Lab. - Speech Analysis & Feature Extraction

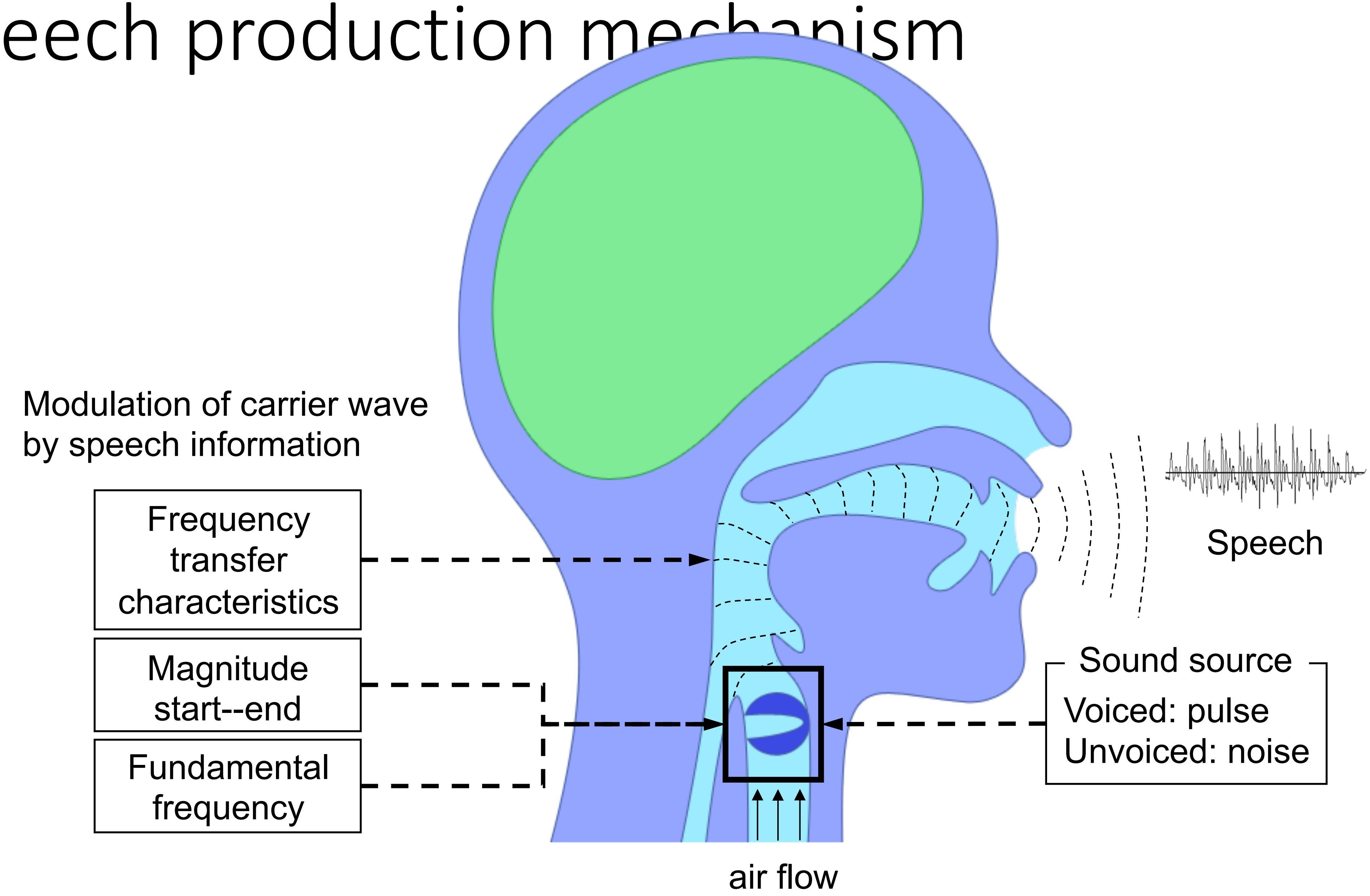
Yuan-Fu Liao

National Taipei University of Technology

Python library for Speech Analysis, Feature Extraction & Data Augmentation

- Speech Analysis & Feature Extraction
  - Librosa - Python library for audio and music analysis
    - https://github.com/librosa/librosa
  - Parselmouth - Praat in Python, the Pythonic way
    - https://github.com/YannickJadoul/Parselmouth
- Data Augmentation
  - Rubberband - An audio time-stretching and pitch-shifting library and utility program
    - https://github.com/breakfastquay/rubberband

