



**WALC 2024**  
**Applied AI**

# Sound Classification Intro

---

Prof. Marcelo J. Rovai

[rovai@unifei.edu.br](mailto:rovai@unifei.edu.br)

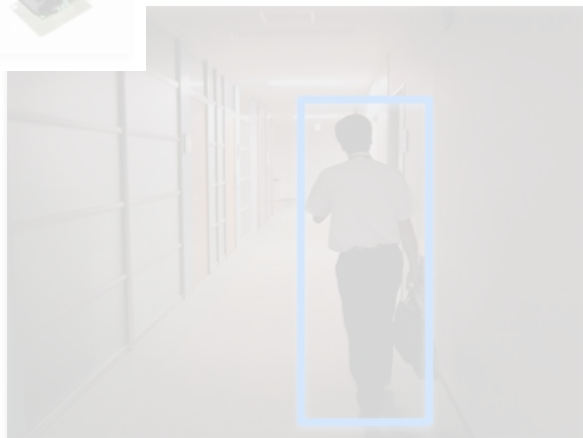
UNIFEI - Federal University of Itajuba, Brazil

TinyML4D Academic Network Co-Chair



**TINYML4D**

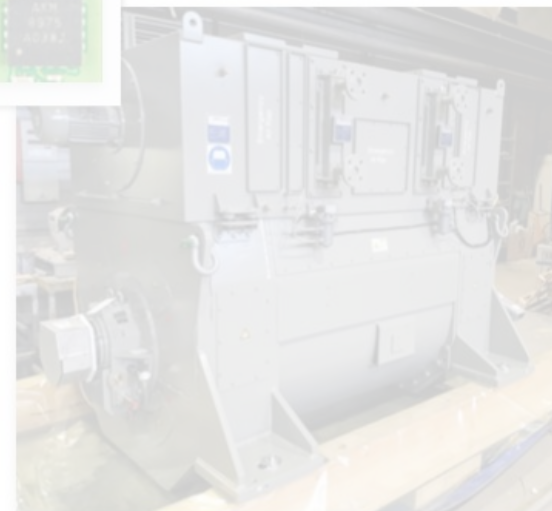
# Vision



# Sound



# Vibration



# KWS (KeyWord Spotting)

## Introduction

# Personal Assistant



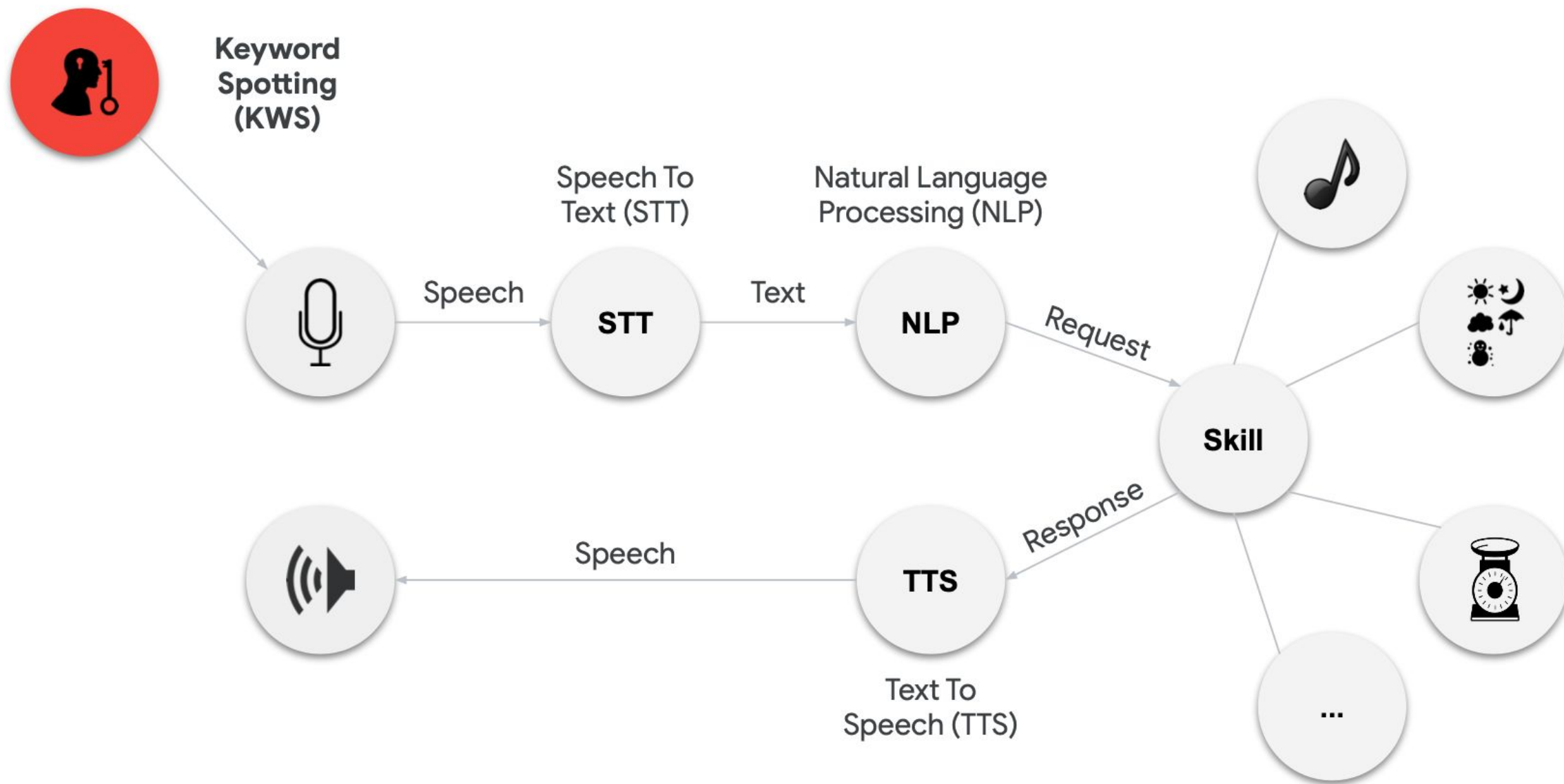
# Keyword Spotting v. General Speech Recognition

- **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

<sup>\*</sup> Single **W**ord **T**arget **S**egment

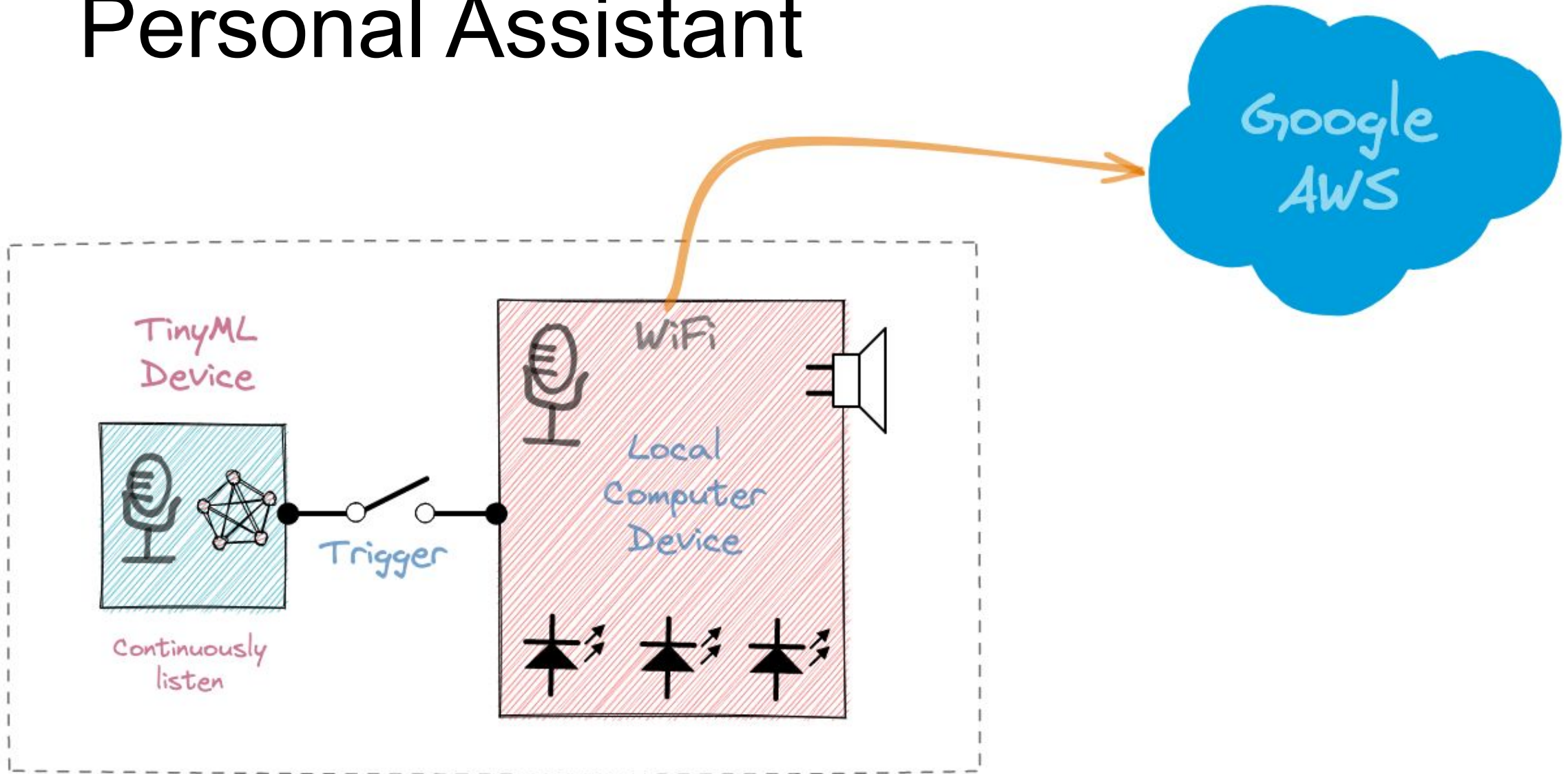
- **General ASR**<sup>\*</sup> still requires **larger, power-hungry models**
  - But it can run on mobile devices (offline dictation on smartphones)

