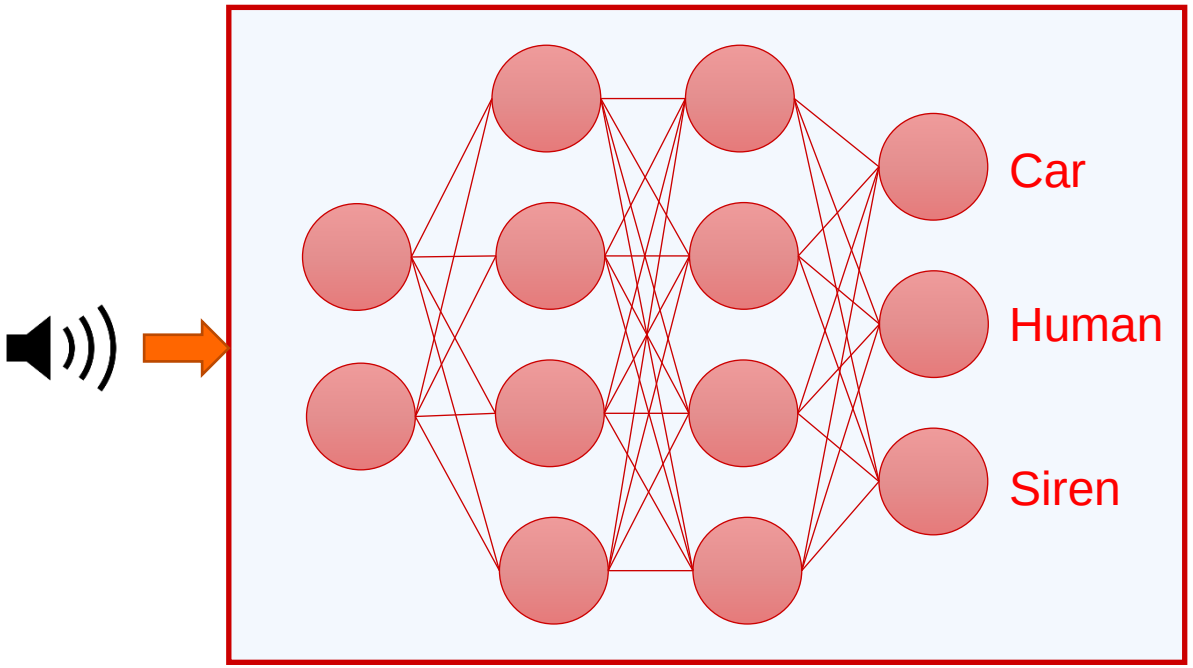
Uncited images created by the author using CoPilot or Adobe FireFly

SINGAPORE INSTITUTE OF TECHNOLOGY

INF2009 – Edge Computing and Analytics [2024/25 T2]

Edge Analytics - Audio

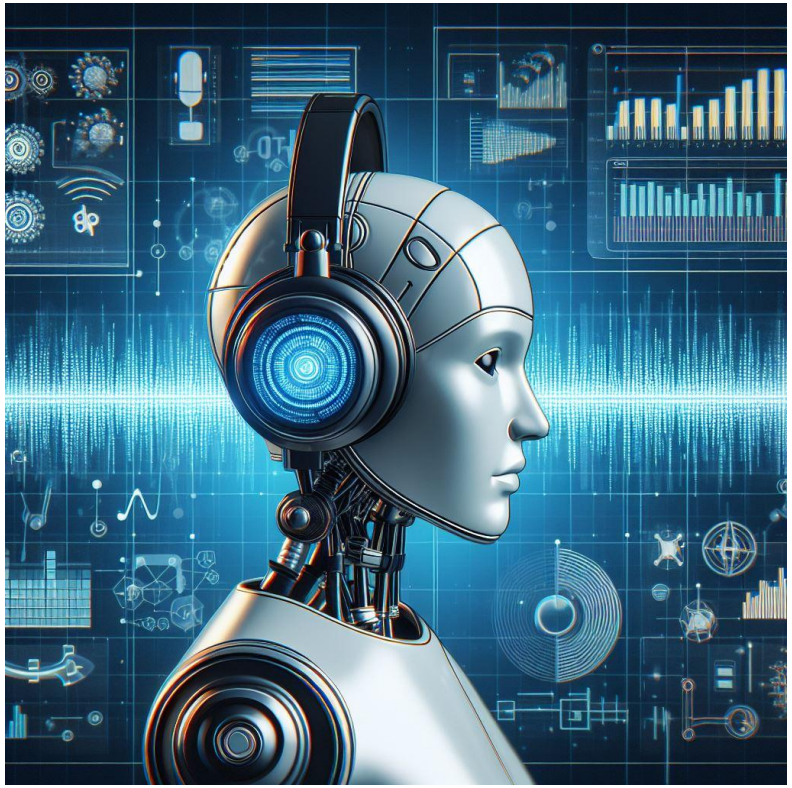
Audio Analytics



Reference:

- Müller, Meinard. *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*. Vol. 2. Cham: Springer, 2021.
- Knees, Peter, and Markus Schedl. Music similarity and retrieval: an introduction to audio-and web-based strategies. Vol. 36. Heidelberg: Springer, 2016.
- Bäckström, Tom, Okko Räsänen, Abraham Zewoudie, Pablo Perez Zarazaga, and Sneha Das. "Introduction to speech processing." (2020). https://speechprocessingbook.aalto.fi/
- https://librosa.org/doc/main/index.html
- https://github.com/musikalkemist/AudioSignalProcessingForML/tree/master

Why Audio Analytics on Edge



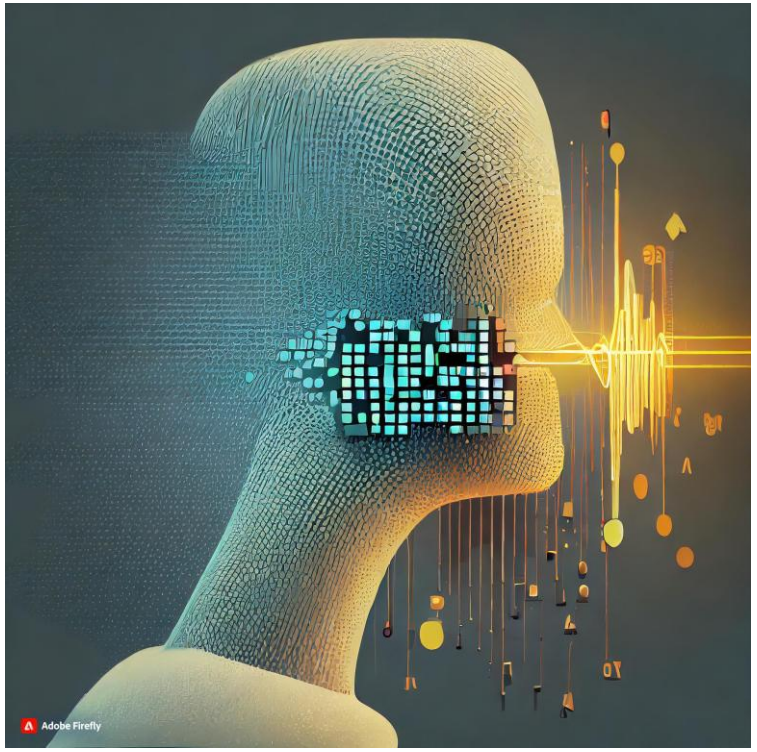
- Low latency is critical for fast action (e.g., audio scene classification)
- Sensitivity/Security of the data is critical for many applications
- Redundancy of the data can be reduced (enable compression) for further analysis (offline)

Audio Analytics on Edge - Applications

- *Audio Scene Classification*
  - *e.g. surveillance*
- *Audio Event Detection*
  - *e.g. screeching brakes*
- *Speaker/Speech Recognition*
  - *Keyword Recognition*
- *Speech to Text (or vice versa)*
- *Conversational User Interfaces (CUI)*

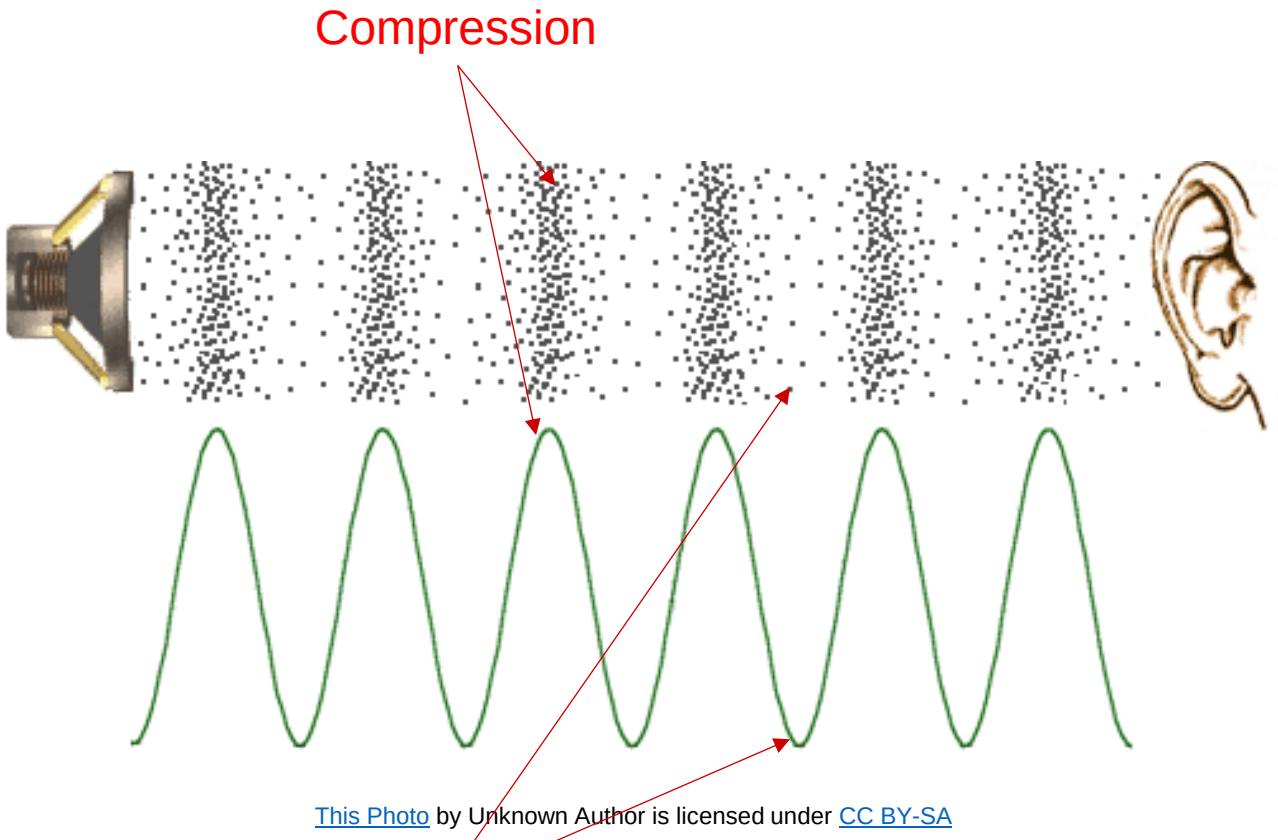
https://www.st.com/content/st_com/en/about/innovation---technology/image-analytics.html

