Acoustic Event Detection in Multichannel Audio Using Gated Recurrent Neural Networks with High-Resolution Spectral Features

Hyoung-Gook Kim and Jin Young Kim

Recently, deep recurrent neural networks have achieved great success in various machine learning tasks, and have also been applied for sound event detection. The detection of temporally overlapping sound events in realistic environments is much more challenging than in monophonic detection problems. In this paper, we present an approach to improve the accuracy of polyphonic sound event detection in multichannel audio based on gated recurrent neural networks in combination with auditory spectral features. In the proposed method, human hearing perception-based spatial and spectral-domain noise-reduced harmonic features are extracted from multichannel audio and used as high-resolution spectral inputs to train gated recurrent neural networks. This provides a fast and stable convergence rate compared to long short-term memory recurrent neural networks. Our evaluation reveals that the proposed method outperforms the conventional approaches.

Keywords: Acoustic event detection, Deep recurrent neural networks, Gated recurrent neural network, Multichannel audio.

I. Introduction

Over the last few years, there has been an increased interest in studies related to sound event detection (SED). Sound events include important information that can be used to describe and understand human and social activities. Therefore, the classification and detection of these sounds, namely acoustic events, can be utilized in a variety of application areas, including multimedia content analysis [1], context-aware devices [2], unobtrusive monitoring in healthcare [3], and audio-based surveillance [4].

SED aims to label temporal regions within the audio, within which a specific event class is active, by estimating the start and end of each event in a continuous acoustic signal. The SED is divided into two scenarios: monophonic and polyphonic. Monophonic sound event detection is a restricted condition in which the number of simultaneous active events is one. By contrast, in polyphonic sound event detection, several sound events may happen simultaneously, or multiple sound events are overlapped. In real-life everyday situations, most of the sounds that reach our ears tend to stem from a multitude of sound sources. The overlapping patterns of acoustic events are unknown, and the features extracted from the mixture do not match with features calculated from sounds in isolation. Therefore, the polyphonic case is much more challenging (formulated as a multilabel classification problem) than a monophonic detection problem.

One of the most popular approaches to polyphonic SED is using Gaussian mixture models (GMMs) or hidden Markov models (HMMs) as classifiers and using mel-frequency cepstral coefficients (MFCCs) as features [5]. In recent years, deep neural networks (DNNs) have achieved tremendous success in various machine learning tasks, and

Manuscript received Mar. 2, 2017; revised May 15, 2017; accepted June 7, 2017.
Hyoung-Gook Kim (hkim@kw.ac.kr) is with the Department of Electronics Convergence Engineering, Kwangwoon University, Seoul, Rep. of Korea.
Jin Young Kim (corresponding author, beyondi@jnu.ac.kr) is with the Department of Electronics and Computer Engineering, Chonnam National University, Gwangju, Rep. of Korea.

have also been applied to sound event detection tasks [6], [7]. Further improvements over DNNs have been achieved with alternative types of neural network architectures, including convolutional neural networks (CNNs) [8], recurrent neural networks (RNNs) [9], and long short-term memory recurrent neural networks (LSTM-RNNs) [10]. LSTM-RNN works well on sequence-based tasks with long-term dependencies. However, it has a relatively high model complexity, and parameter tuning for LSTM-RNN is not always simple. Compared to LSTMs, gated recurrent neural networks (GRNNs) [11] are temporal deep neural networks with reduced computational complexity. More recently, a multilabel GRNN was successfully applied to acoustic event classification in [12].

State-of-the-art polyphonic SED systems have been using a single channel of audio. On the other hand, humans have only two ears and yet are capable of identifying each of the overlapping sound events by isolating and localizing the source spatially. Reflecting the auditory perception principle, Adavanne and others [13] successfully used spatial and harmonic features from stereo-channel audio in combination with LSTM-RNN and showed considerable improvement over just using monochannel audio. This motivates us to use multichannel audio features in combination with GRNNs for automatic polyphonic sound event detection tasks in real-life audio.

In this paper, we propose acoustic event detection in multichannel audio using GRNNs with high-resolution spectral features. The proposed method has the following three advantages: (1) Just like humans use their two ears (two channels) to localize and recognize the sound events around them, we also potentially trained machines to learn sound events from multichannel audio. Three sets of features (pitch range, time difference of arrival, and spectrogram patches) are extracted from the stereo audio. (2) Instead of low-resolution features such as MFCCs or Mel band energies, high-resolution spectral features such as a noise-reduced spectrogram patch are used to preserve detail that is critical to deal with overlapping signals. To alleviate the effect of the noises, a pitch-based noise reduction approach is performed for feature enhancement. (3) To efficiently classify overlapping sound events, spatial and spectral-domain features extracted from multichannel audio are used as high-resolution spectral inputs to train GRNNs. This provides a fast and stable convergence rate compared to LSTM-RNN.