<sup>\*</sup> Automatic Speech Recognition

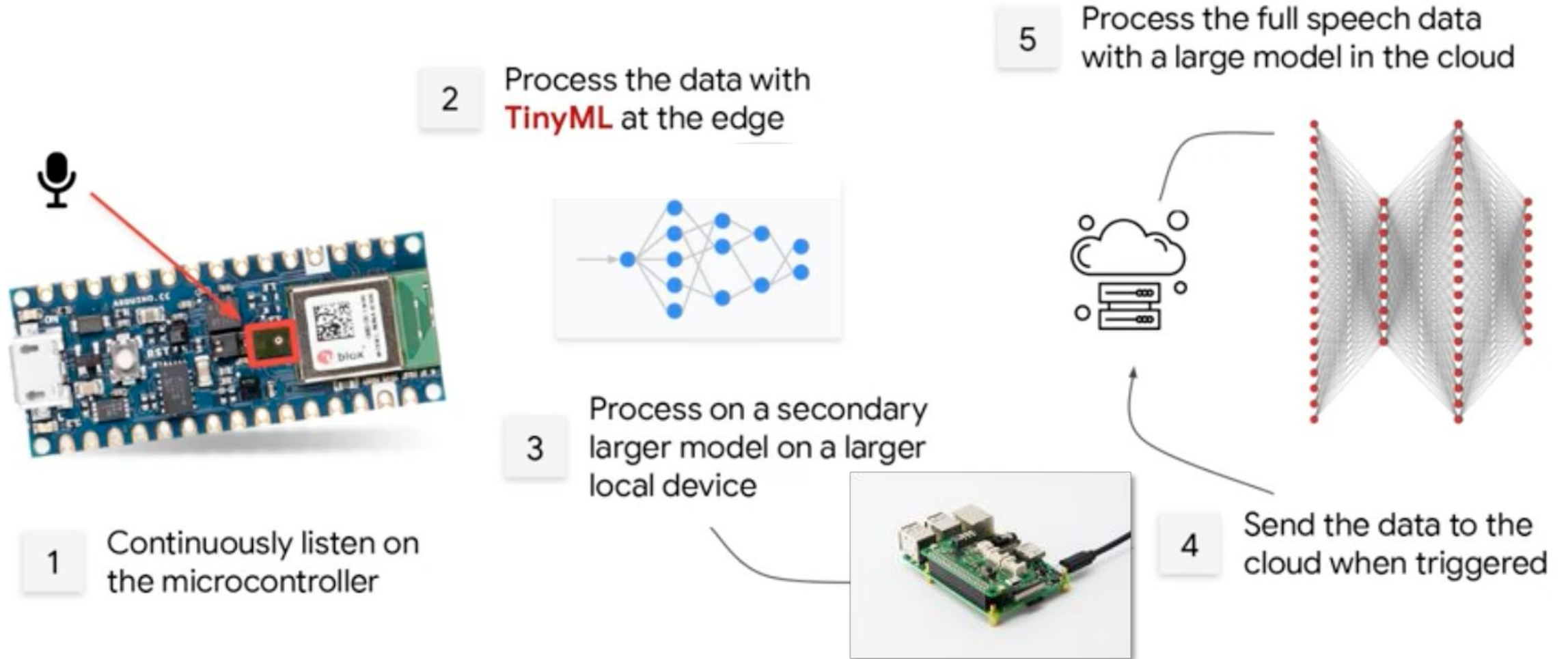




# Personal Assistant

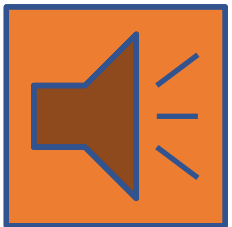
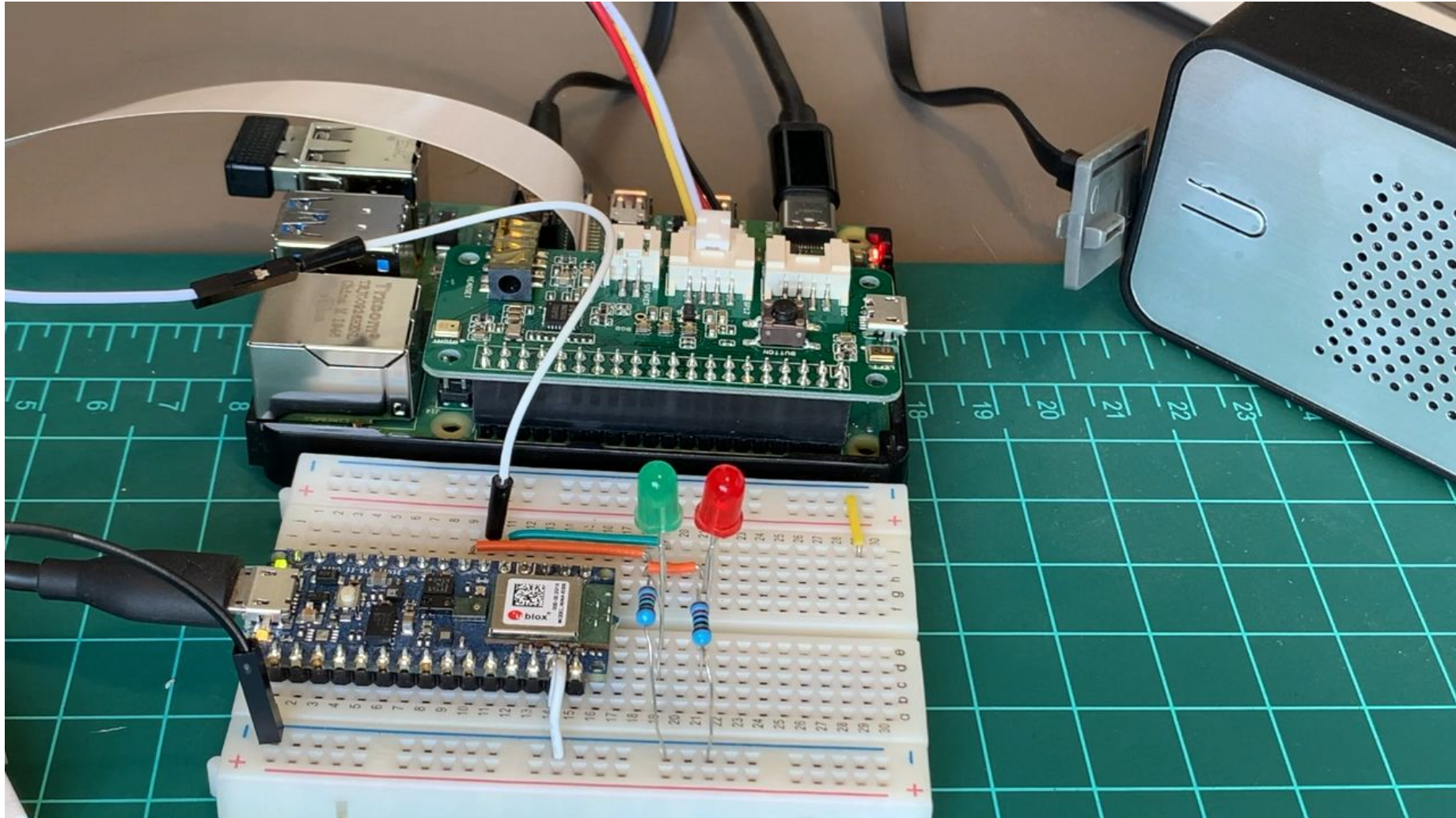


# “Cascade” Detection: multi-stage model





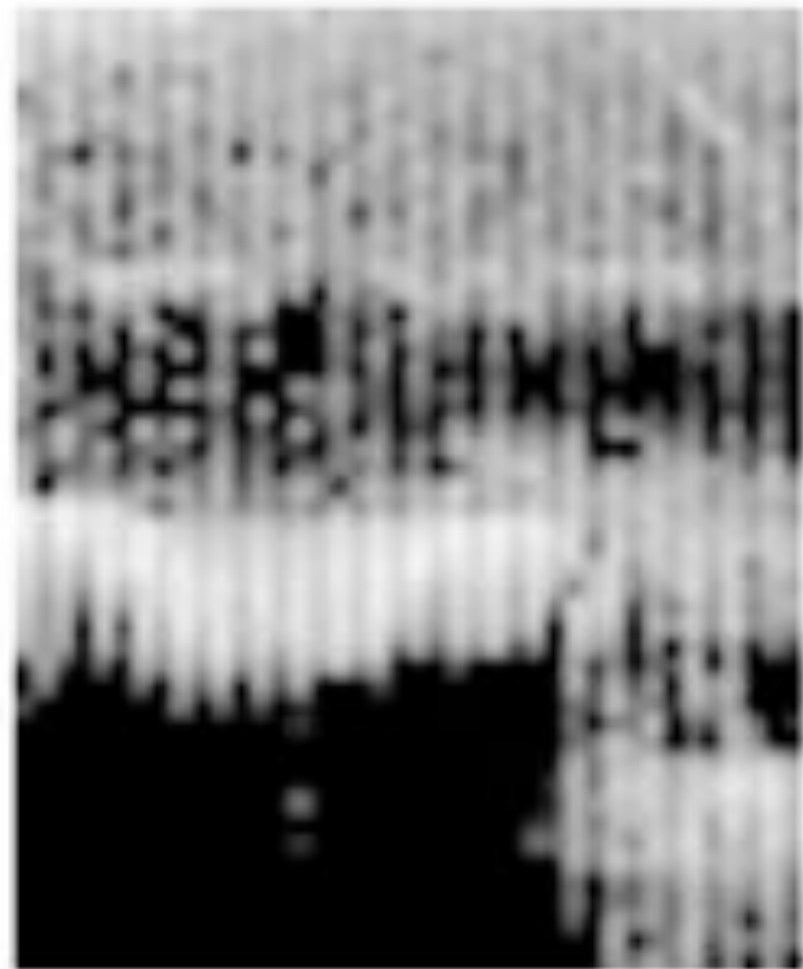
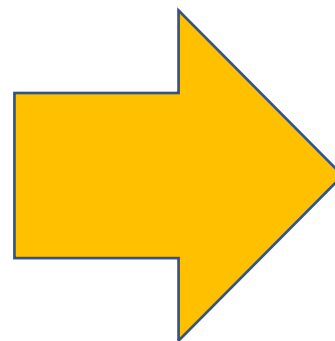
# KeyWord Spotting (KWS)



<https://mrobot.org/2021/01/27/building-an-intelligent-voice-assistant-from-scratch/>



