


RESEARCH

Open Access



Music detection from broadcast contents using convolutional neural networks with a Mel-scale kernel

Byeong-Yong Jang¹, Woon-Haeng Heo¹, Jung-Hyun Kim² and Oh-Wook Kwon^{1*} 

Abstract

We propose a new method for music detection from broadcasting contents using the convolutional neural networks with a Mel-scale kernel. In this detection task, music segments should be annotated from the broadcast data, where music, speech, and noise are mixed. The convolutional neural network is composed of a convolutional layer with kernel that is trained to extract robust features. The Mel-scale changes the kernel size, and the backpropagation algorithm trains the kernel shape. We used 52 h of mixed broadcast data (25 h of music) to train the convolutional network and 24 h of collected broadcast data (ratio of music of 50–76%) for testing. The test data consisted of various genres (drama, documentary, news, kids, reality, and so on) that are broadcast in British English, Spanish, and Korean languages. The proposed method consistently showed better performance in all the three languages than the baseline system, and the *F*-score ranged from 86.5% for British data to 95.9% for Korean drama data. Our music detection system takes about 28 s to process a 1-min signal using only one CPU with 4 cores.

Keywords: Music detection, Music segmentation, Convolutional neural networks, Mel-scale filter bank

1 Introduction

Broadcast contents consist of various signals, such as music, speech, background music, noise, and sound effects. The music type of broadcast content is diverse, for example, classical music, popular song, rap, and instrumental music. On average, music has a higher proportion of broadcast contents than the other signals. Therefore, music detection is an elementary factor in research related to the processing of broadcast data, such as automatic data tagging, speech/music classification, music and background music detection, and music identification. Among these, music identification is a research topic that can be applied to various services to provide music information to users, or identification when claiming royalties. Music detection results can be helpful in the preprocessing step for fast and robust music identification.

IberSPEECH held audio segmentation challenges in 2010 [1] and 2014 [2], while MIREX held music detection challenges in 2015 [3] and 2018 [4]. The goal of the audio

segmentation challenges in the IberSPEECH 2014 was to segment and label audio documents indicating where speech, music, and/or noise were present. Because the 2010 and 2014 IberSPEECH challenges used the Catalan broadcast news database [5], the proportion of speech in the test set was high. Consequently, research works focused on detecting speech, rather than music. The best system in the 2014 [2] challenge combined the results of two subsystems. The first subsystem uses the hidden Markov model (HMM) to classify the non-overlapping class. The second subsystem used the Gaussian mixture model (GMM) and multilayer perceptron (MLP) to classify the detailed classes of speech. In this system [2], they extracted the Mel-frequency cepstral coefficient (MFCC) [6] and i-vector [7].

In the music/speech classification and detection challenge of MIREX2015, the music/speech detection was attempted for the data set of the British Library's World and Traditional Music collections. The challenge provided a sample of the evaluation data set [3, 8]. The data had a clear boundary between music and speech, i.e., there was less overlap between voice and music. The highest performance system [9] in this challenge consisted of two

* Correspondence: owkwon@cbnu.ac.kr

¹School of Electronics Engineering, Chungbuk National University, Cheongju, South Korea

Full list of author information is available at the end of the article

steps. In the first step, a symmetric Kullback Leibler (KL) distance was used to detect the music/speech boundary. In the second step, a second music/speech classifier was used to classify each segment. The MFCC was used as a feature to detect the music/speech boundary. In the music and/or speech detection challenge of MIREX2018, there are four tasks of music detection, speech detection, music and speech detection, and music relative loudness estimation. The system [10] that showed the best performance in music detection was the CNN model that was trained using TV and radio broadcasting data and annotation and that used the Mel-spectrogram.

Theodorou et al. [11] summarized the structure of audio segmentation in three broad categories, i.e., distance-based, model-based, and hybrid of distance- and model-based. The distance-based audio segmentation is an algorithm that detects the boundary of specific acoustic categories by the use of a distance matrix of acoustic features. For distance-based audio segmentation, the MFCC and zero crossing rate (ZCR) were the main features used. The distance matrix was computed using these features with various distance measures. In their paper, they introduced Euclidian distance, Bayesian information criterion (BIC), KL distance, generalized likelihood ratio (GLR), and so on as distance measures. The model-based audio segmentation is a method of classifying each frame using a trained model. GMM-HMM and support vector machine (SVM) were mainly used as machine learning algorithms for the audio segments. MFCC, ZCR, and spectrum-based features (such as energy, pitch, bandwidth, flux, centroid, roll-off, spread, and flatness) were mainly used as the features for training the model of machine learning algorithms.

