Preliminary report

Authors: Lars Vonk, Aditya Patel, Luka Železnik

This preliminary report gives a short description of our project for image and speech recognition "speech controlled calculator" with used algorithms and a short explanation of the selected datasets.

Description of project

Description

Our project will focus on making a calculator that can be controlled using speech. This means that digits and operators can be input using speech into a microphone or other speech detection device. The audio input will be converted into digits and operators in the system. The system will output the given input and the result of the equation after each spoken digit e.g. 1 + 10 = 11. After this a special word can be spoken to indicate to the system that the input is done and the recording will stop.

Motivation

We chose this challenge since we are interested in working with speech recognition and want a project that is interactive. We thought of a project that not only recognizes selected words from a dictionary but also uses them in an interactive and potentially useful way. This application will be lightweight and domain specific. Potential end users are people who have difficulty controlling input devices where physical contact is needed, such as people with disabilities.

Minimal viable product

Our product will be at least able to pick up single digits as input (0-9) and the simple math operators, which are plus, minus, divided by and multiplied by. Where the input has minimal background noise, this means something similar to a quiet classroom or a bedroom. We will also add an indicator for the timing of each input which will be like a progress bar of some sort that resets every X seconds and will display to the user what timing their inputs are needed.
If the project is completed prematurely we will make an effort to extend the inputs with multi digit numbers e.g. ten, twelve, twenty-three. And we will try to add more complex math operators. With a noisier input like a cafe or open street. The focus of these will be ordered as follows, increased background noise, more complex operators and multi digit input.

