dataset which appears to be useful for the machine learning side of things and for datasets related to respiratory data.
↳ can arrange to speak to Evelyn once I have a better idea of what I want to do with this.

→ Looked over the first parts of the OPERA paper and stopped at the spectogram section. Producing spectrograms will be something I need to do in order to conduct analysis on audio datasets of my choice.

→ Followed referenced papers (stored these on Zotero) until I reached one that spoke of the Librosa library in python that is used for audio analysis and producing the spectrograms

Librosa

→ Followed instructions to install Librosa library along with other modules it requires to perform all of its functionality.

→ Have encountered research using machine learning to identify various environmental sounds (~~train~~ Animals, Natural soundscapes, Human, Interior/Domestic, Exterior/Urban) perhaps I could test machine learning methods on spectrograms to identify instruments / genres?
↳ found paper doing just that, will read and see where I could advance. Submitted 1st November 2024

Ideas

→ should focus on conclusions of papers for inspiration for extension ideas on what has already been done
↳ Speak to Pietro about these ideas.

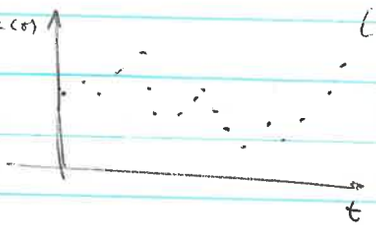
22/11

XGBoost

→ Mentioned by Josh as more powerful than the Convolutional Neural Networks he studied for his dissertation. Extreme Gradient Boosting has been applied to disturbance classification. 2022 study

Gramian Angular Field

→ Scale the setup values $X = \{x_i\}$ to be within the range $[-1, 1]$ then convert these values into cosine of polar angle ϕ.



→ plot the values of ϕ with r increasing with the time to get a polar plot from your time series.

$$GASF = \cos(\phi_i + \phi_j) \quad \text{(summation)}$$

$$GADF = \sin(\phi_i - \phi_j) \quad \text{(difference)}$$

(matrices (i, j) image?

Time average of d (spectogram) as function of delay time

→ Figured out what poetro suggests is to compute these spectrogram images and then somehow take the difference between them and then average it. Then do this method at multiple delay times (move the spectrogram and take averages?) and plot the time average as a function of delay time.

↳ can be classified with image classification

↳ Maybe also try the GAF with this as would be a time series? then do image classification with the GAFs.

29/11/24

Progress

→ implemented Evelyn's framework on the W1 and managed to run tasks 10 and 11, both returned sensible results similar to those she was measuring:
↳ Importantly different though as these are my own iterations of the process.

OPERA - In detail

Audio recordings can be used to estimate respiratory rate and lung function, detect snoring and apnea events during sleep, assess the effect of smoking on health and diagnose diseases like flu and asthma.

Supervised deep acoustic models have been proposed, but their performance relies heavily on the volume and quality of available labels → not easy to have for respiratory data
↳ maybe just generally.
↳ ∴ training models with large amounts of unlabelled data has high potential to improve performance through transfer learning and supervised fine-tuning

Supervised vs. Unsupervised.
→ Supervised learning uses labelled data, meaning each input (feature) comes with a corresponding output (label).
→ Unsupervised ~~data~~ learning uses unlabelled data, meaning only input features are provided without corresponding output labels.

Feature: measurable property or characteristic of data that is used as input to a model.

OPERA curates unlabeled respiratory ~~data~~ audio datasets, pretrain three pioneering foundational models, and evaluates them against existing pretrained acoustic models across various applications.

→ ~ 136 k samples, 400+ hours orders of magnitude greater than number of respiratory audio samples in datasets used for training existing open acoustic models.

→ 3 generalizable acoustic models pretrained with the unlabeled data (contrastive, generative).
leaning based Transformer / CNN

→ 10 labelled datasets to formulate 19 respiratory health tasks

Results show the OPERA models are generalizable across multiple downstream tasks, including new datasets and unseen respiratory audio modalities
e.g. clinical imaging, electronic health records and medical time series.

→ Contrastive model is better for classification-based downstream tasks, while the generative model performs better in regression tasks.

Classification - discrete data categorizing
Regression - predicting trends in continuous data (like stock markets

Before pt training, all recordings are resampled to 16 kHz and merged into a mono channel. They are then transformed into spectrogram using 64 Mel filter banks with a 64 ms Hann window that shifts every 32 ms. ⇒ Spectrogram of 1 × 126 × 64 dimension.

epoch: complete pass through the entire training dataset.
→ if a dataset contains 10,000 samples and you train for 5 epochs, the model will see each sample 5 times but not necessarily in order

there are other pretrained acoustic models: OpenSmile, VGGish and AudioMAE and CLAP.

→ How do the tasks ~~use~~ split the data up if there could be many distinguishing features in the audio file?

30/11/24

Plan Skeleton (Plan is due 06/12/24!)

(Prior speak about motivations)

(1.) You have identified a state of the art framework (OPERA) developed by collaborators.

OPERA curates a large dataset of unlabelled respiratory data
→ not currently available blah blah blah.
→ also trains.

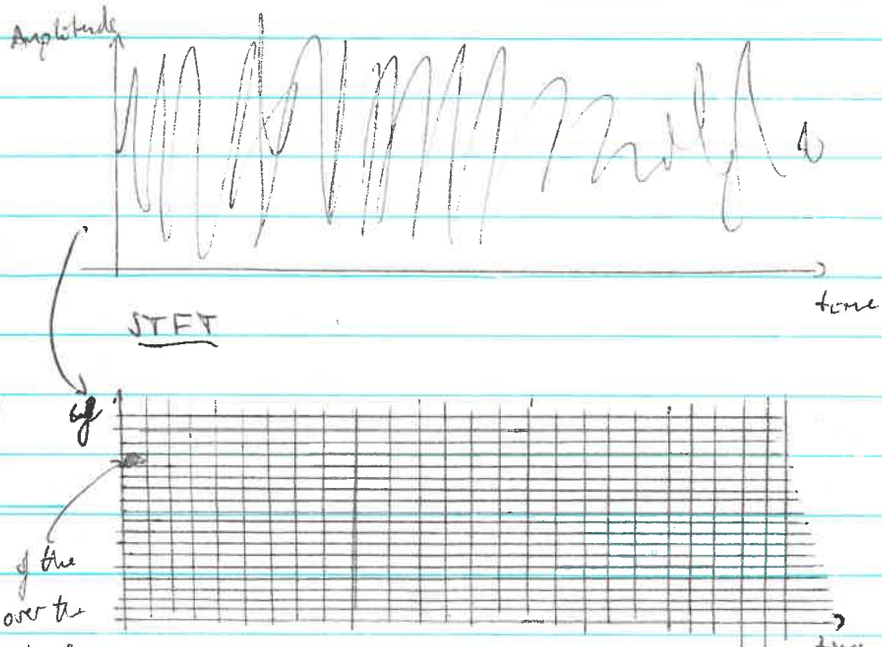
(2.) OPERA uses features in an optimal way, within a ML framework

(3.) We have discussed some new features, based on average differences of spectrograms, which we'd like to add to the standard features.

(4.) We'd like to know on samples of real audio and basket of classification / discrimination, what role our new features can have, and see if they can have an impact in the field.

(5.) There is a dataset connected to the development of OPERA, want to do a like for like comparison. If have found other data and if we see value on these respiratory applications, could be good to test in wider applications.

1/12/24



Big mess of oscillations

02/12/24

Gantt chart.

1. Creation of new images + replacing mel spectrograms in Evelyns process. 21/01 → 04/02 (2 weeks)

2. Pretraining models on the OPERA investigation
28/01 → ~~04/02~~ 11/02 + 1 week

3. Completing tasks with the pretrained models and new images.
11/02 → ~~25~~/02 + 2 week →

4. Collation of results and comparison with mel spectrogram results.
18/02 → 11/03 + 2 week

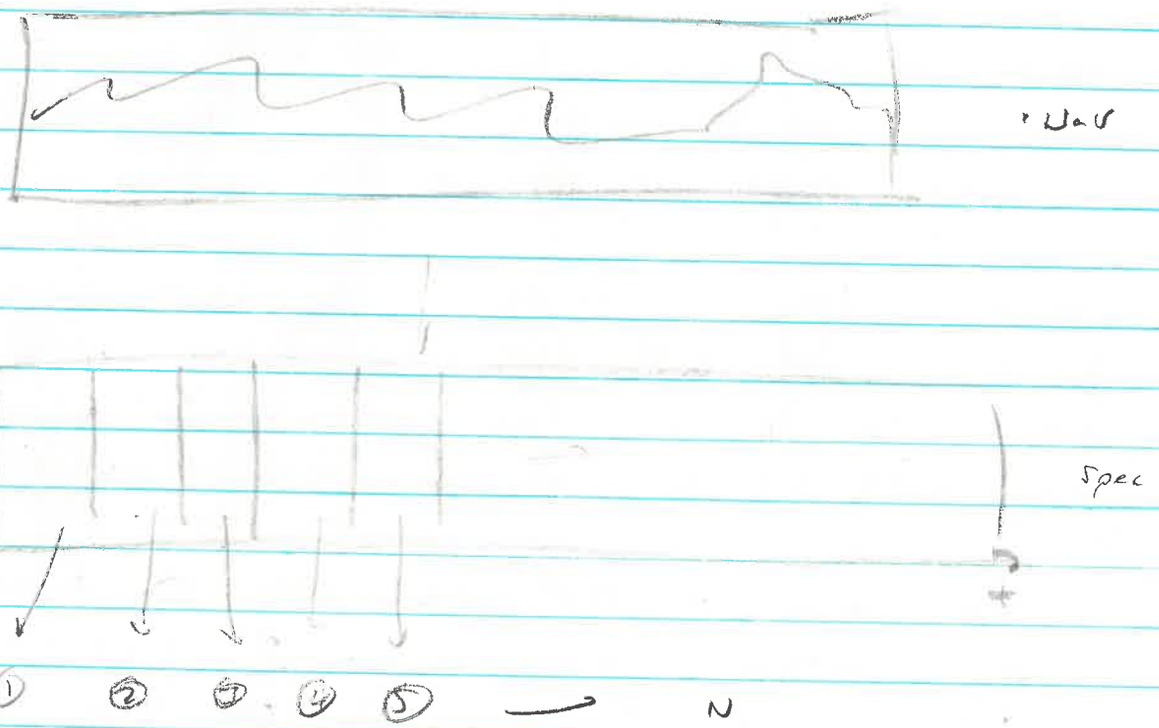
5. Application of new images to other datasets with different objectives 11/03 → 29/04

6. Report writing. (Deadline 2nd June)
29/04 → 02/06

7. Presentation preparation (7th week of last)
4/03 → 11/03

8. Poster creation. (Deadline 12th June)
02/06 → 12/06.

29/1/25



Potential functions to modify:

→ from src/utils.py, pre_process_audio_mel_t

⇒ Conversation with Chat gpt:

Might need to do some tuning to get the best results with this new form of spectrogram

→ Reading code, clearly pre_process_audio_mel_t gets used in other important functions

- get_split_signal_librosa (1)
- get_entire_signal_librosa (2)
- get_individual_segments_librosa (3)

(1) loads an audio file, trims silence, and splits it into smaller segments.
Adds a filter to pick out frequencies within a certain range (bandpass) → between 200 and 1800 Hz.
→ Can return either the split audio chunks or the spectrograms.

