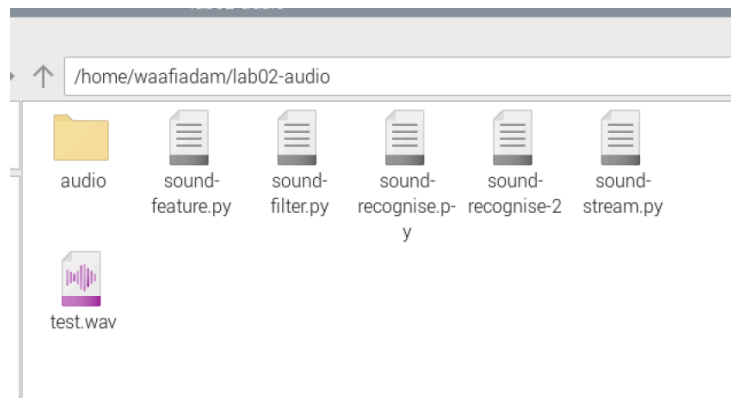
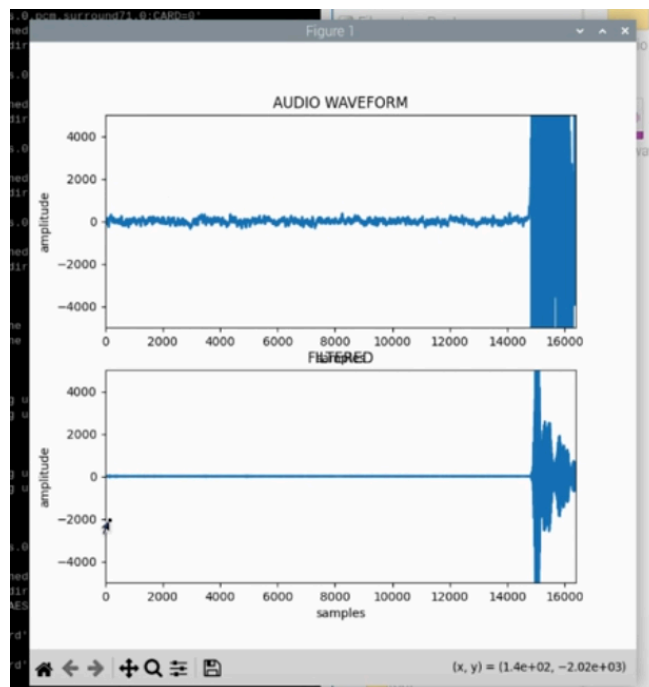