Speech production mechanism



Modulation of carrier wave by speech information

Frequency transfer characteristics

Magnitude start--end

Fundamental frequency

Speech

Sound source

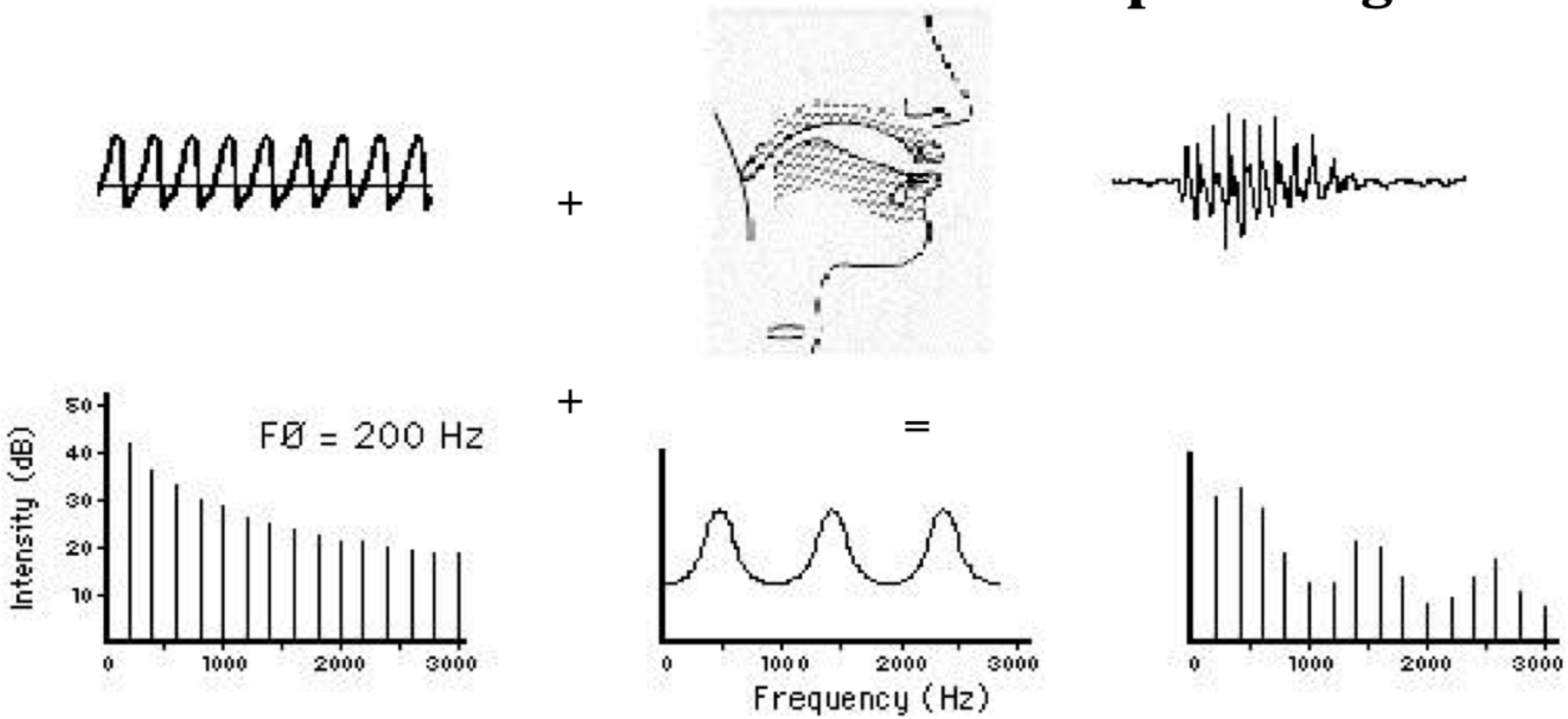
Voiced: pulse

Unvoiced: noise

air flow

Speech Production

Glottal Pulses | Vocal Tract | Speech Signal

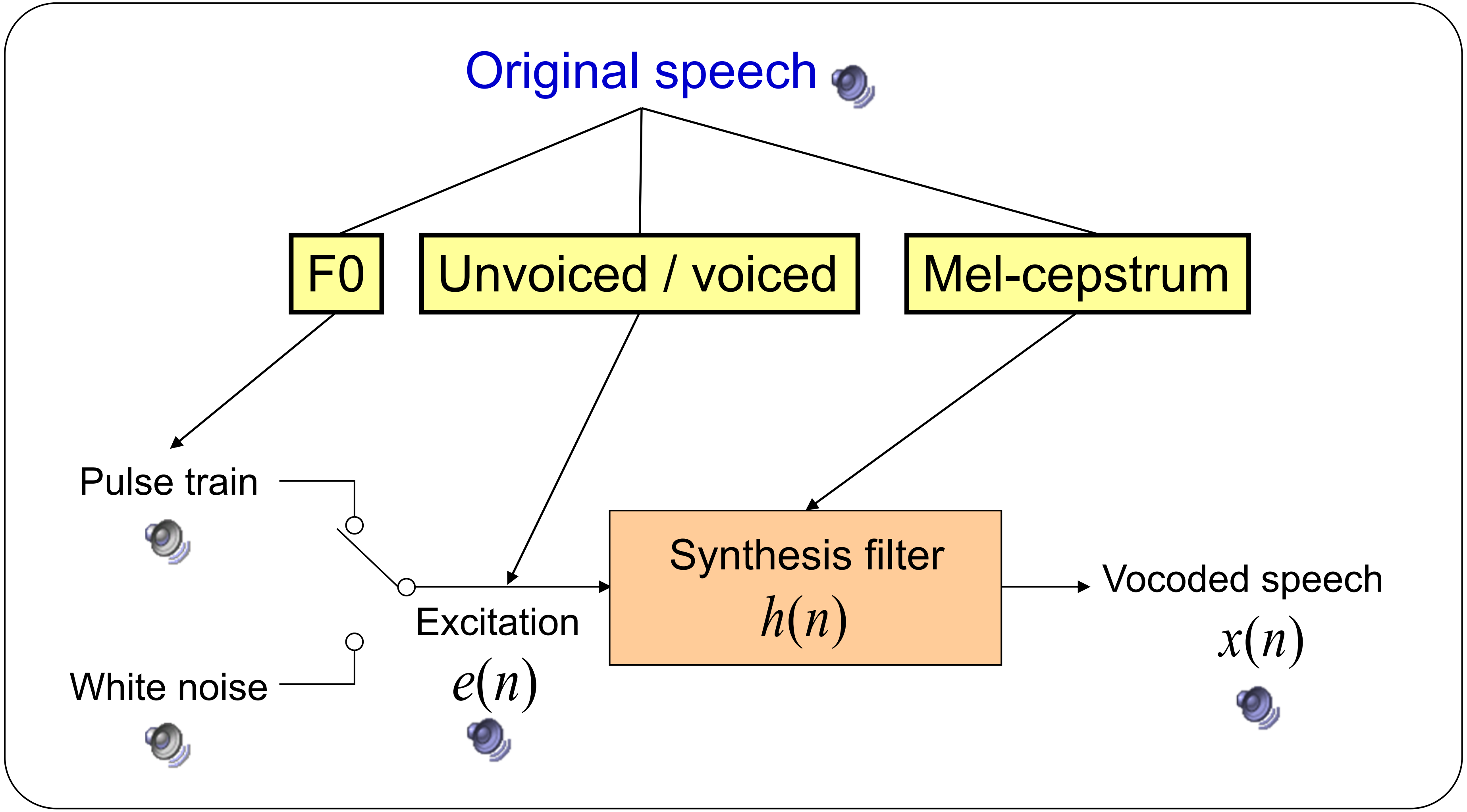


(a) Source Spectrum

(b) Filter Function

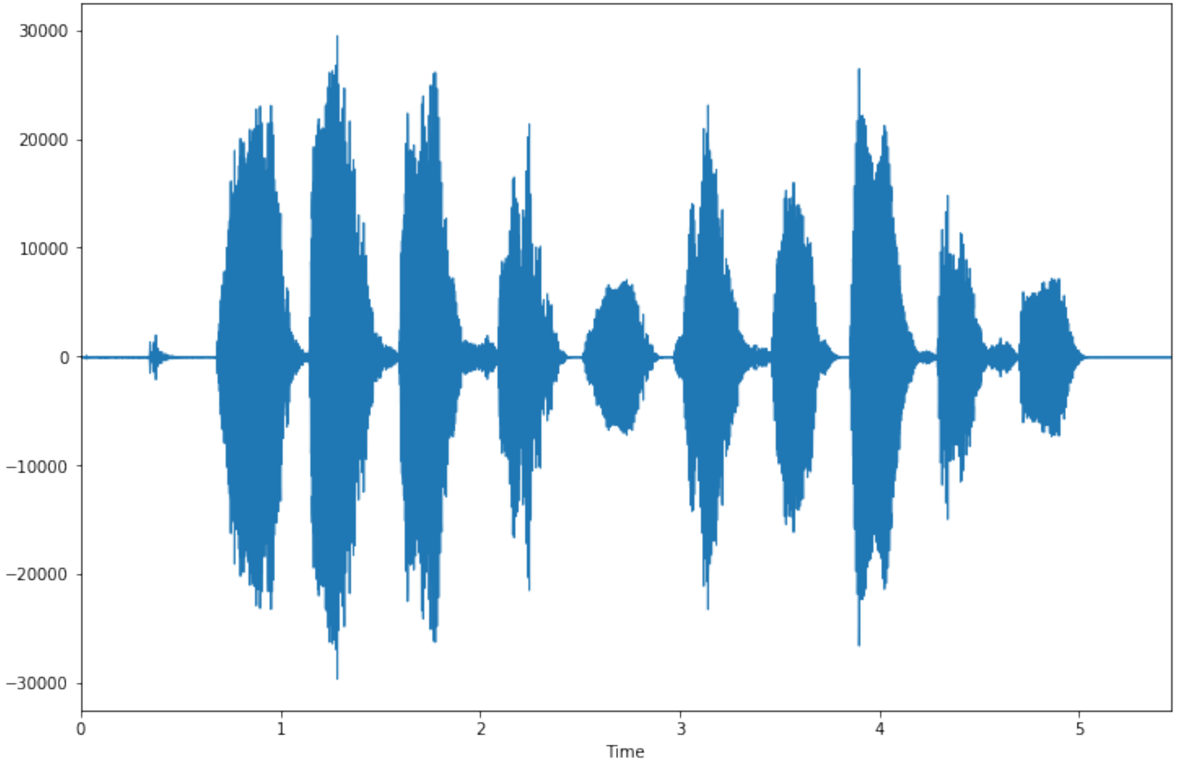
(c) Output Energy Spectrum