② Processes entire signal and uses librosa function to remove silent parts. Pads out the sample to get to expected time (8s) if removing silences makes sample too short.

③ this function trims out the silences then produces a spectrogram of a specified length. This must be very useful for the job at hand, can split a clip into very small segments and then do delay time measurements?

↳ difference between ③ that does make audio file containing noise silences ⑤ splits into the sections above there or noise and makes them into separate spectrograms.
(Osborne)

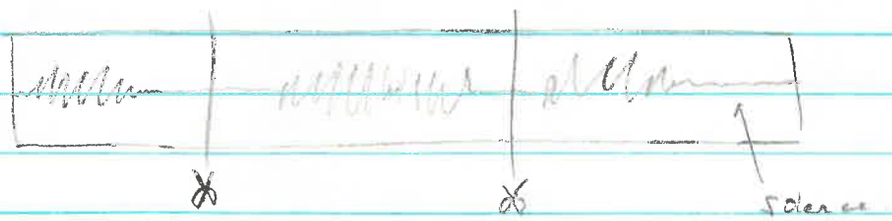
Question

→ Can I just create a load of these delay images and feed them into 'extract_opera_feature' and see what happens?

30/1/2025

Pietro's Words:

- Time average of the differences of spectograms as a function of delay time between spectograms.



take 1 snippet

split sample into N slices of the spectogram so $dt = \frac{t_f}{N}$.

Take the difference of each snippet with all other snippets keeping tab of which the delay time (how many dt's the two snippets are away from each other.

Group the differences between the spectrograms into their respective delay times and average over the time axis. Now plot the time averaged strips as a function of delay time.

Variables → 1. length of snippet

→ other ways we discussed mentioned nearest neighbours.

Have managed to successfully plot spectograms using these functions.

→ need to see how the removal of silences will affect how I intend to do the delay time plots.

→ pre processing and spectograms to be done at the end of all the manipulations or functions ① ② and ③ therefore if I ~~I~~ only modify and open to do what I want then that seems to make sense.

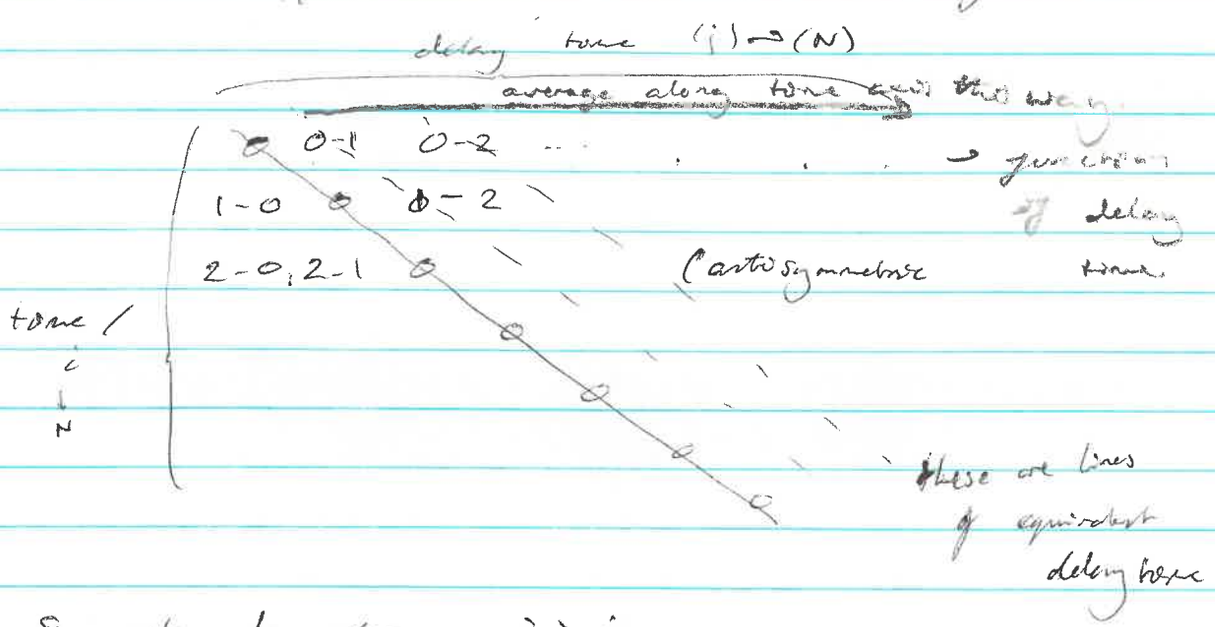
31/1/25

→ Have managed to create snippet plots of spectrograms, ready to create delay plots with.

First going to try Method ① outlined 30/1/25

Outline of function:

- loop through snippets,
  take i^{th} snippet, Subtract all j snippets and append to row / column of matrix.



so only do for $i > j$

Doesn't appear to make a great deal of sense to do averages over all the same delay times.

- seems to just result in one image on my hands

→ only have one unique decorrelation for largest delay time e.g. 2 → 1 is large delay time crossing each other or the negative of short delay time which we ignore.

$$\sum_{n=1}^{N} N - \sum_{n=1}^{N} n \quad : \quad N^2 - \frac{N(N-1)}{2}$$

$$= \frac{N(N+1)}{2} .$$

→ Average each entry along a row along its time axis and then append the frequency averages together → spectrogram as a function of delay time.

→ doesn't make sense.

4/2/25

— Call with Pedro:

- ① only take delay tone averages where the number of snippets averaged is $\geq \frac{1}{3}W$
  ↳ include the diagonal of zeros. This is characteristic on the video analogue of this technique.

② After tackling this, there is the idea of episodes
- two different time scales, episode length and snippet length.
- For long audio clips makes sense.
- do normal delay images but separately for each episode.
- length of episodes can vary - something to explore.

③ Another angle is the windowing function length, parameter involved in creating the spectrogram impacts how pronounced certain frequencies are. by taking the window length as a third axis, has the effect of similar to looking at wavelength (let?) transform of audio instead of FT spectrogram.

6/2/25

Delay plots

- Had query over whether more sensible to leave conversion of spectrogram to decibel units ~~to~~ before or after the delay time averaging process.

- Pietro believes that decibel conversion before is gone, the images look better this way, worth at some point putting some solid evidence behind this.

- Now have capability to create delay plots of any window length for any audio file → all functions written in spec-delay file.

DDM and Multi-DDM process

DDM algorithm takes the algebraic difference of several couples of frames separated by a lag time τ.

The differences are then Fourier transformed in space and the results averaged

- Fourier transform after taking algebraic difference? I am not currently doing this.

- The idea of episodes in the video content is wished to be described as multi-multi-DDM, varying box size and also episode length (subset of time).

- Have downloaded Luigi's Matlab code for DDM with motile cilia, will need significant modification if I wish to use.

Main things to check coming up

① Fourier transform before or after time delay manipulation (or just try both).

② Need to see how to get feature from my new data.

7/2/25

New problem: if an audio clip going in with a sample rate of 44000 gets put through pre process mel spec with sample rate 16000, this appears to be reducing the time of the sample (may be just my visualisation).

→ Now big misunderstanding over sample rate and how spectrogram manipulation is reducing the lengths of the audio clips, need to investigate.

Have created a process to unpack a directory of .wav files and transform them, an into delay images in a new specified directory alongside another folder holding images of the spectrograms.

10/2/25

→ Meeting Pedro Wednesday, emailed Evelyn want to get training

• Have implemented a delay image output to get entire signal - librosa not entirely confident it will help much.

11/2/25

→ Delay episode functions.

1st divide and/or clip into time bins.
↳ need to figure how to extract time length.

2nd perform snippets function on each episode.

3rd just the same as delay image function.

(Probably best to make two separate functions.

12/2/25

Meeting

- Convert the delay plots to take the modulus of the snippet differences rather than the actual values.

- Get stuck into writing an image classification process using google thing (Pytorch or tensorflow). Should be relatively simple tutorials step by step online.

- Definitely worth investigating what is happening to the time dimension, generate random .wav file and see what happens.

14/2/25

Have located a step by step walkthrough on github for an example of audio classification using Tensor flow CNN model.

Have been able to directly substitute my delay tone function to generating the spectrograms and am now following through to replicate the CNN machine learning.

