한국소음진동공학회 AI혁신위원회 주관
2022년 인공지능 강습회

AI 기계이상진단을 위한 인공지능 학습 기법

제 7강 소음 기반 이상진단을 위한 데이터 구성

한국과학기술원 전기및전자공학부

최정우
jwoo@kaist.ac.kr

KAIST EE

목차

- DCASE 2020 Intro & Audio Data Preprocessing
  - Introduction to DCASE 2020 task 2

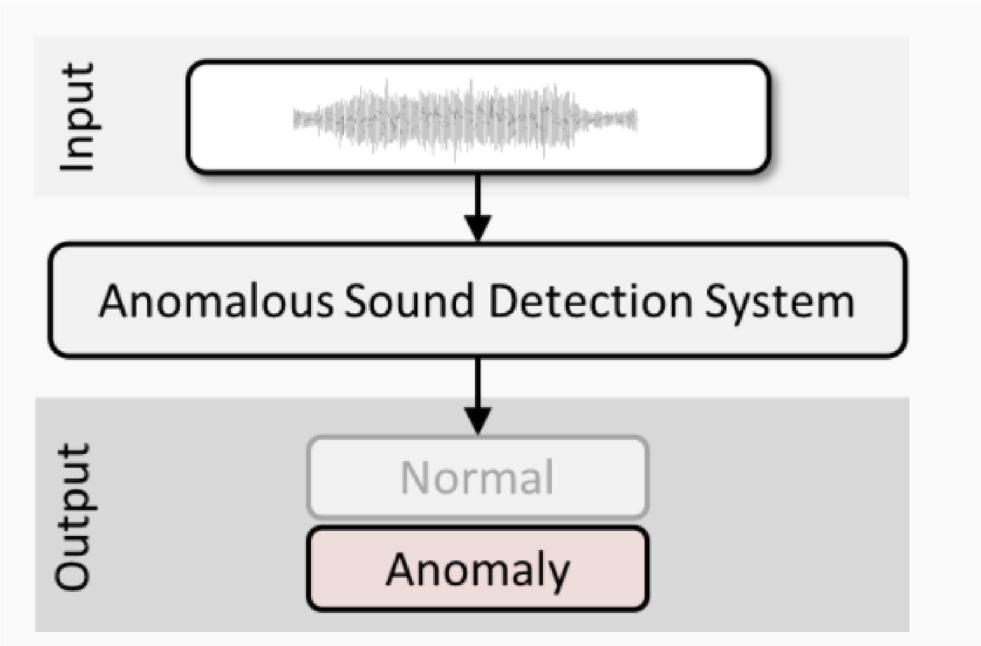
- Audio Data Preprocessing
  - Audio Data Preprocessing (Acoustic feature)
  - Audio Data Preprocessing (Front end system)

DCASE 2020 Dataset & Audio Data Preprocessing

Introduction

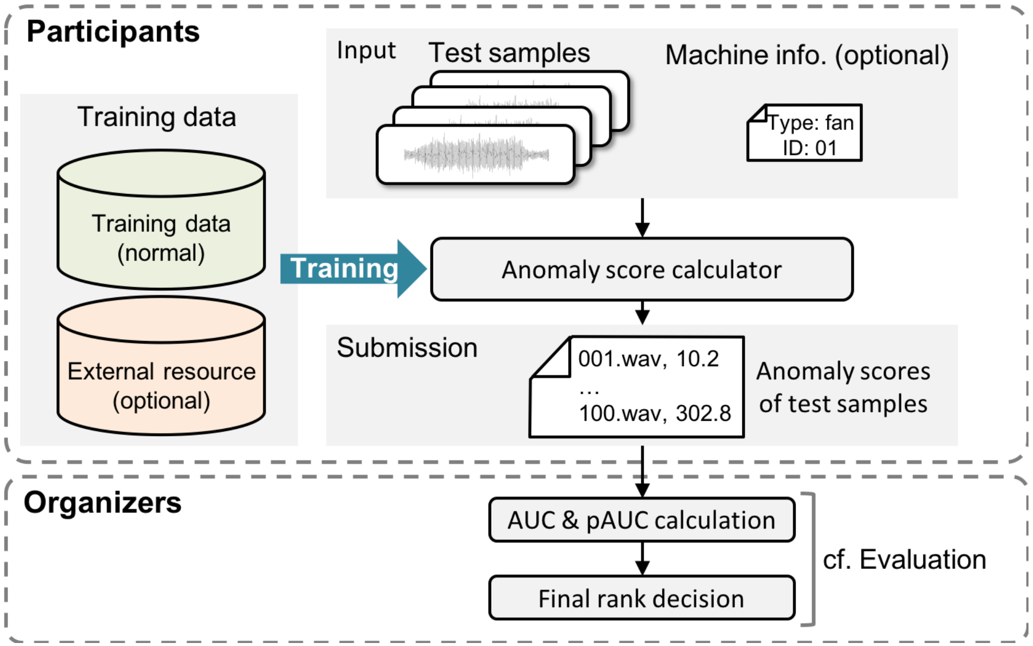
DCASE 2020

- Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring
  - http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds
  - Unsupervised learning: No normal data in train data



Task description

- Anomalous sound detection (ASD)
  - Task to identify whether the sound emitted from a target machine is normal or anomalous
- Main challenge
  - Detect unknown anomalous sounds under the condition that only normal sound samples have been provided as training data



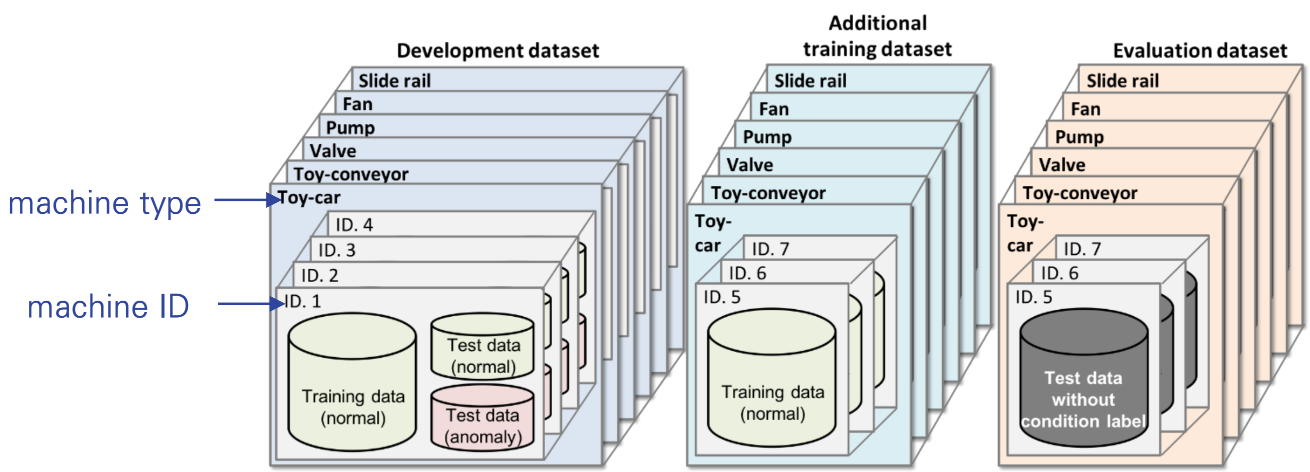
DCASE 2020 dataset

- Development dataset: https://zenodo.org/record/3678171
- Machine Type and Machine ID
  - Six machine types: toy-car, toy-conveyor, valve, pump, fan, and slide rail
  - Each Machine Type has three or four Machine IDs.
  - Machine ID: identifier of each machine of the same type

    (i) around 1,000 samples of normal sounds for training

    (ii) 100-200 samples each of normal and anomalous sounds for the test.

    The normal and anomalous sound samples in (ii) are only for checking performance therefore the sound samples in (ii) shall not be used for training.