Due to the development of deep learning, many recent studies utilize the deep neural network (DNN) algorithms. Grill and Schluter [12] used the CNN and self-similarity lag matrices (SSLMs) for music boundary detection. They trained the network with two input features that used the Mel-scaled log-magnitude spectrogram and SSLMs as inputs of the CNN. They employed four different network architectures to combine the two input features. Among the four architectures, the fusion in the convolutional layer showed the best performance. When used independently without combining the two features, the Mel-scaled spectrogram showed better performance for boundary detection than the SSLM, indicating that the Mel-scaled spectrogram is a good feature in the CNN-based processing.

Doukhan and Carrive [13] used a CNN model for music/speech classification and segmentation. They extracted the Mel-frequency cepstra, corresponding to 40 Mel-scale bands, to train the model. The CNN model consisted of two convolutional and two dense layers. They trained the first convolutional layer with an unsupervised

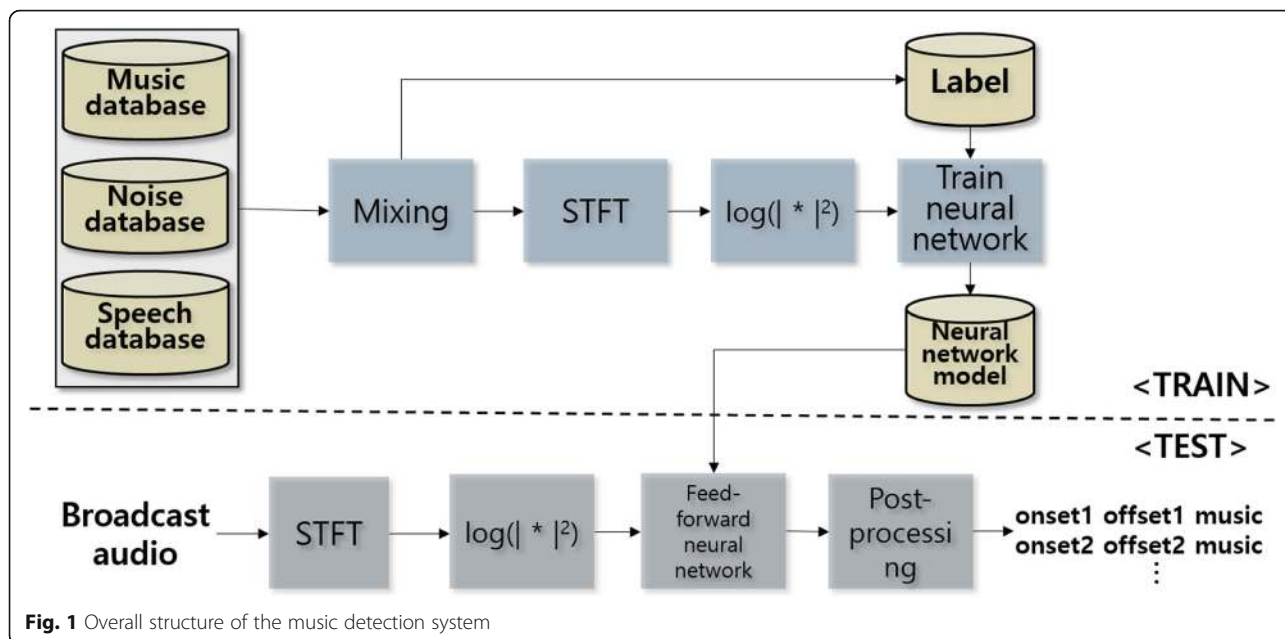
procedure based on the spherical k-means and zero-phase component analysis (ZCA)-based whitening [14]. They used the CNN model to classify each frame into either speech or music. They then determined the music/speech segments using the Viterbi algorithm. Using the MIREX 2015 music/speech detection training example material [3], they achieved a recall performance of 82.73% for only music segment without post-processing and 91.07% with post-processing (Viterbi algorithm). They also published and distributed their systems [15].