Content



- *Sound Waves and Digitization*
- *Time-domain and frequency-domain features*
- *Audio transformations*
- *Lightweight Machine Learning for Audio*

Introduction to Sound Waves

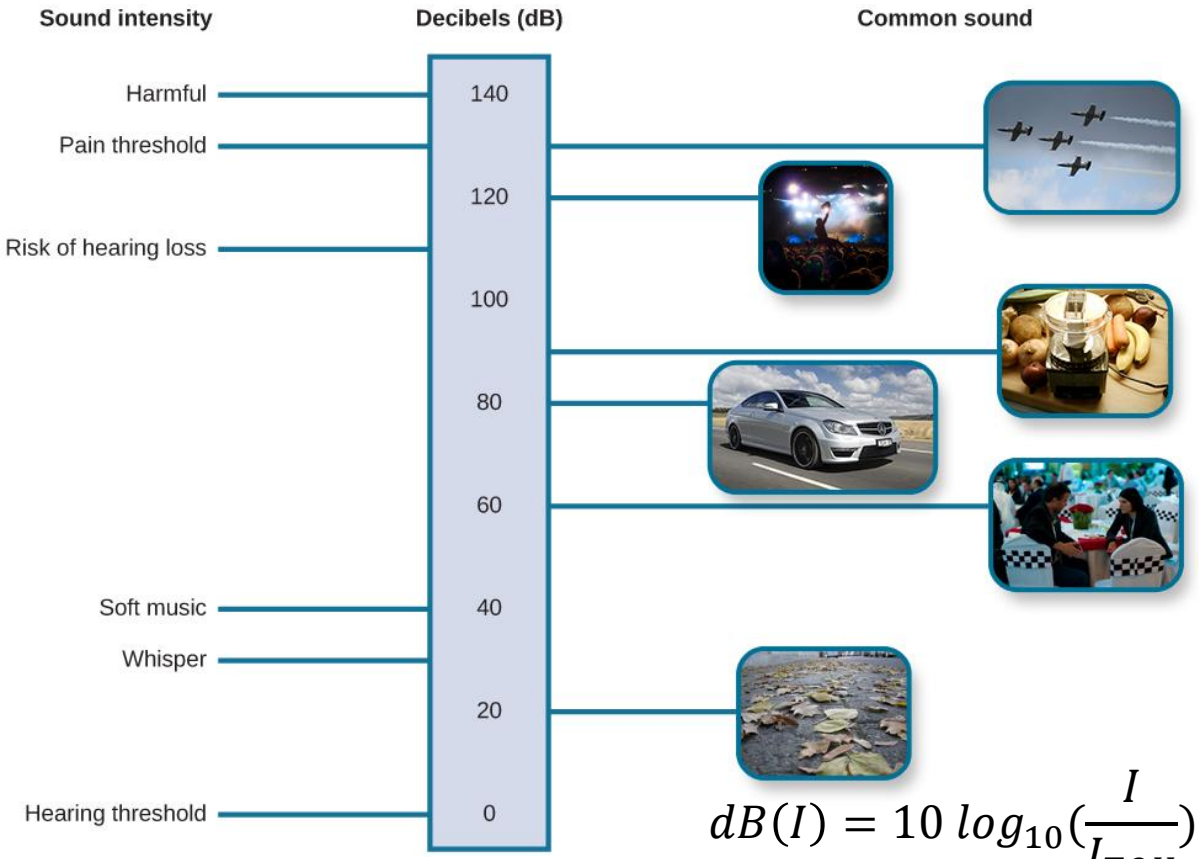


$$y(t) = A\sin(2\pi f t + \varphi)$$

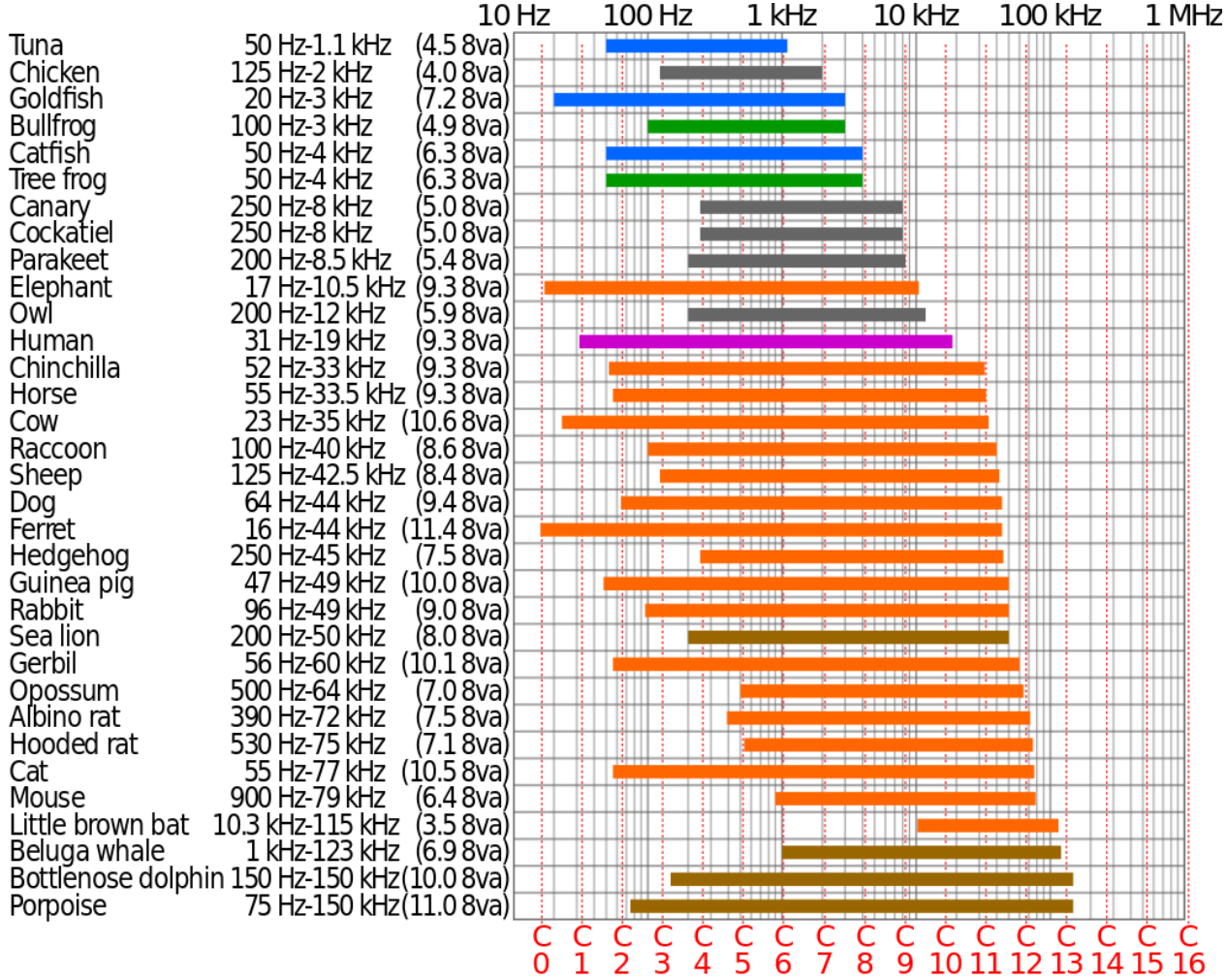
A → Amplitude
f → frequency
t → time
φ → phase

Sound Amplitude and Frequency

https://pressbooks.umn.edu/sensationandperception/chapter/auditory-sensitivity-function/



