High Accuracy Speech Recognition for Industrial Machinery Operation

Liam Kelly
Executive Summary Report
September 26th, 2022

1. Introduction

Speech recognition is a powerful tool that can be used as an alternative method for accepting user inputs. In industrial settings, traditional methods of user input (e.g., buttons, levers, pedals) can be difficult to operate when wearing protective clothing, or if workers are physically impaired. By developing a highly accurate speech recognition model, voice commands can be used to replace traditional user inputs in areas where they are cumbersome [1].

For the speech command dataset used in this report, existing benchmark models have demonstrated a keyword spotting accuracy of over 98%. While it’s unlikely to exceed that accuracy here, the goal of this report is to investigate the use of various machine learning models and to better understand their advantages and limitations.

2. Speech Commands Dataset

Dataset Information

The dataset was obtained from “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition” [2]. It is a well-known audio dataset for training and evaluating keyword spotting systems.

Dataset Description

The dataset consists of 30,596 .wav audio files of eight different spoken words: up, down, left, right, start, stop, yes, and no. Each audio file is provided at the same sampling rate (16 kHz), however there is some variance in the length of audio recordings, varying from 4778 data points to 16000 data points. Note 91% of all samples have a length of exactly 16000 data points (i.e., are 1 s in duration).

The dataset is provided as a folder with 8 subfolders, one for each spoken word, such that the spoken content of each audio file can be determined from its subfolder’s name. The distribution of files by word is fairly even, the word with the most audio samples, “yes”, having 4044, and the word with the least, “up”, having 3723. It should be noted that there is some variance in the quality of the spoken words. Some are easily distinguishable by ear, while others are difficult to understand due to poor spoken quality or due to the audio being cut-off. This should be considered when interpreting model accuracy.

3. Data Cleaning and Preprocessing

3.1 Audio Feature Creation

To use the audio files from the data set in traditional machine learning models, features must be created to describe properties of the waveform. To do this, several equations were adopted from Signal Processing Methods for Music Transcription [3], such as the signal’s centroid, bandwidth, and skewness among others. These audio features can be used to characterize the sound’s properties. Since audio files can be analyzed in both the time domain and frequency domain, the equations can be applied in both the time domain and frequency domain to extract additional information from the audio file.

The process of creating a data frame of all signals and their signals is outlined in Figure 1. Waveforms of each signal (in both the time and frequency domains) are subjected to a series of equations. In the resulting data frame, the audio file becomes a row, with each column storing the output of a particular equation applied to the audio file.