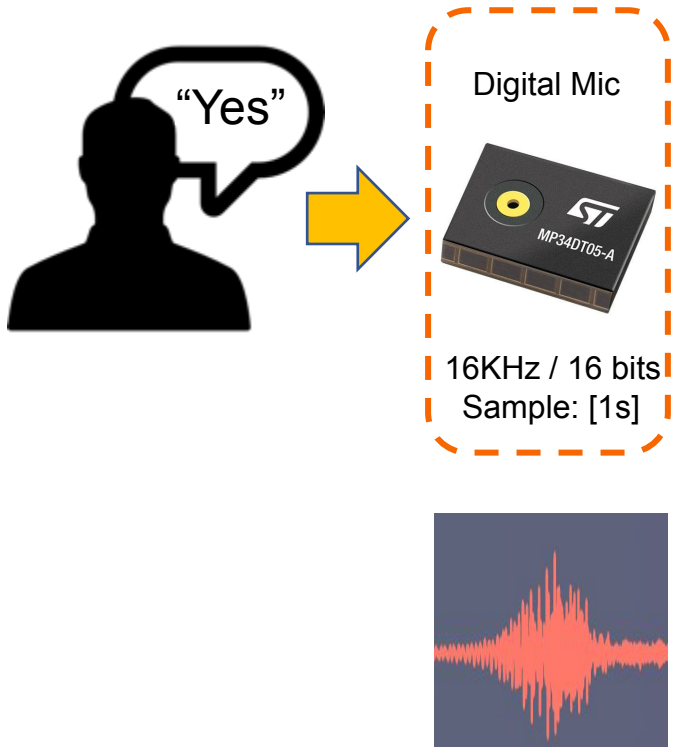
Sound



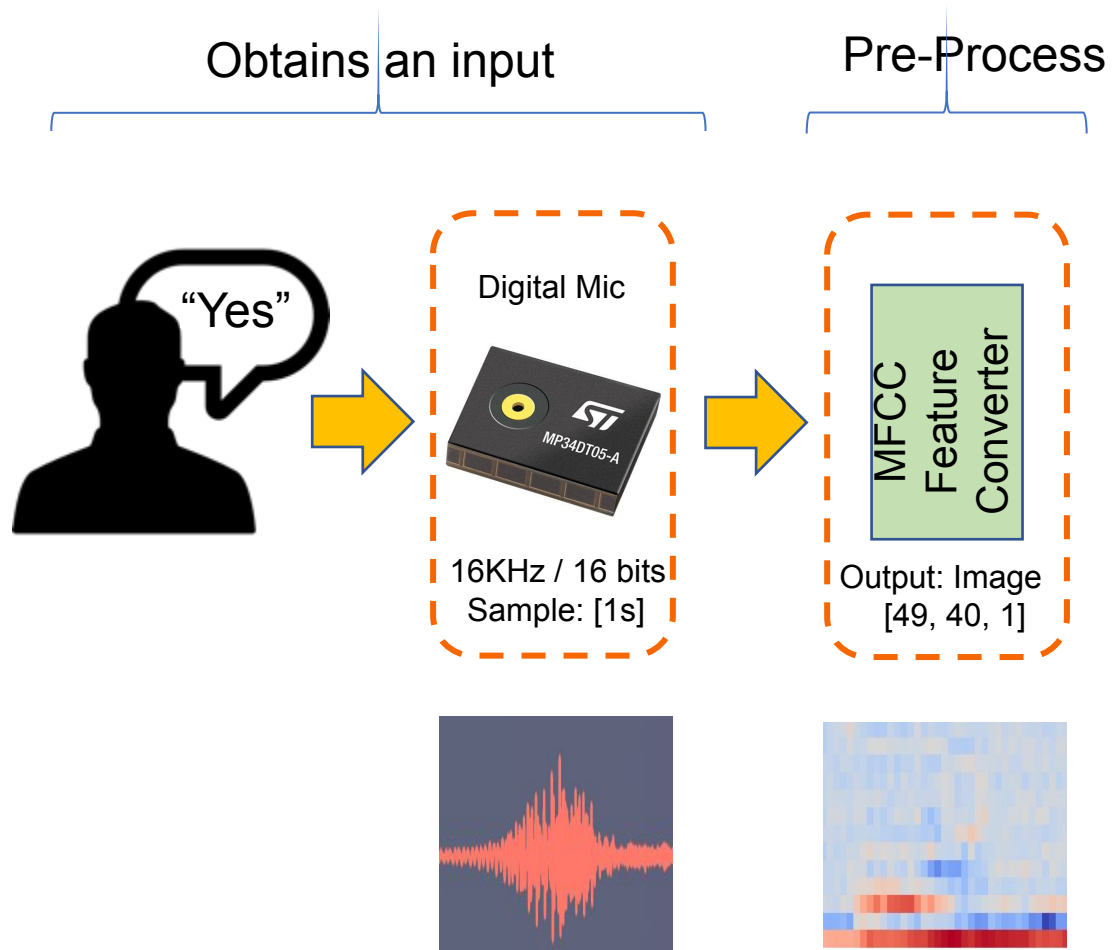
Image

# KeyWord Spotting (KWS) - Inference

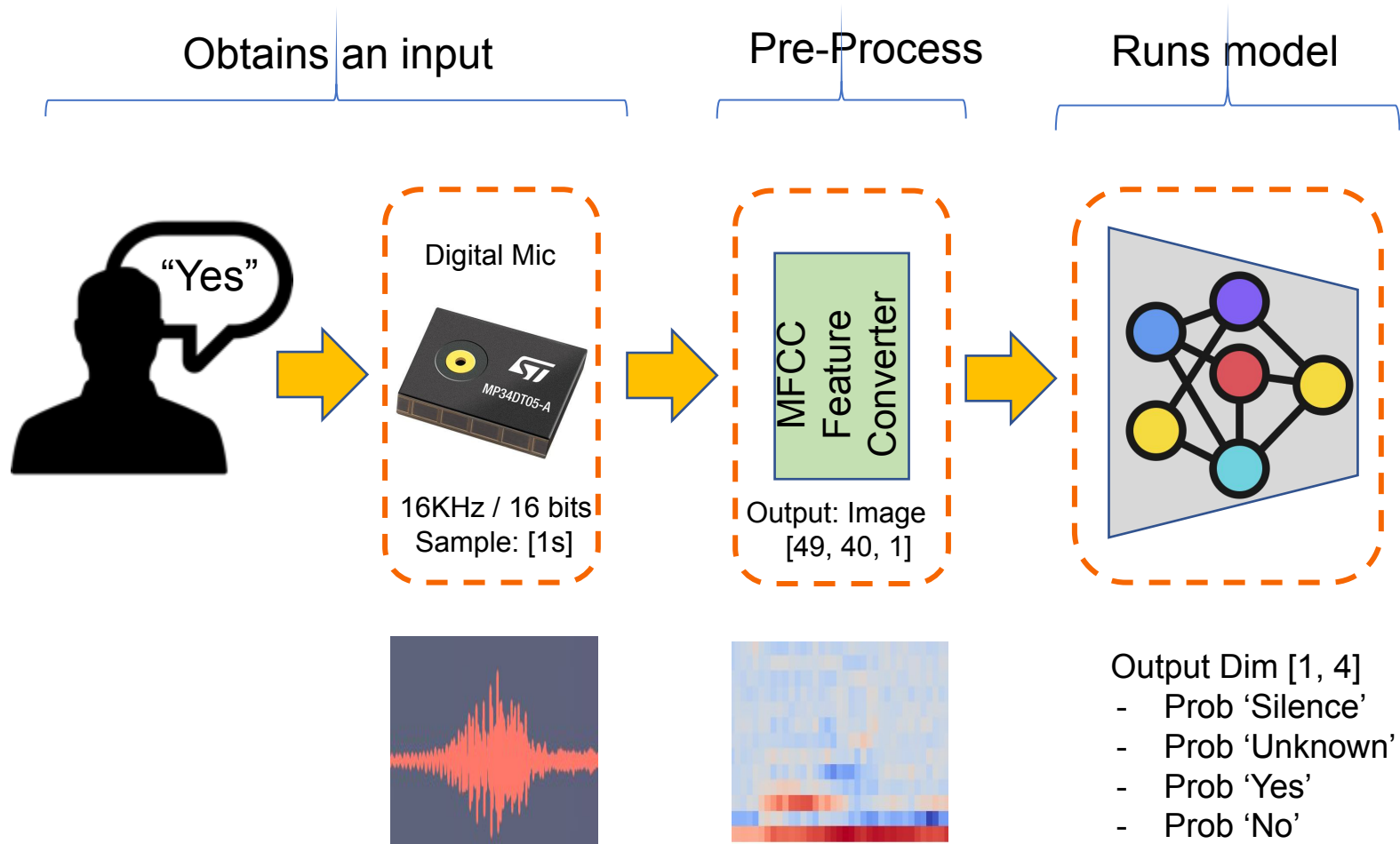
Obtains an input



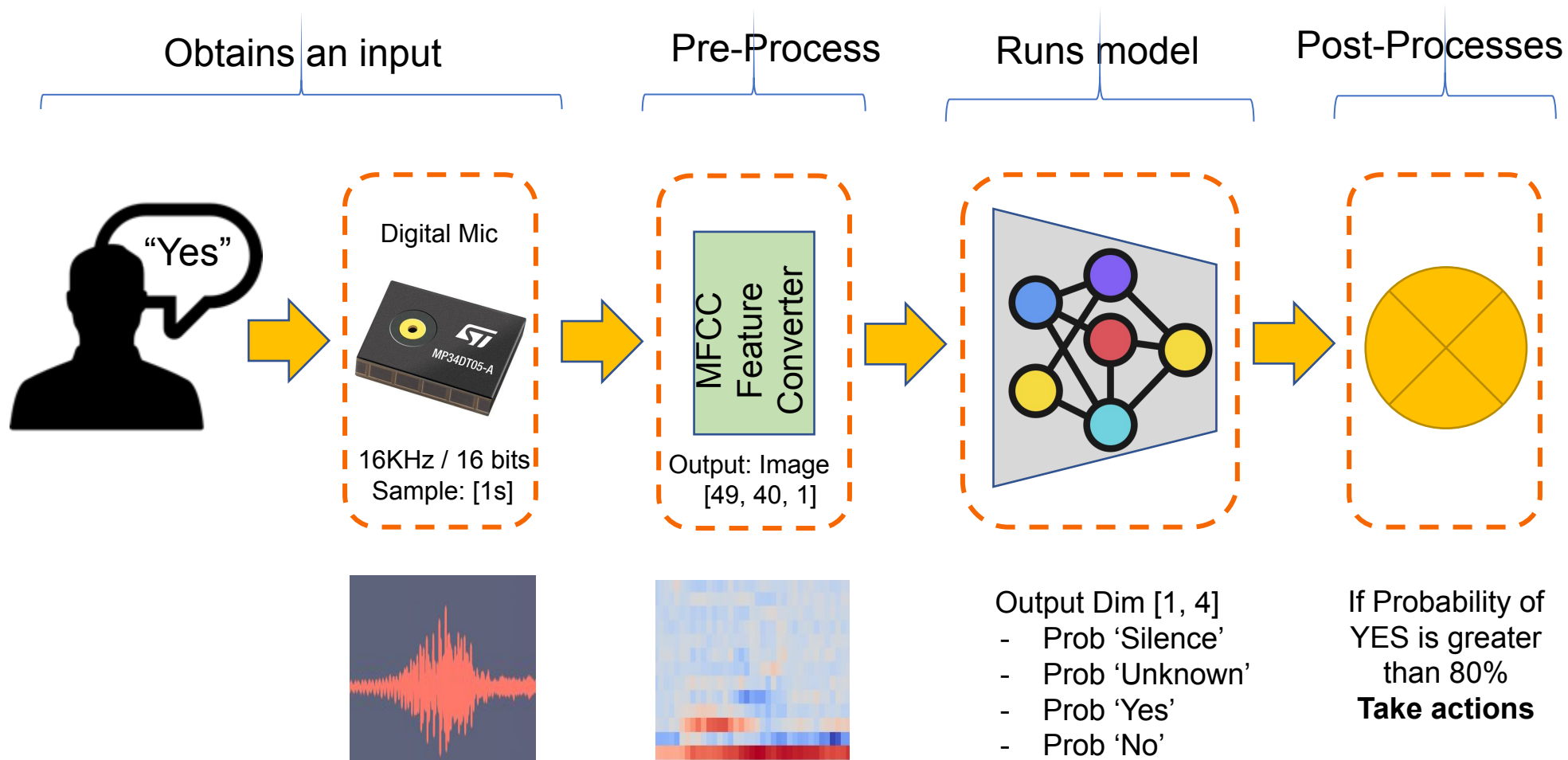
# KeyWord Spotting (KWS) - Inference



# KeyWord Spotting (KWS) - Inference

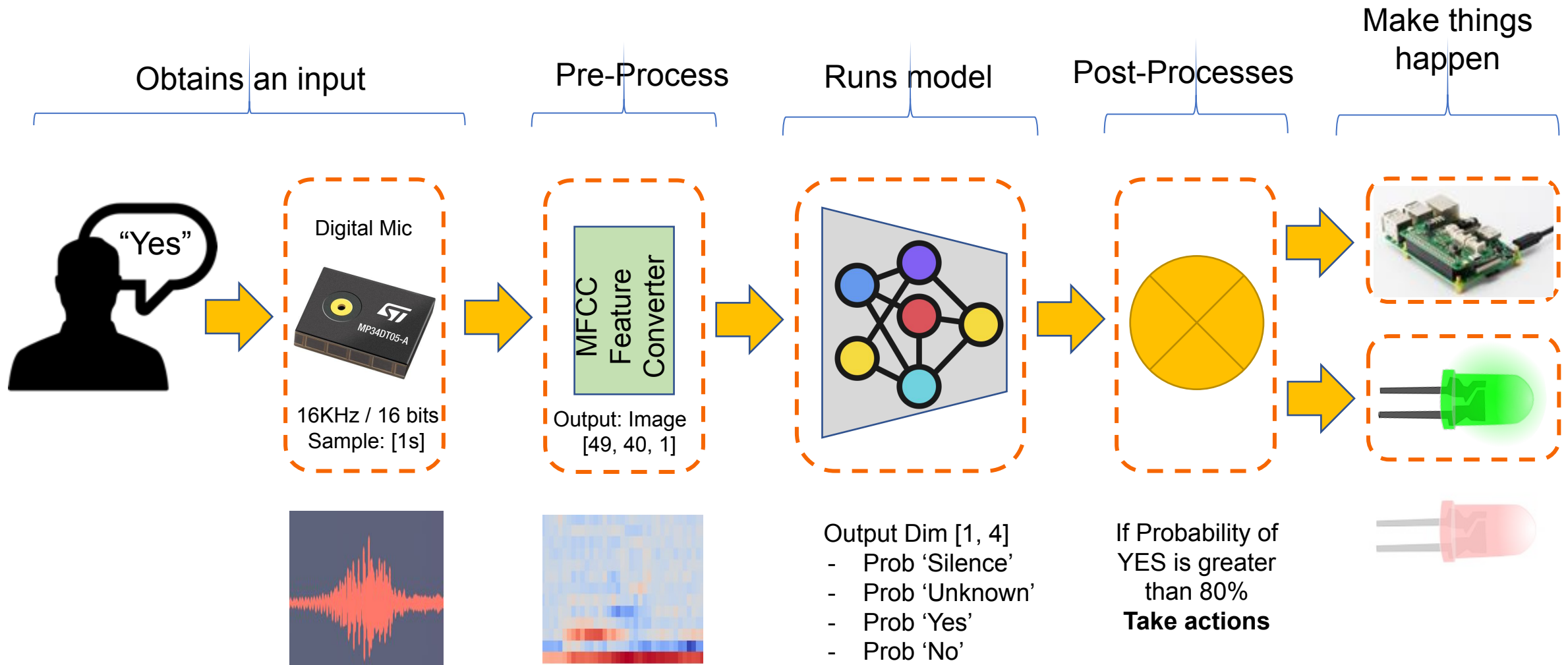


# KeyWord Spotting (KWS) - Inference

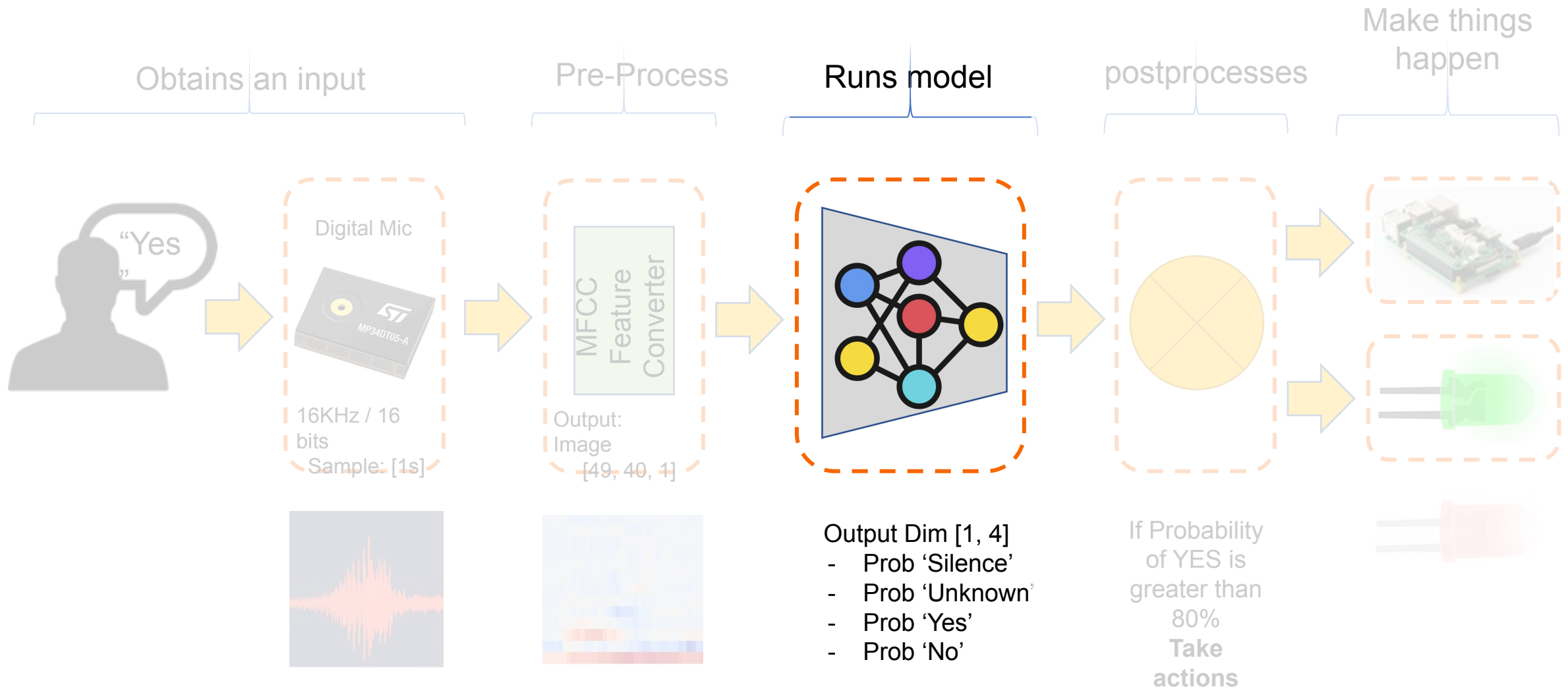




# KeyWord Spotting (KWS) - Inference



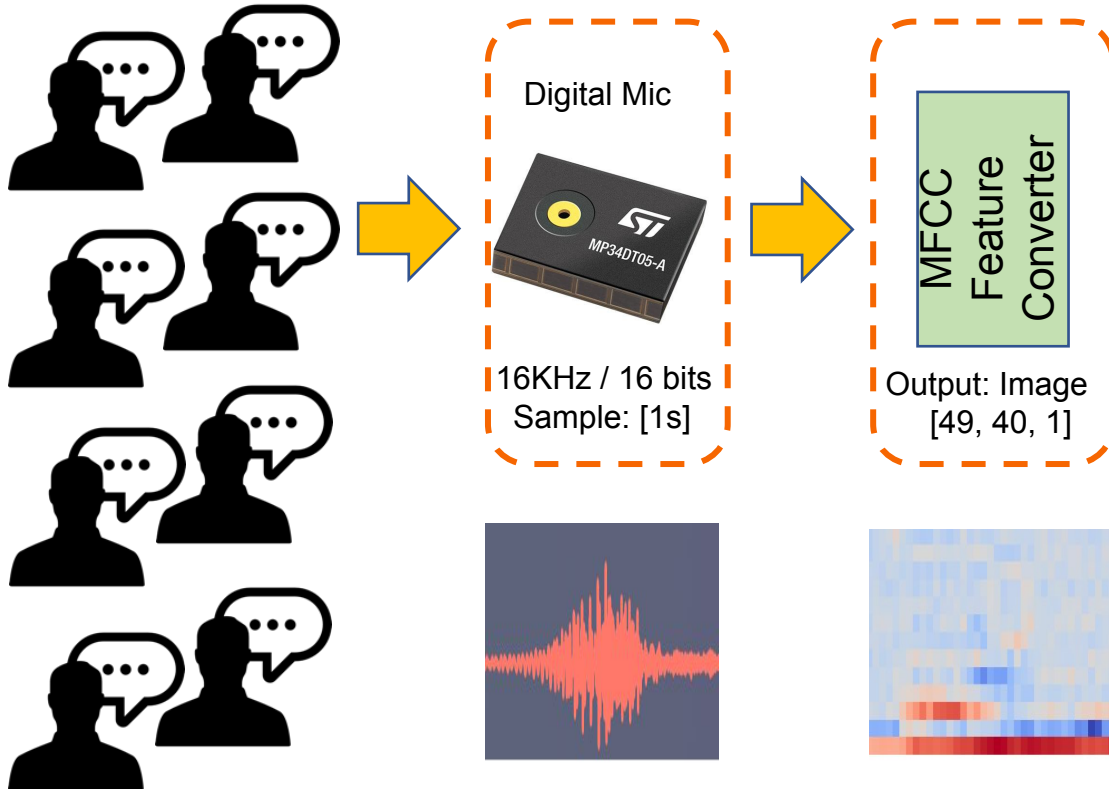
# KeyWord Spotting (KWS) - Model



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

Obtains data

Pre-Process



