Smartphone-based Real-time Classification of Noise Signals Using Subband Features and Random Forest Classifier

Fatemeh Saki, Abhishek Sehgal, Issa Panahi, Nasser Kehtarnavaz

Department of Electrical Engineering, University of Texas at Dallas, USA

ABSTRACT

This paper presents the real-time implementation and field testing of an app running on smartphones for classifying noise signals involving subband features and a random forest classifier. This app is compared to a previously developed app utilizing mel-frequency cepstral coefficients features and a Gaussian mixture model classifier. The real-time implementation has been carried out on both the Android and iOS smartphones. The field testing results indicate the superiority of this newly developed app over the previously developed app in terms of classification rates.

Index Terms**—**Smartphone implementation, real-time background noise classification; noise classification for hearing devices; band-periodicity and band-entropy features; random forest classifier

1. INTRODUCTION

The problem of environmental background noise classification has been previously examined in many papers for various applications. Some example applications include classifying environmental sound signals in robotics [1], in smart homes for elderly people [2], and in automatic tagging of sound files [3]. In addition, noise classification has been utilized as part of speech enhancement or noise suppression pipelines for hearing aid and cochlear implant devices [4-6], where the speech enhancement parameters are adjusted depending on the environmental background noise.

A typical environmental background noise classification algorithm consists of two major components: a feature extractor and a classifier. Signal features which have been previously considered for noise classification are many. The major ones include: mel-frequency cepstrum coefficients (MFCC), matching pursuit [7], zero crossing rate, centroid and roll-off point [8], spectral centroid, spectral spread, spectral flatness, spectral flux, change chirp rate spectrum, Hilbert envelope, local energy and discrete curvelet transform [9], harmonic ratio, upper limit of harmonicity, and audio fundamental frequency [10]. A combination of these features is often used to achieve a high classification rate [11].

As far as classifiers are concerned, Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), Support Vector Machine (SVM), neural networks, deep belief network, k-nearest neighbor classifiers have been utilized [7-12] for noise classification.

As discussed in [13], one issue that has not been adequately addressed is the real-time computation aspect of such features and classifiers. In [13], band-periodicity and band-entropy features and Random Forest (RF) classifier were used to achieve background noise classification for cochlear implants applications. It was shown that computationally efficient subband features along with an RF classifier (subband+RF) outperformed a previously developed MFCC and GMM (MFCC+GMM) approach [4, 14].

In this paper, a real-time implementation of the subband+RF noise classification is reported on both Android and iOS smartphones together with a performance comparison with the MFCC+GMM noise classification.

The rest of the paper is organized as follows. An overview of our previously developed background noise classifier using subband features and random forest classifier is provided in Section 2. The steps taken towards the smartphone implementation of this classification approach are then reported in Section 3. Section 4 includes the results corresponding to both the offline analysis as well as the real-time field testing. Finally, the conclusion is stated in Section 5.

2. OVERVIEW OF PREVIOUSLY DEVELOPED BACKGROUND NOISE CLASSIFICATION

Although MFCC features have been extensively used in the literature for noise signal classification, it is found that they have limitations in realistic noise environments. That is why additional features are often used in addition to MFCC features to gain high classification rates. However, a practical problem that arises as a result of utilizing many features is the computational complexity associated with running a classification signal processing pipeline in real-time on handheld devices, in particular on smartphones. In [13], subband features and a random forest (RF) classifier were used as an alternative to MFCC features and a GMM classifier that had been shown to be computationally suitable to achieve real-time throughputs compared to many other approaches [4].

