



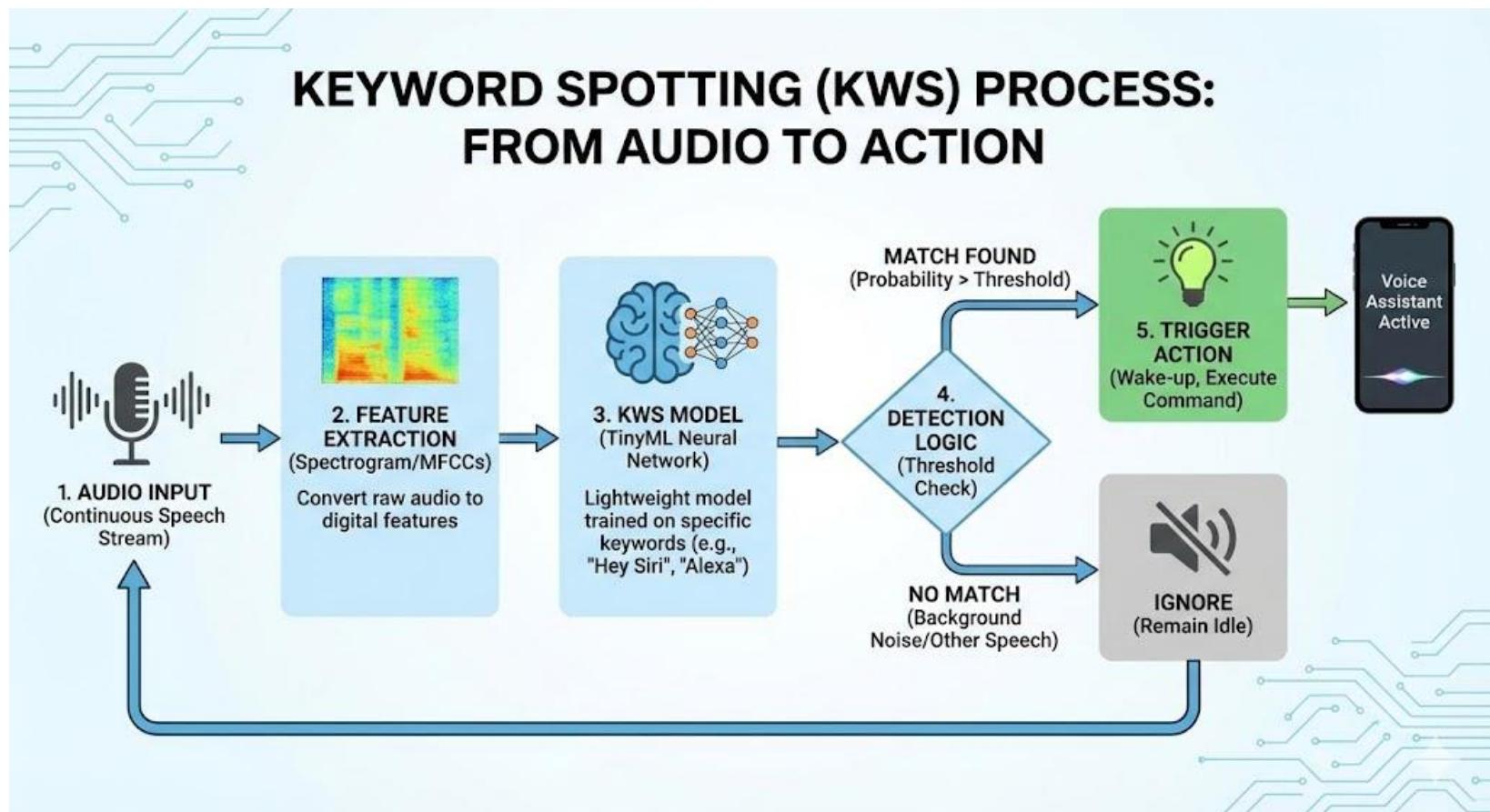
Audio – Key Word Spotting

What is KeyWord Spotting (KWS)

Embedded ML



- **Keyword Spotting (KWS)** is a subfield of speech recognition that focuses on detecting specific, pre-defined words or phrases within a continuous stream of audio.





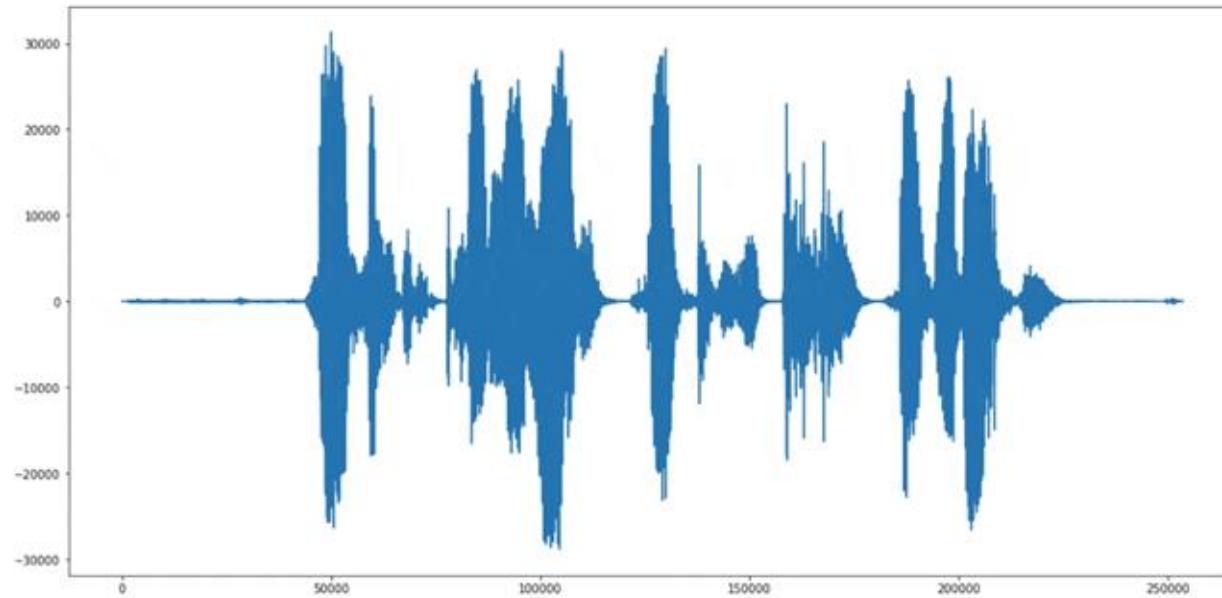
Keyword Spotting vs. General Speech Recognition