Dataset details

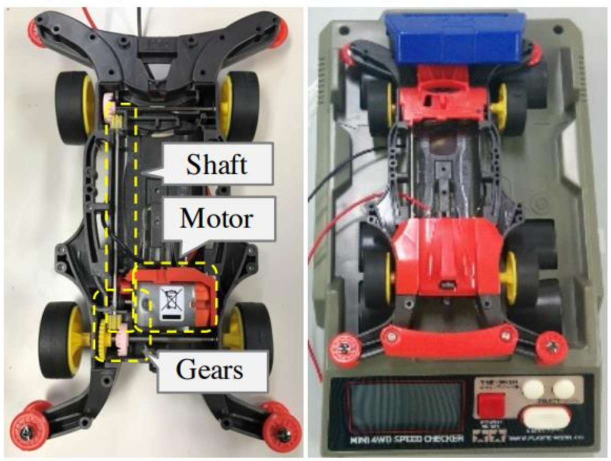
- Machine data from two datasets
  - ToyADMOS (https://arxiv.org/abs/1908.03299)
  - MIMII (http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf)

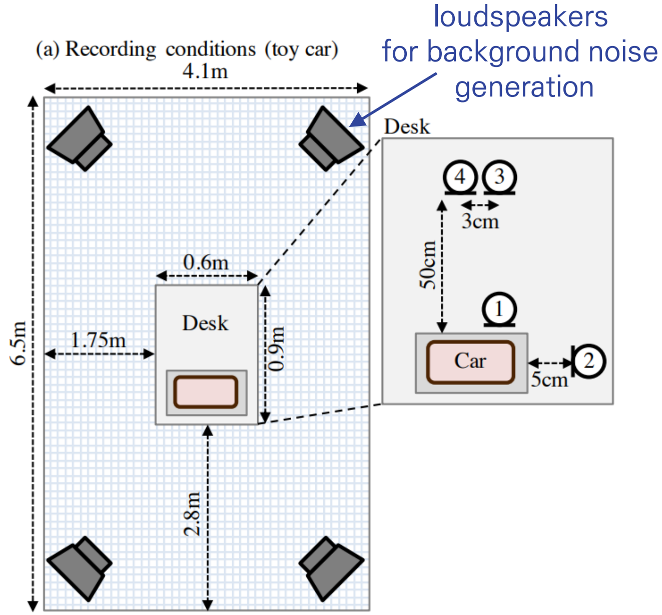
- ToyADMOS dataset
  - Toycar: product inspection
  - Toy conveyer: fault diagnosis for a fixed machine
  - Machine-operating sounds and environmental noise are individually recorded for simulating various noise levels.
  - Multiple machines of the same class are used; each machine belongs to the same class of toys but has a different detailed structure

ToyADMOS: Toycar

- Four toy machines with two types of motors and bearings
- Mimicking the product inspection

(a) Toy car (b) Inspection device



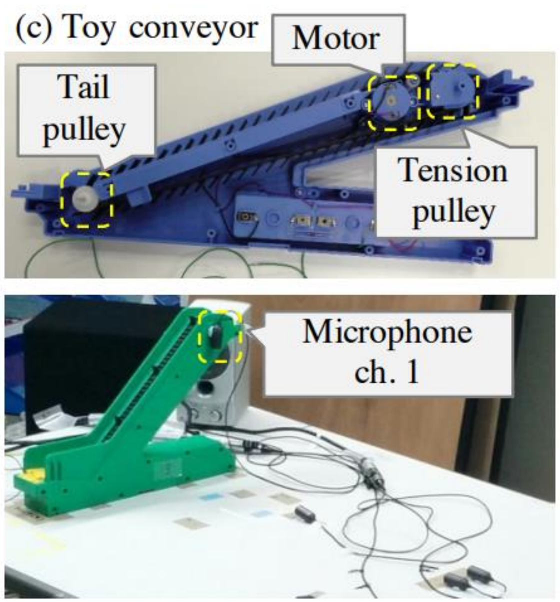


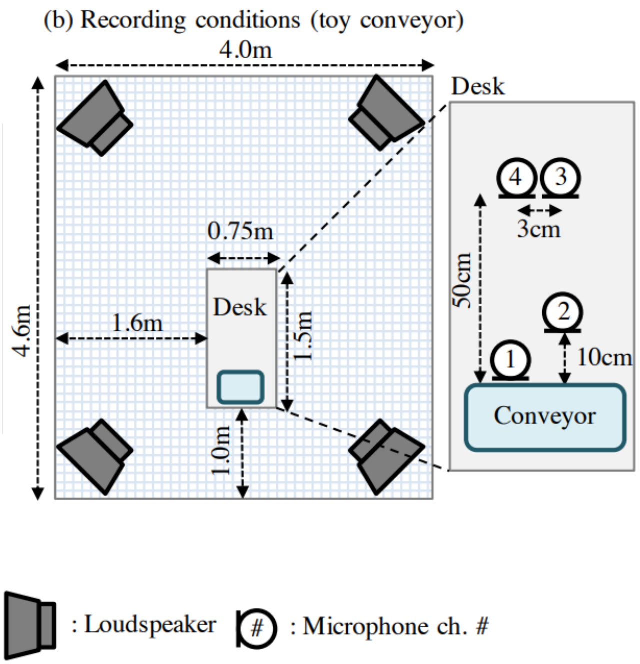
  - Environmental noise: recorded in an actual factory (collision, drilling, pumping, and airbrushing)

- Anomalous sounds were generated by deliberately damaging the shaft, gears, tires, and voltage

ToyADMOS: Toyconveyor

- Toy conveyor transporting a small tin toy car by driving a belt
- Three types of conveyors of different sizes
- Fault diagnosis task





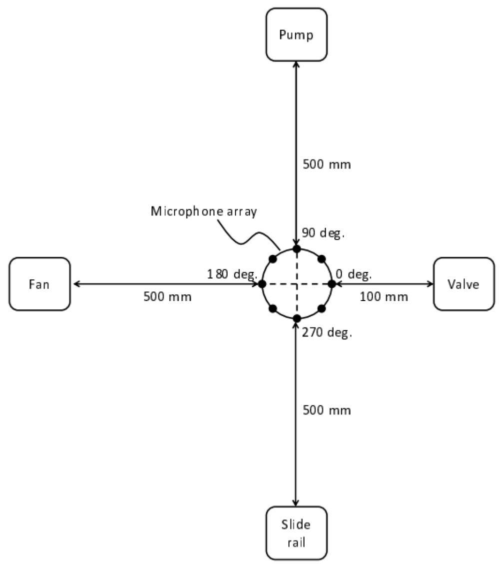
- Anomalous sounds: damaging the tension pulley, trail pulley, and belt and excessively lowering/raising the voltage

MIMII

- Measurement
  - Six microphones (circular array)
  - 10 second sound segments for each data
  - Six machine IDs, total 26092 samples
  - Sampling at 16 kHz, reverberant environment
  - Mixed with factory background noises

Table 1: MIMII dataset content details.

Machine type / model ID		Segments for normal condition	Segments for anomalous condition
Valve	00	991	119
	01	869	120
	02	708	120
	03	963	120
	04	1000	120
	05	999	400
	06	992	120
Pump	00	1006	143
	01	1003	116
	02	1005	111
	03	706	113
	04	702	100
	05	1008	248
	06	1036	102
Fan	00	1011	407
	01	1034	407
	02	1016	359
	03	1012	358
	04	1033	348
	05	1109	349
	06	1015	361
Slide rail	00	1068	356
	01	1068	178
	02	1068	267
	03	1068	178
	04	534	178
	05	534	178
	06	534	89
Total		26092	6065



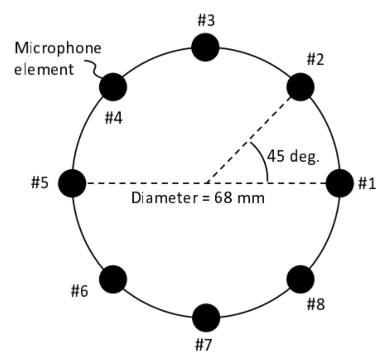


Figure 1: Circular microphone array.

