



# Sound and KWS



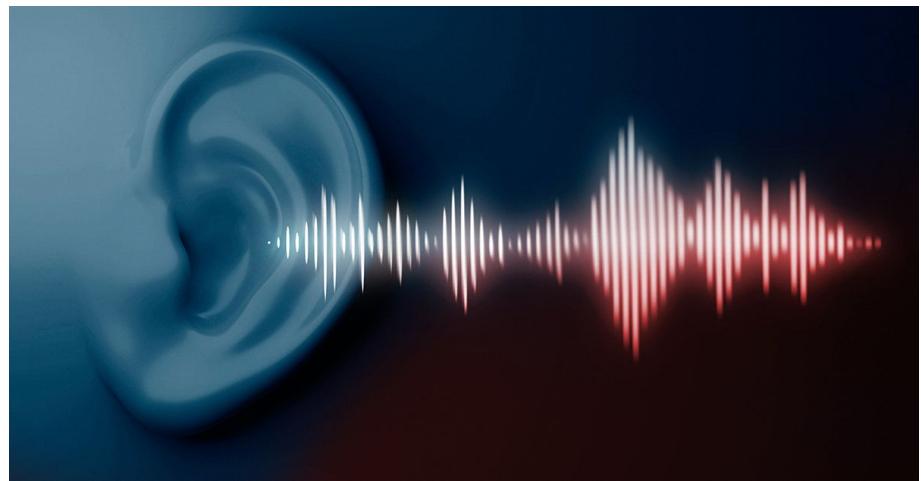
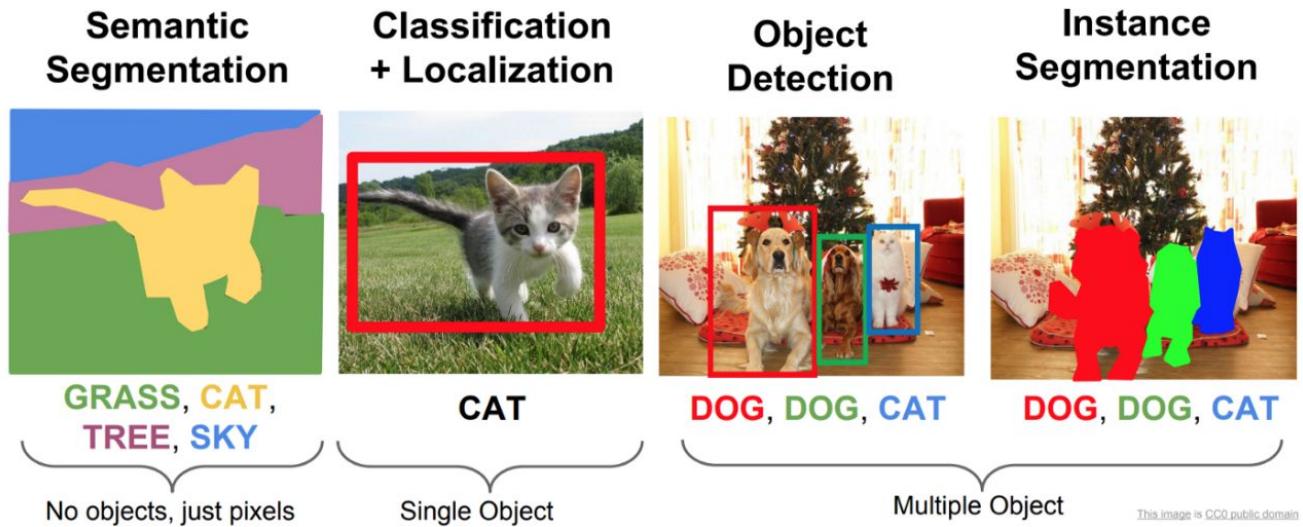
Prof. Jesús Alfonso López  
[jalopez@uao.edu.co](mailto:jalopez@uao.edu.co)

Universidad Autónoma de  
Occidente

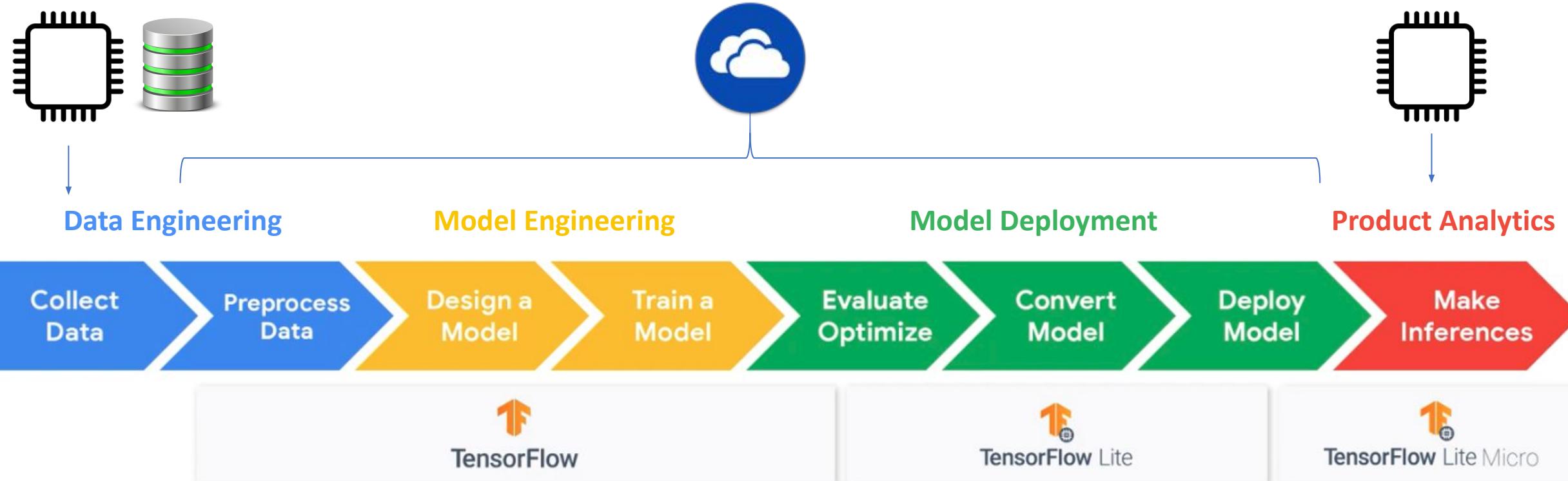


Images are Ok but, What  
About Sound?

# Unstructured Data

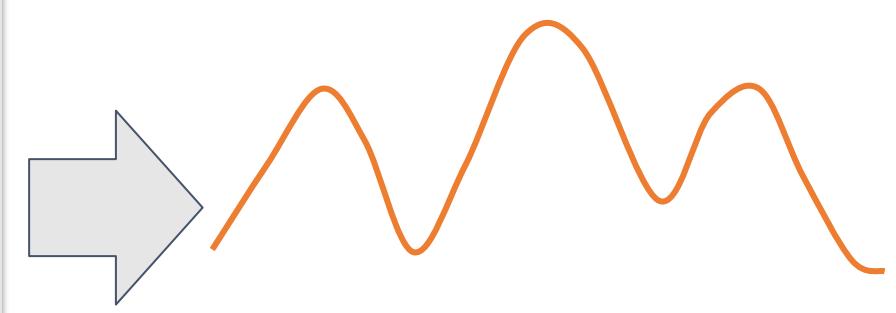
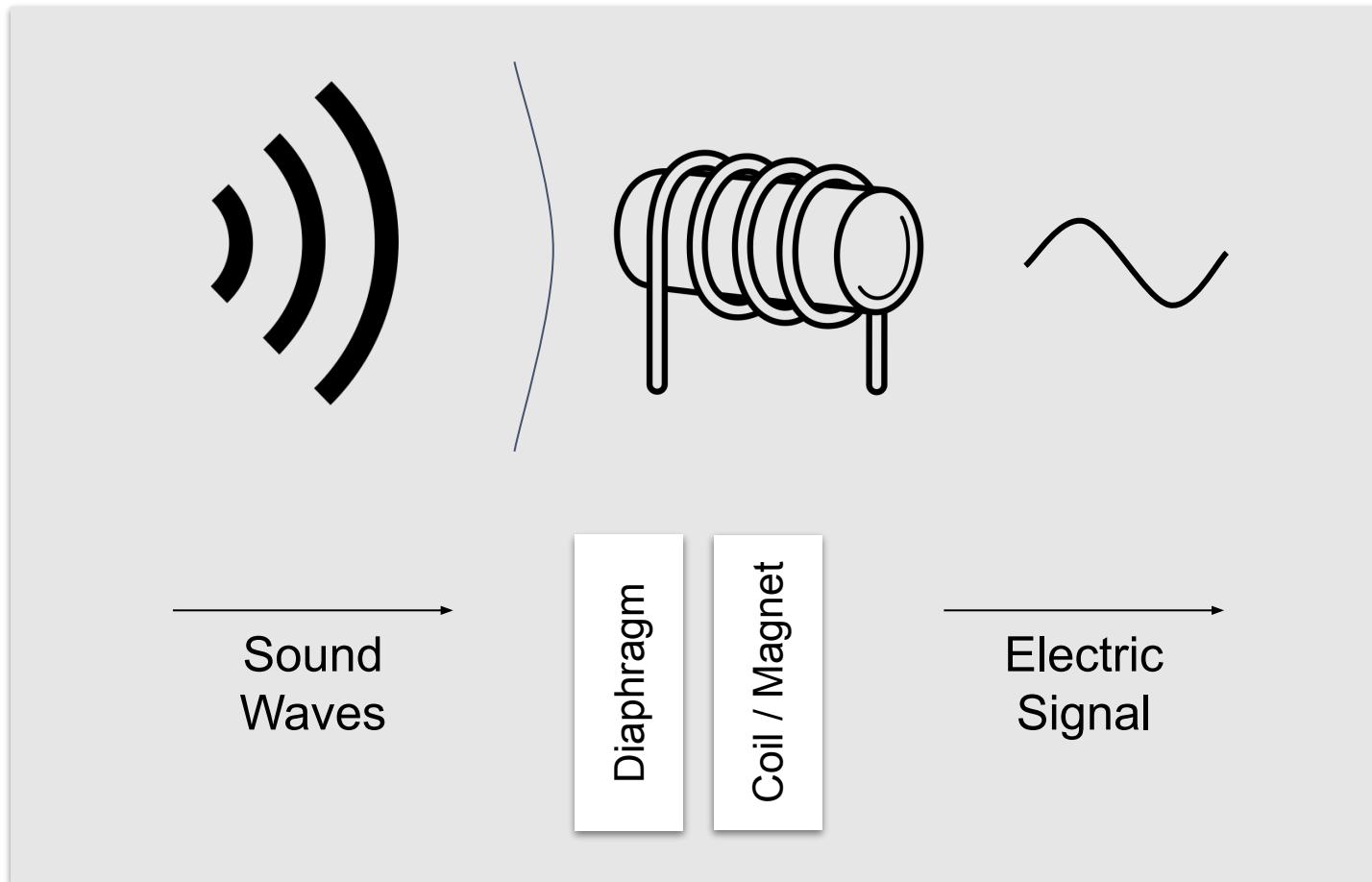