$$dB(I) = 10\ log_{10}(\frac{I}{I_{TOH}})$$

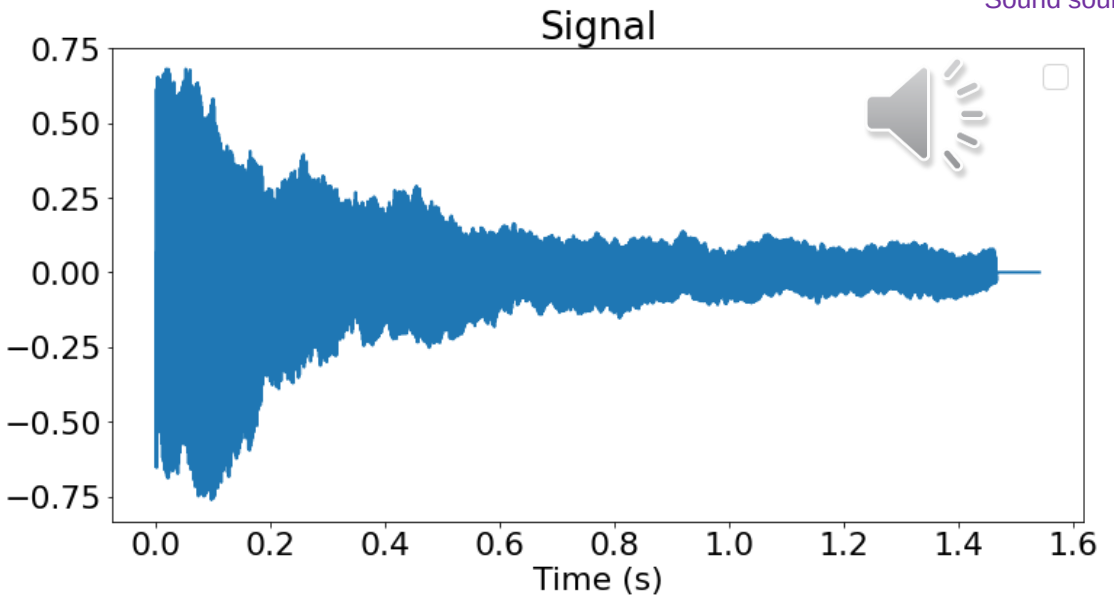


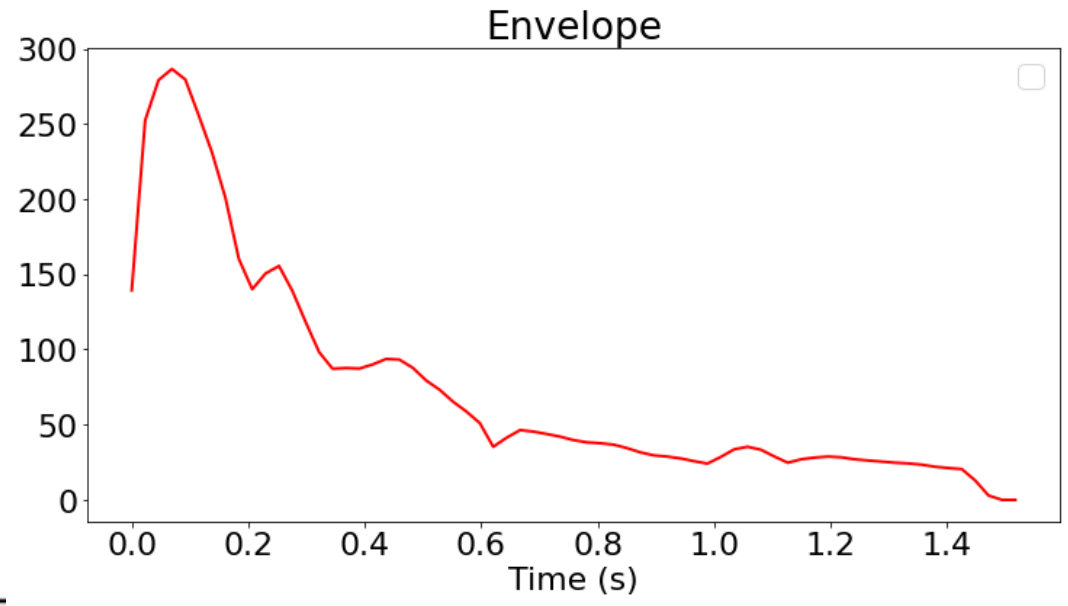
Simple machine learning is to put thresholds on amplitude and frequency (but world is noisy!)

Sound Amplitude (Envelope)

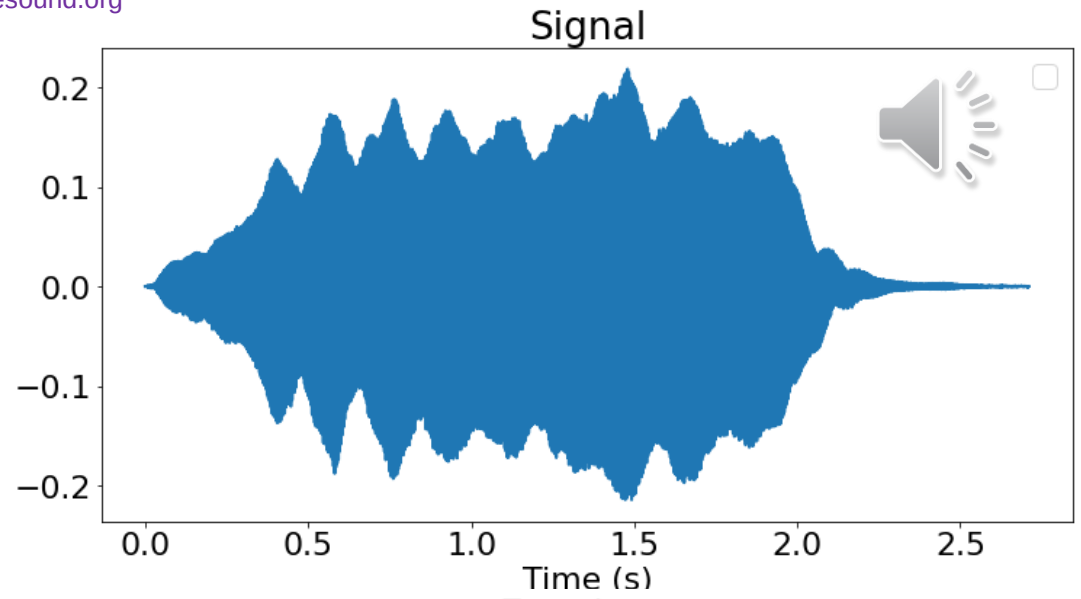
Sound source: https://freesound.org

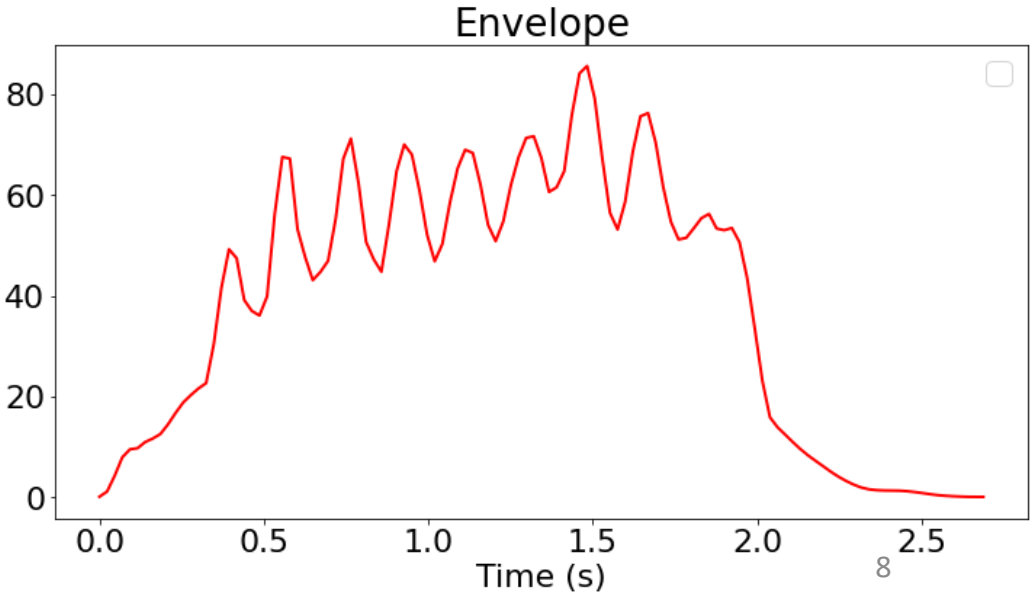
Piano





Violin

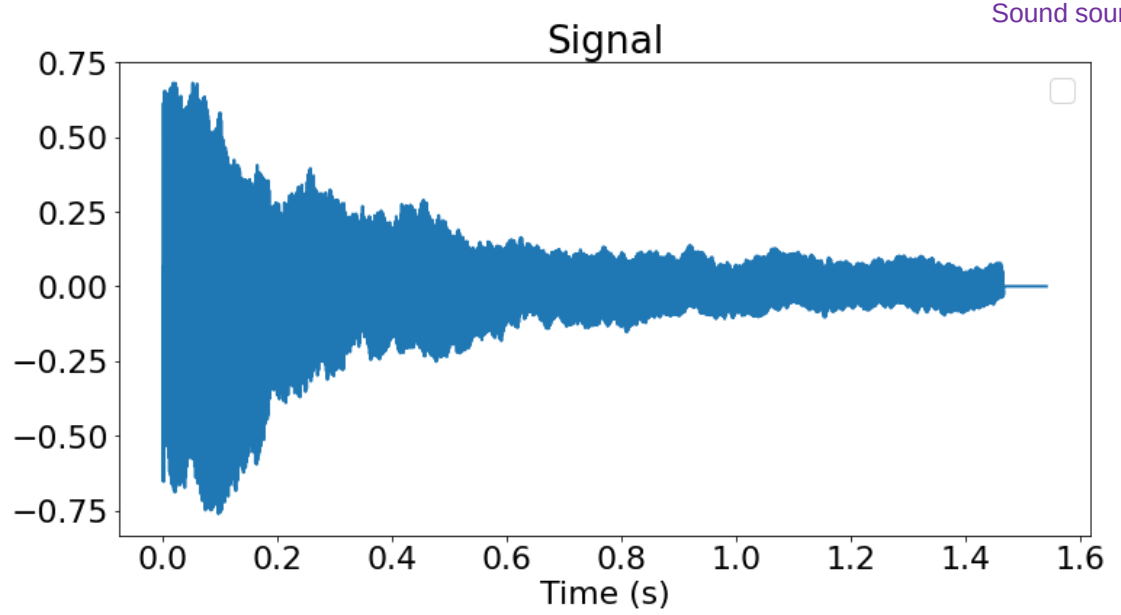


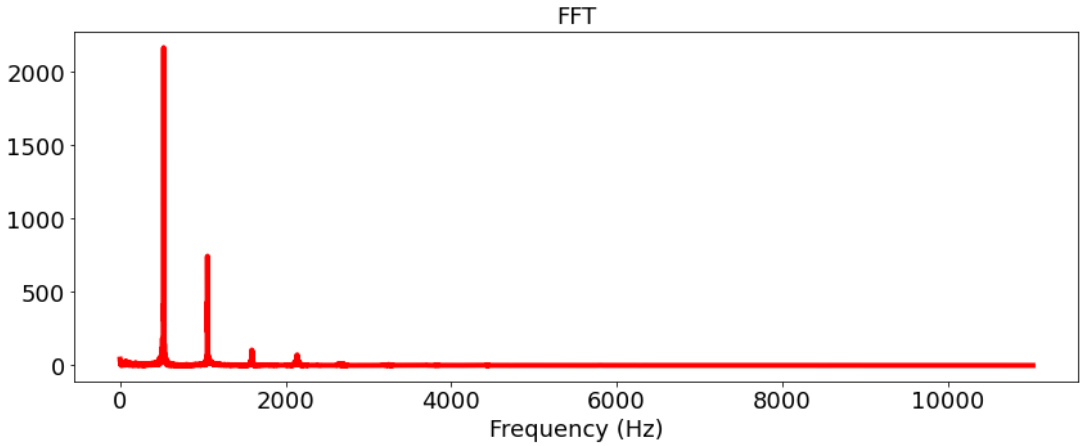


Sound Amplitude (Frequency)