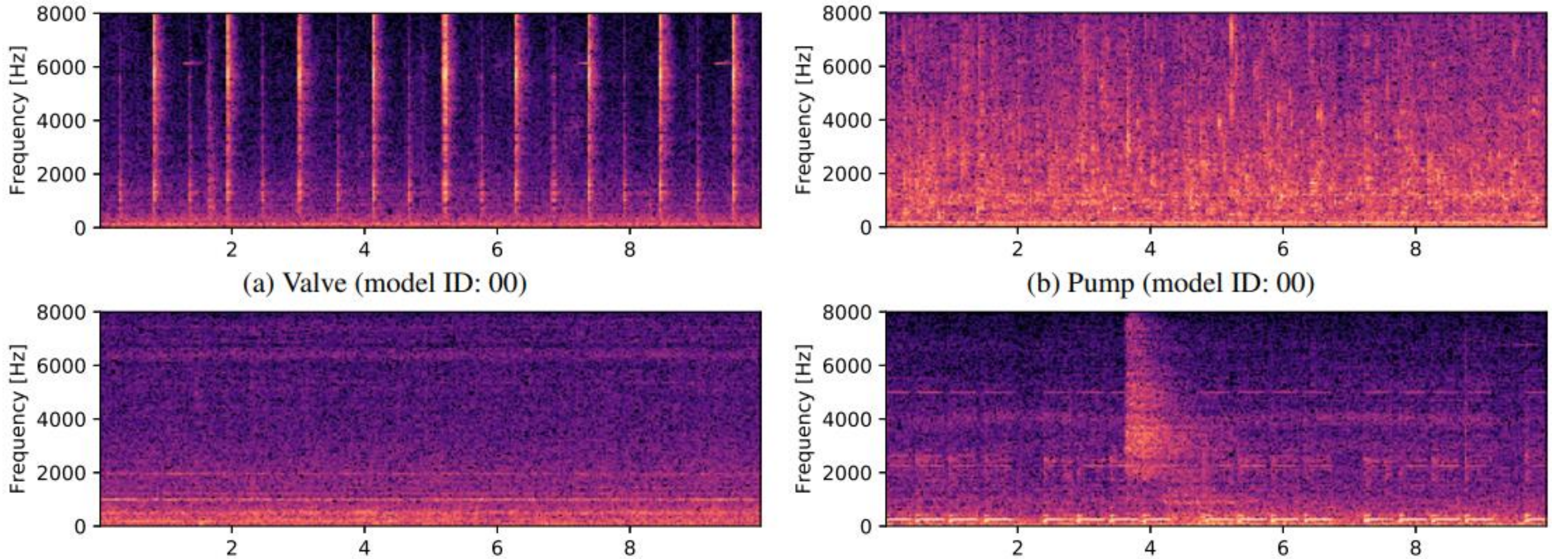
MIMII

Solenoid valves repeatedly open/close

Water pumps continuously drain/discharge waters

Normal

Anomaly



(a) Valve (model ID: 00)

(b) Pump (model ID: 00)

(c) Fan (model ID: 00)

(d) Slide rail (model ID: 00)

Figure 3: Examples of power spectrograms under normal condition at 6-dB SNR.

Industrial fans providing continuous gas/air flow into the factory

linear slide rails consisting of moving platform and a stage base

DCASE 2020 dataset

- Data labels : Machine Type, Machine ID, condition
  - Machine Type : Toy Car, Toy Conveyor, Valve, Pump, Fan, Slider rail
  - Machine ID : 1,2,3,4 / 1,2,3 / 0,2,4,6 / 0,2,4,6 / 0,2,4,6 / 0,2,4,6
    Different machine ID means different recording situation

Table 2: List of operations and anomalous conditions

Machine type	Operations	Examples of anomalous conditions
Valve	Open/close repeat with different timing	More than two kinds of contamination
Pump	Suction from/ discharge to a water pool	Leakage, contamination, clogging, etc.
Fan	Normal work	Unbalanced, voltage change, clogging, etc.
Slide rail	Slide repeat at different speeds	Rail damage, loose belt, no grease, etc.

Normal Anomaly

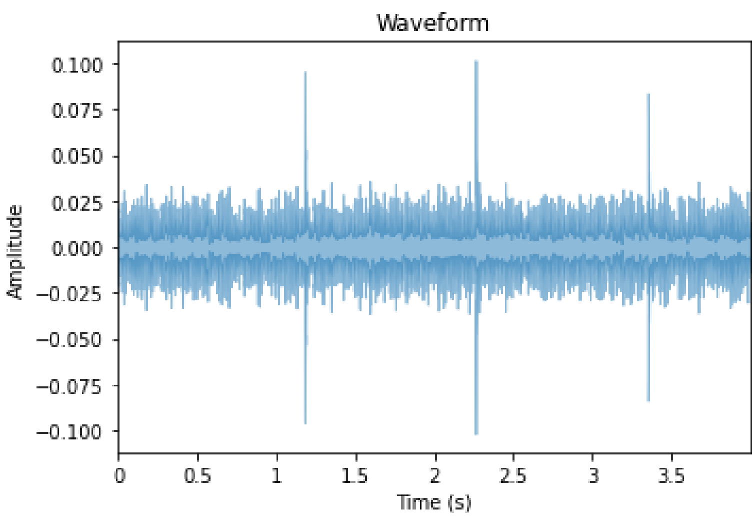
DCASE 2020 Dataset
& Audio Data Preprocessing

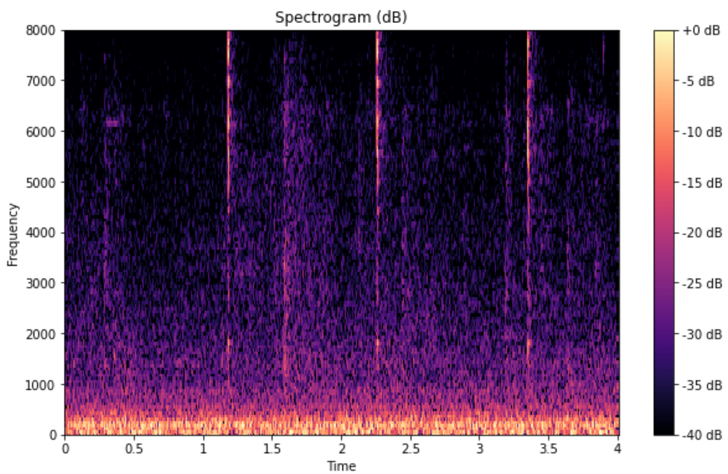
Audio Data Preprocessing

Audio Data Preprocessing

- Contents
- Audio Data Preprocessing
- Acoustic Feature
  - Waveform, Spectrogram, Melspectrogram
- Front End System
  - Restricting dB
  - Normalization
    - Normalization using single mean, std
  - Filtering
    - Pre Emphasis
    - De Emphasis