- **Keyword spotting** (KWS) is one of the most successful examples of **TinyML**
 - Low-power, continuous, on-device
- General Automatic Speech Recognition (**ASR**) still requires **larger, power-hungry models**
 - But it can run on mobile/edge devices (offline dictation on smartphones)

Audio



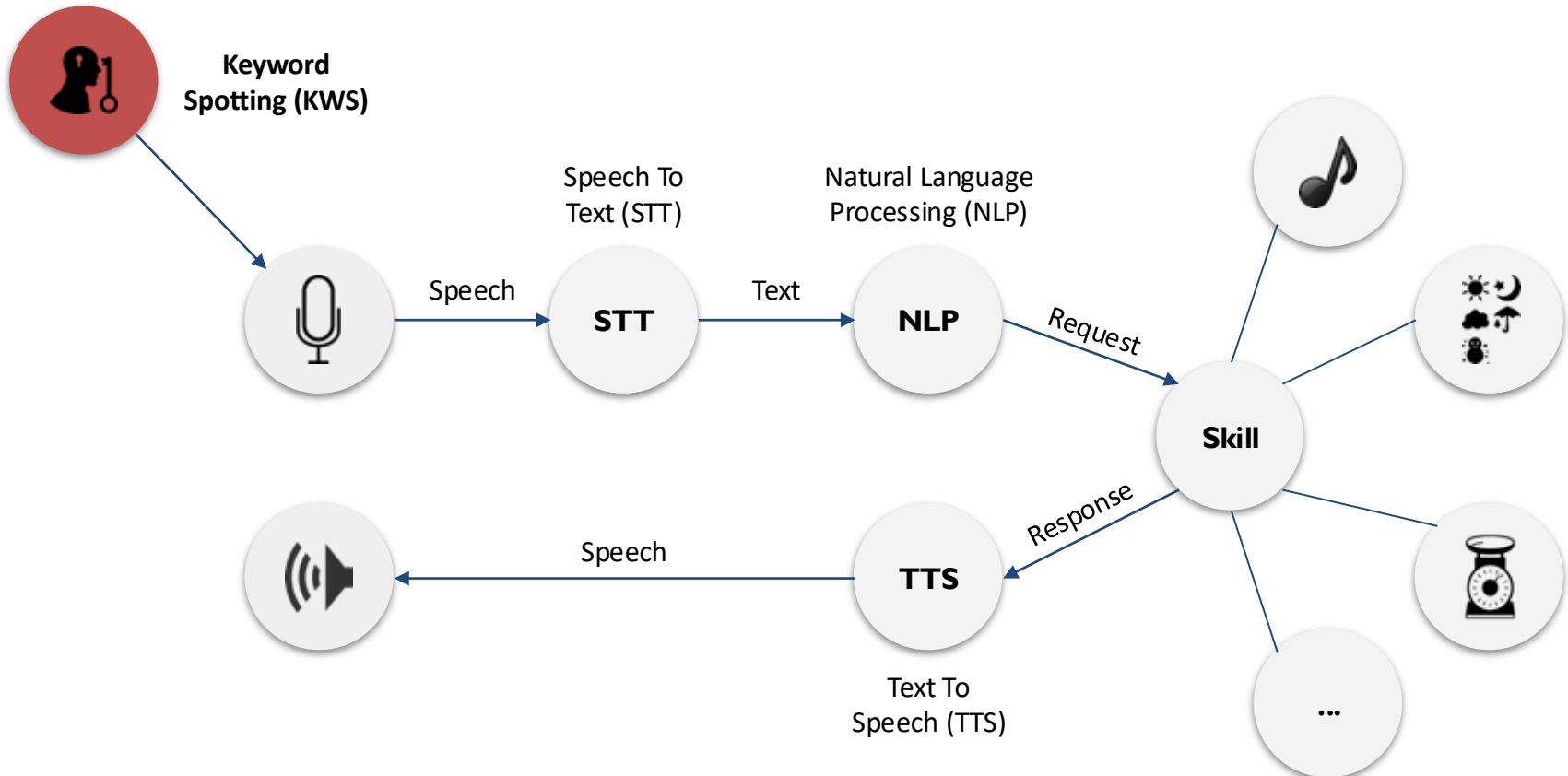
An **audio file** is a digital record of sound. It stores acoustic waves—like voice or music—as a series of binary numbers (0s and 1s)



WAV (Waveform Audio File Format) - Pulse Code Modulation (PCM) data – convert smooth signals into digital format (staircase)
MP3 (MPEG-I Audio Layer III) – **lossy compressed** format. It uses algorithms to throw away sounds the human ear (psychoacoustics) cannot hear well to save space.