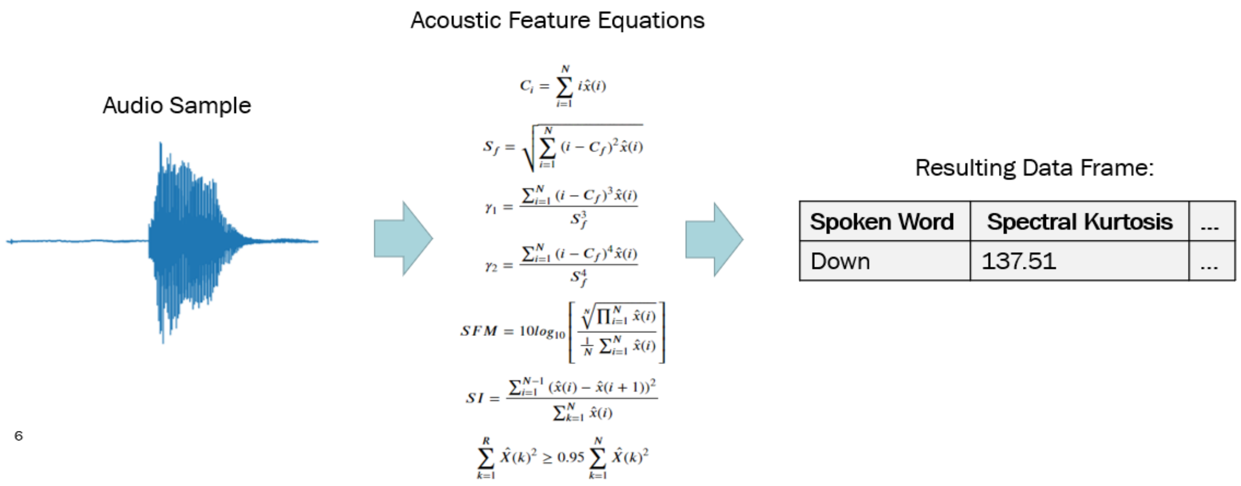

Acoustic Feature Equations
Audio Sample
6
$C_i = \sum_{i=1}^{N} i\hat{x}(i)$
$S_f = \sqrt{\sum_{i=1}^{N} (i - C_f)^2 \hat{x}(i)}$
$\gamma_1 = \frac{\sum_{i=1}^{N} (i - C_f)^3 \hat{x}(i)}{S_f^3}$
$\gamma_2 = \frac{\sum_{i=1}^{N} (i - C_f)^4 \hat{x}(i)}{S_f^4}$
$SFM = 10log_{10}\left[\frac{\sqrt[N]{\prod_{i=1}^{N} \hat{x}(i)}}{\frac{1}{N}\sum_{i=1}^{N} \hat{x}(i)}\right]$
$SI = \frac{\sum_{i=1}^{N-1} (\hat{x}(i) - \hat{x}(i+1))^2}{\sum_{k=1}^{N} \hat{x}(i)}$
$\sum_{k=1}^{R} \hat{X}(k)^2 \geq 0.95 \sum_{k=1}^{N} \hat{X}(k)^2$
Resulting Data Frame:
Spoken Word
Spectral Kurtosis
...
Down
137.51
...

Figure 1: Flowchart for transforming an input audio sample waveform into a row in the resulting data frame by applying the equations for each acoustic feature.

The resulting data frame is then cleaned by inspecting for null and inf values and dropping rows which contain them. Classification models are often built on the assumption that the input data has a normal distribution along each of its features. To improve the normality of the distributions for each feature, mathematical operations such as the log, inverse, or square root can be applied to the data. A trial-and-error approach was adopted, and operations which produced improvements in the feature's normality were selected to overwrite the data in the data frame.

Box plots were then used to visualize the distribution of the different spoken words along the different audio features as seen in Figure 2. The box plots showed some good signs of inter-group variance for different spoken words, indicating that these features will be effective in determining spoken words through a classification model.

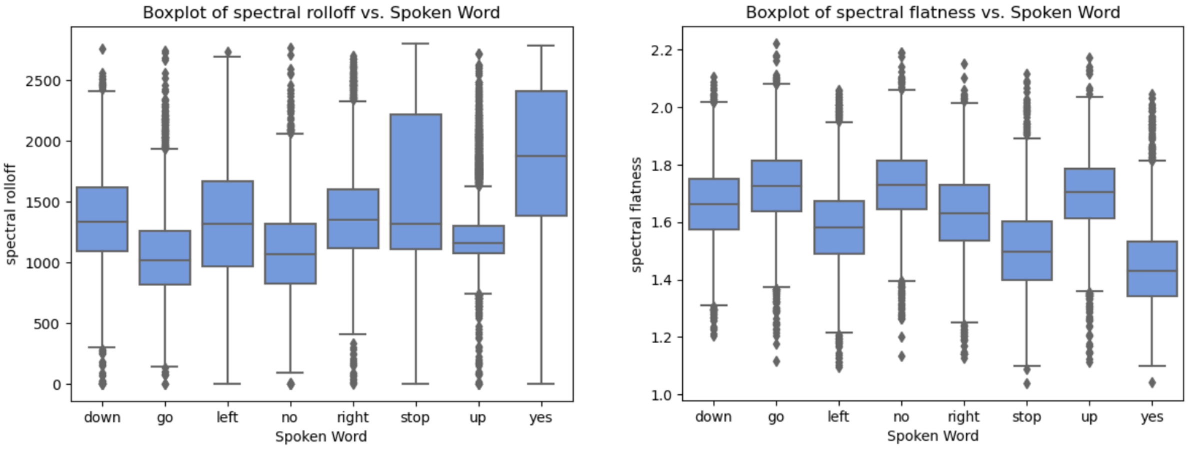


Figure 2: Box plot visualizations of the distribution of each spoken word along the spectral roll-off feature (left) and the spectral flatness feature (right).