The outline of this paper is as follows. In Section II, the proposed acoustic event detection method is explained. Section III presents the experimental results. Section IV provides conclusions and suggestions for further study.

II. Proposed Acoustic Event Detection

The main objective of the proposed acoustic event detection system is to temporally locate the sound events in a recording collected from a realistic auditory scene and give each event a label from a set of possible labels. Figure 1 shows the process flow of the proposed sound event detection system, which is composed of a training stage and a test stage.

First, in the training stage, data augmentation is performed to generate artificial sound event scenes, which are used to train the classifier to deal with the data sparsity problem. In this paper, event scenes are corrupted only by white, blue, and pink noises without manipulation by time stretching, pitch shifting, and dynamic range compression for data augmentation. The sound event signals and noise sources were mixed by digital addition and scaling to create three sets of noisy sound event signal recordings. The mixed sound event database was created at multiple SNRs: 0 dB, 5 dB, 10 dB, 15 dB, and 20 dB.

Second, we compute acoustic features in each frame for the left and right channels. Third, the extracted acoustic features from multichannel audio are concatenated and used for training the GRNN. In the test stage, we perform acoustic event classification by inputting the high-dimensional acoustic features from multichannel audio into the trained GRNNs. Similar to [14], we post-process the events by detecting contiguous regions neglecting events smaller than 0.1 s, as well as ignoring consecutive events with a gap smaller than 0.1 s.

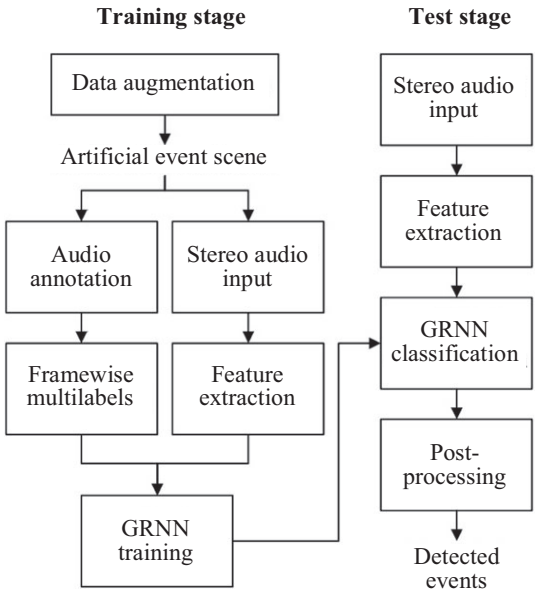

Fig. 1. Framework of training and testing procedure for proposed system.

1. High-Resolution Spectral Feature Extraction from Multichannel Audio

Humans have only two ears and yet are capable of analyzing an auditory scene in multiple dimensions. By increasing the number of audio channels, the spatial properties of a sound scene can be captured and reproduced more accurately. Thus, we extract three sets of features: noise-reduced spectrogram, pitch range, and time difference of arrival (TDOA) [13]–[15] from the stereo-channel of audio, as shown in Fig. 2. All features are extracted at a hop length of 20 ms to achieve consistency across features.

A. Pitch Range Extraction

Most acoustic sounds can be classified according to how rapidly they change over time. Pitch reflects the time-varying features of sound events and can be determined only in sounds that have a frequency that is clear and stable enough to be distinguished from noise. In an acoustic environment, human listeners are able to identify different sounds using pitch cues, and can make efficient use of pitch to acoustically separate each of the mixtures in an overlapping sound event.

Uzkent and others [16] introduced a pitch range-based feature set in order to improve the accuracy rates of non-speech environmental sound classification. Garner and others [17] used a linear dynamical system and associated Kalman smoother to generate continuous pitch estimates given discontinuous f0 observations. In using the Kalman smoother, some heuristic aspects of the pitch extractor were rendered moot, thus enabling simplification.