Smart Assistants

Embedded ML





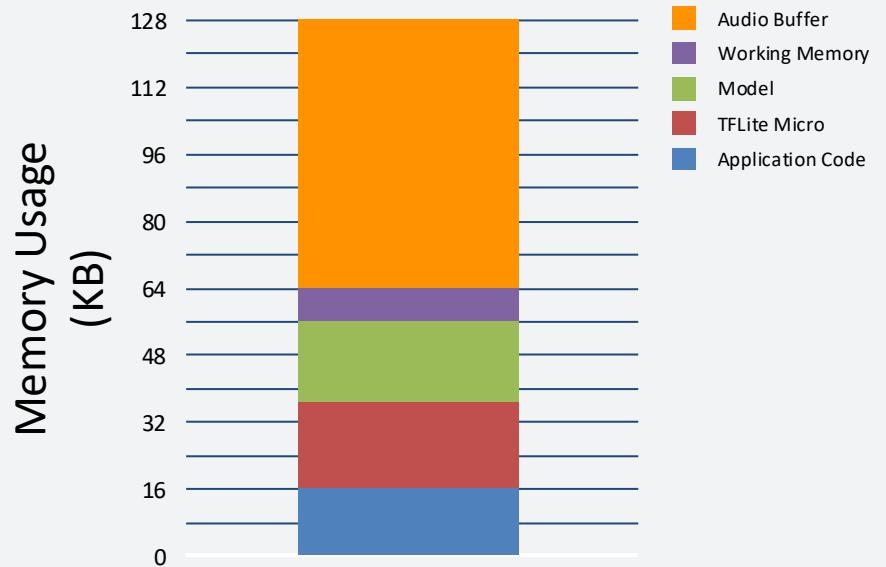
Challenges and Constraints

- Latency - Provide results quickly, respond in real-time to the user
- Bandwidth - Minimize data sent over the network (slow and expensive)
- Accuracy - Listen continuously, but only trigger at the right time
- Personalization - Trigger for the user and not for background noise
- Security & Privacy - Safeguarding the data that is being sent to the cloud
- Battery - Limited energy, operate on coin-cell type batteries
- Memory - Run on resource constrained devices



Memory Usage

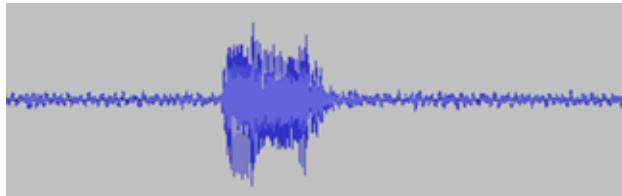
- Need to be **resource aware**
- **Less** compute
- **Less** memory
- Use **quantization**



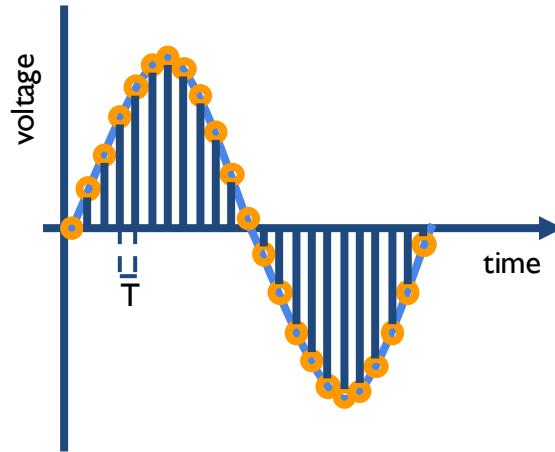
Sample Rate



Normally Signals are recorded @ 44100 Hz (44.1 KHz)



- Sample rate
- Bit depth
- Length



Sampling period (T): 62.5 μ s
Sampling rate (f_s): $1/62.5 \mu\text{s} = 16 \text{ kHz}$

1 second of audio at 16 kHz and 16 bits = 16,000 samples (16 bits each)

Bit Depth (The Y-Axis / Amplitude) 16 bits
- KWS standard - The precision of the measurement at each sample point.

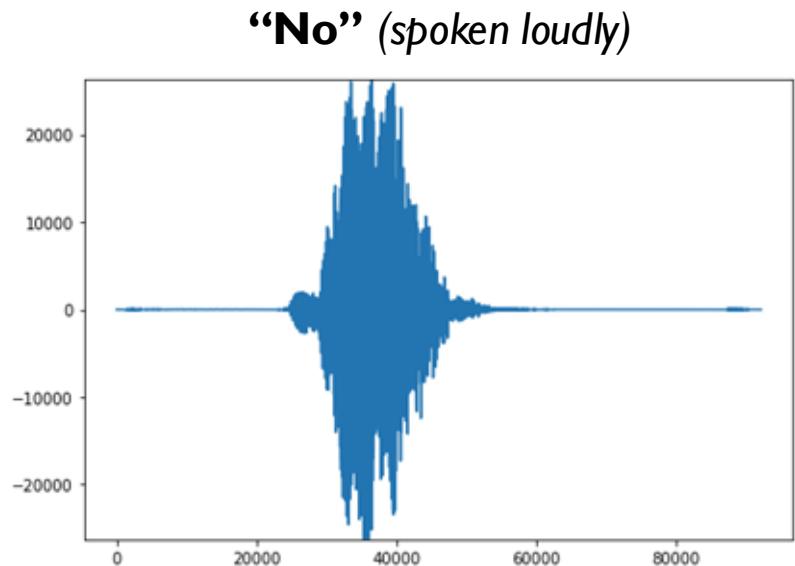


The Speech Commands Dataset

- Recorded as individual **words** not sentences
- 1000-4000 examples of each word
- >2,500 volunteers
- Representative of **real world audio** and includes background noise as well
- **25 “IoT keywords” + 10 “unknown words”** (with phonetic similarities:
“three” vs “tree”)

What are interesting challenges?

- It is a continuous signal, so **when does the word start?**
- How do you “align” on the starting point?
- How do we **extract the vital parts** of the signal that matter?



