RTDSP Final Report

CYEE 10828241 Chen Da-Chuan

June 18, 2023

Contents

List of Figures

List of Tables

1 Introduction

This report focuses on classifying the sound in the environment [1]. Possible usage of this is to detect the anomaly in the background. For example, if the environment's sound is always the same, but suddenly there is the sound of a crying baby, then there may be something wrong. Another thing about this report is that this requires very little time for classifying, so it can be used in real-time.

2 Dataset

The dataset I used, ESC-50, is possibly the only option in this field that provides high-quality and well-labeled data.

2.1 ESC-50

This dataset has 40 5-second samples for each category. It provides samples for categories like animals, natural soundscapes & water sounds, human & non-speech sounds, interior/domestic sounds, and exterior/urban sounds.

Table 1: Metadata of ESC-50 dataset

Metadata	**Description**
Total number of audio samples	2000
Categories	50
Sample length (sec)	5
Sample rate (Hz)	44100

2.2 5 Stages of Categorization

Based on the requirement of 5 progress stages from the professor, I assign related categories from the ESC-50 dataset to each stage. The categories are listed in Table 2. Each stage of the experiment will use its categories and the categories from all previous stages.

Table 2: 5 Stages of Categorization

Stage	Description	ESC-50 Categories
1	Cry, Laughter	301 - Crying baby
		307 - Laughing
2	Low Freq, Fan, Motor	501 - Helicopter
		502 - Chainsaw
		505 - Engine
		510 - Hand saw
3	High Freq, Conversation	504 - Car horn
		507 - Church bells
		508 - Airplane
4	Phone/Door Ring	401 - Door knock
		408 - Clock alarm
		409 - Clock tick
5	Traffic, Police, Ambulance	503 - Siren

3 Methodology

The methodology of this report is to extract multiple features from the 5-second audio samples and then use ML algorithms like k-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) to classify the samples. These algorithms, compared with other large models like CNN, RNN, DNN, etc., are much faster and easier to implement, and they require a lot less computing resources and time to perform the classification.

3.1 Process Pipeline

There are 3 stages in developing a working system, model training and evaluating, offline processing, and online processing. For this report, only the first stage is implemented.

The processing pipeline of this report is shown in Figure 1.

1. Acquire audio samples in array format.
2. Extract all of the necessary features from the audio sample arrays and store them in a Pandas data frame.
3. Use the data frame to train the KNN, RF, and SVM models. The samples are divided into five-folds, and each fold will be used as the test set once.
4. Evaluate the models with 5-fold cross-validation.

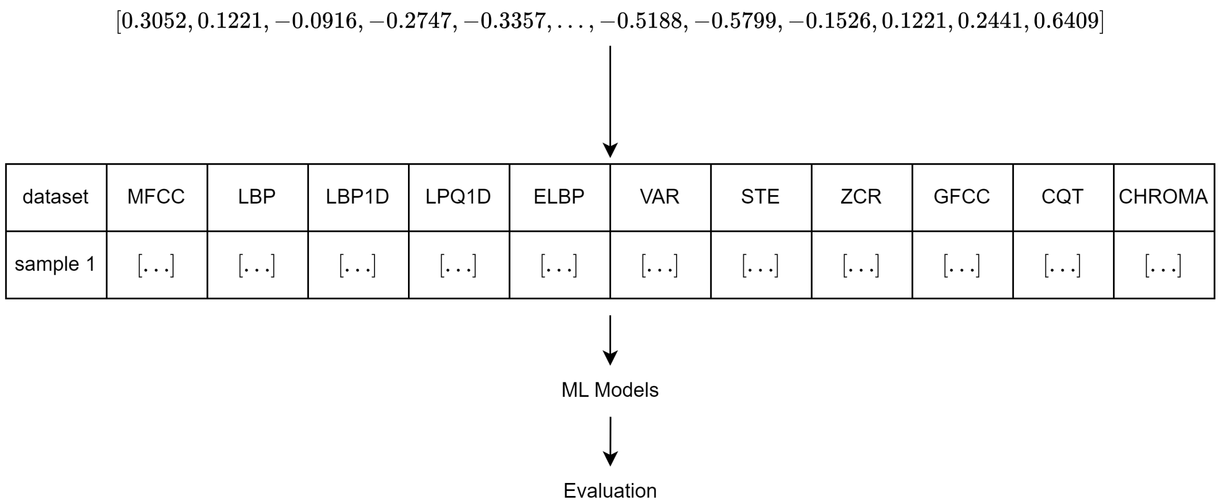

Figure 1: System Flow chart

3.2 Feature Extraction

The features I used in this report are the following:

1. MFCC, Mel Frequency Cepstral Coefficients, figure 4. [2]
2. LBP, Local Binary Pattern, figure 9. [3] [4] [5]
3. LBP1D, LBP for 1D, figure 14, 15. [1]
4. LPQ1D, Local Phase Quantization for 1D, figure 16, 17. [6]
5. ELBP, Extended LBP, figure 11. [3]
6. VAR, Variance LBP, figure 13. [3]
7. STE, Short Time Energy, figure 18. [7]
8. ZCR, Zero-Crossing Rate, figure 19. [7]
9. GFCC, Gammatone Frequency Cepstral Coefficients, figure 5. [8] [9]
10. CQT, Constant-Q Transform, figure 6. [10]
11. CHROMA, Chromagram, figure 7. [11]

These features are extracted with parameters in Table 3.

Table 3: Parameters of Feature Extraction

Frame Size	2048
Frame Shift	1024
LBP Digits	4

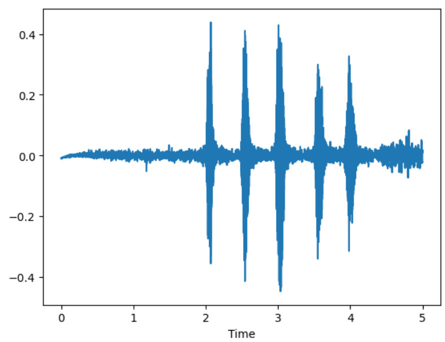

Figure 2: Raw Data

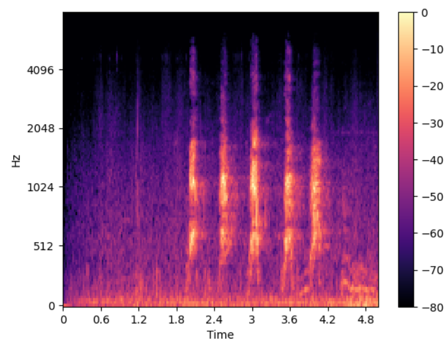

Figure 3: Mel Spectrogram

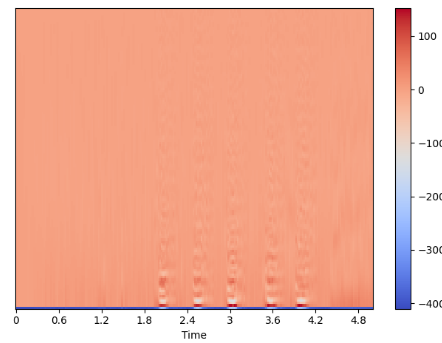

Figure 4: MFCC

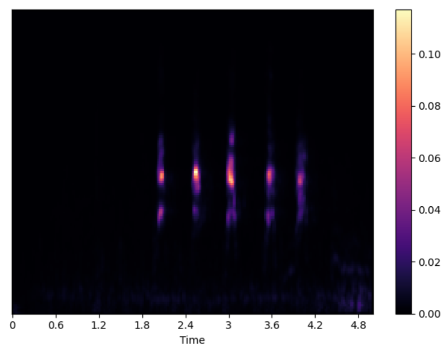

Figure 5: GFCC

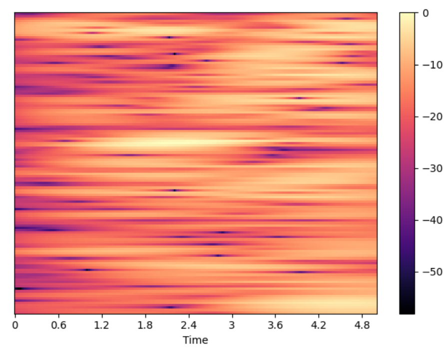

Figure 6: CQT

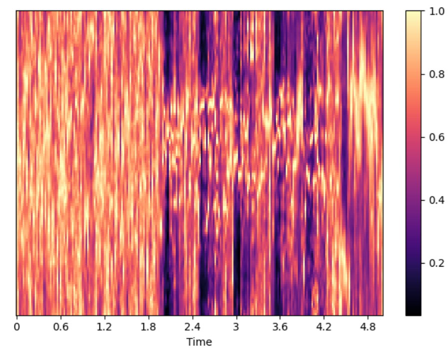

Figure 7: Chromagram

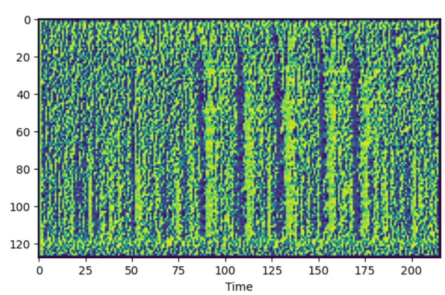

Figure 8: Raw LBP

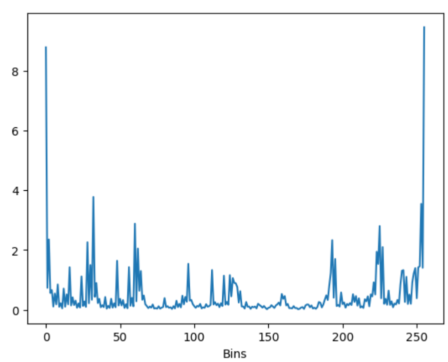