Tsipas et al. [16, 17] introduced an audio-driven algorithm for the detection of speech and music events in multimedia contents. They used a distance-based method using a self-similarity matrix and a model-based method using the SVM. They first computed the self-similarity matrix with cosine distance, to find the boundary between music and speech. They then used the SVM to classify each frame into speech or music. Finally, by combining the classification results of the frame unit and the detected boundary result, the result of the segment index was output. They extracted the ZCR, flux, spectral roll-off, root mean square energy, MFCC, and spectral flatness per band for boundary detection and classification.

Seyerlehner et al. [18] proposed a new feature, called continuous frequency activation (CFA), which is especially designed for music detection. They focused on the fact that music tends to have more stationary parts than speech. They extracted features that represented the horizontal component of music. They detected music using the extracted features and machine learning algorithms. They detected the music in TV production data with a lot of speech and noise and provided a sample audio to publicize their data environment [19]. They classified music and non-music by comparing the proposed CFA feature value with the threshold value. Their method of music detection improved the classification accuracy from 81.21 to 89.93%. Moreover, they released some examples of those misclassifications [19]. Wieser et al. [20] improved the performance of speech/music discrimination using the SVM with CFA and MFCC.

In addition, Choi et al. [21] improved the automatic music tagging performance using Mel-spectrogram and CNN structure, while Jansson et al. [22] improved the performance of singing voice separation using spectrogram and CNN structure.

We propose a convolutional layer with a Mel-scale kernel (*melCL*) for music detection in the broadcast data. We detected music from real broadcast data, which included music mixed with noise, and speech-like background music. Our test data is very similar to the data used in [19]. We trained the CNN model for a model-based music detection system and applied the proposed *melCL* to the first convolutional layer. We used the test data and a public data set to compare the music detection



performance between the proposed method and the baseline systems.

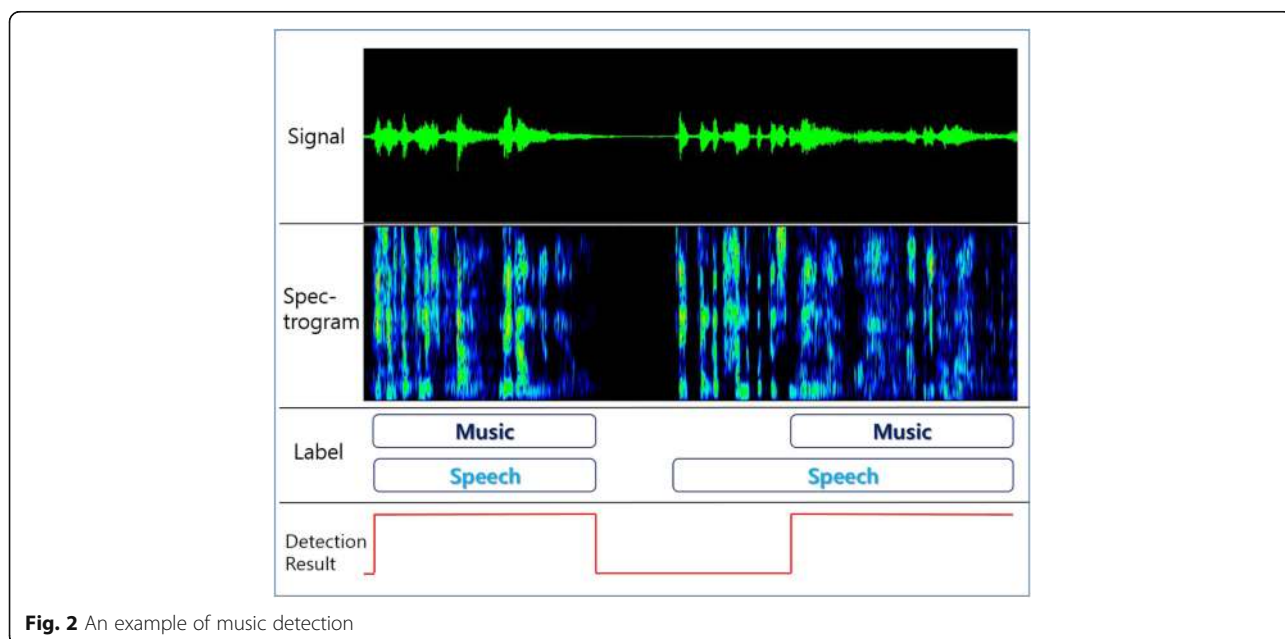
The paper is organized as follows: Section 2 describes the overall structure of music detection and the proposed algorithm. Section 3 reports the experimental results and discussion. Section 4 presents the conclusions and describes future works.