Using a combination of the above two methods, we extract two features based on the pitch range as follows:

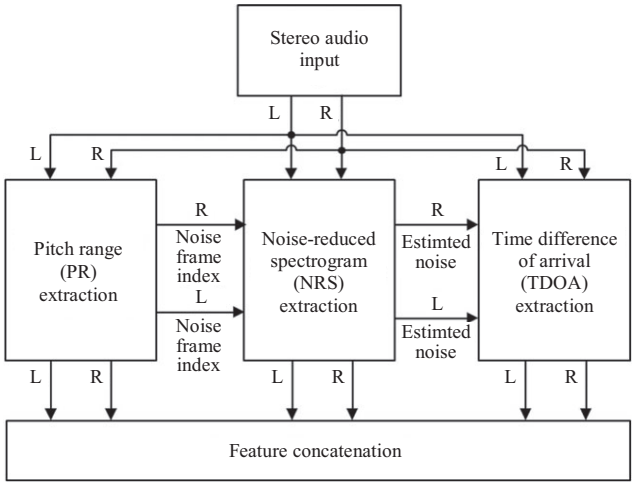

Fig. 2. Spectral feature extraction from stereo-channel audio.

First, pitch values are calculated using the short-time autocorrelation function (STAC). The STAC generates the instantaneous pitch for the input signal, which will invariably contain some tracking errors.

Second, the Kalman smoother [17] is applied to pitch values to smooth the pitch contour. This smoothing is needed because the pitch provided by the autocorrelation function is not continuous enough (mainly at low SNRs). The resulting contour is robust to the wrong choice of peak in the autocorrelation.

Our feature extraction focuses on the range of the pitch of the sound instead of the pitch itself, since environmental sounds may change their acoustical characteristics in time and do not have a constant pitch value, but a range of values. Then, we compute two features using the pitch range: the ratio of the maximum to the minimum of the pitch range (RPD) and the ratio of the standard deviation to the mean value of the pitch range (RPM).

The RPD and RPM are calculated as follows:

$$RPD = \max\{P(i)\}/\min\{P(i)\}, 1<i<M, \quad (1)$$

$$RPM = std\{P(i)\}/P_{\text{mean}}, \quad (2)$$

using

$$P_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{M} P(i), \quad (3)$$

and

$$std\{P(i)\} = \left(\frac{1}{N-1}\sum_{i=1}^{M}\left(P(i)-P_{\text{mean}}\right)\right)^{1/2}, \quad (4)$$

where P represents the pitch and M is the total number of short windows for any sound event.

B. Noise Reduced Spectrogram Extraction

In real-life recordings, various noises exist and make it difficult to detect sound events correctly. To alleviate the effect of the noises, noise reduction is performed for feature enhancement.

Each input sound event signal $x(n)$ is transformed into the complex spectrogram $X(\omega, l)$ of the ωth frequency at time frame l using a short-time discrete Fourier transform. The noise is estimated by extrapolating the noise from frames where the sound event signal is believed to be absent. These frames are detected by the pitch detector, which is applied to pitch range-based feature extraction.

Next, the noise estimation is revised in sound event frames using a harmonic tunneling technique. Given a

https://doi.org/10.4218/etrij.17.0117.0157

sound event frame contaminated with some background noise, it is possible to estimate the magnitude shape of the noise spectrum using samples of the noisy spectrum in the gaps between the harmonics. If the input frame has been classified as a sound event frame by the pitch detector, spectral peak detection as harmonic analysis is performed on the spectral envelope. The noise estimation technique based on harmonic tunneling is well described in [18].

The estimated noise is subtracted from the input magnitude spectrogram for noise reduction. Finally, the noise-reduced magnitude spectrogram is normalized by a logarithmic operation.

State-of-the-art feature extraction techniques such as MFCC or Mel-filter bank features in acoustic event detection tend to obtain broad characterizations but end up reducing detail that is critical to deal with overlapping signals. Therefore, we use log magnitude spectrograms as useful features to detect overlapping sounds [19].

C. TDOA Extraction

The human ear hears sounds in stereo, and the brain uses the subtle differences in sounds entering the left and right ears to locate sounds in the environment. To locate the source of sound within the environment, we need to calculate the TDOA of the wave front at the two transceivers, which in biological terms is the equivalent of the interaural time difference cue used in the auditory cortex of the mammalian brain. The TDOA can be estimated using the generalized cross-correlation with phase-based weighting (GCC-PHAT) owing to our need for computational efficiency.

