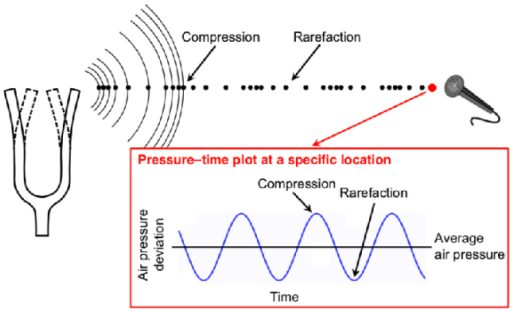
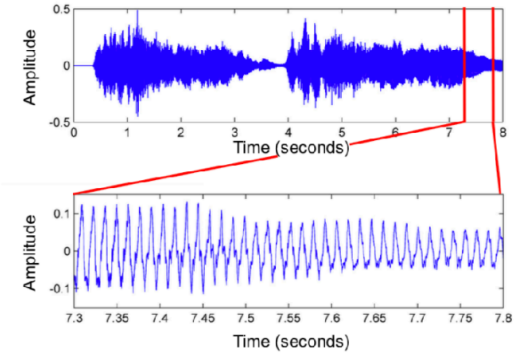
Lecture8 Audio Understanding

1. Basic of Audio and Sound

Audio Signal / Waveform

Change in air pressure at a specific point in space over time registered by a microphone.

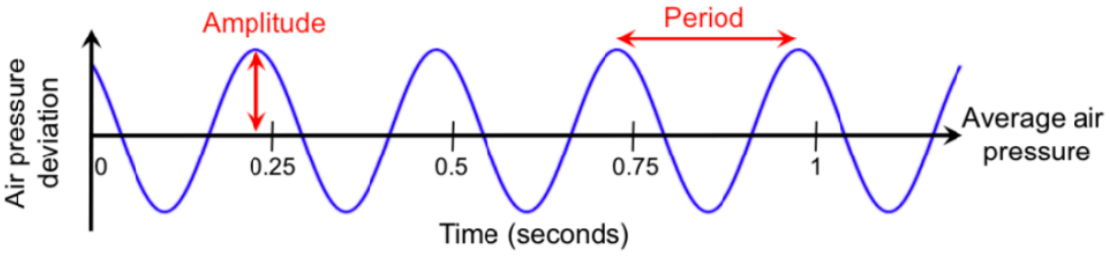




Frequency and Pitch

- A signal is **periodic** if it repeats
- Formally, a signal $x[n]$ is periodic if there is a such that $x[n] = x[n+p]$ for all n. p is the period of the signal, typically measured in seconds
- **Frequency** is the reciprocal (倒数) of the **period**, $f = \frac{1}{p}$, typically measured in Hertz (Hz) (cycles per second)

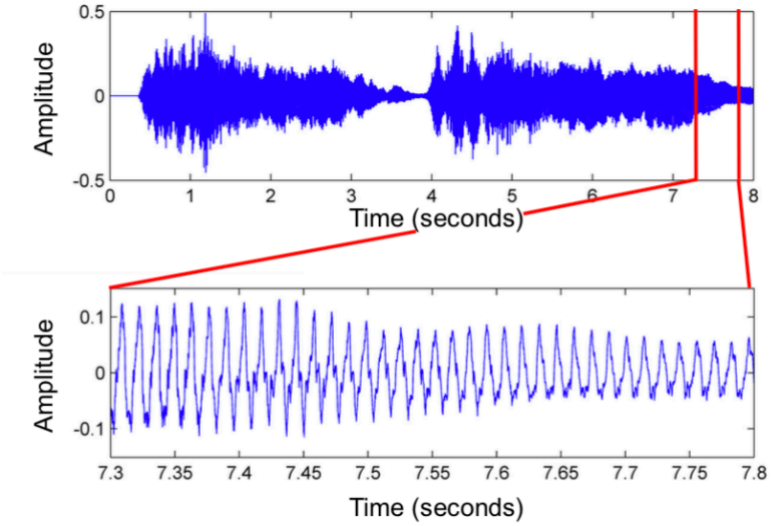
Elements



A sinusoid is the simplest type of periodic waveform, fully described by its:

- **frequency** f (number of cycles per second)
- **amplitude** a (peak deviation from the mean)
- **phase** ϕ (where in the cycle is the sinusoid at time zero)

Music Frequency



- “Real” sounds are more complex than pure tones
- Musical tones can be modeled as a combination of different pure tones (**partials**), each with different frequencies, amplitudes, and phases
- The frequency of the lowest partial is the **fundamental frequency (f0)**, which often determines the perceived pitch
- The frequencies of the other partials which are an integer number of the f0 are called **harmonics**

Real World Frequency

- The larger the **frequency** of a sinusoidal waveform, the “**higher**” it sounds
- Human hearing range: 20 Hz — 20 kHz
  - Orchestra tuning frequency: 440 Hz
- **Frequency** is closely related to **pitch**, which is subjective
  - For pure tones, we can consider pitch and frequency equal

Pitch

- Pitch is the human perception of the frequency of a sound wave.
- It's a **subjective** quality that can be described as "how high" or "how low" a sound seems to the human ear.

Loudness and Decibel

- **Loudness**: a perceived measure of intensity of a sound, correlated with the objective measures of **sound intensity** and **sound power**

