

Pico Weather Assistant - ECEN 5003 Final Project

Team members

Sriram Yadavalli

Goals

- To build a weather assistant device that runs completely locally.
- Train and use a weather model using self collected data.
- Train and use a voice model for text-to-speech functionality.
- Experiment with more complex voice model creation and training.
- Using pre-trained models to improve text-to-speech functionality

What was produced

- Two executables are produced as part of this application
- VoiceApp and WeatherApp
- User can talk to the application with a weather related prompt and get the desired output based on what they asked

VoiceApp

- The VoiceApp application is responsible for recording audio, handling reading keyboard inputs, and text-to-speech through the DeepSpeech tflite model

WeatherApp

- The WeatherApp application is responsible for parsing the text from VoiceApp, reading sensor data, and inference through the polynomial regression tflite model

How to build and run?

Hardware Requirements

- Host PC/Mac to clone Git repo
- Target Hardware, Raspberry Pi 3B+
- USB Microphone connected to Raspberry Pi 3B+
- DHT22 Temperature/Humidity Sensor connected to GPIO4 on Raspberry Pi 3B+

Prepping hardware

- To update and install required packages
 - `sudo apt update`
 - `sudo apt upgrade`
 - `sudo apt install -y build-essential libpigpio-dev libasound-dev libsndfile1-dev libdeepspeech-dev`

Transferring, building, and running

- Clone the repo
 - `git clone https://github.com/Sriram-Y/Pico-Assistant`
- Go into repository directory
 - `cd Pico-Assistant`
- Modify the Makefile variable `REMOTE_HOST` to point to `username@ip_address`
- Modify the Makefile variable `REMOTE_PATH` to point the desired destination directory (make sure this directory exists on the target)
- Run `make transfer` to transfer all required files to build on target
- Go into `REMOTE_PATH`
- Run `make` on target hardware
- Run `./VoiceApp`
- Press `[SPACE]` when prompted
- Try one of the prompts
 - "What is the temperature?"
 - "What is the humidity?"
 - "What will the temperature be tomorrow?"
 - "What will the humidity be tomorrow?"
- Press `[SPACE]` to stop recording
- Wait for response

Takeaways and Learnings

- Understood how to use, build, and train a multilayer perceptron for polynomial regression
- Understood how to use a Recurrent Neural Network for speech recognition
- Experimenting with implementing my own speech to text model for keyword recognition

- Embedded device deployment (keeping the model small, responsiveness, etc.)
- Audio capture, export, and passing data to model in main application code

GitHub Repo Link

- [Pico Weather Assistant GitHub](#)