First, we compute the inverse Fourier transform of the signal cross-power spectrum scaled by a weighting function. One instance of a GCC weighting function is the cross-power spectrum phase, also called the phase transform (PHAT) [15]. This weighting places equal importance on each frequency band. By dividing the spectrum by its magnitude, it deemphasizes portions of the spectrum that are suspected to be corrupt. This process results in a constant energy concentrated over all frequencies such that the correct TDOA can be found by high coherence between the two signals. GCC-PHAT is computed as

$$R_{12}(\tau, l) = \sum_{\omega=0}^{N-1} \frac{w^2(\omega, l) X_1(\omega, l) X_2^*(\omega, l)}{|X_1(\omega, l)||X_2^*(\omega, l)|} e^{j2\pi\omega\tau/N}, \quad (5)$$

where $X(\omega, l)$ are the FFT coefficients of the ωth frequency at time frame l, and w is a weighting function for the spectrum. PHAT weighting is computed as follows:

$$w(\omega, l) = \begin{cases} 1, & \text{if } Y(\omega, l) \le Y_N(\omega, l), \\ \left(\frac{Y(\omega,l)}{Y_N(\omega,l)}\right)^{\alpha}, & \text{otherwise.} \end{cases} \quad (6)$$

The weighting function gives more weight to regions in the spectrum where the local signal-to-noise ratio (SNR) is the highest. Let $Y(\omega, l)$ be the mean power spectral density for all the microphones at a given time, and $Y_N(\omega, l)$ be a noise estimate based on the time average of previous $Y(\omega, l)$.

The TDOA is estimated as $\Delta\tau_{ij} = \arg\max R_{ij}(\tau)$ because $R_{ij}(\tau)$ is expected to have a global maximum at the location of correlation peak magnitude $\tau = \Delta\tau_{ij}$.

2. Multilabel Recurrent Neural Networks Using Gated Recurrent Units

A recurrent neural network (RNN) is a computational neural network that has feedback connections, so it works efficiently and flexibly with time-series signals like audio. In fact, owing to the exploding and vanishing gradient problem, a simple RNN is not easy to train and is unable to deal with long ranges. To avoid these problems, two variants of RNN have been proposed using a gating approach: long short-term memory (LSTM) and gated recurrent unit (GRU). GRU is a relatively simplified architecture alternative to LSTM [20]. It is easy to notice similarities between LSTM unit and the GRU from Fig. 3. Therefore, our focus is mainly on the GRU. For polyphonic sound event detection, in this paper we use a multilabel RNN with a GRU (GRU-RNN).

While the LSTM uses three gates to control the information flow of the internal cell unit, the GRU only uses two gates: an update gate z and a reset gate r. The update gate controls the information that flows into memory, while the reset gate controls the information that flows out of memory. Similar to the LSTM unit, the GRU

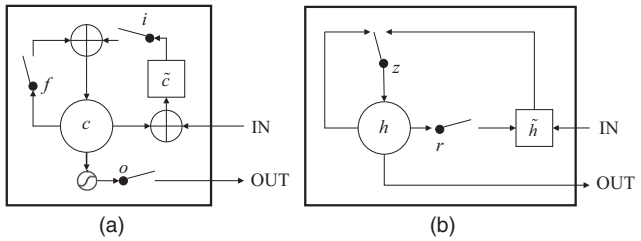

Fig. 3. (a) LSTM unit and (b) GRU. (a) i, f and o are the input, forget, and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation.

unit has gating units that modulate the flow of information inside the unit. However, it does this without having a separate memory cell. Mathematically, the process is defined by (7) to (10).

$$z_t = \sigma\left(W_z[h_{t-1}, x_t] + b_z\right), \tag{7}$$

$$r_t = \sigma\left(W_r[h_{t-1}, x_t] + b_r\right), \tag{8}$$

$$\tilde{h}_t = \tanh\left(W_h[r_t \circ h_{t-1}, x_t] + b_h\right), \tag{9}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t, \tag{10}$$

where x_t, h_t, $\tilde{h}_t$, and b denote the current input, activation, candidate activation, and bias vector, respectively. W, $\sigma(a)$, and $\circ$ are the input-to-hidden weight matrix, element-wise logistic sigmoid function, and element-wise multiplication, respectively.

Equation (7) describes the update gate, (8) represents the reset gate, (9) calculates the candidate activation, and (10) shows how the output h_t is calculated [21]. The activation of the GRU at time t is a linear interpolation between the previous activation and the candidate activation (That is, the memory is handled by a simple linear interpolation between a hidden-like state in the previous time step and an RNN-like component representing a current time step).

The reset mechanism helps the GRU to use model capacity efficiently by allowing it to reset whenever the detected feature is no longer necessary. The update mechanism helps the GRU to capture long-term dependencies. Whenever a previously detected feature or the memory content is considered to be important for later use, the update gate will be closed to carry the current memory content across multiple time steps. The data flow and operations are illustrated in Fig. 4.