Decibel

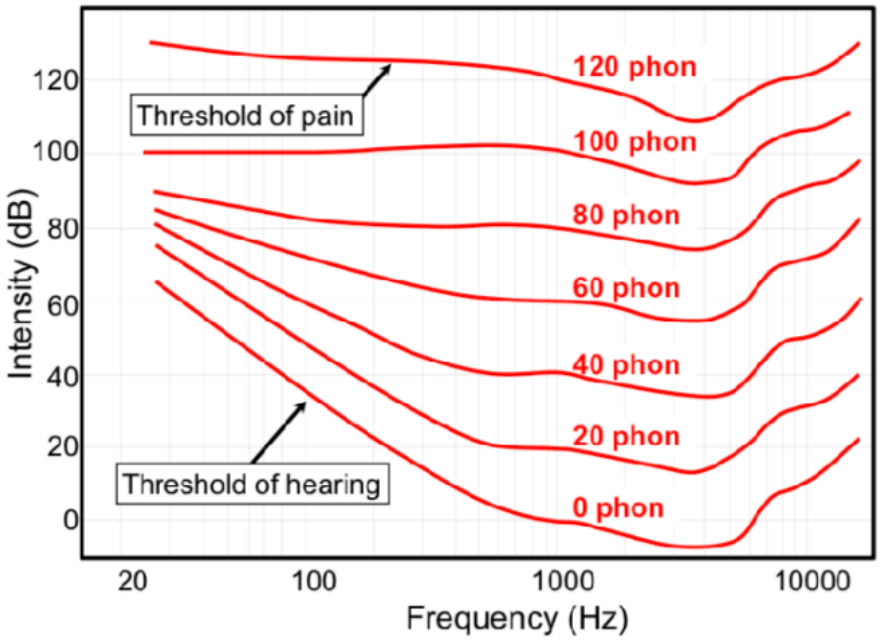
- **Decibel** (dB): a scale, measuring the **logarithmic** ratio between a **sound’s intensity** and the **threshold of hearing (the most silent sound human can notice)**

Source	Intensity	Intensity level	× TOH
Threshold of hearing (TOH)	10^{-12}	0 dB	1
Whisper	10^{-10}	20 dB	10^{2}
Pianissimo	10^{-8}	40 dB	10^{4}
Normal conversation	10^{-6}	60 dB	10^{6}
Fortissimo	10^{-2}	100 dB	10^{10}
Threshold of pain	10	130 dB	10^{13}
Jet take-off	10^{2}	140 dB	10^{14}
Instant perforation of eardrum	10^{4}	160 dB	10^{16}

Midtown traffic noise is around 70 to 85 dB's on any given day

- **Intensity**: A physical measure of the power per unit area, expressed in watts per square meter.
- **Intensity level**: The intensity of the sound converted into decibels (dB). This is a logarithmic scale used to describe the intensity level in a way that is more manageable for human perception.
- **× TOH**: This column shows how many times the intensity of each sound source is greater than the threshold of hearing. The threshold of hearing is typically taken as 10^{-12} watts per square meter and is equivalent to 0 dB.

Phon Level (Loudness)



- The 'phon' is a unit that is used to describe **loudness** levels. Equal loudness contours show curves with a **constant perceived loudness** (in units of "phons")
- The **perceived** quantity, **loudness**, depends on both **intensity** and **frequency**
- Our ears are most sensitive between **2-4kHz**, which is where the curves dip the lowest. This means it takes **less intensity** (fewer decibels) for sounds in this frequency range to be perceived as **loud**.
- The intensity of the threshold of hearing/ pain depends on the frequency

Timbre (音色)

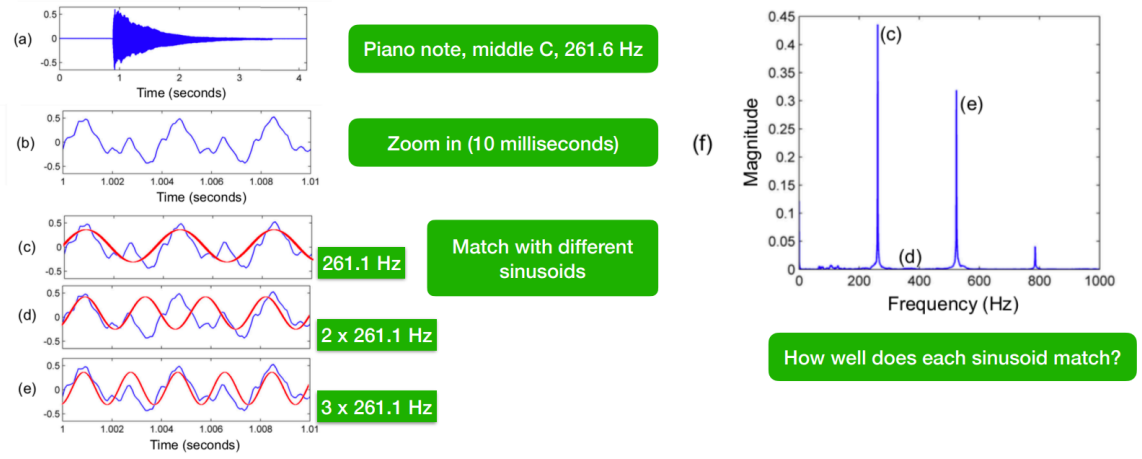
- The perceptual property of sound that allows us to **distinguish between a instruments** if they played the same note at the same volume
- A perceived quality of sound, related to
  - The evolution of the sound over time
  - The distribution of energy across partials
  - The relative amount of noise and sinusoidal components

Fourier transform

Waveform not the most informative to look at. There are other tools that make the spectral content of the audio more **explicit**.

Which frequencies are present in an audio signal?

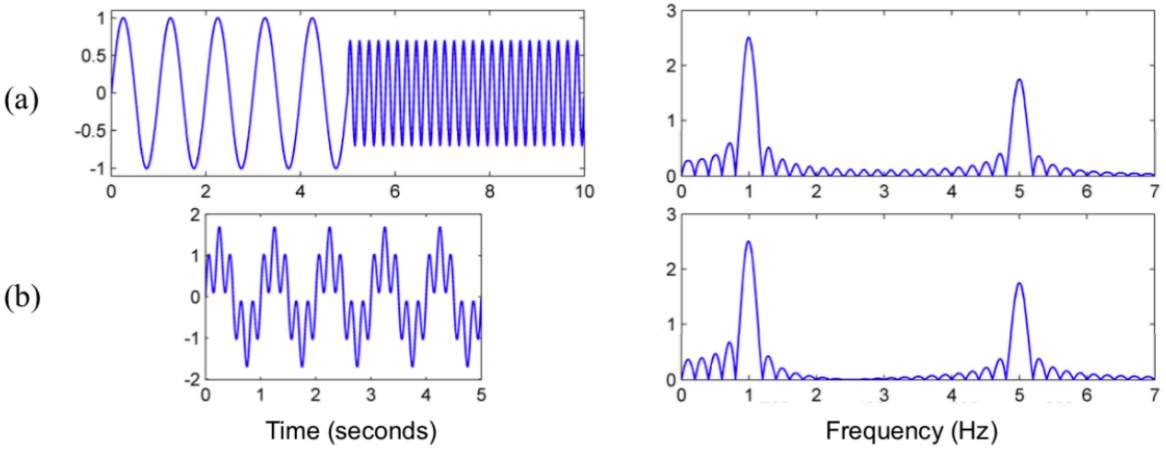
1. Get a set of “sinusoids” with the frequencies of interest
2. For each “sinusoid”, compare to original signal: how well do they match?
3. You can recover the signal by adding up those sinusoids