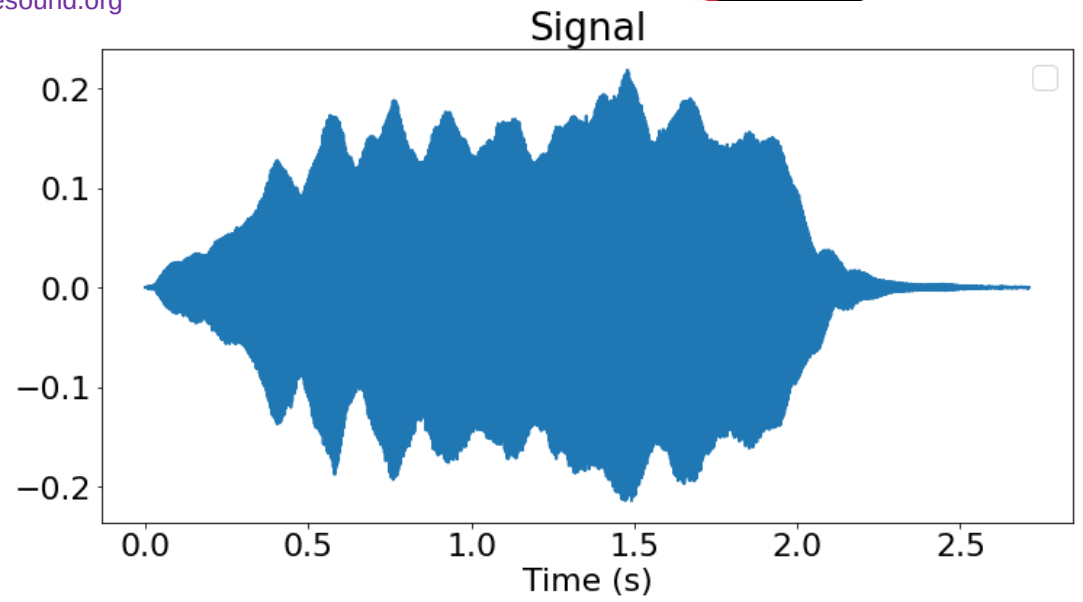
Sound source: https://freesound.org

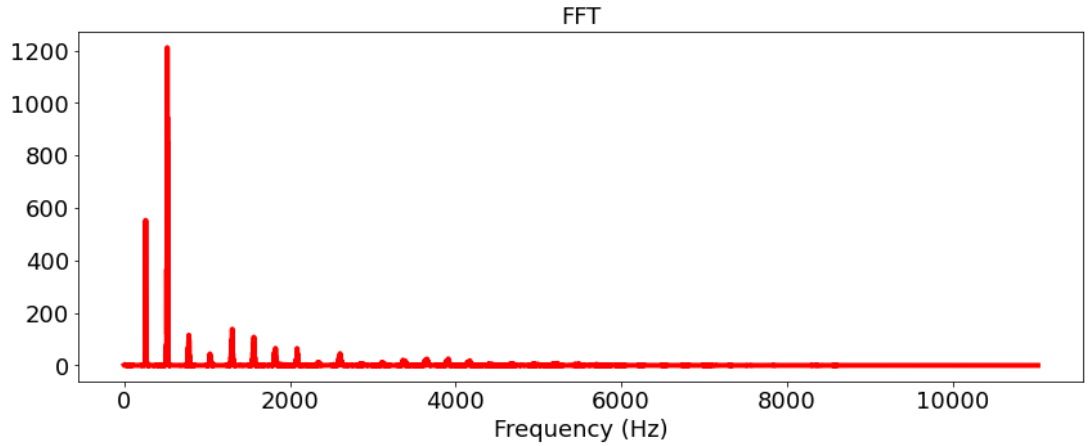
Piano



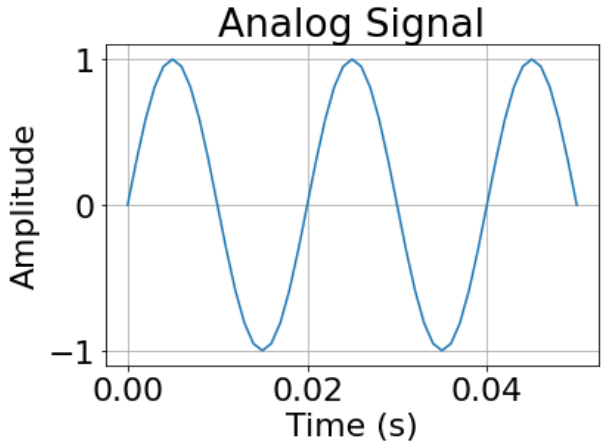


Violin





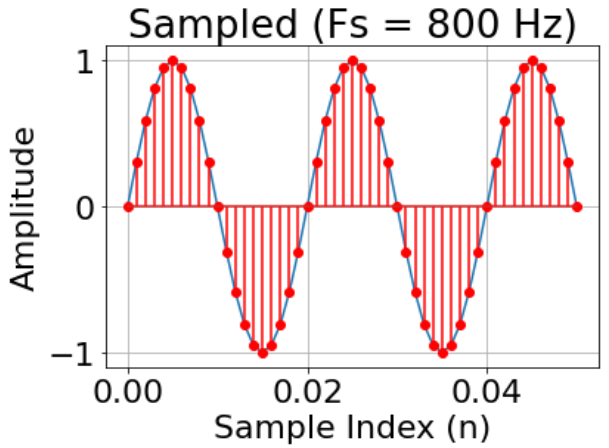
Digitization of Sound Waves

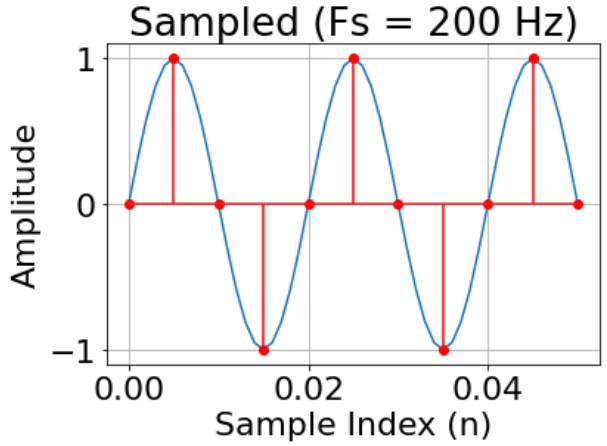


Sampling

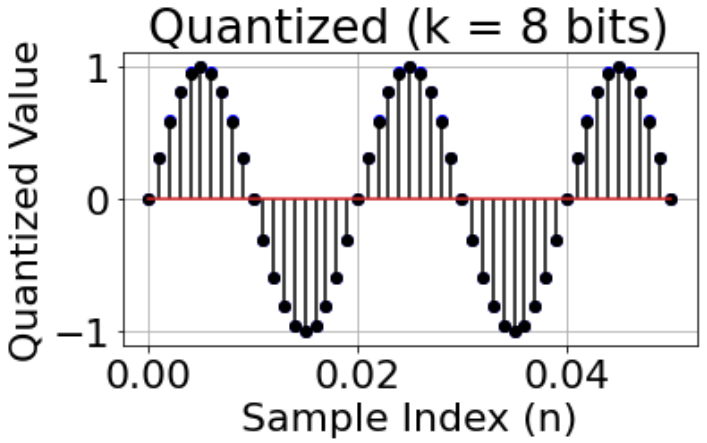
Sound Card (Analog to Digital Conversion)

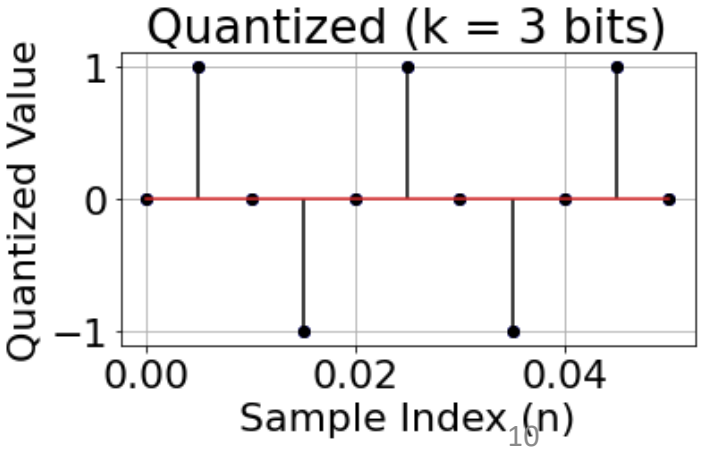
Quantization





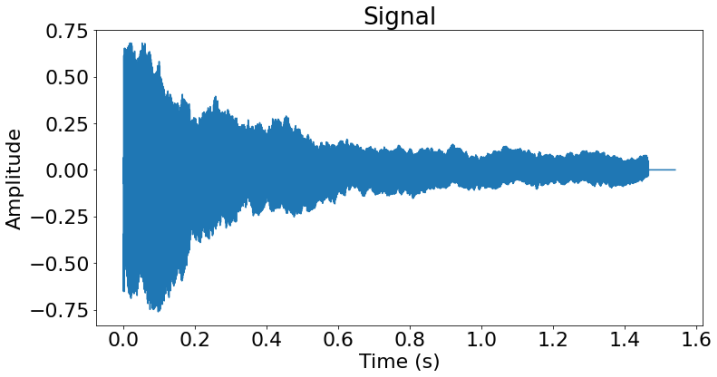
- Sampling frequency > 2 times the max frequency (bandwidth)
  - Nyquist Criteria
  - Sound cards typically support up to 48 kHz.
- Higher bits (resolution) is better!