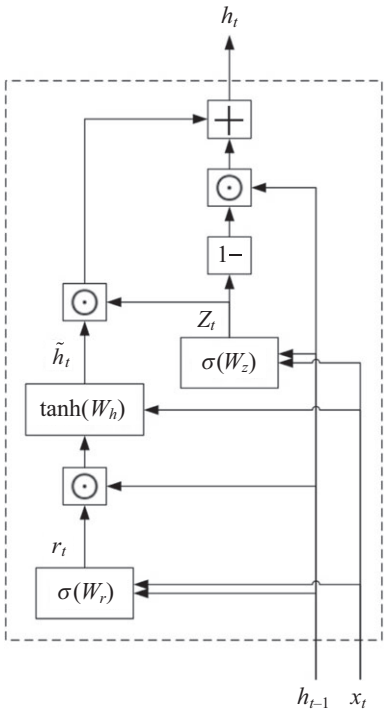

Fig. 4. Procedure of GRU.

III. Experiments and Results

In this subsection, the performance of the proposed method is evaluated for the SED in real-life audio. The subsections contain the data set used for our evaluation, the experimental setup, and a description of the performances obtained with the proposed approach.

1. Evaluation Data Set

In our experiments, the development subset of the TUT sound events detection 2016 database [14], called DCASE 2016, was used. Detection and Classification of Acoustic Scenes and Events (DCASE) 2016 is an official IEEE Audio and Acoustic Signal Processing (AASP) challenge. The DCASE 2016 task3 database, that is, sound event detection in real-life audio, is divided into a training and test set containing 22 recordings. It has two scene categories, that is, home and residential area. The home scene contains 10 recordings with 11 sound event classes, whereas the residential area scene has 12 recordings with seven sound event classes. All of the sounds were stored in PCM format in stereo with 24-bit depth and a 44.1-kHz sampling rate. The length of these recordings is between 3 min–5 min.

In the development subset provided, each of the context data is already partitioned into four folds of training and testing data. Given the training set, data augmentation was explored. The test data was collected such that each recording is used exactly once as the test, and the classes in it are always a subset of the classes in the training data. In addition, 20% of the training data recordings in each fold were selected randomly to be used as validation data. The same validation data were used across all evaluations. More details about the data are in [14].

2. Experimental Setup

During the implementation of the SED system, several experiments were conducted to find the most suitable setup. The performance of the proposed method was compared with those of different classifiers in combination with different features as follows:

https://doi.org/10.4218/etrij.17.0117.0157

- GMM-OF: As the baseline system, a GMM classifier with single-channel features was applied. A GMM consisting of 16 Gaussians was trained for each of the positive and negative values. Sixty dimensional features (20 MFCC, 20 delta, and 20 acceleration features) were extracted from single-channel audio with 40-ms frames and 20-ms hop lengths.
- GMM-SF: Instead of the single-channel features, stereo-channel features including MFCC, delta, and acceleration per frame were applied to the GMM classifier.
- LSTM-OF: Instead of GMM, an LSTM-RNN was used as a classifier with single-channel features. For single-channel features, 20 Mel-band energies per frame were extracted. The LSTM-RNN had an input layer with 40 units (each reading one component of the feature frame) and two hidden layers with 200 LSTM units. The output layer had logistic sigmoid functions and one neuron for each class. The network was trained by back-propagation through time using binary cross-entropy as a loss function.
- LSTM-SF: Instead of the single-channel features, stereo-channel features per frame were applied to the LSTM-RNN classifier. A 512 log-magnitude spectrogram and two pitch range values were extracted from each frame for each of the left and right channels. In addition, 1 TDOA value per frame was extracted using stereo-channel audio.
- LSTM-SP: A 512 log-magnitude spectrogram and two pitch range values were extracted from each frame for each of the left and right channels and were applied to the LSTM-RNN classifier. The TDOA feature was not used.
- GRNN-OF: Instead of LSTM-RNN, a GRU-RNN was used as a classifier with single-channel features. For single-channel features, 20 Mel-band energies per frame were extracted. The GRU-RNN had an input layer with 40 units (each reading one component of the feature frame) and three hidden layers with 200 GRU units. Three-layer GRNNs were initialized with orthogonal weights and rectifier activation functions. The output of GRU-RNN was mapped to the posterior of the target audio events through one feed-forward neural layer, with a logistic sigmoid output activation function. The network was trained by back-propagation through time using binary cross-entropy as a loss function. Regarding the training procedure, we extended the on-the-fly shuffling routine in two ways: we dropped frames with a probability of 50% and used smaller permuted sequence batches.
- GRNN-SF: Instead of the single-channel features, stereo- channel features per frame were applied to the GRU-RNN classifier. A 512 log-magnitude spectrogram and two pitch range values were extracted from each frame for each of the left and right channels. In addition, 1 TDOA value per frame was extracted using stereo-channel audio.
- GRNN-SP: A 512 log-magnitude spectrogram and two pitch range values were extracted from each frame for each of the left and right channels and were applied to the GRU-RNN classifier. The TDOA feature was not used.