Figure 9: LBP

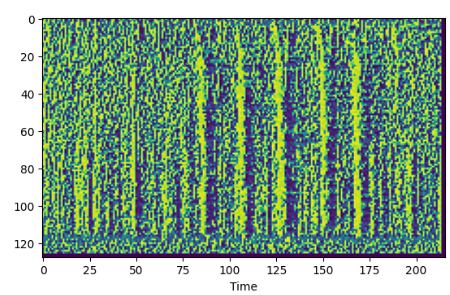

Figure 10: Raw ELBP

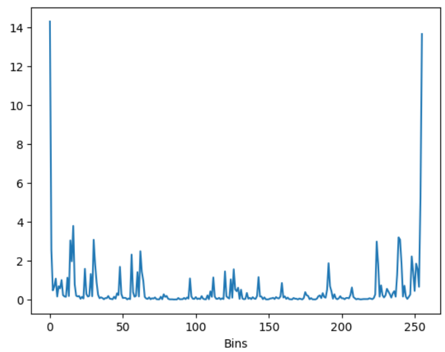

Figure 11: ELBP

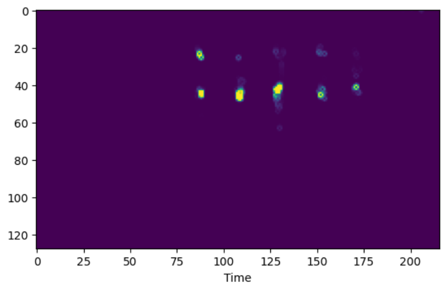

Figure 12: Raw VAR

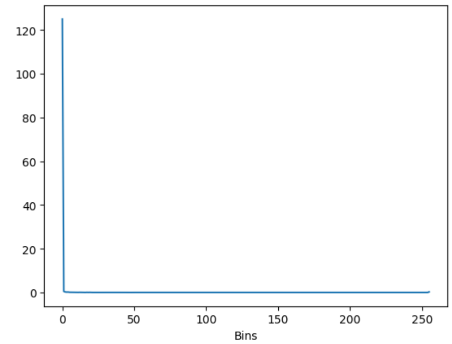

Figure 13: VAR

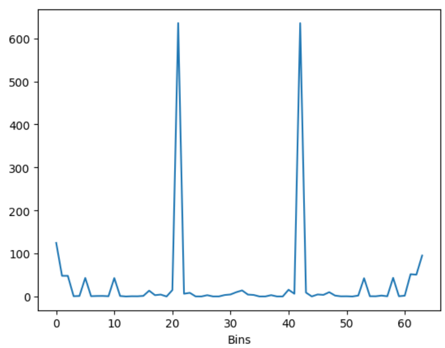

Figure 14: LBP1D 64

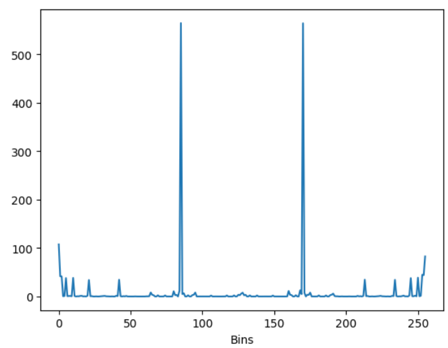

Figure 15: LBP1D 256

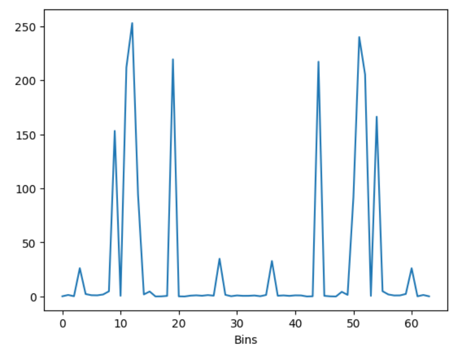

Figure 16: LPQ1D 64

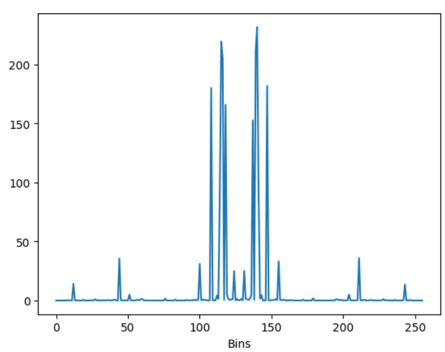

Figure 17: LPQ1D 256

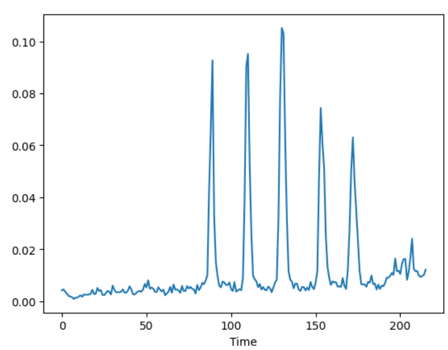

Figure 18: STE

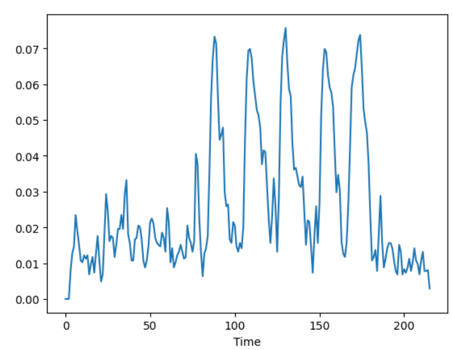

Figure 19: ZCR

4 Results

The following results are based on the machine in table 4.

Table 4: Execution Platform

CPU	Intel Core i7-10750H
Memory	48G
Disk	SP P34A60

4.1 Accuracy

From the accuracy data in table 5, 6, and 7, we can see that RF performs the best and KNN performs the worst. All three models have relatively high accuracy in lower stages, and the performance degrades after more categories are added. Based on these results, there is still room for improvement to get the accuracy high enough for real-world implementation.