1D Waveform → 2D Image

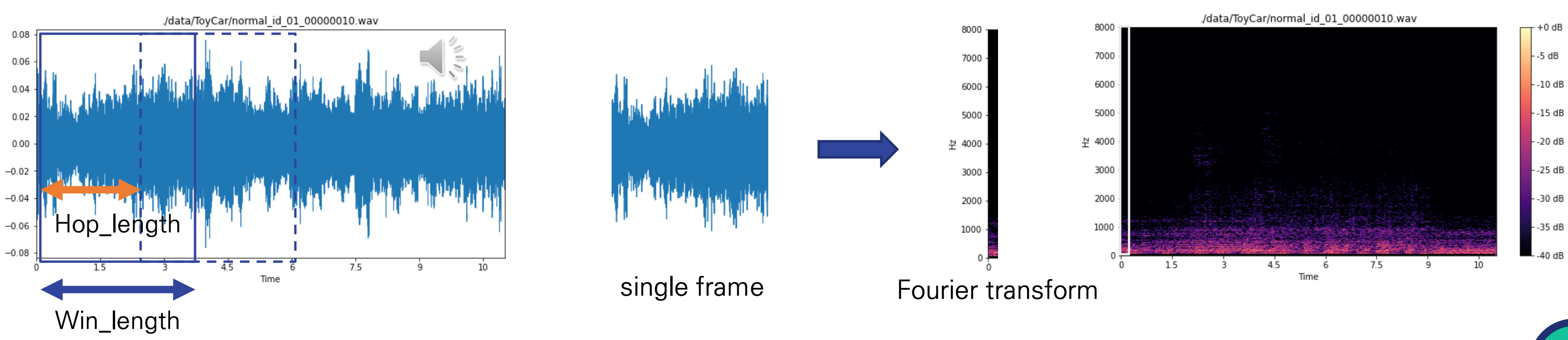




Audio Data Preprocessing

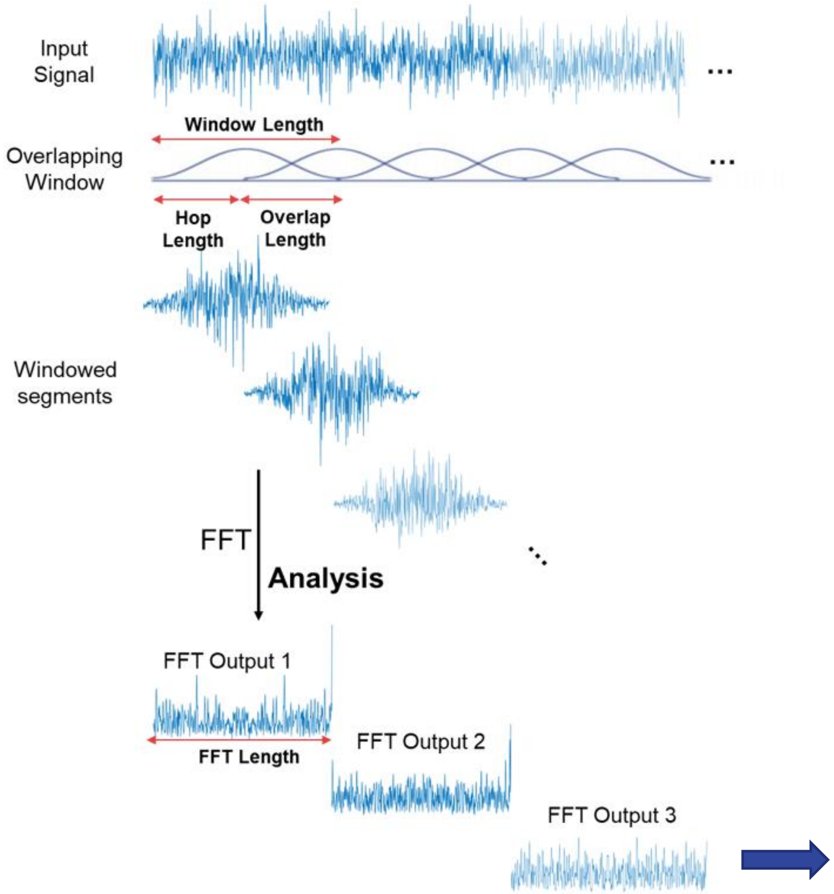
- Audio Data
  - Use librosa.load to load audio file (sr = 16 kHz sampling rate)

- Spectrogram
  - STFT (Short time Fourier Transform) for frame-wise processing
  - Use librosa.feature.stft to convert audio to spectrogram
  - Win_length
    - temporal width of a single analysis window (default = n_fft)
  - Hop_length : shifting of each frames



Audio Data Preprocessing

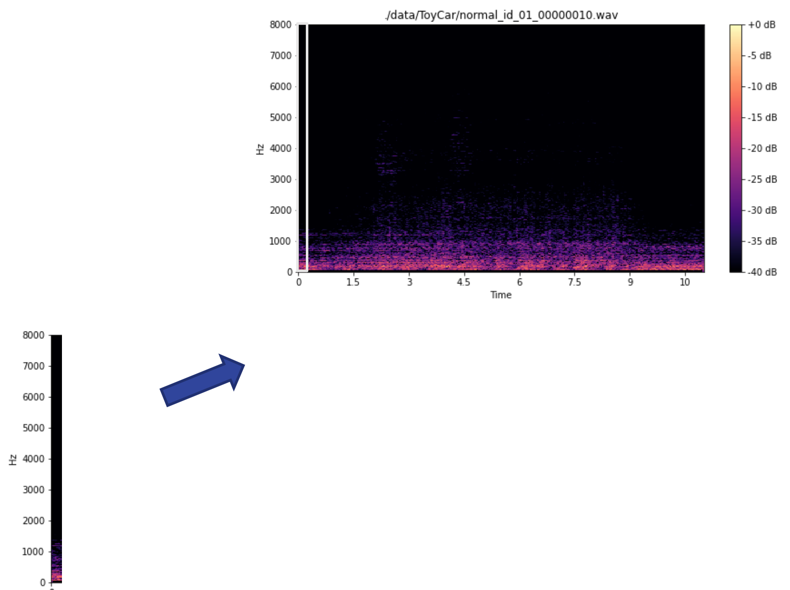
- Overview



1. Number of data in time = Number of data in freq.

   N data in a single window = N frequency data

2. hop & window size determines temporal resolution

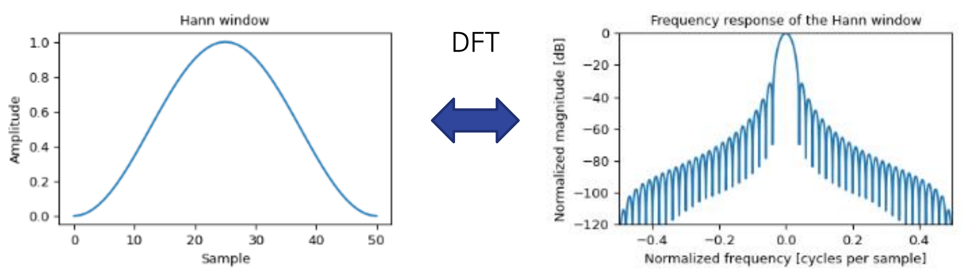


Fourier transform

Audio Data Preprocessing

- n_fft
  - Length of Fourier transform
  - Spectrogram contains frequency from 0 to n_fft / 2 * sr (Nyquist Frequency) so total length is 1+n_fft/2
  - Frequency resolution:
    - (n_fft > win_length) pad zeros for increasing frequency resolution

- window = ‘hann’
  - Window function for reducing spectral leakage
  - List of all supported windows: https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.signal.get_window.html#scipy.signal.get_window

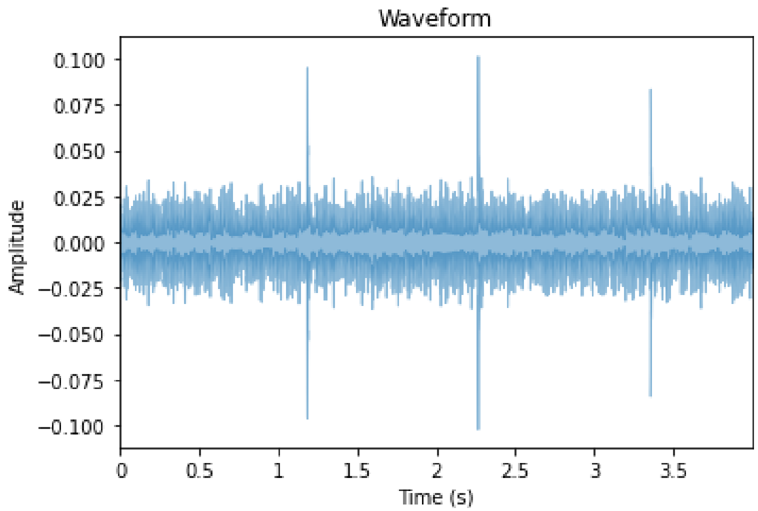


$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1$$

Audio Data Preprocessing

- Waveform displayed in time

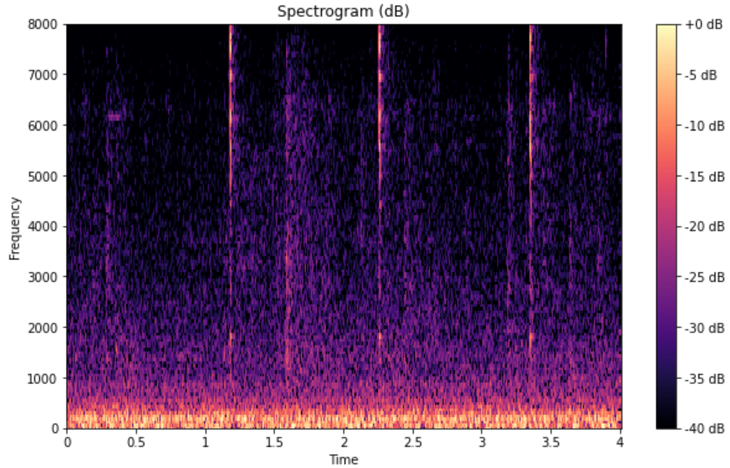
```
mydata = '/content/gdrive/MyDrive/Dcase/valve_test/anomaly_id_00_00000000.wav'

y, sr = librosa.load(mydata, sr=None)
y = y[0:(sr*4)]

plt.figure()
librosa.display.waveplot(y, sr, alpha=0.5)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.title("Waveform")
```



