CS522 Assignment 2 Report
Team Members: Hsun Jui Chang, Jaachinma Okafor, Timo Salisch
October 21, 2022

Data Collection Process:

For each class, we used the microphone of a smartphone (the recorder) to record the audio signals.

Blender:

Ninja BL450 was used in the data collection. The blender was filled with 500 cc of water and blended for continuous 30 seconds and had a break of 10 seconds between different recordings to prevent overheating of the blender. The recorder (phone) was placed on the same surface as the blender sits on. 20 samples of 30 seconds were recorded.

Vacuum:

Dyson v8 was the vacuum cleaner used for the data collection. The phone was held by the same hand as the one holding the vacuum cleaner while recording the sound. The vacuum cleaner was turned on continuously for 30 seconds. 20 samples of 30 seconds were recorded.

Alarm:

Alarm data collection was recorded using a smartphone. A YouTube video of a fire alarm was played for this recording process. 20 samples of 30 seconds were recorded.

Music:

The first 30 seconds of the same song from the same singer was recorded for 20 times. A YouTube video: The One by the dirty youth was played. The playback device used for music data collection was the built-in monitor speaker. The smartphone (recorder) was placed on the desk facing up for recording.

Microwave:

The microwave was set to defrost mode for 20 minutes to make sure the consistency of the noise. 20 examples of 30 seconds were recorded using a smartphone that was facing up and was placed on top of the table where the microwave sits on**.**

Silence:

The recording was done in a silent dormitory room with only one person in it to operate the recorder. 20 examples of 30 seconds were recorded using a smartphone that was facing up and was placed on the surface of a desk in the dorm**.**

NB: The collection of each class was done across multiple days to account for variations in the sampling instances.

Rationale for features:

When the different sounds are recorded using a microphone, the sounds are shown to possess different features. It has been shown that a computer model can be trained to predict what a sound is using these features.

For the pre-processing, we implemented a high frequency filter that removes frequencies that are higher than 10,000hz.

We then tried two approaches: 1) windowing with a window size of 10 seconds and 5 seconds overlap and 2) using a single window. After that, we buffered the incoming time-domain signal from the microphone and computed the Fast Fourier Transform (FFT) with FFT size = 1024. From the FFT, we created bins of 5 sizes each for the time and frequency bins. Figure 1 shows how these bins vary across the different classes but might not be enough as they can be somewhat similar amongst a few classes.

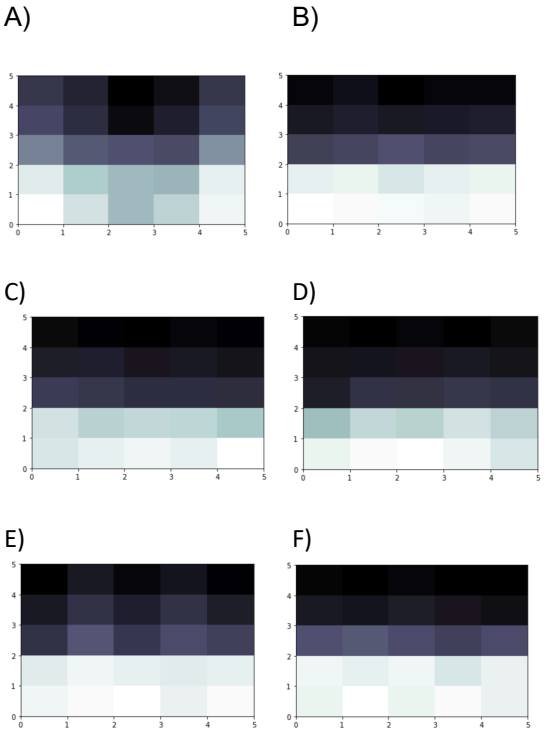

Figure 1: Bins of the different classes A) Alarm B) Blender C) Microwave D) Music E) Silence F) Vacuum

To add to our features selection, we choose some domain specific features: three frequencies with the highest magnitude, the mean magnitude of the

frequencies, median magnitude of the frequencies and the variance of the magnitude of the frequencies. We chose these features because they highlighted the similarities of samples within each class while indicating the differences of samples with other classes.

We tried different combinations of features to come up with the optimal features to build an optimal system. The combination of features tried included:

- 1: Multiple windows with only the spectrogram data as features
- 2: Multiple windows with only the binned spectrogram as features
- 3: Multiple windows with the spectrogram data and the other domain features
- 4: Multiple windows with the binned spectrogram data and the other domain features
- 5: Single window with only the spectrogram data as features
- 6: Single window with only the binned spectrogram as features
- 7: Single window with the spectrogram data and the other domain features
- 8: Single window with the binned spectrogram data and the other domain features

The result of the features will be discussed in the next session.

Results:

We developed two models: an SVM and a Random Forest model. For the SVM model, we had to normalize our features to improve accuracy but that is not required for the random forest. Also, we had to use the linear kernel for the SVM as the other kernel types produce low accuracies, which signifies the features have linear relationship with the classes.

We also compared using the single window and multiple windows (with different window sizes) approach and realized they produce different accuracy levels with regards the SVM and did not affect the random forest accuracy as shown in Figure 2. The lowest accuracy around 30 seconds for SVM may be caused by the length of our data being exactly 30 seconds long. Therefore, a +/- 5 second window at 30 seconds can cause some inaccuracy because there is no data we can compare after 30 seconds. However, the SVM still contributes more than 90% of accuracy and we conclude this is not a significant cause of the accuracy.

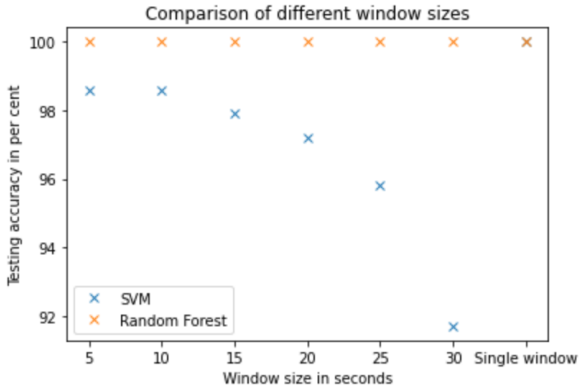

Figure 2: Comparison of Different Window Sizes

With different combination of features selection and window selection approaches, we came up with 8 cases which produced the following results:

Features selection	1	2	3	4	5	6	7	8
SVM	100.0%	66.3%	100.0%	99.4%	100.0%	72.2%	100.0%	100.0%
Random Forest	100.0%	97.8%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

The Features selection are as explained in session 2. It is seen from the table above that the random forest performs very well notwithstanding the feature selection. This shows that the decision tree algorithm performs better with our data for classifying the signals than the SVM algorithm with a linear kernel.

Therefore, we chose the single window approach with binned spectrogram and other domain features. We developed models with accuracy of 100% with the SVM and Random Forest Model.