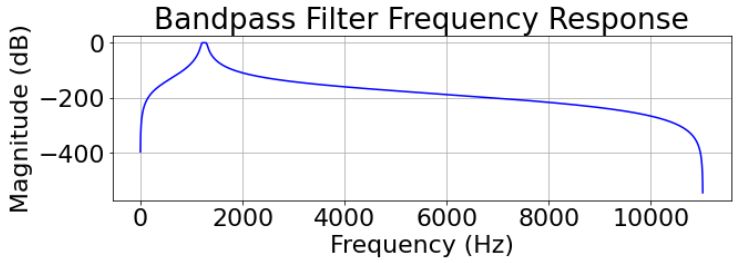


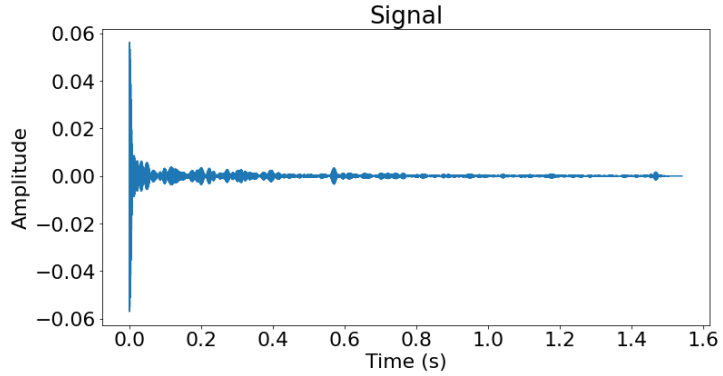
Pre-processing

- Noise reduction: background, artefacts
- Standardization: formats, sampling rate, varying lengths
- Normalization: varying signal amplitudes
- Improved learning models: efficient training, low complexity



Sound source: https://freesound.org





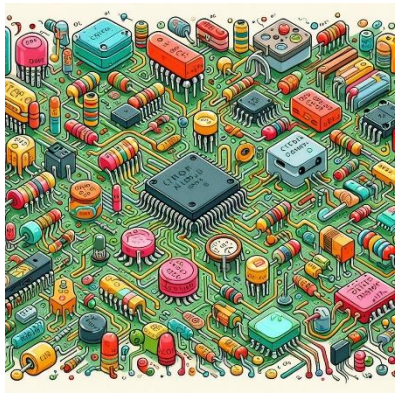
Features from Sound

High-Level

Examples: Melody, rhythm, tempo, genre etc.

Mid-Level

Examples: Pitch and beat related, Spectrogram, Mel-Spectrograms, MFCCs

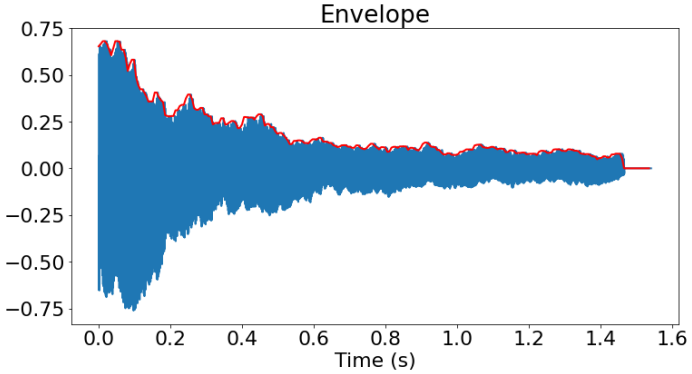


Low-Level

Examples: Amplitude envelope, energy, spectral centroid, zero crossing rate

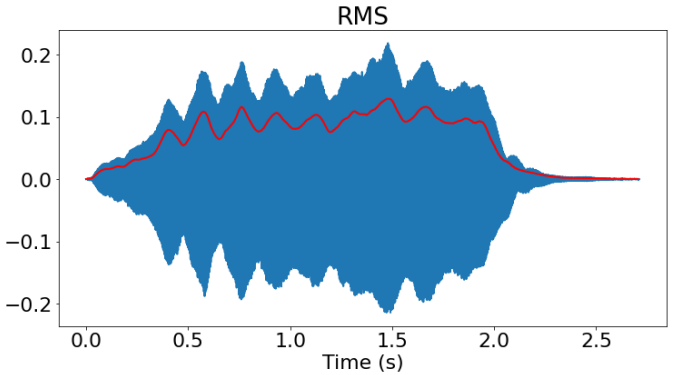
Time Domain Features

Sound source: https://freesound.org



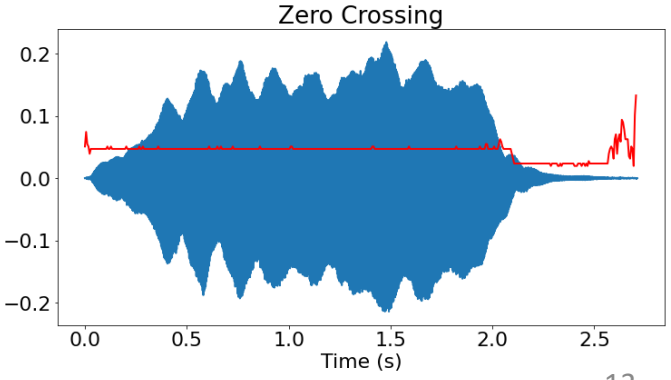
Envelope, $AE_k = max(s(k))\ for\ k - K/2 : k + K/2$

- Divide the total time into small time windows
- Find the max/min/both in the window
- Useful in onset detection, music genre classification
- Very sensitive to outliers!
- Sensitive to noise!



Root Mean Square, $RMS_k = \sqrt{\frac{1}{K}\sum_{k-K/2}^{k+K/2} s(k)^2}$

- Divide the total time into small time windows
- Find the root mean square in the window
- Less sensitive to outliers
- Audio segmentation, music genre classification
- Sensitive to noise!

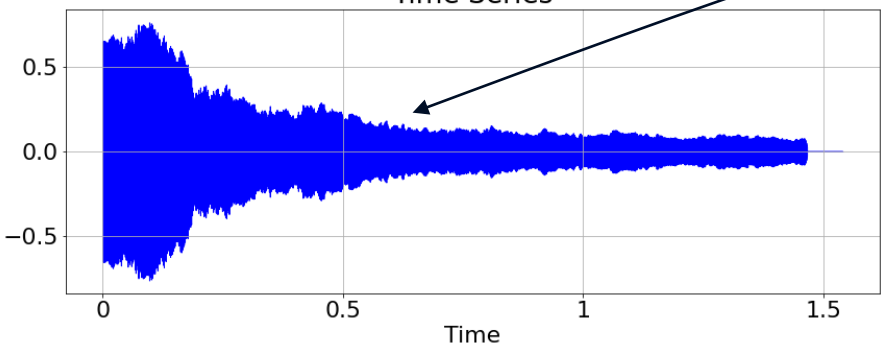


Zero Crossing Rate, $ZCR_k = \frac{1}{2}\sum_{k-K/2}^{k+K/2} \left| sgn\left(s(k)\right) - sgn\left(s(k+1)\right) \right|$

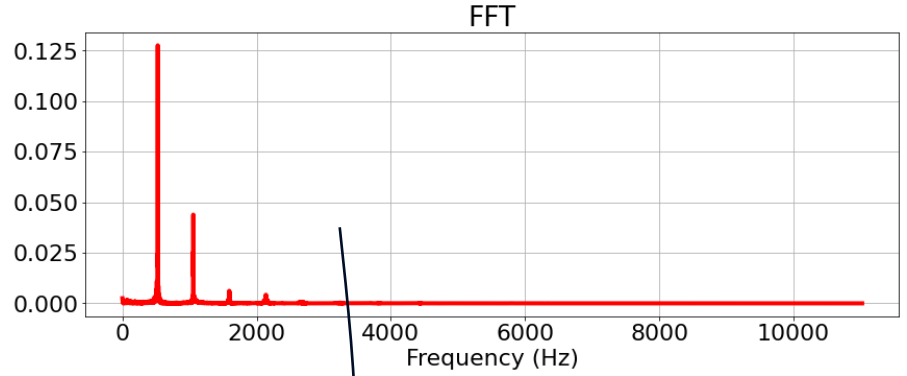
- Number of times a signal crosses the horizontal axis
- Music genre recognition, pitch estimation, speech segmentation
- Sensitive to noise!