All files created here



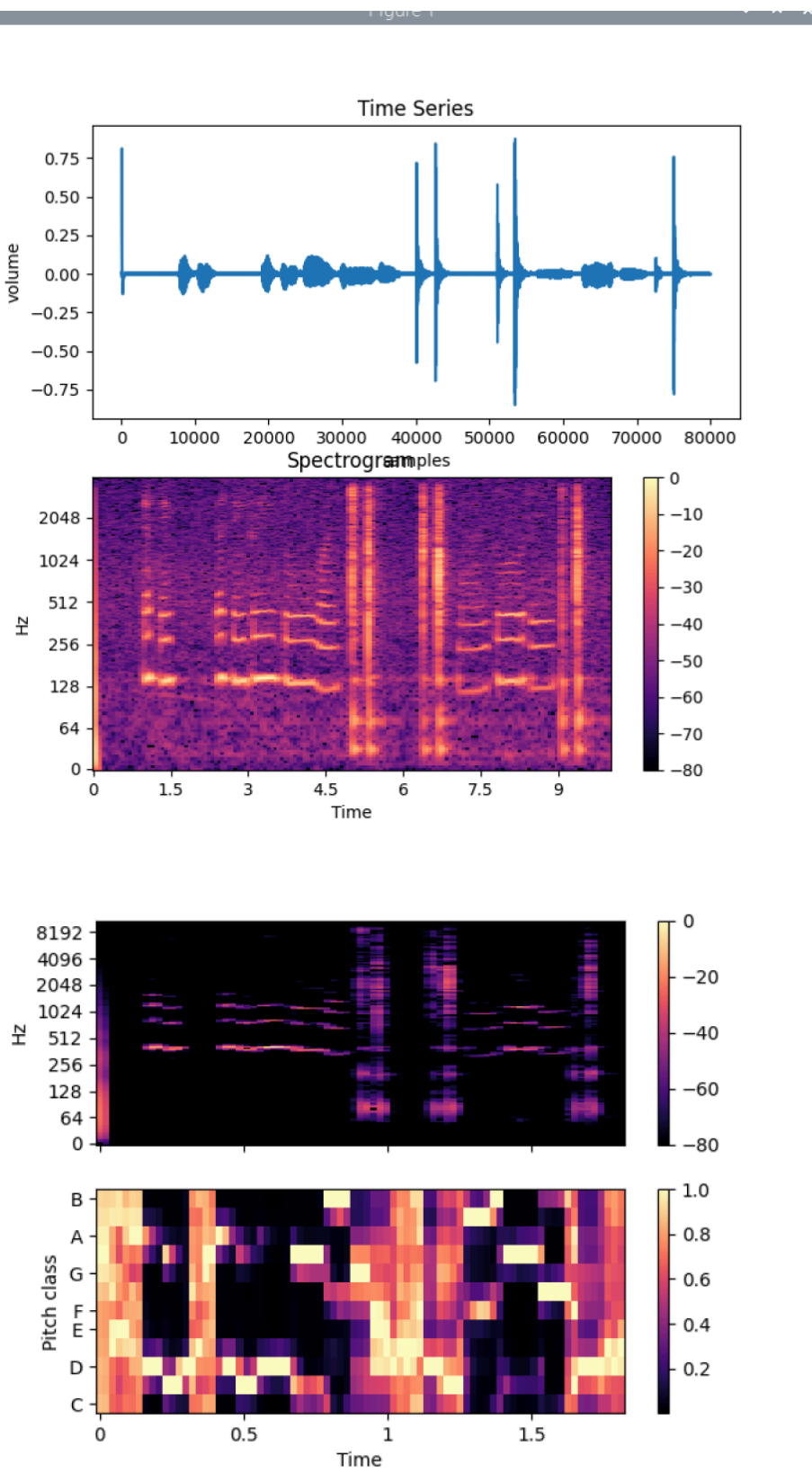
Adjusted the low and high frequency to see filter only claps which has a higher frequency range

