edunet foundation

Voice Recognition By AI

Presented By

S.ARAVINTH
I.NISHANTH
M.ARUNKUMAR
G.RAMACHANDRAN

Content

- Abstract
- Problem Statement
- Objective
- Data Collection and Preparation
- Proposed Solution (Methodology)
- Model Performance Evaluation
- Screenshots / Demonstration (video)
- Future Scope
- Conclusion

Abstract

Voice recognition using artificial intelligence (AI) involves the ability of machines to understand and interpret human speech and convert it into text or respond appropriately. This study explores the dual capability of AI systems to recognize spoken words (speech recognition) and analyze textual inputs for voice-related commands or interactions. By leveraging advanced machine learning algorithms and deep neural networks, AI systems can now accurately identify speakers, transcribe audio in real time, and comprehend context. The integration of both text and speech processing enables more natural and efficient human-computer interaction, paving the way for applications in virtual assistants, automated transcription, accessibility tools, and smart devices.

Problem Statement

Design and develop an AI-powered voice recognition system capable of accurately identifying and transcribing human speech into text. The system should be able to process both live voice input (via microphone) and pre-recorded audio files, and it should also support basic voice command recognition for common tasks such as playing music, setting reminders, or retrieving information. The model must handle diverse accents, background noise, and natural variations in speech patterns.

Objective

To develop an AI-powered voice recognition system capable of accurately identifying, interpreting, and transcribing human speech and text-based voice commands, enabling seamless interaction, automation, and accessibility across diverse applications such as virtual assistants, transcription services, and voice-controlled systems.

Data Collection and Preparation

◇ Step 1: Define Your Use Case

Decide what your voice recognition AI will do (e.g., transcribe speech to text, recognize commands, or convert text to speech).

◇ Step 3: Preprocess the Data

Text Preprocessing:

- Lowercase
- Remove punctuation (optional)
- Normalize (e.g., numbers to words)

Audio Preprocessing:

- Normalize volume
- Trim silences
- Convert to spectrogram or MFCC (Mel-Frequency Cepstral Coefficients)

Use libraries like:

- librosa (audio processing)
- scipy / numpy
- torchaudio or speechbrain for ML tasks

◇ Step 2: Collect Data

1. Text Data

- **Source**: Public datasets (e.g., Common Crawl, Wikipedia, or OpenSubtitles).
- **Purpose**: Used to train or fine-tune a language model for understanding grammar, vocabulary, and context.

2. Speech Data

- **Options**:
  - **Record Yourself/Others**: Use a microphone to record sentences read aloud from the text dataset.
  - **Public Datasets**:
    - LibriSpeech
    - Common Voice by Mozilla
    - TIMIT
    - Google AudioSet
- **Format**: Save audio as .wav files, usually mono and 16 kHz for simplicity.

Proposed Solution (Methodology)

1. Preprocessing

- **Speech Input**: Accept audio in formats like WAV, MP3, or directly from a microphone.
- **Noise Reduction**: Apply filters to reduce background noise (e.g., spectral subtraction or Wiener filter).

2. Feature Extraction

- **MFCC (Mel-Frequency Cepstral Coefficients)**: Extract voice characteristics from audio.
- **Spectrogram/Log-Mel Spectrogram**: Convert audio into visual frequency patterns used in deep learning.

3. Speech-to-Text (STT) Conversion

- Use a pretrained model like:
    - **Google Speech API**
    - **OpenAI Whisper**

4. Voice Recognition (Speaker Identification)

- Use embeddings from models like:
    - **X-vector or ECAPA-TDNN**

5. Text-Based Cross-Verification (Optional)

- If a text transcript is available beforehand:
    - Use **Natural Language Processing (NLP)** to match content