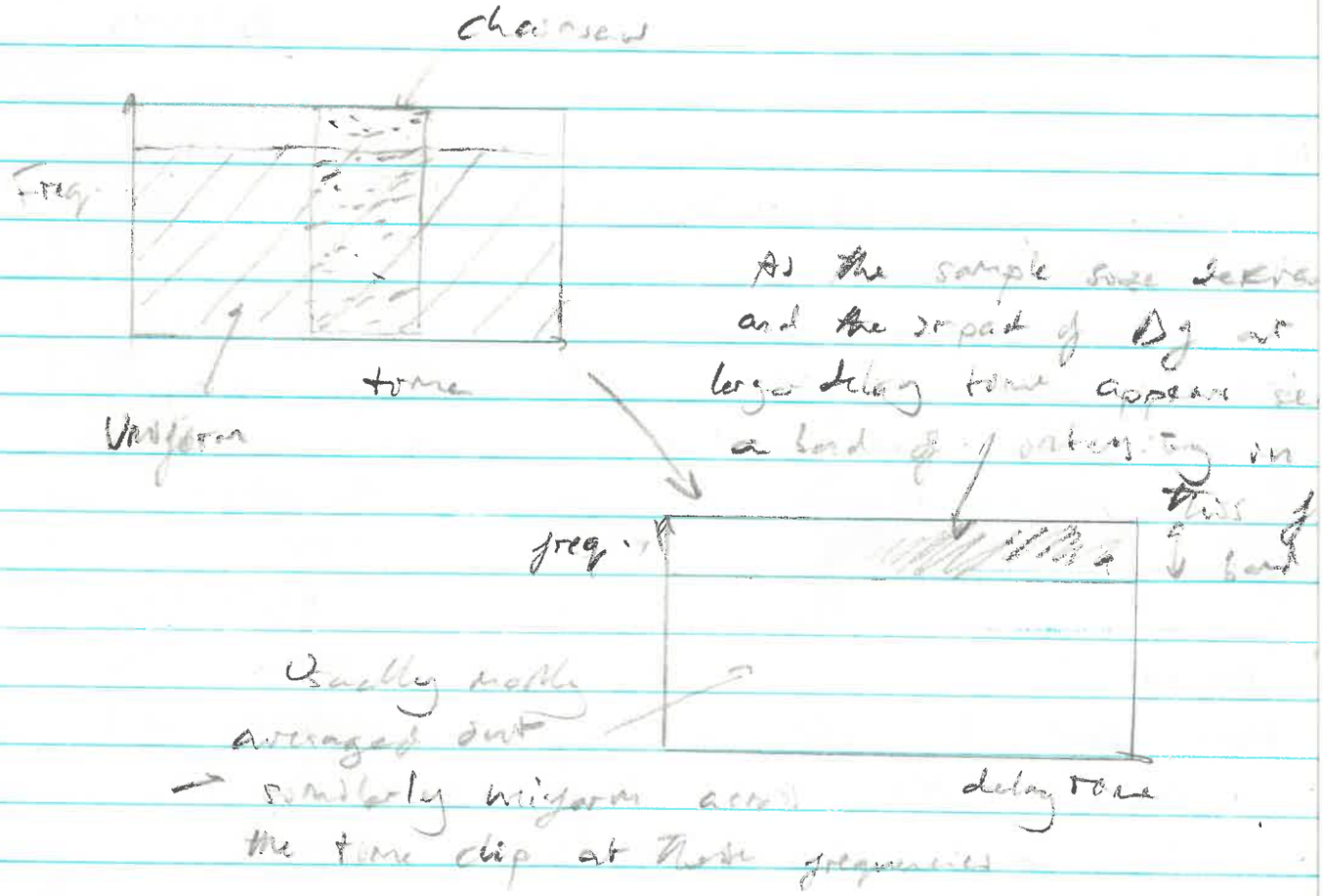
16/2/25

Progress

Have replicated the use of a CNN to distinguish the sounds of chainsaws, background noise, engines and storms.

I did this using the delay representation of the image with a step of 1 pixel. (Aside: next vary the size of window and see how this impacts the accuracy of the CNN). The results on the whole were that using the delay image had very little impact on the performance of the CNN, but importantly it did not seem to perform worse. These conclusions, I have made by comparing the accuracy curves as a function of epoch for spectrogram representation and for delay representation.

Something to note is that the spectrograms for background noise vs those in which there are chainsaws / engines / storm rumblings are a uniform dist. (ish) with an overlaying patch of intensity in same tone zone where the intensity extends to higher frequency.



Perhaps delay images are less useful in this context as there is so much noise so the presence of a distributing noise such as a chainsaw just appears as a disturbance to the uniform low frequencies and hence a band on the high frequencies like I've drawn rather than exposing 'hidden' rhythms in an isolated audio recording → episodes might be required to try and add value to this analysis.

There was one instance in which there was a log feature in identifying ergones from shawms which was a lot more than working with spectrograms. This was just one time though and lots of repeats with would be needed for a conclusion on whether my method struggles with this distinction.

To do

- try analyses again with larger window length and see how this affects performance. ✓
- Check some concrete examples displaying what I actually do to the images and find out what is happening to the interpreted time when I put audio clips through Evelyn's spectrogram generating functions.
- Follow up email to Evelyn showing her the accuracy curves and ask where to go to do this with her models (good start and might allow me to figure out how the code works a bit better.) ✓

17/2/25

Evelyn's response:

OPERA is a pretraining framework. Both pretraining or end to end training are feasible for testing the effectiveness of the new images.

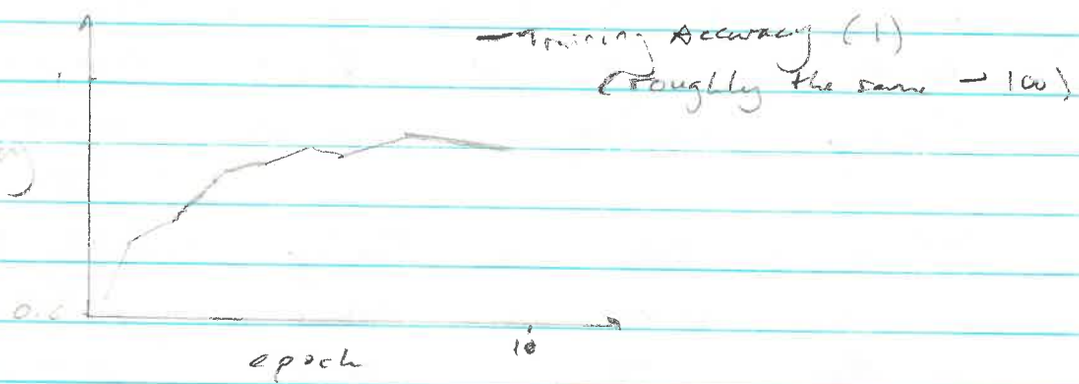
↳ Same advice as before to utilize the pretraining scripts.

also scripts/eval_all.sh for downstream evaluation.

Testing larger window sizes

→ the effects of increasing the window size are surprisingly small.

→ going from delay time steps of 1 ($\times \frac{1}{\text{sample rate}}$) to 100 ($\times \frac{1}{\text{sample rate}}$) still results in fairly good success rate



→ tested window sizes from 1, 5, 10, 20 50, 10

→ By window size $\frac{100}{\text{sample rate}}$, the window size was ~~$\frac{1}{5}$ of the~~ resulted in only 3 delay time averages being suitable including the initial 0 steps so the images were $\frac{1}{3}$ 0 intensity.

→ Interesting that despite this the accuracy curves didn't became significantly worse.

→ The one more notable difference was that the discernibility or distinguishing chainsaw / engine grew with window size and for the largest window size, the storm was badly confused for the background.

Generating .wav files

- ① Sine, sample rate 22050 Hz, 10 s, frequency 440 Hz
  ② Top hat, sample rate 44100 Hz, 10 s or between 3 and 7 seconds (44100 Hz is CD quality standard good for sharp edges (top hat))
  → Used $\frac{SR}{2}$ for sine to save memory.

  ③ White noise, SR: 44100 Hz, 10 s currently only or only off (could try $\frac{1}{4}$ / periodic).

How the time and resulting spectrogram shape is handled:

time-axis size of the spectrogram is given by (time × sample rate)

$$\text{frames} = \frac{\text{Samples}}{\text{hop length}} + 1.$$

hop length ← could be resulting to vary.

⇒ Solved time offset

↳ problem was on the samplerate or the specshow librosa function, this must be the sample-rate of the original audio in order for the image to be labelled with a correct time axis.

In order to modify delay-snag to take time as S as an input:

$$\text{frames} = \frac{\text{Sample rate} \times \text{time}}{\text{hop-length}} + 1$$

Unnecessary $\left\{ \text{time} = \frac{\text{hop-length}\,(\text{frames} - 1)}{\text{Sample rate}} \right.$

So feed function time and then convert to the index using this equation.

remove the +1 when specifying a number of indexes to store

$$dt = int(rint(\cdot round\left(\frac{SR \times time - store}{hop}\right)))$$

(this could cause an issue if dt rounds to zero)

→ Some delay examples

Some wave tables record (on powerpoint) to help that it's not build up of rounding errors?

OR issue with the throwaway at the end if dt is not an exact multiple of the length of the array.

→ think must be this, if method shows problem will need to address this.

→ note sample rate 16000 does not seem to be anywhere any thing? → Maybe will only increase SR not decrease?

hop knob is not a sensible audio signal for this!

SR is impacting the resolution of the delay / there is small scale variation that appears when increased so interpolation says keep at 16000 but this is not trivial.
The other sample rates should be tracked as they keep the same axes sensible.

18/2/25

Progress:

1. Example plots

- have implemented examples mainly of modulated white noise such that the frequency bins are all populated. These are modulated with periodic step functions or sinusoidal envelopes
  → could try to think of more examples

- The results of the delay image transformation preserves the frequency of the patterns in the spectrogram representation but appears to smoothen step functions and general fluctuation.
  Some appears to go to one

- One major thought was that the cut off due to keeping the averaging sample size large enough eliminates ~ 35% of the original time axis. When the patterns are so obvious such as constant frequency or off, it is clear to predict / you could extrapolate the trend further out to delay time = time.
  → nice to see if there was an option to match axes.

2. Coord Ok download

- in progress, each file ≈ 2GB and there are ~ 20 so it will take some time.

19/02/

Meeting

① Continue with current approach until 12/03/25 (Candice event day)

② No point in increasing window size (not to help diff

③ try to see if can attribute success to the cutoff sample size (right now $\frac{1}{3}$)

④ After 12/03/25 pivot point of whether to look into health audio delay time stuff or move to the episode approach to see if can find some golden time seq of episode in which the algorithm performs best!

OPERA

I believe I have got on task 11 with the delo images rather than normal and get these res

Model:	Opensmile	VGGish	Audio MAE	CLAP	OPERA CT	OPERA CE	C
AUC	0.494 ± 0.054	0.590 ± 0.053	0.510 ± 0.021	0.559 ± 0.049	0.725 ± 0.05	0.707 ± 0.00	0.60

also

appears a lot better

20/02/25

- Tried Task 10 with the new images and have confirmed that they are giving different results to the OPERA paper.

- Task 10 performed worse with the new images it seems, data is logged in the OPERA file

- Still only able to use OPERACT and CE for the delay time images as GT uses different functions that would require some talk with Evelyn to implement my current method into.

- Figured out how to write and execute .sh scripts which is useful to execute the other tasks.

- Struggling with installing the Covid UK dataset as initially got the wrong one and the size of the dataset is about 120 GB so trying to find a place to put it.

- Should look at the other datasets and tasks.

25/02/25

Progress

→ Lots of difficulty downloading and sorting datasets. But have cracked it now.