- (c) matches the best

Discrete Fourier transformation (DFT)

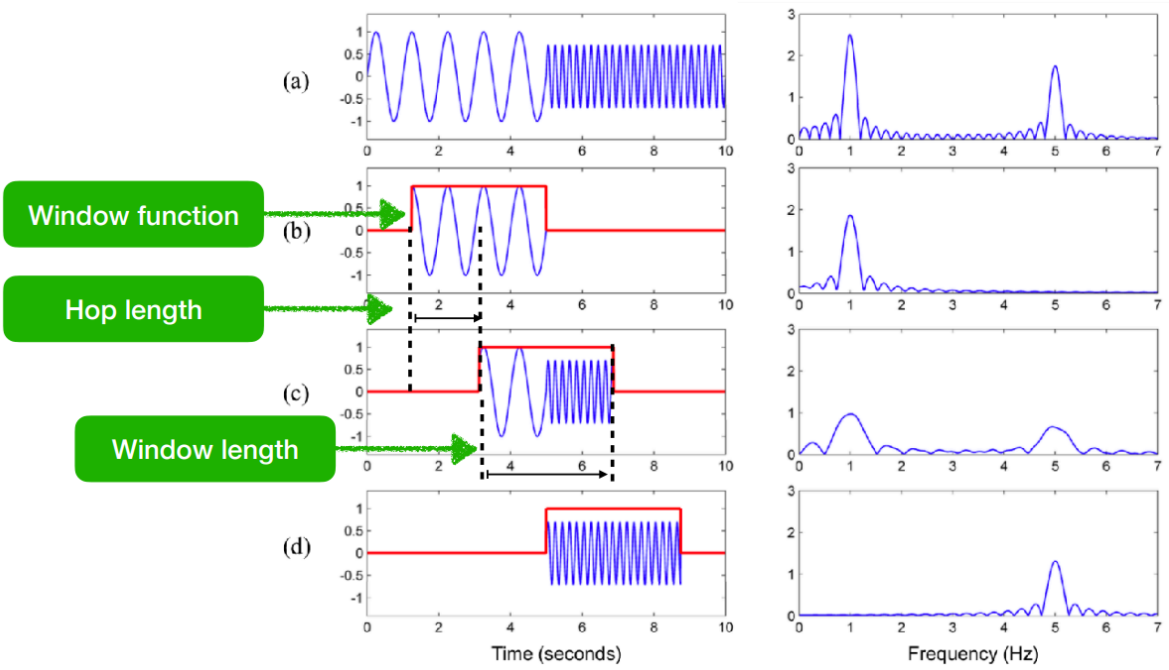
The DFT (discrete Fourier transformation) shows the overall frequency composition of a sound, but we lose information about changes over time



- **Time Domain Signals** (left)
  - Shows a simple periodic signal, likely a pure sine wave, followed by a segment of higher frequency oscillations. This could represent a sound that changes pitch over time
- **Frequency Domain Representations** (right)
  - The corresponding frequency domain graph, as analyzed by the DFT.
  - These graphs show how much of each frequency is present in the original time domain signal. Peaks in the frequency domain represent the frequencies that are most prevalent in the time domain signal.

Short-Time Fourier transform (STFT)

In practice, sound is often analyzed using a **Short-Time Fourier Transform (STFT)**, which divides the signal into shorter segments and applies the Fourier Transform to each one.

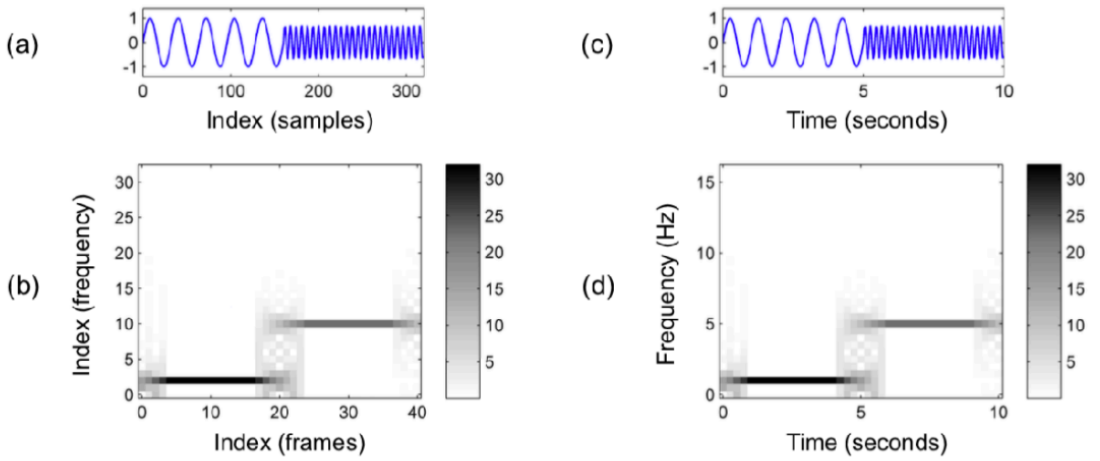


- H: hop length
- N: window length

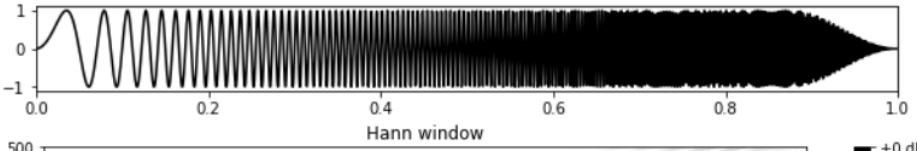
Spectrogram (声谱图)

The magnitude $|X[m, k]|$ of the STFT is called a **spectrogram**.