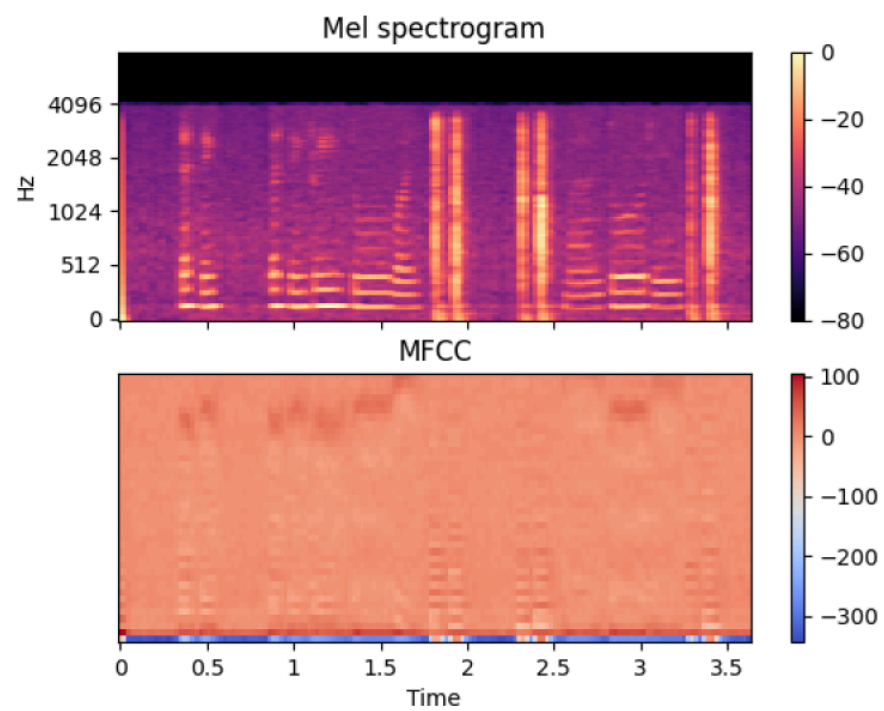
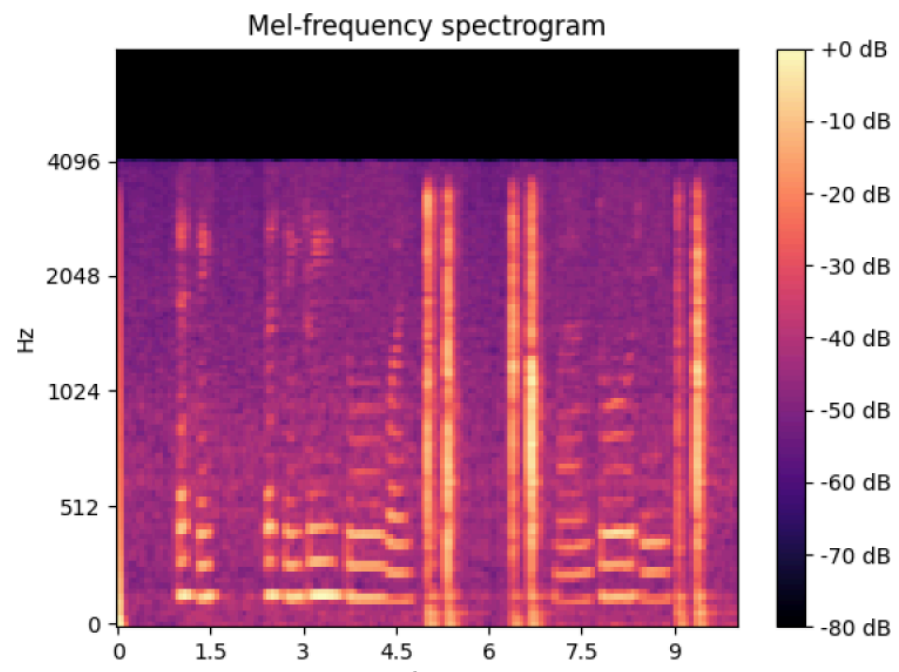
```
# design the filter
LOW_FREQ = 7000
HIGH_FREQ = 8000
sos = design_filter(LOW_FREQ, HIGH_FREQ, 48000, 3) #
```



Feature extraction  
with test.wav

Spectrogram





## Speech recognition

Modified with one extra speech recognition API, vosk, and with wake word identification.

I tested all three with the phrase, "Hello there, ok google, what is up"

```
===== SUMMARY =====  
  
Google Recognition:  
- Recognized Text: hello there OK Google what is up  
- Processing Time: 0.64 seconds  
- Wake Word Detected: Yes  
  
Sphinx Recognition:  
- Recognized Text: lou the team will lead us  
- Processing Time: 9.62 seconds  
- Wake Word Detected: No  
  
Vosk Recognition:  
- Recognized Text:  
- Processing Time: 1.52 seconds  
- Wake Word Detected: No  
  
=====
```