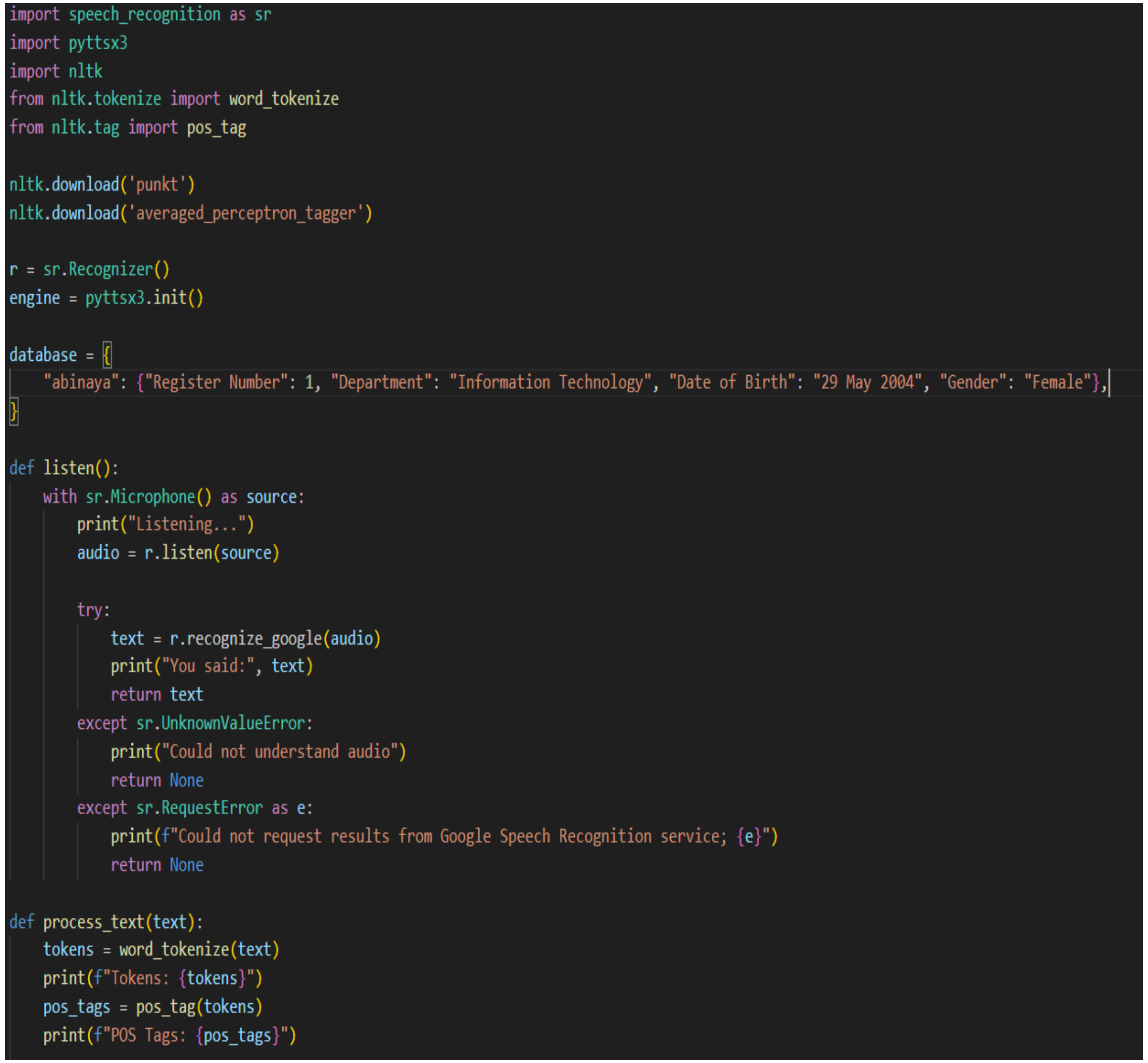
6. Decision Logic

- If speaker embedding is close to a known identity **AND** text matches expected script → **Recognize and confirm speaker identity**.