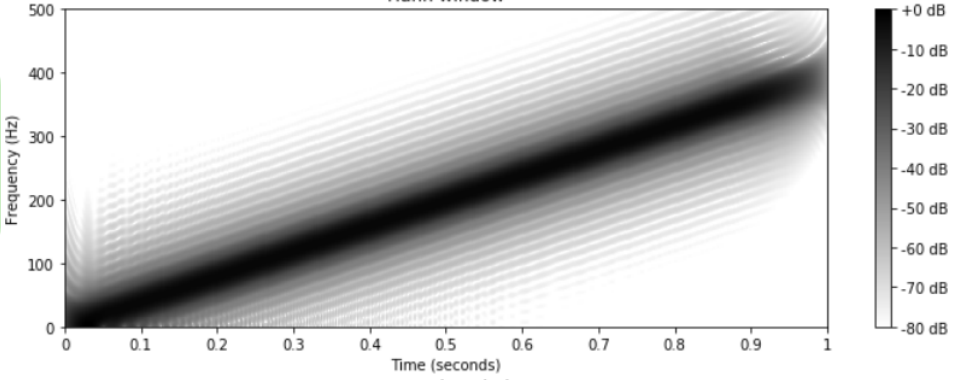
We can plot a spectrogram like an image.



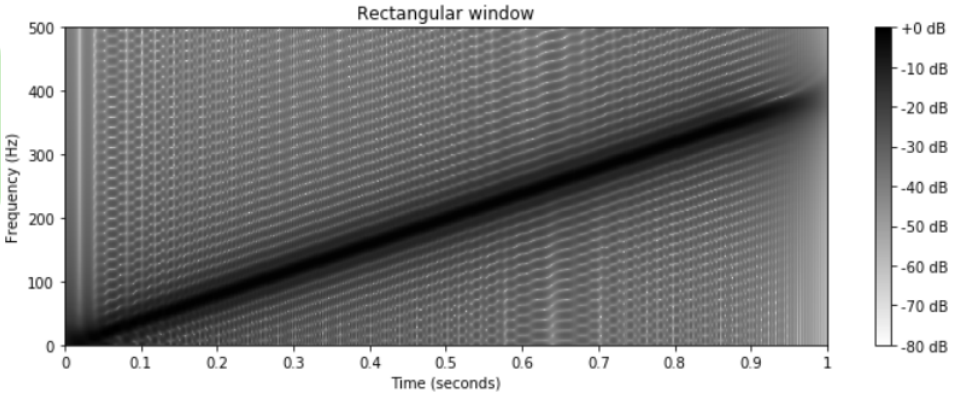
Sweep



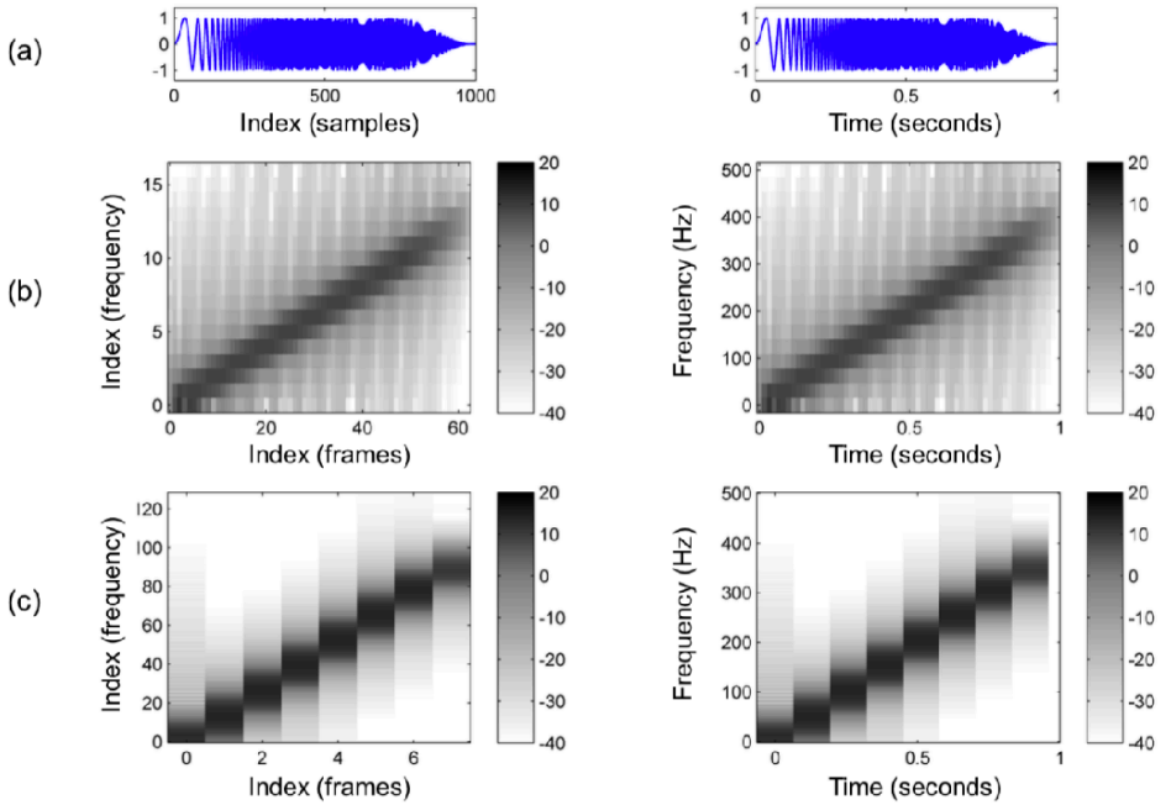
Spectrogram with Hann window



Spectrogram with rectangular window

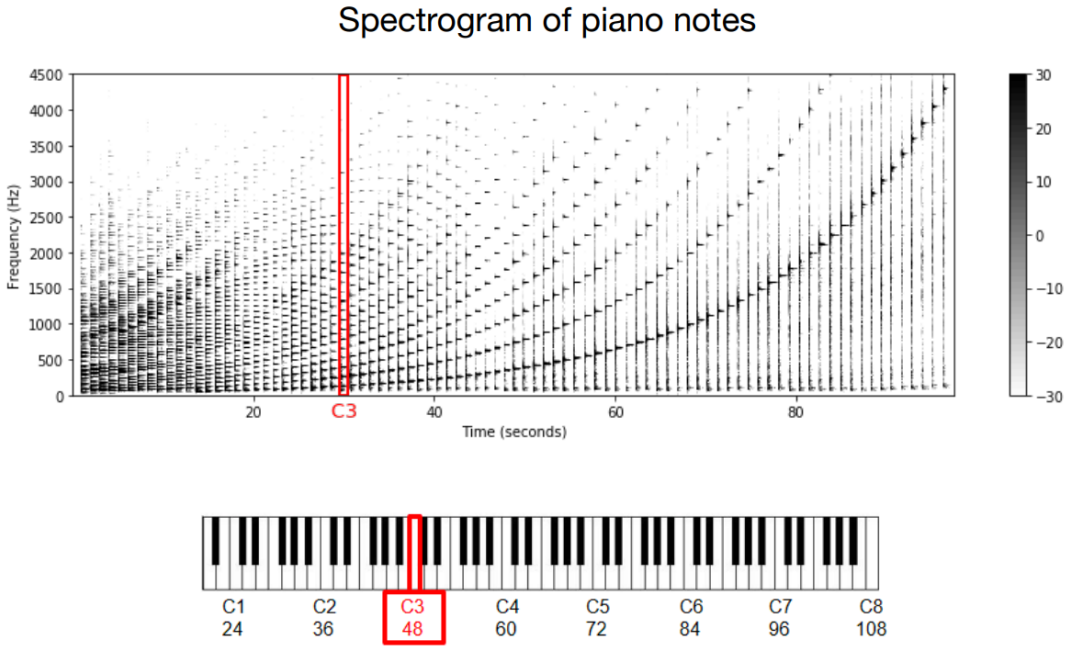