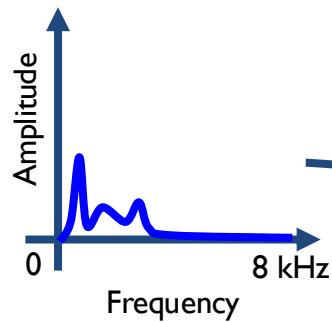


Fourier Transform

1 second audio sample (“hello”)

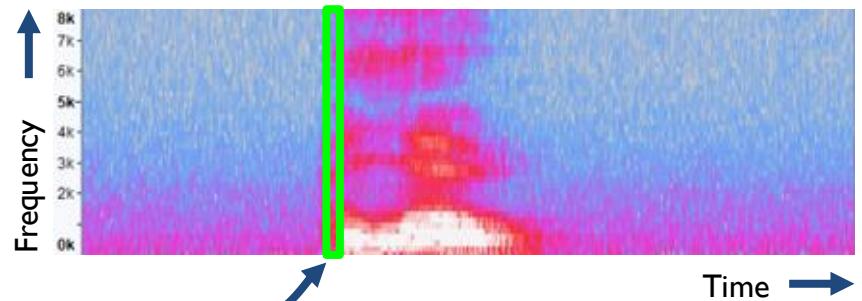


Fast Fourier Transform (FFT)



Voice frequency range: 300 - 3400 Hz

Spectrogram

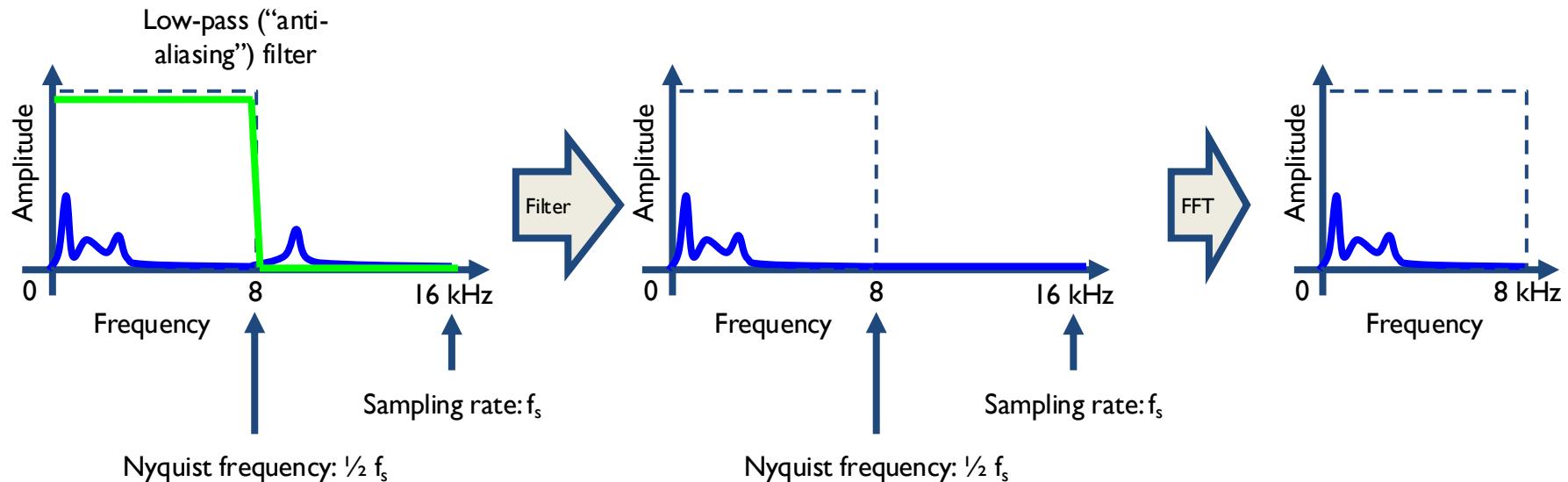




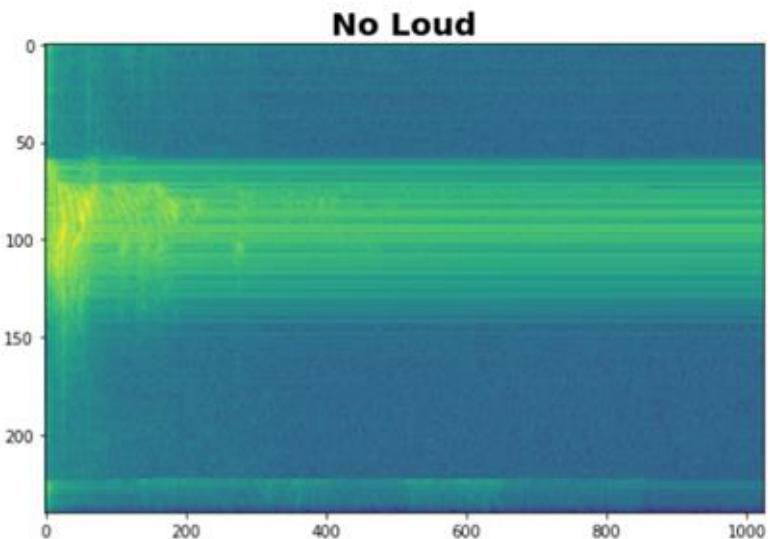
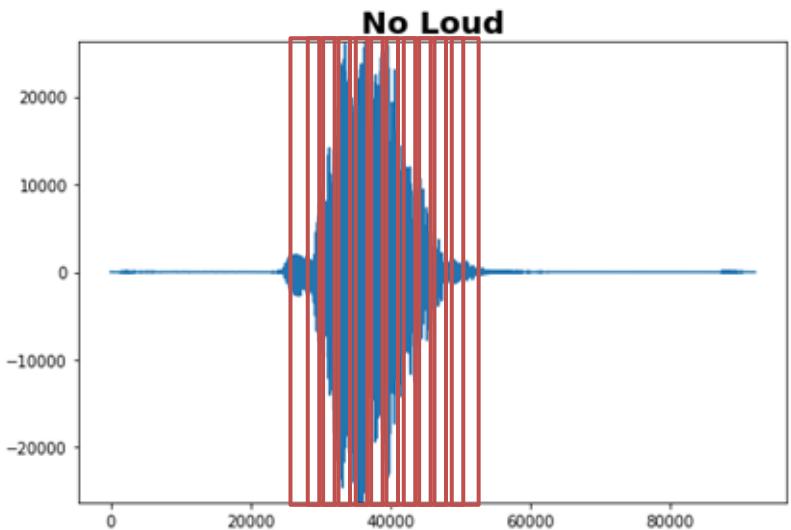
Nyquist-Shannon Sampling Theorem

$$f_s > 2B$$

f_s is the sampling frequency (Hz)
 B is the highest frequency component (Hz)