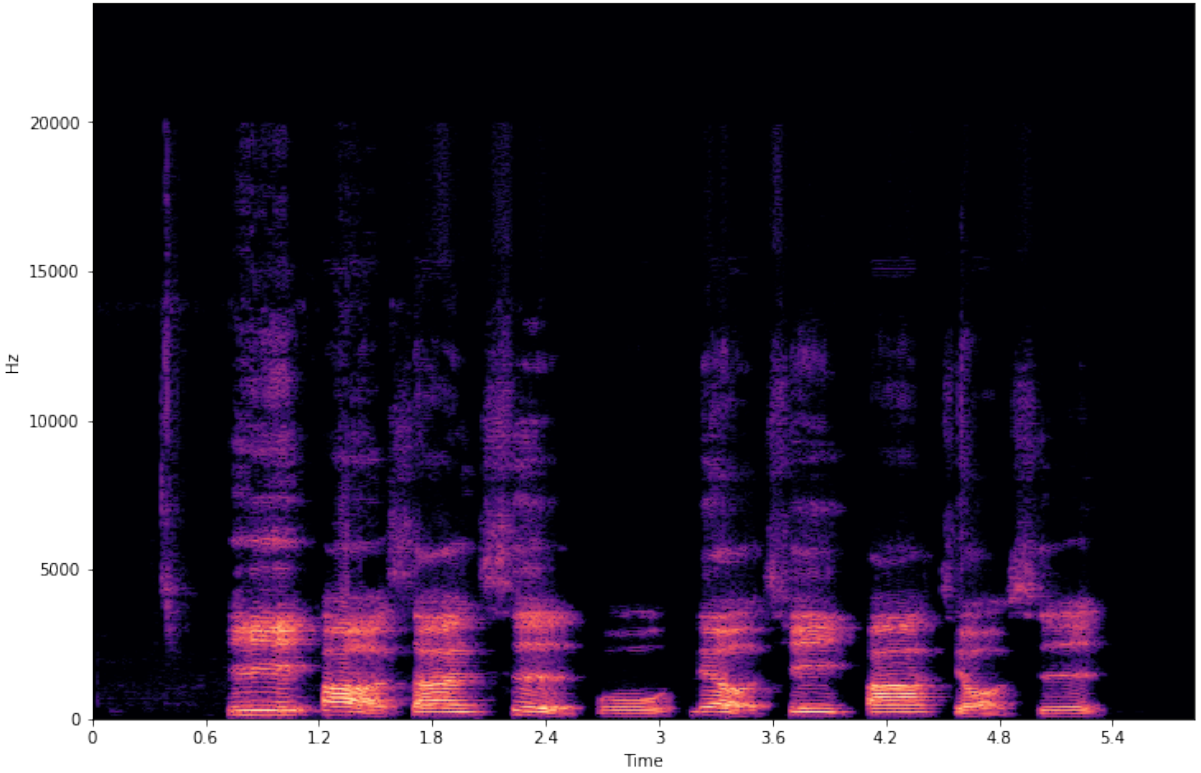
Speech vocoding



Spectrogram

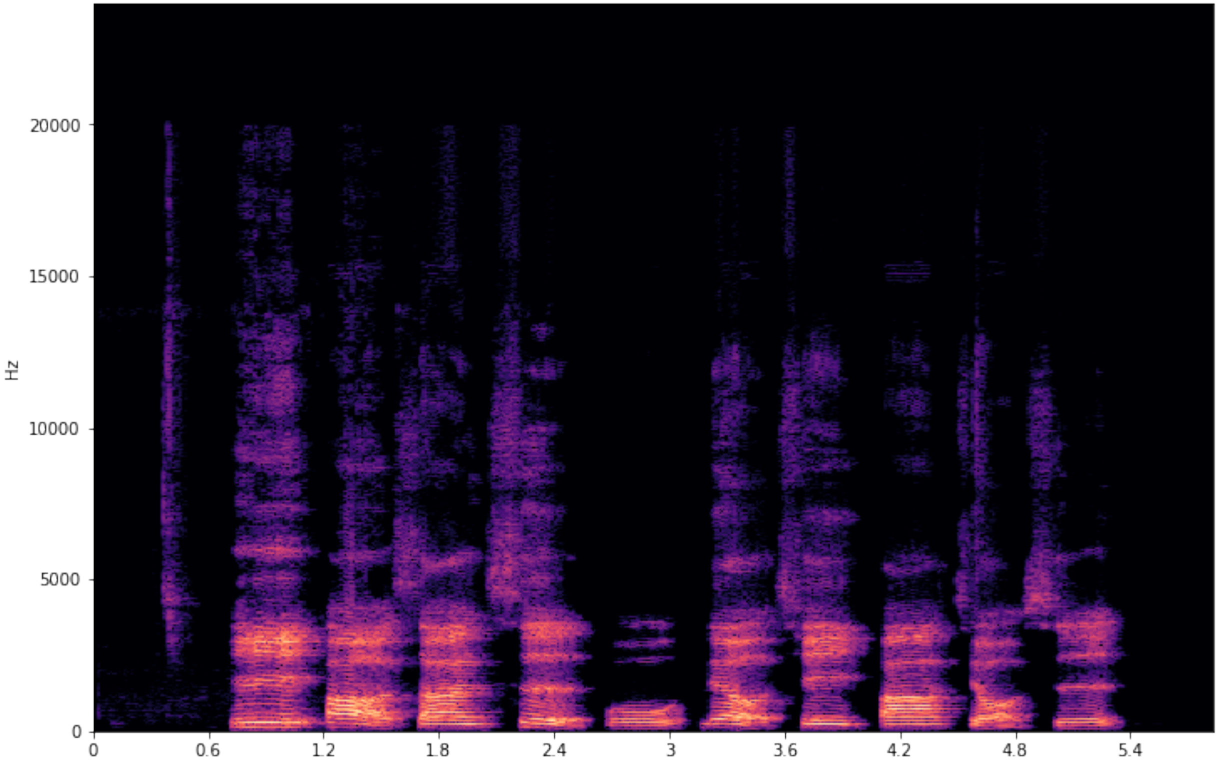
- x = librosa.stft(audio, n_fft=2048, hop_length=480)
- librosa.display.specshow(librosa.amplitude_to_db(np.abs(x)), sr=sr)

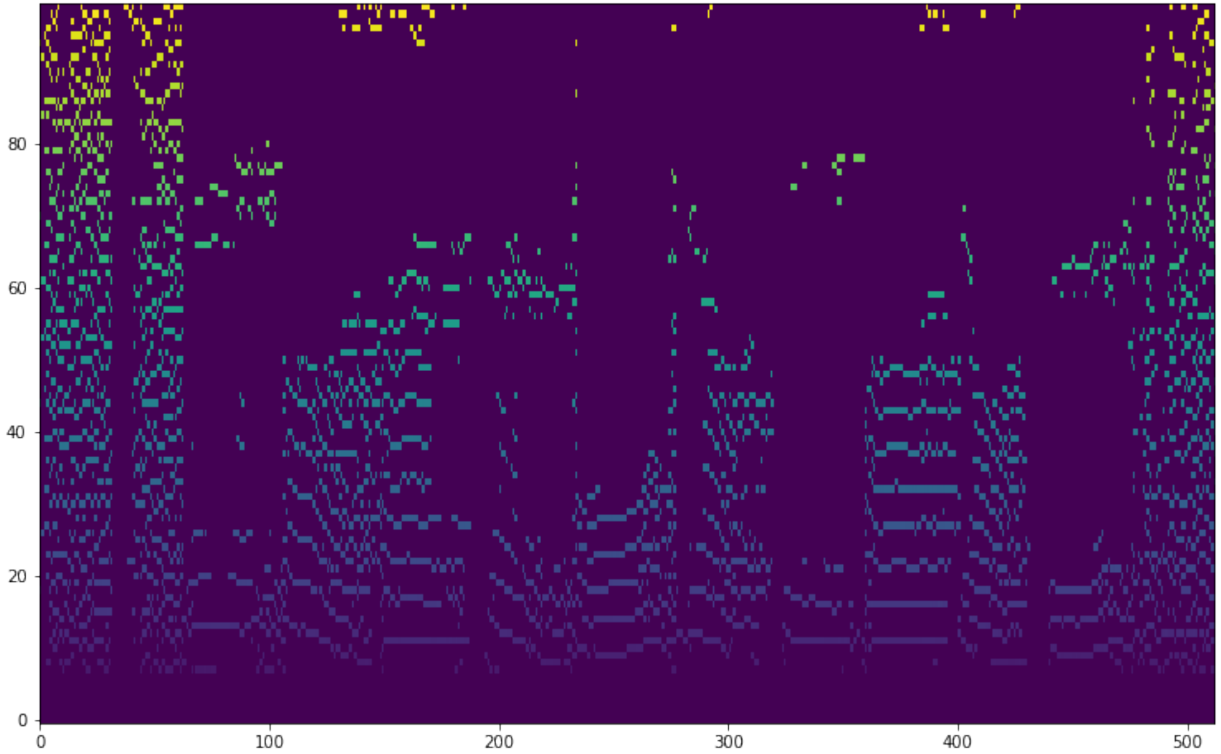




Pitch Tracking

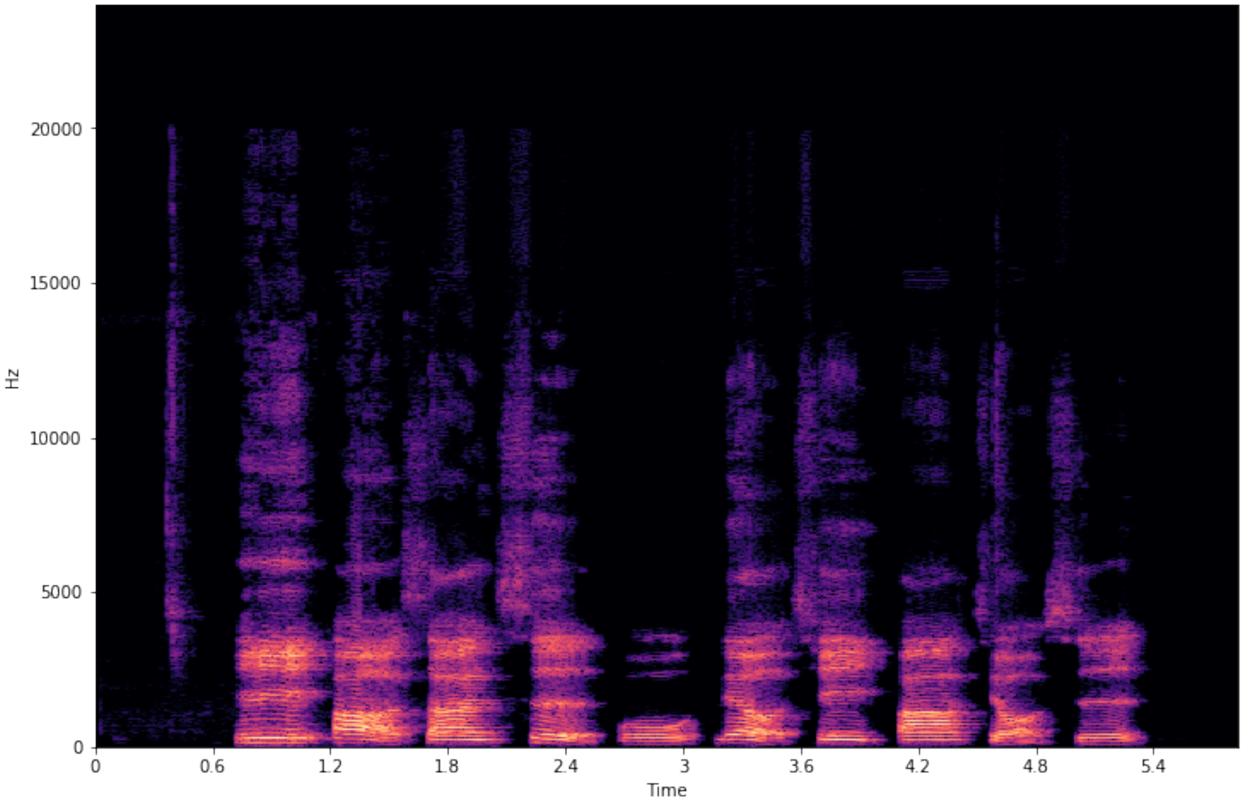
- pitches, magnitudes = librosa.piptrack(y=audio, sr=sr)
- plt.imshow(pitches[:, :], aspect="auto", interpolation="nearest", origin="bottom")