Time vs frequency resolution tradeoff

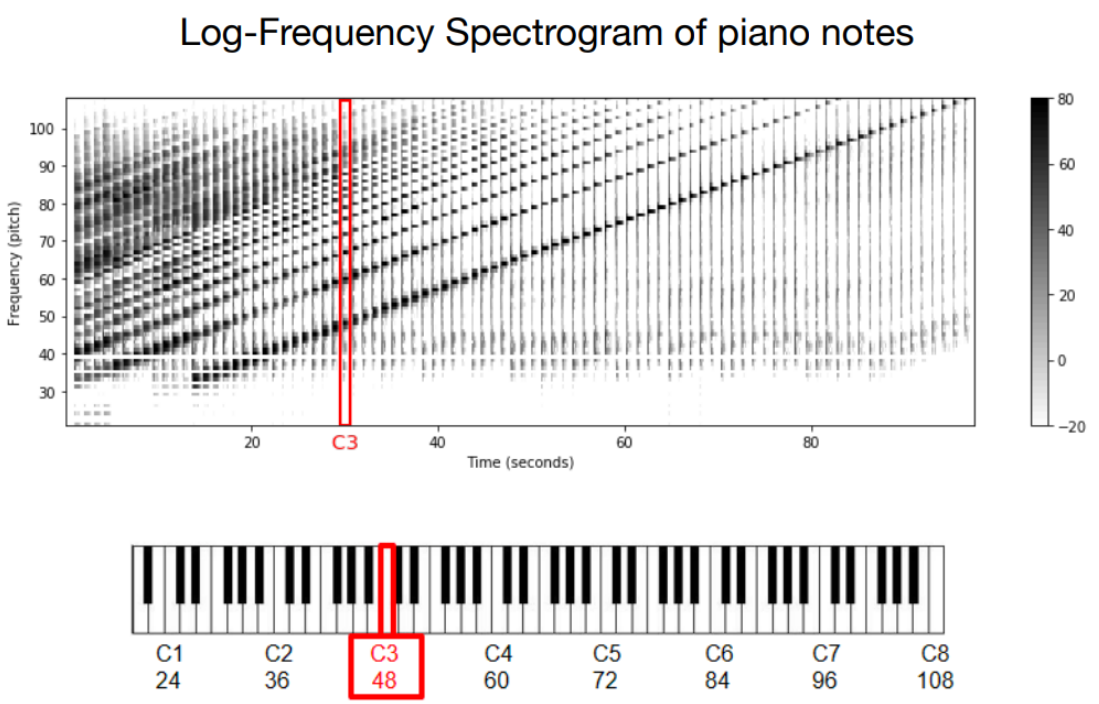


- (b) small N: poor frequency resolution, frequencies localized in time
- (c) large N: good frequency resolution, frequencies smeared in time
- hop size H: smaller hop sizes give better time resolution, but more computation time