2 Proposed method

2.1 Overall structure for music detection

Figure 1 shows the overall structure of the music detection system. The upper part of the figure shows the

model learning process for music detection. The lower part shows the music detection process of broadcast data. We employed mixed data obtained by mixing music, speech, and noise signals to train the CNN model. The log power spectrogram was used as input to CNN for model learning. For music detection, we calculated the log spectrograms and then classified music and non-music on a frame-by-frame basis using the trained CNN model. We post-processed the frame-by-frame CNN results, and output music sections whose onset and offset positions were annotated. Figure 2 shows an example of music detection.



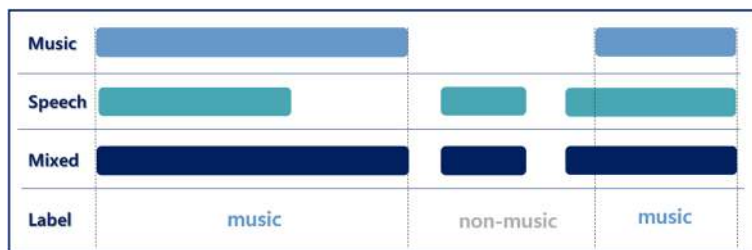


Fig. 3 Mixing and labeling

2.2 Mixing data

We mixed music, speech, and noise database to train the model for the music detection. To create data similar to the broadcast data, we created “music with speech,” “music with noise,” and “speech with noise,” by mixing each pure data (music, speech, and noise). Equation 1 shows the mixing method:

$$\text{Mixed data} = g \cdot s_1 + s_2, g = 10^{k/20} \cdot \frac{E(s_2)}{E(s_1)} \quad (1)$$

where s is signal, g is gain, k is target decibel (dB), and $E(\cdot)$ is energy. We automatically generated the label of the mixed data, and Fig. 3 shows the criterion. That is, the music duration of pure music is labeled as the music duration of the mixed data. The “speech with noise” data is labeled as non-music in all durations.

2.3 Feature extraction for CNN input

We computed the log power coefficients (spectrogram) of the short-time Fourier transform (STFT) with a window size of 25 ms (400 samples at 16 kHz sampling rate), shift size of 10 ms (160 samples at 16 kHz sampling rate), and 512-point FFT (fast Fourier transform). In contrast to the conventional processing that was used in the previous studies [2, 11–13, 16, 20], we employed the use of a convolution layer with a Mel-scale kernel, instead of a Mel-scale filter bank. The final dimension of

a feature vector of CNN input was (257×101) by splicing 50 frames on either side.

2.4 Proposed convolutional layer with a Mel-scale kernel

The Mel-scale filter bank is similar to the human auditory characteristics. Whereas the interval of the filter banks in the low-frequency region was narrow, the interval in the high-frequency region was wide. The filter shape of the common filter banks [6] was as shown in Fig. 4. Until recently, the features with the Mel-scale (Mel-scaled spectrogram, MFCC) have been widely used for processing audio signals and have produced good performance [2, 11–13, 16, 20, 21, 23]. However, several studies have recently attempted to detect music by suggesting or adding new features [12, 18, 20] because there are limitations in music detection through using features extracted from *fixed* filter bank shapes. To overcome the limitations, we attempted to implement a new filter bank whose center frequency was located in the Mel-scale, but whose filter shape was learned from input data. From this reasoning, we proposed to use a convolutional layer with a Mel-scale kernel (*melCL*).