As discussed in [13], subband features consist of band-periodicity and band-entropy features. Band-periodicity features capture the periodicity aspect of noise signals whose characteristics remain more or less stationary over time; whereas band-entropy features capture the non-stationary characteristics of noise signals. Band-periodicity and band-entropy features are computed from signal segments of duration S seconds. Each segment is divided into M overlapping frames of length N, with the m^{th} frame specified by $F_m := \{x_n \mid x_n \in R, n = 1,..N\}$, where x_n represents the n^{th} sample in the frame. Assuming the sampling rate of Fs, the frequency range $[0, Fs/2]$ is divided into B non-overlapping subbands. The cross-correlation between every two consecutive frames, that is F_m and F_{m-1} in each band, is computed and the peak value of the cross-correlation is denoted by $P_{b,m}$, where b and m represent the band and frame index, respectively. The band-periodicity feature in band b is then defined as [15]:

$$BP_b = \frac{1}{M}\sum_{m=1}^{M} P_{b,m} \quad ,b = 1,\ldots,B \tag{1}$$

where M is the total number of frames over duration S.

The band-entropy feature in each band over duration S is defined as:

$$BE_b = \frac{1}{M}\sum_{m=1}^{M} H_{b,m} \quad ,b = 1,\ldots,B \tag{2}$$

where $H_{b,m}$ represents the entropy of the m^{th} frame in band b. Considering B bands, a feature vector of $2 \times B$ components is thus used to capture the signal characteristics over a duration of S seconds. The extracted feature vector is then fed into an RF classifier to find a matched class to the incoming signal frames. It is worth noting that band-periodicity and band-entropy features unlike the MFCC features are not sensitive to the sound loudness, thus they do not require any preprocessing normalization as part of their extraction.

An RF classifier [16] is an ensemble of T number of classification trees. Each tree is trained independently from other trees using a randomly selected (with replacement) subset of a training set. At the start of the training, or at the root node, the entropy is high since all training samples from all the classes are used at this stage. Then, the tree is built in such a way that the entropy is decreased as layers are added until the tree reaches its leaves with the lowest entropy allowing classification of all the training data.

MFCC features are widely used in speech processing. MFCC features attempt to capture the spectral information corresponding to the human auditory response. MFCC features are computed by grouping the short time Fourier transform coefficients of a frame into a set of L coefficients based on L mel-scale non-overlapping filters or filterbank, followed by a discrete cosine transform for decorrelation purposes. Normally, the first 13 coefficients are used to serve as MFCC features. Likewise, the Gaussian Mixture Model (GMM) classifier is extensively used for signal classification. In this classifier, the data or samples corresponding to a class is modeled by a mixture of several Gaussians in the feature space whose parameters are estimated using the iterative expectation-maximization algorithm.

3. REAL-TIME IMPLEMENTATION ON SMARTPHONES

The subband feature extraction and the random forest classifier were coded in C which were then integrated into the Android and iOS smartphones using the guidelines provided in the book "Smartphone-Based Real-Time Digital Signal Processing" [17]. The shell provided in the book was used for the microphone interfacing and the GUI. The software tools that were used to achieve the smartphone implementation are noted below: For Android smartphones, the IDE (Integrated Development Environment) of Android Studio was used together with the Android SDK (Software Development Kit) [18]. To support C codes within Android smartphones, the Android NDK (Native Development Kit) [19] was used. For the iOS implementation, the IDE of Xcode [20] was used. C codes were interfaced with Objective-C of iOS by importing the header file. Interested readers are referred to the above book for the details of embedding and running C codes within the Android and iOS environments.

For feature extraction, signals were captured in frames of length 25msec with half a frame overlap, i.e. 12.5msec overlap. MFCC features were extracted from every frame and the extracted feature vector was fed into a GMM classifier. The implementation was done using 13 MFCC features with a mel-filter bank of 40 filters together with two Gaussians in the mixture model per class.

Band-periodicity and band-entropy features were computed per signal segment of duration S =1 second. Each incoming frame was divided into B = 8 non-overlapping bands of width 1kHz in the frequency domain. Thus, a feature vector of 16 subband features (8 band-periodicity and 8 band-entropy features) was obtained over every 1 second which was then fed into an RF classifier consisting of 20 trees.

Screen snapshots of the app on an Android smartphone are provided in Fig. 1. The user has the option to perform online classification of sound signals that are captured by the smartphone microphone or to save captured sound signals for later examination. The app allows adjusting the sampling rate, frame length, frame overlap amount, and decision buffer length (in frame unit) for majority voting classification.