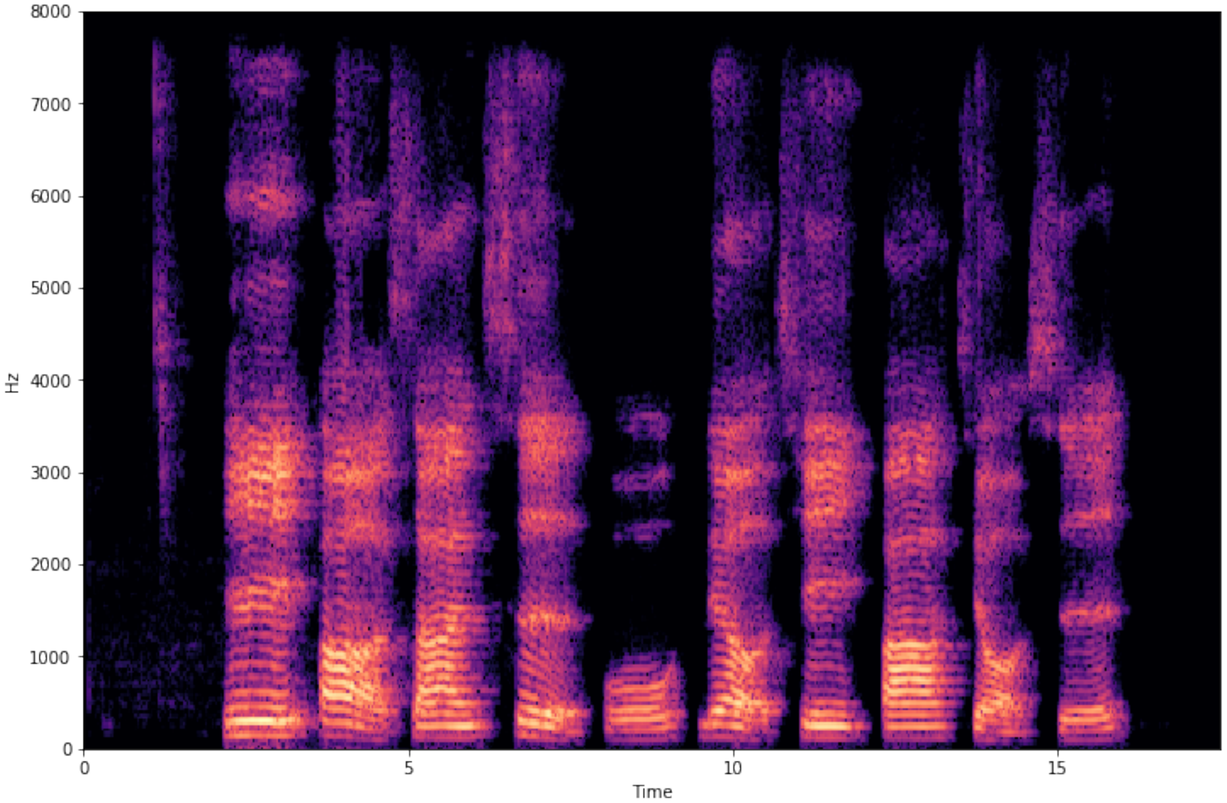




Resampling

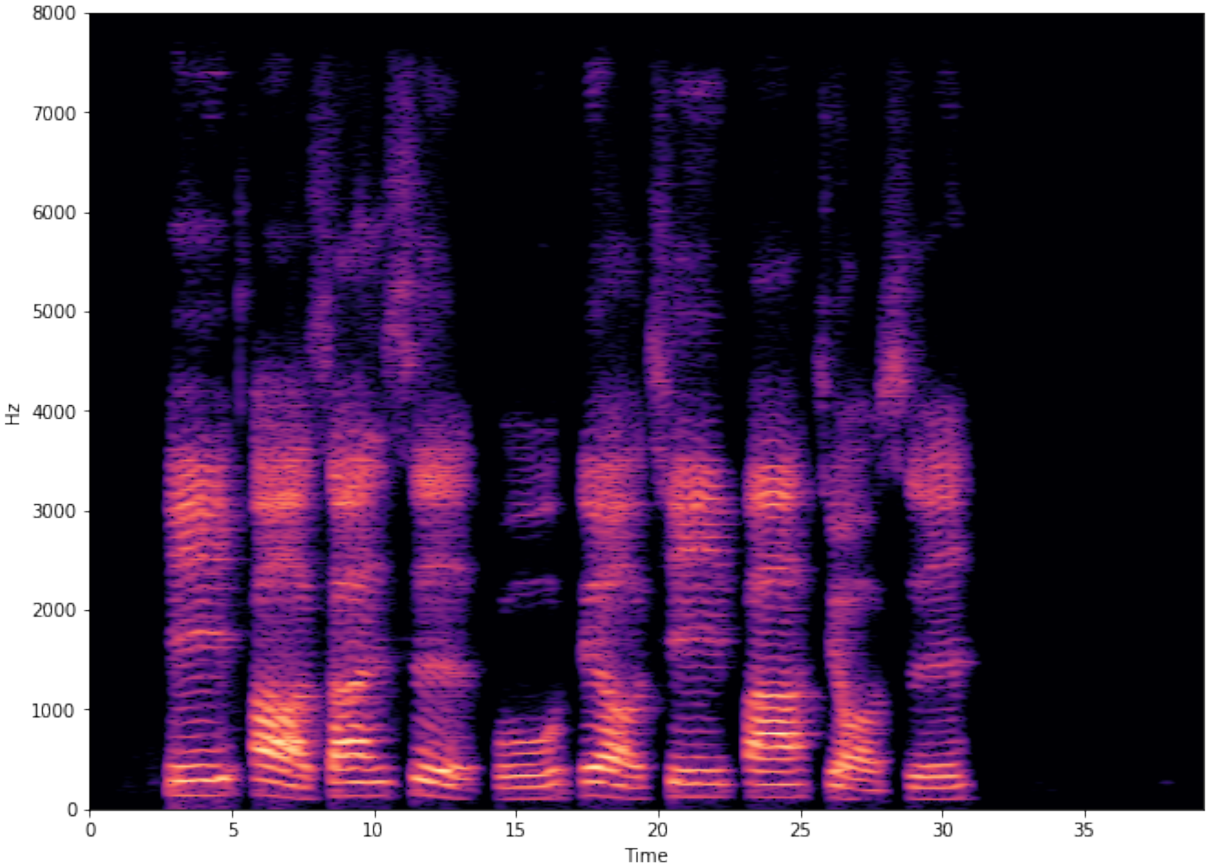
- audio = librosa.resample(audio, orig_sr=sr, target_sr=target_sr)

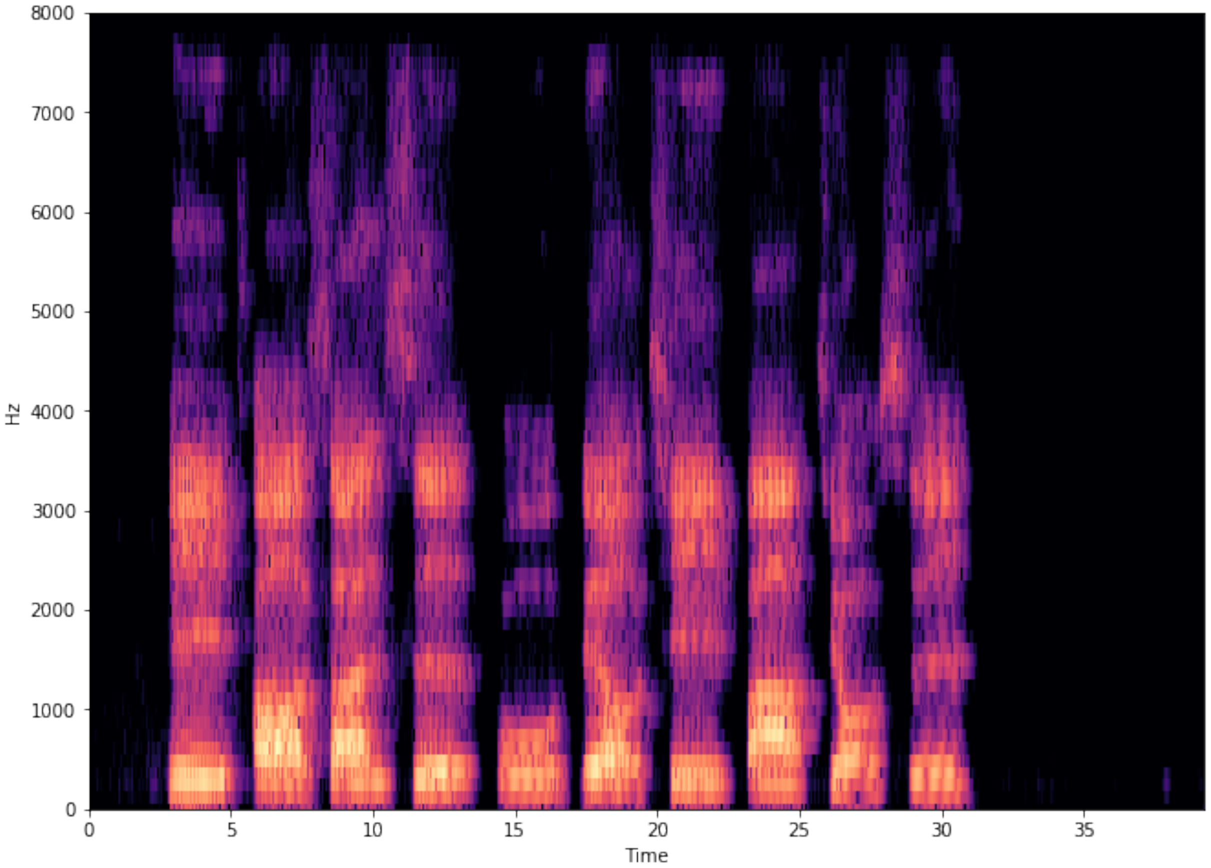




NarrowBand and WideBand Spectrogram

- x = librosa.stft(audio, n_fft=2048, hop_length=80)
- x = librosa.stft(audio, n_fft=128, hop_length=80)



