UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

Reconocimiento de palabras con TinyML

Juan Pablo Gómez
Danilo Tovar

Una Facultad abierta y transformadora

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

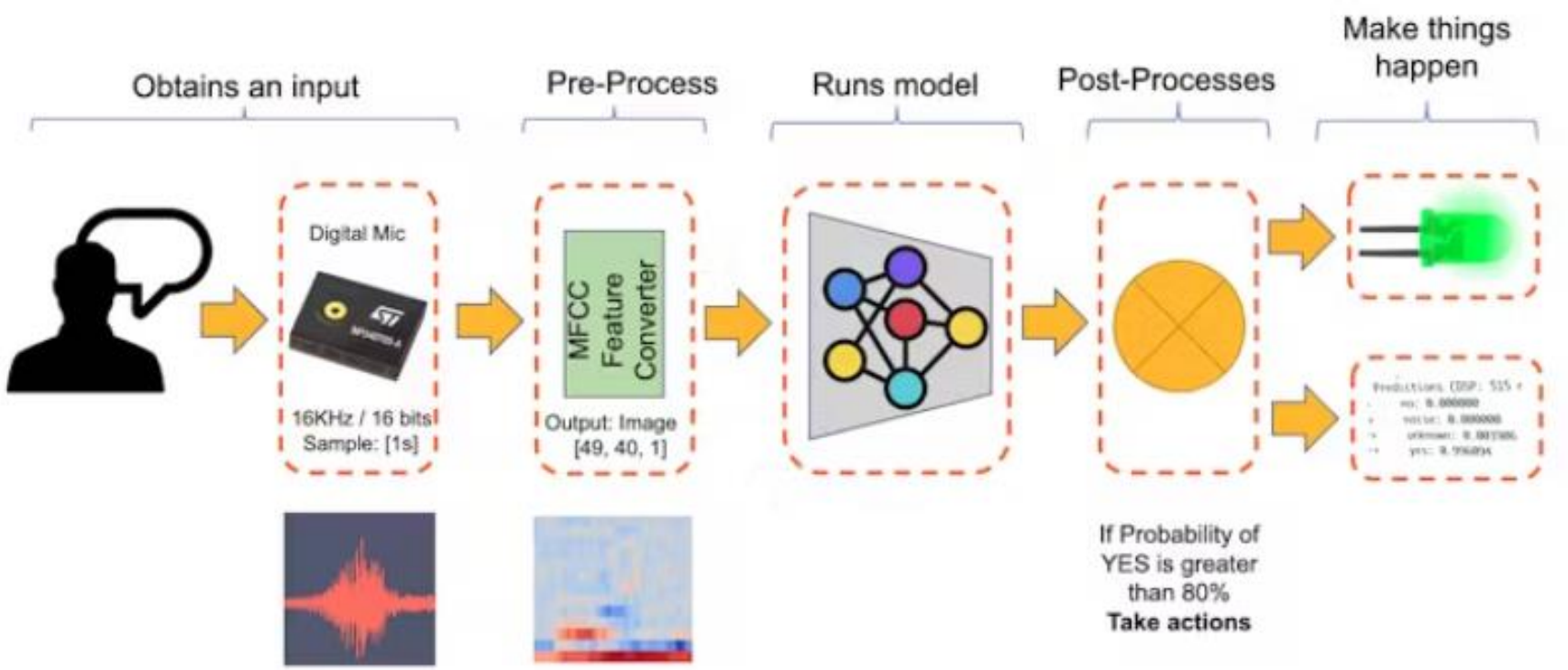
Objetivo

Implementar un modelo de inteligencia artificial (IA) para identificación de palabras clave (keyword spotting) y desplegarlo en un Raspberry Pi 4

Objetivos específicos

- Reconocimiento de dos comandos específicos: "encender" y "apagar"
- Captura de audio en tiempo real desde micrófono USB
- Inferencia local del modelo de ML sin conectividad externa
- Control de LED mediante GPIO de Raspberry Pi