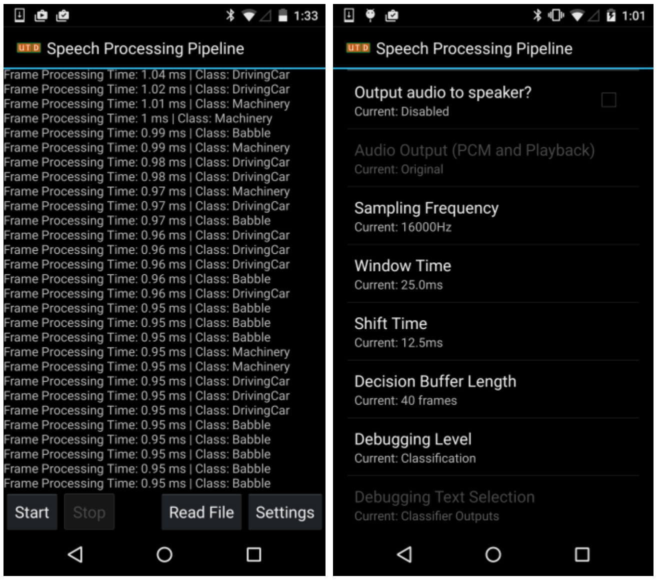

Fig.1. Snapshots of the developed noise classification smartphone app

4. EXPERIMENTAL RESULTS AND COMPARISON

The developed classification app was examined by considering three widely encountered noise types of babble, car driving and machinery. The examination was done in offline and field testing manners which are explained in more details in the subsection that follow.

3.1. Dataset

As part of the app development, a comprehensive dataset of 120 sound files for the three noise types of babble, car driving and machinery were put together which is accessible for public use at the website noted in [21]. The machinery class contains noise signals of home appliances. For each noise type, 40 sound files of duration 30 seconds were collected at different times at a sampling frequency of 16kHz using a Nexus 5 smartphone. For both the data collection and the real-time operation of the classifier, only one microphone of the smartphone was used.

TABLE I. OFFLINE EVALUATION OF SUBBAND+RF-AVERGED OVER 100 DIFFERENT TRAINING AND TESTING

Detected class Actual class	Babble (%)	Car Driving (%)	Machinery (%)
Babble	**98.9**	0.1	1
Car Driving	0	**99.7**	0.3
Machinery	3.9	0.1	**96**

TABLEI II. OFFLINE EVALUATION OF MFCC + GMM–AVERGED OVER 100 DIFFERENT TRAINING AND TESTING

Detected class Actual class	Babble (%)	Car Driving (%)	Machinery (%)
Babble	**86.5**	11.5	2
Car Driving	3.3	**95.6**	1.1
Machinery	2.1	0.9	**97**

3.1. Offline evaluation and comparison

The MFCC+GMM and subband+RF classification approaches were evaluated in an offline manner first as follows. The dataset was randomly divided into a training (80%) and a testing set (20%) with no overlap between them. This procedure was repeated 100 times. Each time the classifiers were trained using a different training and testing sets and the averaged results are indicated in Tables I and II. As can be noted from these tables, the subband+RF approach provided a higher overall classification rate compared to the MFCC+GMM approach, in particular for babble type of noise. This is attributed to the discriminatory power of subband features as compared to MFCC features as evident by computing the Fisher discriminant measure [22]:

$$J = trace(S_w^{-1} S_b) \qquad (3)$$

where S_w denotes the within-class scatter matrix and S_b the between-class scatter matrix. Higher J values indicate that samples in the multi-dimensional feature space are more separated. When using the subband features, this measure was found to be J=2350, while when using the MFCC features, this measure was found to be J=38, indicating a high level of spread or overlap between the babble class and the other two classes in the MFCC feature space.

The next experimentation involved running the classifiers in real-time on actual smartphones in the field which is mentioned next.

