Module 1

Speech Recognition Calculator

Yvette Gerber, Hayden Somach

I. INTRODUCTION

The objectives of the speech recognition extension were to improve accuracy and to expand capability of the speech recognition algorithm. Accuracy was improved by comparing frequency-domain correlation over entire spectrograms, optimizing spectrogram parameters, increasing the number and variability of templates, applying frequency filtering, and leveraging the inherent "mathematical grammar" of the input. Capability was expanded through real-time word detection and the development of a speech-driven calculator, initially limited to adding numbers one through six and incrementally extended to support additional mathematical operations.

II. METHODS

A. Hardware Setup

Audio was obtained through a microphone and sent to a desktop computer via a sound card. All digital processing was competed using MATLAB. Hardware specifics are as follows:

- USB duplex sound card with 3.5-mm input/output
- 3.5-mm and USB microphone
- 3.5-mm aux cable
- Computer with MATLAB and MATLAB audio toolbox installed

B. Algorithm

Initialization & Saving Audio: The algorithm begins by loading in all example templates for each word, along with storing basic parameters. The set of templates associated with a specific word are stored in their own respective matrices. Spectrograms of each template are created using MATLAB's spectrogram(). A mask is applied to each spectrogram to only keep frequency data from 50 Hz to 5000 Hz, and the spectrograms are stored in matrices for each dictionary word.

The input audio from the microphone is saved in an array that appends consecutive audio frames, continuously updating while the program is running. The last two seconds of the real-time audio are displayed to ensure proper functionality. Word recognition is attempted after each newly appended audio frame.

Word Recognition State Machine (Detect, Save, Process):

Detect: The algorithm searches for the beginning of a word by computing the average energy of the most recent frame and checking if its value is greater than a predetermined threshold.

Save: If successful, the "successful" frame and the five following frames are appended chronologically. An additional time-domain plot of the six-frame word is displayed on the screen. The algorithm detects and saves words until the number of detected words is equal to three, corresponding to the grammatical structure of all the expected mathematical equations.

Process: Word recognition occurs when three words have been detected. The algorithm's main goal is to determine which dictionary word has the highest correlation with a given detected word. The correlation of each detected word and the dictionary of words is calculated by creating the spectrogram of the detected word, calculating the correlation between the detected word and the template spectrograms with MATLAB's corr2(), saving the max correlation for each set of word-specific templates, and finding the actual max correlation of the word-specific maximums. To improve accuracy, since the grammatical structure is predetermined (number, operator, number), the first and third words in the sentence will only be compared to the number templates. Subsequently, the second word will only be compared to operator templates.

The final piece of processing is the calculator function. The words are turned into numerical values with MATLAB's mapping structure. The numerical values are used to find the real answer to the recognized equation. The three words are displayed in a full sentence, and everything in the algorithm resets, primed to detect another equation.

C. Methods Used to Determine Initial Algorithm Parameters

Calculating the average power threshold: To determine the detection threshold, Yvie was recorded saying each word in the preliminary dictionary: {"One," "Two," "Three," "Four," "Five," "Six," "Plus," "Add"}. A MATLAB script computed the average power per frame for each word. The first non-zero frame power across all words was identified. The average power per frame of ambient noise was calculated to ensure the determined threshold was large enough to remove ambient noise false positives.

Frame Collection: The number of frames collected for each detected word was determined by recording a one-second

audio input for all dictionary words. The amplitude and power of each dictionary word were plotted in the time domain, adding red vertical lines at each frame to the power plot to visualize the length of each word in frames. The number of frames of the longest word was used to save detected words.

D. Methods Used to Calculate Accuracy & Revise Algorithm

Baseline Accuracy Calculation: To establish baseline accuracy, we conducted ten consecutive trials, one for each word in the dictionary. In each trial, the real-time recognition algorithm ran continuously while Yvie repeated the target word ten times. The program output the dictionary word with the highest correlation in both the time and frequency domains. The results were collected for each dictionary word ("one"–"six," "plus," "add"), and word counts were compiled into raw confusion matrices. ChatGPT was used to generate the confusion matrices from these counts, and the accuracy of ChatGPT was verified by cross-referencing with the raw data.