Audio Data Preprocessing

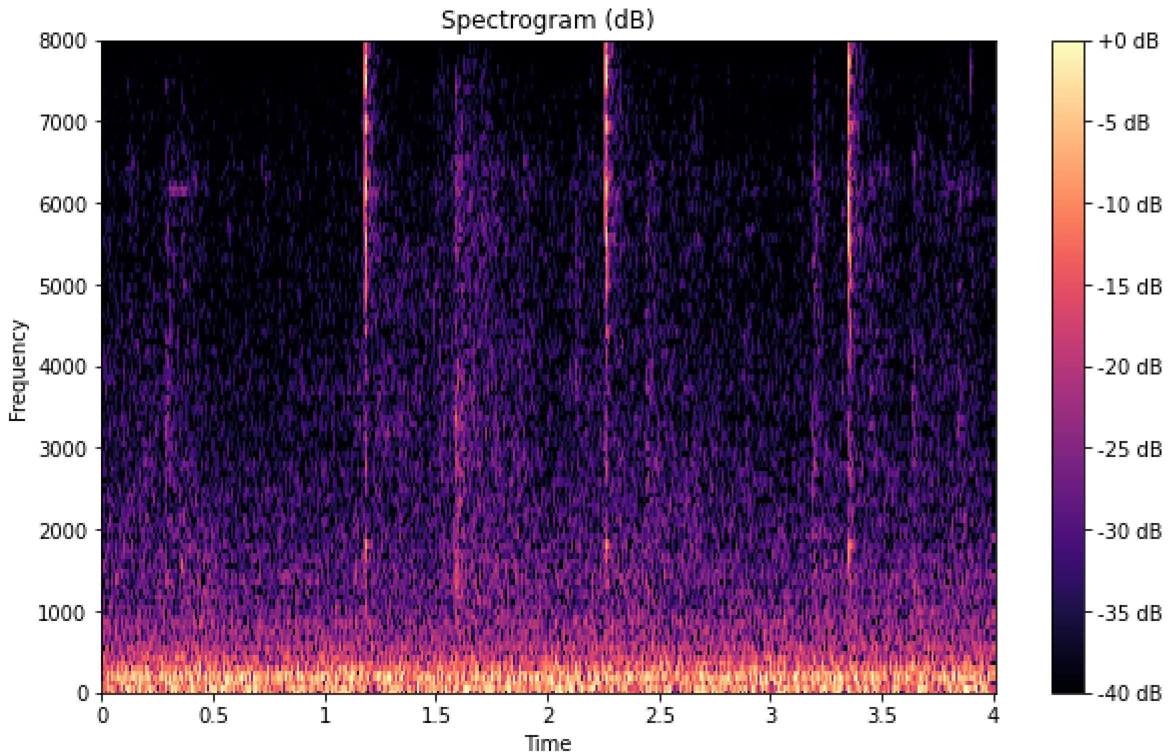
- Spectrogram parameters

```
stft = librosa.stft(y, win_length=nwin,  n_fft=nfft, hop_length=nhop)

mag = np.abs(stft)
log_spec = librosa.amplitude_to_db(mag)

fig = plt.figure(figsize=(10, 6))
librosa.display.specshow(log_spec, sr=sr, hop_length=nhop,cmap=plt.get_cmap('magma'),
                         vmin=-40, vmax=0, x_axis='time', y_axis='linear')
fig.gca().label_outer()
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.colorbar(format="%+2.0f dB")
plt.title("Spectrogram (dB)")
```



Audio Data Preprocessing

- Spectrogram parameters



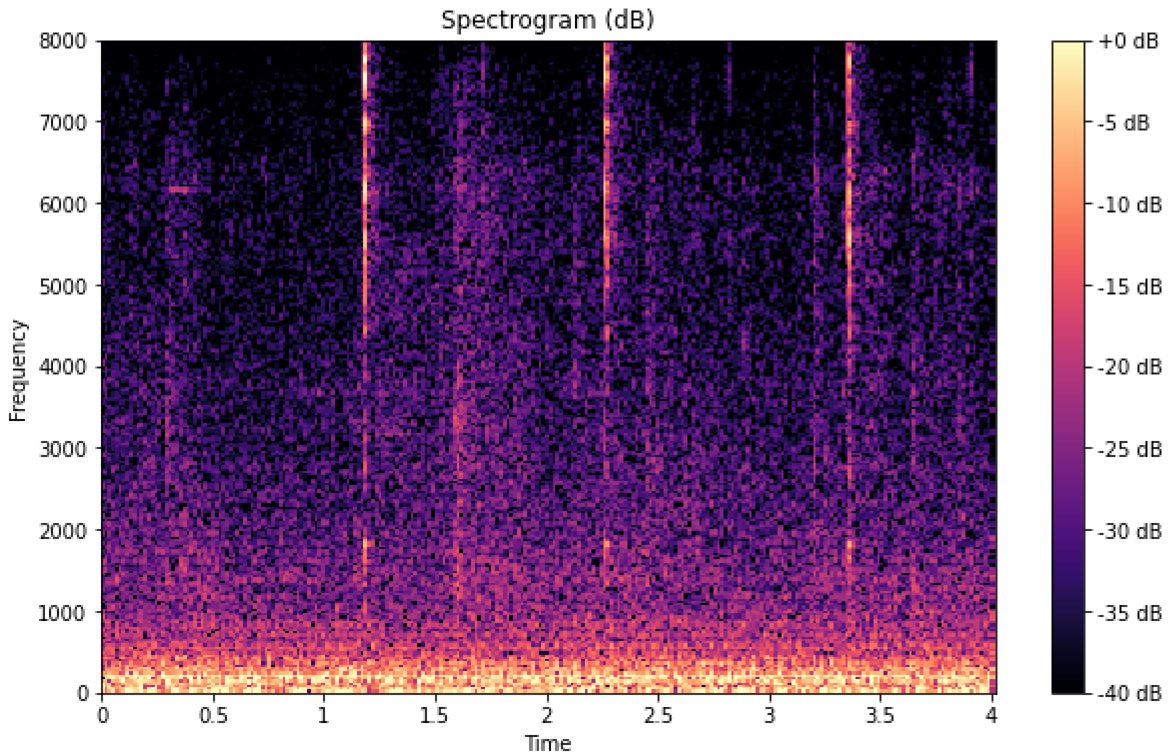
https://librosa.org/doc/main/generated/librosa.display.specshow.html

```
sr = 16,000 Hz
Win_length = 256
Hop_length = 128
n_fft = 256


waveform size= 4 s
Image size?

(f=###,t=313)
```

Win length ↓ : temporal resolution ↑ frequency resolution ↓
Win length ↑ : frequency resolution ↑ temporal resolution ↓

Audio Data Preprocessing

- Spectrogram parameters



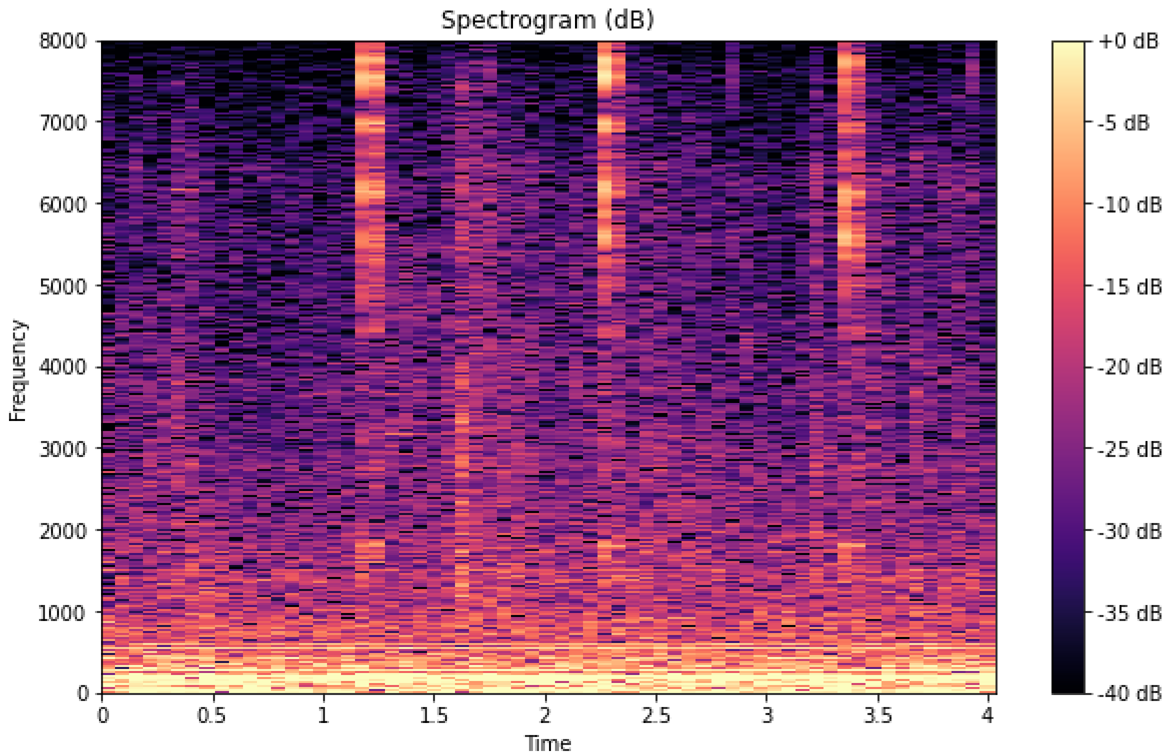
https://librosa.org/doc/main/generated/librosa.display.specshow.html

```
sr = 16,000 Hz
Win_length = 512
Hop_length = 256
n_fft = 512

waveform size= 4 s
```