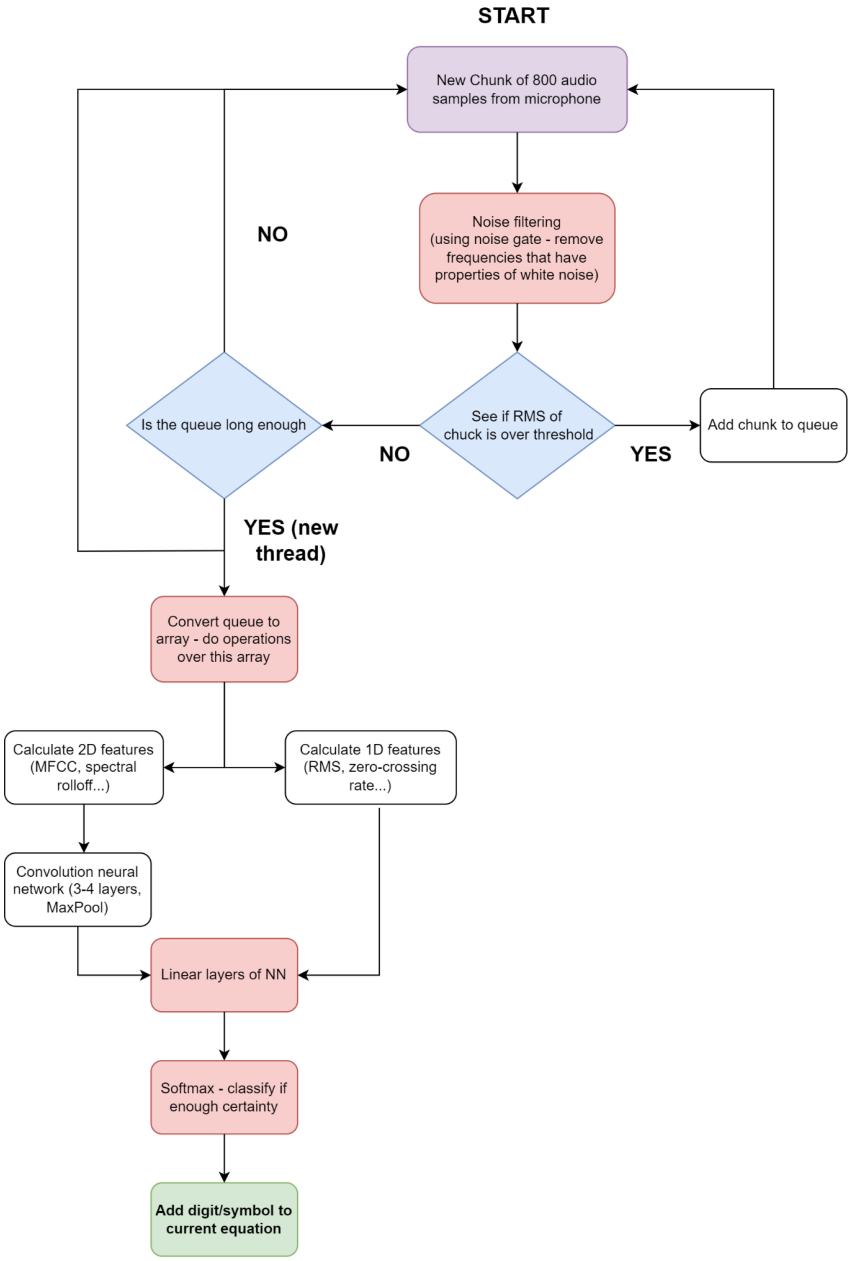
Description of the algorithm

We are going to use a classification neural network with audio preprocessing steps. The input data will be a live microphone recording of audio with a sampling rate of 8000 Hz. We will do an analysis on 20-80 sample chunks of the live recording. The first step is noise reduction of the chuck - filtering to remove unwanted (noisy) frequencies. After that, we are going to calculate the energy of the now filtered chunk using RMS. If the value is above the threshold we will add this chunk to a queue of chucks that represent a word. When RMS is below the threshold we know the word is over - we can begin the classification.

The main features for the classification of digits and symbols, are going to be a MFCC (Mel-frequency cepstral coefficients) of known dimensions that will be used with convolutional neural networks. We will make sure that our MFCC feature has a constant amount of columns with overlapping frames - so if we have a short sample, the bins will overlap, but for longer samples not as much. With this modification we make sure that the MFCC matrix will have constant dimensions.

The CNN will have at least 2 convolutional layers and have Max Pooling operations between the layers. We will also experiment with different configurations. At the end, linear NN layers will be added. We also want to experiment with the Mel spectrogram as the main feature, since it is more robust and easier to interpret. Additional features such as RMS, zero crossing rate, and spectral rolloff may be added in linear layers if they produce better results. If the feature produces a 2d matrix as does spectral contrast we will use a CNN. Our goal is to experiment with different features to find a good compromise of speed and accuracy. For feature extraction, we will use the librosa python library, and for neural networks, we will use Pytorch.

We also want to try other classification methods such as SVM with RBF kernel, which typically performs well and is faster and more efficient than a neural network. However, using neural networks for classification is our main focus.



Flowchart of algorithm

Existing solutions

Here are some excerpts from papers covering this topic. The first is a general overview of methods for speech recognition and when to use them, the second is a system for detecting digits over the phone for use in call centers and third is covering advantages of convolutional neural networks for speech recognition.

"This paper gives an overview of techniques developed in each stage of speech recognition and helps with choosing the technique along with their relative merits & demerits" [1].

"The zero to nine digit utterances for speech data were collected. The speech digit recognition mainly involves two parts, one is feature extraction and other one is feature matching. The main approach is to isolate the speech recognition by Cepstrum and vector quantization. The cepstrum technique is used for feature extraction and vector quantization is used for feature matching. The result shows that all digits give good performance. The proposed speech digit recognition algorithm is implemented by using MATLAB software" [2].

"This paper aims to provide some detailed analysis of CNNs. By visualizing the localized filters learned in the convolutional layer, we show that edge detectors in varying directions can be automatically learned. We then identify four domains we think CNNs can consistently provide advantages over fully-connected deep neural networks (DNNs)" [3].

[1]
S. K. Gaikwad, B. W. Gawali, and P. Yannawar, 'A review on speech recognition technique', *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, 2010.
[2]
M. D. Rudresh, A. S. Latha, J. Suganya and C. G. Nayana, "Performance analysis of speech digit recognition using cepstrum and vector quantization," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 2017, pp. 1-6, doi: 10.1109/ICEECCOT.2017.8284580.
[3]
J.-T. Huang, J. Li, and Y. Gong, 'An analysis of convolutional neural networks for speech recognition', in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4989–4993.

Characteristics of dataset

While searching for data that can be used in this project we found a large dataset of spoken digits. This fits very well with our goal and can be used to train our model. For operators, this was a little harder and we found another project that has similarities to our project. This project contains a small dataset of spoken basic math operators. We have decided to add to this dataset with our own voices and some extra voices that we can gather (about 5 audio files per basic math operator per other person). We are hoping to get at least 40-50 audio files per math operator to train the model. The training samples have a low sampling rate, but as we are doing speech recognition that should be fine (phones operate with the same sampling rate). Before we start training with this dataset we will do all necessary preprocessing steps, including noise reduction. We will split this data into a training, validation and test set.

We are also going to add our own voices to the dataset (saying operators).

Name	**Source**	**Sampling rate**	**Format**	**Number of recordings**
Digits	Kaggle	8000Hz	WAV	3000
Operators	Github	8000Hz	WAV	2500
Operators	Self	8000Hz	WAV	160-200
Special word	Self	8000Hz	WAV	40-50

One thing we have to keep in mind with the audio samples is that the audio samples that come from the dataset are probably clean audio from voice actors who have no to very little accent in the English language and will have low background noise. Because of this, it will be good to create our own samples which contain varying levels of accents and background noise and will also contain audio recordings of both males and females.