Spectrogram Analysis and Algorithm Revisions: To investigate possible improvement across spectrogram correlation, spectrograms were generated for all template words (80 total) using the baseline parameters defined in the algorithm. The visual change in the plotted spectrograms was observed as parameters varied. Qualitative data were used to create greater visual distinction between dictionary words, therefore increasing the difference between max correlations across templates. *This test was a continuation of work completed in Module 1 Demo 3, where additional spectrograms were generated for the words "Hi" and "Map," varying the window type, length, and overlap while keeping parameters consistent across templates.

Frequency mask: Spectrograms and FFT plots of ambient noise were also recorded to verify that ambient noise shows a higher amplitude at low frequencies (< 200 Hz). Additionally, the upper bound of dictionary words was also observed.

Baseline on Updated Spectrogram in Algorithm: These tests led to algorithmic updates (Rev0 → Rev1). Specifically:

1. The correlation function was updated from *xcorr2()* to *corr2()* since the new real-time recognition synchronized spectrograms.
2. Spectrogram parameters were modified to: window length = 300, hop size = 30, overlap = 90%, NFFT = 2048, frequency range = [50, 5000] Hz.

An additional accuracy test replicating the baseline evaluation was conducted using the Rev1 algorithm. Results were compiled into confusion matrices.

Final Algorithm Evaluation (Rev2 Algorithm)**:** The final evaluation tested the real-time detection and recognition program, which used Rev1's parameters but contained a larger dictionary, {"One," "Two," "Three," "Four," "Five," "Six," "Plus," "Add," "Minus," "Times," "Over"}. To construct the test set, ChatGPT was prompted to generate 20 random equations using all available numbers and operations. Yvie spoke these equations sequentially into the Rev2 algorithm. For each equation, the spoken words were recorded manually along with the algorithm's detected values. Using the previous information, the final percent accuracy was calculated.

III. RESULTS

Calculating the average power threshold: Final Algorithm Evaluation (Rev2 Algorithm): The final evaluation tested the real-time detection and recognition program, which used Rev1's parameters but contained a larger dictionary, {"One," "Two," "Three," "Four," "Five," "Six," "Plus," "Add," "Minus," "Times," "Over"}. To construct the test set, ChatGPT was prompted to generate 20 random equations using all available numbers and operations. Yvie spoke these equations sequentially into the Rev2 algorithm. For each equation, the spoken words were recorded manually along with the algorithm's detected values. Using the previous information, the final percent accuracy was calculated.

Table I

Average Power Values in First Frame

Spoken Word	Computed Power
"One"	0.00160
"Two"	0.00380
"Three"	0.00087
"Four"	0.00300
"Five"	0.00080
"Six"	0.00580
"Plus"	0.01070
"Add"	0.00062

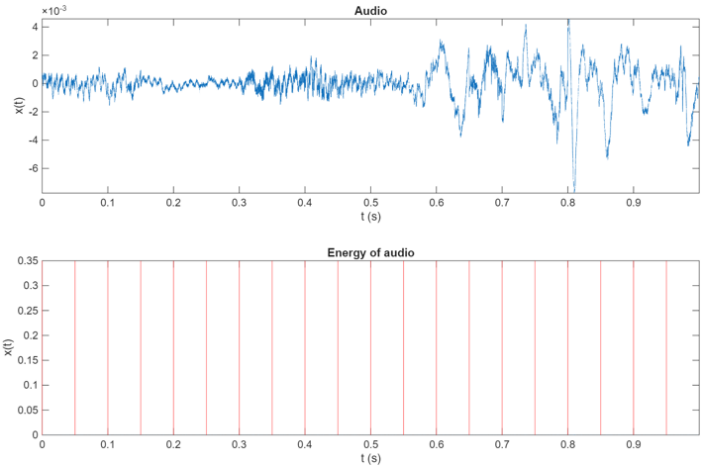

Fig. 1: *Raw audio input of ambient noise (Top) and corresponding power with red lines indicate sampling frames (Bottom). Using a constant y-axis scale for word and background noise plots, average power of noise is practically nonexistent for background noise.*

