Audio-Based Early Warning System of Sound Events on the Road for Improving the Safety of Hearing-Impaired People

Chia-Yi Chen, Pei-Yi Kuo, Yunh-Hsin Chiang, Jia-Yun Liang, Kai-Wen Liang, and Pao-Chi Chang

Department of Communication Engineering
National Central University
Taoyuan City, Taiwan
pcchang@ce.ncu.edu.tw

***Abstract*—This paper presents an audio-based early warning system which helps to protect the safety of people with hearing impairment. The work is implemented on Android platform with a human-machine interface APP rendering. The prediction model will identify the result and send it to the device interface in order to alert the user. The experimental results showed that the accuracy of the system is 66.4% for eight class classification, including vehicle approaching events and some environmental sound on the road.**

Keywords—Android application, warning, audio detection, machine learning

I. INTRODUCTION

In our lives, we feel and identify some information through our five senses. Hearing plays a vital role on the absorption and comprehension of information. Hence, when we lose our eyesight, hearing becomes a crucial way to identify the surrounding. To hearing-impaired people, losing the hearing system would cause enormous inconvenience and difficulty of their lives, which can make them feel more uneasy.

Nowadays, environmental sound recognition applications have become popular on mobile platforms.

Angelos and his colleagues presented the realization of real-time environmental sound recognition on Android operation system[1].

We hope that through the technology of machine learning, we can design an auxiliary system to improve the convenience of hearing-impaired people and make their lives more secure.

In this paper, we propose an early warning system for detecting vehicle approaching events and some environmental sound on the road. And we design this system as the app on the smart phone. This app is designed to be a cost-effective and feasible approach to detect environmental sound on the road using a smart phone with microphone. Our goal is to alert the user to some events on the road in a short period ahead. The alert should be triggered ahead of this 4-second period. When the alert is triggered, the mobile phone interface notification will jump out and the block of that event will twinkle to alert the user..

II. SYSTEM ARCHITECURE

In our system, we use CNN architecture to train our sound data, and classify them according to their features. Afterwards, we input our prediction model to Android application. The prediction model will identify the environmental sounds and send the result to the device interface in order to alert the user.

The flowchart of our system is shown in Fig1. There follows important blocks described. Audio input with 22050 Hz sampling rate and 16-bit floating-point resolution is applied in our system and then extract the audio input feature by MFCC as the audio input should be matched to our input of CNN. The short-time frame size is 8192 samples. The detection is performed in every single frame and every four seconds.

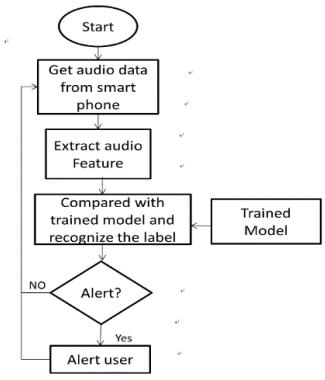

Fig. 1. System work flow of the proposed system

III. EXPERIMENTS AND FUNCTIONAL ASSESSMENT

We use the decisoin tree to classify the sounds detected to confirm whether the car is approaching or not. J48 decision tree are choosed after some experiments and performance evaluations among methods such as random tree , SPC, SPR and RMS. J48 is characterized by selecting the best judgment basis and the critical point of judgment under all conditions. As shown in the Fig.2 , it is judged that the best classification basis is that the SPC classification critical point is 71.4, and the correct rate is 0.96, which is not the highest, but it can judge whether the car approaches by only one feature, and improves the efficiency of the proposed system.

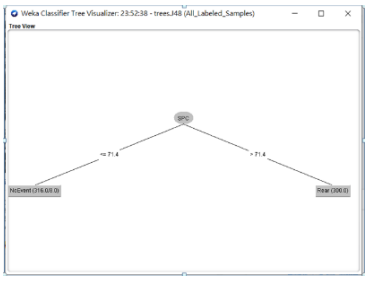

Fig. 2. Experiment on J48 decision tree

978-1-7281-2575-5/19/$31.00 ©2019 IEEE

For the sake of integration of the system, the time factor, compatibility of Android Studio and the consistency of the receiving frequency are all considered. Waveform characteristics captured by MFCC are used be the features because of the complexity of the process of decision tree.The accuracy could reach 80%, which is suitable for capturing single frequency, and actual identification result has almost right. Thus, we did not use Librosa's decision tree for feature extraction. Instead, we use MFCC for our feature extraction.

IV. FEATURE EXTRACTION

Feature extraction function generates a set of features that represent the characteristics of the sound signal. In this work, the analysis method we apply is "MFCC", which is commonly used by speech and environmental sound recognition research. [2]

The use of Mel-Frequency Cepstral Coefficients (MFCC) is a feature widely used in automatic speech and speaker recognition. The MFCC feature extraction consists of two steps: Meyer frequency analysis and cepstrum analysis. The Mel frequency cepstral coefficient MFCC considers human auditory characteristics; First, map the linear spectrum into the Mel nonlinear spectrum based on auditory perception and then convert it to the cepstrum. That is, the spectrum is obtained by passing a spectrum through a set of Mel filters. The formula is log X[k] = log (Mel-Spectrum). At this time, we perform cepstrum analysis on log X[k] and obtain it on the Mel spectrum. The cepstrum coefficient h[k] is called the Mel frequency cepstral coefficient. Usually, before calculating the MFCC, the spectrogram spectrum of the original sound signal and the MFCC are analyzed by pre-emphasis, framing and windowing, short-time FFT, and so on.