We describe the Mel-scale spectrograms and one-dimensional convolutional layers to help understand *melCL*. The conventional Mel-scale spectrogram is the Fourier transform point multiplied by the Mel-scale filter, as shown below:

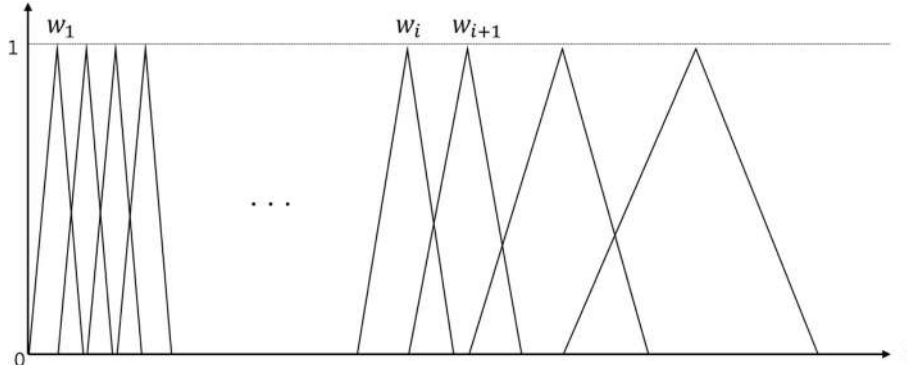


Fig. 4 Mel-scale filter bank

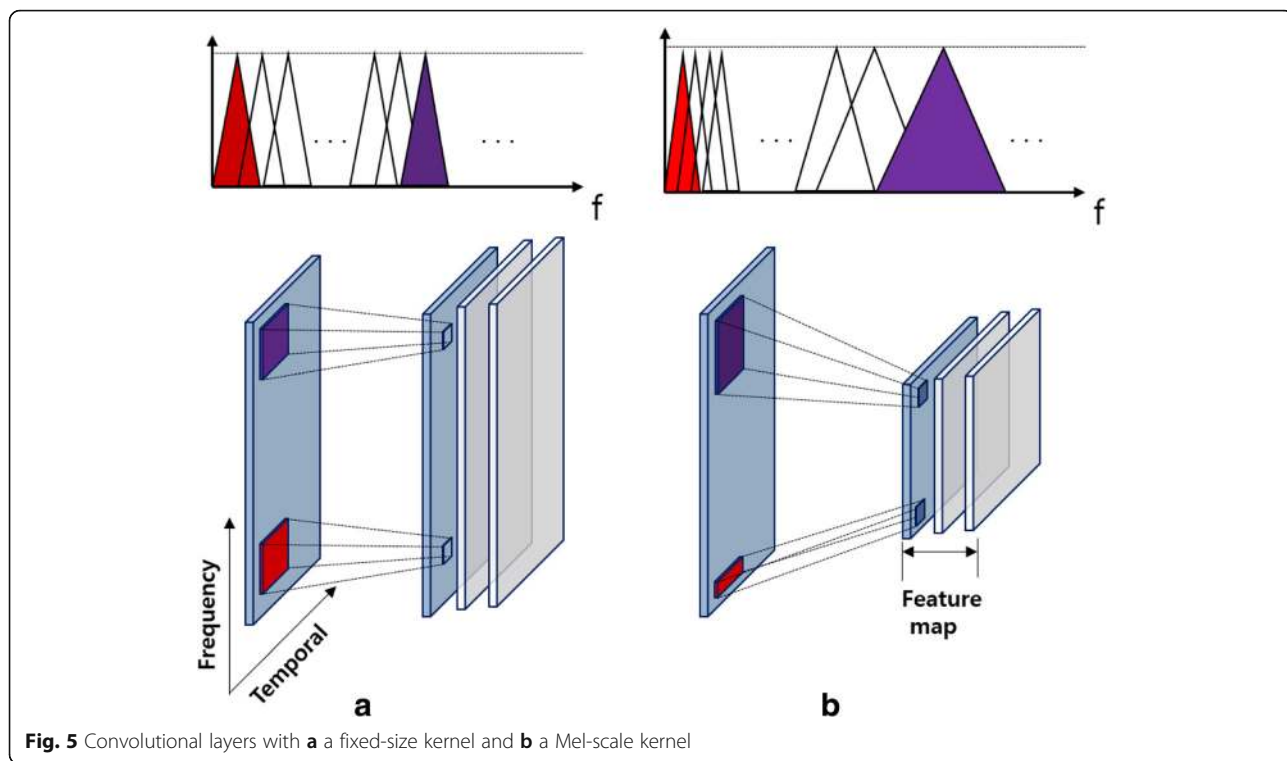


Fig. 5 Convolutional layers with a a fixed-size kernel and b a Mel-scale kernel

$$y_i^t = w_i \cdot x^t \tag{2}$$

where x^t is a Fourier transform point vector for the t -th frame, and w_i is the weight of the i -th bin of Mel-scale filter bank. The weight of the i -th bin has fixed values with a triangular shape, and the size of the w_i bin changes according to the Mel-scale, as shown in Fig. 4. The Mel-scale filter bank is advantageous for audio processing, because of its different sizes depending on the frequency. Because the low-frequency regions contain more information than the high-frequency regions,

the Mel-scale helps to extract robust features from the audio data.