Frame Collection: All words in the preliminary dictionary were plotted as shown for "One" in. 2. Among these, the

shortest word “Six” and the longest word “One” required one and four frames, respectively. On average, each word spanned approximately three frames. Fig. 2 shows a microphone tap plotted within a frame for reference, providing an intuitive understanding of frame duration. Based on these results, the total sample length was set to six frames or 0.3 s, allowing for variability in word length. While this approach worked well for monosyllabic words, the increased length of multisyllable words like “minus” increased the probability of an error.

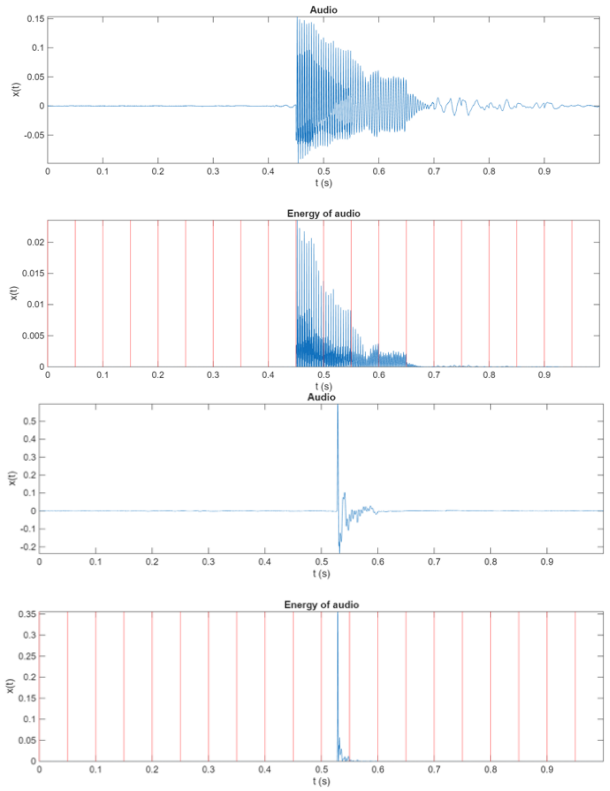

Fig. 2: *Raw audio input of “One” (Top) and corresponding power (Second from Top). Raw audio input of a microphone tap (Second from Bottom) and corresponding power (Bottom).*

Baseline Accuracy Calculation: The baseline time-domain test achieved an accuracy of 70% as shown in Fig. 3. The frequency-domain test achieved 52.5%, as shown Fig. 4.

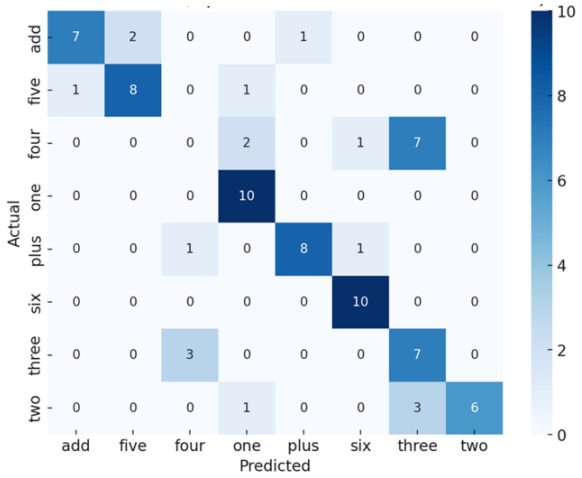

Fig. 3: *Time-domain raw count confusion matrix. Each element represents the number of times a spoken word (columns) was identified as the highest correlation match using xcorr() in the time domain.*