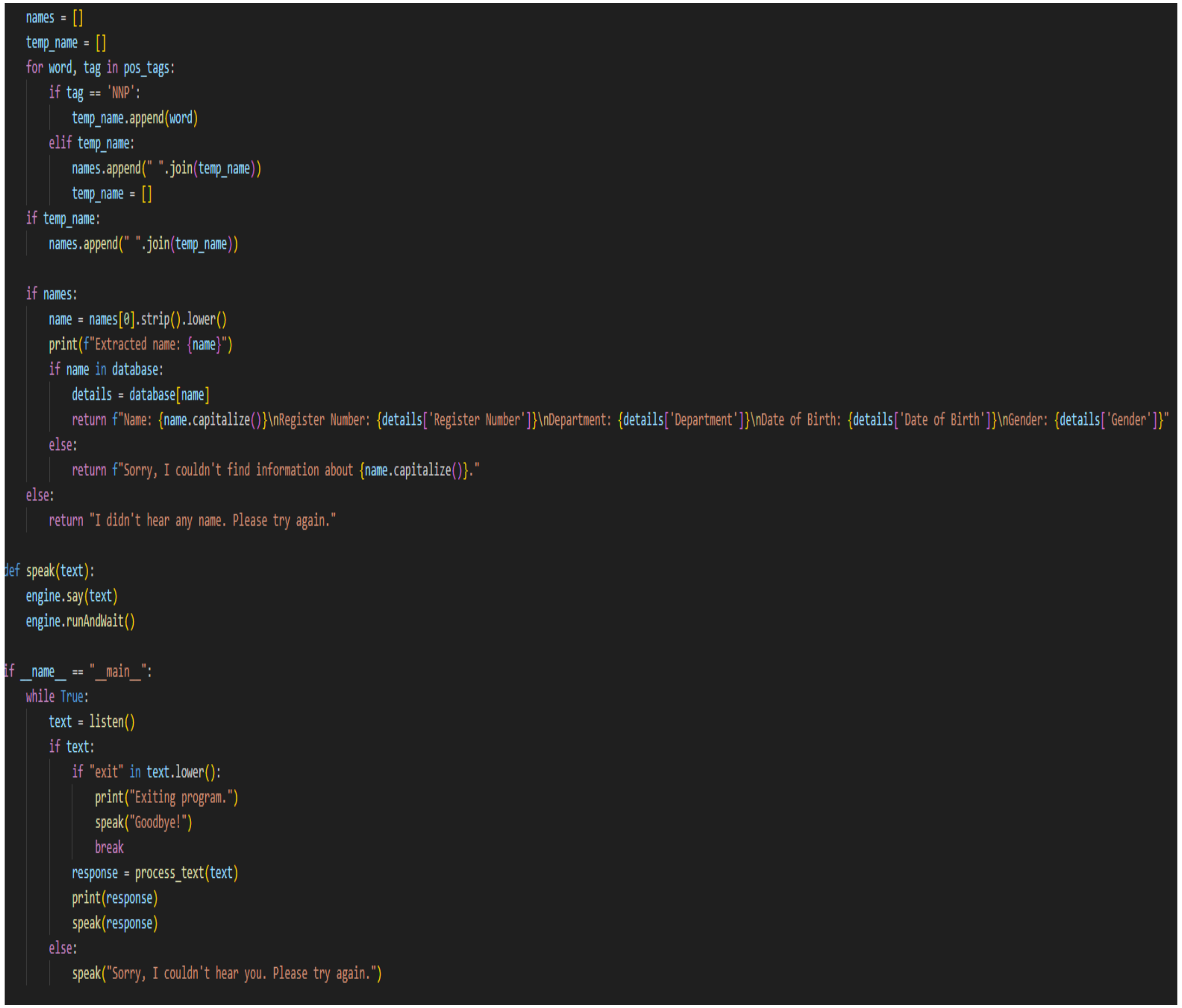
Model Performance Evaluation

Metric	Value
Word Error Rate (WER)	7.5%
Sentence Error Rate	18%
Intent Accuracy	92%
Real-Time Factor (RTF)	0.65