Single-channel audio was created by averaging the stereo channels in order to compare the performance of the TDOA and pitch range features. We used the PyBrain Toolbox [22] to implement the LSTM and GRU. In the configuration of LSTM-RNN and GRU-RNN, the size of the input layer was changed according the feature dimensions.

As evaluation metrics of system performance for sound event detection, we used the error rate (ER) and F-scores calculated on 1-s-long segments:

- ER measures the amount of errors in terms of insertions (I), deletions (D), and substitutions (S). After counting the number of substitutions per segment, the remaining false positives in the system output were counted as insertions, and the remaining false negatives as deletions. The error rate was then calculated by integrating segment-wise counts over the number of segments K, with $N(k)$ being the number of active ground truth events in segment k.

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)}. \quad (11)$$

- F-score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score. p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F-score can be interpreted as a weighted average of the precision and recall. It reaches its best value at 100% and its worst at 0%.

$$F = \frac{2 \times precision \times recall}{presision + recall}. \quad (12)$$

The F-score has an intuitive meaning. It indicates how precise the classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). Therefore, good scores on both precision and recall will be favored over very

good performance for one and poor for the other. This makes the *F*-score a more meaningful and robust metric, which is widely used in the context of SED.

The detailed measures are explained in [23].

3. Results

Table 1 presents the experimental results of the proposed method compared to the baseline and the several methods including alternative types of neural network architecture and different feature sets.

As shown in Table 1, the overall performance of the baseline system, GMM-OF, is a context average *ER* of 0.90 and *F*-score of 28.7%. This value is slightly higher than those of the method using GMM-SF. An ideal system should have an *ER* of 0 and an *F*-score of 100%. From the results, we can see that the best SED performance is obtained with the proposed method, GRNN-SF, in terms of both *ER* and *F*. The *F*-score and *ER* of GRNN-SF on segment-based metrics are 0.76 and 50.3%, respectively. In addition, the results of GRNN-OF are less than the performance of GRNN-SF.

The results of GRNN-SP are also less than the performance of GRNN-SF, but they are better than the performance of GRNN-OF. The LSTM-SF achieves better results than the LSTM-OF and LSTM-SP in terms of *ER* and *F*, and is slightly lower than the proposed method. The LSTM-SP provides a slightly better performance than the LSTM-OF in terms of *ER* and *F*. Compared to the other five methods, the method using GMM-SF attains the worst results.

The overall performance of sound event detection based on two variants of RNN such as LSTM and GRU increases significantly by using stereo-channel feature extraction. Although the GRU incurs a 19.23% smaller run time compared with that of LSTM, the performance of the GRU-RNN is slightly higher than that of the LSTM.

In Tables 2 and 3 and Fig. 5, more detailed comparisons of GRU vs. LSTM are presented.

In Table 2, we test the effect of the number of recurrent layers. Here, the hidden layer size is 200 for both LSTM and GRU cases. From the results, we can see that the performance is degraded when the number of layers is > 3 for GRU and when the number of layers is > 2 for LSTM.

In Table 3, we test the effect of the hidden layer size. Here, the number of hidden layer is 3, according to the above experiments. From Table 3, we confirm that GRU-based systems outperform the LSTM-based systems. This may be because the GRU based systems can better avoid overfitting, and the unbounded output and gradient help the GRU-based systems train sufficiently.

In Fig. 5, we test the effect of the convergence epoch. In practical applications, training is conducted for many epochs, and the network parameters are saved if the

Table 1. Comparison of segment-based error rate and *F*-score for different combinations of classifiers and features.

Methods	Home		Residential area		Average	
	ER	*F* (%)	*ER*	*F* (%)	*ER*	*F* (%)
GMM-OF	0.95	19.6	0.85	37.8	0.90	28.7
GMM-SF	0.96	18.9	0.88	36.9	0.92	27.9
LSTM-OF	0.95	35.6	0.79	51.2	0.87	43.4
LSTM-SF	0.91	39.8	0.71	57.6	0.81	48.7
LSTM-SP	0.96	38.9	0.76	53.2	0.86	46.1
GRNN-OF	0.93	37.3	0.73	55.1	0.83	46.2
GRNN-SF	0.84	41.5	0.68	59.1	0.76	50.3
GRNN-SP	0.90	40.7	0.72	54.9	0.81	47.8

Table 2. Comparisons of *ER* results between LSTM and GRU networks while varying the number of layers.