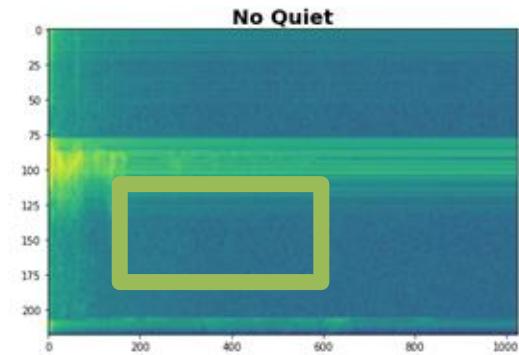
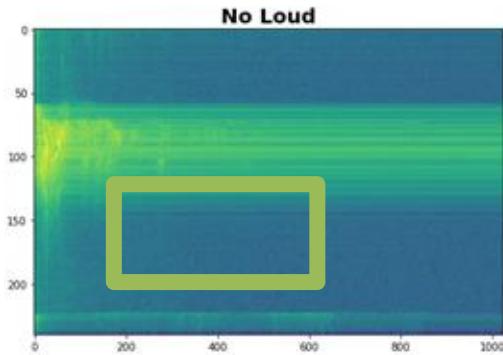
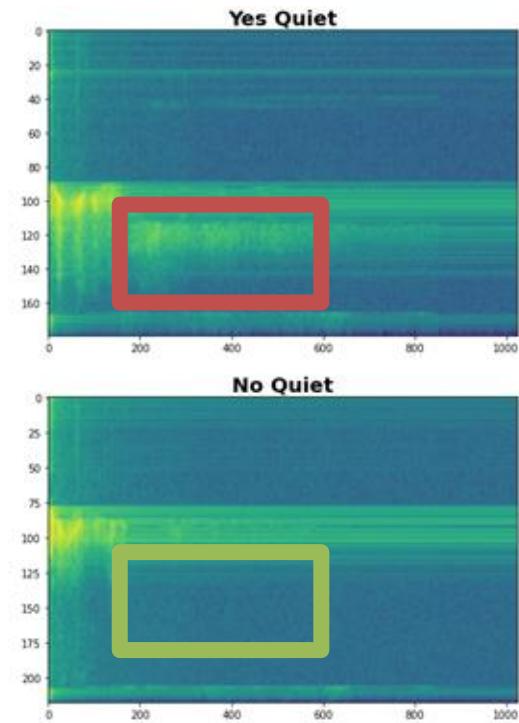
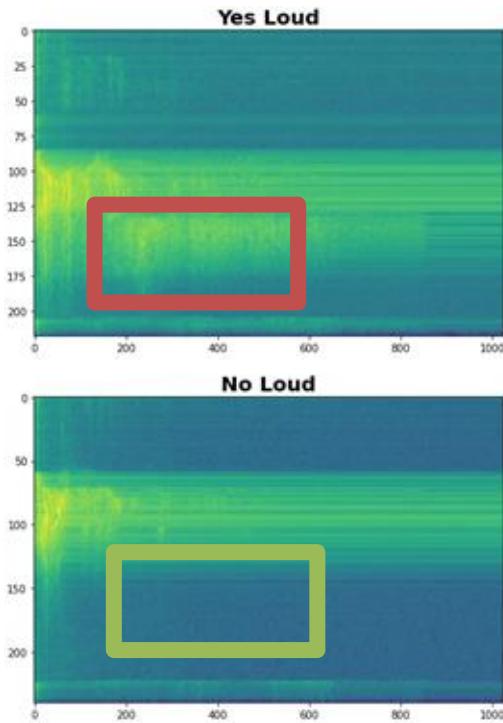