Win length ↓ : temporal resolution ↑ frequency resolution ↓
Win length ↑ : frequency resolution ↑ temporal resolution ↓

Audio Data Preprocessing

- Spectrogram parameters



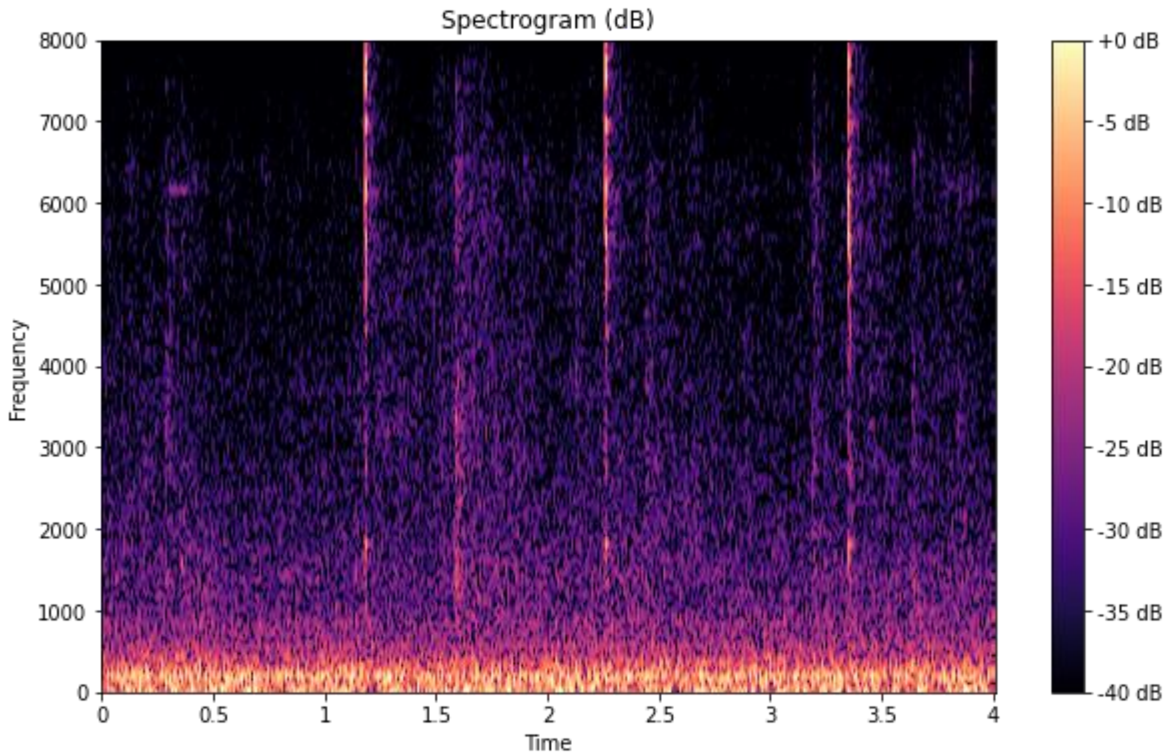

```
sr = 16,000 Hz
Win_length = 2048
Hop_length = 1024
n_fft = 2048

waveform size= 4 s
```

https://librosa.org/doc/main/generated/librosa.display.specshow.html

Win length ↓ : temporal resolution ↑ frequency resolution ↓
Win length ↑ : frequency resolution ↑ temporal resolution ↓

Audio Data Preprocessing

- Spectrogram parameters



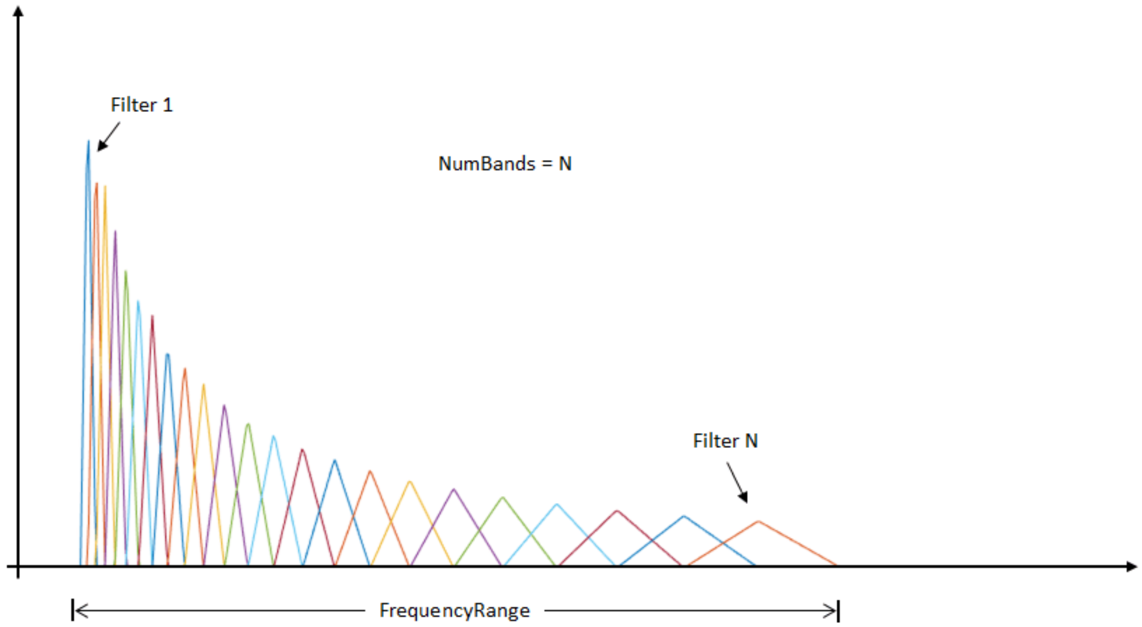
https://librosa.org/doc/main/generated/librosa.display.specshow.html

```
sr = 16,000 Hz
Win_length = 256
Hop_length = 128
n_fft = 2048

waveform size= 4 s
```

Win length ↓ : temporal resolution ↑ frequency resolution ↓
Win length ↑ : frequency resolution ↑ temporal resolution ↓

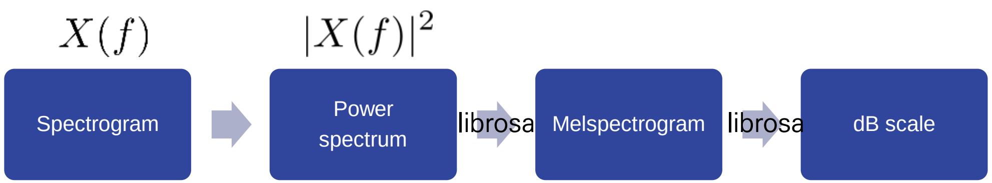
Audio Data Preprocessing

- Melspectrogram
  - Frequency scale similar to human perception scale
  - Fine resolution at low-frequencies