→ Still only able to evaluate OPERA CT/CE

→ Have now got bigstore account and have moved OPERA directory here

TASK	OPERA-CT	OPERA-CE
Could do #1	0.551 ± 0.002	0.513 ± 0.010
#2	0.644 ± 0.005	0.636 ± 0.001
#3		
#4		
#5	0.571 ± 0.003	0.599 ± 0.006
#6	0.765 ± 0.001	0.694 ± 0.001
#7	0.789 ± 0.021	0.758 ± 0.006
#8	0.715 ± 0.021	[redacted] tricky.
#9	0.877 ± 0.002	
#10	0.725 ± 0.005	0.707 ± 0.004
#10	0.685 ± 0.009	0.688 ± 0.032

26/02/25

Definition of AUROC:

ROC Curve: A plot of True Positive Rate vs False Positive Rate at different classification thresholds

AUC (Area Under Curve): The total area under the ROC curve, which represents the model's ability to rank positive instances higher than negative ones.

AUROC ~ 0.7 - 0.8 → Fair to good classification
AUROC = 0.5 → No discrimination, random chance.

Finish Evelyn for next weekend. ✓

Focus on having delay time down and see if the performance (time) improves.

↓

Make sure to do the thing where the feature that gets thrown away increases and compare with Spectrogram → seems decently promising to be fair.

The plan

Make it so that I can do classification with a dataset → look in on specific task maybe count UK.

→ fitted auc as a function of image size (and subsequently run time?)

→ have this by next Wednesday.

27/02/25

→ Having difficulty OPERA CE Task#8 input sec =2 but the framework is not creating files that give the input sec = 2 part.

→ Need to get an idea creating this way of testing the speed/accuracy with image size.

28/02/25

Going up with the pretraining stuff for the tonic being, focussing on testing for speedup.

My cleanup code should be easily adapted as on the OPERA analysis, the audio files are all processed and saved so I have arrays with corresponding label arrays and train test splits.

↓

bits of this in linear_eval.py and {dataset}_processing

Process

① the pre processing of the spectrograms.

② the assigning of labels

③ Sub straight into cleanup code.

Have successfully preprocessed set of Covid Uk data (Task #1) and fed through training with labels.

The labels were taken from covidUK_eval / and potentially not aligning as expected as the

results are giving just a flat curve.

① May need to read csv. file myself and create a label list.

② Also might have to redig all the chainsaw ML because that was designed for four categories and potentially better options for binary classifications.

③ Still going array → png → array which could likely kill the speed up benefits of delay image and so will need to go further in depth with this.

01/03/25

- Checked number of +ves and -ves in the dataset currently using. There are 884 positive cases and 1616 negative cases — could be issue?

→ Next try with spectrograms not delay!

02/03/25

Tried using spectrograms and running into same issue, can't seem to break past the ~0.64 limit which indicates the model is just picking one category every time to maximise accuracy.

→ Try with trimming effects!

05/03/25

Meeting

- Joined by Evelyn, I hadn't trained a model so up to now what I had been doing was fairly hopeless.

- The google net training approach hadn't really made any ground, still unable to see growth on the validation data accuracy indicating over-fitting

- Need to: go deep into multiple-pretrain and prepare-data.sh.

  Have been going through each dataset to prepare the delay images - quite likely I will have to modify this and do again.

(5) Problem with Covid UK, only have 'downsample' file names and supposed to have normal.

(10) Problem (potentially) with big lung, big blacked out (small value) region at top of each Check delay image, maybe these frequencies aren't spec present in ~~the~~ big lung data? representative.

- Might also need to reduce the largest delay time. (relate: max delay time to grad-throw!).

(5) - Mentioned not to have gap between windows be larger than 0.45

Have done icbhi, coughvid and hf_lung so far.

Need to check, dcbhi does get_individual_cycles for one function so should see why that is necessary.

① Problem Covid-19 sounds is a nightmare so many files to download → just play away 1 by 1.

② dcbhi also does extract spectrograms → need to add make_dcbhi_time.

09/03/25

→ Nothing too major since last update except I now have all the datasets and have converted all into delay spectrograms.

→ Now processing on multiple pretrain.sh and having issues, currently main is

"EOFError: No data left in file".

↳ located the problem, coughvid file: UID - 26811_t4RWR...

↳ I have removed this from necessary arrays.

10/3/25

→ Number of epochs was not meant to be 512, actually 100, I exceeded this so it will not take 4 days!

→ There are cropping elements to the pre processing that do not suit delay images i.e. if the audio clip is too long it will take a random clip of the right

lengths so as to bypass this

→ all images go in with shape (256, 64) with delay, or I could naturally reduce this potentially / at least remove the elements of random cropping that could improve the process.

↳ Note: already junction crop - just that could easily sub for random crop.

→ Issues with OPERA CE and OPERA CT doing random cropping as part of the method and encountering shapes lower than the max time lengths presumably due to delay effects

OR if its a random crop and starting at the end, it might not have enough remaining time to get the correct lengths.

→ Yes, wherever the lengths of the delay sample is not long enough, the random crop fails. Currently there are low instances of these failures for now so can just discard but

↳ could also reduce the max length by the factor getting thrown away as an idea

Probing 'bad files' (ones of less than optimal max len):

(icbhi cycle : 1 in 5024) incorrect.
(covid UK exhalation : 1 in 1800.)

covid 19 sounds breath : 1470 out of 33766 ~ 4.3%
↳ fixed → 452 out of 33766 ~ 1.3%

covid 19 sounds cough : 32 out of 43677 ~ 0.07%
↳ fixed.

icbhi - zero bad files
icbhi cycle - 882 out of 8024 ~ 18.7%
covid UK exhalation : 295 out of 1800 ~ 19.6%
↳ fixed
covid UK cough : 129 out of 1499 ~ 8.61%
↳ fixed.

lung : 0 out of 10554 ~ 0%

11/03/25

Progress

→ Big revelations: Evelyn has sent the correct training files for covid UK.

→ Need to retrain the input_sec and padding functionality on the get_entire_signal_librosa because otherwise clips get through that are too short and random crops on OPERA CT and OPERA CE fail.

13/09/25

Audio Day

→ Spoke to someone saying that using mel spectrograms was detrimental in machine learning and that is a predisposition from a human perspective that biases certain frequencies and the idea is for the cough and breath data we are trying to use ML to capture the hard to spot characteristics of respiratory data that can distinguish it.

↳ try and see how linear spectrograms could get onto everything.

→ Pietro saying we don't want any cropping we are just going to fix the delay tone that we go to

→ I think I should direct my attention on the generative OPERA GT model first and then ~~move~~ see what I get.

Notes

→ hf lung stethoscope and tone ↓ data doesn't contain frequencies passed ~ 1000 Hz.

→ icbhi too, seems all stethoscope data loses the high frequency data in the averaging process

14/05/25

Progress

→ Finally OPERA CT has trained using delay images.

→ In ~~OPERA~~ extract_opera_feature amended the use of get split signal to get entire signal then cropped the post length to the desired length.
→ for OPERA CT.

Aside:

→ for the paper after conversations at audio day
Some motivations / introduction:

Paper?

The use of mel-spectrograms is commonplace in the field of respiratory audio data analysis leveraging AI. We question whether their use is logical with AI and respiratory audio where AI doesn't hold the same bias as only detecting using human senses and in order to search for hidden time frames we shouldn't be using frequency time representations that are suited to human ears and instead be more general

→ Issues with now extracting feature as requires 2500 audio clips of the correct length (8 seconds)

→ Now padding (with zeros I believe) to ~~1.5 seconds~~ 1.5x the input second length, then the delay image will cut that back to the correct length (or greater)
→ will result with all images coming out at correct length of 8 delay seconds.

→ the next big step will be to modify the size of the image used throughout all of OPERA-CT down from 256 to smaller and smaller and see how performance holds up.

16/03/25

The OPERA CT feature extraction and benchmarking ran but failed for most of the tasks.

For now I believe debugging will be too time consuming for little reward as all the processes are designed to optimally process spectrograms and are not general to images. It also takes a long time to run pre training and other tests which makes mistakes costly so goal I need to find another way.

→ Have found an open baseline method for analysis of the ICBHI dataset (lung sounds) that seems easy to play around with.

→ Currently the accuracy isn't great nor does it seem to be growing so much for mel_spectrograms but it seems much simpler and faster to run and also looks at acoustic features so I think it has potential.

↳ Would be useful to implement transfer learning to the current CNN and FNN methods

→ ~~I definitely want to~~

17/03/25

Papers

One very similar to my work:

A Novel Melspectrogram Snippet Representation Learning Framework For Severity Detection of Chronic Obstructive Pulmonary Diseases

→ 'It has been investigated that as spectrograms contain immense information of audio signal it is difficult to extract promising feature from image based CNN models [45]'.

→ YAMNet has been trained on the mel-spectrograms extracted from the Audio signals of Audio set, which is the largest dataset for audio deep learning.

Idea

→ In the acoustic feature vector creation aspect of the code, there is the option to add 'deltas'.

↪ deltas are how all of the feature values have changed from the window before to the next.

→ In the same style could add another 'deltas' set where instead of the difference between the features with the window before, do the deltas of fixed delay times and do my averaging technique!.

18/03/2

Feature investigation

Currently, there are 84 features evaluated at 155 windows

the windows are twice the loop length I believe meaning there is ½ overlap between one window to the next.

The implementation of delta's : True is essentially the shortest delay time implementation of my idea.
→ All I have to do is add delta but one, delta but two, delta but three etc

18/03/25

- emailed Pietro a summary
- Returned to OPERA

Owen GT

	AUC
Task 1	0.549 ± 0.006
Task 2	0.594 ± 0.003
Task 3	0.673 ± 0.001
Task 4	

	AUC	OWEN GT − OPERA GT
Task 1	0.549 ± 0.006	− 0.056
Task 2	0.670 ± 0.001	− 0.007
Task 3	0.594 ± 0.007	− 0.019
Task 4	0.673 ± 0.001	± 0
Task 5	0.571 ± 0.002	+ 0.019
Task 6	0.761 ± 0.000	+ 0.026
Task 7	0.646 ± 0.007	− 0.095
Task 8	0.654 ± 0.020	+ 0.004
Task 9	0.867 ± 0.000	+ 0.042

AUC

Task 10	0.693 ± 0.012	− 0.010
Task 11	0.716 ± 0.024	+ 0.110

↔ Way better again.

→ Note lots of symmetry on Goswami delay images.
↪ Could be cut down and retain eyes.