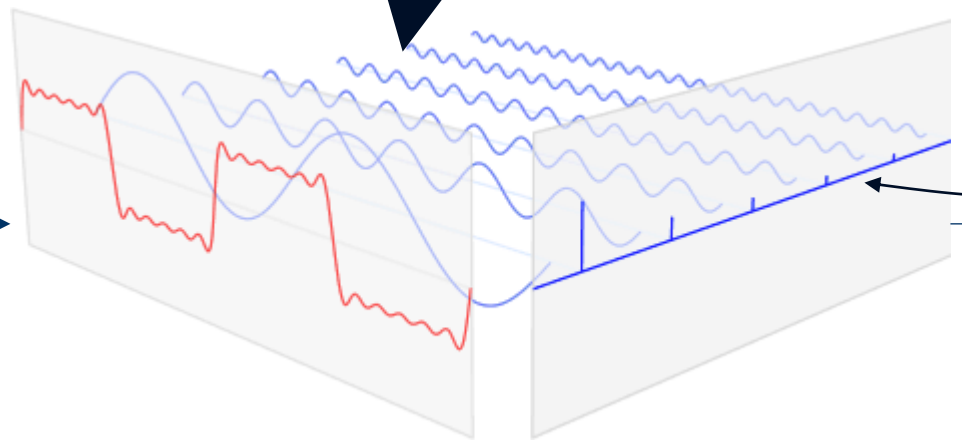
Frequency Domain

Sound source: https://freesound.org



$$X[k] = \sum_{n=0}^{N-1} x[n]\, exp^{-i2\pi\frac{k}{N}n}$$





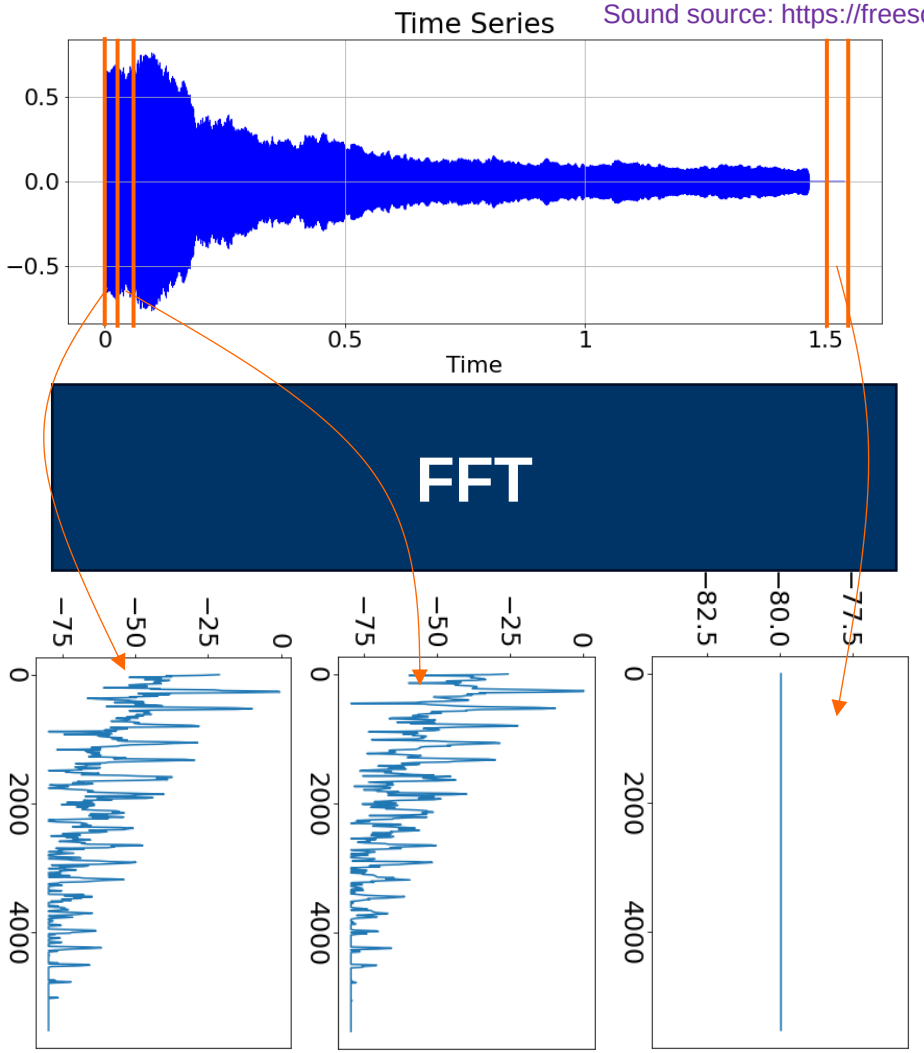
Discrete Fourier Transform

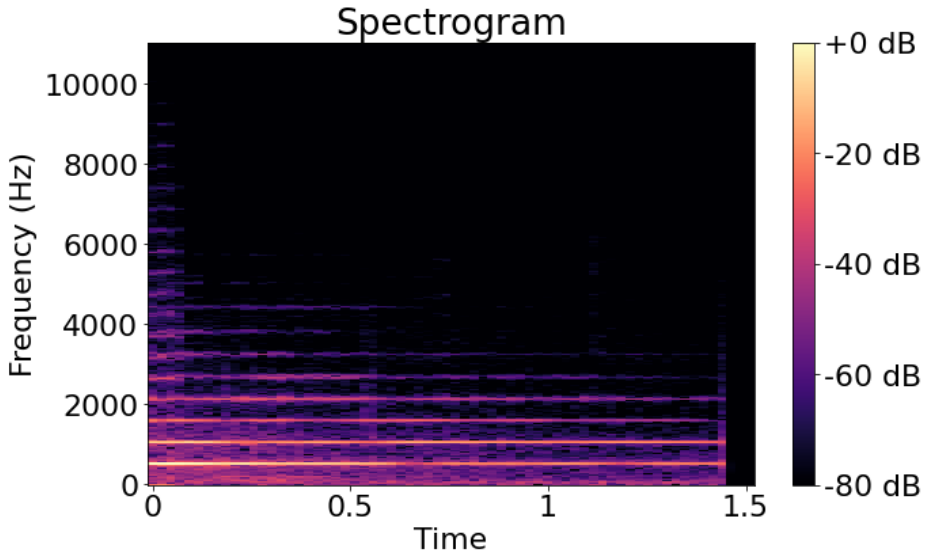
***Calculated via Fast Fourier Transform (FFT)**

Trivia: DFT/FFT helps us to find which frequencies exist. But at what time instance do they occur?

https://www.ritchievink.com/blog/2017/04/23/understanding-the-fourier-transform-by-example/

Short Time Fourier Transform - Spectrograms





$$S(k,m) = 20\,log10\left(\left|\sum_{n=0}^{N_w-1} x[n]\,w[m-n]exp^{-i2\pi\frac{k}{N}n}\right|\right)^2$$

1. **Divide the total time into small time windows**
2. **Find FFT for each window (after multiplying with a short decaying windows (e.g. Gaussian)**
3. **Stack all the FFTs (actually FFT power) to plot an image**
4. **Take log of FFTs to show small changes (result will be dB)**

Mel - Spectrograms

Humans perceive sound in log frequency scale and not linear frequency scale

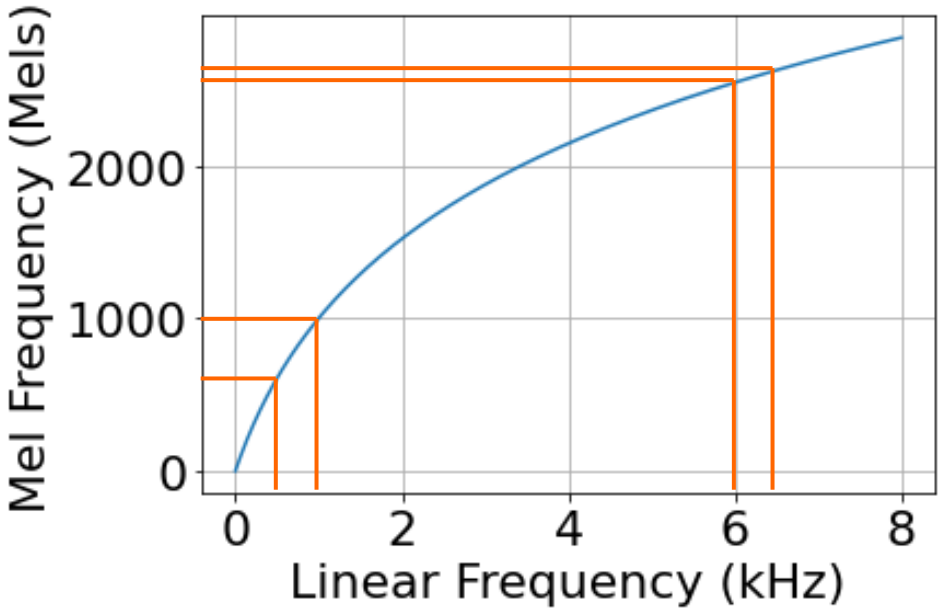
The mel-scale is "a perceptual scale of pitches judged by listeners to be equal in distance from one another"

Linear Scale Change from 100 Hz to 200 Hz

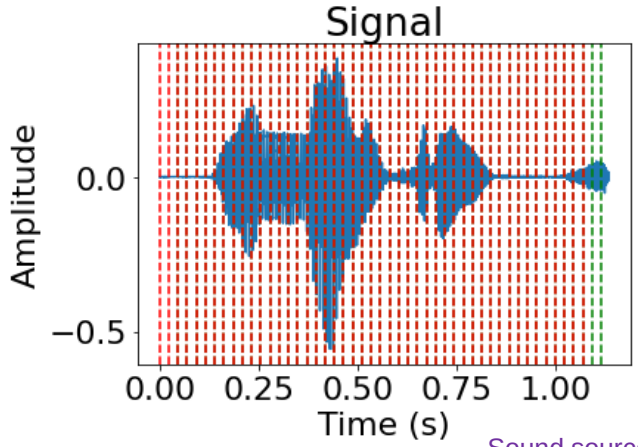
Mel Scale Change from 100 Hz to 200 Hz

$$mel_f = 2595\, log_{10}\left(1 + \frac{lin_f}{700}\right)$$

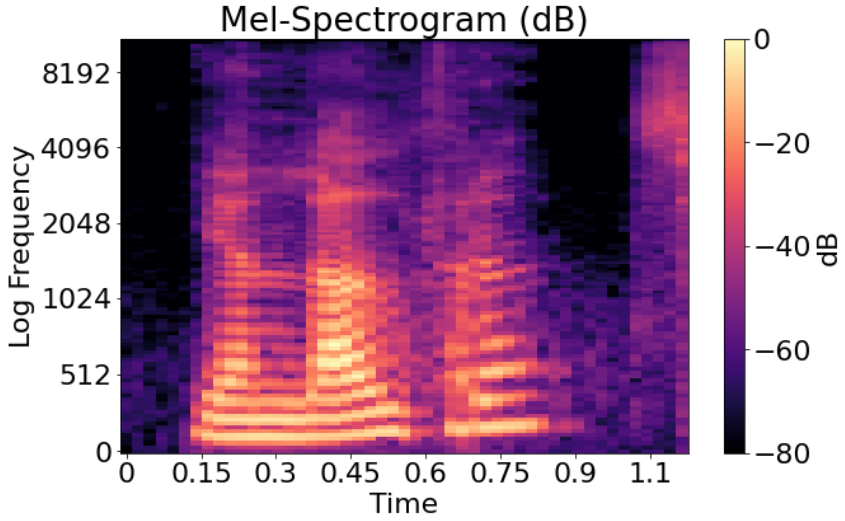
(O'Shaughnessy 1987)