Methods	Number of layers					
	1	2	3	4	5	6
GRNN-SF	0.831	0.783	0.761	0.765	0.765	0.768
LSTM-SF	0.829	0.812	0.814	0.817	0.817	0.821

Table 3. Comparisons of *ER* results between LSTM and GRU networks with various units per layer.

Methods	Number of layers					
	50	100	150	200	250	300
GRNN-SF	0.813	0.787	0.775	0.761	0.761	0.761
LSTM-SF	0.843	0.831	0.823	0.814	0.821	0.825

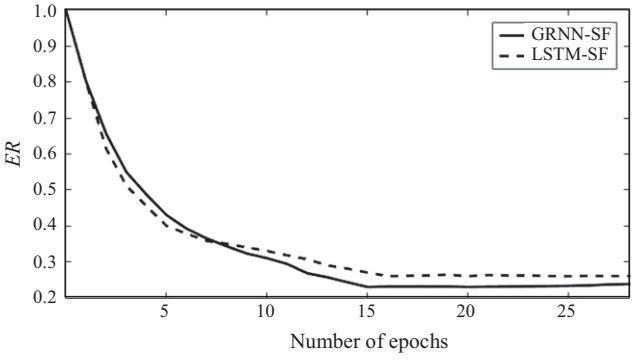

Fig. 5. Comparisons of *ER* results between LSTM and GRU networks while varying the number of epochs.

https://doi.org/10.4218/etrij.17.0117.0157

validation set performance increases. Here, the number of hidden layers is 3, and the hidden layer size is 200 for both the LSTM and GRU cases. The evolution of the training with the small set is depicted in Fig. 5 with a best validation set performance of 0.23 *ER* in epoch 15 for the GRU. The converence curve of the LSTM is slower than that of the GRU. For training, we chose 100 sound files with four sound classes selected from the home scene category in the training set. We refer to this as the small set.

As expected, we observed that the proposed method using GRNN in combination with auditory high-resolution spectral features of stereo-channel audio is more efficient and effective compared to the state-of-the-art mono or stereo-channel methods.

IV. Conclusion

In this paper, GRNNs with high-resolution spectral features of stereo-channel audio were proposed and evaluated for multilabel sound event detection. The experimental results showed that the proposed method obtained better results compared with other previously reported methods. The multichannel features were seen to be performing significantly better than the GMM baseline system using monochannel features. The accuracy of the GRU was slightly higher than that of the LSTM, although the GRU has one less hidden state and one less gate than the LSTM. In addition, the GRU requires a smaller run-time compared to the LSTM.

In future work, we will focus on extending GRNNs by coupling them with convolutional neural networks and deep neural networks to improve the detection accuracy of overlapping sounds. The method will be applied to audio-based context sensing in wireless acoustic sensor networks to the Internet-of-Things.

Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Rep. of Korea (NRF-2015R1D1A1A01059804). And the present Research has been conducted by the Research Grant of Kwangwoon University in 2017.

References

[1] D. Zhang and D. Ellis, "Detecting Sound Events in Basketball Video Archive," Dept. Electronic Eng., Columbia Univ., New York, USA, Speech & Audio Processing Class Project Rport, 2001.

[2] T. Heittola et al., "Audio Context Recognition Using Audio Event Histograms," *Eur. Signal Process. Conf.*, Aalborg, Denmark, Aug. 23–27, 2010, pp. 1272–1276.

[3] Y. Peng et al., "Healthcare Audio Event Classification Using Hidden Markov Models and Hierarchical Hidden Markov Models," *IEEE Int. Conf. Multimedia Expo*, New York, USA, June 28–July 3, 2009, pp. 1218–1221.

[4] A. Harma, M.F. McKinney, and J. Skowronek, "Automatic Surveillance of the Acoustic Activity in our Living Environment," *IEEE Int. Conf. Multimedia Expo*, Amsterdam, Netherlands, July 6–8, 2005, pp. 634–637.

[5] T. Heittola et al., "Context-Dependent Sound Event Detection," *EURASIP J. Audio Speech Music Process.*, vol. 2013, no. 1, Feb. 2013, pp. 1–13.

[6] E. Cakir et al., "Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks," *Int. Joint Conf. Neural Netw.*, Killarney, Ireland, July 12–17, 2015, pp. 1–7.