20/03/25

	Task 12			
	Task 13	0.906 ± 0.575	+ 0.014	
-ve is good (lower)	Task 14	0.764 ± 0.595	− 0.061	
	Task 15	0.124 ± 0.137	− 0.004	
	Task 16	0.859 ± 0.570	− 0.019	
	Task 17	0.851 ± 0.581	+ 0.077	
	Task 18	0.133 ± 0.145	+ 0.003	
	Task 19			

26/03/25

Set training running over weekend with max-len zeroed for all datasets at 64.

→ made various changes to codes accomodating all input-sizes to be 2 seconds of delay time

↪ Query on the image size (256, 64) being independent of the max-len chosen.

→ will need to check this, have emailed Evelyn.

Youtube video on audio features

Traditional ML

→ hand pick set of audio features and pass through pipeline

Deep learning

→ Just pass the whole audio representation into the pipeline.

↪ haven't looked at doing this.

Valerio Velardo - useful channel

→ something to consider, the delay images aren't normalized between 0 and 1.

29/3/25

→ Pietro happy with 2-3 delay times + keeping OPERA random but my images look crap.

→ Followed up with Evelyn and Tong to make sure they are happy with their course of action.

→ In the mean time going to create a function to engineer delay time features.

Currently:

→ function stores previous feature vector values to compare with current feature vector and loops

→ need to create a 2 x feature_dim array that stores the previous 2 features and then compares the earliest feature vector with current and then re assigns the later feature vector back one step.

↳ doesn't make sense to change the feature extraction function - make a new one to manipulate the matrix output of feature extraction.

8/4/25

Weekly update

→ I created the MFCC delayed Δ style feature and saw minimal increase in accuracy or trend with delay time.

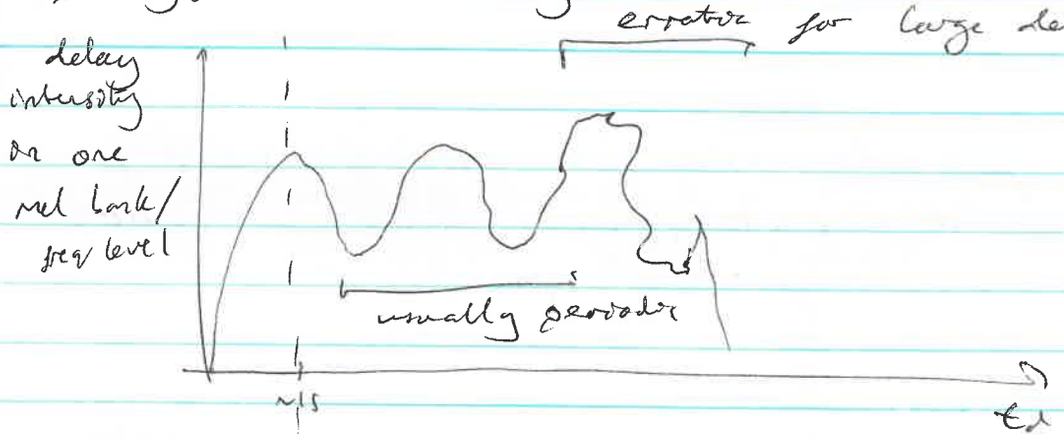
→ Have been reading around papers and found that the covid 19 sounds original paper has an accessible github so is ideal to use for my feature engineering

→ Spoke to Pedro about feature engineering and he would like to see an exponential fitted to the onset of the delay signal up to the first peak. From this two parameters can be extracted

$$F(t_d) = a_0\left(1 - \exp\left(-\frac{t_d}{a_1}\right)\right)$$

fitting function

t_d ~ delay time



onset.

→ The OPERA models are still in progress but training has sped up so hope to have all by end of the week.

9/4/25

Call with Pietro

→ to deal with diggicult time scale values can just pass a jilter / only keep a certain range of frequencies

→ SR = 22050) ÷10
n_fft = 2205

Max frequency = $\frac{SR}{2}$ = 11080 Hz

Frequency resolution = $\frac{SR}{n\_fft}$ = 10 Hz

Some clips only have to ~ 3800 Hz and others more like 7600 Hz.

→ initially going to just use 0 – 3800 Hz of values i.e. only take first 380 y axis values.

→ Also for the tables I'm producing, would be better to create figures like a scatter graph.

→ Could just increase time resolution and lower frequency in order to get less features and better number of samples for fitting consider that time scale is what we're after.

OPSRA problems with crop lengths, have now resolved to say if input sec = 8.18 or 8, t = 256

2 sec → 64

5 sec → 160 forget the discrepancy between 8 and 8.18 for now.

11/4/25

→ A couple images appear blue to start with [0, 0, 0 ...] vector, investigate...

14/4/25

time resolution ~ $\frac{\text{Hop length}}{\text{Sample rate}}$ ← (secs samples).

Have decided on gust delay goatwee to best:

gust: $f = a_1\left(1 - \exp\left(-\frac{t}{b_2}\right)\right)$

order : 10 (0.255 either side).

hop_length : $\text{int}\left(\frac{SR}{20}\right)$

Frame length : $\text{int}\left(\frac{SR}{10}\right)$ = window length

Sample_rate : 22050 (SR).

delay_seconds : 3s.

will append (mean, median, std) × 2 to the extracted features

timescale + multiplier
(a_2) (a_1)

→ 6 extra features for now.

→ nevermind, if I do std for the functions on the multipliers and timescales, then I get 11 features from each param

→ 22 extra features.

Seems to be very clear that these features are making the performance worse...

20/4/25

KDD paper wasn't ideal for what I wanted to do

→ not a great baseline as lots of parameters were varied and then the results were cherry picked.

However neurips covid19 sounds paper has a good openly accessible code and easy to modify opensmile + SVM pipeline that I can add my delay features to.

→ the exponential fits aren't working as nicely as I'd like → need to speak with Pietro.

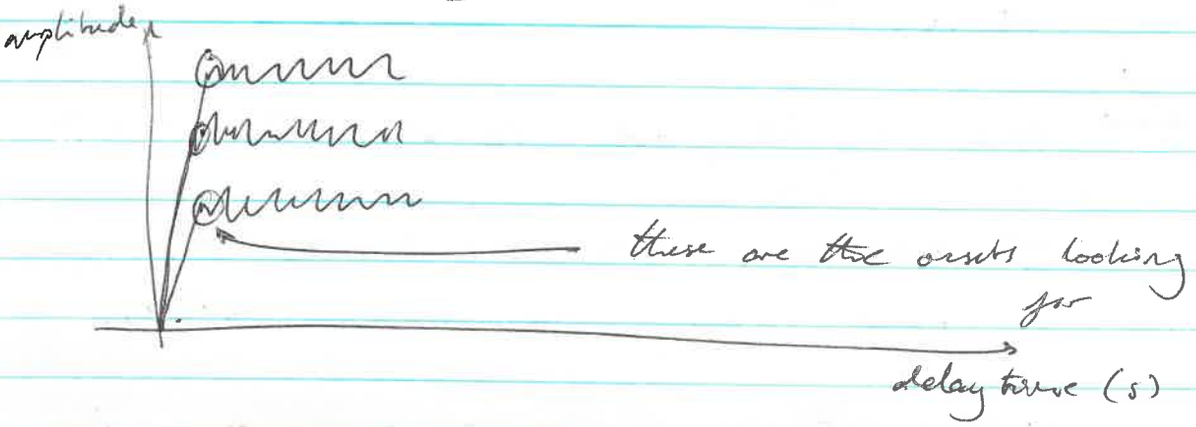
Lots of updates in the Powerpoint from this week! → Figures and comments.

23/4/25

Call with Pietro

① Can't say much more on OPERA front.
→ Train 3 models for each length (2 extra per) to try and probe if the nature of evaluation on ⑦ and ⑪ is correct

② Feature engineering: Focus on voice data



→ was this style that Pietro had been expecting at the beginning

→ will lead to reduce order parameter

→ Evaluate the frequency dependence of parameters.

28/4/25

Problematic breath indices - 142, 2753, 7240

7/5/25

Post-minors update

→ Implementing tonescale, multiplier and power features has considerably introduced improvements to the SVM classification albeit by only ~ 0.5% AUC on the ~~largest~~ highest AUC case

→ Not too convinced that tone scale is doing a lot for the classification, there are very outlierish values to get the curves

↳ Need to try and combat this.

→ potentially over parameterising by allowing power and tone scale variables.

→ Haven't explored smoothing the delay signal to avoid picking out the wrong maximas.

→ Proposed to take the delay onset tone (end-index) as the tonescale and then allow the power fitting to give the curvature required.

↳ Have implemented these changes as well as modifying process to look for a threshold gradient rather than explicit turning point

↳ So far the values look much more reasonable in comparison to the raw data and the fits are good.

→ Going to try and start writing up Destroys

Method layout

Table of Contents

Introduction (How does it fit into the bigger picture) → Maybe try and talk some physics like how the sounds generated by the throat and how they are modified by the presence of mucus acting like a filter.
↳ learn more about this.

Literature review → maybe introduce the papers I am following and how they fit into everything.

now → Method

- How I make the images (general)
  get a figure explaining how to go from sound clip to delay spec.
- Feature engineering → ISO? (maybe 6???)
  → timescales, averaging in Opensmile... SVM
  (parameter values)
- OPERA
  → MAE, linear evaluation → (will take some work to break this down)
  (parameter values)

Results + Discussion

- Feature engineering
- OPERA

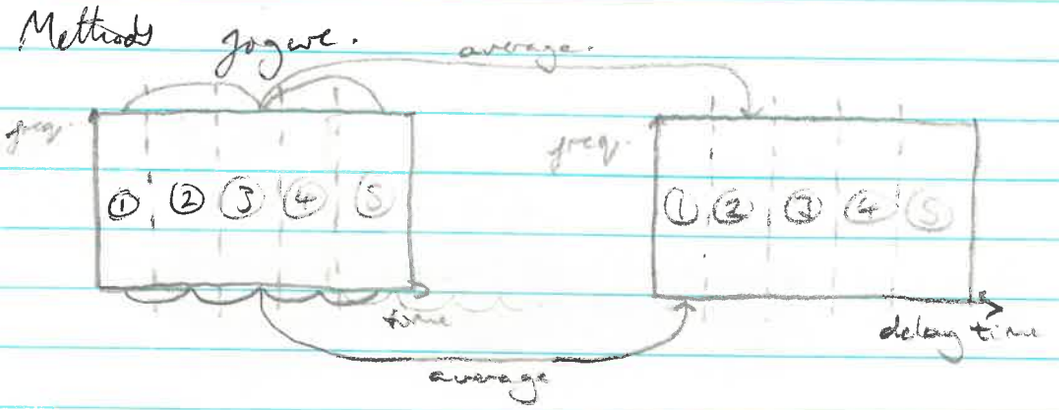
Conclusion

References

Appendices.