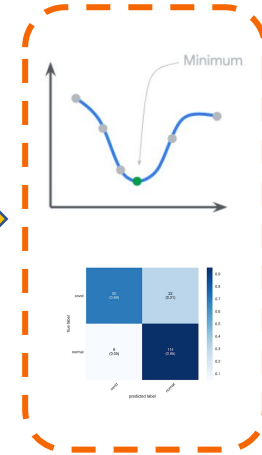
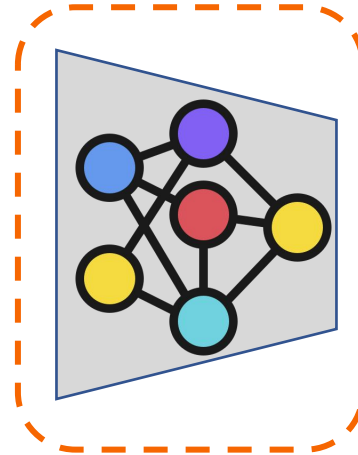
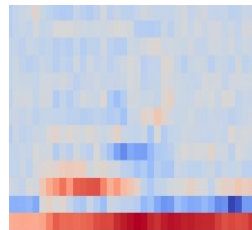
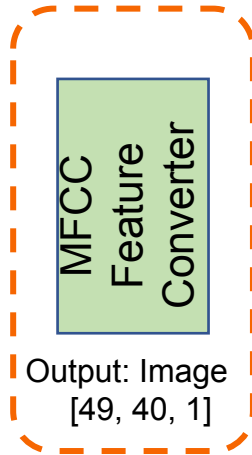
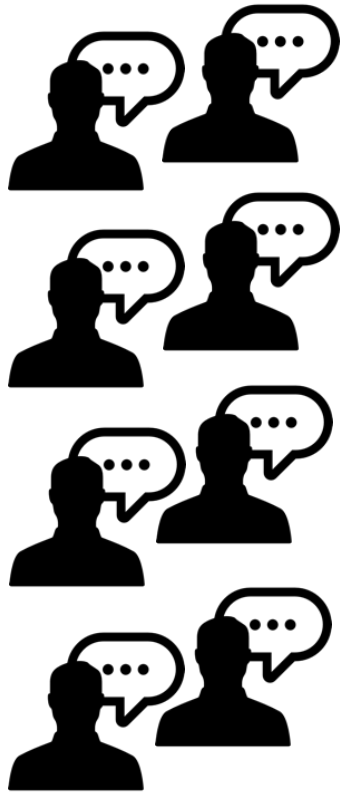
# KeyWord Spotting (KWS) – Create Model (Training)

Obtains data

Pre-Process

Train model

Evaluate Model



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

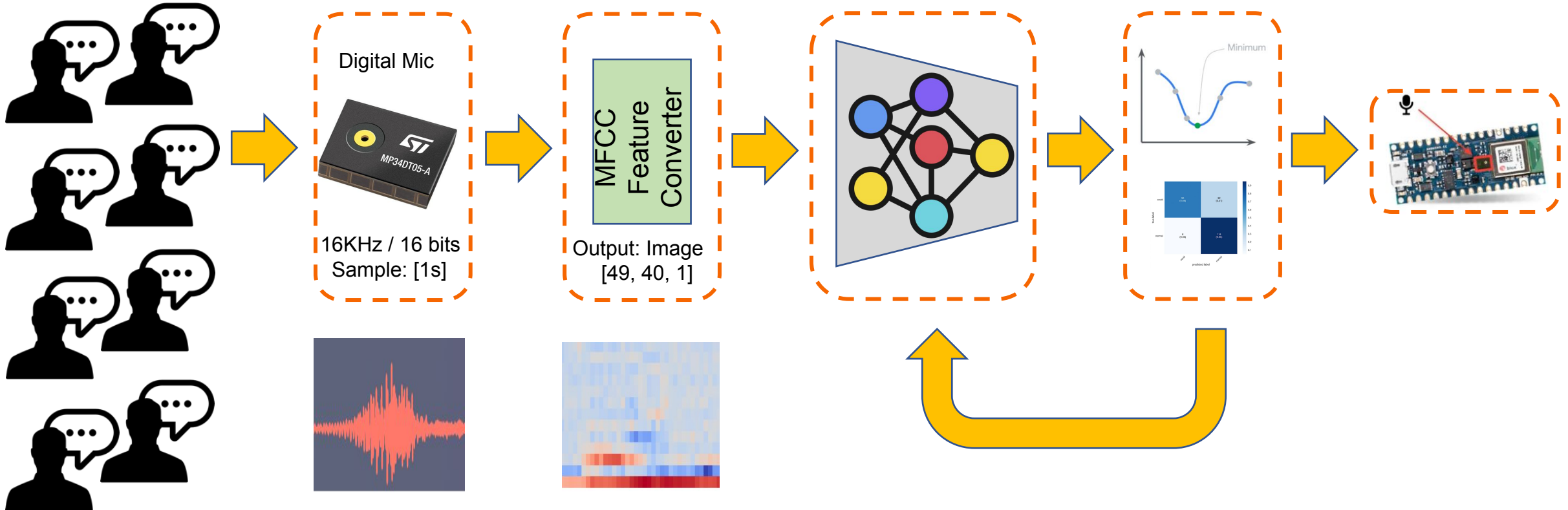
Obtains data

Pre-Process

Train model

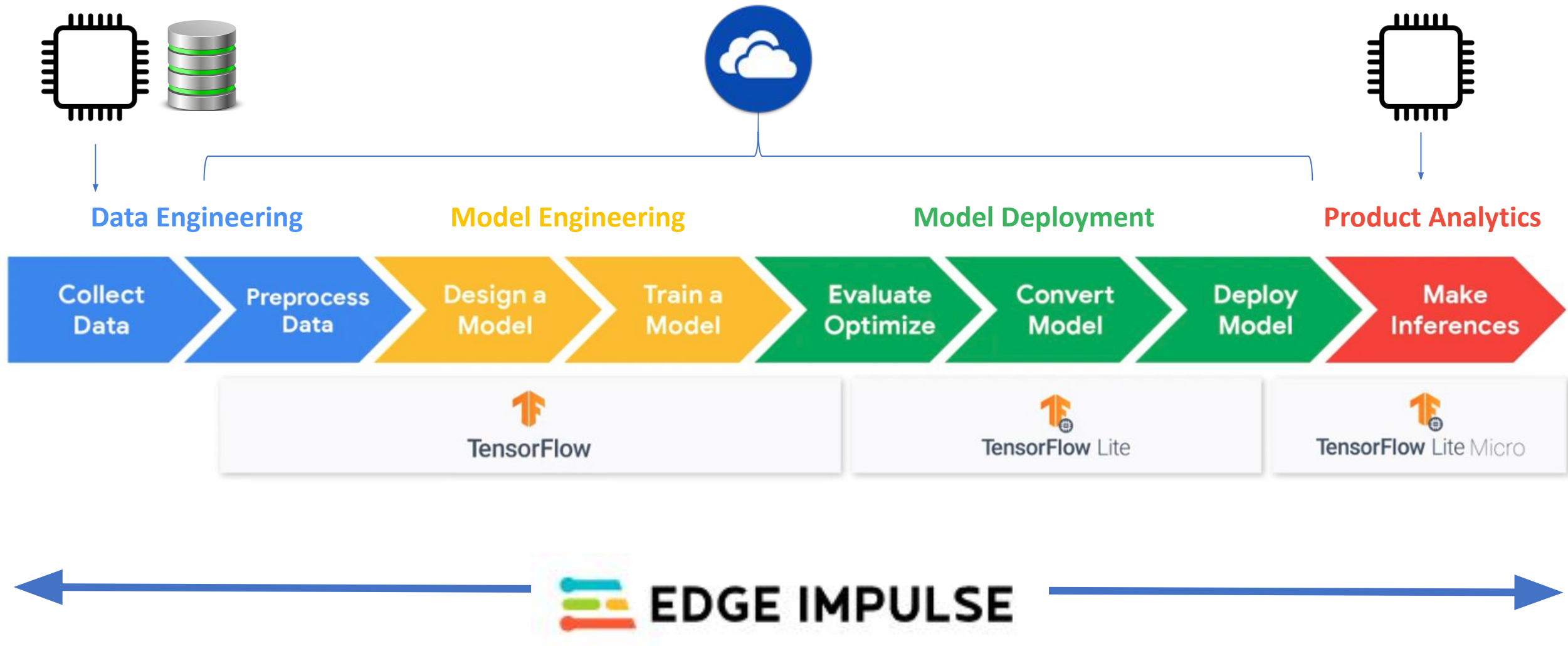
Evaluate Model

Deploy



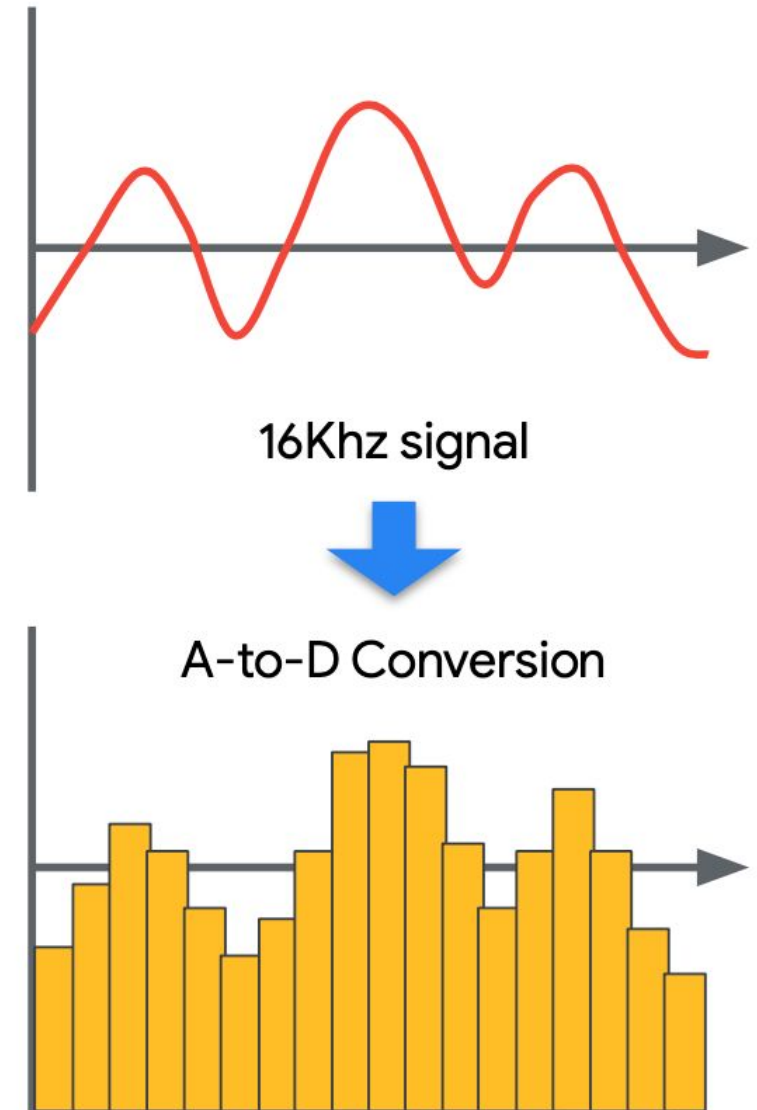
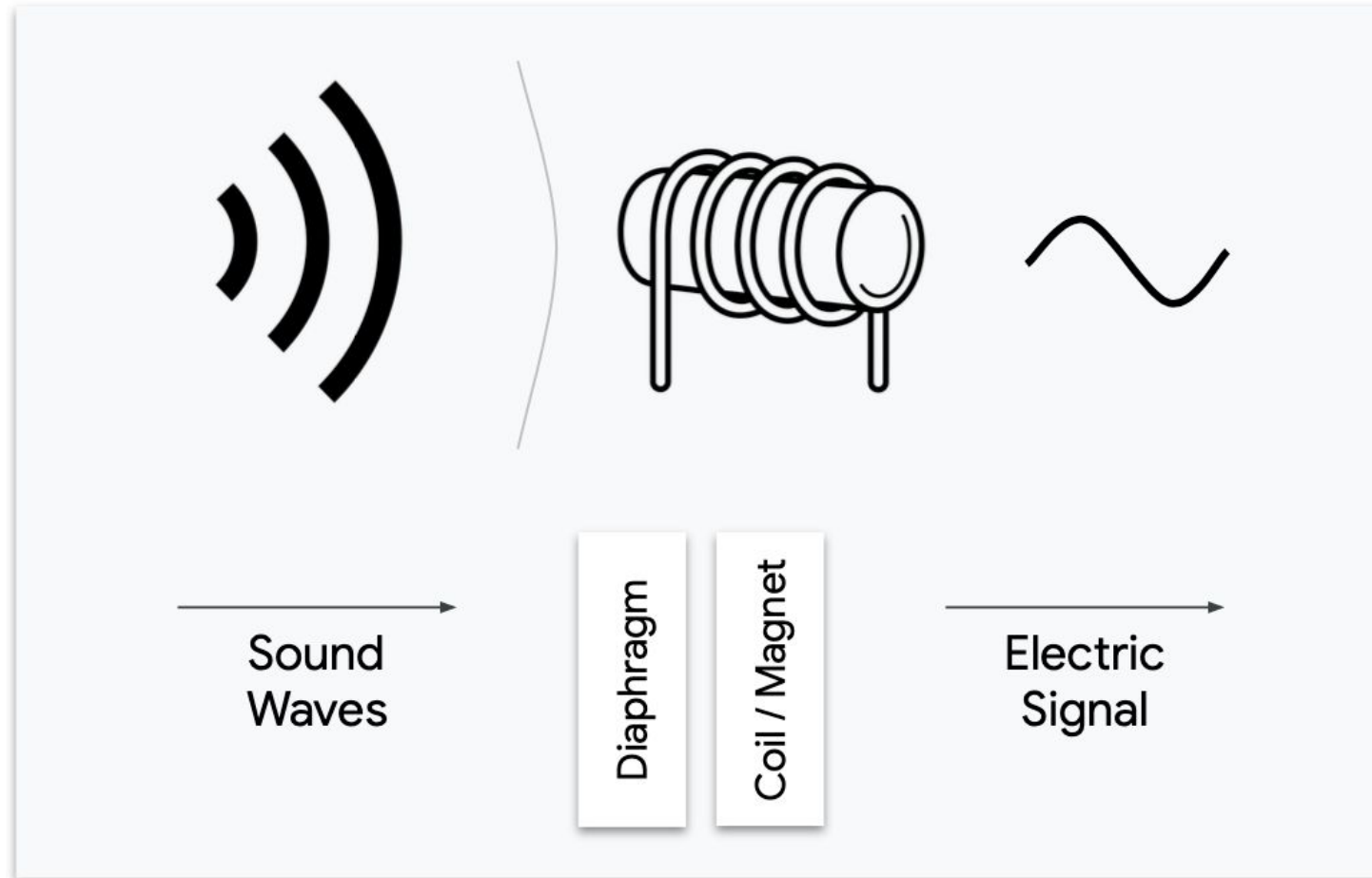
# KWS Data Collection & Pre-Processing