Mel-frequency cepstral coefficients (MFCCs) are derived from a type of cepstral representation of the audio clip. MFCC uses spectral centroid, and the spectral centroid is a feature on the frequency domain, which uses calculation of a weighted average to describe the centroid position of the spectral energy distribution in a frame. The formula is as follows:

$$\mathrm{SPC} = \frac{\sum_{k=0}^{N-1} kF[k]}{\sum_{k=0}^{N-1} F[k]}$$

Where, F[k] is the spectral power of the frequency bin.

V. CLASSIFICATION ALGORITHMS

Convolutional Neural Networks (CNNs) have basic architecture that consists of the convolutional layer, the pooling layer, also named as the subsampling layer, and the fully connected layer. As we know, CNNs have proven very effective in image classification and show promise for audio. Thus, we use CNN architecture to classify the soundtracks of a dataset with eight labels. We investigate varying the size of both training set and labels, finding that analogs of the CNNs used in image classification do well on our audio classification task, and larger training data and label sets help up to a point. A model using embeddings from these classifiers does much better than raw features classification task.

Therefore, we chose the CNN as our model, and our architecture consists of two convolutional layers, two max pooling layers, and one fully-connected layer. Our model is shown in Fig3.

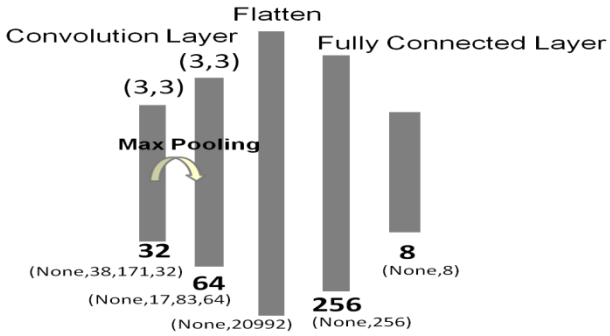

Fig. 3. CNN model of this work

In Fig. 3, after MFCC converts the sound to image, the 38x171 image is fed into the training model. Through a convolutional layer, the images are captured and feature mapping is generated. In order to reduce the complexity of the feature mapping, a max pooling layer is utilized. The model obtains the image by using the two convolution and two max pooling operations, where the size of the convolution kernel is 3× 3, respectively, and the Rectified Linear Units (ReLU) is added which means that the negative value caused by the convolution is turned off to become 0. The max pooling kernel size is 2 ×2. After one convolution and one max pooling, these features will go through the dropout layer which is used to reduce overfitting and the dropout rate is 0.3.Eventually, the model rearranges these high-dimension features into one-dimension feature vectors called flatten ,and then go into the fully-connected layer to classify eight labels and the softmax is added which means that , each component will be in the interval (0,1), and the components will add up to 1.

VI. DATA PREPROCESSING

First, we use the urbansound 8k dataset and connected sound with a microphone on smart phone, and each audio clip of the urbansound 8k is almost 4 seconds, but some are less than four seconds, so we fill it with 0.The sampling rate of these sounds are 22050 Hz, and thus, there are 88200 sample points in 4 seconds. Second, we use the Librosa kit to convert each audio file into a feature vector of the MFCC. Third, extract the fragments of MFCC and normalize them. Finally, Convert the MFCC feature vector to the CNN input format. Since the MFCC is two-dimensional data, to input the conv2D layer, those inputs are changed to 3D data. Because it is a cluster, it converts labels into categories.

VII. APP DEVELOPMENT

Before we start to produce the app, we convert our model to TFlite. After then, input them into the Android Studio to develop the android version of the app. Therefore, this app can receive the sound around us every four seconds, and the sound will be stored in the buffer afterwards. As we apply MFCC to extract our features of the sounds, we also do that in this app. Then, apply the MFCC to analyze the sound in the buffer and compare them with our model. This work is shown in Fig4. If the result show that it matches with the classification model,

the smart phone will vibrate and flash on the corresponding graphic of the event on app interface.

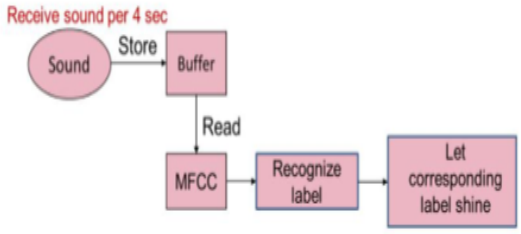

Fig. 4. The work on Android studio

VIII. EXPERIMENTAL RESULTS

We used urbansound 8K and connected sound with a microphone on smart phone for this research. We use 60% of our dataset as the training set, 20% of it as the validation set, and 20% of it as the testing set. The objective of the Sound Classification is to classify eight labels: Car-approaching, Car-horn, Children-playing, Dog-barking, Gun-shot, Construction, Siren, Engine-idling. To our research, the sounds of Car-appraoching are restricted to the speed with 40-50 km/hr and when the car distance from the users is about 30-40m, the smart phone will notify the users.