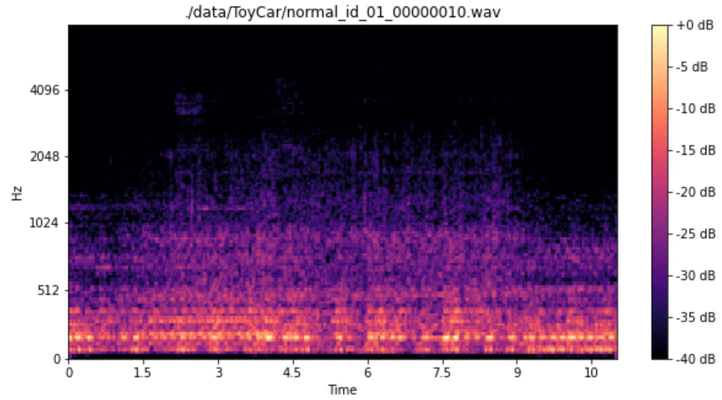


Audio Data Preprocessing

- Mel-spectrogram
  - Process

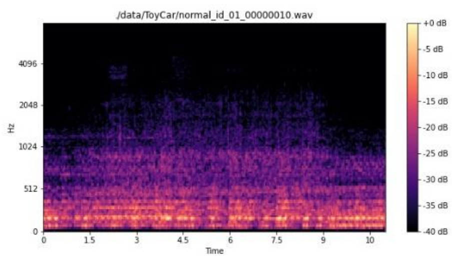
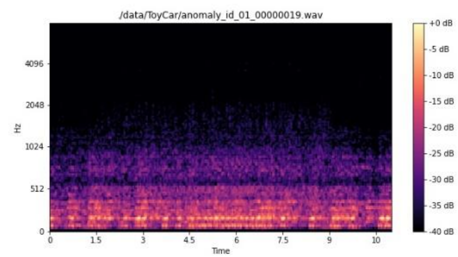
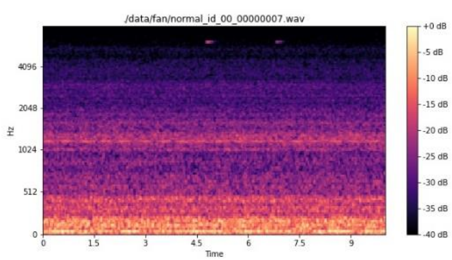
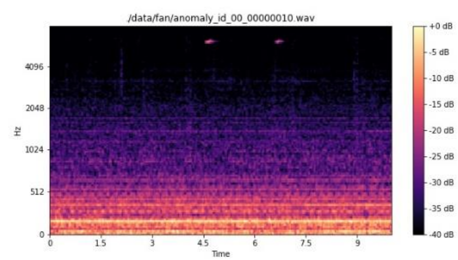
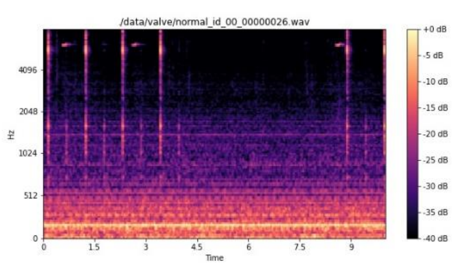
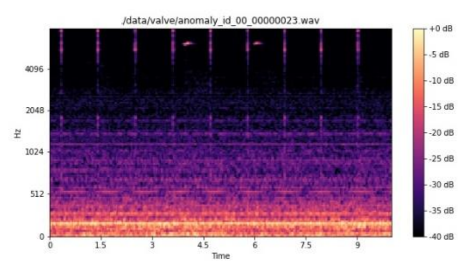



```
mel_spec = librosa.feature.melspectrogram(y=y, sr=sr, n_fft=n_fft, hop_length=hop_length, n_mels=n_mels, power=power)
log_mel_spec = 10.0 * np.log10(mel_spec + sys.float_info.epsilon)
librosa.display.specshow(log_mel_spec, cmap=plt.get_cmap('magma'), vmin=-40, vmax=0, x_axis='frames', y_axis='mel')
plt.title('Log Mel-Spectrogram')
```



Audio Data Preprocessing

- Mel-spectrograms of normal and anomaly data

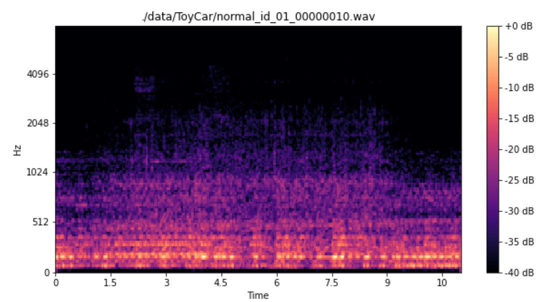
	Normal	Anomaly
Toy Car		
Fan (Stationary)		
Valve (Temporal)		

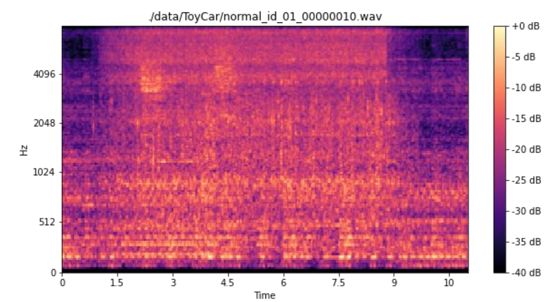
Audio Data Preprocessing

- Front End System
  - Filtering
    - Filtering can emphasize audio feature or reduce noise.
    - <ex> Pre Emphasis

$$y[n] \rightarrow y[n] - 0.97 * y[n-1]$$

Pre emphasis can emphasize high frequency component while de emphasize stationary signal (low frequency component).

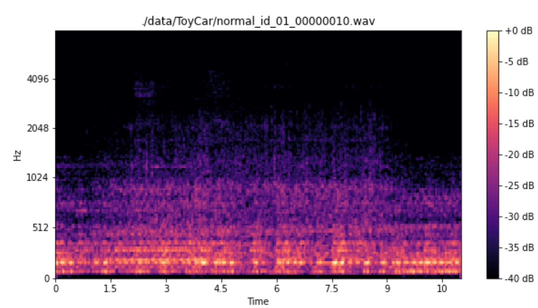


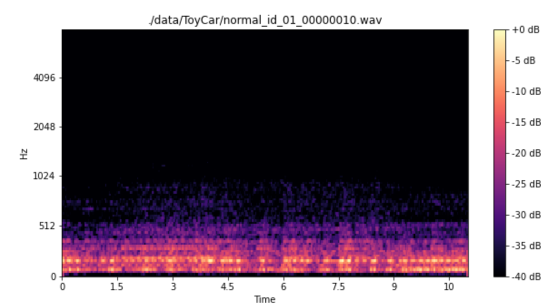


- <ex> De Emphasis

$$y[n] \rightarrow y[n] + 0.97 * y[n-1]$$

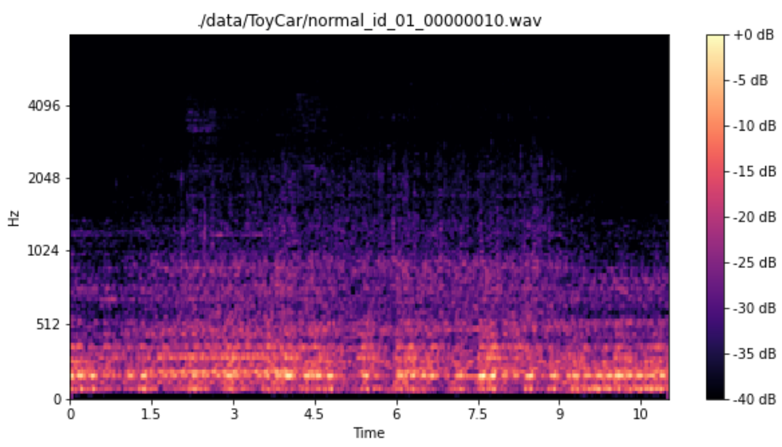
De emphasis can de emphasize high frequency component and relatively emphasize stationary signal which is low frequency component.