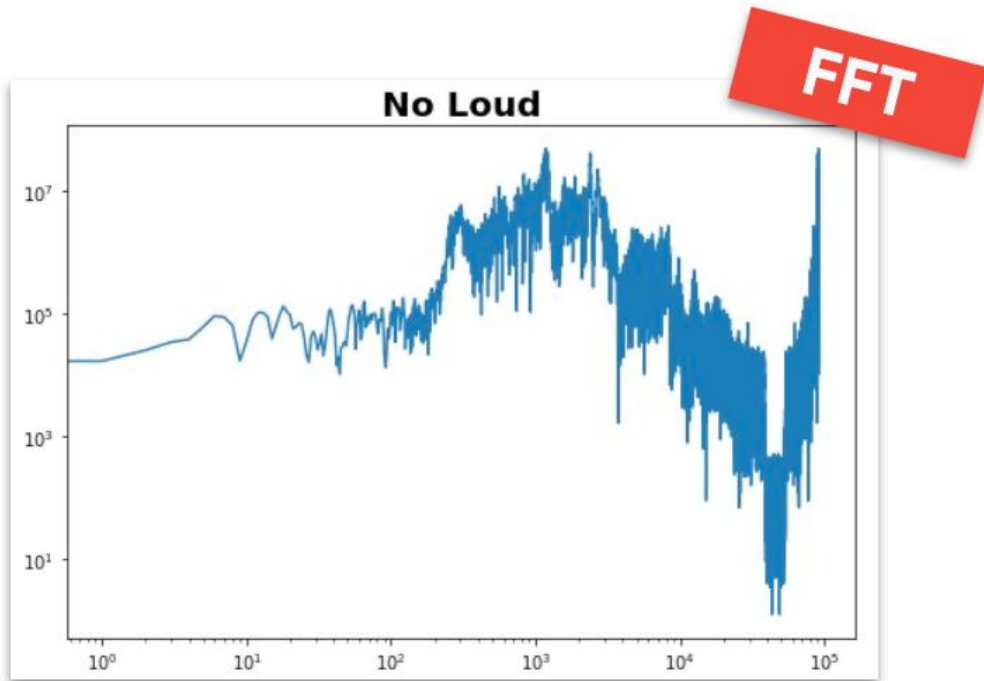




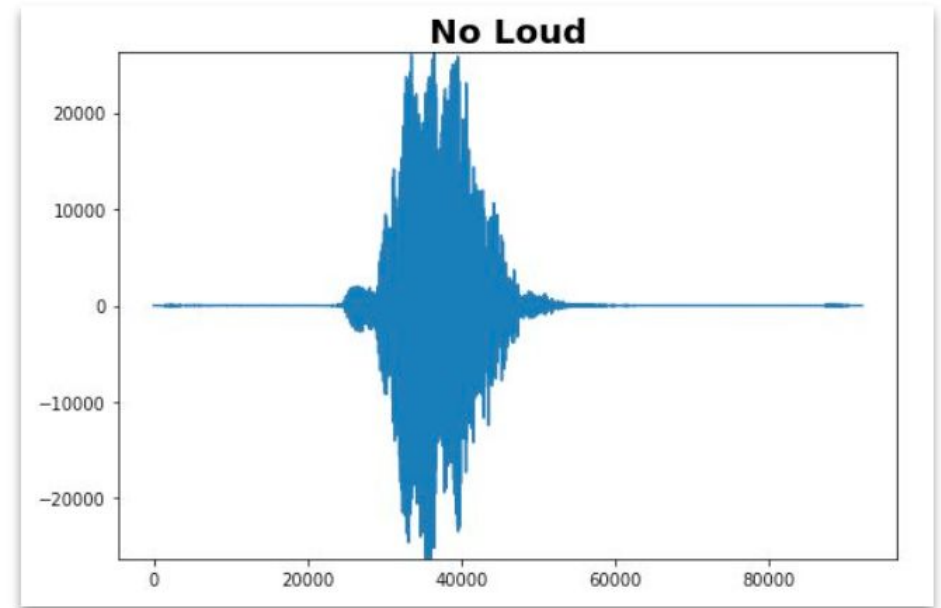
# Sensor Data



# Signal Components?



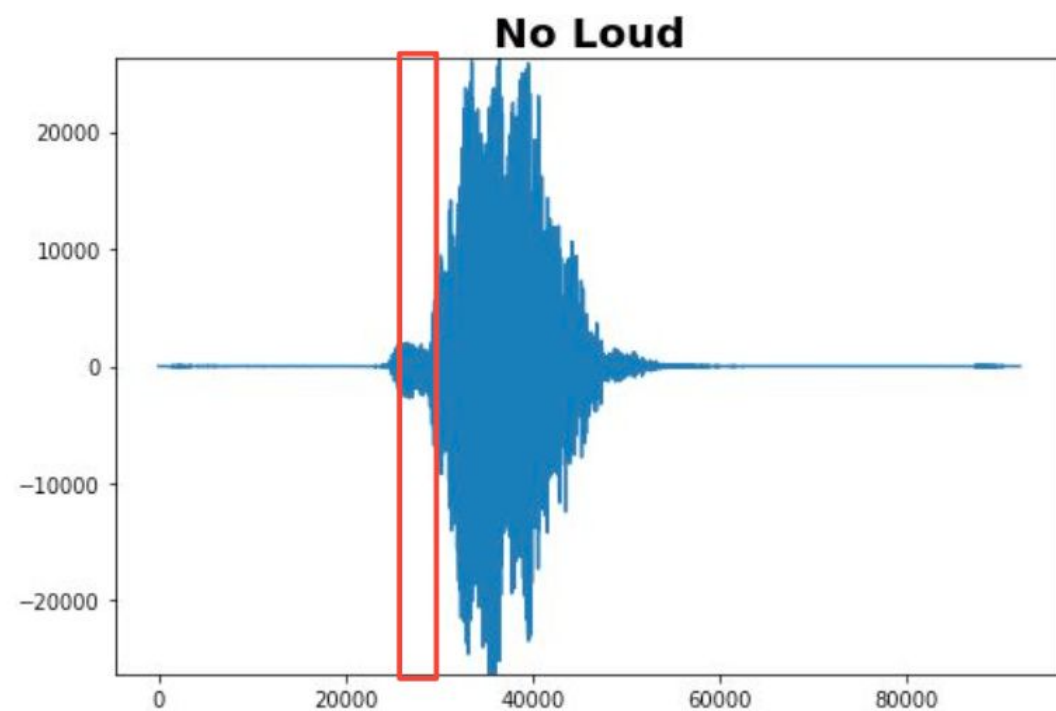
=



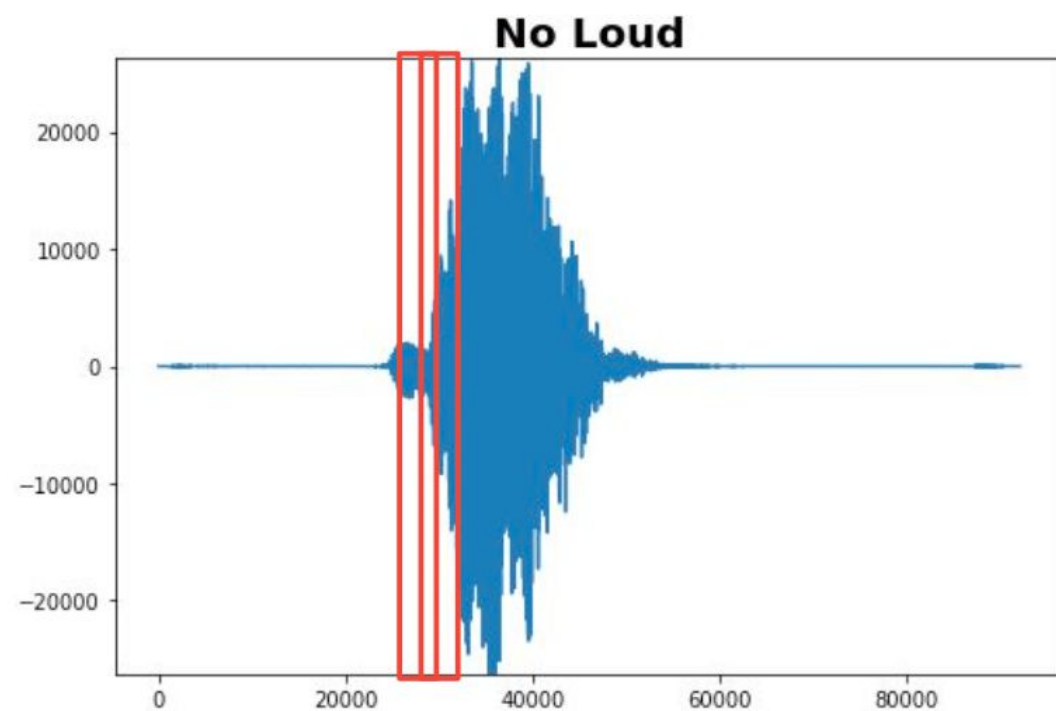
Frequency

Time

# Data Preprocessing

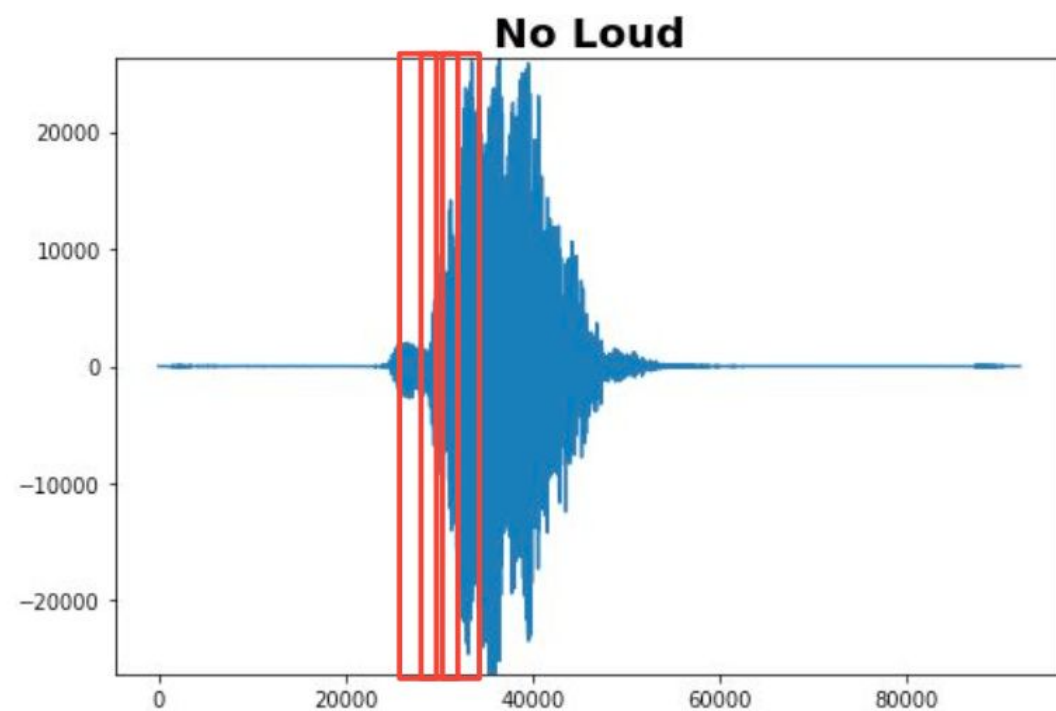


# Data Preprocessing

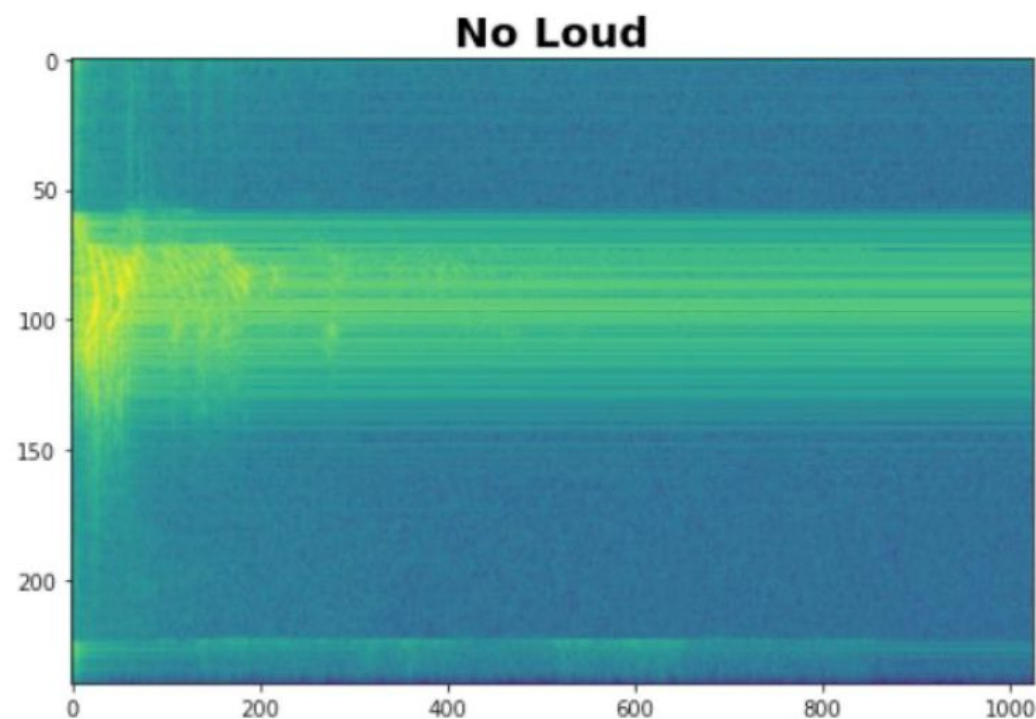
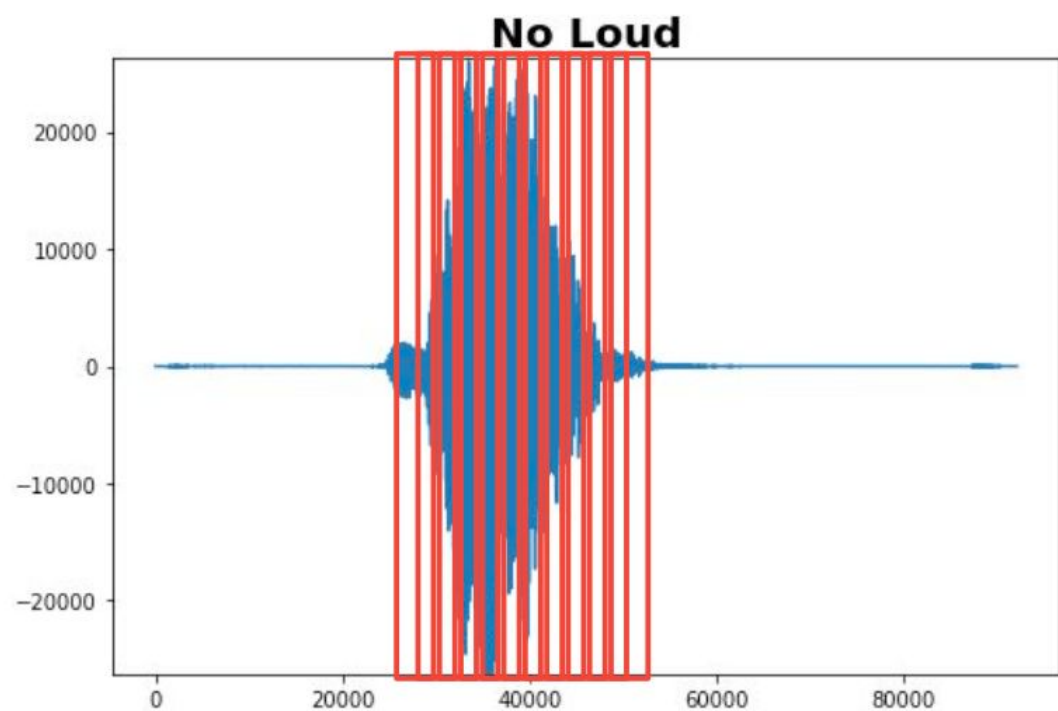




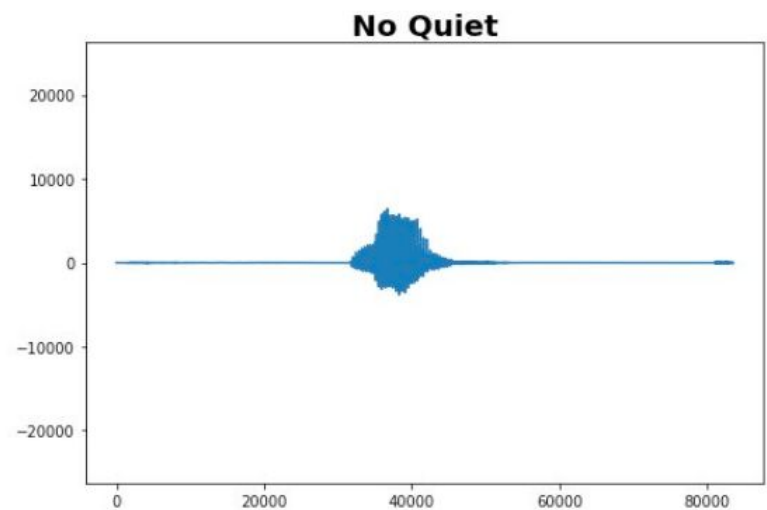
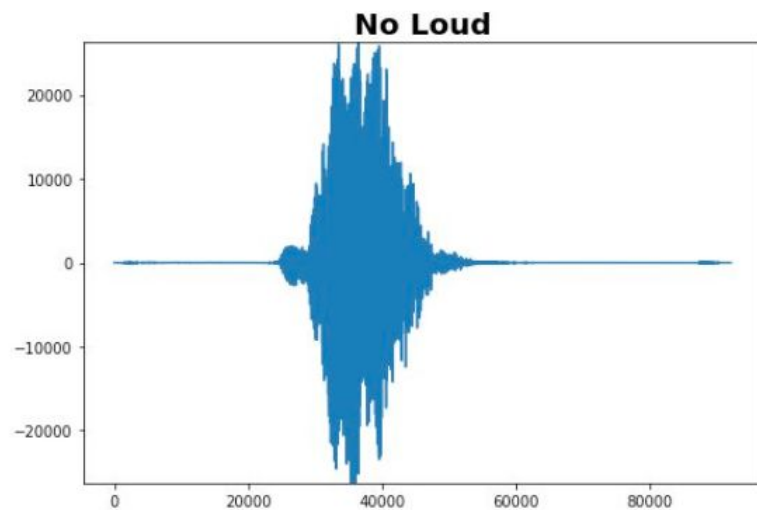