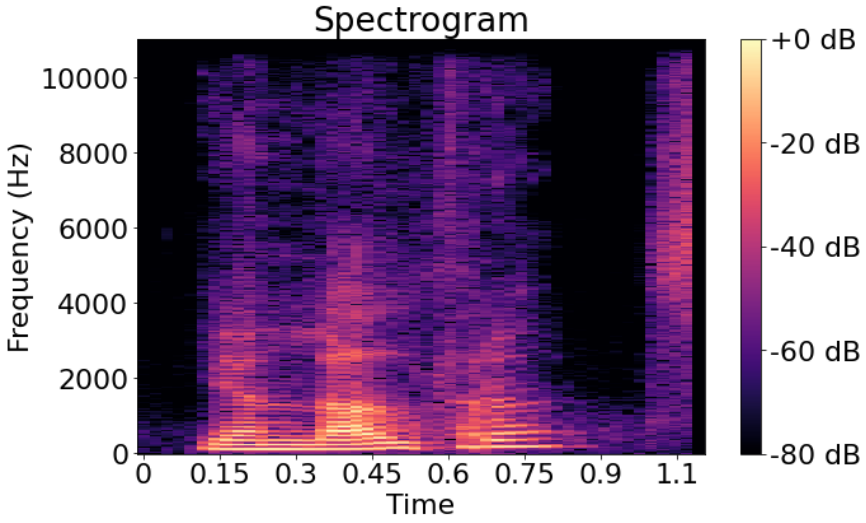


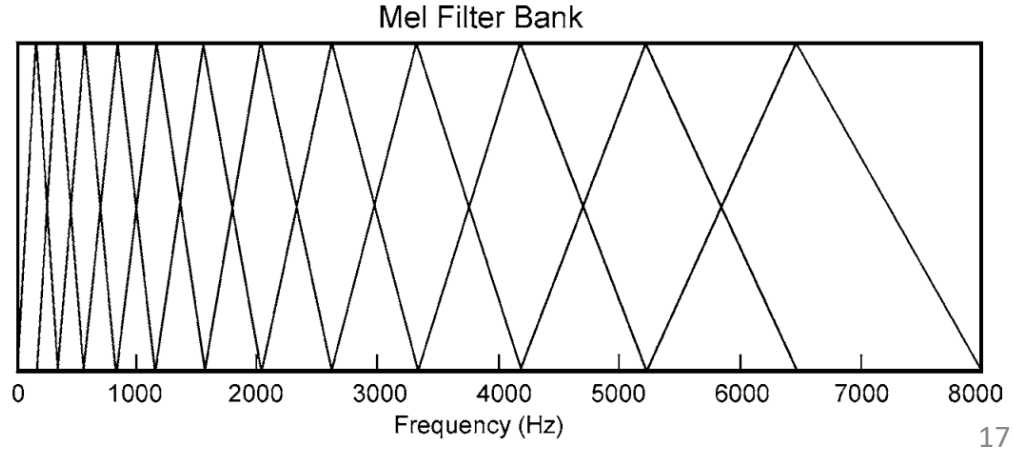
Mel - Spectrograms



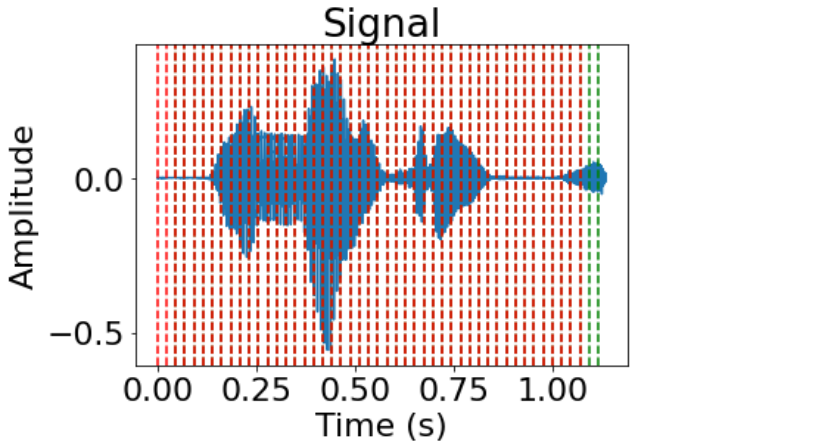
Sound source: https://freesound.org





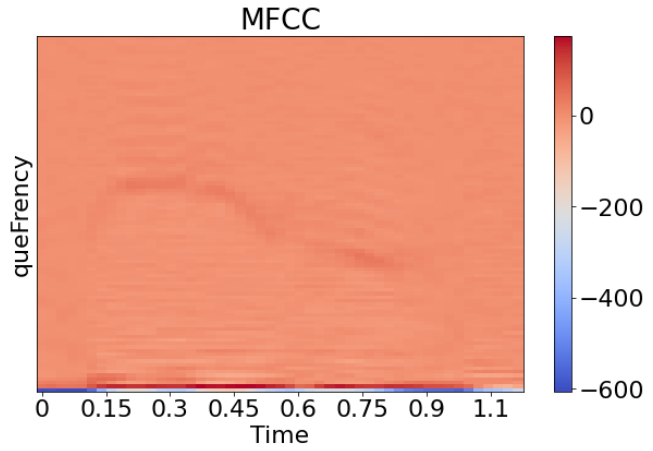


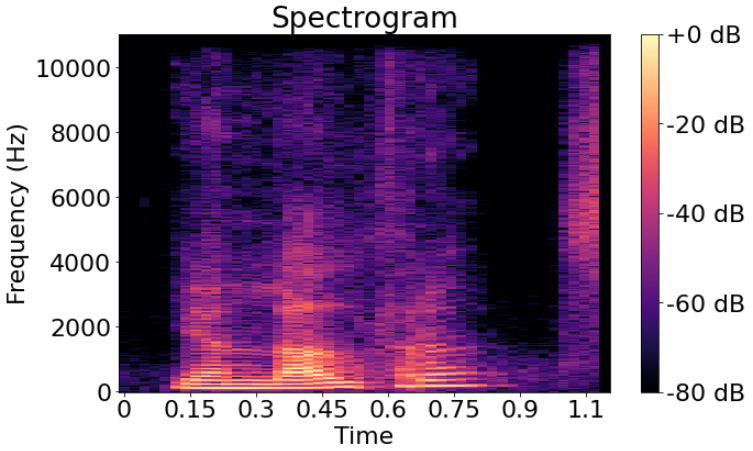
Mel Frequency Cepstral Coefficients (MFCC)

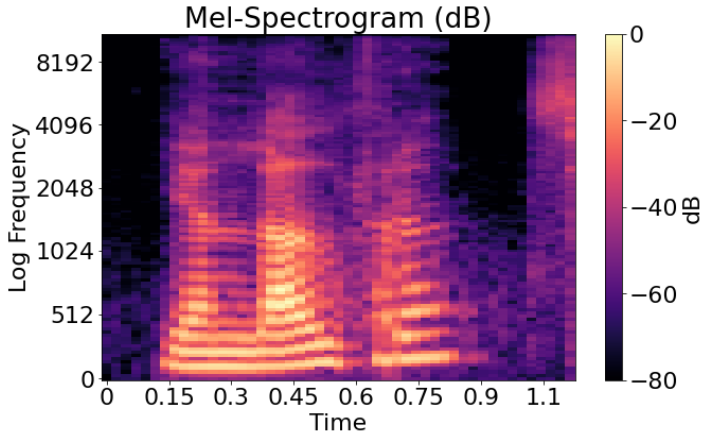


Sound source: https://freesound.org

Discrete Cosine Transform







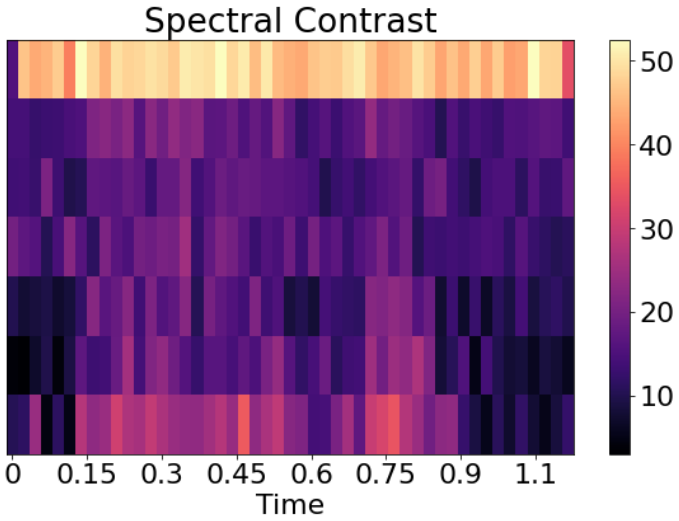
- A spectrum of spectrum is termed as “Cepstrum”
- MFCC is in time-domain and is much less decluttered and hence useful for Machine Learning tasks
- MFCC is considered as a representation of timbre *(perceived sound quality)*

Reference for an intuitive in-depth illustration of MFCC: https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779

Frequency Domain Features

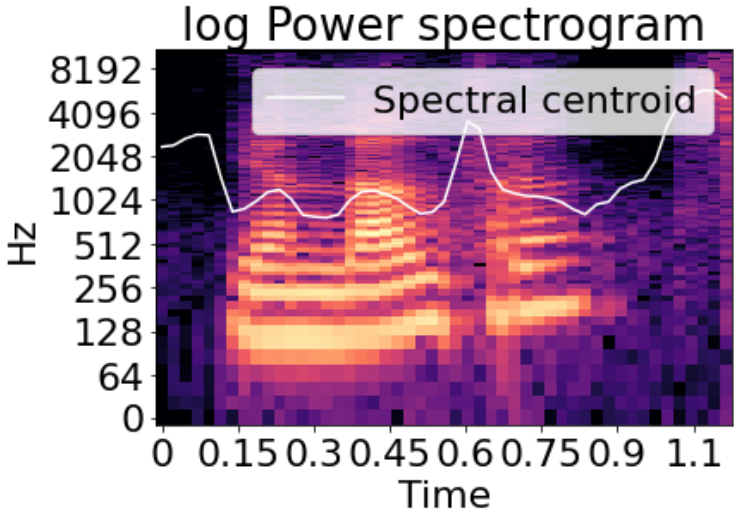
Spectral Contrast, $BER[n] = \frac{\sum_{k=0}^{F-1} S_n(n)^2}{\sum_{k=F}^{N-1} S_n(n)^2}$

- Comparison of energy in the lower/higher frequency bands
- Music/speech discrimination