# Data Collection & Pre-Processing

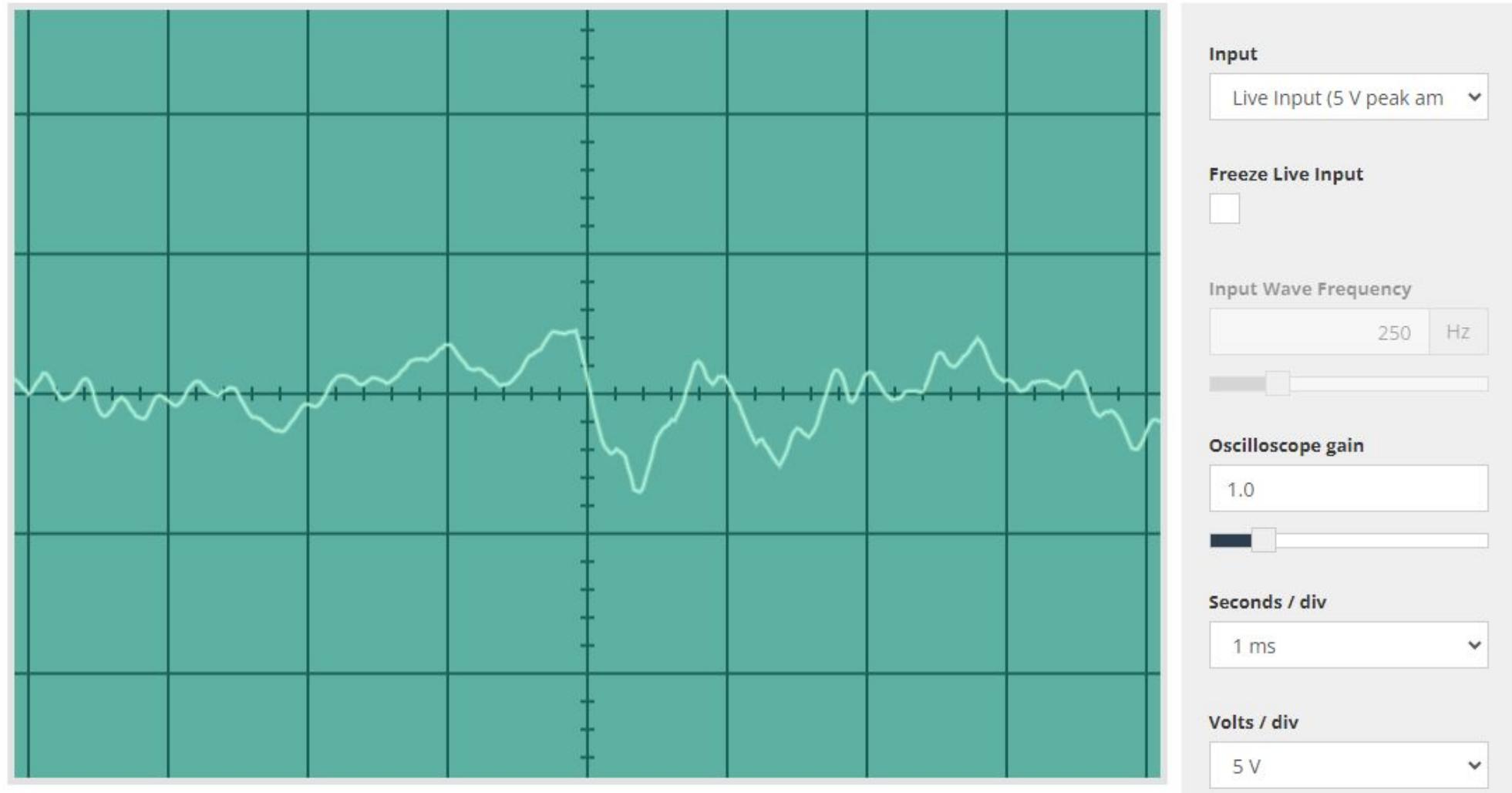




# Sensor Data



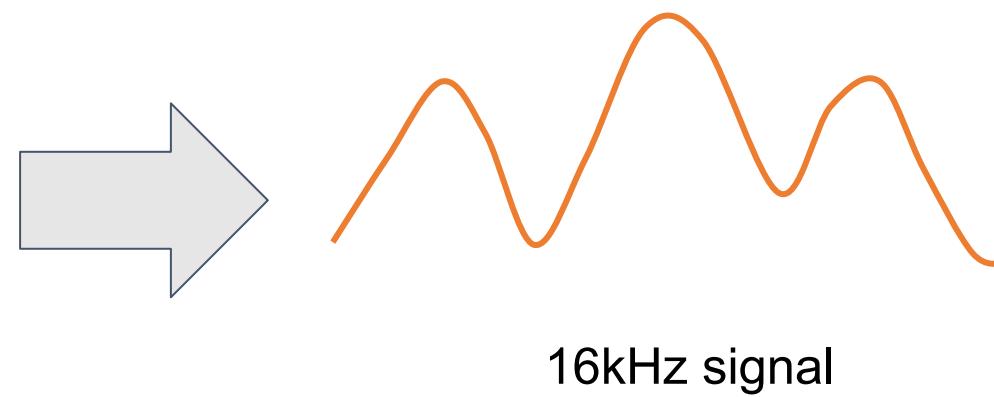
# Sensor Data



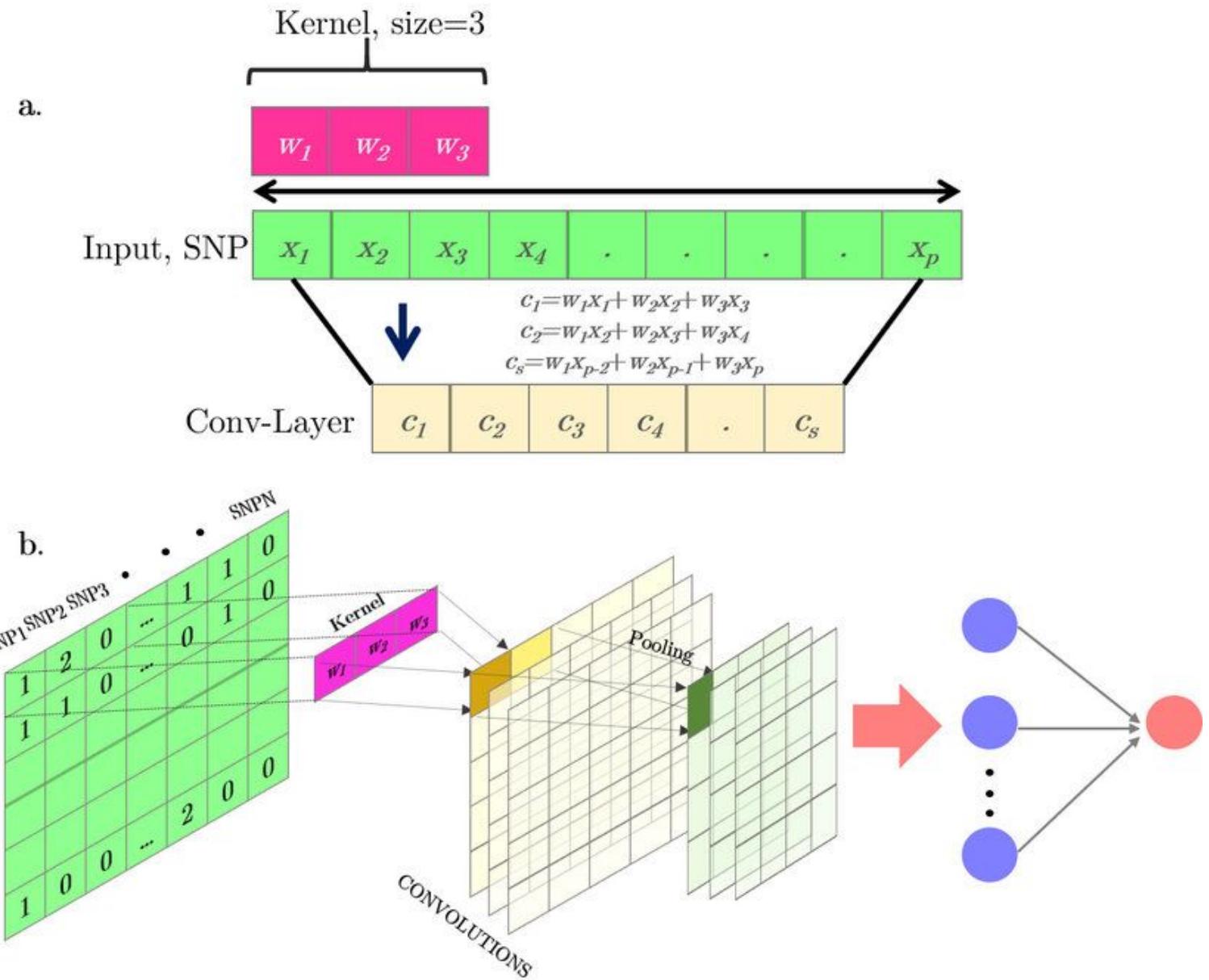
<https://academo.org/demos/virtual-oscilloscope/>

# Sensor Data

- 1D signal
- 16kHz signal, so that's **16000** samples (points / second)
- How do you feed ***all*** of that data into the model?
- Need to **think creatively** about the input signal!

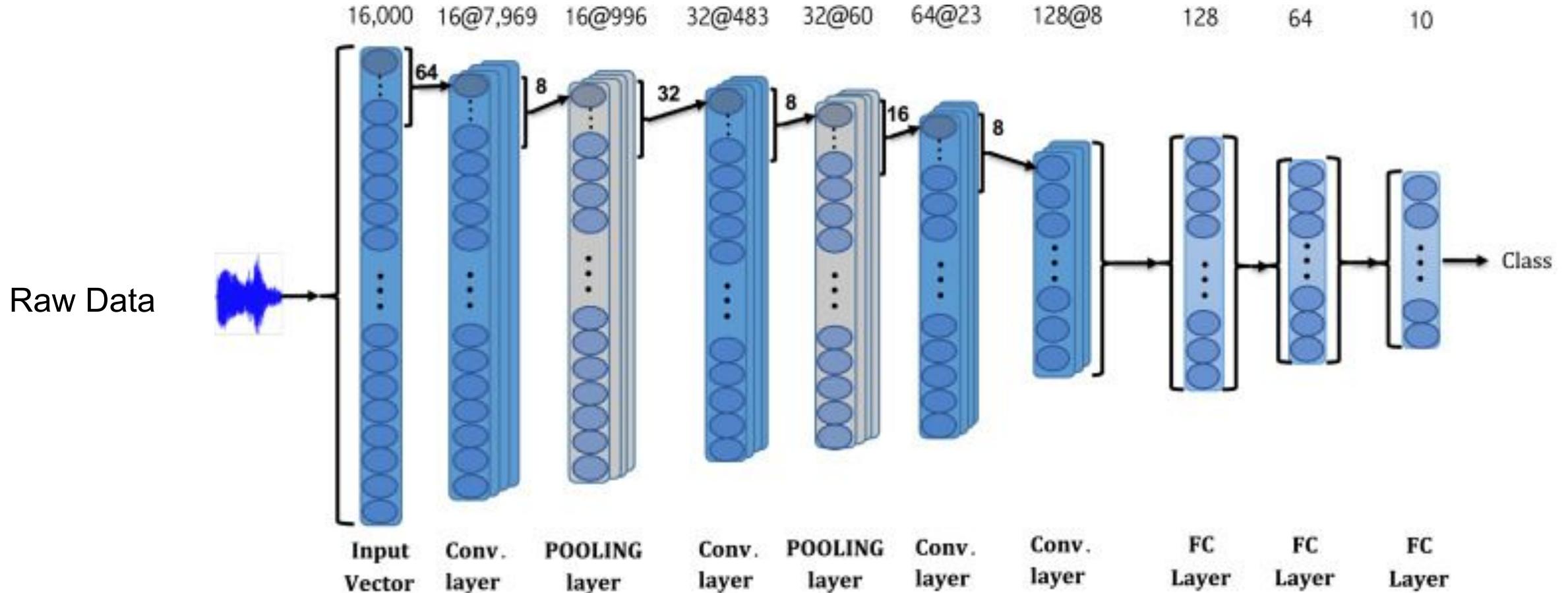


# 1D Convolution



[https://www.researchgate.net/figure/a-Simple-scheme-of-a-one-dimensional-1D-convolutional-operation-b-Full-fig2\\_334609713](https://www.researchgate.net/figure/a-Simple-scheme-of-a-one-dimensional-1D-convolutional-operation-b-Full-fig2_334609713)

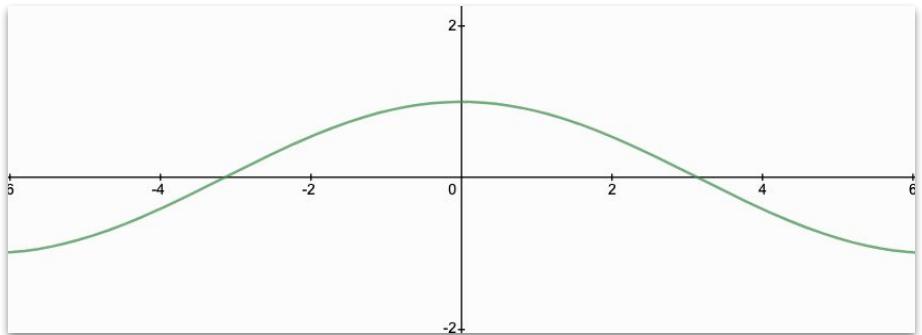
# 1D Convolution



End-to-End Environmental Sound  
Classification using a 1D Convolutional Neural  
Network

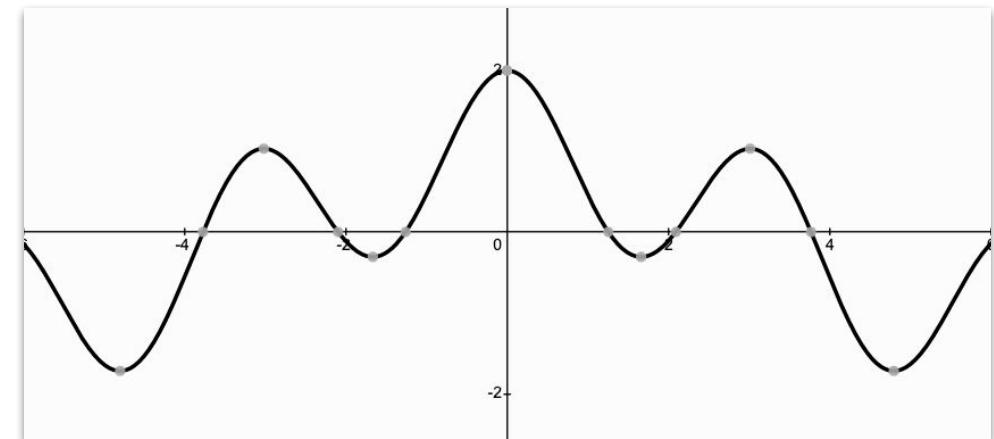
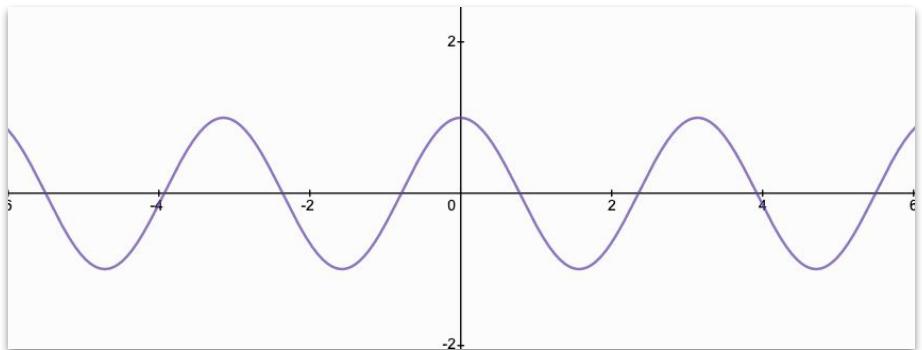
<https://arxiv.org/pdf/1904.08990>

# Signal Components?



+

=



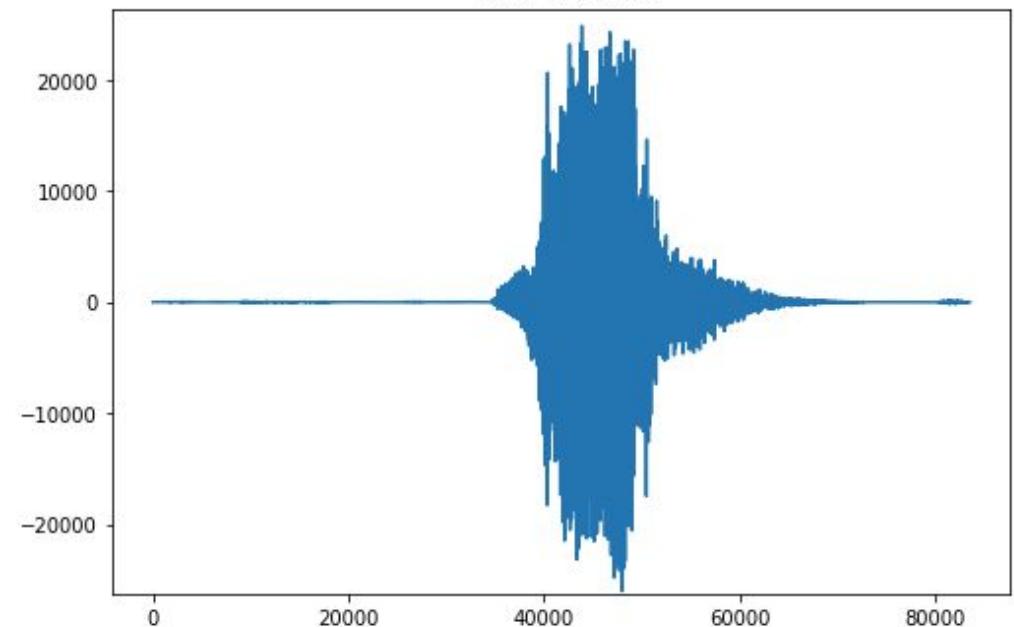
# Signal Components?

?

+

?