3.2 Spectrogram Creation

Convolution Neural Networks (CNN) are extremely powerful models when it comes to image classification. Fortunately, there is a representation of sound files that appears as an image: spectrograms. Spectrograms visualize how the frequency components of an audio sample evolve over time. They plot frequency on the y-axis and time on the x-axis, the value of each point in the spectrogram represents the amplitude of a given frequency at a given time. To create spectrograms, a custom function was developed that calculates the spectrogram of an input waveform, as shown in Figure 3. This function was used to create a unique spectrogram for every audio file in the dataset.

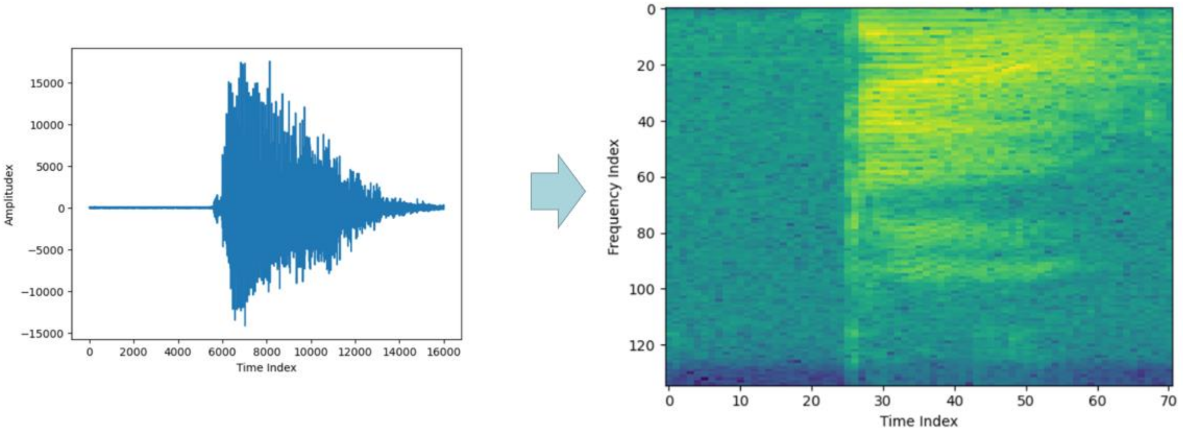


Figure 3: The audio sample's waveform (left) is used to create a spectrogram (right) using the spectrogram function.

4. Modelling and Results

4.1 Traditional Classification Models

Three different classification models were evaluated for their ability to accurately predict the spoken words of audio samples: logistic regression, random forest, and XGBoost. The hyperparameters of each model were optimized using GridSearchCV from sci-kit learn. The test set accuracy results from each model are summarized in Table 1.

Table 1: Test set performance metrics of the machine learning models explored in this report.

	Logistic Regression	Random Forest	XGBoost
Accuracy (%)	42.2	51.3	**53.3**
Macro Precision (%)	41.4	50.7	**52.8**
Macro Recall (%)	42.1	51.3	**53.3**

Out of the three models, XGBoost performs the best. Not only does it have the best accuracy, but it also benefits from being a decision tree based model, meaning it is insensitive to feature scale and multicollinearity. However, the model performance is still underwhelming. In the confusion matrix (Figure 4), it can be seen from the off-diagonal elements that the model frequently misclassifies audio samples in the test set. Furthermore, the test set accuracy is only 53%. While this accuracy is much better than the baseline accuracy of 12.5% (accuracy of selecting the same word every time), it needs to be much better to be implemented as a method of user input for industrial machinery.