Next, the kernel equation of the one-dimensional convolution layer is as shown below:

$$y_i^t = F(w \cdot x_t + b) \tag{3}$$

where w and b are the weight and the bias of kernel and F is an activation function. The kernel weight of CNN can be trained to benefit music detection through the

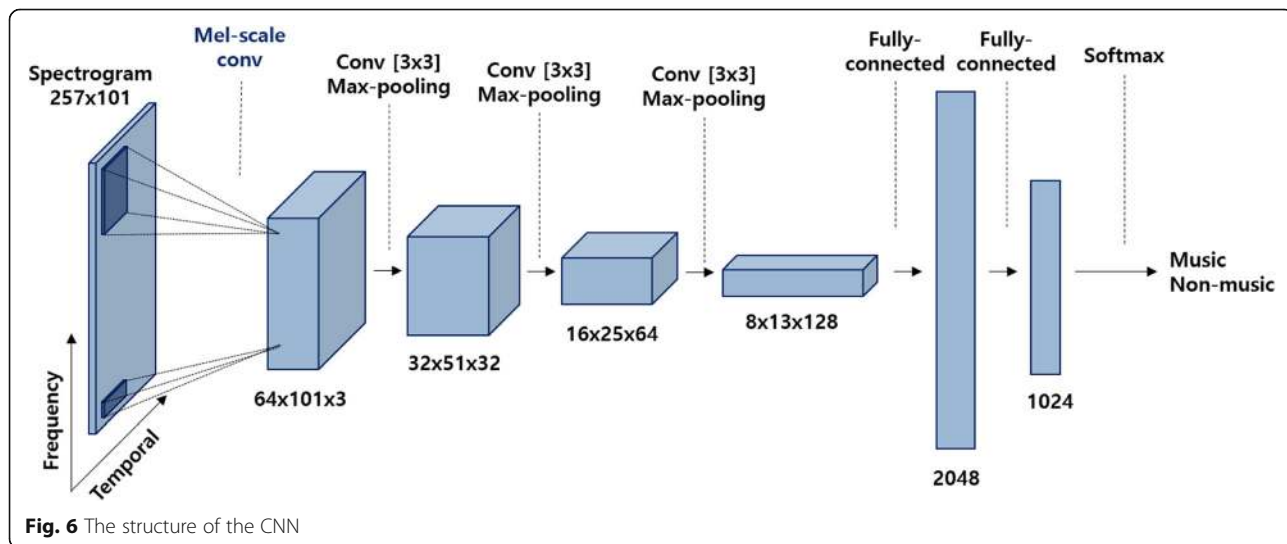


Fig. 6 The structure of the CNN

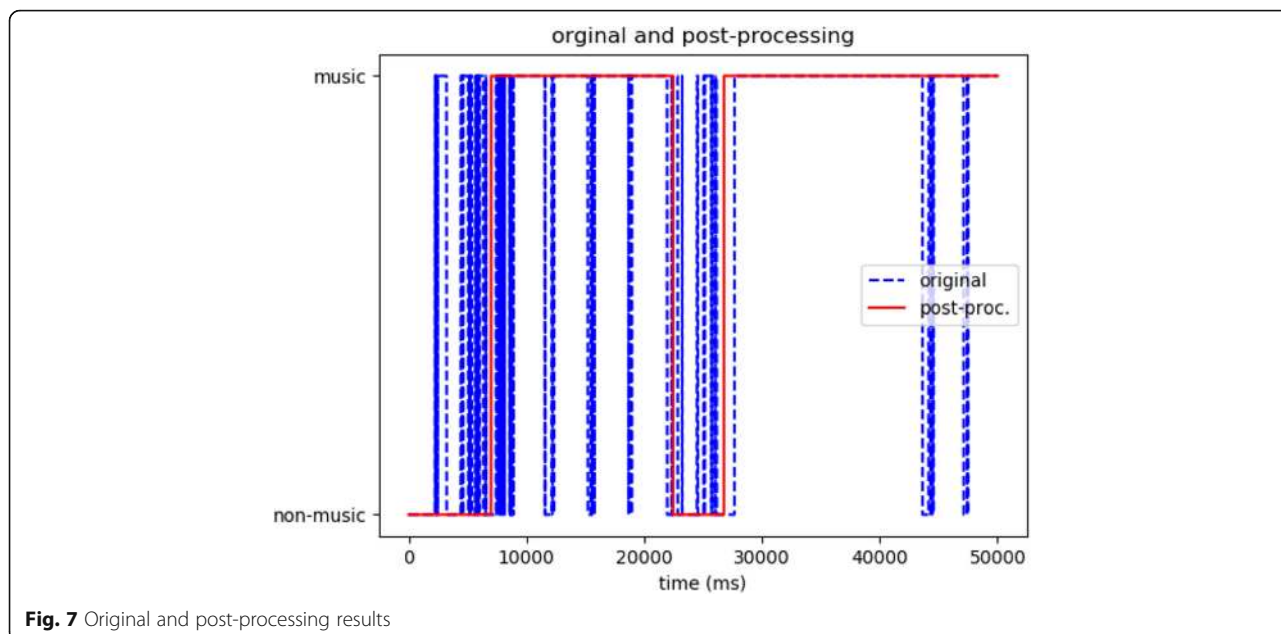


Fig. 7 Original and post-processing results

backpropagation algorithm. However, a fixed-size kernel will be applied for all frequency regions.

We have implemented the *melCL* by combining these two advantages. We also implemented a two-dimensional kernel composed of a two-dimensional convolution layer to learn more about optimized filters. Figure 5 shows the basic convolution layer with (a) a fixed kernel and (b) a convolution layer with the Mel-scale kernel. Figure 5 a shows the temporal and frequency dimensions of the kernel fixed at all times. However, Fig. 5 b shows the frequency dimension of the kernel is large in the high-frequency region and small in the low-frequency region. We initialize the kernel weight of the *melCL* to the weight of the Mel-scale filter bank, to obtain the stability of the learning process and improved performance. The *melCL* has a temporal dimension of kernel of 5 with stride 1, the hyperbolic tangent activation function, and a feature map of 3.

2.5 CNN

We used a CNN with a Mel-scale convolutional layer and 3 convolutional layers appended with 2 fully connected feed-forward layers, and a softmax layer for class output. The three subsequent conventional convolutional layers of the CNN had feature maps of 32, 64, and 128, respectively.

Table 1 Information on mixed data

Mixed data	Data 1	Data 2	Target dB (k)	Total duration (h)