=



# Signal Components?

R// Fourier Transform

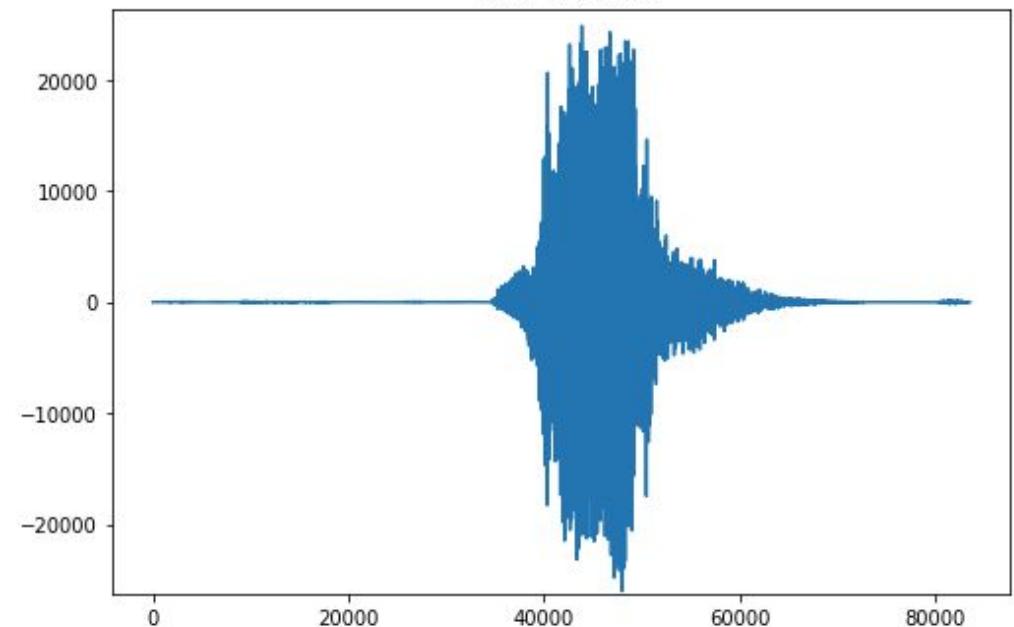


+

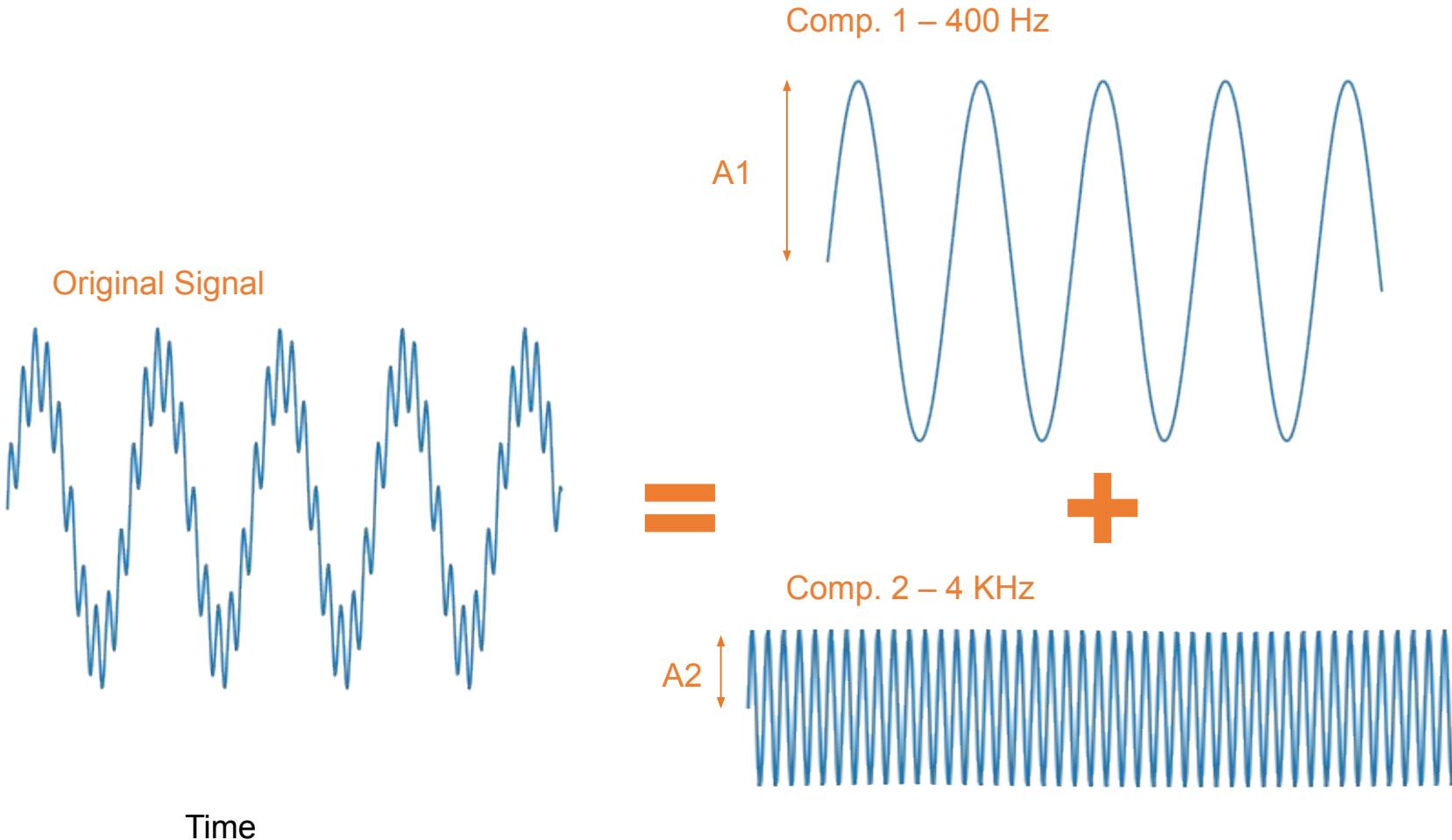
=



[https://www.youtube.com/watch  
?v=spUNpyF58BY&sttick=0](https://www.youtube.com/watch?v=spUNpyF58BY&sttick=0)

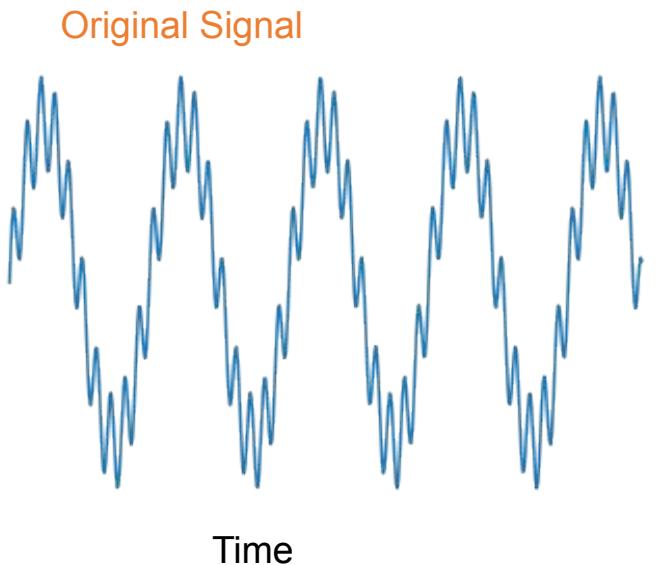


# Fourier Transform



<https://www.youtube.com/watch?v=spUNpyF58BY&sticker=0>

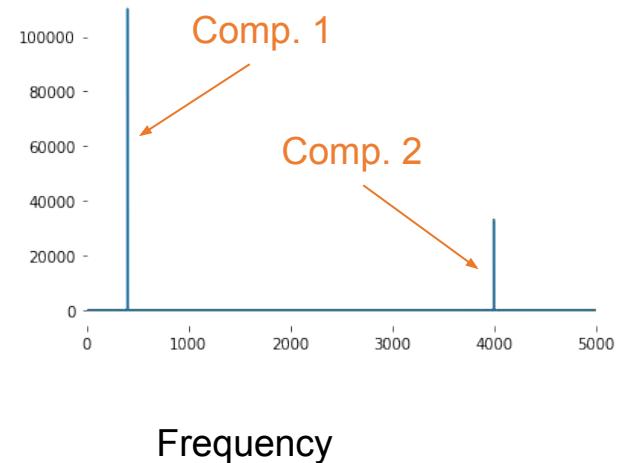
# Fourier Transform



$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

**Fourier Transform**

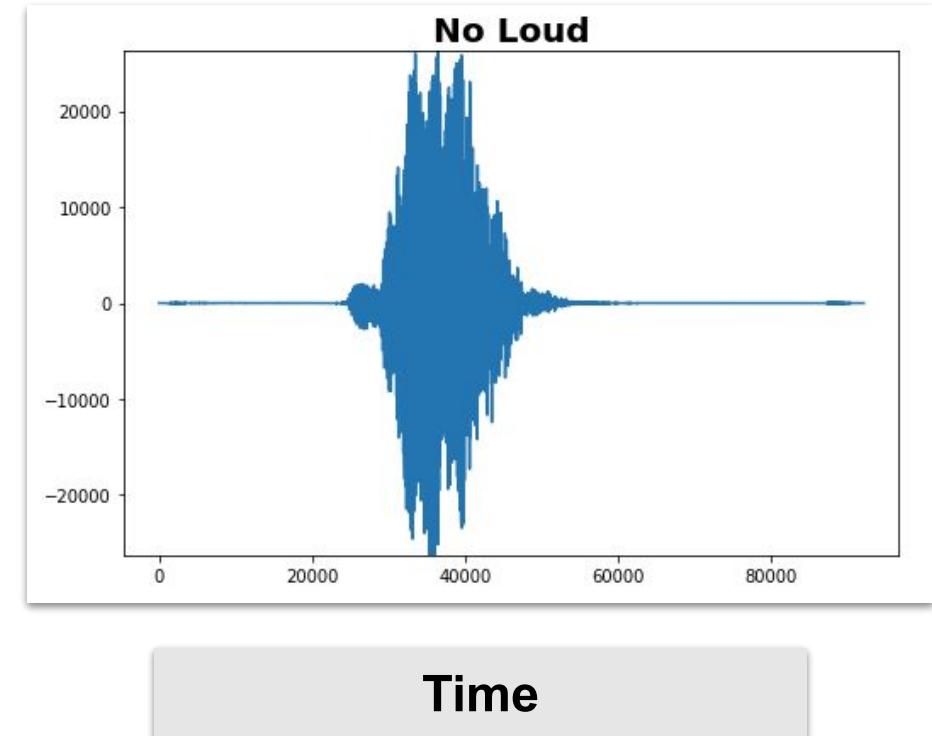
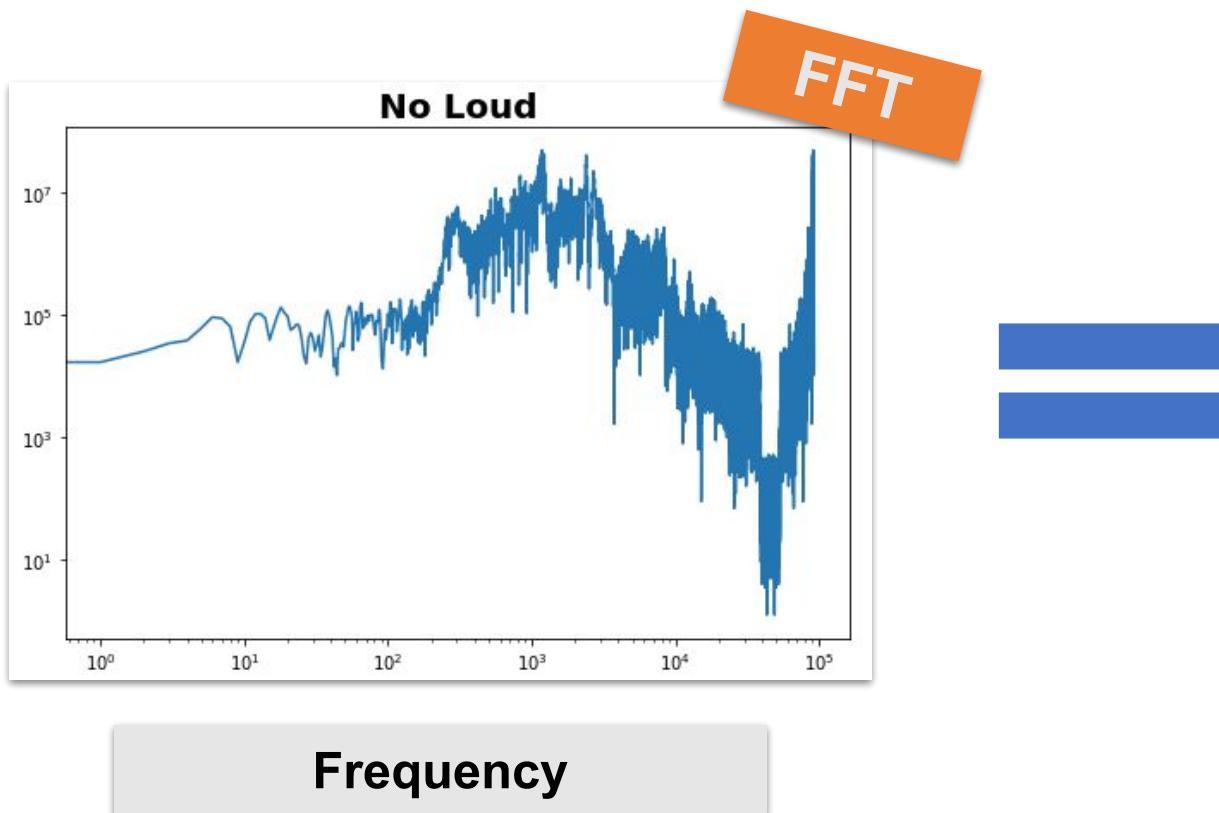
```
from scipy.fft import fft  
yf = fft(raw signal)  
plt.plot(xf, np.abs(yf));
```



<https://prajwalsouza.github.io/Experiments/Fourier-Transform-Visualisation.html>