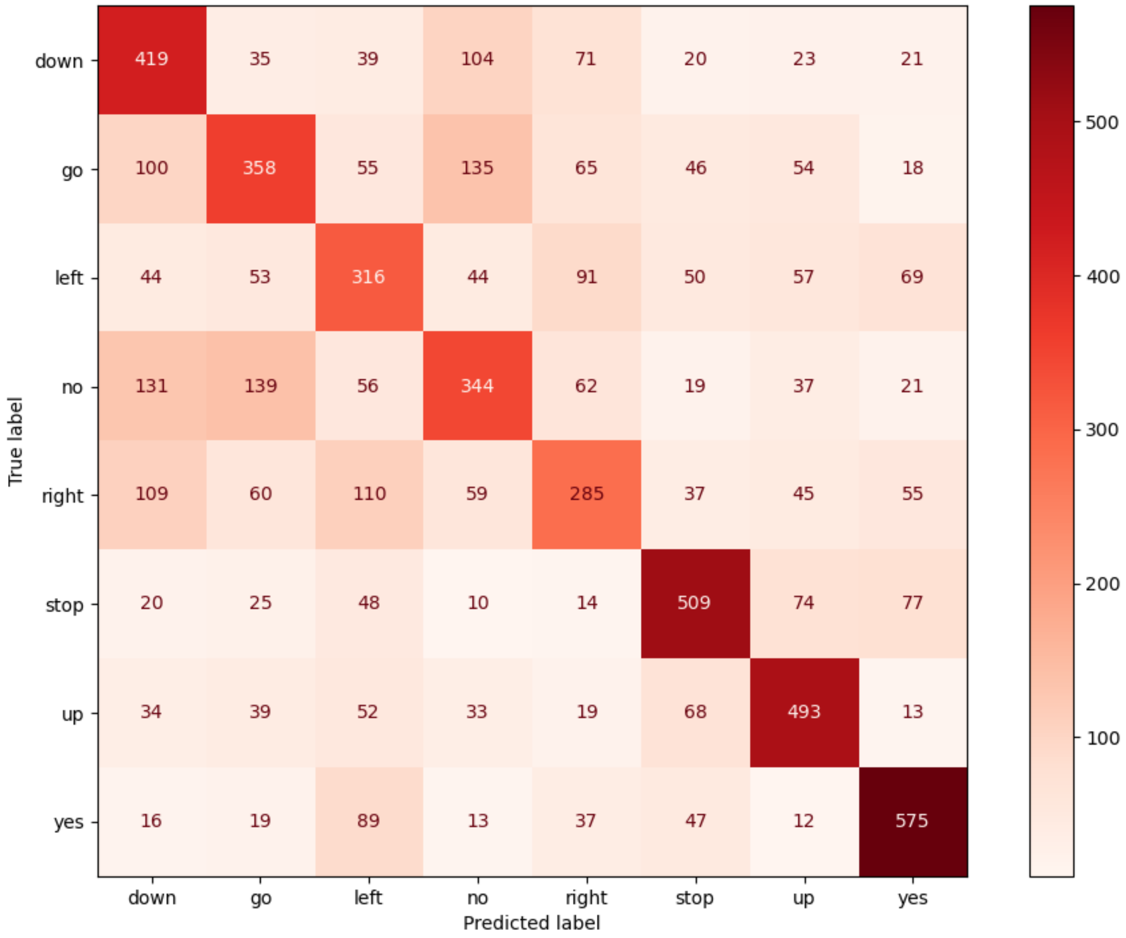
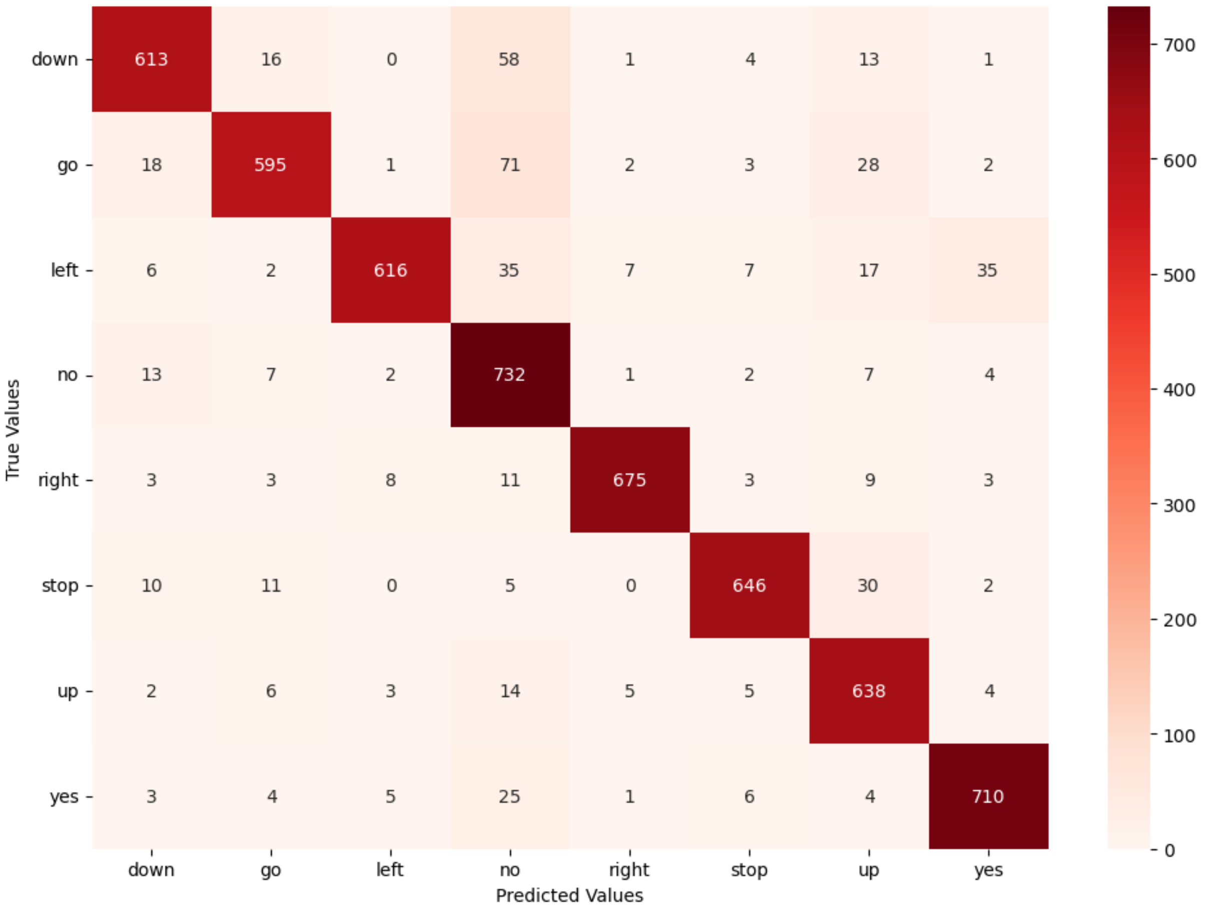