[7] E. Cakir et al., "Multi-label vs. Combined Single-Label Sound Event Detection with Deep Neural Networks," *Eur. Signal Process. Conf. (EUSIPCO)*, Nice, France, Aug. 31–Sept. 4, 2015, pp. 2551–2555.

[8] H. Zhang, I. McLoughlin, and Y. Song, "Robust Sound Event Recognition Using Convolutional Neural Networks," *IEEE Int. Conf. Acoust., Speech Signal Process*., Brisbane, Australia, Apr. 19–24, 2015, pp. 559–563.

[9] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *IEEE Int. Conf. Acoust., Speech Signal Process*., Vancouver, Canada, May 26–31, 2013, pp. 6645–6649.

[10] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings," *IEEE Int. Conf. Acoust., Speech Signal Process*., Shanghai, China, Mar. 20–25, 2016, pp. 6440–6444.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated Feedback Recurrent Neural Networks," *Int. Conf. Mach. Learn.*, Lille, France, July 6–11, 2015, pp. 2067–2075.

[12] M. Zoehrer and F. Pernkopf, "Gated Recurrent Networks Applied to Acoustic Scene Classification and Acoustic Event Detection," *Detection Classification Acoust. Scenes Events*, Budapest, Hungary, Sept. 3, 2016, pp. 1–5.

[13] S. Adavanne et al., "Sound Event Detection in Multi-channel Audio Using Spatial and Harmonic Features," *Detection Classification Acoust. Scenes Events*, Budapest, Hungary, Sept. 3, 2016, pp. 1–5.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Database for Acoustic Scene Classification and Sound Event Detection," *Eur. Signal Process. Conf.*, Budapest, Hungary, Aug. 29–Sept. 2, 2016, pp. 1128–1132.

[15] C. Knapp and G. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoust. Speech Sig. Process*., vol. 24, no. 4, Aug. 1976, pp. 320–327.

[16] B. Uzkent, B.D. Barkana, and H. Cevikalp, "Non-speech Environmental Sound Classification Using SVMs with a New Set of Features," *Int. J. Innov. Comput. I.*, vol. 8, no. 5 (B), May 2012, pp. 3511–3524.

[17] P.N. Garner, M. Cernak, and P. Motlicek, "A Simple Continuous Pitch Estimation Algorithm," *IEEE Sig. Process. Lett.*, vol. 20, no. 1, Jan. 2013, pp. 102–105.

[18] J.A. Morales-Cordovilla et al., "A Pitch Based Noise Estimation Technique for Robust Speech," *IEEE Int. Conf. Acoust., Speech Sig. Process*., Prague, Czech Republic, May 22–27, 2011, pp. 4808–4811.

[19] M. Espi, M. Fujimoto, and T. Nakatani, "Acoustic Event Detection in Speech Overlapping Scenarios Based on High-Resolution Spectral Input and Deep Learning," *IEICE Trans. Inform. Syst.*, vol. E98–D, no. 10, Oct. 2015, pp. 1799–1807.

[20] J. Chung et al., "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," CoRR., Accessed 2017. https://arxiv.org/abs/1412.3555

[21] G.-B. Zhou et al., "Minimal Gated Unit for Recurrent Neural Networks," *Int. J. Autom. Comput.*, vol. 13, no. 3, June 2016, pp. 226–234.

[22] T. Schaul et al., "Pybrain," *J. Mach. Learn. Res.*, vol. 11, Feb. 2010, pp. 743–746.

[23] A. Mesaros, T. Heittola, and T. Vitrtannen, "Metrics for Polyphonic Sound Event Detection," *Appl. Sci.*, vol. 6, no. 6, May 2016, pp. 1–17.

Hyoung-Gook Kim received a Dr.-Ing. degree in Electrical Engineering and Computer Science from the Technical University of Berlin, Germany. From 1998 to 2005, he worked on mobile service robots at Daimler Benz, and speech recognition at Siemens, Berlin, Germany. Since 2007, he has been a professor in the Department of Electronics Convergence Engineering, Kwangwoon University, Seoul, Rep. of Korea. His research interests include audio signal processing, audiovisual content retrieval, and speech enhancement.

Jin Young Kim received the Ph.D. degree in Electronic Engineering from Seoul National University, Rep. of Korea. He worked on speech synthesis at Korea Telecom from 1993 to 1994. Since 1995 he has been a professor in the Department of Electronics and Computer Engineering, Chonnam National University, Gwangju, Rep. of Korea. His research interests are speech synthesis, speech and speaker recognition, and audiovisual speech processing.

https://doi.org/10.4218/etrij.17.0117.0157