(audio) waafiadam@waafiadam-pi:~/lab02-audio \$

```
#!/usr/bin/env python3  
import speech_recognition as sr  
import os  
import time  
import json  
from vosk import Model, KaldiRecognizer  
  
# Define wake words  
WAKE_WORDS = ["ok google", "alexa", "siri"]  
  
# Dictionary to store results  
results_summary = {}  
  
def check_wake_word(text, model_name):  
    """Check if any wake word is in the recognized text."""  
    for word in WAKE_WORDS:  
        if word in text.lower():  
            print(f"\nWake word detected in {model_name}: {word}")  
            print(f"Wake word activated ({model_name})! Executing  
command...\n")  
            os.system("echo 'Command executed'") # Replace with  
actual command  
            return True  
        return False  
  
def record_audio(model_name):  
    """Records audio separately for each model."""
```

```

with sr.Microphone() as source:
    r.adjust_for_ambient_noise(source)
    print(f"\n[INFO] Recording audio for {model_name}, please
speak now...")
    audio = r.listen(source)
    print(f"[INFO] Finished recording for {model_name}.
Processing...\n")
    return audio

def recognize_with_google(audio):
    """Recognize speech using Google Speech Recognition (Online)"""
    try:
        start_time = time.time()
        text = r.recognize_google(audio)
        duration = time.time() - start_time
        print(f"Google recognized: {text}")
        print(f"Time taken: {duration:.2f} seconds")
        wake_word_detected = check_wake_word(text, "Google")
        results_summary["Google"] = {"text": text, "time": duration,
"wake_word": wake_word_detected}
        return text
    except sr.UnknownValueError:
        print("Google Speech Recognition could not understand the
audio")
    except sr.RequestError as e:
        print(f"Google Speech Recognition request error: {e}")
        results_summary["Google"] = {"text": "Not recognized", "time":
None, "wake_word": False}
        return None

def recognize_with_sphinx(audio):
    """Recognize speech using CMU Sphinx (Offline)"""
    try:
        start_time = time.time()
        text = r.recognize_sphinx(audio)
        duration = time.time() - start_time
        print(f"Sphinx (Offline) recognized: {text}")
        print(f"Time taken: {duration:.2f} seconds")
        wake_word_detected = check_wake_word(text, "Sphinx")
        results_summary["Sphinx"] = {"text": text, "time": duration,
"wake_word": wake_word_detected}
        return text
    except sr.UnknownValueError:
        print("Sphinx could not understand the audio")
    except sr.RequestError as e:
        print(f"Sphinx error: {e}")
        results_summary["Sphinx"] = {"text": "Not recognized", "time":
None, "wake_word": False}
        return None

```

```

def recognize_with_vosk(audio):
    """Recognize speech using Vosk (Offline)"""
    try:
        model = Model(os.path.expanduser("~/vosk/model"))
        recognizer = KaldiRecognizer(model, 16000)

        start_time = time.time()
        recognizer.AcceptWaveform(audio.get_wav_data())
        result = json.loads(recognizer.Result())
        duration = time.time() - start_time

        text = result.get("text", "")
        print(f"Vosk (Offline) recognized: {text}")
        print(f"Time taken: {duration:.2f} seconds")
        wake_word_detected = check_wake_word(text, "Vosk")
        results_summary["Vosk"] = {"text": text, "time": duration,
        "wake_word": wake_word_detected}
        return text
    except Exception as e:
        print(f"Vosk error: {e}")
        results_summary["Vosk"] = {"text": "Not recognized", "time": None,
        "wake_word": False}
        return None

# Initialize recognizer
r = sr.Recognizer()

# Record and recognize with Google
audio_google = record_audio("Google Speech Recognition")
recognized_text_google = recognize_with_google(audio_google)

# Record and recognize with Sphinx
audio_sphinx = record_audio("CMU Sphinx (Offline)")
recognized_text_sphinx = recognize_with_sphinx(audio_sphinx)

# Record and recognize with Vosk
audio_vosk = record_audio("Vosk (Offline)")
recognized_text_vosk = recognize_with_vosk(audio_vosk)

# Print summary at the end
print("\n===== SUMMARY =====")
for model, data in results_summary.items():
    print(f"\n{model} Recognition:")
    print(f" - Recognized Text: {data['text']}")
    print(f" - Processing Time: {data['time']:.2f} seconds" if data["time"]
    else " - Processing Time: N/A")
    print(f" - Wake Word Detected: {'Yes' if data['wake_word'] else
    'No'}")
print("\n=====\\n")

```