Table 5: Accuracy of 5 Stages of Categorization with KNN

Fold	**Stage 1** (%)	**Stage 2** (%)	**Stage 3** (%)	**Stage 4** (%)	**Stage 5** (%)
1	62.5	37.5	34.7	32.2	33.6
2	62.5	41.6	31.9	36.4	33.6
3	75.0	56.2	47.2	47.9	44.2
4	56.2	52.0	37.5	37.5	34.6
5	62.5	45.8	36.1	35.4	32.6
AVG	63.7	46.6	37.5	37.9	32.6

Table 6: Accuracy of 5 Stages of Categorization with RF

Fold	**Stage 1** (%)	**Stage 2** (%)	**Stage 3** (%)	**Stage 4** (%)	**Stage 5** (%)
1	100.0	60.4	51.3	52.0	51.9
2	75.0	58.3	44.4	50.0	54.8
3	93.7	62.5	51.3	58.3	57.6
4	81.2	70.8	55.5	60.4	57.6
5	56.2	50.0	52.7	51.0	53.8
AVG	81.2	60.4	51.1	54.3	55.1

Table 7: Accuracy of 5 Stages of Categorization with SVM

Fold	**Stage 1** (%)	**Stage 2** (%)	**Stage 3** (%)	**Stage 4** (%)	**Stage 5** (%)
1	75.0	52.0	41.6	43.7	45.1
2	75.0	45.8	41.6	51.0	51.9
3	81.2	64.5	52.7	58.3	55.7
4	68.7	47.9	43.0	44.7	42.3
5	43.7	45.8	40.2	43.7	45.1
AVG	68.7	51.2	43.8	48.3	48.0

4.2 Time

The data in the following sections are gathered when experimenting stage 5 dataset, which contains all the included categories and takes the most time out of the five stages.