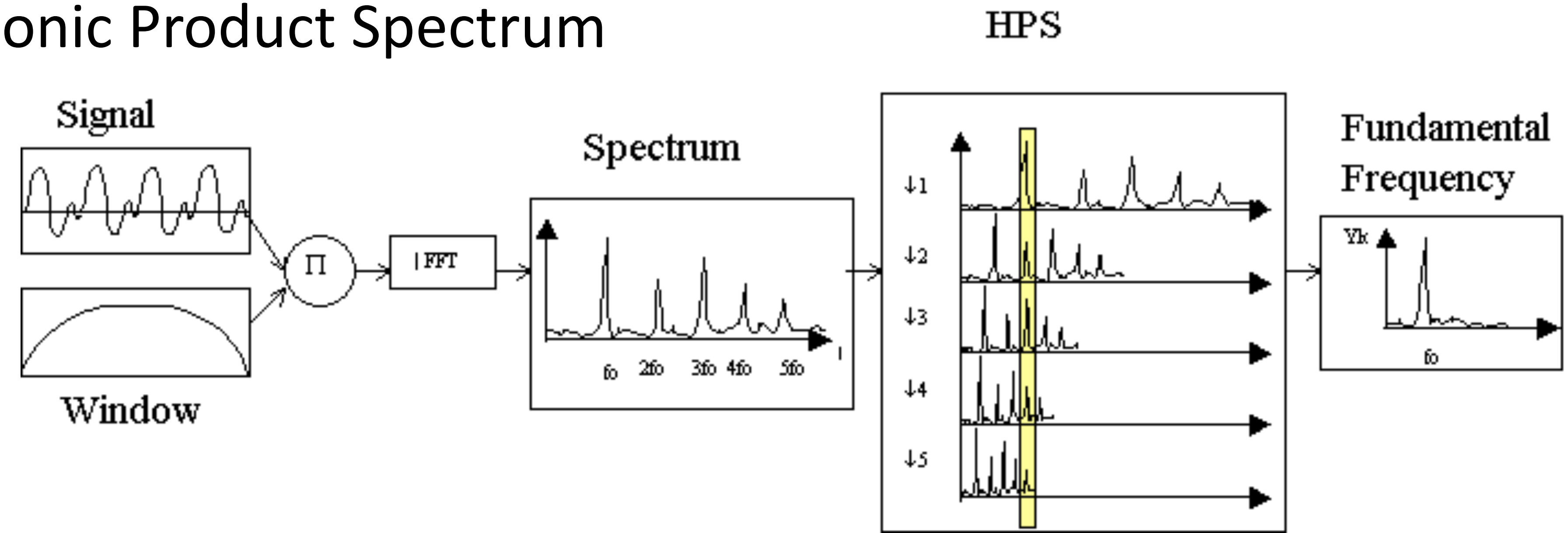


Pitch Detection Algorithms

- Normalized Cross Correlation Function (NCCF)

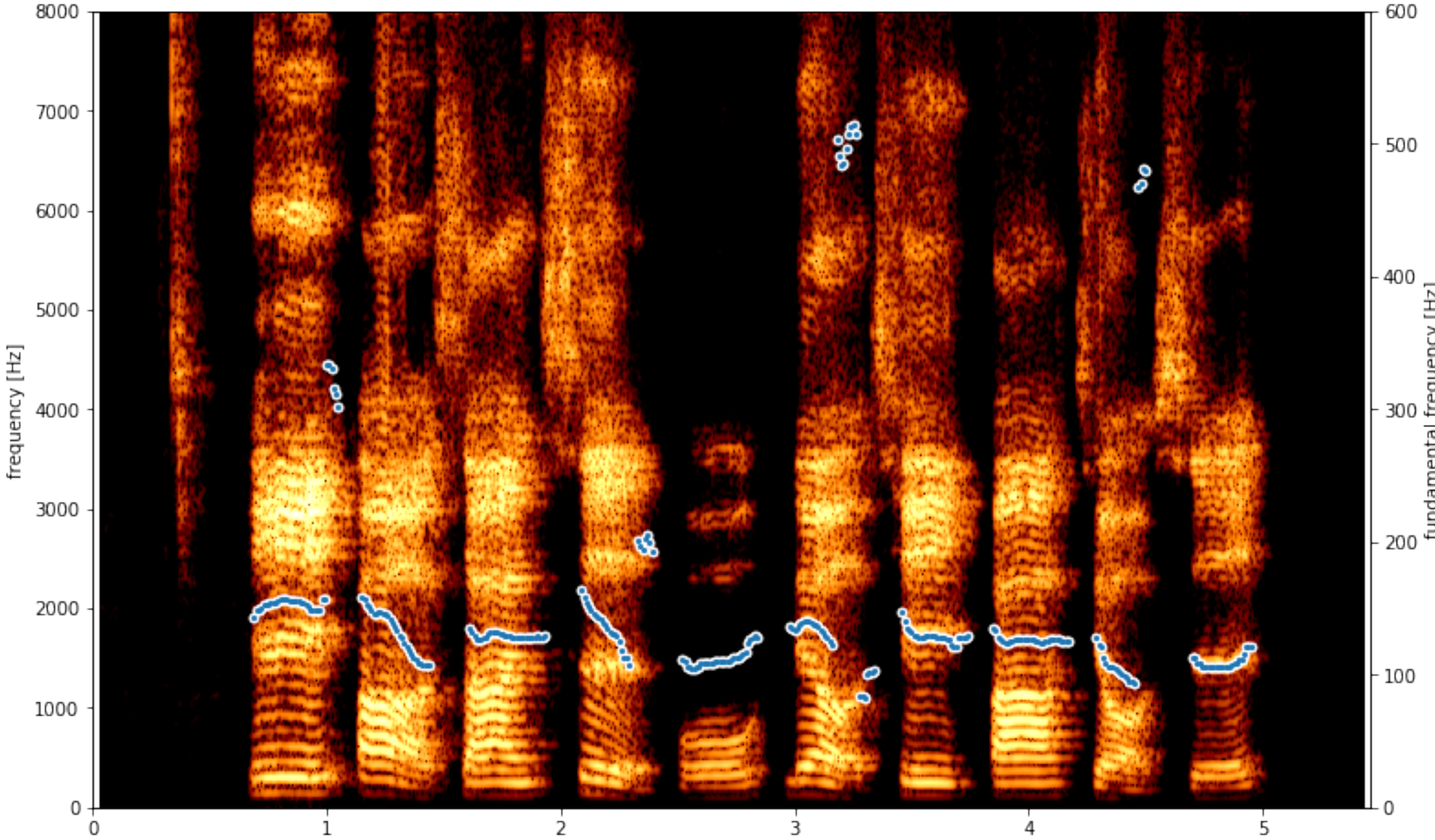
$$NCCF(m) = \frac{\sum_{n=0}^{N-m-1} x(n) \cdot x(n+m)}{\sqrt{\sum_{n=0}^{N-m-1} x^2(n) \cdot \sum_{n=0}^{N-m-1} x^2(n+m)}}, \quad 0 \le m < M_0$$

- Harmonic Product Spectrum

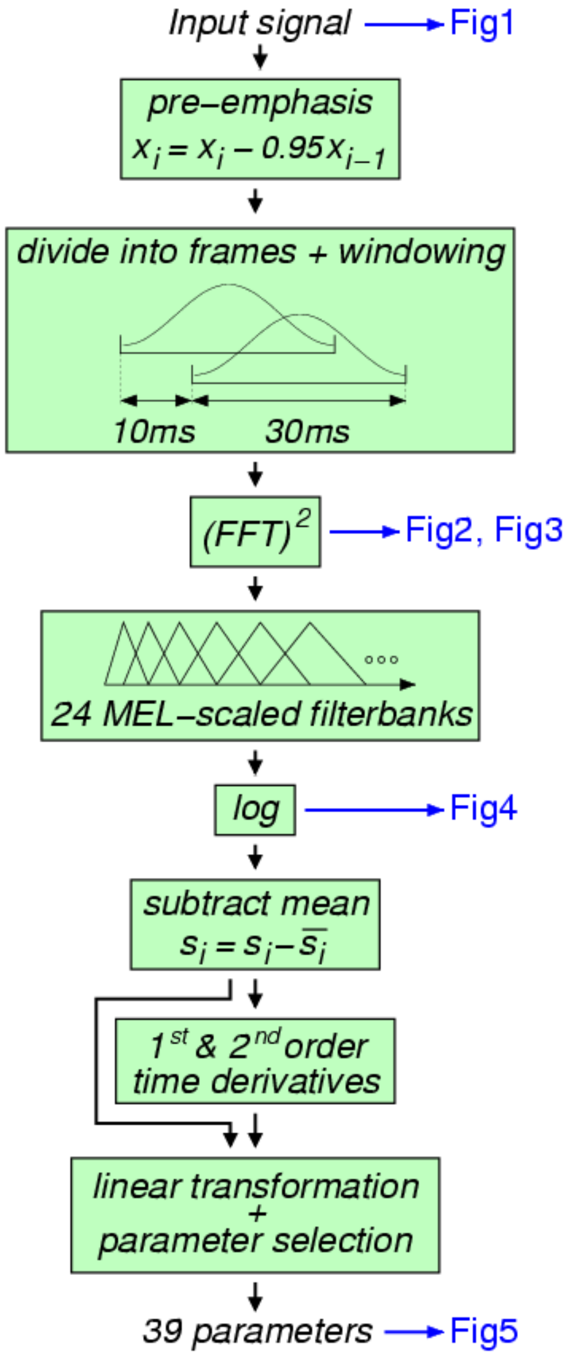


Pitch Contour Extraction

- snd = parselmouth.Sound(human_sound_file)
- snd.resample(new_frequency=16000)
- pitch = snd.to_pitch()