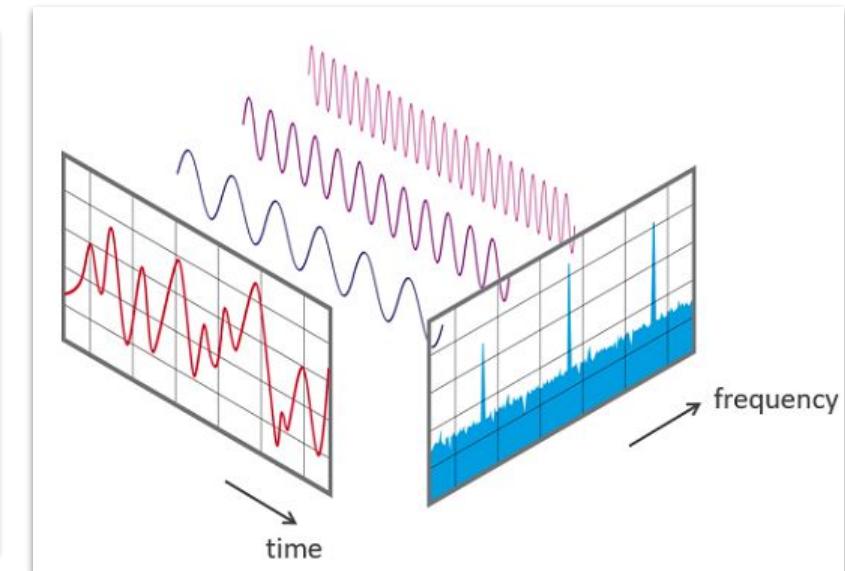
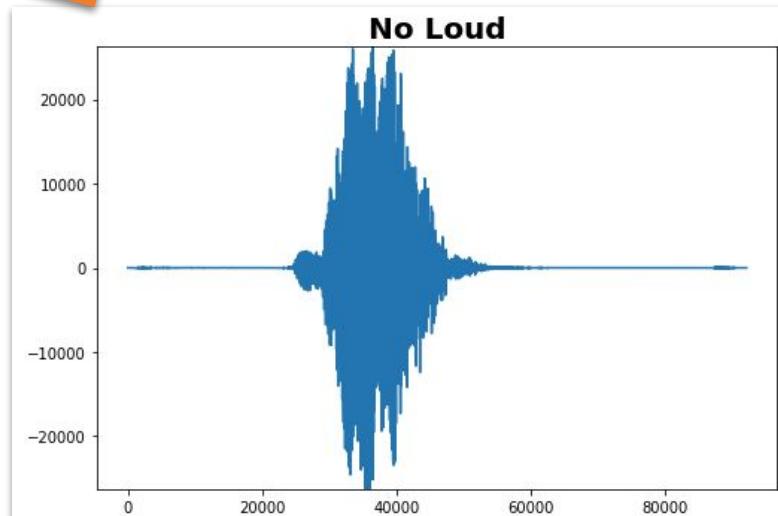
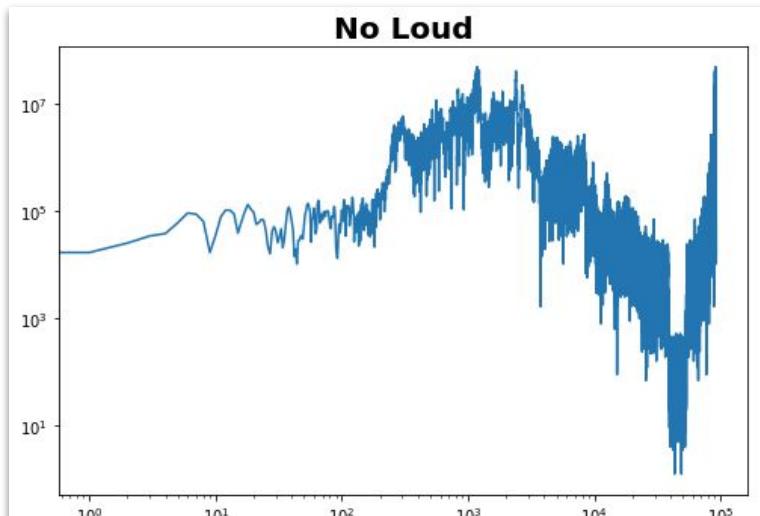
<https://www.youtube.com/watch?v=spUNpyF58BY&sttck=0>