Data Preprocessing: Spectrograms





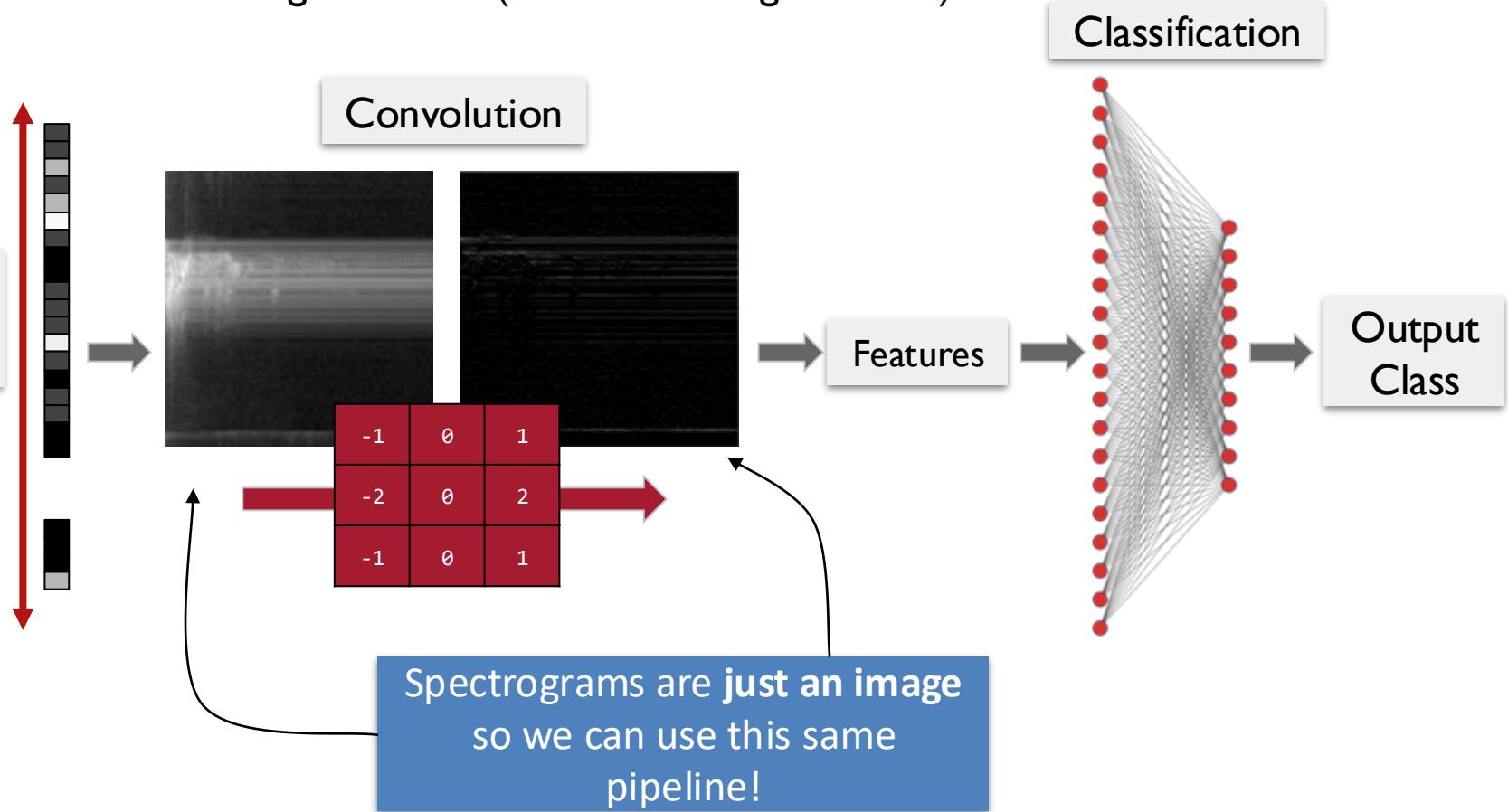
Data Preprocessing: Spectrograms



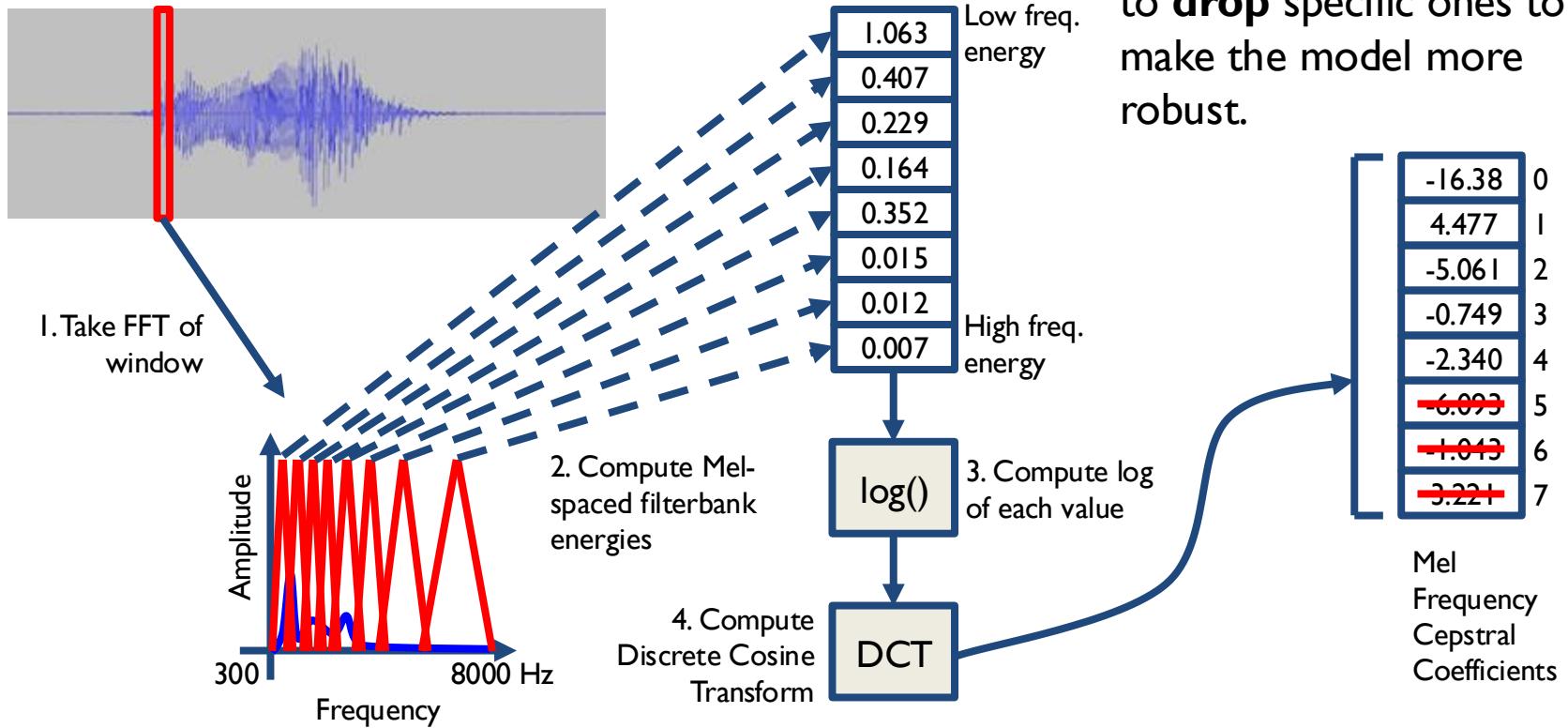


Spectrogram for Keyword Spotting

- **CNNs love images:** Convolutional Neural Networks (CNNs) treat audio features like an image. They look for "edges" and "shapes".
- Produces larger models (suitable for edge devices)

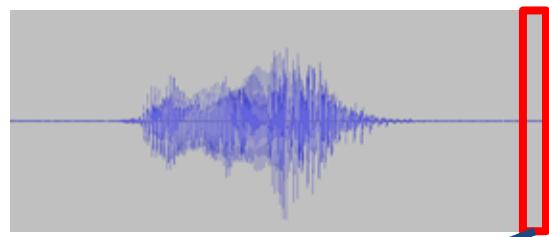


Mel Frequency Cepstral Coefficients (MFCCs)



MFCC mimics the Human Ear (Mel Scale): Humans are great at distinguishing low frequencies (bass) but bad at distinguishing high frequencies. MFCCs allocate more data to low frequencies and less to high ones, matching our biology.