MFCCs



Input signal → Fig1

pre–emphasis
$x_i = x_i - 0.95x_{i-1}$

divide into frames + windowing

10ms 30ms

$(FFT)^2$ → Fig2, Fig3

24 MEL–scaled filterbanks

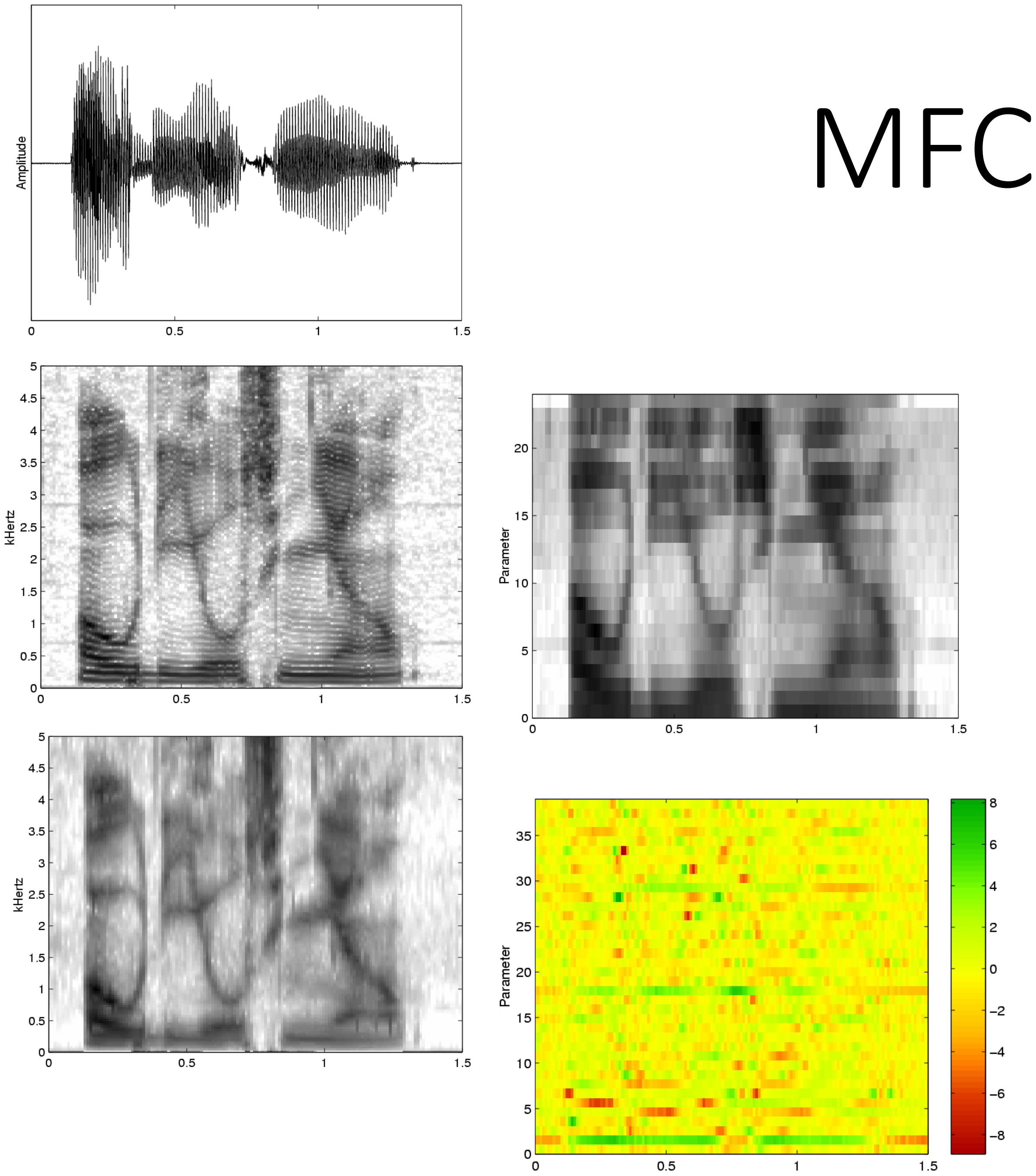
log → Fig4

subtract mean
$s_i = s_i - \bar{s}_i$

1st & 2nd order time derivatives

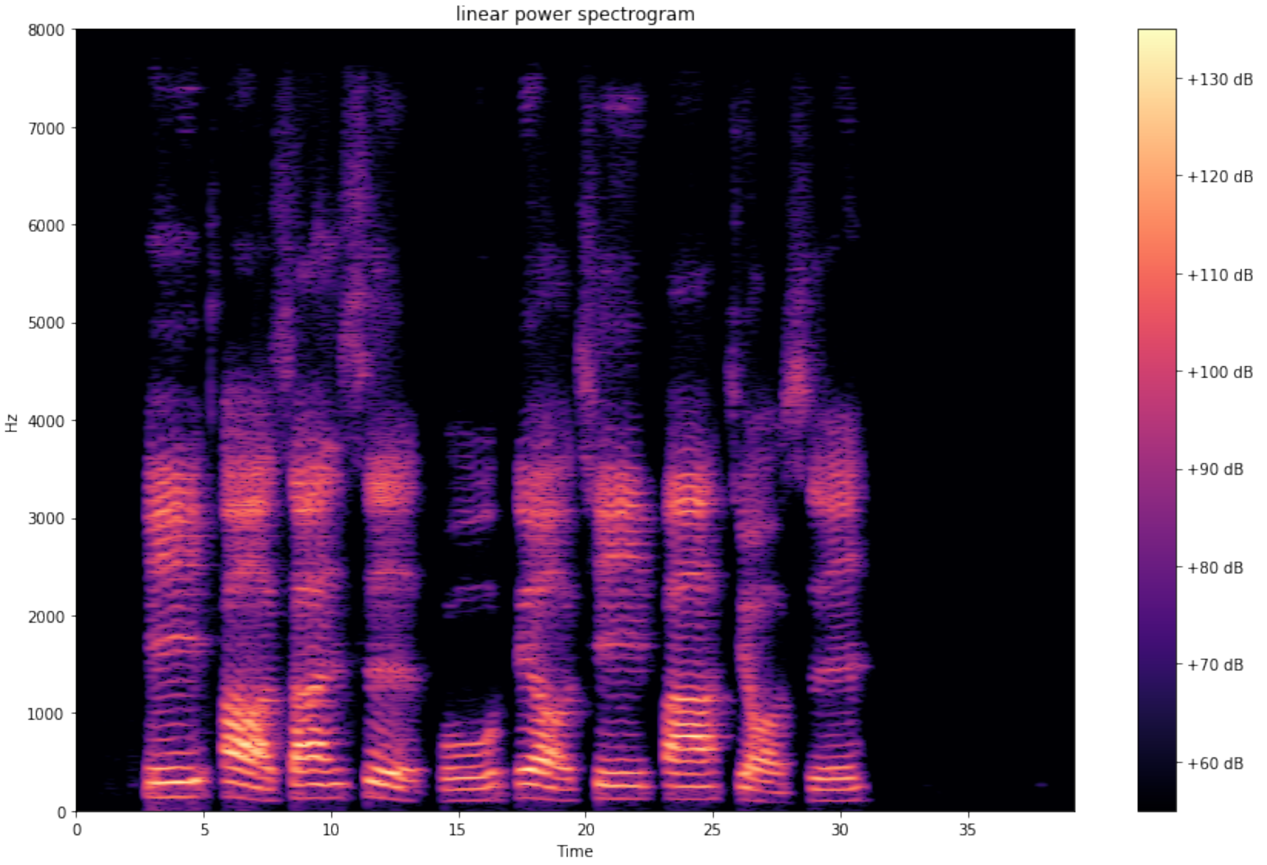
linear transformation + parameter selection