# Signal Components!

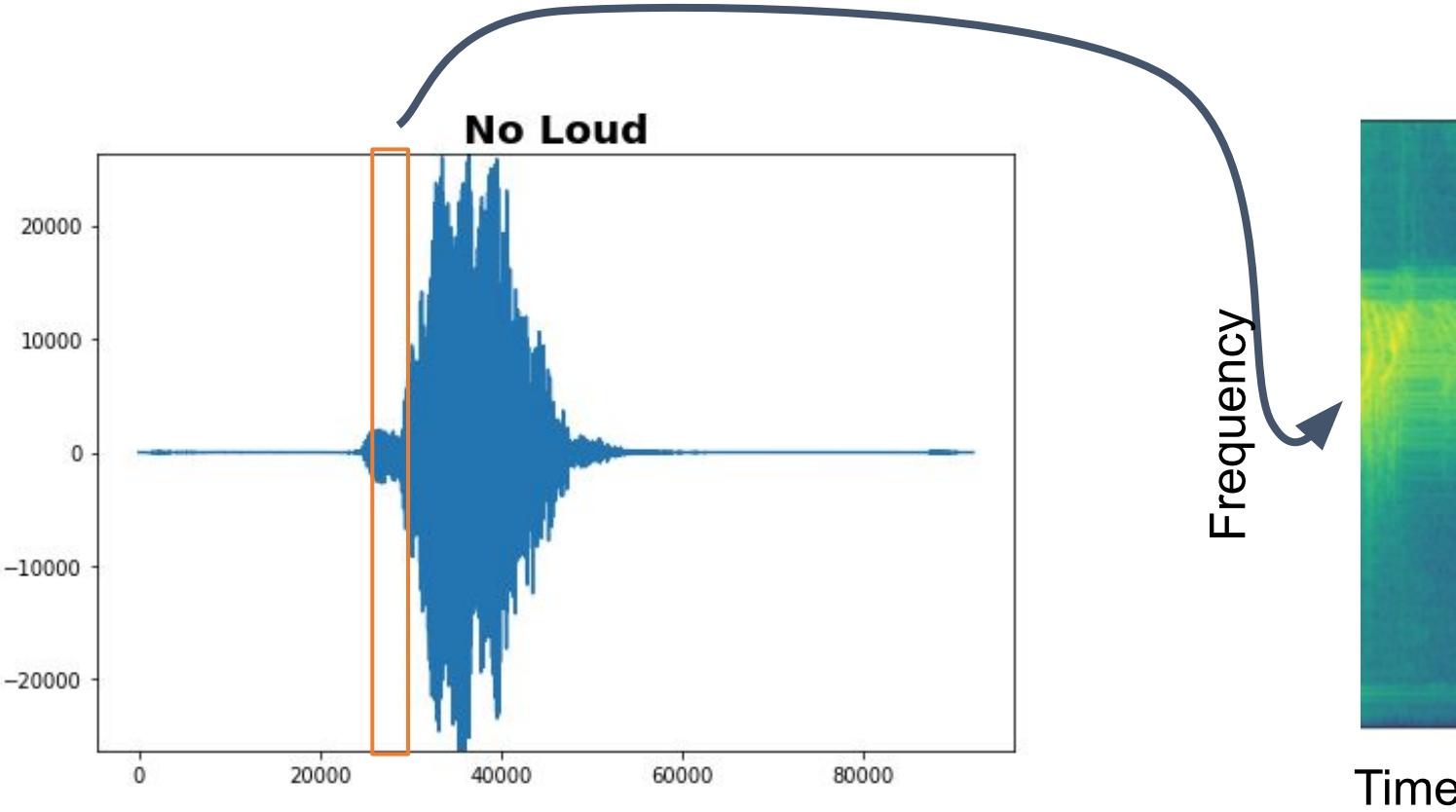


# How Can We Mix Time and Frequency Information?

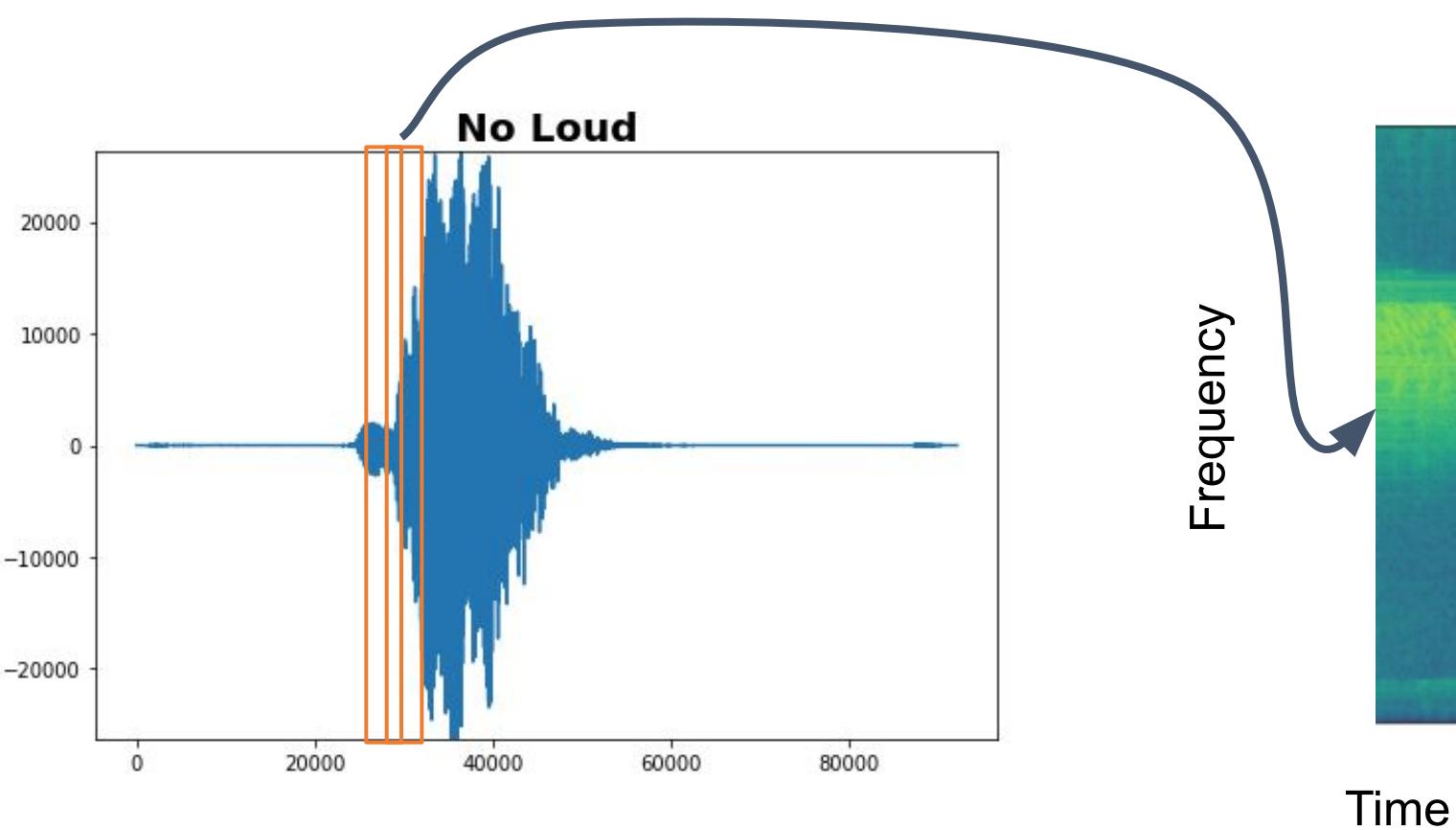
FFT



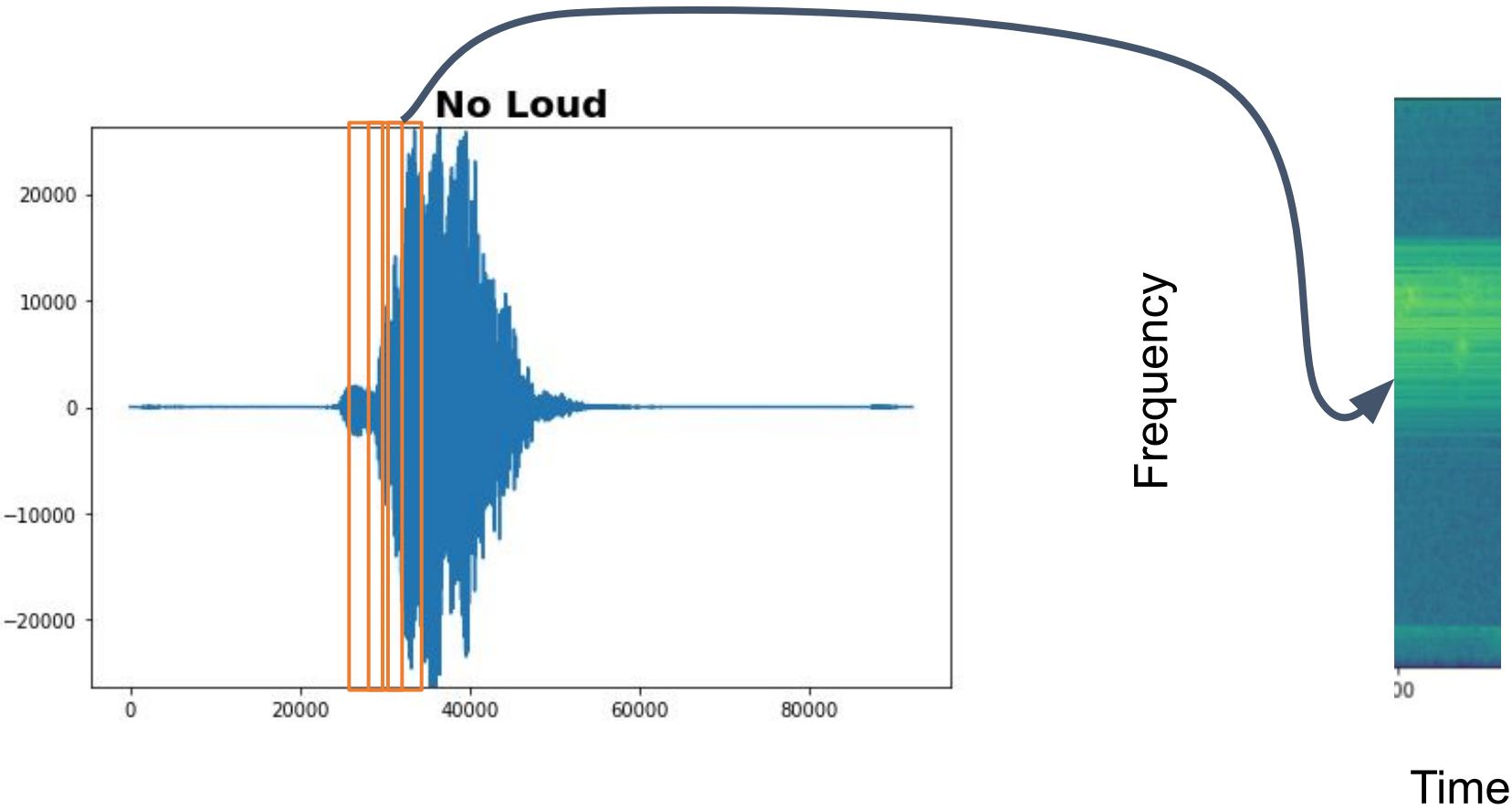
# Data Preprocessing



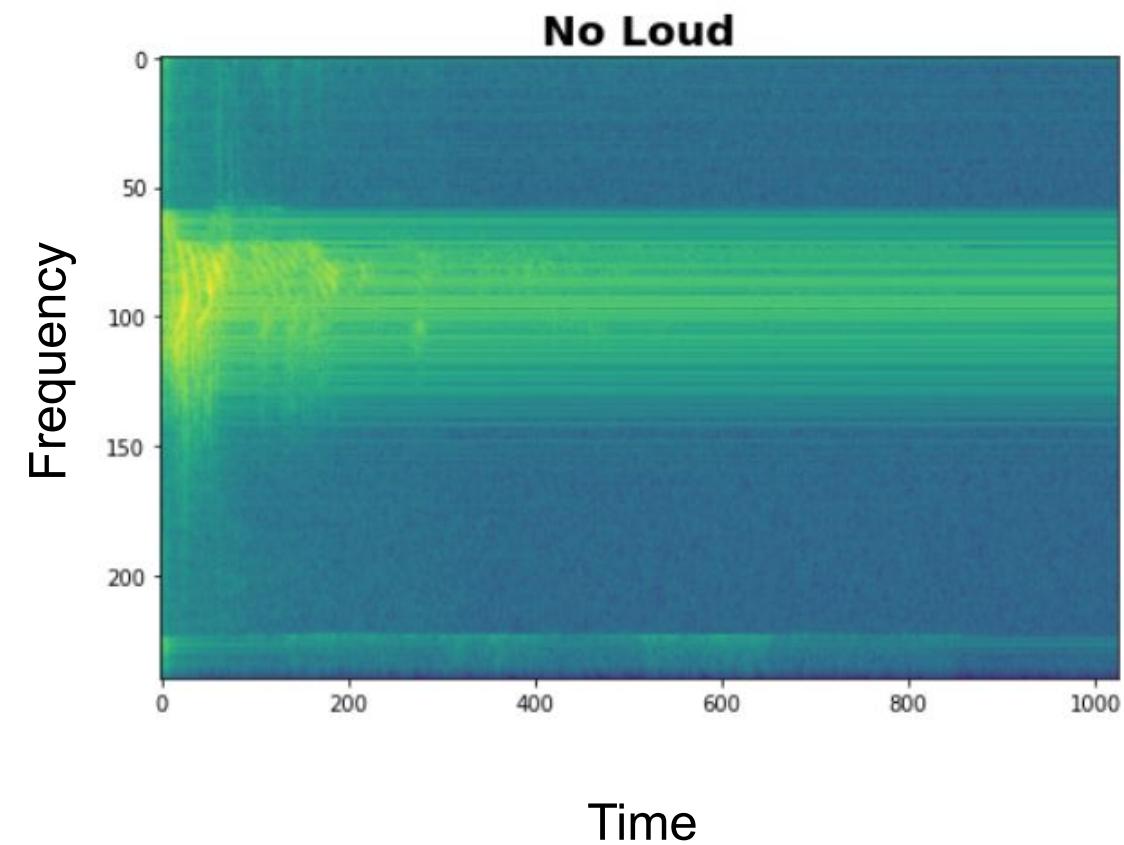
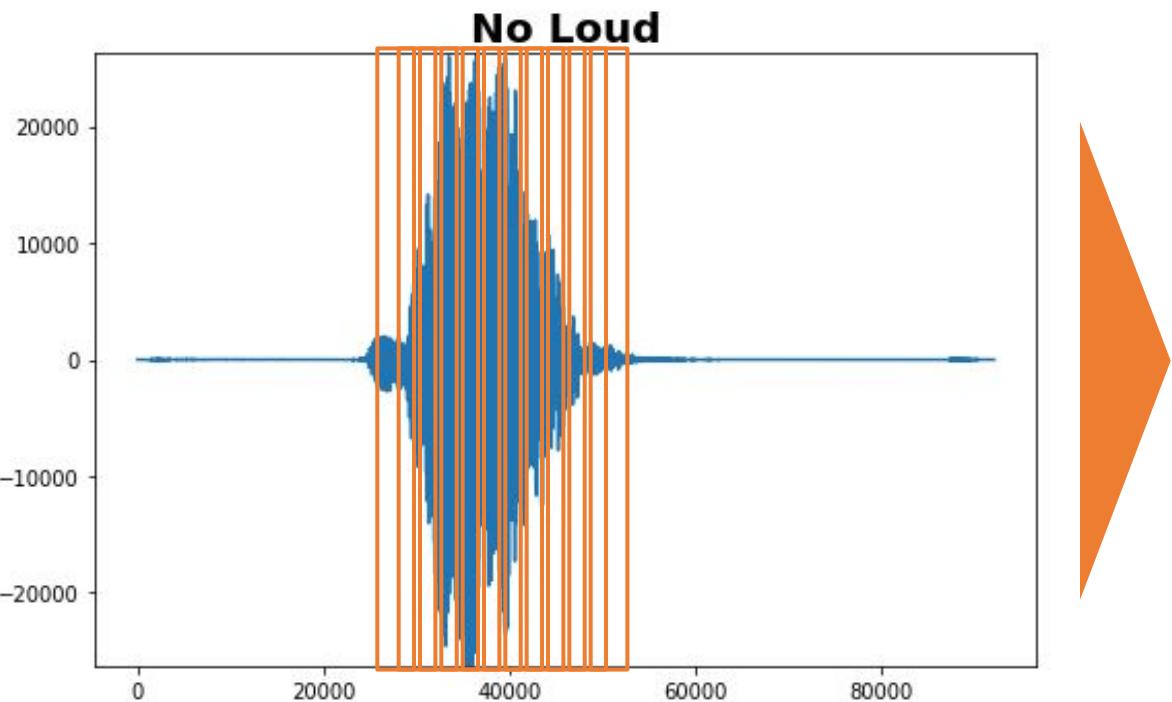
# Data Preprocessing



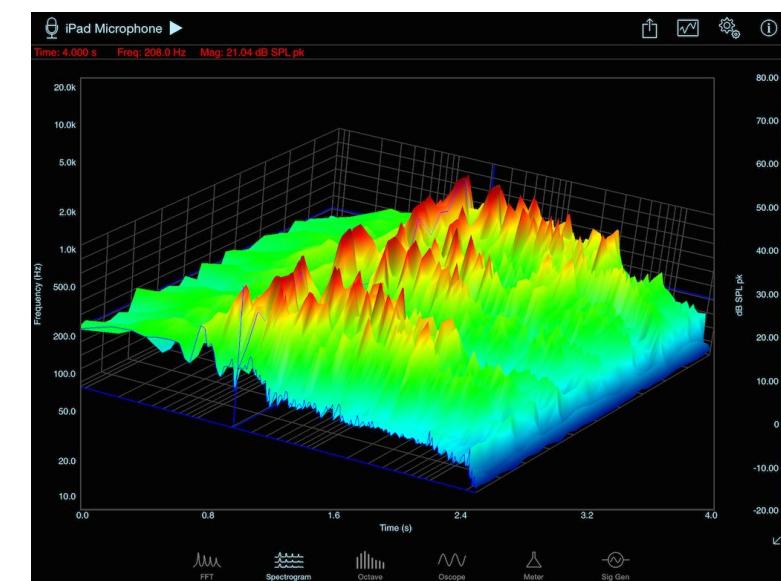
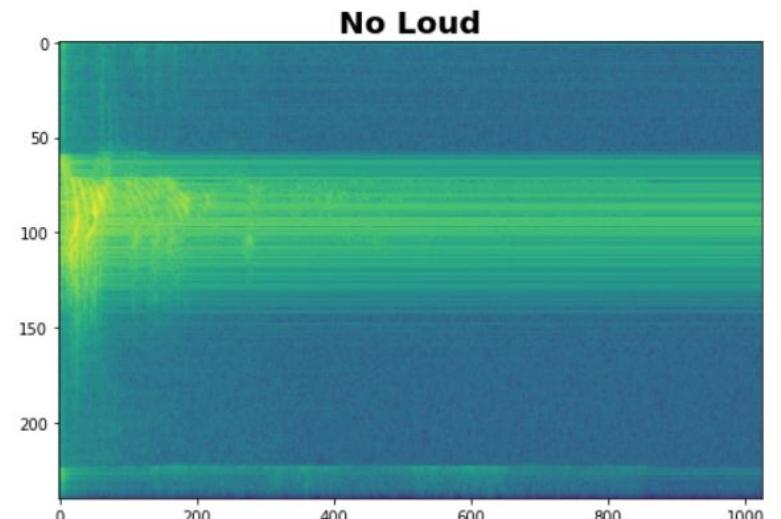
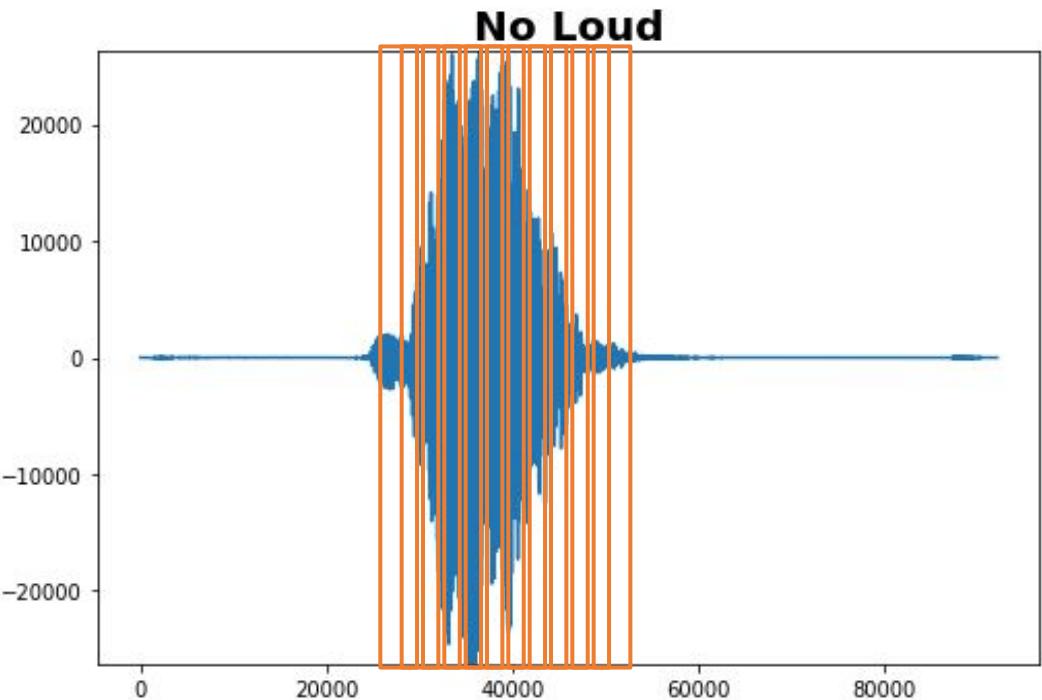
# Data Preprocessing