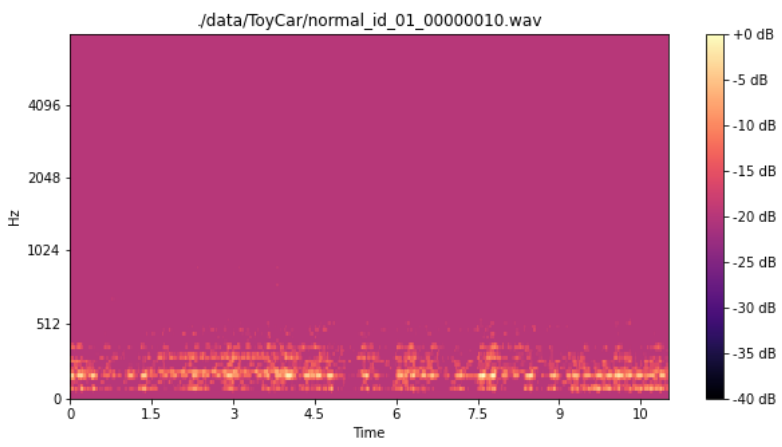




Audio Data Preprocessing

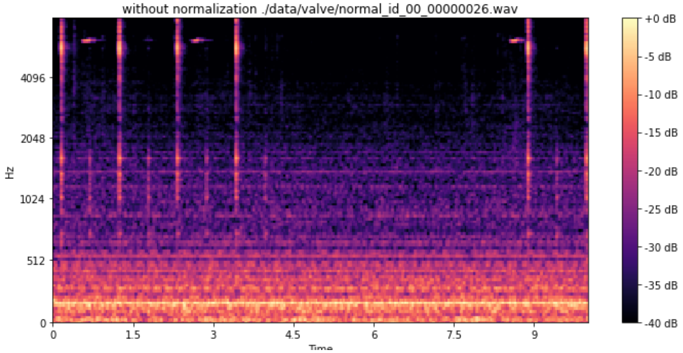
- Front End System
  - Restricting dynamic range
    - Dynamic range: maximum dB – minimum dB of signal
    - If dynamic range is too wide, model can focus on unnecessary low dB region

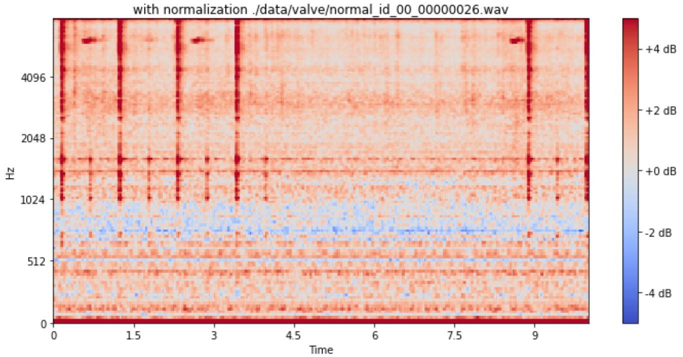




Audio Data Preprocessing

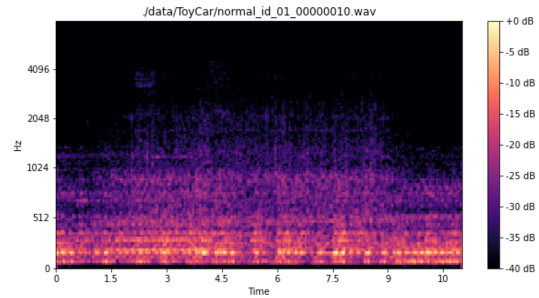
- Normalization
  - Normalization is useful for emphasizing specific acoustic feature.
    - Normalizing spectrogram in time direction
      - Reduce stationary noise
      - Temporal variation is emphasized.
  - In practice code, we normalize the same machine type data
    - Using mean and std calculated from the same machine type
    - Batch normalization or layer normalization can also be applied

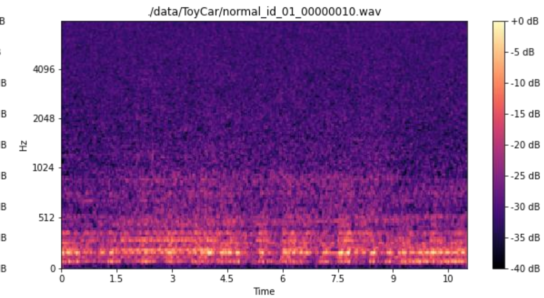




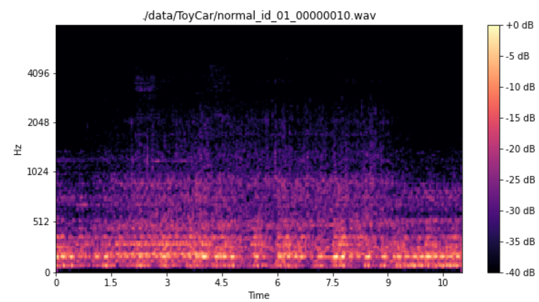
Audio Data Preprocessing

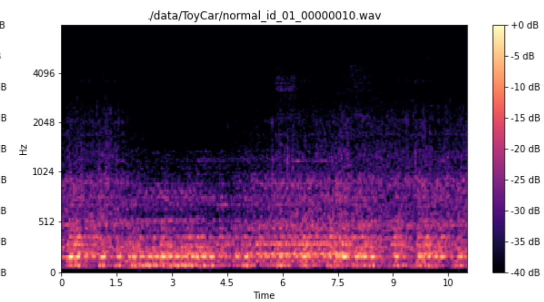
- Data Augmentation
  - Adding noise
    - White noise (AWGN)
      Additive White Gaussian Noise
      → practice code
    - Real background noise



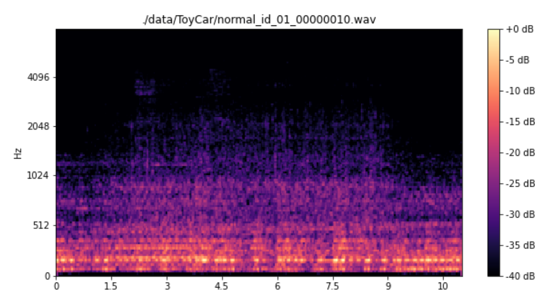


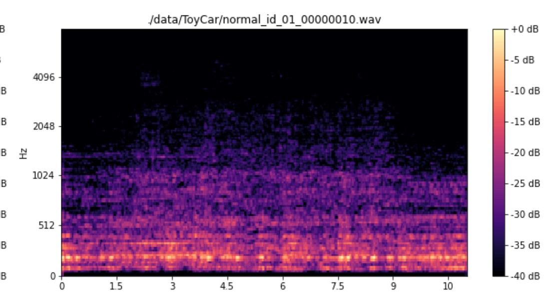
  - Time Shift





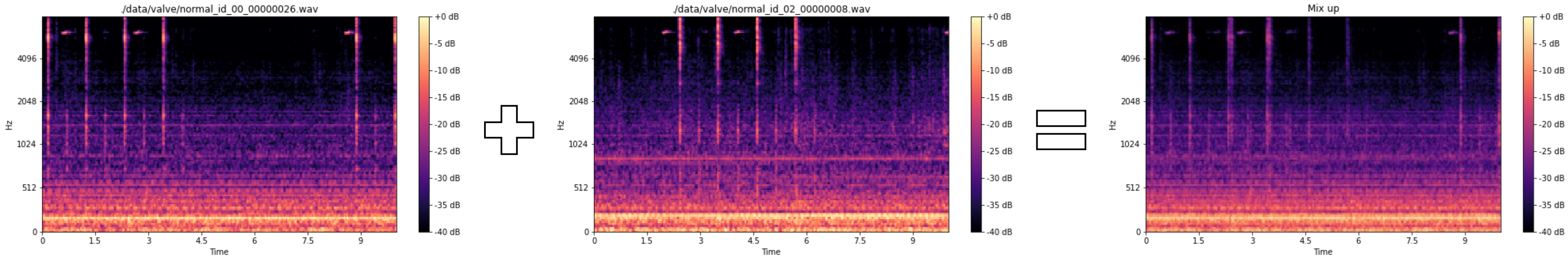
  - Pitch Shift
    - Pitch shift in librosa can shift pitch in ‘n_steps’.
    - Here, 12 steps means 1 octave for default and this step size can be modified by ‘bins_per_octave’





Audio Data Preprocessing

- Audio mix-up
  - Mix-up



[1, 0, 0, 0]　　[0, 1, 0, 0]　　[0.6, 0.4, 0, 0]