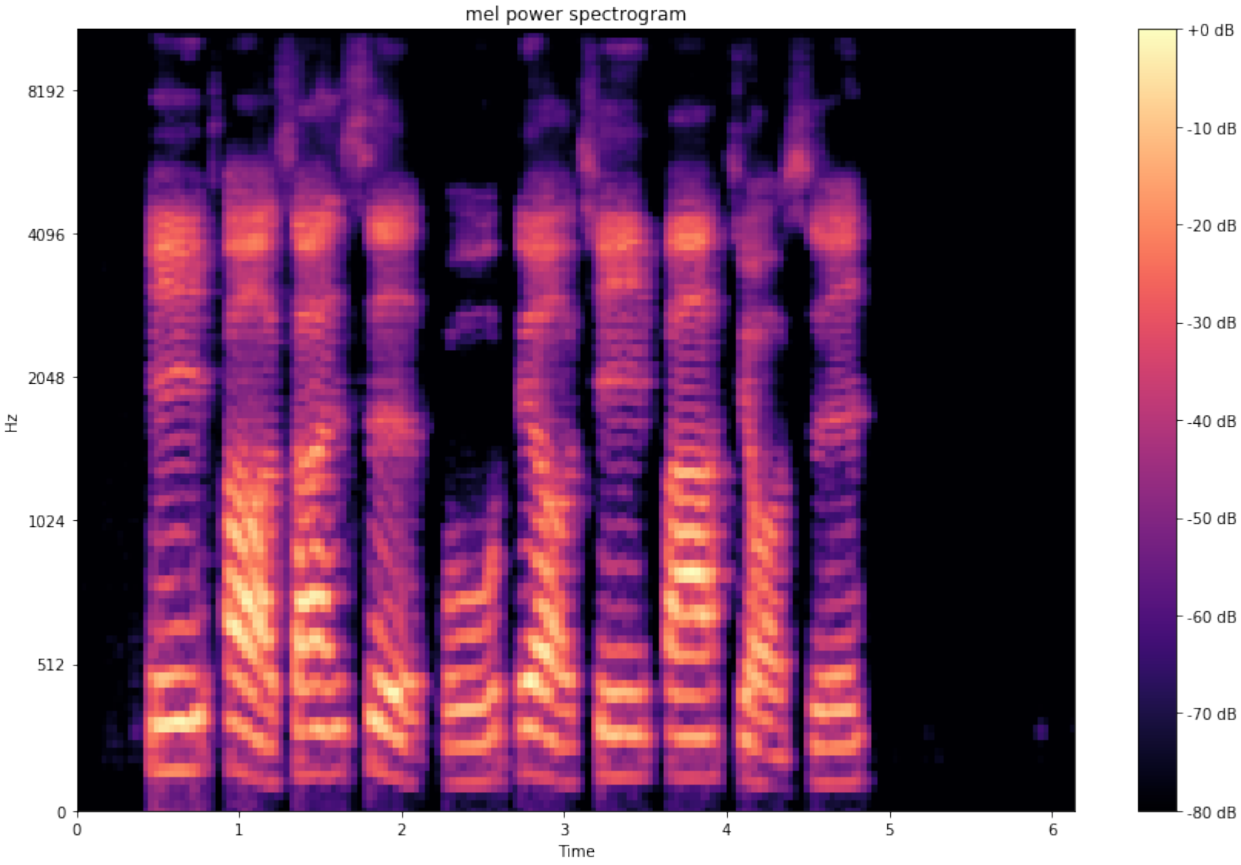
39 parameters → Fig5



Mel-Scaled Spectogram

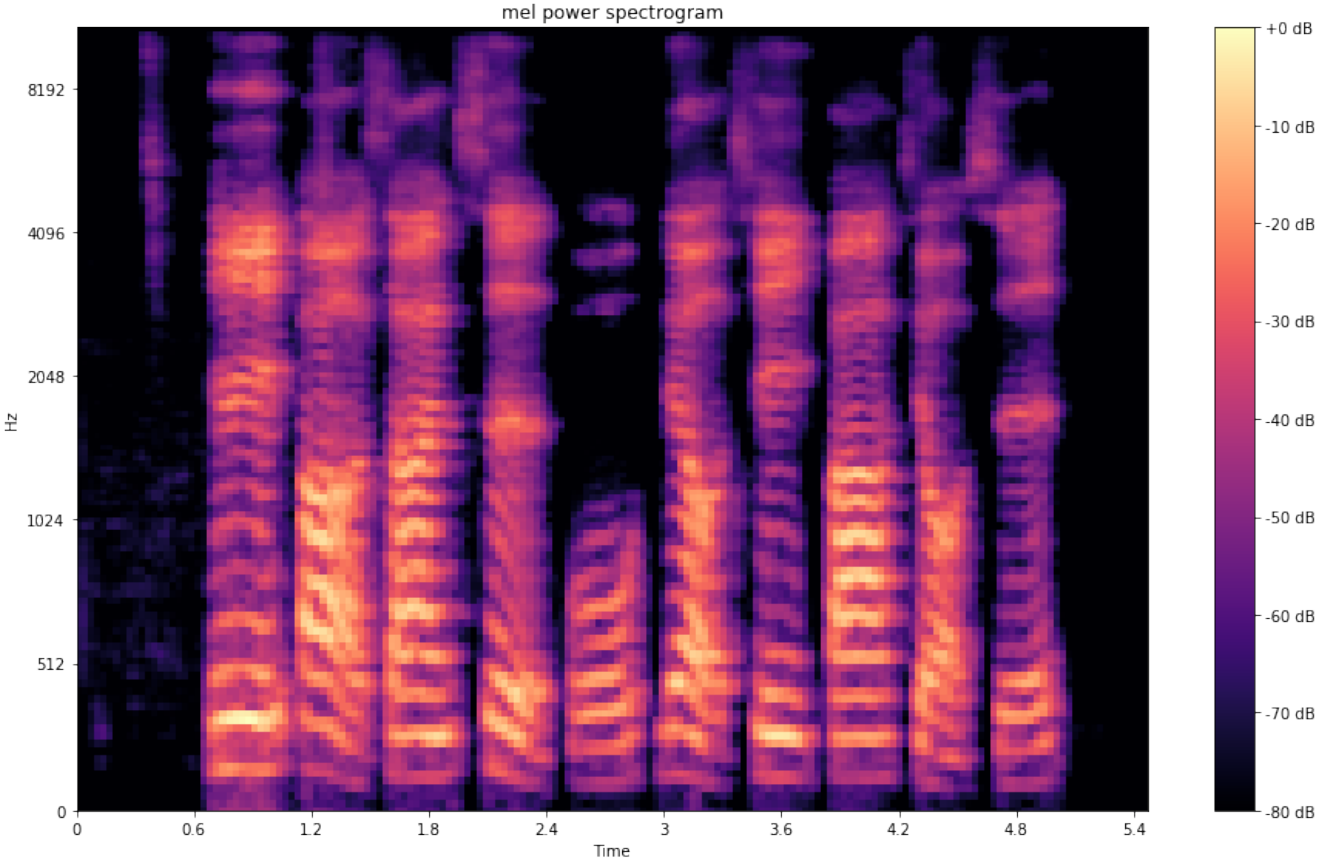
- audio = librosa.resample(audio, orig_sr=sr, target_sr=target_sr)