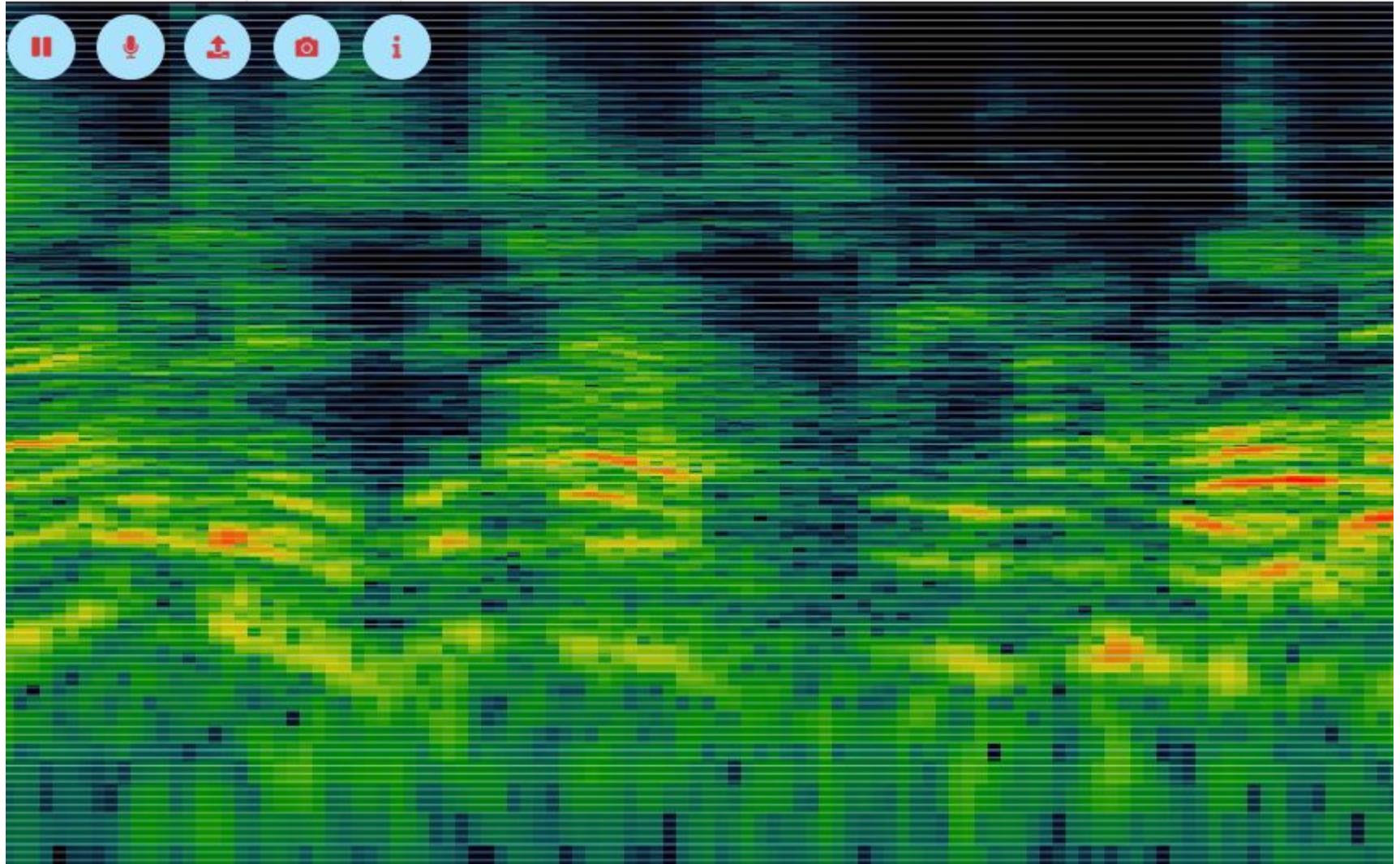
# Data Preprocessing: Spectrograms



# Data Preprocessing: Spectrograms

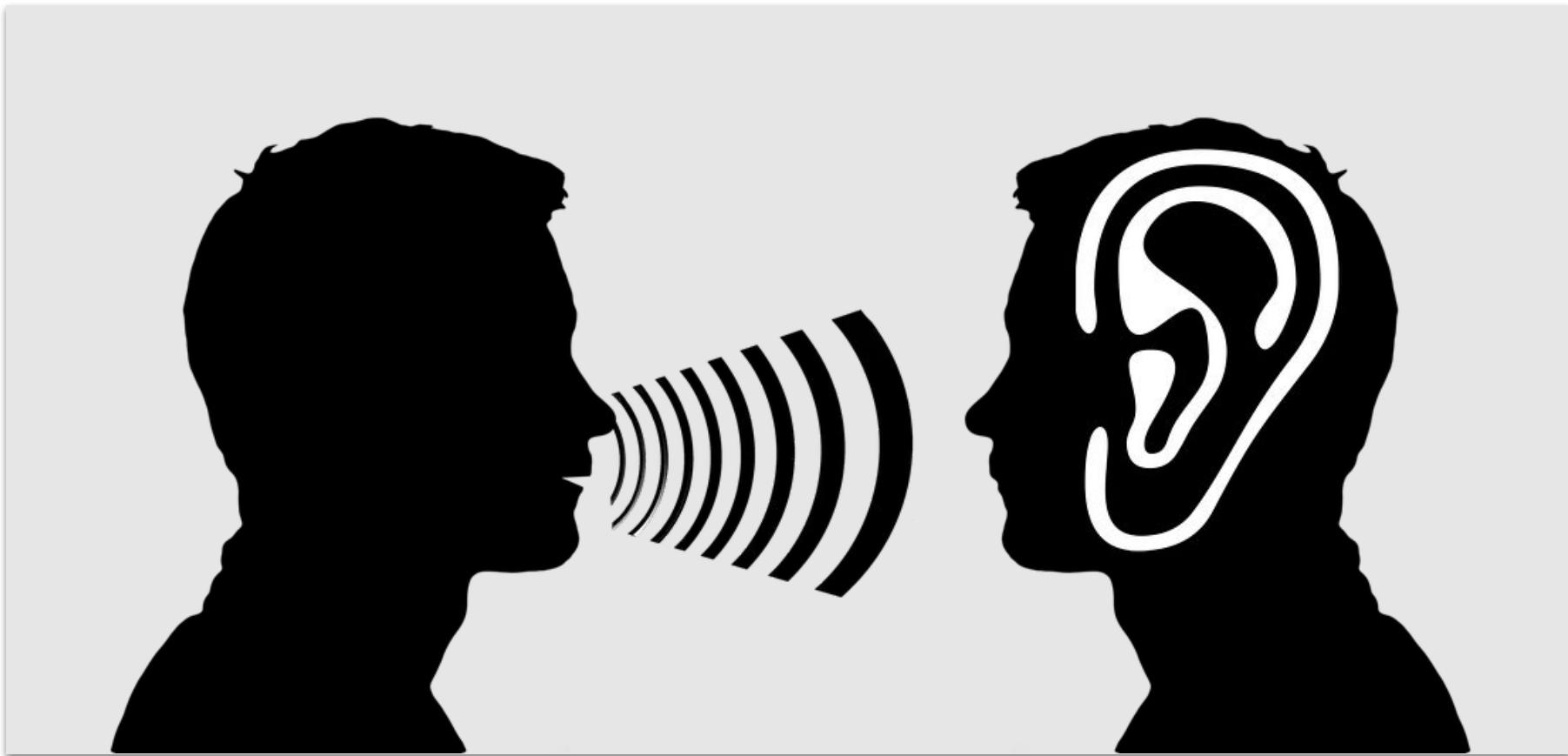


# Data Preprocessing: Spectrograms

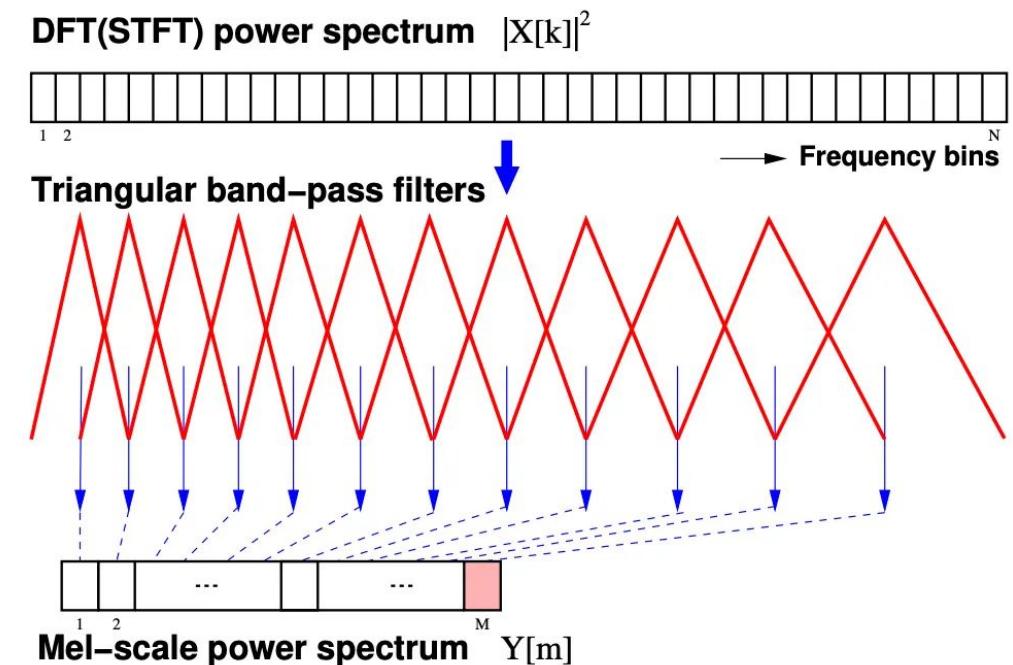
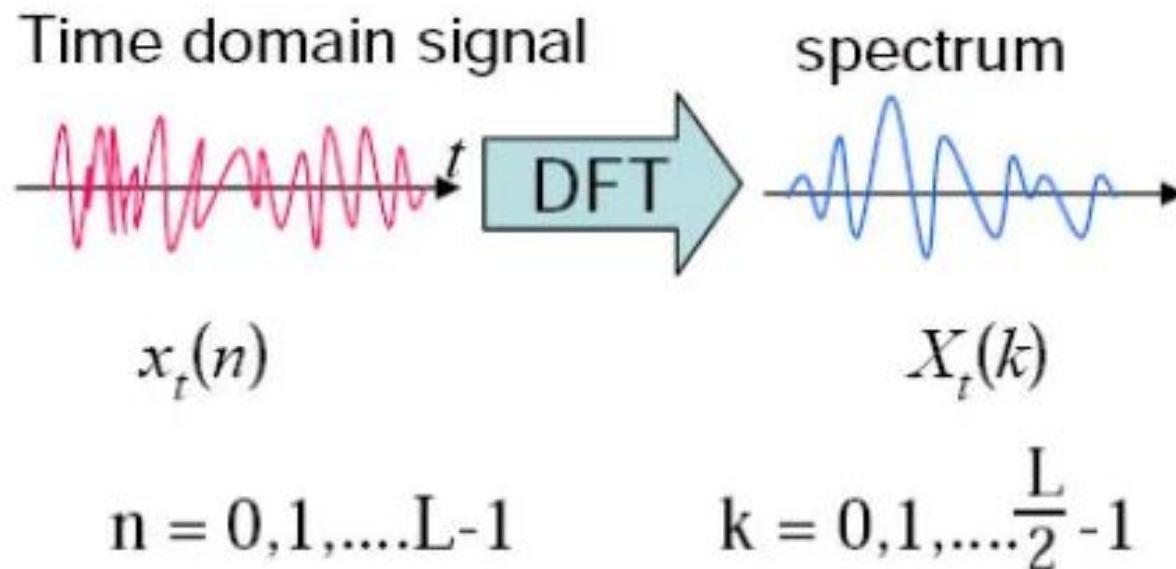


<https://spectrogram.sciencemusic.org/>

Can we find more  
*salient* features?



# Mel-Frequency Cepstral Coefficients : MFCC

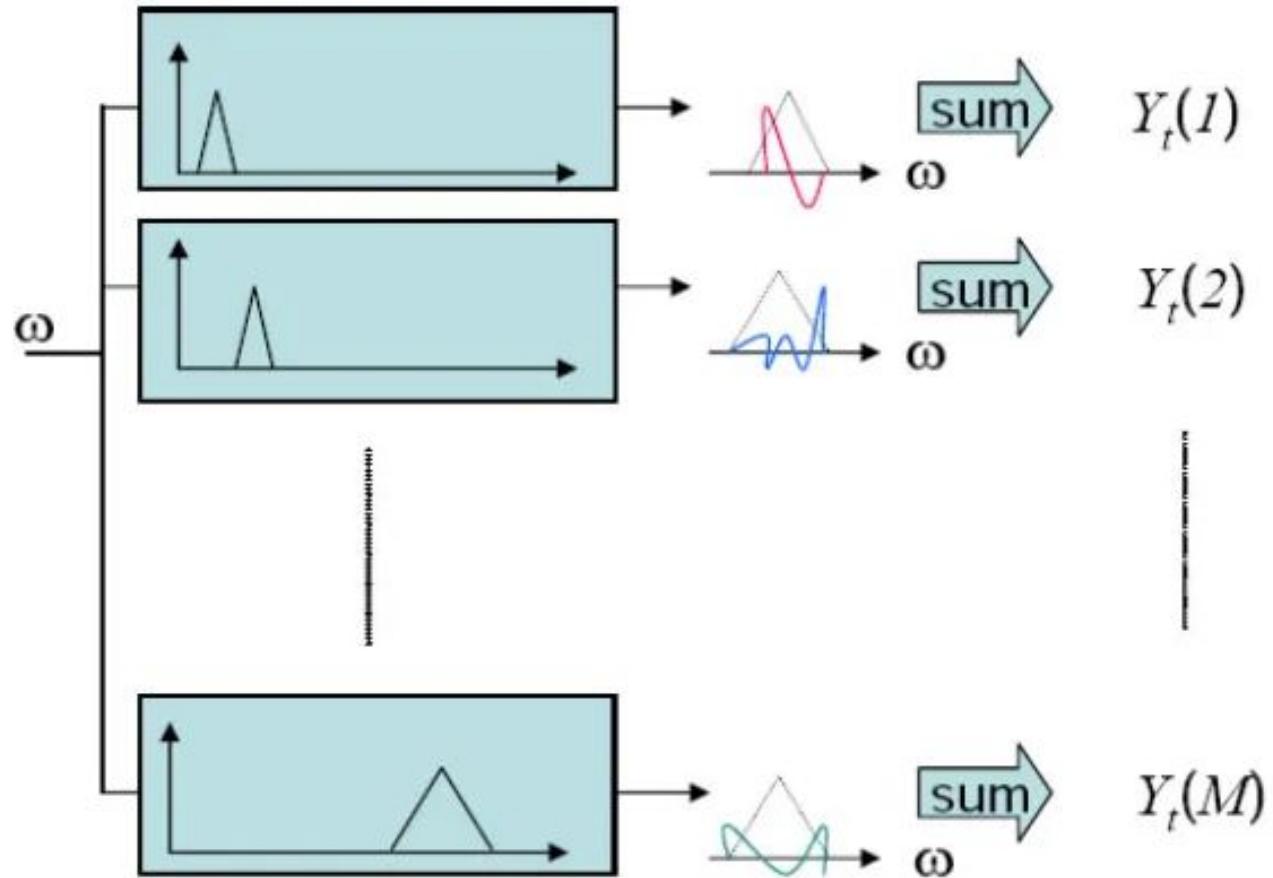


<https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9>

# Mel-Frequency Cepstral Coefficients : MFCC

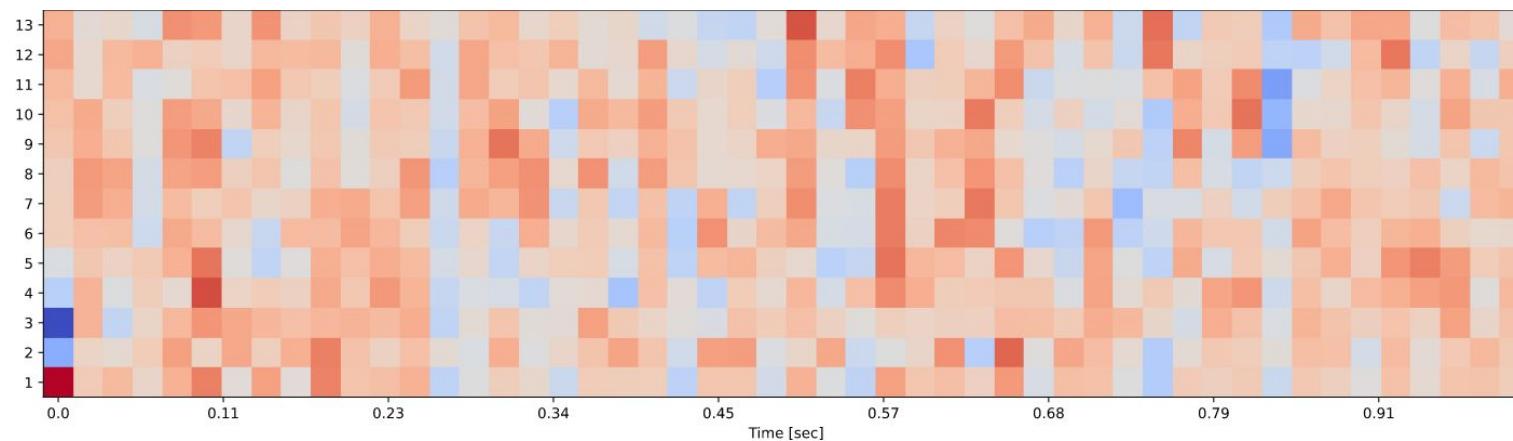
$$Y_t[m] = \sum_{k=1}^N W_m[k] |X_t[k]|^2$$

where  $k$  : DFT bin number ( $1, \dots, N$ )  
 $m$  : mel-filter bank number ( $1, \dots, M$ )



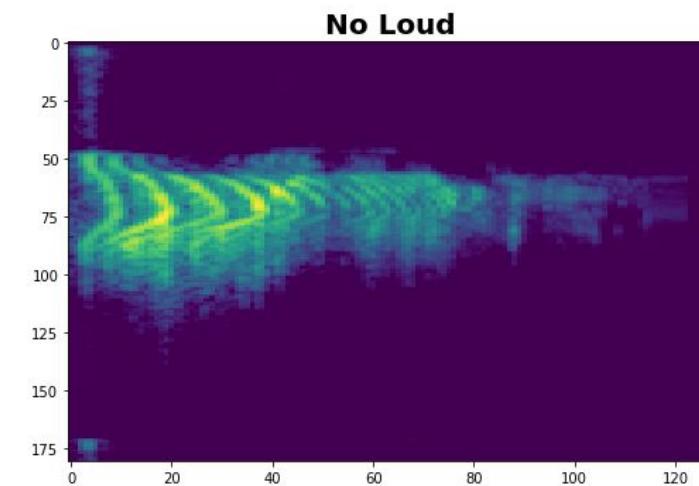
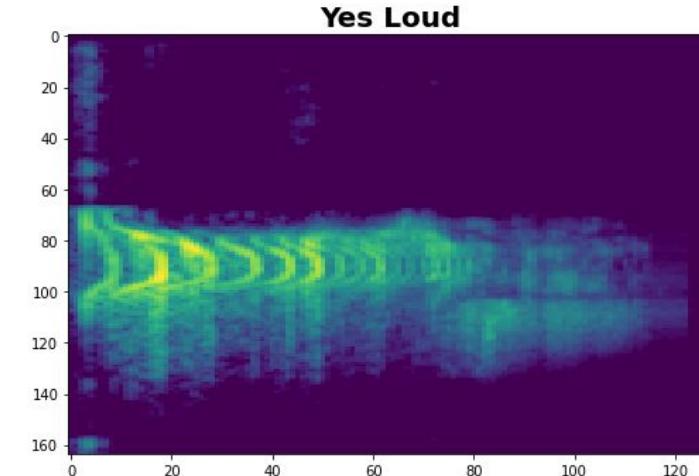
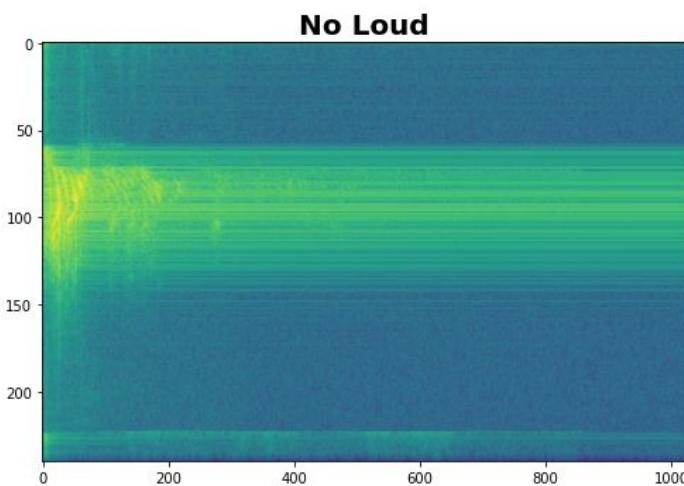
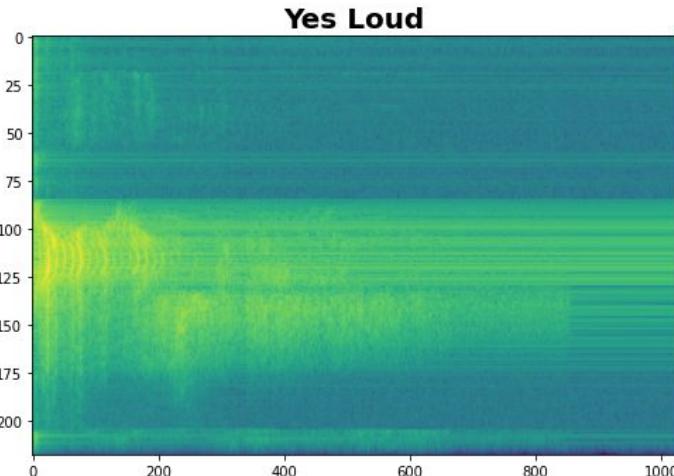
<https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9>