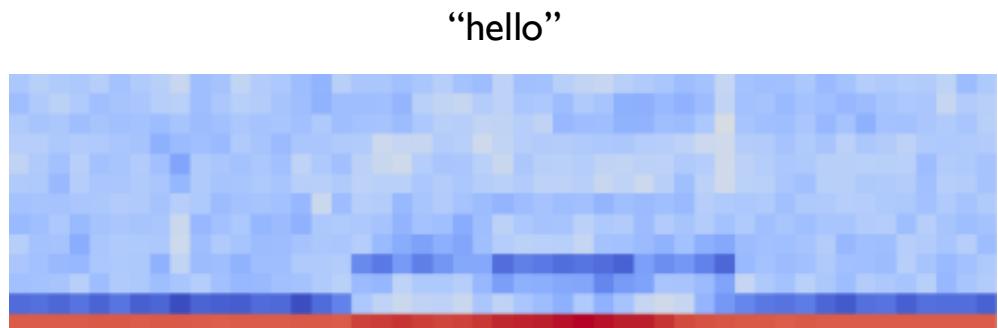
Mel Frequency Cepstral Coefficients (MFCCs)



MFCCs

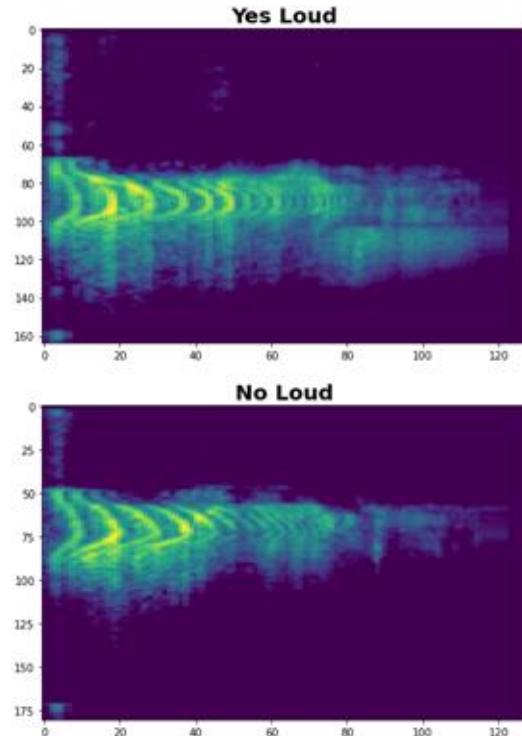
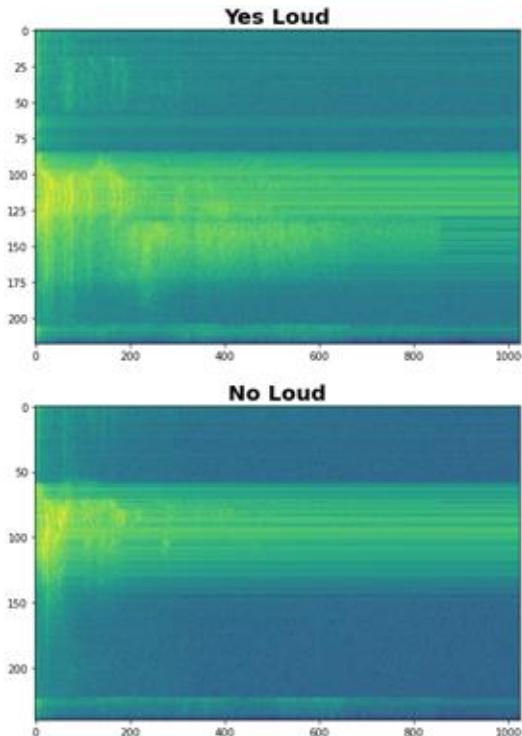
I2	-1.043	-0.816	...	-0.184
:	:	:	...	:
3	0.5467	0.442	...	-0.523
2	0.0476	0.836	...	0.185
1	0.153	-0.671	...	-0.248
0	-1.173	0.462	...	-1.218

0 I 48





Spectrograms v. MFCCs



MFCC breaks the image: The DCT step in MFCCs scrambles the spatial relationship of frequencies. It destroys the "image," making it harder for a CNN to find those shapes.