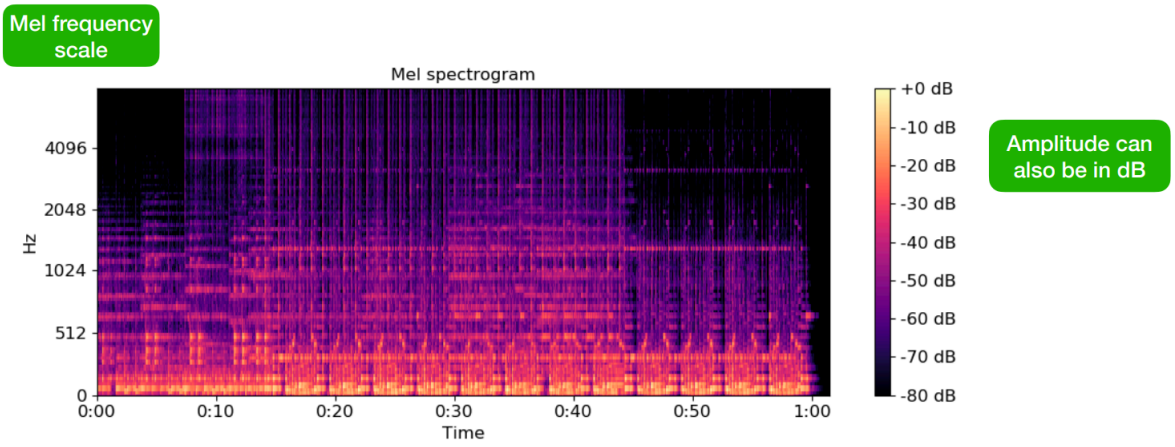
Linear-frequency Spectrogram



Log-Frequency Spectrogram

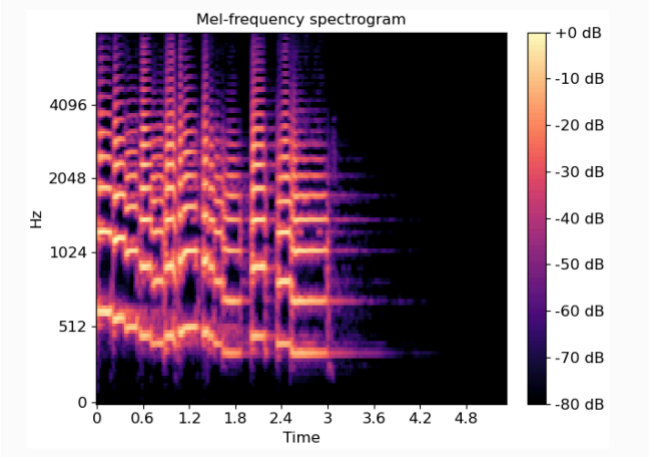


Mel-Spectrogram



Audio signal processing tool (Python)

Librosa



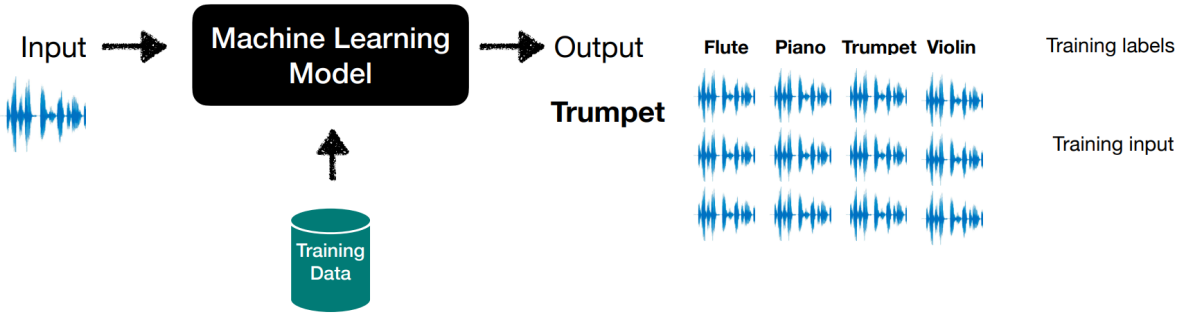

```
>>> y, sr = librosa.load(librosa.ex('trumpet'))
>>> librosa.feature.melspectrogram(y=y, sr=sr)
array([[3.837e-06, 1.451e-06, ..., 8.352e-14, 1.296e-11],
       [2.213e-05, 7.866e-06, ..., 8.532e-14, 1.329e-11],
       ...,
       [1.115e-05, 5.192e-06, ..., 3.675e-08, 2.470e-08],
       [6.473e-07, 4.402e-07, ..., 1.794e-08, 2.908e-08]],
      dtype=float32)
```

Audio Datasets Soun[D]ata

Supported Datasets and Annotations — mirdata 0.3.8 documentation

Supported Datasets and Annotations — soundata 0.1.3 documentation

2. Audio Classification

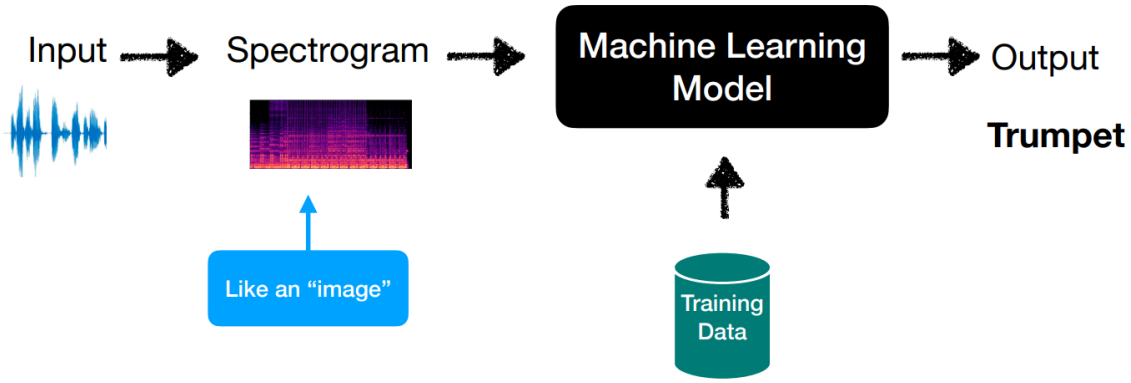


Application

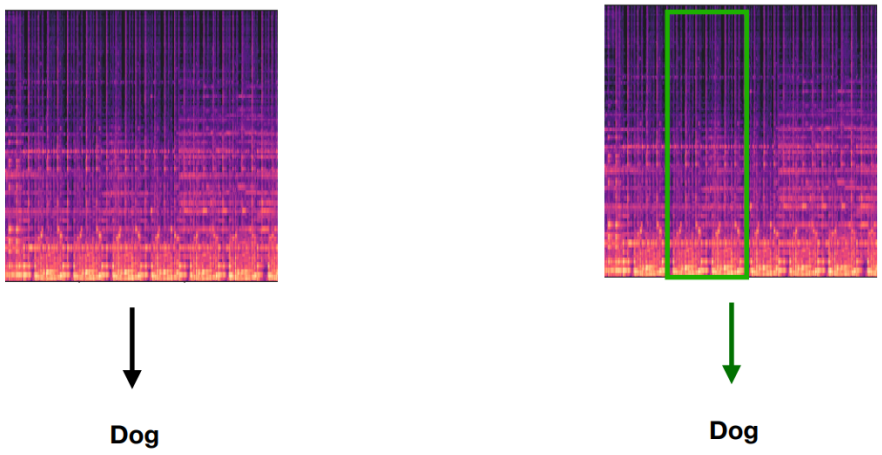
- **Instrument Identification**
  - Input: recordings of solo instruments, Output: instrument label
  - Input: recordings of songs, Output: labels of active instruments over time
- **Music vs. Speech**
  - Input: audio recording, Output: music or speech label
- **Chord recognition**
  - Input: music recording, Output: chord labels over time
- **Drum Transcription**
  - Input: music recording, Output: active drum instruments over time
- **Music Captioning**
  - Input: music recording, Output: description of the given music. (Ref: “LP-MusicCaps”)
- **Sound tagging**
  - Input: recordings of everyday sounds, Output: sound label (e.g. “dog”)
- **Sound event detection**
  - Input: audio recording, Output: sound label and timestamp (e.g. “car, 1.24s, 2.03s”)
- **Sound event detection and localization**
  - Input: multi-channel audio recording, Output: sound label, timestamp, azimuth and elevation (e.g. “car, 1.24s, 2.03s, 30°, 15°”)
- **Acoustic scene classification**
  - Input: audio recording, Output: scene label (“e.g. shopping mall”)