3.1. Actual field testing and comparison

The developed classifier apps were run on smartphone platforms in the three noise environments to evaluate their

TABLE III. FIELD TESTING OF SUBBAND+RF

Detected class Actual class	Babble (%)	Car Driving (%)	Machinery (%)
Babble	**80.4**	0	19.6
Car Driving	0.4	**99**	0.6
Machinery	0	0	**100**

TABLEI IV. FIELD TESTING OF MFCC + GMM

Detected class Actual class	Babble (%)	Car Driving (%)	Machinery (%)
Babble	**47.4**	1.1	51.4
Car Driving	1	**99**	0
Machinery	0	0	**100**

TABLE V. TREATMENT OF OTHER NOISE ENVIRONMENTS, SUBBAND+RF VS. MFCC+GMM

	Subband + RF			MFCC + GMM		
Matched class / Other classes	**Babble** (%)	**Car Driving** (%)	**Machinery** (%)	**Babble** (%)	**Car Driving** (%)	**Machinery** (%)
Crowded Restaurant	**89.5**	3.4	7.1	1	**1**	98
Street	18.5	**13.2**	**68.3**	0	**17.5**	**82.5**
Loud Indoor AC	0	16.3	**83.7**	**98.9**	**1.1**	**0**
Washer	7.8	0.8	**91.4**	**51.3**	**8.7**	**40**
Dryer	0	8.3	**91.7**	**50.5**	**49.5**	**0**
Vacuum	0	0	**100**	0	**0**	**100**

actual performance in the field. The outcome of this experimentation appears in Tables III and IV. Here it is worth pointing out that since microphones on different smartphones have different frequency responses, the data collection and thus training were repeated for each device to remove any frequency dependency on the device microphone. As noted form these tables, the MFCC+ GMM app and the subband+RF app performed similarly in the noise environments of car driving and machinery. However, in the babble environment, the subband+RF app by far outperformed the MFCC+GMM app.

The reason for the poor performance of the MFCC+GMM app in the field was traced back to the sensitivity of MFCC features versus subband features. MFCC features were found to be quite sensitive to various variations that occur in babble type of noise environments in the field whereas the subband features were found to be much less sensitive to various variations that occur in babble type of noise environments. As a percentage, it was found that MFCC features exhibited a large variation of 173% in the field testing performed whereas subband features only exhibited a variation of 2% when encountered with variations of babble type of noise for which the classifiers had not been trained.

Another study was conducted to assess the behavior of the apps in the presence of other noise types for which no training had been done. The outcome of this study appears in Table V. As seen from this table, these other noise types got matched to the closest class with similar sound characteristics when using the subband+RF app, while the MFCC +GMM app could not distinguish between the babble and machinery noise types. For example, the crowded restaurant with music in the background, which was not part of the training data, was classified as machinery noise type and the loud indoor air conditioning (AC) noise, which was not part of the training data, was classified as babble noise type.

The average processing times per 25msec frames with a frame overlap of 12.5msec for the subband+RF classification on an Android platform (Nexus 5) and on an iOS platform (iPad Mini 2) are shown in Table VI. This time incorporates the i/o delay time associated with these devices. To achieve real-time throughputs, the total processing time needed to remain below 12.5msec for no frame to get skipped. When using the Vector Floating-Point (VFP) coprocessor hardware on the smartphones, the timings naturally improved. The table lists the timings with and without using VFP. In all the cases, real-time throughputs were achieved. A video clip of the subband+RF classification app can be viewed at the link stated in [23].

TABLE VI. AVERGED FRAME PROCESSING TIMES OF SUBBAND+RF MODEL (25 MSEC FRAMES WITH HALF FRAME OVERLAP AT 16 KHZ SAMPLING FREQUENCY)

Frame processing time in msec	Subband feature extraction + RF classifier
Android without using VFP	**3.1ms**
Android with using VFP	**1.5ms**
iOs without using VFP	**3.4ms**
iOS with using VFP	**3.1ms**

5. CONCLUSION

This paper has provided an app for carrying out background noise classification in real-time on smartphone platforms. Two classification approaches having low computational complexity which allowed them to be run in real-time on smartphone platforms, namely MFCC+GMM and subband+RF, were implemented and compared in the field. The extensive experimentations carried out have shown that the subband+RF approach provides both real-time throughputs and high classification performance for the three commonly encountered noise environments of babble, car driving and machinery.