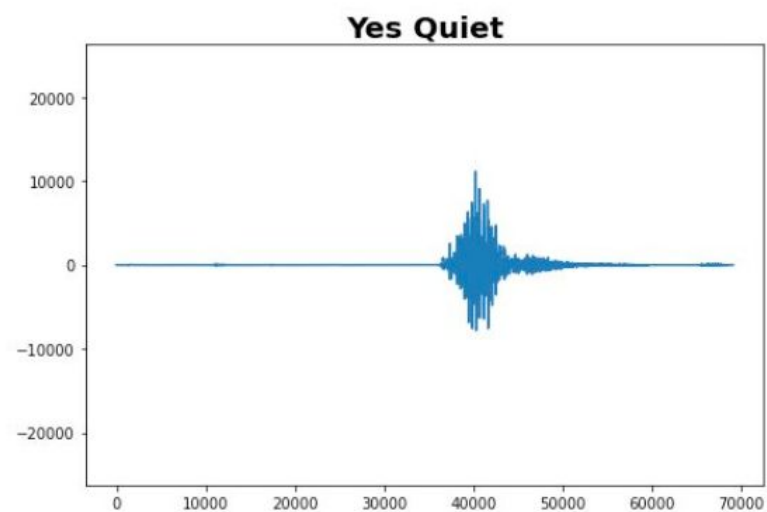
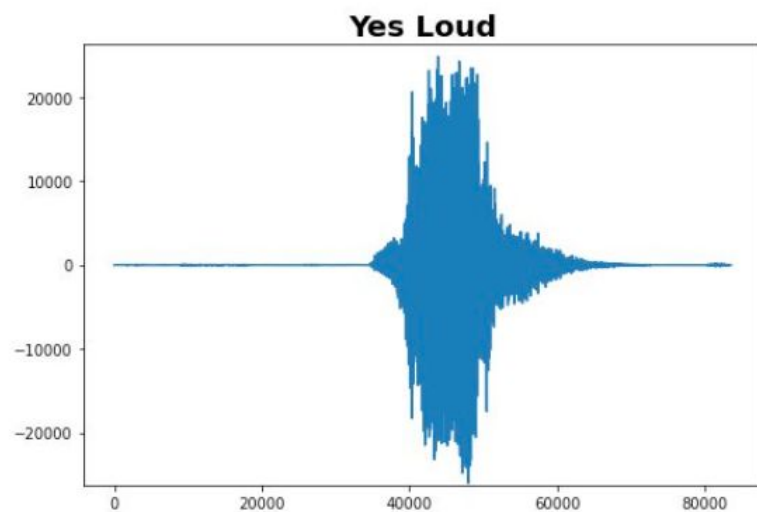
# Data Preprocessing



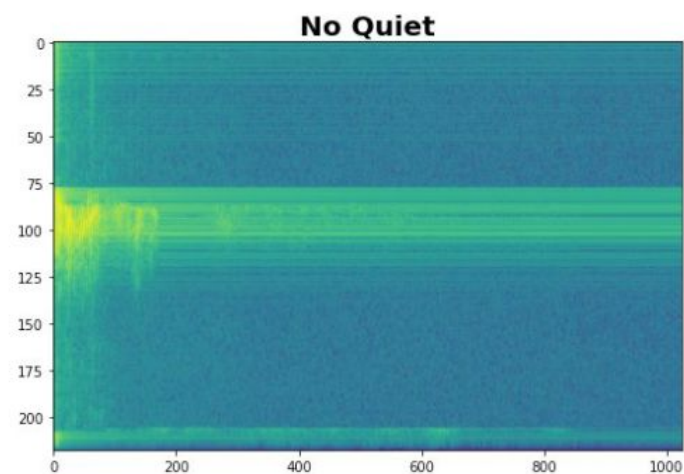
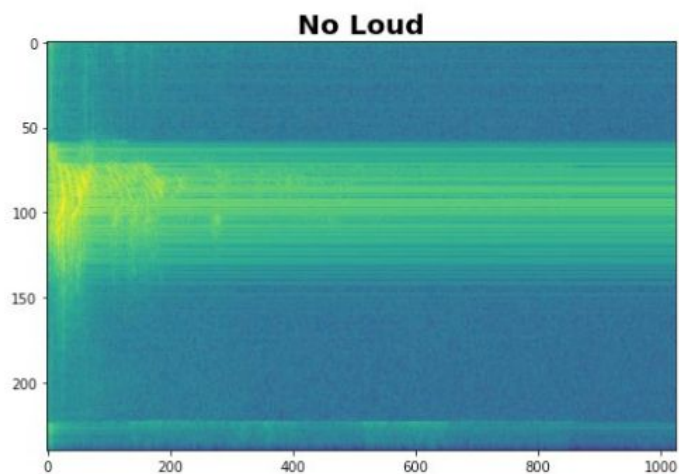
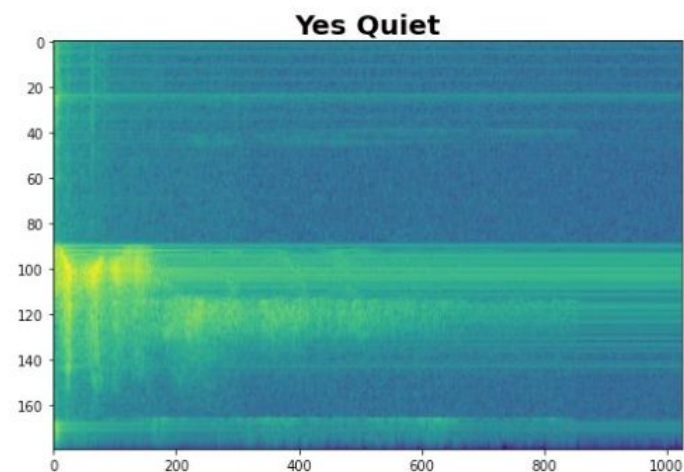
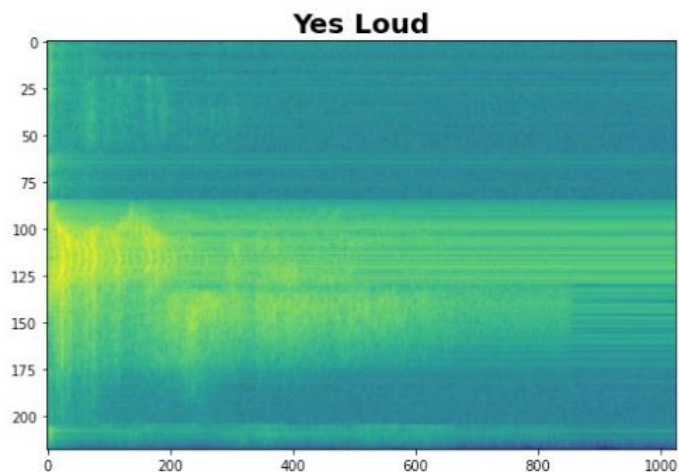
# Data Preprocessing: Spectrograms



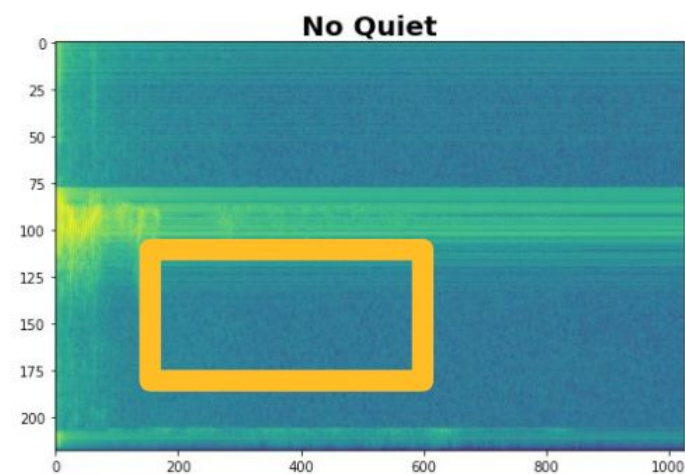
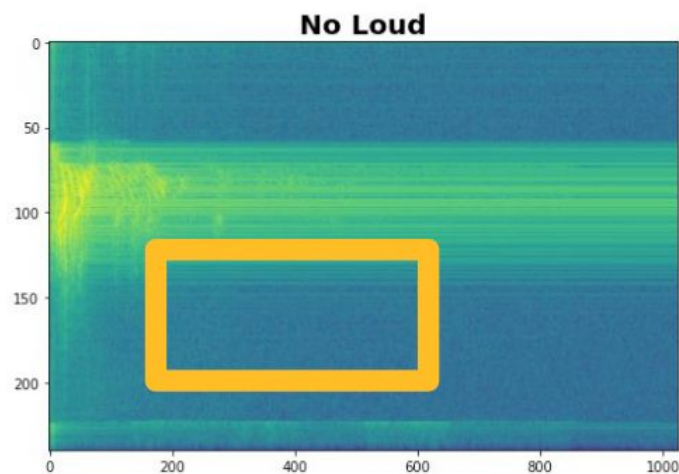
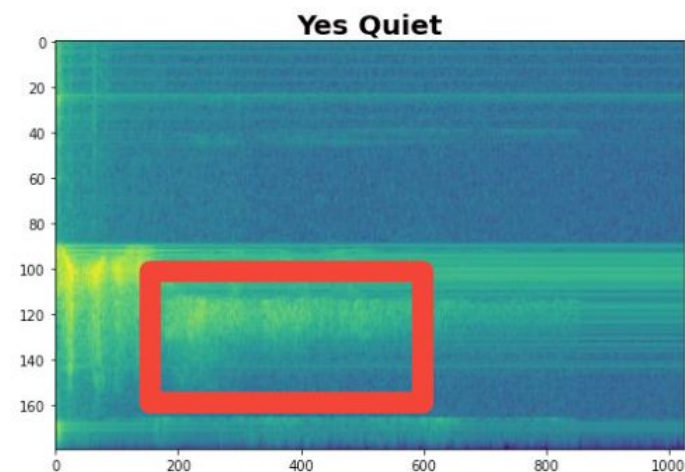
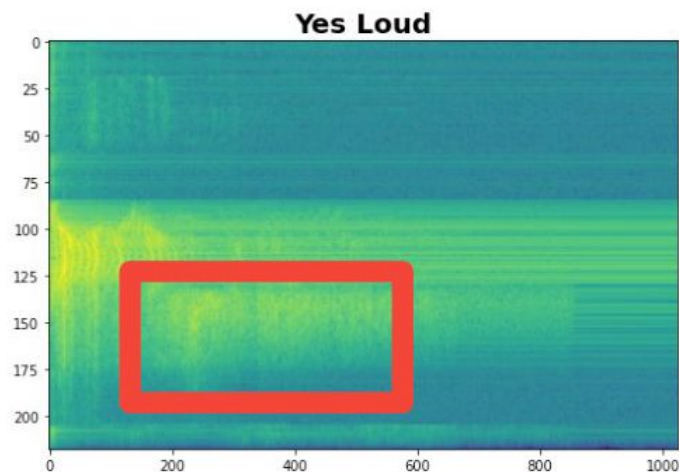
# Data Preprocessing: Spectrograms