→ Might be an idea to seperate two chapters are for feature engineering (methods + results + discussion) and are for OPE same style.

Methods go here.



8/5/2025

→ by making changes to gitlong yesterday, performance has improved.

→ just realised that by using different loop len and frame lengths to original, opensmile feat are slightly worse → should probably change these back for better comparability.

→ Currently extracting VGG ish feature to concatenate with my delay feature and see if they add there.

10/5/23

Something I overlooked is the extraction of eigenvalue and other feature plots in OPERA itself that I can use as baselines.

↳ I have implemented my delay features but with this frame and hop length, the time resolution is not great
→ I'll have to play with these parameters
→ gradient threshold been modified to 0.5

↳ The figure is nearly complete, make the axis labels bigger and make the titles bit neater.

Intuition on delay images

- As you delay the image, the decorrelation is going to grow and grow.
- It ~~was~~ is limited to the max amplitude - min amplitude or this case (2 degrees of freedom)
- This is why the decorrelation saturates.
- The timescale of this saturation (the time at which the audio is completely uncorrelated) it could hold information like some time scale related to the random nature of oscillations in vocal tract / lungs or something
- This is related to aircraft or things that affect how the sounds we make, hence we propose that finding this time scale may add to classification.

11/5/25

Feature engineering + OPERA datasets.

→ tried engineering features for coviduk, covid19sounds and coughvid.

→ ST works better for breathing now with the smoothing approach potentially

→ have seen decline when implemented to coviduk and covid19 sounds so far which is weird.

→ seems to have slightly improved AUC for gender classification with coughvid through power
↳ again, ~~multiplier~~ is strongest feature

AUC before : 0.672 ± 0.007
AUC after : 0.679 ± 0.004

→ DCBSHS better, Oswave better

! Think its definitely worth changing to mel-spectro for this experiment → more focus on lower frequencies.

Notes on the generation of sound.

air coming from the lungs generates 'subglottal pressure' which counters the resistance given by the vocal folds (muscles and ligaments attached to vocal chords).

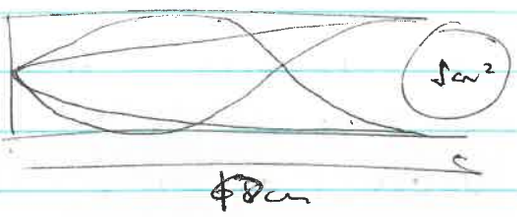
The alternating opening and closing of the vocal cords due to the force balance forms the basis of sound generation from (larynx).

the factors that affect the frequency of vibration are the mass, length and tension of the vibrating structures.

laryngeal muscles modify the amplitude, fundamental frequency and waveform of the glottal pulses. Men use lower frequency (more mass) → slower oscillation, women higher, children highest.

↳ considering a more complicated model of coupled oscillators due to the interconnected muscles involved in generating sound introduces non linearity and 'chaos'.

Mouth:



One end open tube

$v_{sound} = 346 m/s$

Fundamental frequency = $\frac{346}{(17 \times 4)\times 10^{-2}} = 500 Hz$

you get resonances due to the nature of the vocal tract → I know this.

All the sounds are the fundamental frequency + harmonics modulated by the laryngeal muscles to change the amplitude of those present.

We introduce aperiodic sounds commonly by introducing turbulence to the air escaping the tract for example the 's' sound by constricting the tongue to leave a narrow gap (turbulent flow).

12/5/25

AUC, SPE, SEN

→ Have included sensitivity and specificity to the CBERA metrics.

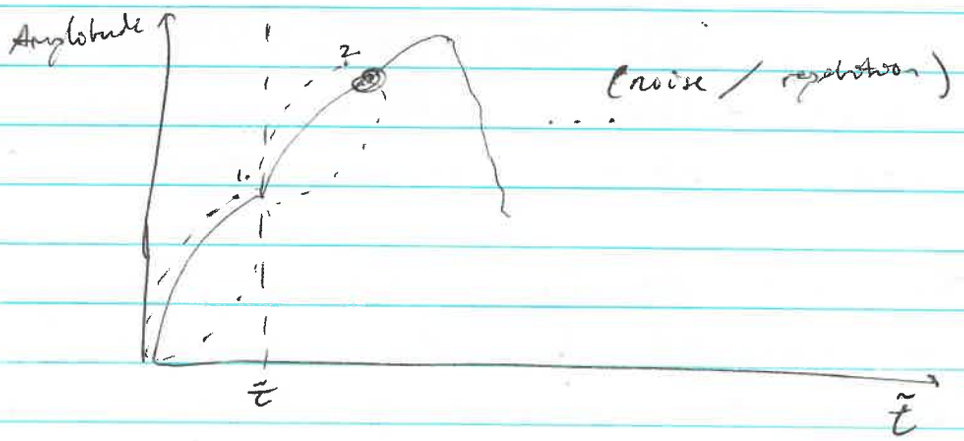
→ DCBHI is much better on all metrics.

Mel Delay Images

Made the transition to mel delay images, it seems logical to target the lower frequencies / weight the timescales there because these are where the sounds come from

→ Main changes,

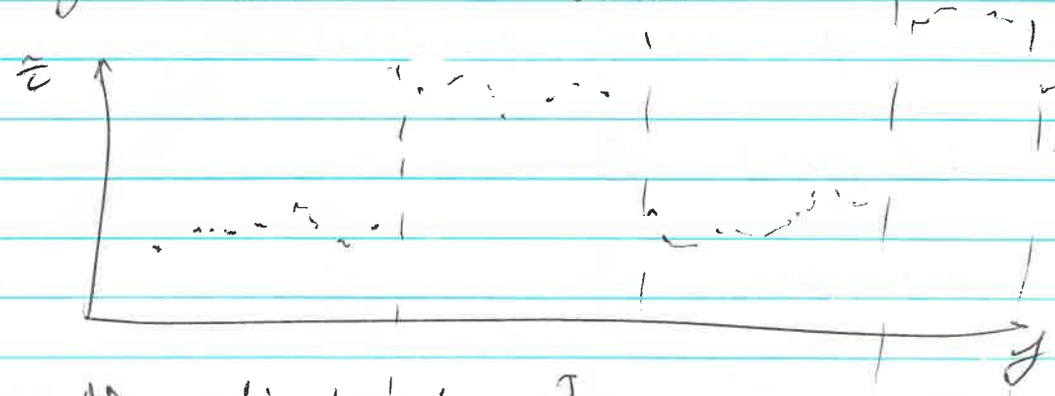
- created a cap on successive timescales to prevent the peak finder from locking on to successive onsets. Cough onsets look like so



I believe the onsets stack due to some timeframe on the data and the most important to catch is the first / need to stay consistent.

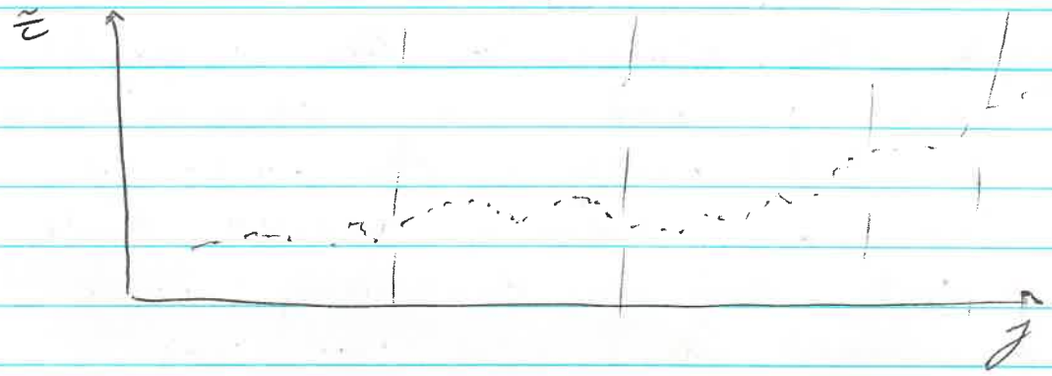
I think this because the timescale with frequency scatterplots have large jumps associated with catching other maxima

and if these were cut out, there would be a more natural trend, hence I constrain the location of consecutive timescales



No adjustment

Adjusted



14/5/25

Meeting with Pietro

→ Add random labels to DCBHD to make sure AUC improvement isn't random.

→ Place the delay profile shapes in results. (example of each.)

→ Find what the timescale values correspond to in frequency and any literature on what they correspond to.

↳ Think the timescales are on the order of how the laryngeal muscles will flex to adjust the 'formants' location / the

fundamental frequency (as an approx to probing these frequencies themselves.)

→ take average values of timescales for both classes
→ box plot! ✓

Some more intuition on delay

phenomena at delay times from our analysis correspond to a timescale within the data.

For example, the timescale in which changes to the laryngeal muscles occur in order to ~~form~~ articulate different sounds by modifying the position of the formants or producing voiceless sounds would manifest itself in the delay profile at some timescale at which these changes usually occur at.

↳ Paper suggesting average syllable length 226 ms
≈ average timescale I've been finding.

~~After~~ At this timescale, we propose that the delay - amplitude difference should separate as when displaced by this amount, the delayed spectrogram is likely to be a most decorrelated from the original.

→ Breath timescales are very slow, very periodic behaviour not changing the breathing rate at frequencies between 0.5 - 80 Hz
→ unlikely to gain much info

However, you can imagine that by conglom

Sounds or speaking sounds, the muscles flex and modify structure multiple times per second i.e. between 1 - 50 Hz

! Make emphasis on formants and the role of flexing mouth muscle to convey sound changes.

→ This should be a powerful emotion recognition feature.

One change to delay feature; I'm extracting:

→ 90% overlap quoted on a paper make little difference but increase pared density (going for that which minimises the tradeoff on frequency). + 4kHz low pass.

→ Trying out with NewSPS.

15/5/25

Example condition delay could pick up:

Pectoriloquy - abnormal increase on the clarity of whispered sounds during auscultation due to the presence of lung consolidation, which increases the sound speed and thus lowers the damping of the vocal signal due to the efficiency of transmission through fluid.