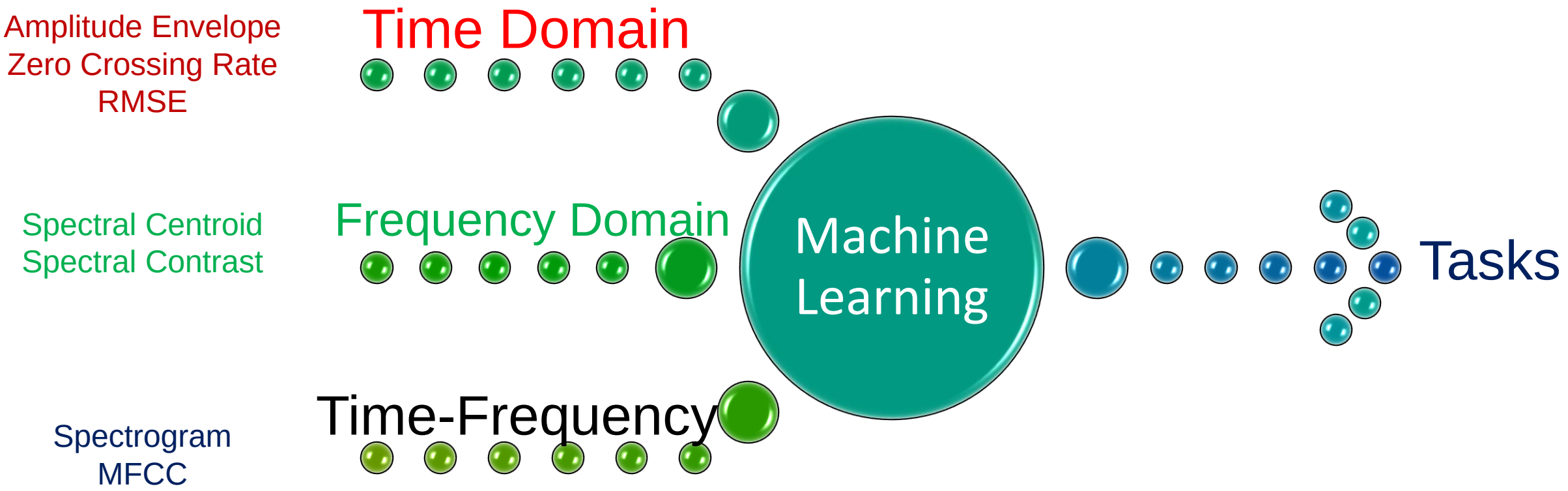


Spectral Centroid, $SC_n = \frac{\sum_{k=0}^{N-1} S_n(n)\, n}{\sum_{k=0}^{N-1} S_n(n)}$

- Centre of gravity of magnitude spectrum
- Frequency band where most of the energy is concentrated
- Measure of brightness of sound!
- Classification

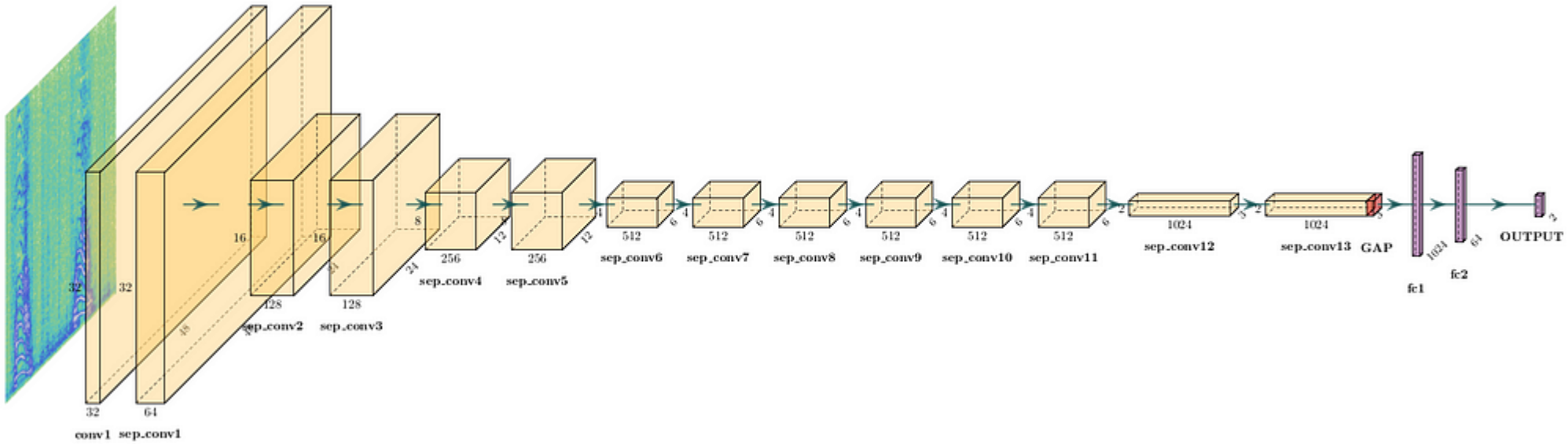


Low Cost (Compute) Audio Analytics



State-of-the art

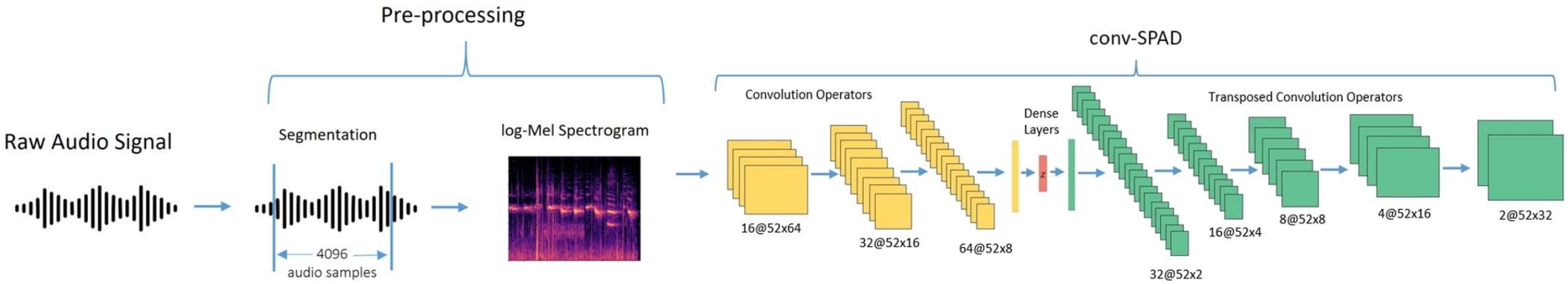
YAMNet: A deep convolutional neural network that predicts 521 audio event classes from the AudioSet-YouTube corpus it was trained on. The architecture is based on the light weight MobileNet v1.



https://www.tensorflow.org/hub/tutorials/yamnet

State-of-the art

conv-SPAD: A simple custom modelled convolutional SPectral audio-based anomaly detection.



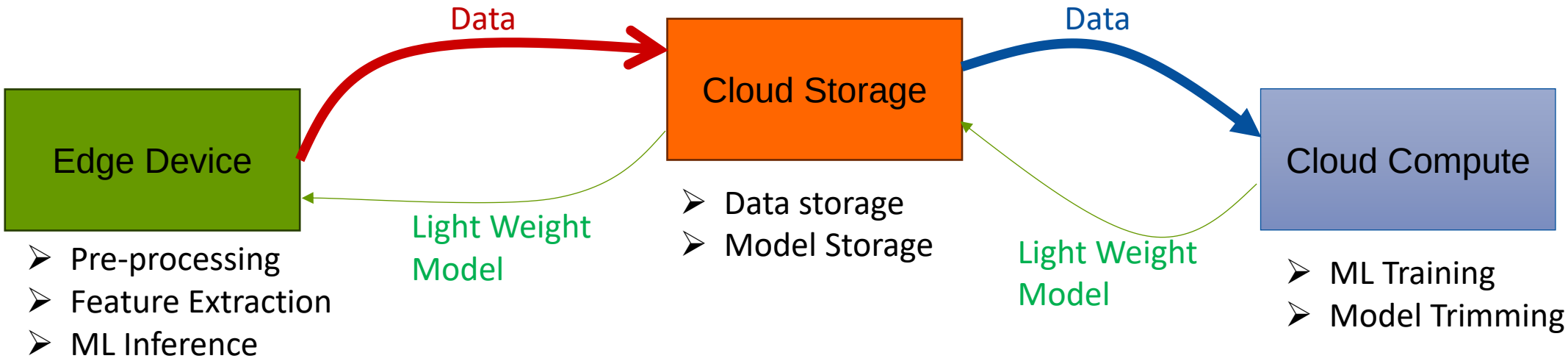
Lo Scudo, Fabrizio, Ettore Ritacco, Luciano Caroprese, and Giuseppe Manco. "Audio-based anomaly detection on edge devices via self-supervision and spectral analysis." Journal of Intelligent Information Systems (2023): 1-29.

Build Your Own Model

- *Pre-processing*
  - *Normalization, Segmentation*
- *Feature extraction*
  - *Spectrogram, MFCC*
- *Model selection*
  - *CNNs, RNNs, Transformers*
- *Transfer learning*
  - *Leverage existing models*
- *Useful datasets*
  - *TAU Urban Acoustic Scenes, TUT Rare sound events, LibriSpeech and Audiset*

Lo Scudo, Fabrizio, Ettore Ritacco, Luciano Caroprese, and Giuseppe Manco. "Audio-based anomaly detection on edge devices via self-supervision and spectral analysis." Journal of Intelligent Information Systems (2023): 1-29.

Current Trend – Cloud to Edge for Scalability



- *Edge Impulse*

- *Speech Processing with AI*
  - *Murf.ai,*
  - *AWS Transcribe and Polly,*
  - *Google Cloud Text-to-Speech,*
  - *Azure Text to Speech API,*
  - *IBM Watson Text to Speech*

Summary

- *Audio analytics on edge helps in faster response, data security and enhanced analysis*
- *Pre-processing of audio is important*
- *Selection of right input features and model is important*
- *Build in the cloud and adapt to edge for scalability*

Demo of audio analytics @ https://edgeaudioanalytics.streamlit.app/

Contact: mahesh.panicker@singaporetech.edu.sg