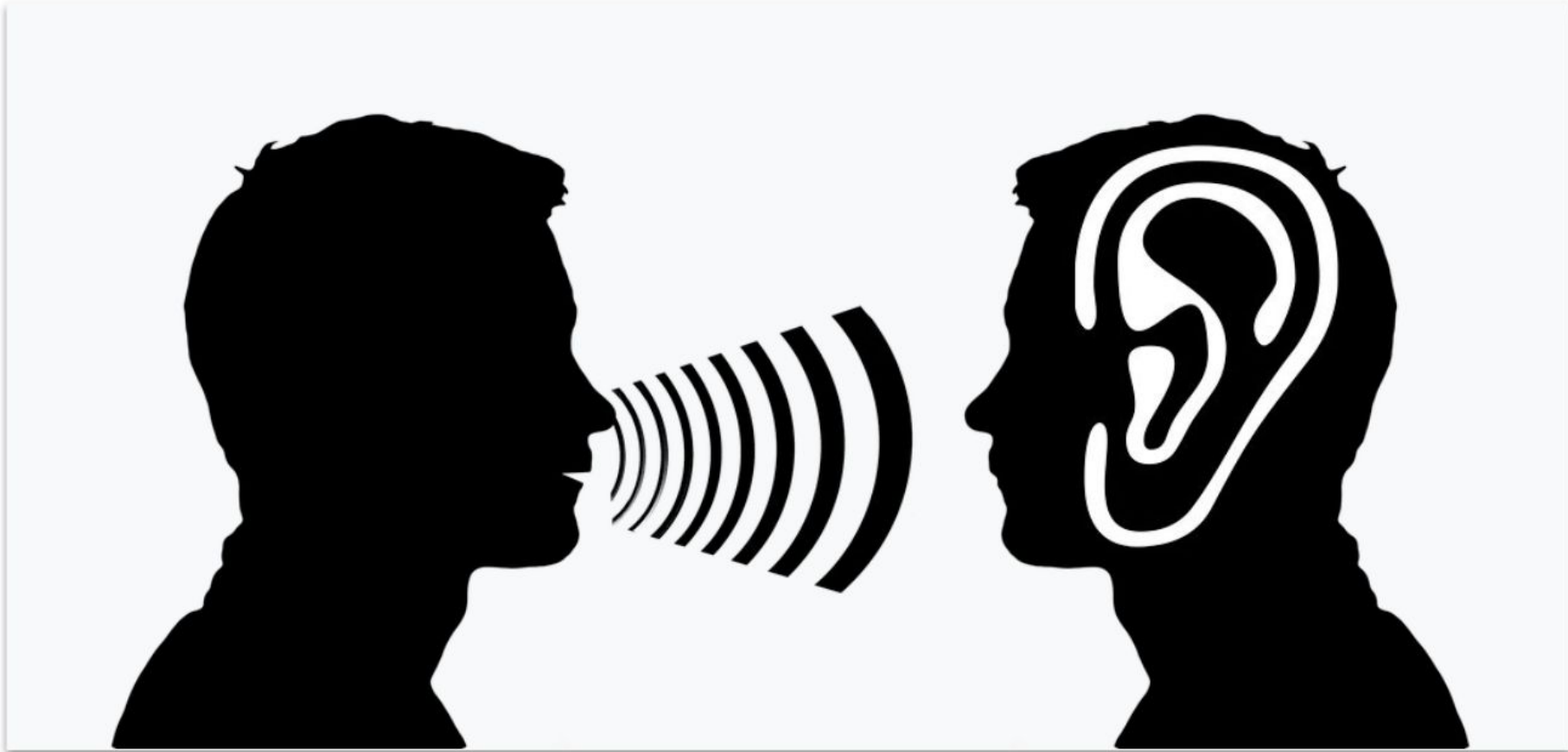
# Data Preprocessing: Spectrograms



# Data Preprocessing: Spectrograms



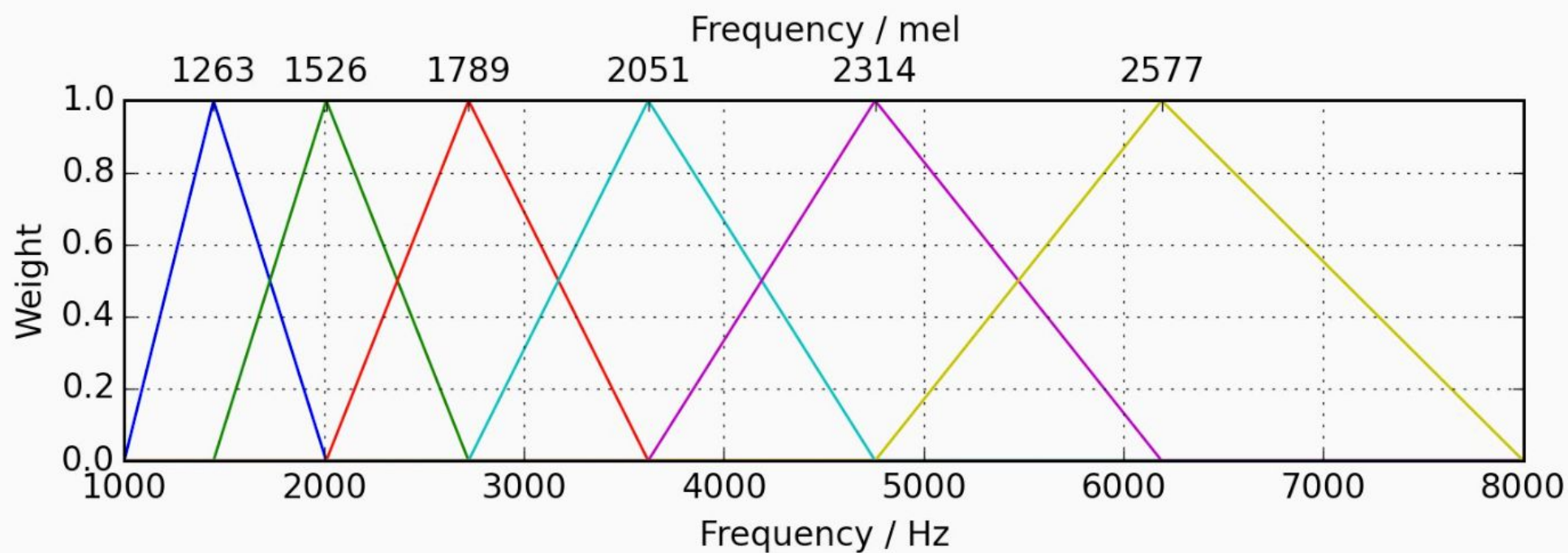




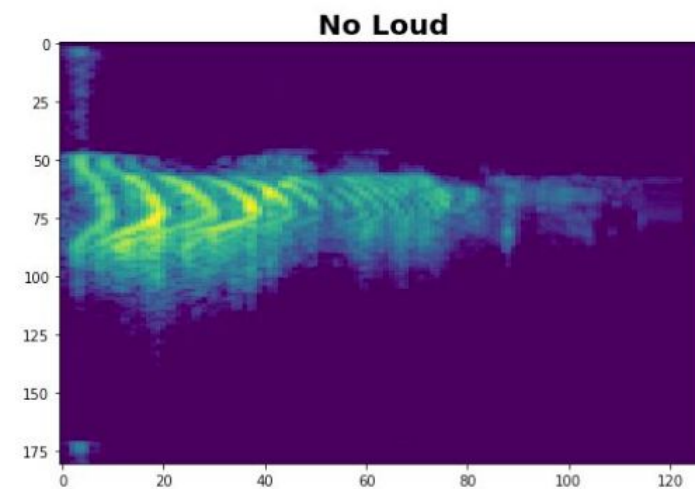
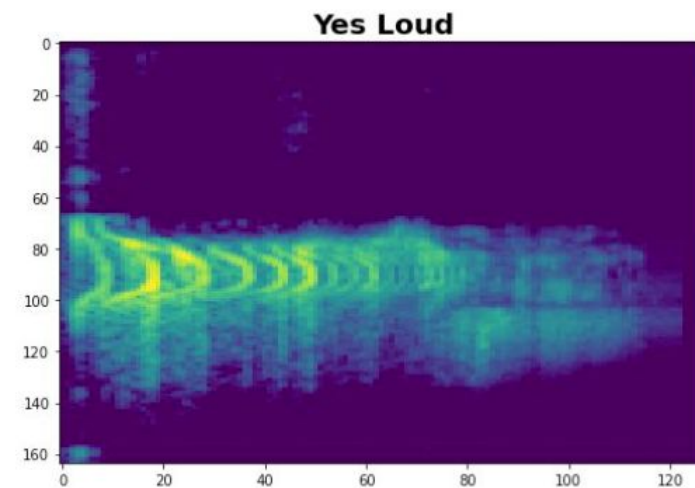
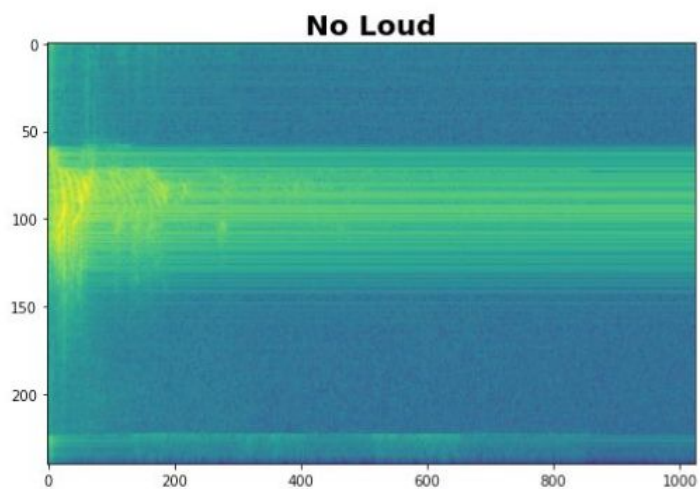
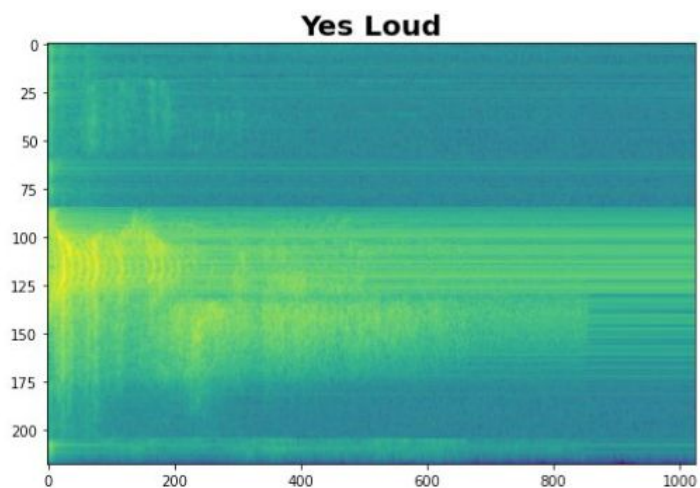
The **lower band frequencies** is much more crisper to us