Tasks

Classification



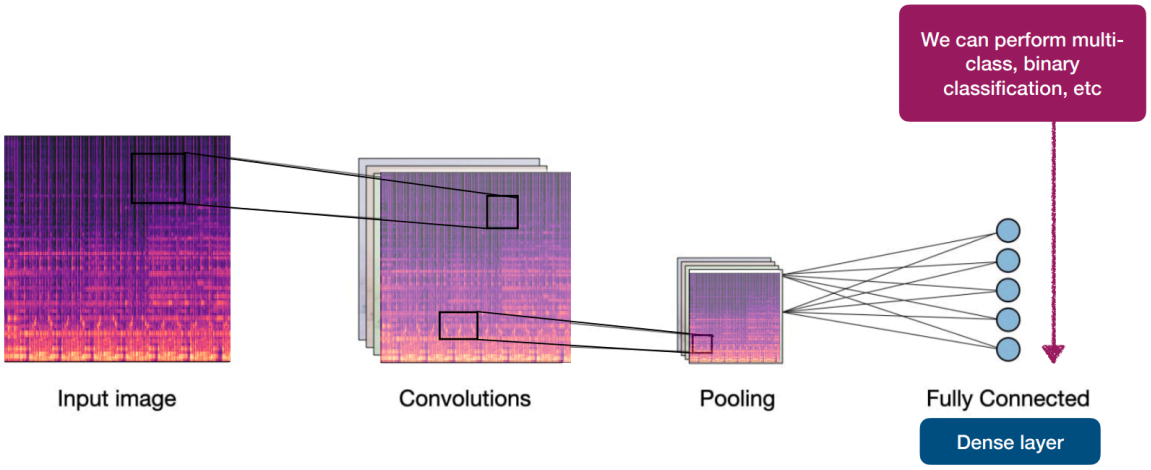
Classification vs. Detection



Single label for the entire clip vs. Label for a given time window

Convolutional Neural Networks in audio

> [1607.02444] Explaining Deep Convolutional Neural Networks on Music Classification (arxiv.org)

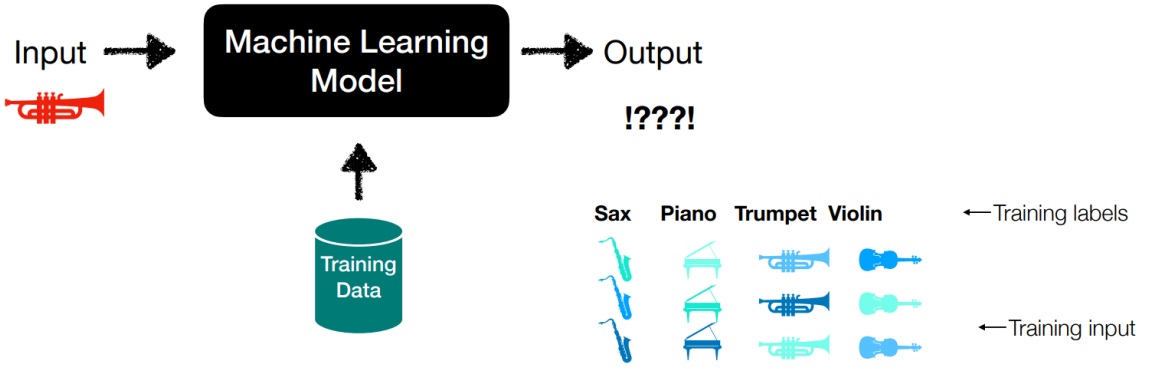


- Idea
  - learn hierarchies of patterns (from edges to concepts) in an effective manner
  - Texture of spectrogram is closely related to timbre of sound and music
- CNNs expect fixed shape input, what do we do?
  - Force fixed size input spectrograms / signals (e.g. padding / truncating)
  - Split the signal into small windows of fixed size, e.g. 1s (add padding to the last one)

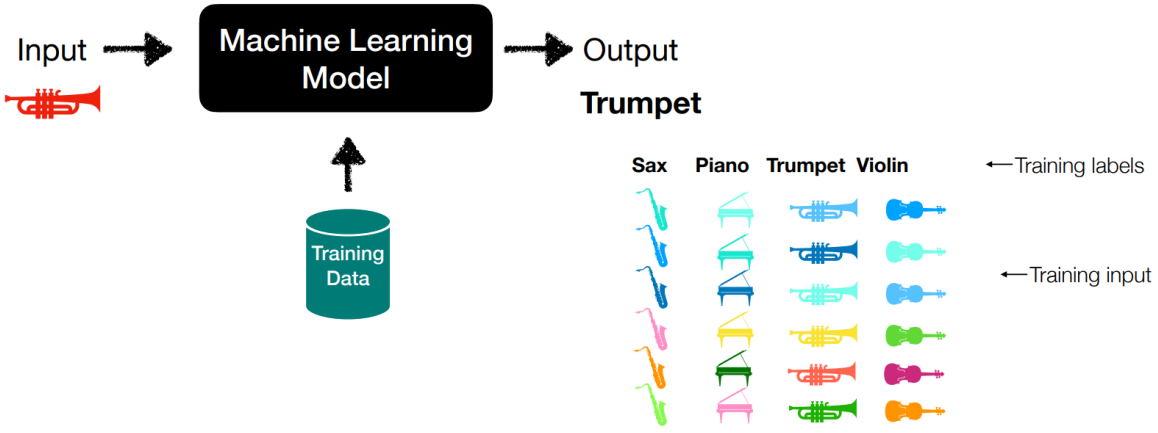
3. Data Augmentation

Dataset Problem

What happens when you don't have a training dataset for the data you want to apply a model to?