# Mel-Frequency Cepstral Coefficients : MFCC

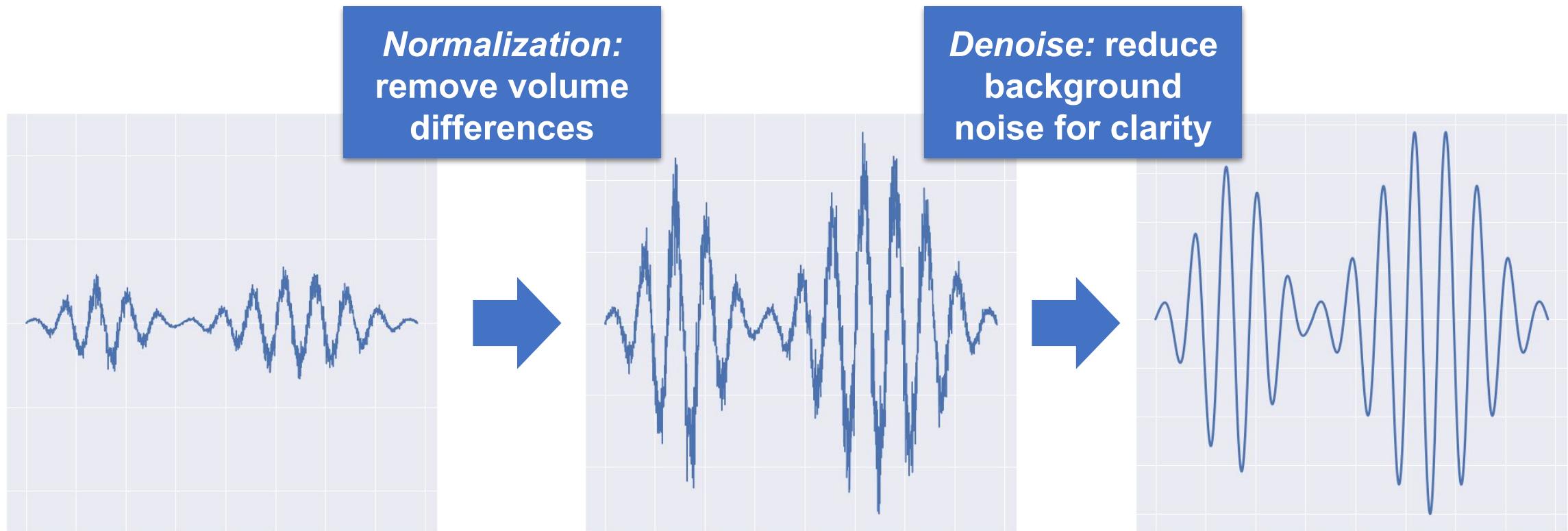


Ejemplo de MFCC

# Spectrograms vs MFCCs



# Additional Feature Engineering

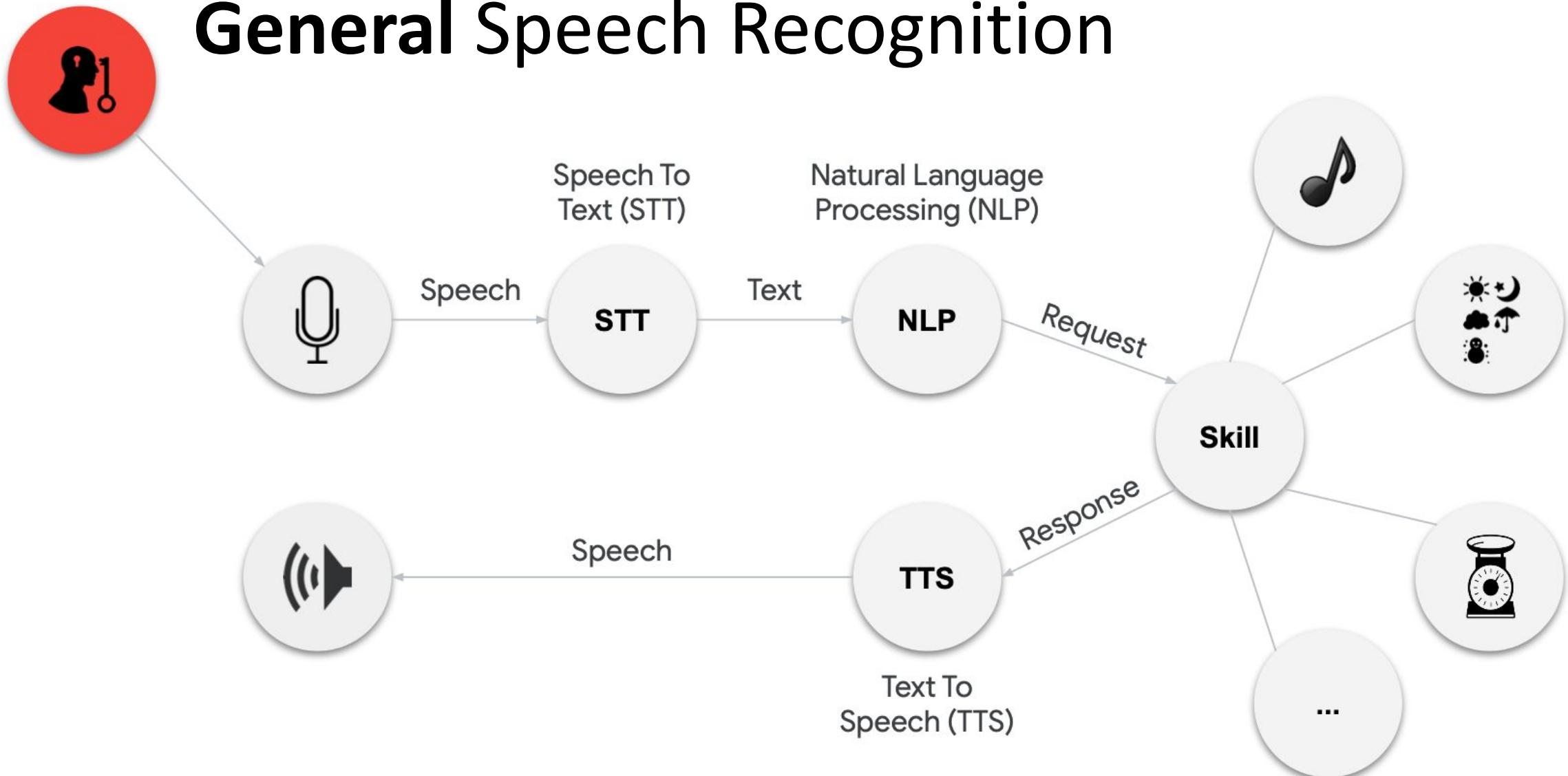


What is  
KeyWord Spotting?

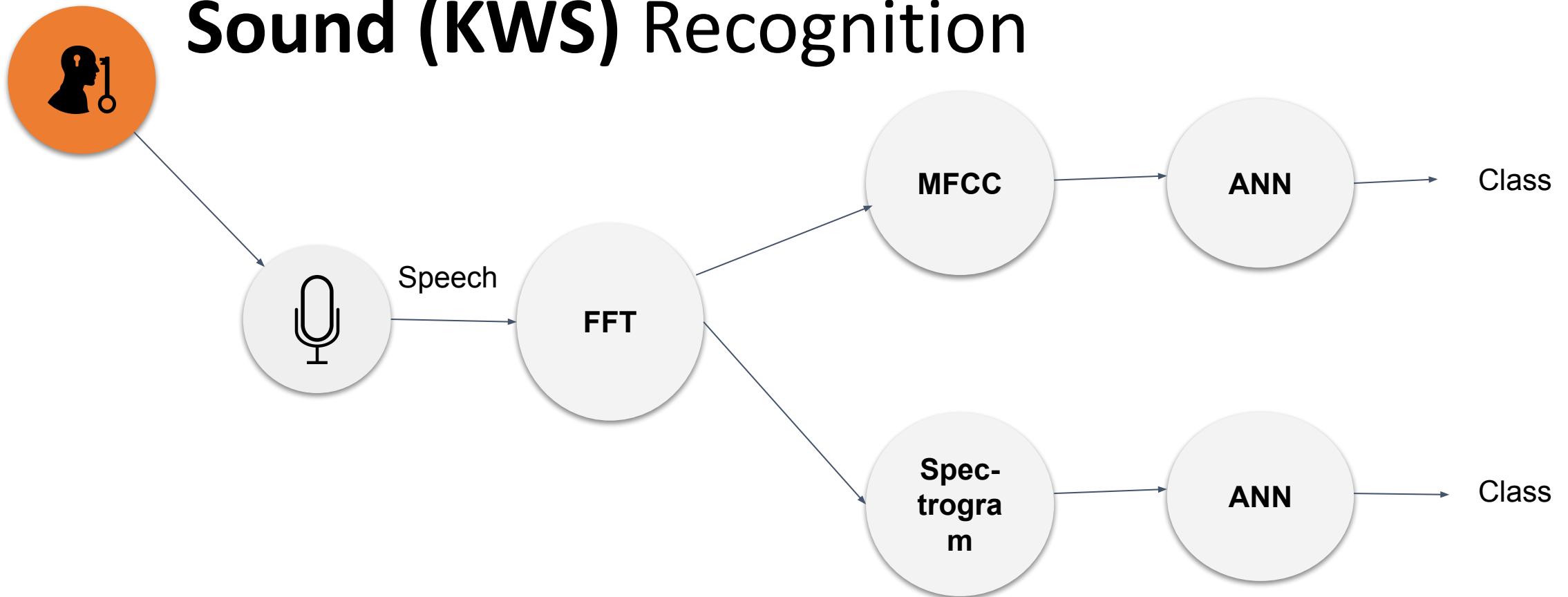
# Keyword Spotting v. General Speech Recognition

- **Keyword spotting** is one of the most successful examples of **TinyML**
  - Low-power, continuous, on-device
  - Common Voice SWTS<sup>\*</sup> expands keyword spotting to more languages
    - \* Single Word Target Segment
- **General ASR**<sup>\*</sup> still requires **larger, power-hungry models**
  - But it can run on mobile devices (offline dictation on smartphones)
    - \* Automatic Speech Recognition

# General Speech Recognition



# Sound (KWS) Recognition









## More than just voice

- **Security** (Broken Glass)
- **Industry** (Anomaly Detection)
- **Medical** (Snore, Toss)
- **Nature** (Bee, insect sound)



# Keyword Spotting Challenges/Constraints

# Challenges and Constraints



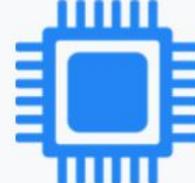
Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

**LATENCY**

Provide results quickly, respond in real-time to the user

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

## **BANDWIDTH**

Minimize data sent over the network (slow and expensive)

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

**ACCURACY**

**Listen  
continuously,  
but only trigger  
at the right time**

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

## PERSONALIZATION

Trigger for the user and **not** for background noise

# Challenges and Constraints



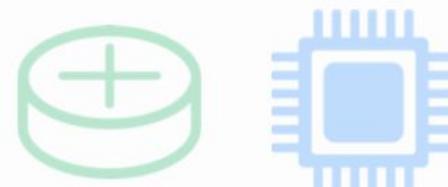
Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