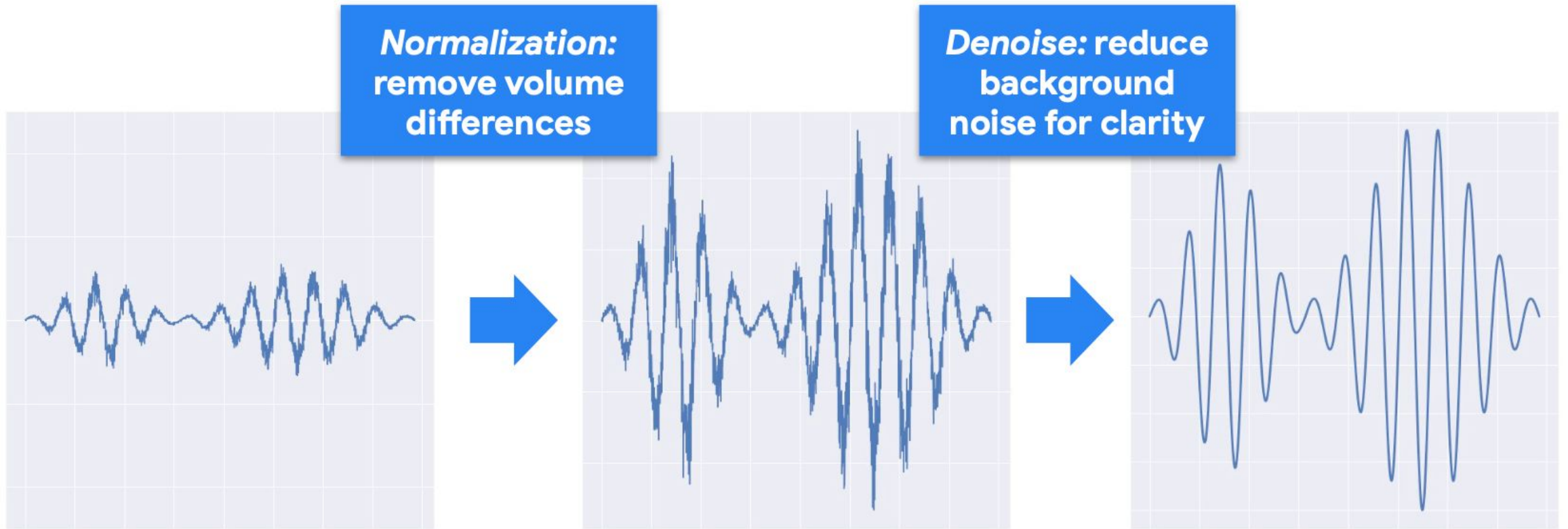
# Mel Filterbanks



# Spectrograms v. MFCCs



# Additional Feature Engineering

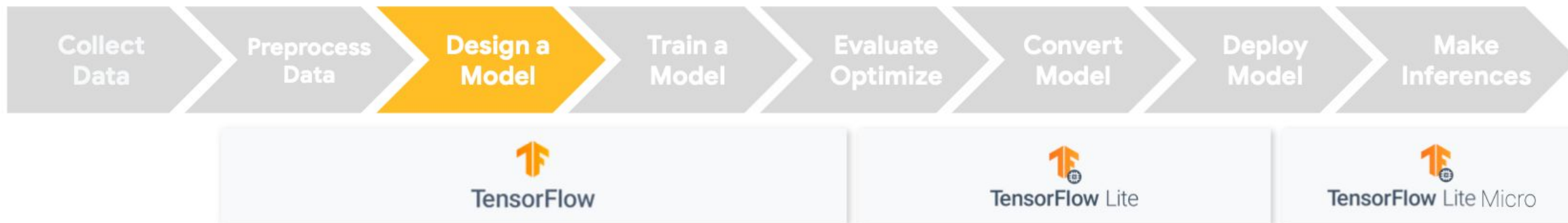


# Spectrograms and MFCCs

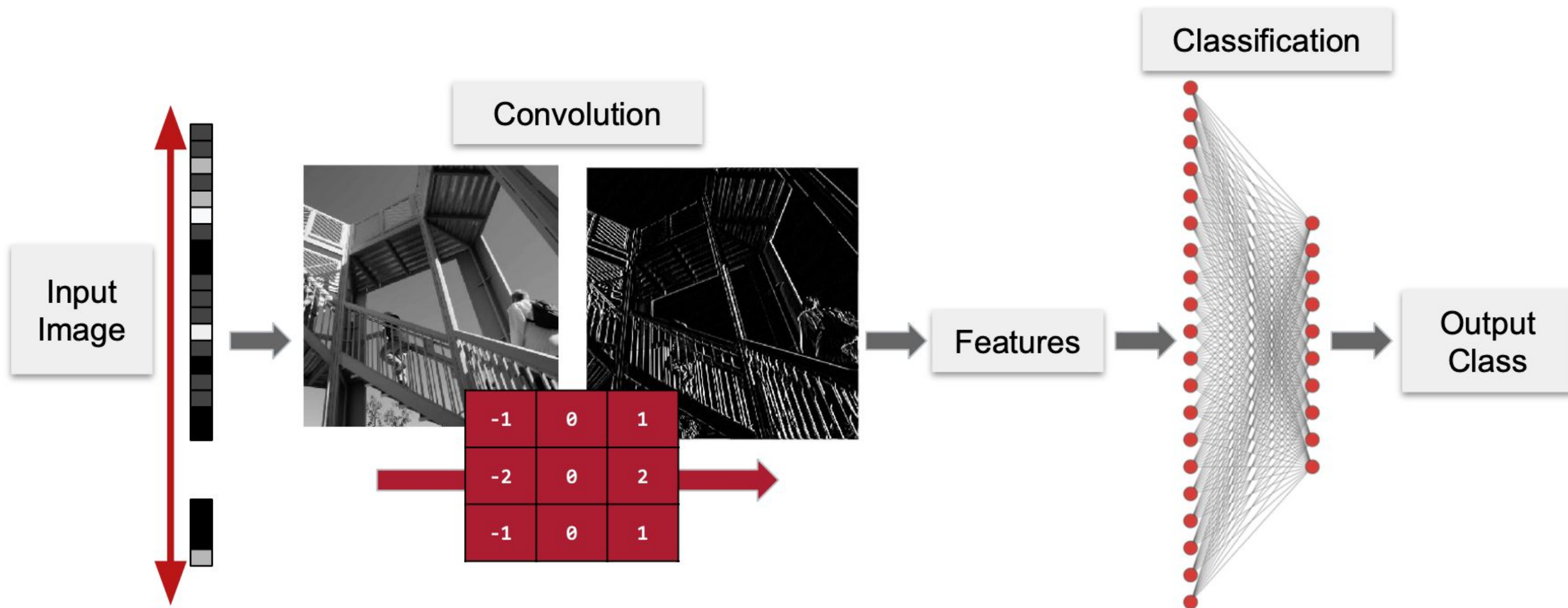
[SpectrogramsMFCCs.ipynb](#)



# A Keyword Spotting Model

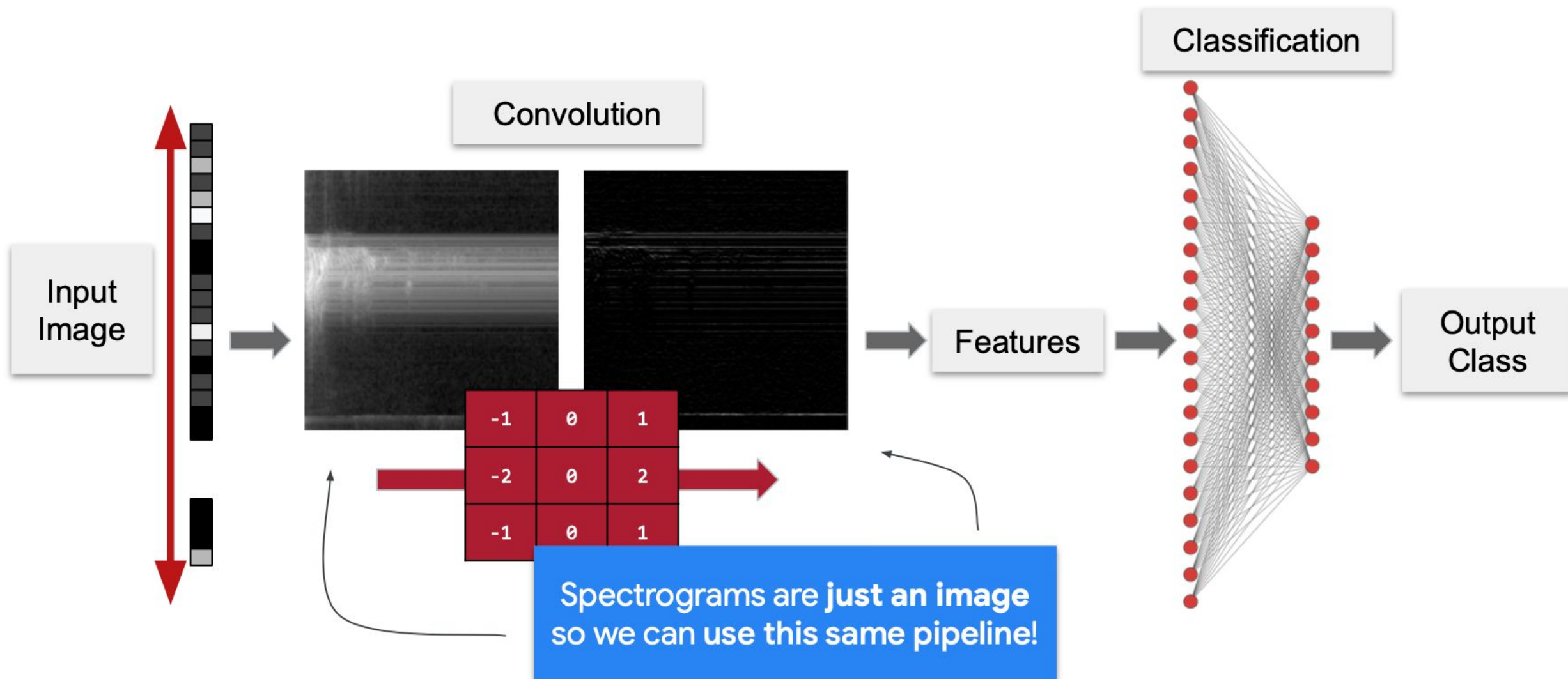


# A model for **Keyword Spotting**





# A model for **Keyword Spotting**

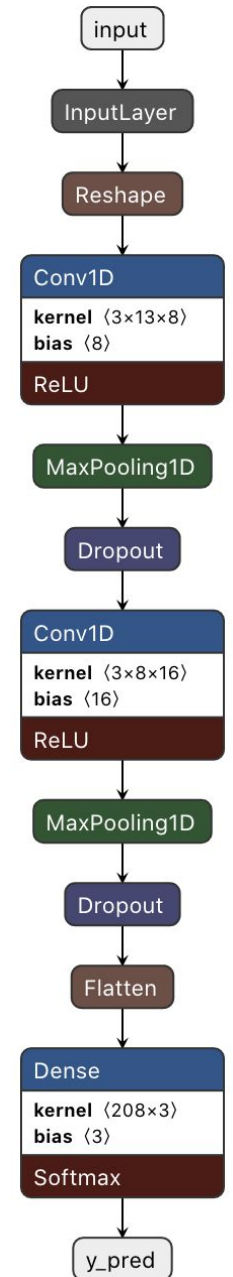


Model: "sequential"

Layer (type)	Output Shape	Param #
reshape (Reshape)	(None, 50, 13)	0
conv1d (Conv1D)	(None, 50, 8)	320
max_pooling1d (MaxPooling1D)	(None, 25, 8)	0
dropout (Dropout)	(None, 25, 8)	0
conv1d_1 (Conv1D)	(None, 25, 16)	400
max_pooling1d_1 (MaxPooling1D)	(None, 13, 16)	0
dropout_1 (Dropout)	(None, 13, 16)	0
flatten (Flatten)	(None, 208)	0
y_pred (Dense)	(None, 3)	627

=====  
Total params: 1,347  
Trainable params: 1,347  
Non-trainable params: 0  
=====

Model size: 200KB



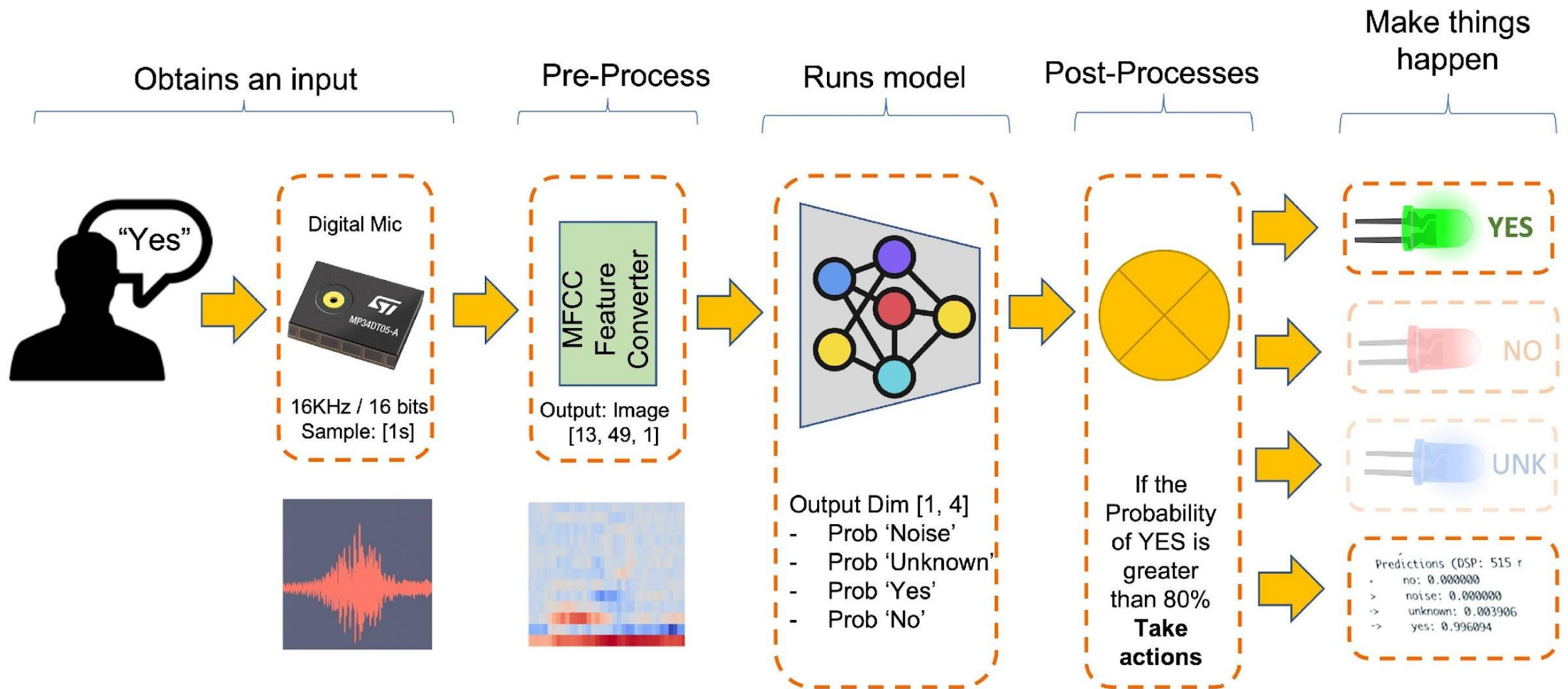
# KWS

## Keyword Spotting Project

- **First keyword:** Yes
- **Second keyword:** No
- **Noise** Background Noise
- **Unknown:** (a mix of different words than Yes and No)

<https://studio.edgeimpulse.com/public/292418/latest>





# Questions?



27°

Taller sobre  
Tecnologías de Redes Internet  
para América Latina y el Caribe

Organización General