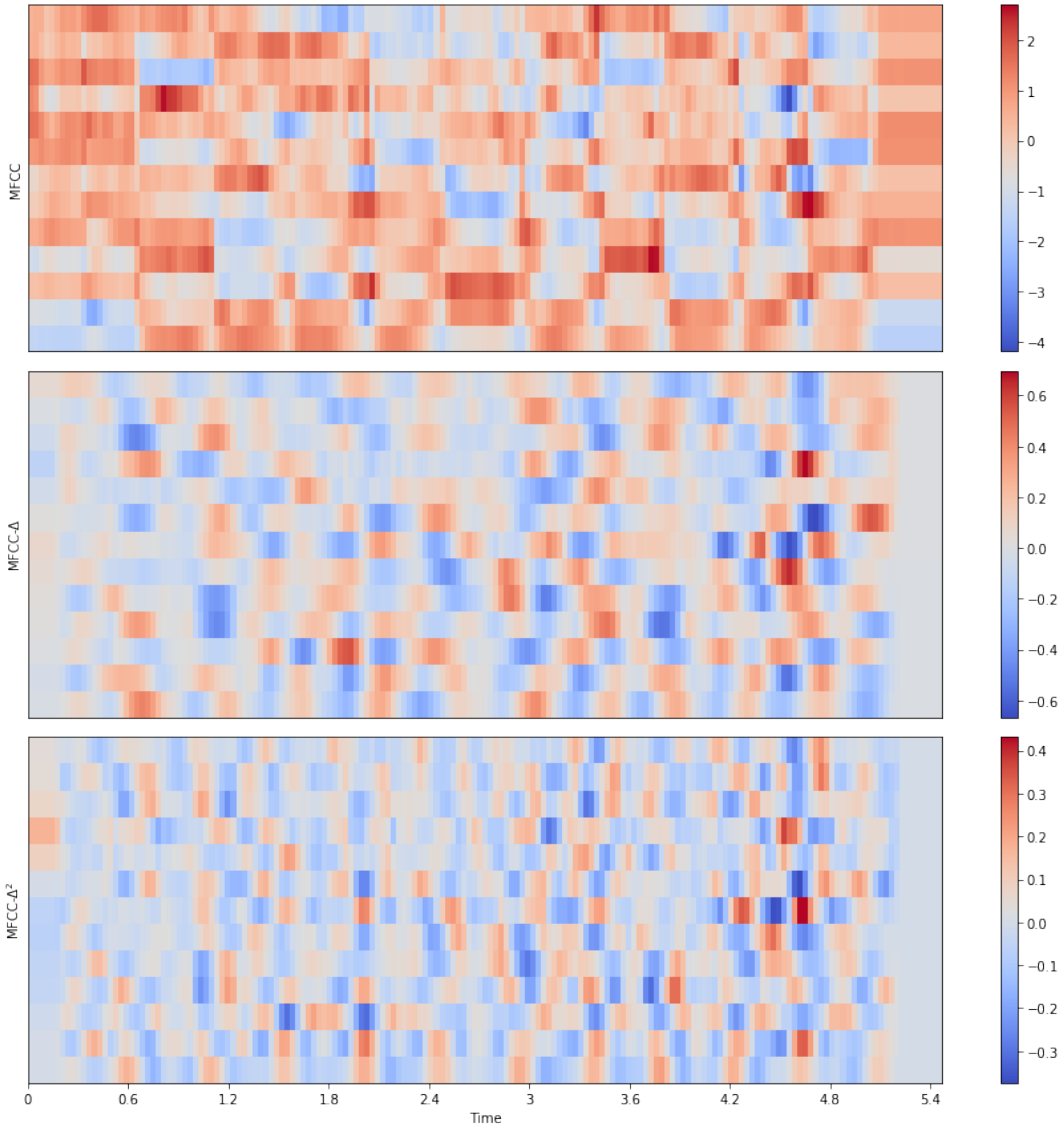




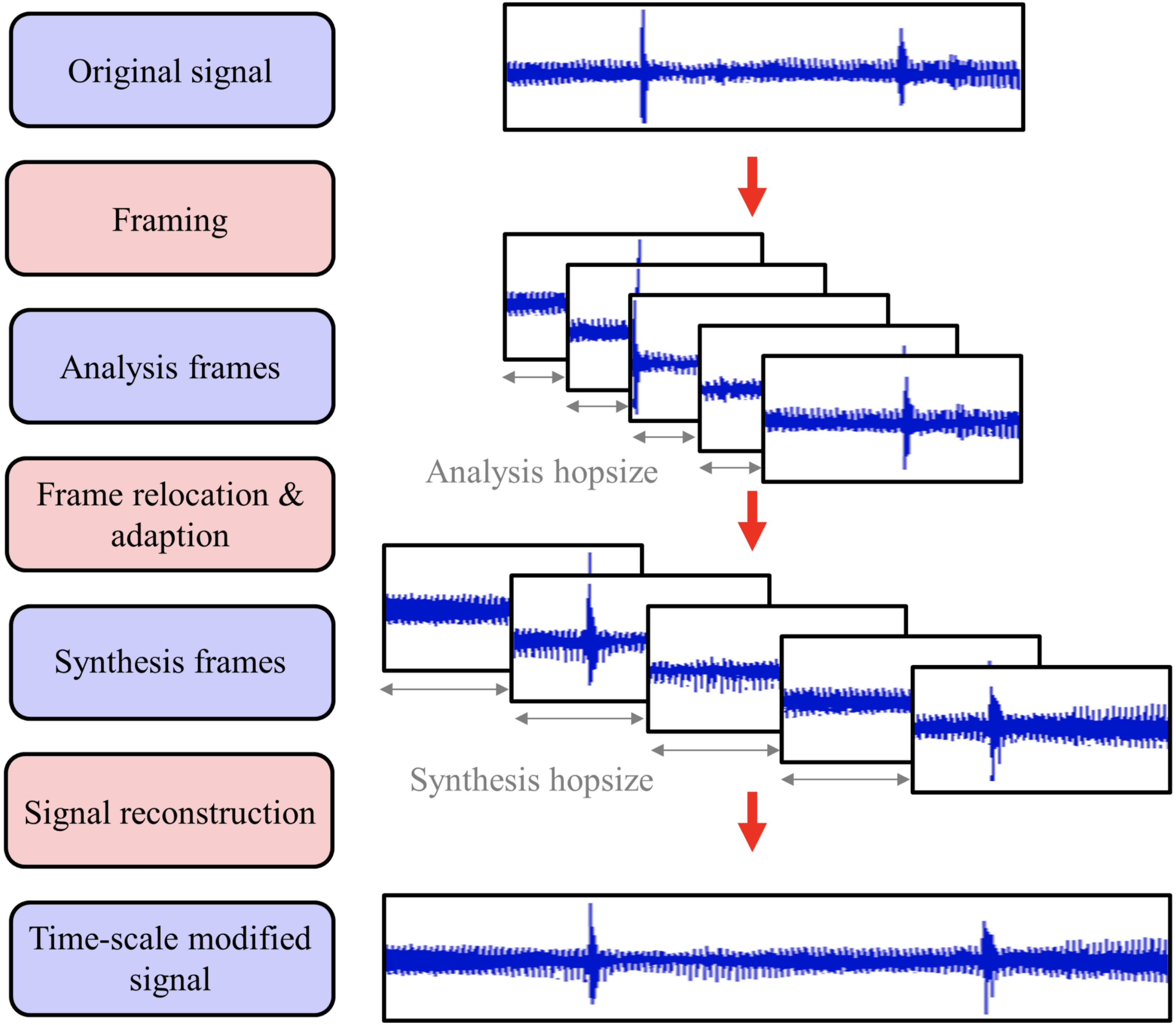
MFCCs

- mfcc = librosa.feature.mfcc(S=log_S, n_mfcc=13)





Audio Time-Stretching and Pitch-Shifting



Original signal

Framing

Analysis frames

Frame relocation & adaption

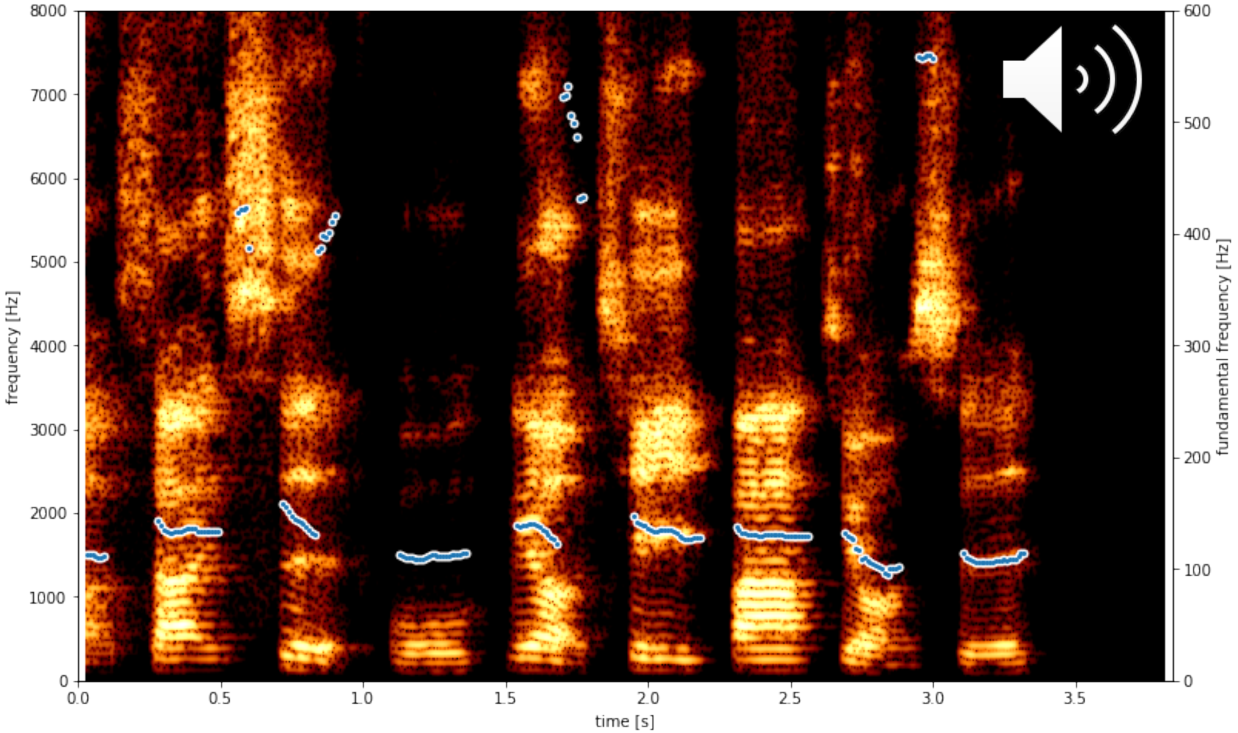
Synthesis frames

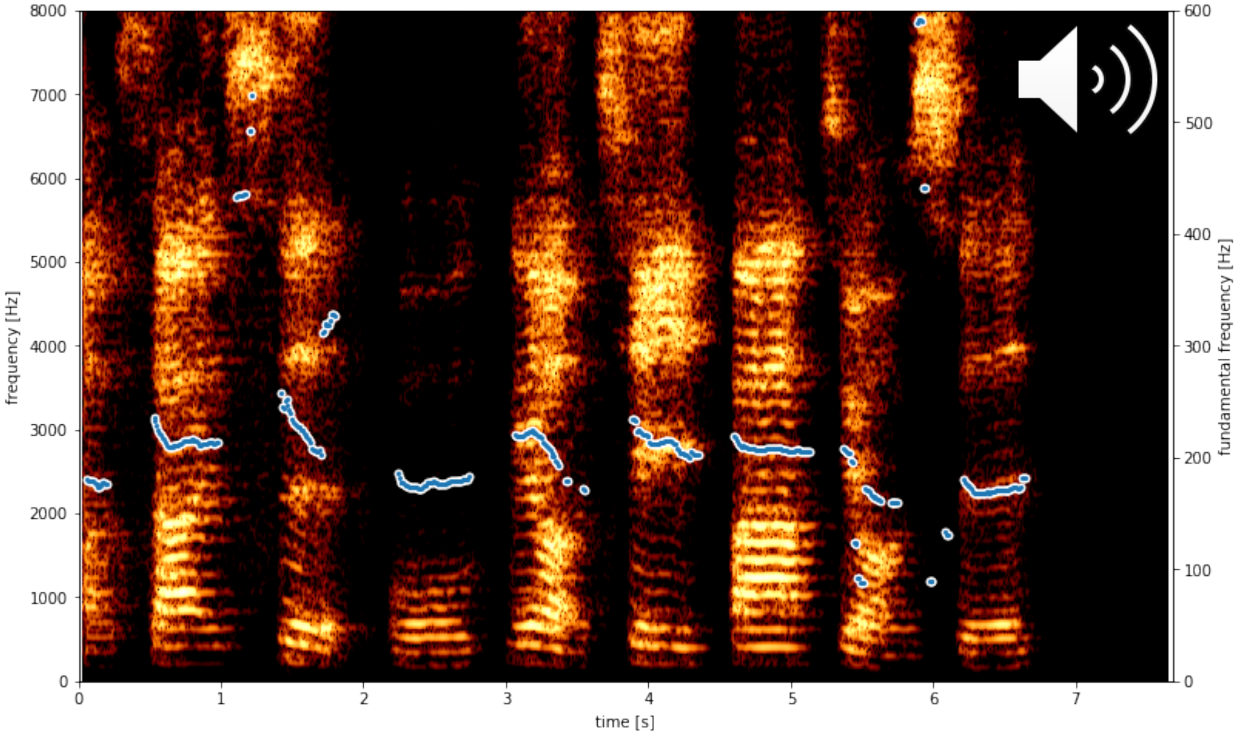
Signal reconstruction

Time-scale modified signal

Audio Time-Stretching and Pitch-Shifting

- time = 2.0
- pitch = 8.0
- !rubberband -t $time -p $pitch $human_sound_file output.wav


frequency [Hz]
fundamental frequency [Hz]
time [s]


frequency [Hz]
fundamental frequency [Hz]
time [s]