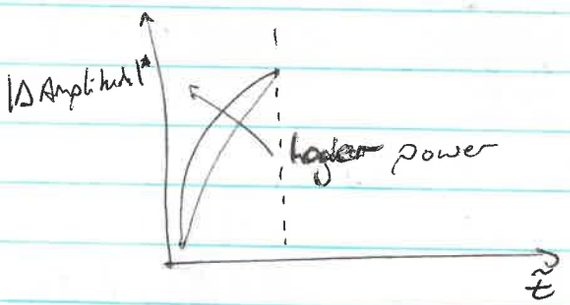
↳ Lowered damping resulting in higher amplitudes of higher frequency voice components which would alter the character of the delay onset at these levels.

Also Bronchophony work is the same but for spoken words.

In these cases, delay could pick out from a sentence whether someone was healthy affected by liquid in the lung / vocal tra by the nature of the delay onsets at certain frequency bins due to the fact that the liq alters the efficiency of transmission in these frequencies.

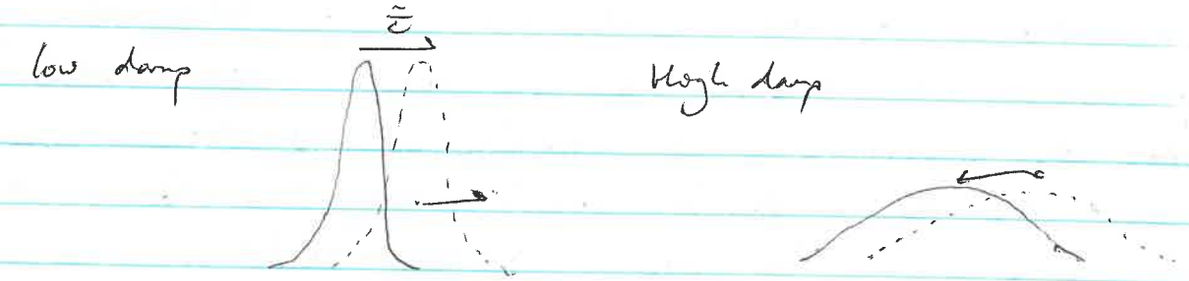
General reference from Doctor ↑ very good.

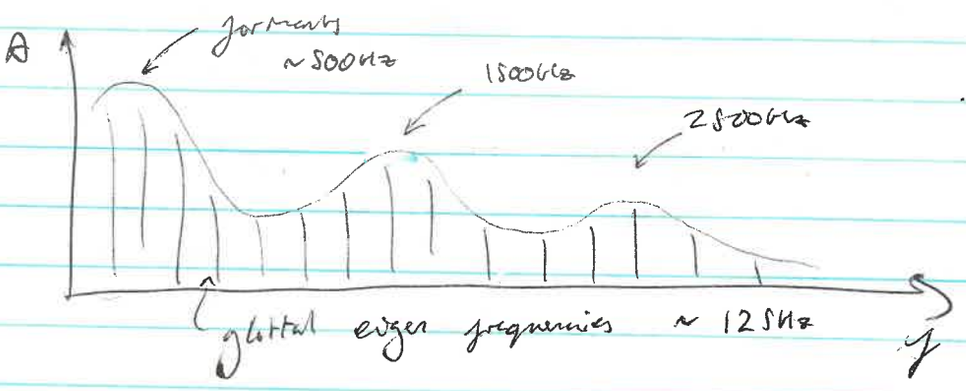
Why might Power parameter be higher for someo healthy vs. someone not healthy for cough?



lower power means delay rose faster for someone unheal

If the higher frequencies are more damped, variation in their presence / not presence will be lower





If you sharpen the higher frequency harmonics and thus move them around on the order of 0.2 s, the correlation between the frequencies present before and after will be a sharper difference hence the decorrelation would rise faster for more pronounced formant peaks

↳ the presence of liquid or general changes to the structure of the thorax during illness could then influence the parameters extracted from the delay onset.

generally associate lower power with more rapid rise in decorrelation

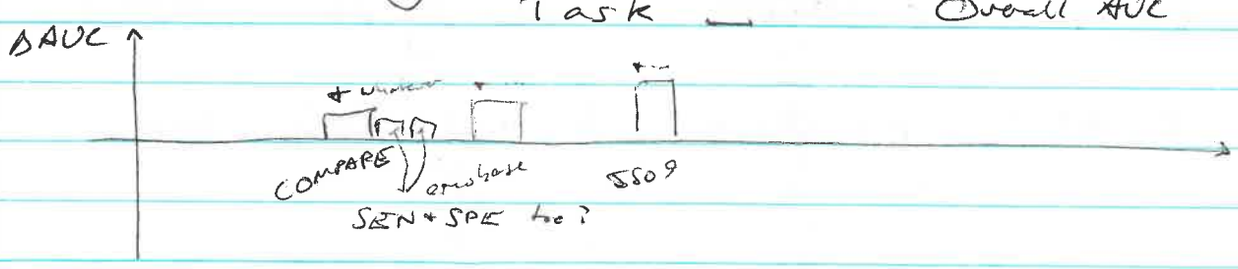
↳ could plot this / make an appendix

→ faster rise in decorrelation → more stop stark / what was described earlier.

FCHI → stethoscopic data.

How to display the AUC values with and without delay.

Task — Overall AUC



→ Table of values in appendix

Features to extract

Coswara: 6377 + 988 + delay { Cough, Breath

Covid 19 sounds: 6377 + 988 + delay { cough, Breath

Coughvid: 6377 + 988 + delay

MF lung: 6377 + 988 + delay

ICBHI: 6377 + 988 + delay

Coswara: 6377 + 988 + delay

KAUH: 6377 + 988 + delay

(For same 988's, need to remember old delay features are appended still.

Got all AUC SEN SPE values for New IPS,

3809 c = 0.001 (only power parameter)

envelope c = 0.001 (only power param)

ComParE c = 0.0001 (all features).

16/5/25

Covid UK cough C = 0.0001 Task 1

Covid UK breath C = 0.0001 Task 2

Covid 19 sounds cough C = 0.001 (0.0001 for 6T?) Task 3

Covid 19 sounds breath. " Task 4

Coughvid Covid C = 0.0001 ~~Task~~ Task 5

Coughvid gender " Task 6

icbhi → only power of value. C = 0.0001 Task 7

Coswara Sex C = ~~0.001~~ 0.0001 for emo Task 8
C = 0.0001 for 6T7T.
(Smoke omitted)

KAUH C = 0.001 for emo. Task 9
C =

Modality based ~~x~~ ~~x~~ y lims.

Cough : timescale - 0 - 0.8
Amplitude - 0 - 100
~~Breath~~ : Power - 0 - 1.2

Breath : timescale - 0 - 0.8
Amplitude - 0 - 80
Power - 0 - 1.2

COPD timescale - 0 - 0.25
Amplitude - 0 - 15
Power - 0 - 0.8.

17/5/2

Lore Plots

For coughs: (multiples of 400Hz)

take 1 at 120 Hz (~ glottal pulse f)

take 1 at ~~500~~ 520 Hz (~F0)

take 1 at ~~2000~~ 1600 Hz (~F1)

3 per plot.

~~take 1 at~~

400Hz resolution so order [1, 13, 100

The reason that there is such variability on the delay onsets is that centre were asked to perform three coughs or 5 or whatever. The spacing was not very uniform at all and in some cases it all blends into one.

Poeppel's References

1) phases of oscillation and the amplitude of vocal oscillation have proven to be a good indicator of Covid-19 / more likely just respiratory disease.
Bottom line: COVID-19 causes asymmetries in vocal fold oscillation (disrupts entrainment) resulting in discriminative features for COVID-19 in symptomatic people.

↳ Could the asynchronization manifest as a periodicity on a timescale > f glottal that co increase the rate of onset in delay featu

~~Could sounds plots 5's choosing.~~

ii) tension asymmetry between the two vocal cords results in modulated signals due to the coupling between the vocal cords causing non linear behaviour

↳ evidenced that vocal cord coupling can create glottal signal modulations of order 5 glottal pulse periods.

↳ tension asymmetry as a result of muscles / organisation (back to (i))!

5x 120Hz → modulation frequency of 240Hz within our range.

↳ How would this non linearity manifest itself in a power spectrogram?

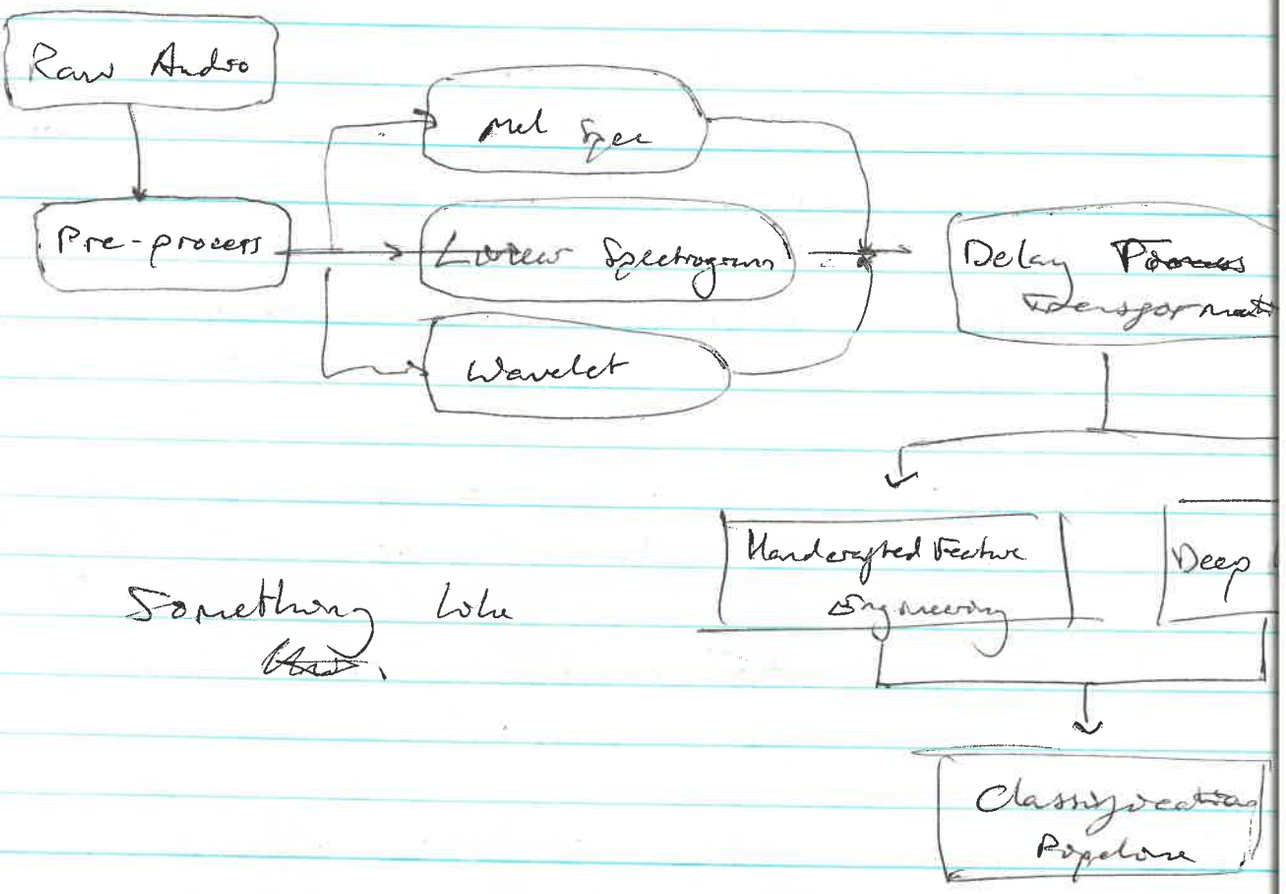
One massive flaw in this approach of the non uniformity across datasets