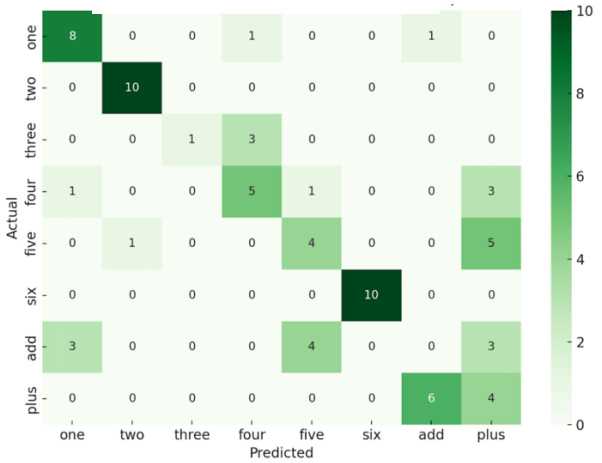

Fig. 4: *Frequency-domain raw count confusion matrix, using corr2(). Spectrogram parameters: window length = 1200, hop size = 400, NFFT = 2048.*

Spectrogram Analysis and Algorithm Revisions: Spectrograms of all templates were analyzed. Fig. 5 shows the change in distinction as spectrogram parameters are changed. From analysis of accuracy, it was concluded that an increase in qualitative distinction was associated with a high quantitative distinction in max template correlations.

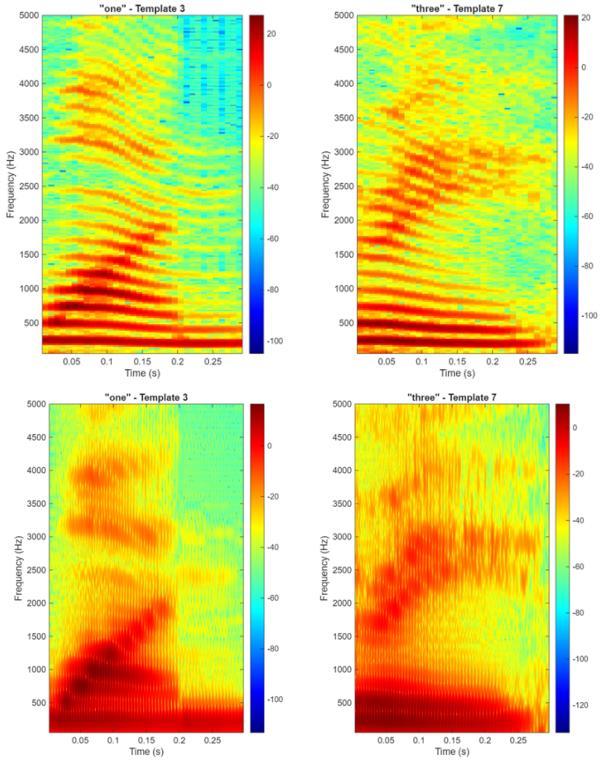

Fig. 5: *Spectrogram plots of the word templates “One” and “Three.” The top row uses parameters of window length = 1200, hop size = 400, and NFFT = 2048. The bottom set uses updated parameters of*

window length = 300, hop = 30, overlap = 90%, NFFT = 2048, and frequency range = [50, 5000] Hz.

Also, Fig. 6 shows that most of the color in a spectrogram occurs between 50 Hz and 5000 Hz. This analysis was consistent across all dictionary words.