Figure 4: Confusion matrix of the test set results from the XGBoost model.

4.2 Convolution Neural Network (CNN) Model

The previously computed spectrograms were then used to train and evaluate a CNN model. The CNN model consisted of several convolution layers which then fed into a series of flattened dense layers. The CNN model demonstrates an impressive test set accuracy of 93.4%. The confusion matrix for this model is shown in Figure 5. The values in the confusion matrix are well confined to the diagonal elements, indicating excellent performance.



True Values \ Predicted Values	down	go	left	no	right	stop	up	yes
down	613	16	0	58	1	4	13	1
go	18	595	1	71	2	3	28	2
left	6	2	616	35	7	7	17	35
no	13	7	2	732	1	2	7	4
right	3	3	8	11	675	3	9	3
stop	10	11	0	5	0	646	30	2
up	2	6	3	14	5	5	638	4
yes	3	4	5	25	1	6	4	710

Figure 5: Confusion matrix of the test set results from the CNN model.

5. Future Work

While the CNN model performs well on words that are in the training set, it is not robust enough to handle words that are not in the training set. As an experiment, the CNN model was tested on how it would classify a test audio sample of someone saying “hamburger”. The model predicted “down” with 95% confidence, as the “hamburger” spectrogram is much closer to the “down” spectrogram than the 7 other spoken words. To improve the model’s performance on recognizing foreign words and sounds as sounds which are not one of the voice commands, the model should be trained on a much larger vocabulary of words and sounds.

6. Conclusion

To conclude, various classifications models were investigated for their ability to correctly classify spoken words. Traditional classification models required feature creation, done by applying equations to the recorded waveforms. However, these models exhibit underwhelming performance, accuracy scores were at best 53%. CNN models require an image input, so spectrograms were created from the audio dataset. The CNN model exhibited excellent accuracy, scoring 93% on the test set. Our CNN model continues to validate the application of speech-recognizing machine learning models to industrial machinery due to its excellent accuracy. Moving forward, this model requires work to improve its robustness in responding to words beyond its vocabulary.

References

[1] Drives and Controls, "Voice control comes to industrial applications," 21 February 2022. [Online]. Available: https://drivesncontrols.com/news/fullstory.php/aid/6956/Voice_control_comes_to_industrial_applications.html.

[2] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," 9 Apr 2018. [Online]. Available: https://arxiv.org/abs/1804.03209.

[3] A. Klapuri and M. Davy, Signal Processing Methods for Music Transcription, Springer US, 2006.