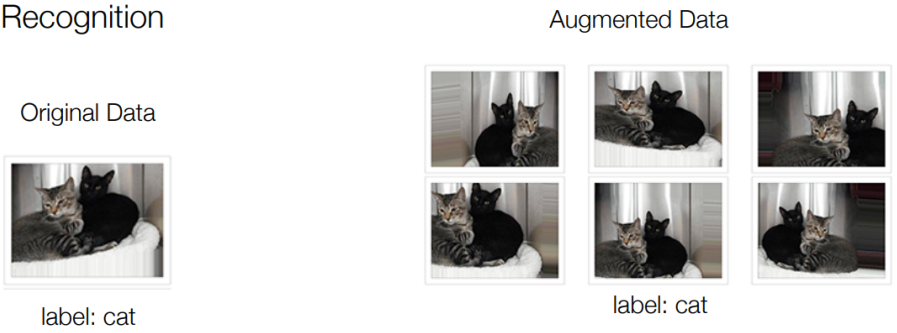


- Training datasets aren't always representative of what you want to model
- **Available training dataset**: Medley-Solos-DB (which contains standard, good quality music recording)
- **Example application**: Classify the instrument in recordings from Freedsound (which is in a noisy and messy environment)
- **One solution**: Add examples that look more like your test data using Data Augmentation



Introduction

Example: Image Recognition



- **Data Augmentation** is the process of increasing the size of an existing dataset by adding modified examples of the original data.

Original Data

label: cat

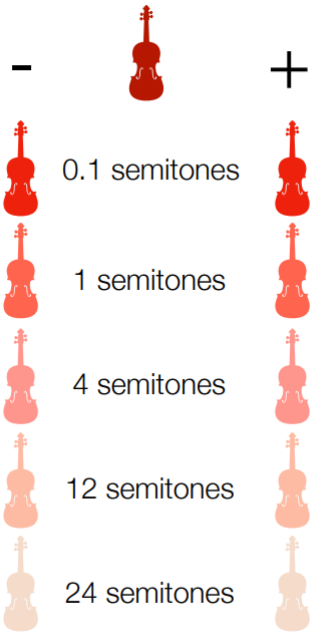
Invalid Augmentation

label: ~~cat~~ !?!?

- The transformations should be **valid** for the task!
- If you change the input, you (sometimes) have to **change the label** to match.
  - Augmentations that **don't change the label** are called **label-preserving** augmentation
    - E.g. adding a small amount of background noise to a solo instrument recording doesn't change the instrument label
  - Other augmentations **deform** the labels
    - E.g. time stretching a song changes the positions of the beats.

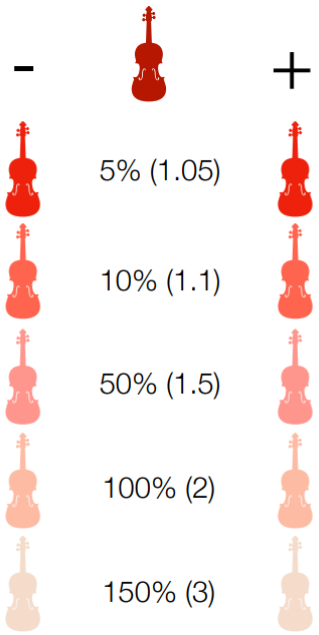
Musical data augmentation

Pitch shifting



- Change the pitch of the audio without changing the speed
  - Label preserving for instrument identification
    - Unless the transformation is extreme
  - Deforms labels for pitch-related data
    - e.g. For chords, the labels should be shifted along with the audio!

Time stretching



- Change the speed without changing the pitch
  - Label preserving for instrument identification*
  - Duration changes - for a time stretching factor of TS:
    - $TS > 1$: speed up, $TS < 1$: slow down
    - $d_{new} = \frac{d_{old}}{TS}$
- Deforms labels for time-related labels
  - e.g. For chords, the labels should be shifted along with the audio!

Resampling (重采样)

- Change the pitch and speed at the same time
  - conceptually equivalent to playing a record at a faster or slower speed
- Resampling by a factor of f:
  - $f > 1$: speed up and increase pitch
  - $f < 1$: slow down and decrease pitch
  - duration changes to $\frac{d_{old}}{f}$
  - shift pitch by $12log_2(f)$
- Labels need to be deformed in both time and pitch

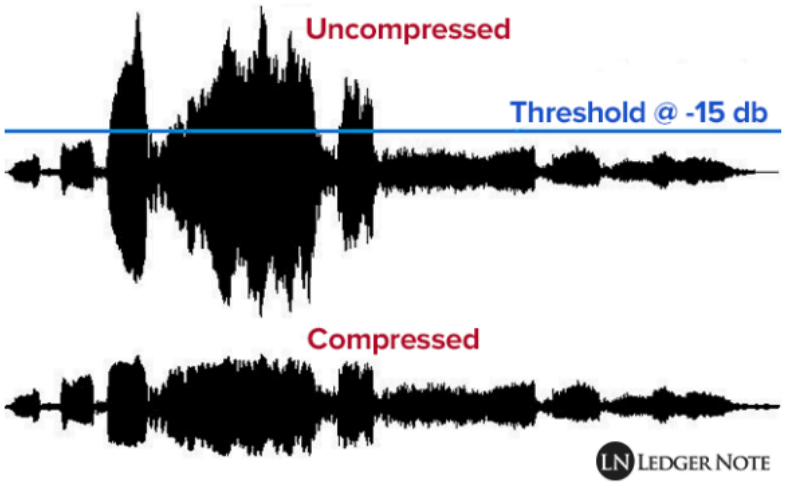
Reverb (混响)

- Add reverb to a recording
- E.g. original recording might be in a studio, test set might be in a concert hall.
- Does not affect the labels

Relative volume adjustment

- If you have individual recordings of different sources (e.g. solo vocals + accompaniment) can create new mixes with different relative volumes
- Doesn't affect labels, unless a source becomes masked by another source

Dynamic range compression



- Non-linear volume adjustment
    - Reduces loud sounds, making the overall volume sound louder
    - Does not affect labels

Some differences

- Recording Conditions
- Recording Quality
- Instrument Characteristics
- Level of musicians
- Musical Style/Genre
- Instrumentation