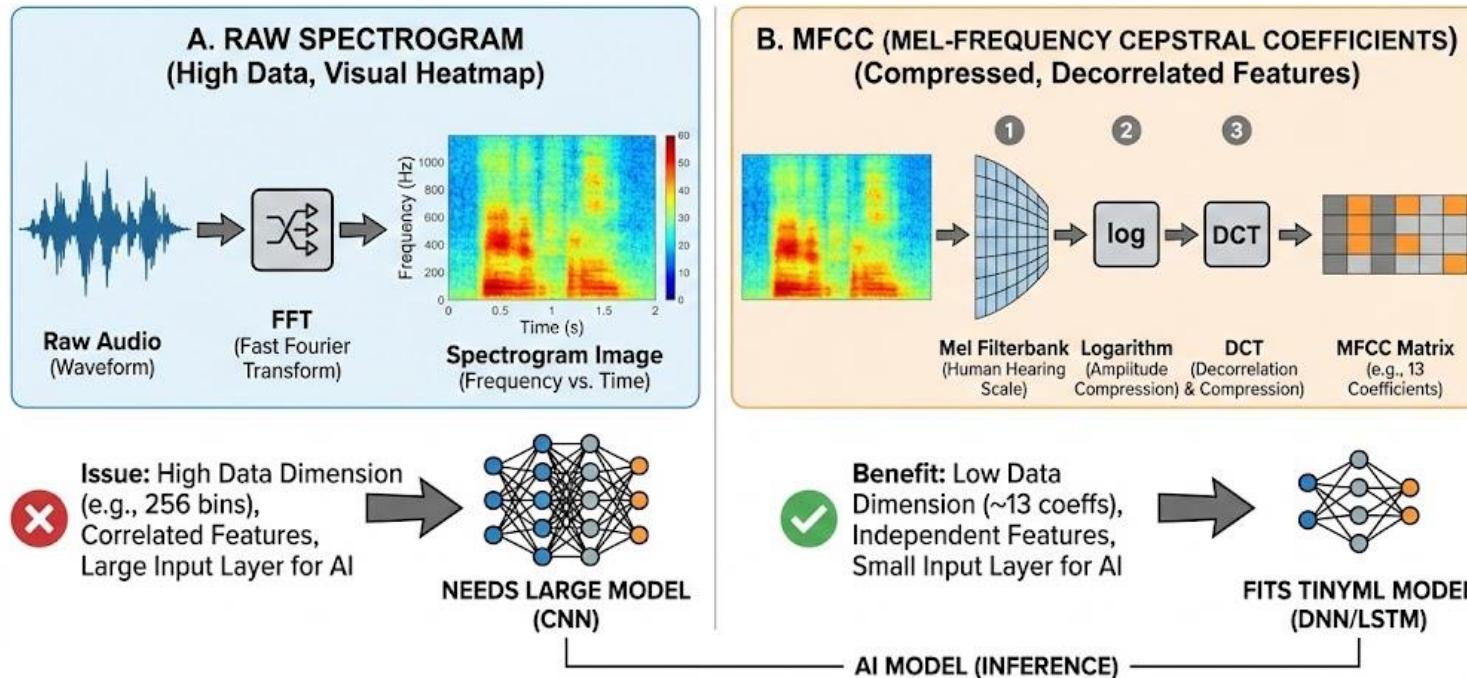
Spectrograms v. MFCCs

Embedded ML



- can use spectrograms (specifically Log-Mel Spectrograms), and for modern Convolutional Neural Networks (CNNs), they are often better.
- **MFCCs** became the standard for Keyword Spotting (especially in ultra-low-power systems),
- comes down to two main factors: **Input Size** and **Feature Independence**.

SPECTROGRAM vs. MFCC for KEYWORD SPOTTING (KWS): FROM RAW SOUND TO EFFICIENT FEATURES

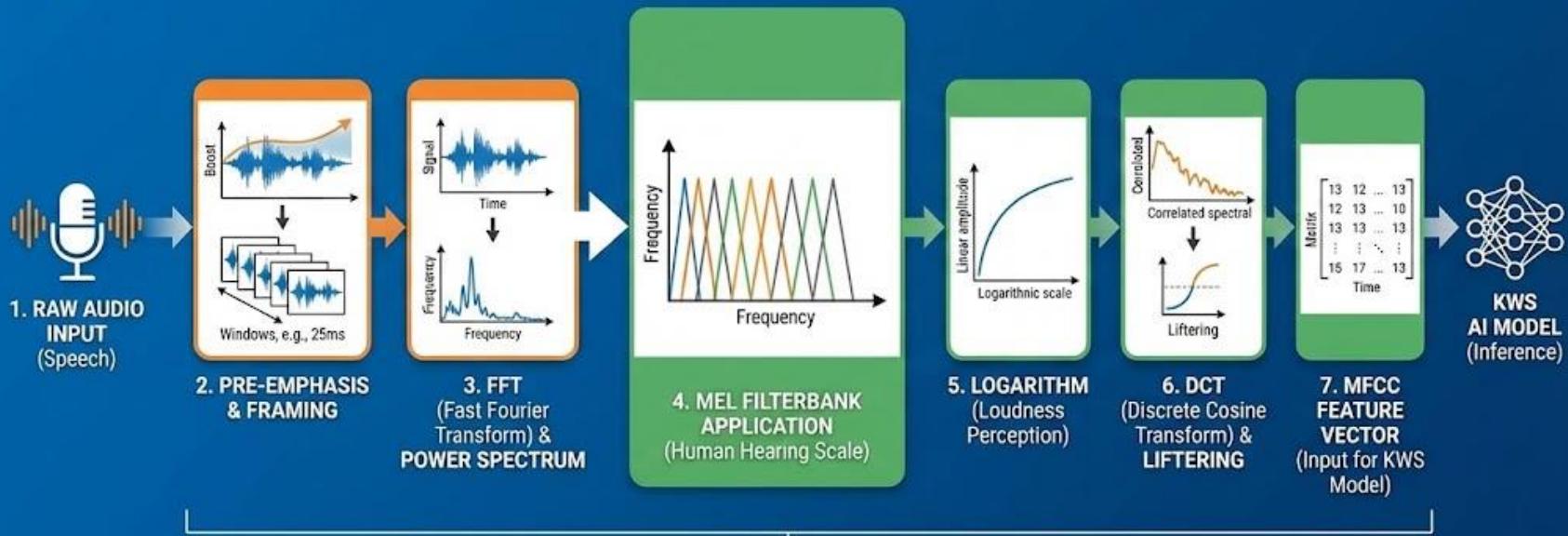


MFCC for Audio Inference - KWS

Embedded ML



MFCC PROCESS for KEYWORD SPOTTING (KWS): FROM AUDIO TO AI FEATURES



Generates compact, decorrelated features that mimic human hearing for efficient KWS.



HandsOn Session

Embedded ML



- Explore IMU interface in RPI Zero W2
- We'll test the microphones in the M5Core2 and RPI Zero W2
- Record the audio in M5Core2 and RPI Zero W2
- Analyze the audio file – Spectrogram and MFCC coefficients.