Automatic Speech Recognition (ASR) with vosk

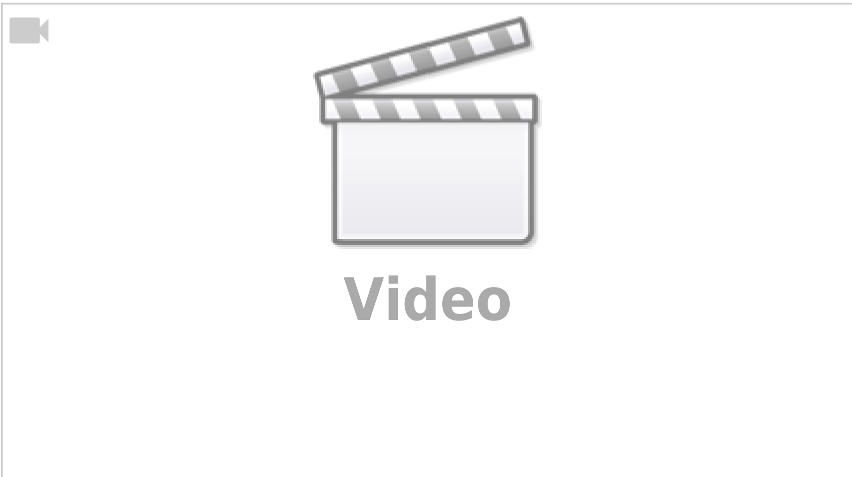
Sources

vosk

- https://alphacephei.com/vosk/
- https://github.com/alphacep/vosk-api

Dataquest

- https://github.com/dataquestio/project-walkthroughs/blob/master/microphone/microphone.ipynb



Installation

```
conda create -n vosk python=3.9
conda activate vosk
conda install -c conda-forge jupyterlab numpy matplotlib pandas
#conda install -c conda-forge ipywidgets
#conda install -c conda-forge scipy scikit-learn
```

```
pip install vosk
pip install pyaudio
```

On Windows the vosk models are **cached here**: C:\Users\<username>\.cache\vosk

Missing:

ffmpeg …

```
import pyaudio
import wave

# Constants for audio recording
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 1024
RECORD_SECONDS = 5  # Adjust this to change the duration of the recording
OUTPUT_FILENAME = "output.wav"

def list_audio_devices():
    audio = pyaudio.PyAudio()
    devices = []

    for i in range(audio.get_device_count()):
        device_info = audio.get_device_info_by_index(i)
        devices.append(f"{i}: {device_info['name']}")

    audio.terminate()
    return devices

def get_input_device_index():
    devices = list_audio_devices()

    print("Available audio input devices:")
    for device in devices:
        print(device)

    while True:
        try:
            print("")
            print("On Becker's Dell Lat. 7330 the following works:")
            print("1: Microphone Array (Realtek(R) Au")
            print("")
            device_index = int(input("Enter the index of the desired input
device: "))
            if 0 <= device_index < len(devices):
                return device_index
            else:
                print("Invalid input. Please enter a valid device index.")
        except ValueError:
            print("Invalid input. Please enter a valid device index.")

def record_audio(device_index):
    audio = pyaudio.PyAudio()

    # Open a microphone stream with the selected input device
    stream = audio.open(format=FORMAT, channels=CHANNELS,
                        rate=RATE, input=True,
input_device_index=device_index,
```

```
                                frames_per_buffer=CHUNK)

    print(f"Recording from: {audio.get_device_info_by_index(device_index)['name']}")

    frames = []

    # Record audio in chunks and store it in frames
    for _ in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
        data = stream.read(CHUNK)
        frames.append(data)

    print("Finished recording.")

    # Stop and close the microphone stream
    stream.stop_stream()
    stream.close()
    audio.terminate()

    # Save the recorded audio to a WAV file
    with wave.open(OUTPUT_FILENAME, 'wb') as wf:
        wf.setnchannels(CHANNELS)
        wf.setsampwidth(audio.get_sample_size(FORMAT))
        wf.setframerate(RATE)
        wf.writeframes(b''.join(frames))

if __name__ == "__main__":
    device_index = get_input_device_index()
    record_audio(device_index)
```

From:
https://wiki.eolab.de/ - **HSRW EOLab Wiki**

Permanent link:
https://wiki.eolab.de/doku.php?id=user:rolf001:vosk:start&rev=1694354788

Last update: **2023/09/10 16:06**