- Vocal differences between users make it clouded what delay features are from someone being ill / being different from other people and the sounds they make.

- ~~Also~~ The uniformity of the audio task can vary a lot for example if someone isn't in a quiet environment, someone coughs 3 or 4 or 5 times or the spacing between their coughs all result in unique characteristics to the delay feature making it

hard to notice trends.

Flow Chart



Something like this.

18/5/25

How could I attempt to probe the ~200 Hz region of delay.

time resolution has to be $\frac{1}{200} = 0.005$ s

4 s clip → 800 samples — doable.

capped at ~ 8000 samples. (4 Seconds)

$\frac{16000}{200} = \underline{80}$ hop lengths.

19/5/25

~ 20% increase in AUC observed by increasing time resolution and honing in on $\sigma \sim 0.1$ time scale for all modalities

↳ this could be tracking modulations on the order of the global pulse!.

20/05/25

→ Call with Pedro

→ Try range of Hop and Frame length to evaluate the AUC increase.

→ CNN (image classifier)

HOP+FRAMELEN.

→ What is the ~~AUC~~ of opensmile?

↳ n-fft same (1024) and hop is (160).

→ It's okay to put OPERA for on each year.

→ Try 512, 256, 128, 64, 32

changing hop length is non-sensical, it means that the jittering moves to the length type peaks where I anticipate there isn't so much information.

Will try n-fft instead

↳ 1024, 512, 256, 2041

Delay image equation?

$$F(\omega, \tilde{t}) = \int_0 |F(\omega, t) - F(\omega, \tilde{t})| \, dt$$

$$F(\omega, \tilde{t}) = \sum_t |F(\omega, t) - F(\omega - \tilde{t})|$$

$$\sum$$

delay spec. ← $\tilde{F}(\omega, \tilde{t})$ = $\sum_0$ $\frac{|F(\omega, t) - F(\omega, t+\tilde{t})|}{T - \tilde{t}}$ → Spectrogram

T ← length of clip

21/5/25

Opensmile is failing on some instances / clips
↳ not sure why only seems to be for 6373 features. and 384 each at off.
↳ would conclude that its the hop_length change but then why does it work on ~3000 clips before it?.

Potential RQs to frame my discussion around:

Can these delay features become standard features?
↳ Lack of real generalisation to different datasets / context - music, emotion detection ...
↳ having such fine scale images could become problematic for computation although it is not uncommon to use different scales for different features;

↳ more rigorous method of jitting required / are these the optimal parameters → could potentially try log of omitted onset to jit linear and extract power law?.
- preclude possibility of holding delay have region constrain i.e. 0 - 0.1 s and forcing jitting.

~~How~~ to Would deep learning contribute to their value.

↳ talk about what failed with OPERA

↳ Currently the value is in the smallest part of the start of the image
- maybe for extremely large passages of audio, this could be an effective way to reduce data size and still find value.

Potential Applications

- Audio spoof detection - natural voice is very unique to people, this feature may capture unique characteristics of voice not ~~shown in other~~ replicated in spoofs.

Indices for f bins: 360 Hz 2000 Hz 4000 Hz

↓ ↓ ↓

~~88~~. 128 256

22/5/25

Can delay features become a standard?

→ They have proven to be strong distinguishers of a respiratory illness.

You should discuss achieving AUC's of up to 92% on classifying whether you previous maximum being 75%. Clearly tells us value of being found useful about that current OpenSMILE features sets + VGG the feature results from original paper cannot make → rate ⟷ AUC from Evaluation results.

↑ decorrelation ⟹ more likely respiratory illness.

→ HOWEVER,

~~Generalization to other tasks and data~~

Process of fitting needs to be refined

- the threshold gradient approach is too variable due to problems of either locking onto the cough length timescale or else.
- The compensation between timescale and multiplier, clearly some reduction can be made. Maybe the ratio of two of the parameters is better than both.

Dependence of parameter results

- The fine resolution spectrogram process is sub optimal. Don't think that it is necessary to go so small hop-lengths.

~~Not sure~~ because of the ~~peak~~ saturation criteria were agreed and more consistent, the same jobs may result from having a ~~scatter~~ more tone resolution.

Frame length of 1024 as same as COMPARE so not needing to change that. Attempted to modify hop for full comparability but it failed on a few occasions.

look to this but or that I don't think that research is necessary if the gradient identification process can be improved.

Once these points are ironed out, then would need to generalise to multiple datasets
↳ could reference Coughvid Cosvara ...
↳ other data (COPD stethoscopic...)

* Also, the frequency dependence might be of importance for speech / cough for example. Just taking the functionals was out of convenience in OPERA. There might be information in the positions of the largest tone scales / features could be rounded up by them
↳ this could be echoing F0 ... formants but clearly work to seeing if there are better ways than OS functionals.

Once all of this addressed, then will have but critical results suggest that they could provide great benefit to respiratory audio.

→ Potentially deep learning discussion

- explain why I think my results aren't meaning ful
- suggest ways in which it could be used to delays advantage

→ Applications outside of respiratory audio.

- Spoof detection
- ~~Audible~~ looking away from the onset at different ~~feature~~ timescales that could benefit ML on other audio datasets.

24/5/25

- Bug development — ~90% AUC was due to a change in process halfway through extra features. This invalidates claim and were back to ~ 1% improvement.

25/5/25

WI as down...

→ tomorrow hopefully back:

- try feature extraction for ing timescale at 20 Hz → remove the feature entirely
- could MFCC approach achieve anything.
- try without gradient as a feat

④ Back to Energy (dizz yuetsoreds?)

4135 – 4226 + [5488 : 4

⑤ Back to spectral + cepstral

4227 – 6376

Feature type	Indices
Energy	[0 : 248] + [4131 : 4226] + [5488 : 5580]
Spectral + Cepstral	[248 : 3658] + [4227 : 5166] + [5648 :
Voicing	[3658 : 4134]

thank this → most likely where information is contained.

separate spectral and cepstral.

Spectral : [248 : 1519] + [1955 : 3224] + [4223 : 5166] + [5488 :

Cepstral : [1519 : 1955] + [32248 : 3658] + [5166 : 5488] +

Redo :

Feature	Indices
Energy	[0 : 248] + [4131 : 4227] + [5488 : 5548]
Spectral	[248 : 1519] + [1955 : 322] [4227 : 5166] + [5548 : 6
cepstral	[1519 : 1955] + [3224 : 36 [5166 : 8488] [6167 :
Voicing	[3658 : 4131]

28/5/25

Results: Spectral + Delay is best feature

What plots to include:

- Feature distributions on their own.
- Maybe baselines + Feature types in same figure?
- Try vertical stack frequency distributions.

Report: Further recommendation, established featuresets may not be the best for respiratory and custom sets targetting in this case spectral features can out perform.

30/5/25

Sample: 23, 24, 25 - comparison loop length

Can find Pitch data showing influence of loop length and smoothing isn't necessary.

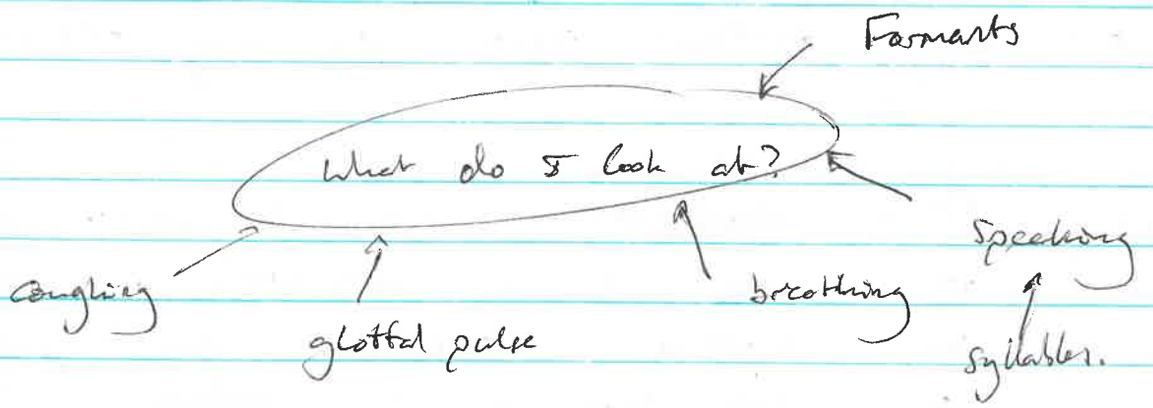
Table of inverse frequencies

~~Forecaste~~

Sound	Typical Frequency
Human Hearing	

Acoustic Phenomena	Typical Frequency
Human Hearing	20Hz – 20 kHz
Syllable	~ 20 Hz
F1 and F2	300 – 750
glottal pulse	120Hz, 225, 265
	60 – 500 Hz

order by length. freq



1/6/25

Abstract:

① Respiratory audio of coughing, breathing and speaking contains a wealth of information that can be harnessed for machine learning healthcare applications. Currently, the standard acoustic feature extraction used to extract this information are outdated. My project..