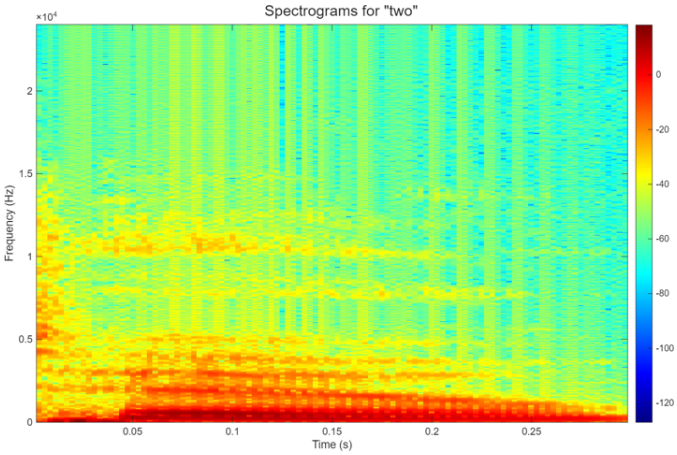

Fig. 6: *Spectrogram of the word "two", frequency range [0, 25000] Hz.*

FFT and spectrogram analysis of ambient noise (Fig. 7) revealed dominant noise contributions below 200 Hz, with peak energy between 0 and 50 Hz. Reducing the window length and increasing the overlap improved variance across word templates over selected frequency bands. This information helped determine the frequency mask [50 Hz to 5000 Hz] for our spectrograms.

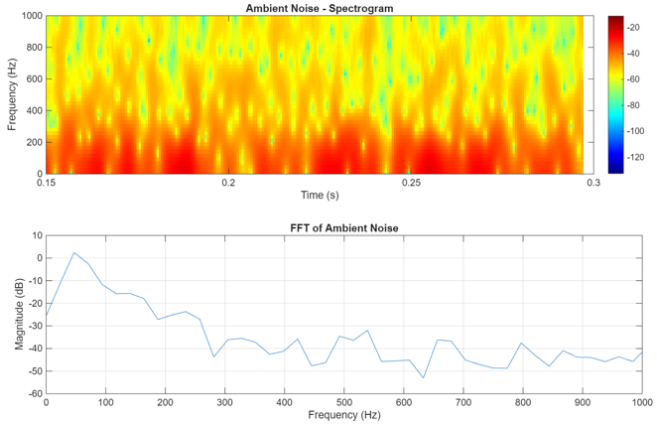

Fig. 7: *Spectrogram of the word "two", frequency range [0, 25000] Hz (Left). Spectrogram (top) and FFT (bottom) of the same noise signal, frequency range [0, 1000] Hz.*

Baseline on Updated Spectrogram in Algorithm: With revised spectrogram parameters, frequency-domain testing achieved an accuracy of 92.5% in Fig. 8. This represents a 76% improvement over the baseline. Due to the jump in accuracy using corr2() and improved spectrogram parameters, time-domain correlation was omitted from the algorithm.

Additionally, the increase in accuracy was associated with a change in the range of correlation difference between a success and the next highest. The original algorithm yielded a range of [0.08 to 0.15], while the improved algorithm ranged from [0.16 to 0.40].

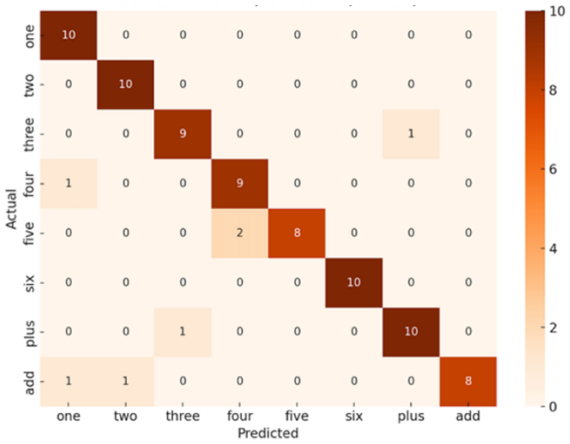

Fig. 8: *Frequency-domain raw count confusion matrix, where each element represents the number of times a spoken word (columns) was identified as the highest correlation match using xcorr2() in the frequency domain (rows). Parameters: window length = 300, hop = 30, overlap = 90%, NFFT = 2048, frequency range = [50, 5000] Hz.*

Final Algorithm Evaluation: The final real-time word detection and speech recognition algorithm achieved an accuracy of 60% (Table II). One incorrect word recognition in an equation resulted in an incorrect overall answer. Misclassifications were as follows:

- "Times" mis-recognized as "Add" (3 instances)
- "One" mis-recognized as "Four" (2 instances)
- "Two" mis-recognized as "Four" (2 instances)
- "Six" mis-recognized as "Two" (1 instance)
- "Three" mis-recognized as "One" (1 instance)

III. Discussions/Conclusions

The speech detection extension aimed to improve accuracy and expand capability. At the week 3 checkpoint, the algorithm achieved 55% overall accuracy, with 44% in frequency-domain comparison. Following redesign, individual frequency-domain word accuracy improved to 92.5%. Overall equation accuracy reached 60%. To have a successful equation, it is necessary to have three individual words successful in a row, explaining some of the drop from 92.5% to 60%.