## **SECURITY**

Safeguarding the data that is being sent to the cloud

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

**PRIVACY**

Safeguarding the data that is being sent to the cloud

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy



Battery & Memory

## **BATTERY**

Limited energy,  
operate on  
coin-cell type  
batteries

# Challenges and Constraints



Latency & Bandwidth



Accuracy & Personalization



Security & Privacy

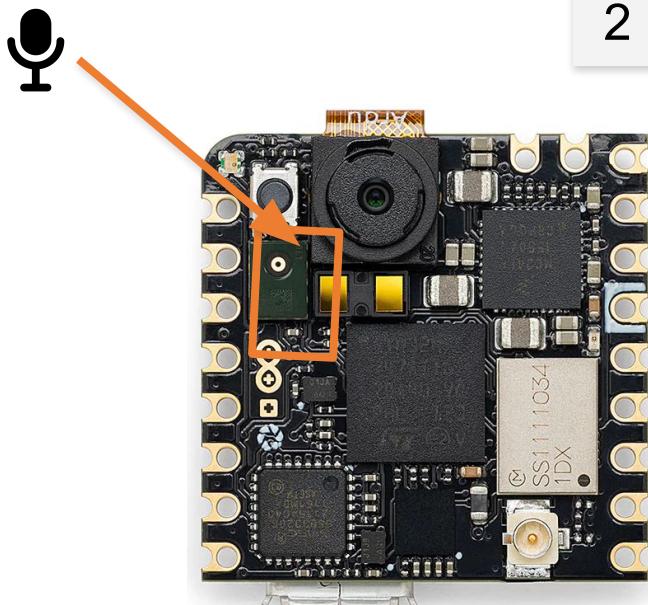


Battery & Memory

**MEMORY**

Run on resource  
constrained  
devices

# Anatomy of a Keyword Spotting Application

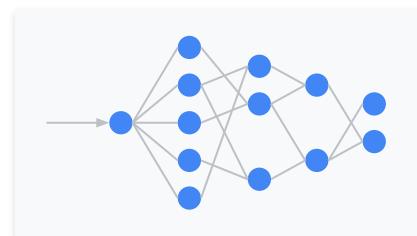


1

Continuously listen on  
the microcontroller

2

Process the data with  
**TinyML** at the edge

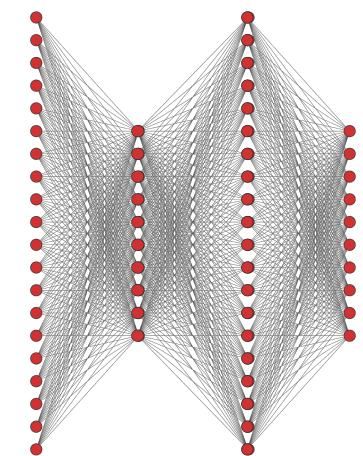


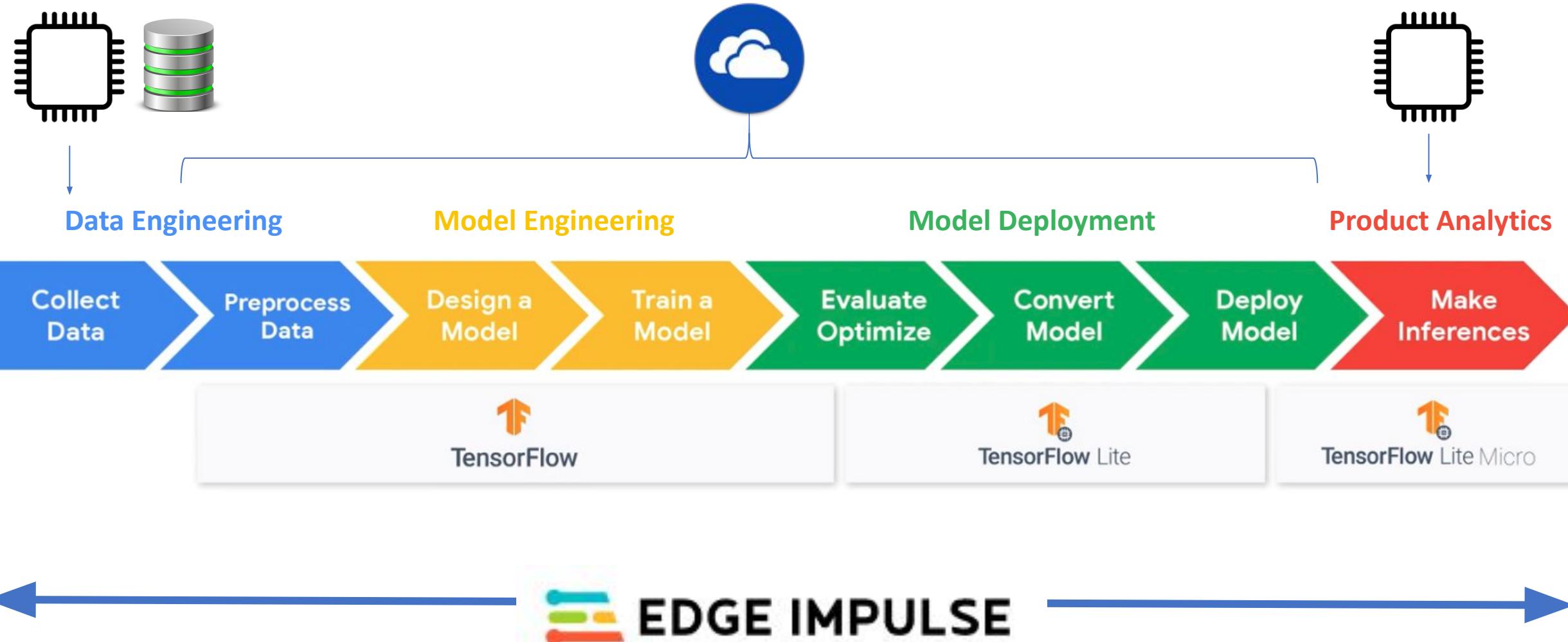
3

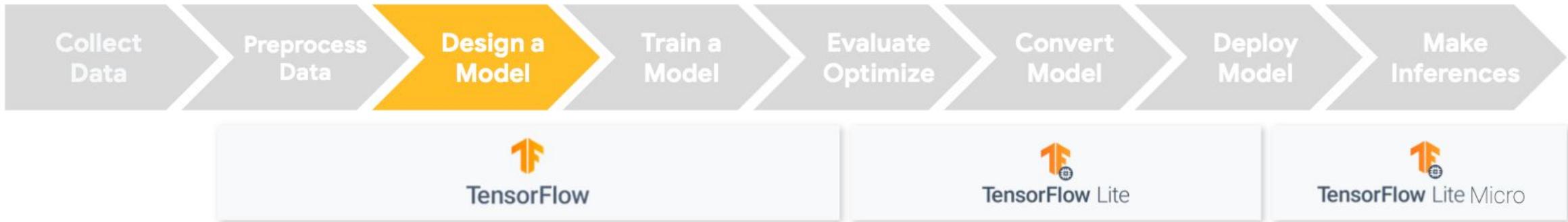
Send the data to the  
cloud when triggered

4

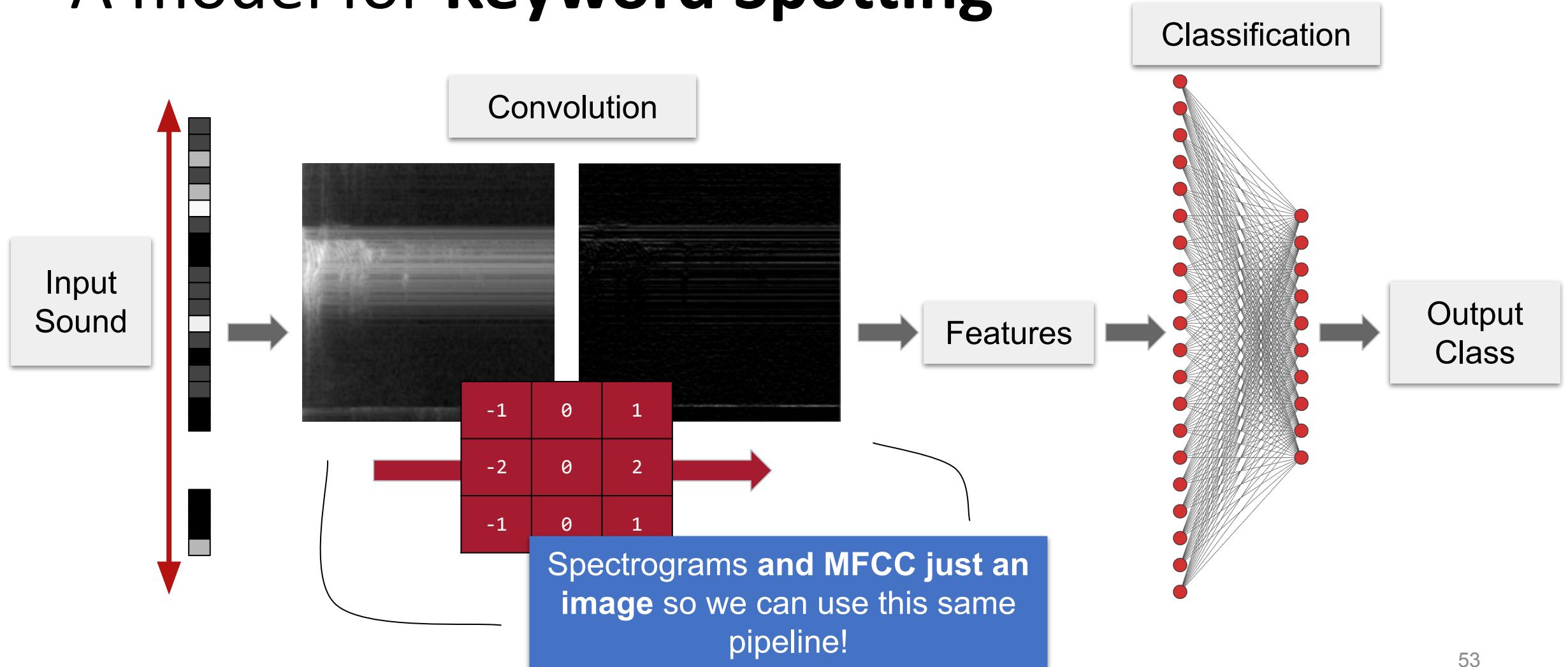
Process the full speech data with  
a large model in the cloud



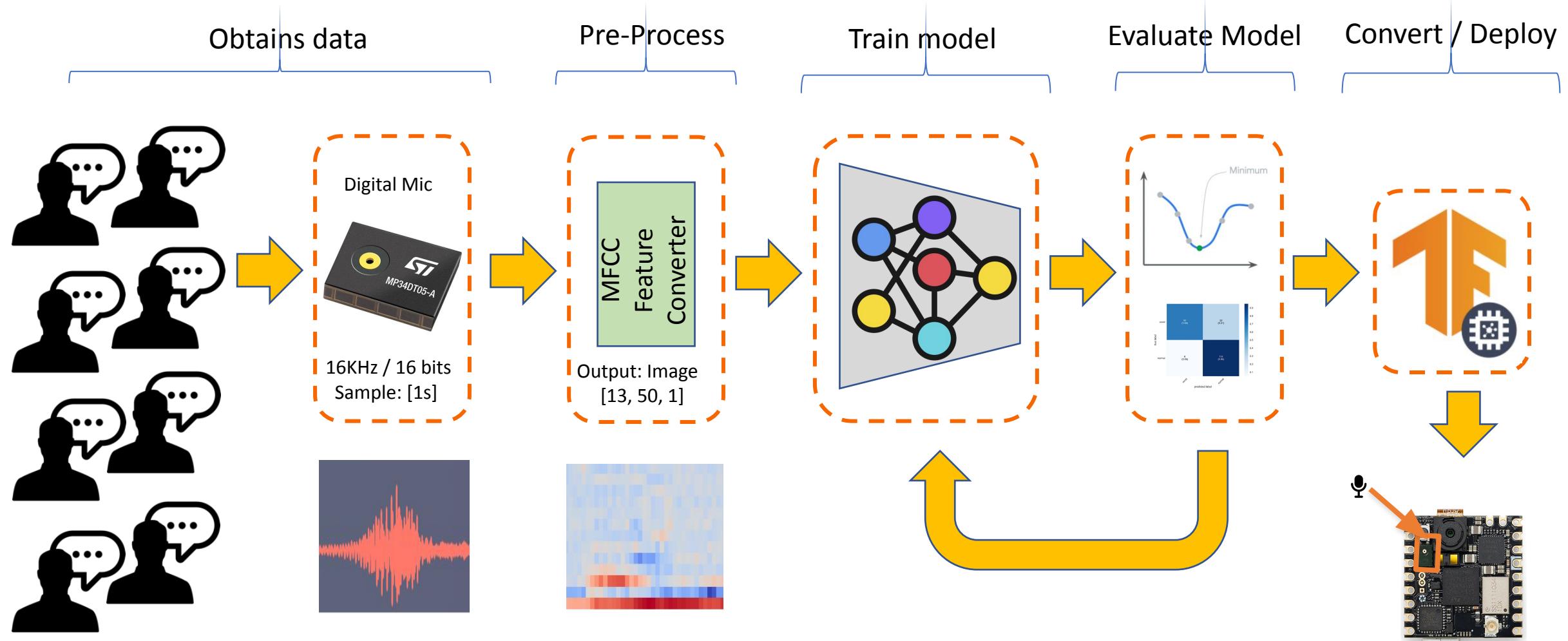




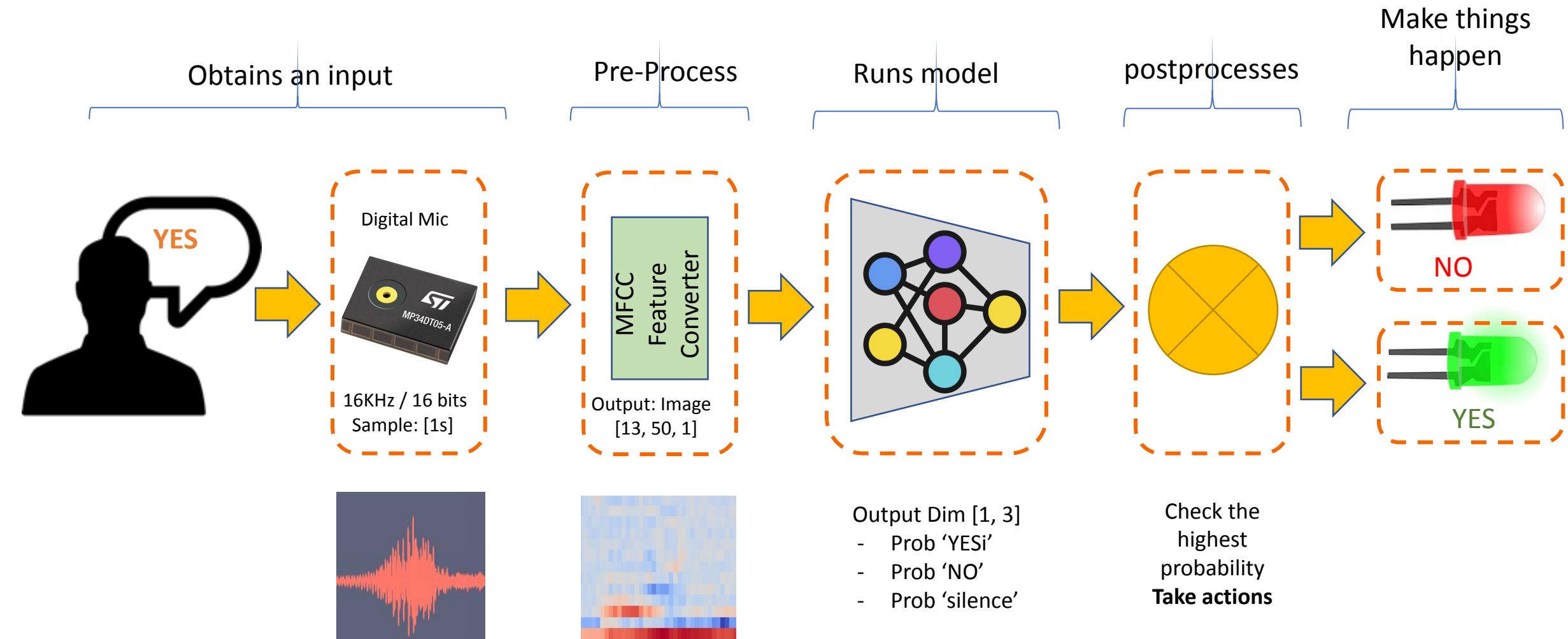
# A model for Keyword Spotting



# KeyWord Spotting (KWS) – Create Model (Training)



# KeyWord Spotting (KWS) - Inference





# Thanks

Prof. Jesús Alfonso López  
[jalopez@uao.edu.co](mailto:jalopez@uao.edu.co)

Universidad Autónoma de  
Occidente