From the time data in table 8, 9, and 10, all models require around or less than 50 milliseconds to classify a single sample. The speed is fast enough to be considered to be real-time processing. On the other hand, the training and testing time, contrary to large models, is blazing fast.

Table 8: Time of Categorization with KNN

Fold	**Train** (sec)	**Test** (sec)	**Single** (sec)
1	0.033308	0.114038	0.031635
2	0.034510	0.052449	0.038730
3	0.036954	0.036913	0.026042
4	0.036363	0.039878	0.029207
5	0.040900	0.041702	0.032058
AVG	0.036407	0.056996	0.031534

Table 9: Time of Categorization with RF

Fold	**Train** (sec)	**Test** (sec)	**Single** (sec)
1	8.243994	0.066902	0.053297
2	8.157935	0.067697	0.054051
3	8.277063	0.067144	0.052809
4	8.396903	0.066978	0.053018
5	8.265938	0.068693	0.053145
AVG	8.268367	0.067483	0.053263

Table 10: Time of Categorization with SVM

Fold	**Train** (sec)	**Test** (sec)	**Single** (sec)
1	0.202691	0.101736	0.019832
2	0.233394	0.109969	0.017555
3	0.231238	0.107044	0.023253
4	0.227675	0.100112	0.019873
5	0.223826	0.147844	0.019531
AVG	0.223765	0.1113341	0.020009

4.3 Accuracy vs. Time

We can consider both aspects in the same tables (table 11, 12, 13) from the accuracy and time data. We can see that the training time for all three models increases as the number of stages increases, and the single classification time stays the same. However, as the stage number goes up, the accuracy decreases.

Table 11: Accuracy vs. Time with KNN

Item	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**
Acc (%)	63.7	46.6	37.5	37.9	32.6
Train Time (sec)	0.03	0.03	0.037	0.035	0.036
Test Time (sec)	0.047	0.048	0.06	0.047	0.057
Single Time (sec)	0.034	0.025	0.033	0.027	0.032

Table 12: Accuracy vs. Time with RF

Item	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**
Acc (%)	81.2	60.4	51.1	54.3	55.1
Train Time (sec)	0.996	3.013	5.195	8.275	8.268
Test Time (sec)	0.061	0.062	0.062	0.099	0.067
Single Time (sec)	0.056	0.055	0.055	0.071	0.053

Table 13: Accuracy vs. Time with SVM

Item	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**
Acc (%)	68.7	51.2	43.8	48.3	48
Train Time (sec)	0.031	0.078	0.13	0.194	0.224
Test Time (sec)	0.025	0.029	0.044	0.088	0.111
Single Time (sec)	0.021	0.018	0.019	0.021	0.02

5 Summary

The advantage of this approach is that it requires a lot less computing resources and time than other approaches. This means it is possible to deploy on many relatively low computing capability devices, eliminating the need for an internet connection to send data to servers for processing.

However, this approach's low accuracy means that more audio features are needed for small models to categorize environmental sounds accurately.

Despite the little time it takes for this approach to classify audio samples, extracting features from audio is usually really long, averaging around 20 to 30 seconds. A new method is needed to accelerate feature extraction so the entire process can be done online.

The full implementation of this approach is available on `https://github.com/belongtothenight/RTDSP_Code/tree/main/src/esc`.

6 References

[1] Ohini Kafui Toffa and Max Mignotte. Environmental sound classification using local binary pattern and audio features collaboration. *IEEE Transactions on Multimedia*, 23:3978–3985, 2021.

[2] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *International Society for Music Information Retrieval Conference*, 2000.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[4] Timo Ojala, Matti Pietikäinen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of 12th International Conference on Pattern Recognition*, 1:582–585 vol.1, 1994.

[5] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.*, 29:51–59, 1996.

[6] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *International Conference on Image and Signal Processing*, 2008.

[7] T. Zhang and C.-C.Jay Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4):441–457, 2001.

[8] Malcolm Slaney. An efficient implementation of the patterson-holdsworth auditory filter bank. 1997.

[9] Xavier Valero and Francesc Alias. Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, 14(6):1684–1689, 2012.

[10] Christian Schörkhuber. Constant-q transform toolbox for music processing. 2010.

[11] Roger N. Shepard. Circularity in Judgments of Relative Pitch. *The Journal of the Acoustical Society of America*, 36(12):2346–2353, 07 2005.