The evaluation is done with our dataset. Our test loss is 1.9904 and our model loss is shown in Fig5. Our test accuracy is 66.4% and the model accuracy is shown in Fig6.

Fig. 5. Fig5. Our model loss

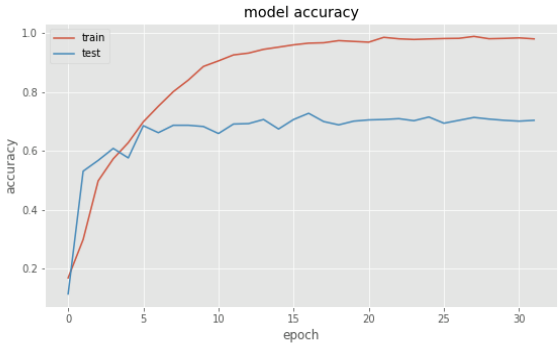

Fig. 6. Our model accuracy

In this confusion matrix, the accuracy of each classification is almost 65% and up. Also we can see the highest accuracy is 93.8%.The result is shown in Table I.

TABLE I. CONFUSION MATRIX FOR EACH CLASSIFICATION

REAL\PREDICTIONH	Car-approaching	Car-horn	Children-playing	Dog-barking	Construction	Engine-iding	Gun-shot	Siren
Car-approaching	82.0%	0.0%	2.0%	0.0%	6.0%	10.0%	0.0%	0.0%
Car-horn	0.0%	51.5%	18.2%	9.1%	21.2%	0.0%	0.0%	0.0%
Children-playing	12.0%	0.0%	71.0%	11.0%	2.0%	0.0%	1.0%	3.0%
Dog-barking	1.0%	2.2%	12.0%	67.4%	10.8%	0.0%	5.4%	1.1%
Construction	12.0%	0.0%	3.0%	8.0%	62.0%	3.0%	7.0%	5.0%
Engine-iding	15.0%	0.0%	0.0%	0.0%	10.7%	73.1%	1.1%	0.0%
Gun-shot	0.0%	0.0%	0.0%	6.3%	0.0%	0.0%	93.8%	0.0%
Siren	12.5%	0.0%	0.0%	0.0%	18.8%	3.1%	18.8%	55.4%

Our App rendering is shown in Fig 4. In this figure, the recognition result of this app is Car-approaching; therefore, the graphic of car-approaching will twinkle.

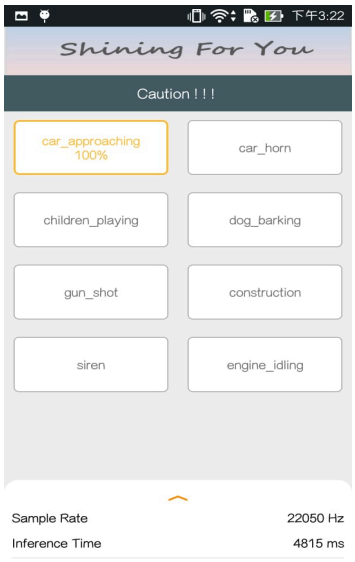

Fig. 7. App rendering

At last, we note that the test accuracy is up to 66.4%. The experimental results show that the accuracy is not really high, but the actual identification result on the app is very correct.

The most important problem for us is the accuracy is not high, but this problem can be solved by expanding the dataset with more diverse ambient noises

At last, we note that the test accuracy is up to 66.4%. The evaluation results showed that the proposed system utilizing devices is feasible for doing the task of detecting the dangerous sound on the road and giving early-warning. The experimental results show that the accuracy is not really high, but the actual identification result on the app is very correct.

IX. Conclusion and Future Work

This work indicates that the CNN is effective for environment sounds classification tasks by appropriate parameter settings and feature sets. Finally, identification results on the app is with certain level of accurracy and therefore can help to ensure the safety of people in need.

The objective of this work is aimed to promote this app to the hearing-impaired people to easy their difficulties when they are walking on the road. Besides, the app can be combined with the bracelet to enhance the users' convenience. In our future works, more environmental sound events will be taken into consideration for achieving higher accurracy and providing more practical applications.

X. DEMO LINK

DEMO Video: https://youtu.be/y-JDc_Zr8CA

Acknowledgment

This work was supported in part by Ministry of Science and Technology under grant no. MOST 108-2634-F-008-004.

References

[1] A. Pillos, K. Alghamidi, N. Alzamel, V. Pavlov, and S. Machanavajhala,"A real-time environmental sound recognition system for the Android OS, " *in Proceedings of Detection and Classification of Acoustic Scenes and Events,* September 2016, Budapest.

[2] W. Zunjin and C. Zhigang, "Improved MFCC-based feature for robust speaker identification," *Tsinghua Science and Technology*, Vol. 10, Issue. 2, pp. 158-161, April 2005