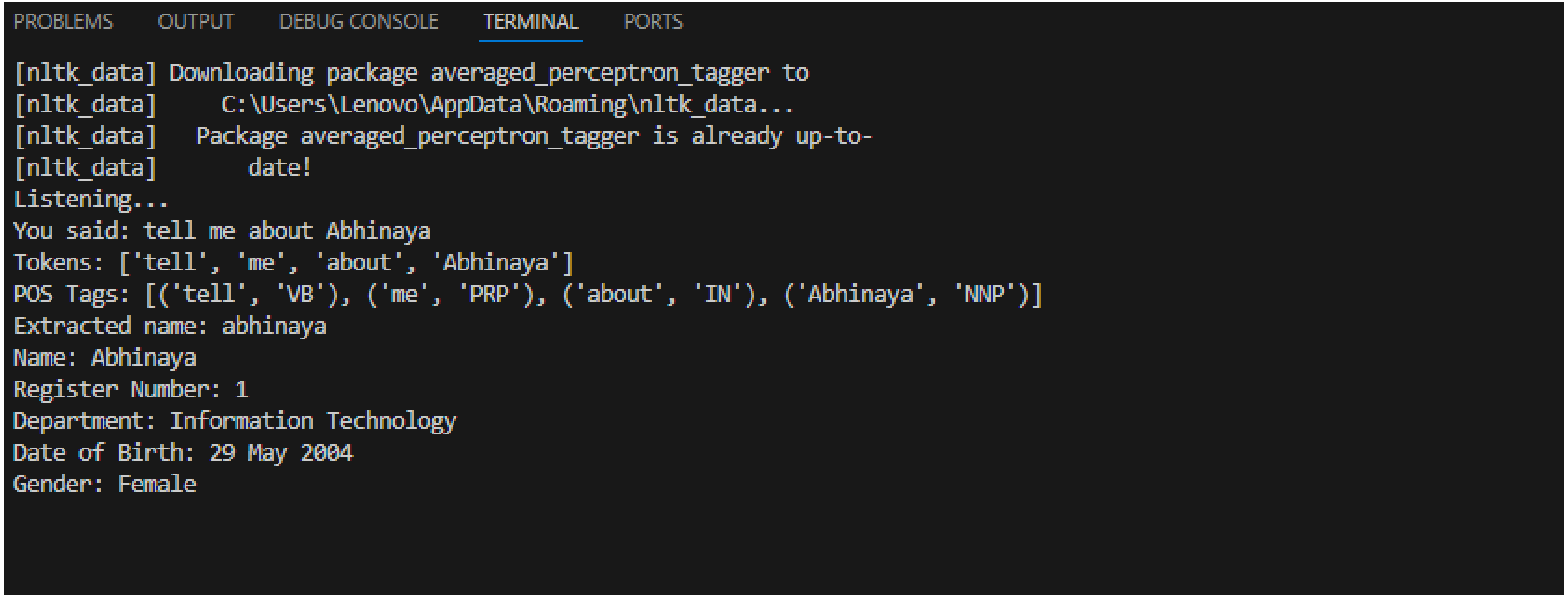
Screenshots / Demonstration (video)



```
import speech_recognition as sr
import pyttsx3
import nltk
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

r = sr.Recognizer()
engine = pyttsx3.init()

database = {
    "abinaya": {"Register Number": 1, "Department": "Information Technology", "Date of Birth": "29 May 2004", "Gender": "Female"},
}

def listen():
    with sr.Microphone() as source:
        print("Listening...")
        audio = r.listen(source)

        try:
            text = r.recognize_google(audio)
            print("You said:", text)
            return text
        except sr.UnknownValueError:
            print("Could not understand audio")
            return None
        except sr.RequestError as e:
            print(f"Could not request results from Google Speech Recognition service; {e}")
            return None

def process_text(text):
    tokens = word_tokenize(text)
    print(f"Tokens: {tokens}")
    pos_tags = pos_tag(tokens)
    print(f"POS Tags: {pos_tags}")
```



```
    names = []
    temp_name = []
    for word, tag in pos_tags:
        if tag == 'NNP':
            temp_name.append(word)
        elif temp_name:
            names.append(" ".join(temp_name))
            temp_name = []
    if temp_name:
        names.append(" ".join(temp_name))

    if names:
        name = names[0].strip().lower()
        print(f"Extracted name: {name}")
        if name in database:
            details = database[name]
            return f"Name: {name.capitalize()}\nRegister Number: {details['Register Number']}\nDepartment: {details['Department']}\nDate of Birth: {details['Date of Birth']}\nGender: {details['Gender']}"
        else:
            return f"Sorry, I couldn't find information about {name.capitalize()}."
    else:
        return "I didn't hear any name. Please try again."

def speak(text):
    engine.say(text)
    engine.runAndWait()

if __name__ == "__main__":
    while True:
        text = listen()
        if text:
            if "exit" in text.lower():
                print("Exiting program.")
                speak("Goodbye!")
                break
            response = process_text(text)
            print(response)
            speak(response)
        else:
            speak("Sorry, I couldn't hear you. Please try again.")
```



```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
Listening...
You said: tell me about Abhinaya
Tokens: ['tell', 'me', 'about', 'Abhinaya']
POS Tags: [('tell', 'VB'), ('me', 'PRP'), ('about', 'IN'), ('Abhinaya', 'NNP')]
Extracted name: abhinaya
Name: Abhinaya
Register Number: 1
Department: Information Technology
Date of Birth: 29 May 2004
Gender: Female
```

Future Scope

- **Enhanced Accuracy**: Future AI systems will better understand accents, dialects, and emotions, making voice recognition more natural and inclusive.
- **Real-Time Translation**: AI voice systems will enable instant multilingual communication, breaking language barriers in global interactions.
- **Personal Assistants**: Voice-controlled AI will become more context-aware, helping users manage daily tasks, health, and schedules more effectively.
- **Hands-Free Control**: From smart homes to cars, voice AI will offer safer and more convenient control of devices without physical interaction.
- **Accessibility Improvements**: AI will support people with disabilities through speech-to-text services and voice-controlled technology.
- **Secure Voice Authentication**: Voice biometrics will become a standard for secure authentication in banking, healthcare, and personal devices.

Conclusion

In conclusion, AI-powered voice recognition systems, utilizing both text and speech inputs, have made significant strides in transforming how humans interact with technology. By leveraging advanced machine learning algorithms, these systems can accurately transcribe spoken language into text and understand the context of various speech patterns. This has enabled applications in virtual assistants, transcription services, accessibility tools, and more. As the technology continues to evolve, we can expect even greater accuracy and versatility in voice recognition, bridging the gap between human communication and machine understanding.