ACKNOWLEDGMENT

This work was supported in part by the National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health under the award number 5R56DC014020-02. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

6. REFERENCES

[1] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Special Issue Socially Interactive Robots, Robot., Autonomous Syst*., vol. 42, no. 3–4, pp. 271–281, 2003.

[2] J. Wang, H. Lee, J. Wang, and C. Lin, "Robust environmental sound recognition for home automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 25–31, 2008.

[3] S. Duan, J. Zhang, P. Roe, and M. Towsey, "A survey of tagging techniques for music, speech and environmental sound," *Artificial Intelligence Review*, vol. 42, pp. 637–661, 2012.

[4] V. Gopalakrishna, N. Kehtarnavaz, T. Mirzahasanloo, and P. Loizou, "Real-time automatic tuning of noise suppression algorithms for cochlear implant applications," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 1691-1700, 2012.

[5] P. Loizou, A. Lobo, and Y. Hu, "Subspace algorithms for noise reduction in cochlear implants," *J. Acoust. Soc. Am.* 118, pp. 2791-2793, 2005.

[6] F. Saki, T. Mirzahasanloo, and N. Kehtarnavaz, "A multi-band environment-adaptive approach to noise suppression for cochlear implants," *Proc. of IEEE International Conference Engineering in Medicine and Biology*, Chicago, August 2014.

[7] S. Chu, S. Narayanan, C. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Transactions on Audio, Speech and Language Processing*, vol.17 pp. 1142–1158, 2009.

[8] H. Lozano, I. Hernaez, A. Picon, J. Camarena, and E. Navas, "Audio classification techniques in home environments for elderly/dependant people," *Proc. of the 12th Int. Conf. on Computers Helping People with Special Needs*: *Part I, ICCHP'10*, Springer-Verlag, pp. 320–323, 2010.

[9] H. Byeong-jun, H. Eenjun, "Environmental sound classification based on feature collaboration," *Proc. of IEEE Int. Conf. on Multimedia and Expo*, New York, pp. 542–545, 2009.

[10] S. Ntalampiras, I. Potamitis, N. Fakotakis, "Automatic recognition of urban environmental sounds events," *Proc. of Int. Association for Pattern Recognition Workshop on Cognitive Information Processing*, pp. 110–113, 2008.

[11] C. Kraetzer, A. Oermann, J. Dittmann, and A. Lang, "Digital audio forensics: a first practical evaluation on microphone and environment classification," *Proc. of the 9th Workshop on Multimedia and Security*, Dallas, TX, 2007.

[12] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Inform. Sciences*, vol. 243, pp. 57-74, 2013.

[13] F. Saki, N. Kehtarnavaz, "Background noise classification using random forest tree classifier for cochlear implant applications," *Proc. of IEEE ICASSP,* Italy, May 2014.

[14] S. Parris, M. Torlak, and N. Kehtarnavaz, "Real-time implementation of cochlear implant speech processing pipeline on smartphones", *Proc. IEEE International Conference Engineering in Medicine and Biology (EMBC)*, Chicago, August 2014.

[15] L. Lu and H. Zhang "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing,* vol. 10, pp. 504-516, 2002.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[17] N. Kehtarnavaz, S. Parris, A. Sehgal, *Smartphone-Based Real-Time Digital Signal Processing*, Morgan and Claypool Publishers, 2015.

[18] http://developer.android.com/sdk/index.html

[19] http://developer.android.com/tools/sdk/ndk/index.html

[20] https://developer.apple.com/xcode/

[21] http://www.utdallas.edu/~kehtar/NoiseData.rar

[22] R. Duda, and P. Hart, *Pattern Classification*, New York: Wiley, 2000.

[23] http://www.utdallas.edu/~kehtar/NoiseClassifierDemo.mp4