Esquema teórico

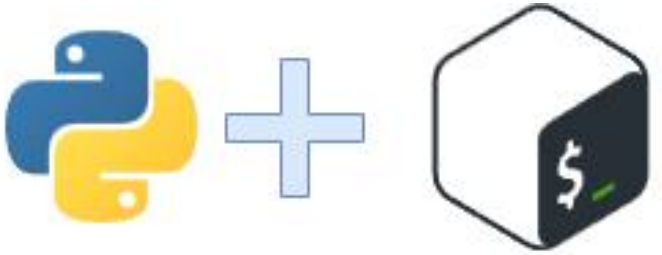


Fuente: https://www.hackster.io/mjrobot/tinyml-made-easy-keyword-spotting-kws-5fa6e7

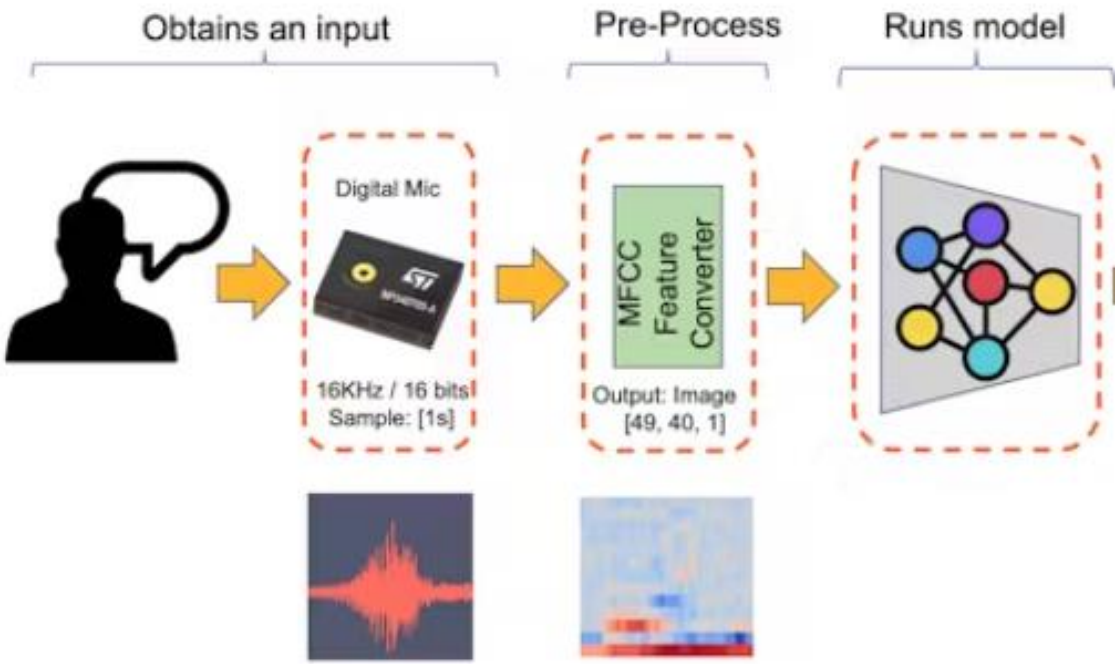
Nuestra implementación

Python
SDK

model.eim



¿Por qué Edge Impulse?



Para ahorrar tiempo

EDGE IMPULSE

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

Etapas del proyecto

1. Conectar Raspberry al micrófono, y conectar a Edge Impluse. (Configuraciones iniciales)
2. Crear modelo en Edge Impulse
    1. Toma de datos
    2. Ajuste de parámetros
    3. ¡Entrenar!
3. Descargar modelo: artefacto model.eim
4. Con el SDK de Edge Impulse de Python, implementar lógica según la etiqueta detectada para encender LED, y posteriormente ejecutar.