A key factor was reducing the audio length of a word: the original two-second window increased computational cost, constrained parameter tuning, and introduced zero-padding effects that reduced word distinctness. Shortening the sampling window increased the average correlation margin between the top two correlating words.

Capability also advanced significantly. The week 3 system relied on timed input, whereas the final implementation enabled true real-time word detection and sampling. In addition, the redesigned spectrogram test eliminated the need for manual frequency band selection and enabled automatic comparison across all templates. Template management was also streamlined, allowing new signals to be added without code modification. These improvements facilitated expansion of the calculator's functionality from simple addition to include subtraction, multiplication, and division.

Future work to improve accuracy would look to fix the thresholding bug that occurs when a word has an average power greater than the threshold after the sixth frame.

IV. Appendix

Table II
Accuracy Test for Rev2-Algorithm,

Spoken Equation	**Algorithm Answer**
3 + 5	True
6 - 2	True
4 x 1	Incorrect
2 / 6	True
5 – 1	Incorrect
1 + 4	True
6 x 3	Incorrect
2 + 2	Incorrect
3 / 1	Incorrect
5 x 4	True
4 – 6	True
1 x 1	Incorrect
6 / 3	Incorrect
2 – 5	True
4 + 6	True
3 x 2	Incorrect
5 / 2	True
6 + 1	True
2 – 4	True
1 / 6	True

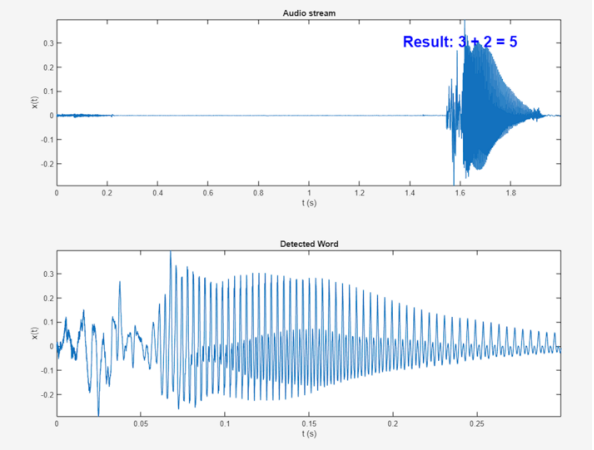

Fig. 9: *Interface of Real-Time word Detection Sampling, Word Recognition, and Algorithm Calculation of Spoken Equation.*

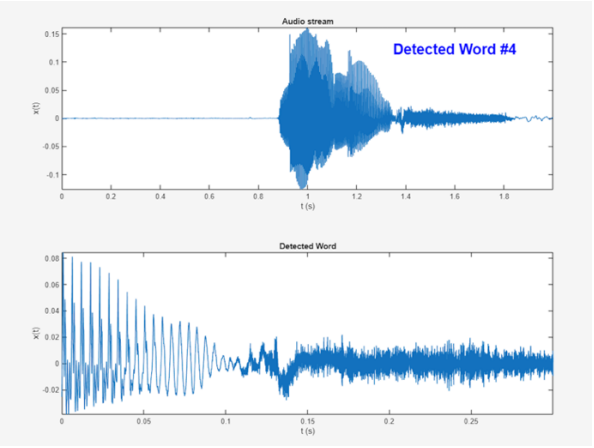

Fig. 10: *"Minus" Spoken as Input, Plot shows Real-Time word Detection Sampling and Word Recognition mis-detecting the start of a word at the end of a dual syllable word "Minus".*