Primeros pasos (1)

Se debe disponer de una Micro SD con algún sistema operativo. Nosotros escogimos Rasberry Pi Os (basado en Debian GNU/Linux). En nuestro caso, tuvimos que conectarnos via SSH porque no disponíamos de monitor con HDMI ☹.

```
(base) PS C:\Users\JuanPabloGomez> ssh juanpablogomez@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:rCsY/noXejOr9i/VPpz2Uu9bYI+D5Re61oyrY1d1lDc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.12' (ED25519) to the list of known hosts.
juanpablogomez@192.168.1.12's password:
Linux raspberrypi 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 12 19:41:59 2025
juanpablogomez@raspberrypi:~ $ xrandr
Can't open display
juanpablogomez@raspberrypi:~ $ sudo cat /boot/config.txt
DO NOT EDIT THIS FILE

The file you are looking for has moved to /boot/firmware/config.txt
juanpablogomez@raspberrypi:~ $ kmsprint | grep Connector
Connector 0 (33) HDMI-A-1 (disconnected)
Connector 1 (42) HDMI-A-2 (connected)
juanpablogomez@raspberrypi:~ $ raindrop
Can't open display
Can't open display
cp: cannot stat '/var/tmp/dispsetup.sh': No such file or directory
^C
juanpablogomez@raspberrypi:~ $ man raindrop
```

```
juanpablogomez@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Primeros pasos (1)

Se debe instalar software de Edge Impulse. Se sigue la guía que ellos mismo proveen:

https://docs.edgeimpulse.com/docs/edge-ai-hardware/cpu/raspberry-pi-4

Luego se ejecuta

$ edge-impulse-linux

Para configurar la Raspberry y comunicarla con la plataforma

Plataforma Edge Impulse

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

¡Mostrar plataforma!

Despliegue (1)

- Una vez se tiene el modelo en la plataforma, se debe descargar este.
- Pero primero, se instala el Python SDK de Edge Implse:

https://docs.edgeimpulse.com/docs/tools/edge-impulse-for-linux/linux-python-sdk

- Luego, descargar el modelo:

```
$ edge-impulse-linux-runner --download model.eim
```

Despliegue (2)

- Se debe crear el script para ejecutar el modelo descargado. Edge impulse provee una plantilla básica para esta labor.
- La ejecución del script se hace mediante este comando:

```
$ python ./model_script.py model.eim <ID_INTERFAZ_MICROFONO>
```

- La tarea ahora consiste en adaptar este código para el **control** del LED.

Configuración del LED

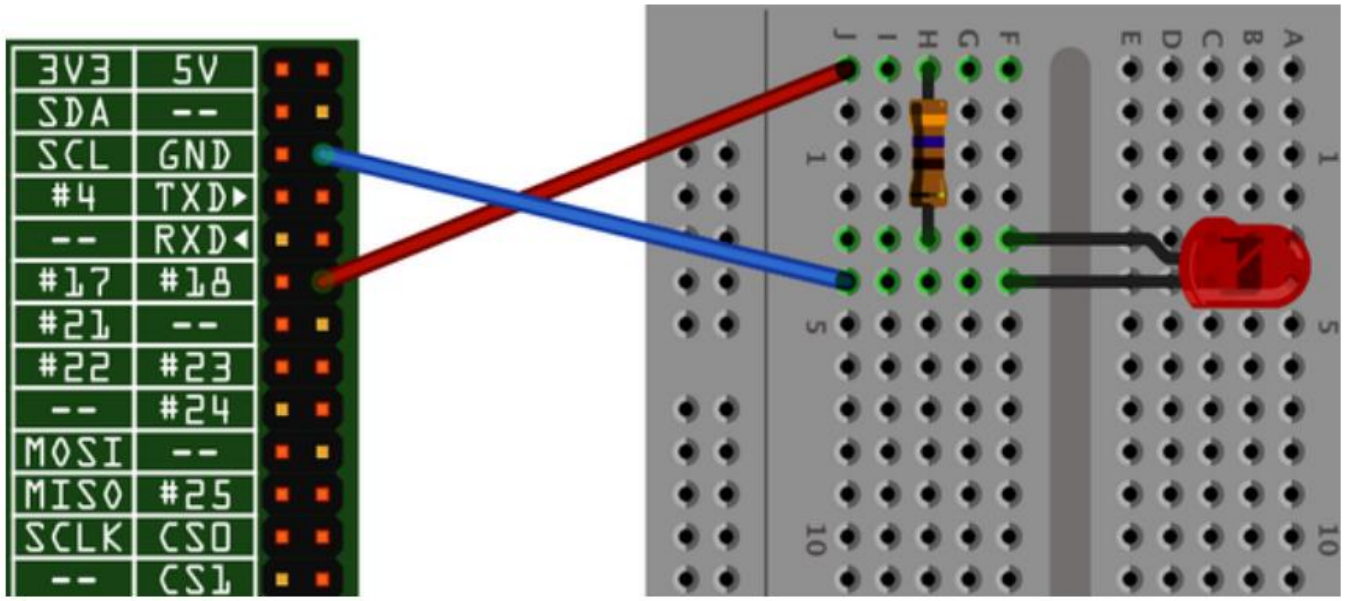


Figure 11-1. Connecting an LED to a Raspberry Pi

(Esquema obtenido de Raspberry Pi Cookbook, Simon Monk, O'Reilly 4ed)

Tabla de estados para el LED

Estado del LED	Comando reconocido	Acción
encendido	'encender'	no hacer nada
apagado	'encender'	encender el LED
apagado	'apagar'	no hacer nada
encendido	'apagar'	apagar el LED
encendido	'desconocido' o 'ruido'	no hacer nada
apagado	'desconocido' o 'ruido'	no hacer nada

Modificación script

El control del LED se puede resumir en este simple condicional

```
if scores.get('encender', 0) > 0.8:
    if not led_status:
        led.on()
        led_status = True
        print("LED turned ON")
elif scores.get('Apagar', 0) > 0.8:
    if led_status:
        led.off()
        led_status = False
        print("LED turned OFF")
```

Threshold

Variable auxiliar de control

Creación de servicio del SO

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

- Recordar el comando para ejecutar el script

```
$ python ./model_script.py model.eim <ID_INTERFAZ_MICROFONO>
```

- <ID_INTERFAZ_MICROFONO> puede cambiar al conectar el micrófono en otro Puerto o simplemente tener otro valor.
- El Sistema solo funciona si se ejecuta manualmente el commando de arriba. ¿qué hacer?

Se crea un script de bash <3

```bash
#!/bin/bash

# Activate the virtual environment
source /home/juanpablogomezlopez/proyecto_so/venv/bin/activate

# Auto-detect microphone using PyAudio in Python
DEVICE_ID=$(python3 - <<EOF
import pyaudio
p = pyaudio.PyAudio()
for i in range(p.get_device_count()):
    info = p.get_device_info_by_index(i)
    if info.get("maxInputChannels", 0) > 0:
        name = info.get("name", "").lower()
        if any(k in name for k in ["usb", "webcam", "mic"]):
            print(i)
            break
else:
    print(0)
p.terminate()
EOF
)

echo "🎤 Using device ID: $DEVICE_ID"

# Run the model script with the detected device ID
python /home/juanpablogomezlopez/proyecto_so/model_script.py model.eim "$DEVICE_ID
```

Se busca dos cosas, automatizar el proceso de selección de la interfaz del micrófono. Se crea el archivo **start_model.sh**

$ sudo nano /etc/systemd/system/keywordspotting.service

```
[Unit]
Description=Keyword Spotting on Boot
After=network.target

[Service]
Type=simple
ExecStart=/home/juanpablogomezlopez/proyecto_so/start_model.sh
WorkingDirectory=/home/juanpablogomezlopez/proyecto_so
User=juanpablogomezlopez
Restart=always
StandardOutput=inherit
StandardError=inherit

[Install]
WantedBy=multi-user.target
```

Se crea el servicio

Ejecutar y disfrutar

```
1 sudo systemctl daemon-reload
2 sudo systemctl enable keywordspotting
3 sudo systemctl start keywordspotting
```