Music with speech	Library music	Librivox	-30-0 dB	25
Music with noise	Library music	ESC-50	0-30 dB	25
Speech with noise	Librivox	ESC-50	0-30 dB	25

Each convolutional layer had a 3×3 kernel with stride 1, ReLU (rectified linear unit) activation function, and a 2×2 max pooling with stride 2. Figure 6 shows the detailed structure of the CNN. The CNN was trained for 50 epochs with cross-entropy loss function, Adam optimizer, mini-batch size of 300, learning rate of 0.001, and dropout probability of 0.4.

2.6 Post-processing

The music detection results obtained from frame-by-frame processing appear as a very small segment. However, the music duration of broadcast data is mostly long, which includes many frames. We applied a median filter [24] with a size of 5 s to the frame-by-frame detection results, in order to obtain smoothed segmentation results. The median filter was repeatedly applied 3 times. The post-processing removes or merges the small segments to represent the detection result as a large segment. Figure 7 shows the original and post-processing results.

3 Experimental results and discussion

3.1 Database

We used the mixed data set to train the model for music detection. This data set was created by mixing music, speech, and noise. We used 25 h of library music (song, classic, instrumental, and so on), 25 h of the librivox (speech) in the MUSAN database [25], and 2 h 46 min of the ESC-50 database (noise) [26]. Table 1 shows the information on the mixed data used in the training. In order to equalize the duration of data 1 and 2, we copied data 2, or cut it to the length of data 1. In Table 1, k is a random value of 5 units in the range. The “music with speech” was mixed so that the energy of

Table 2 Ratio (%) of music and non-music

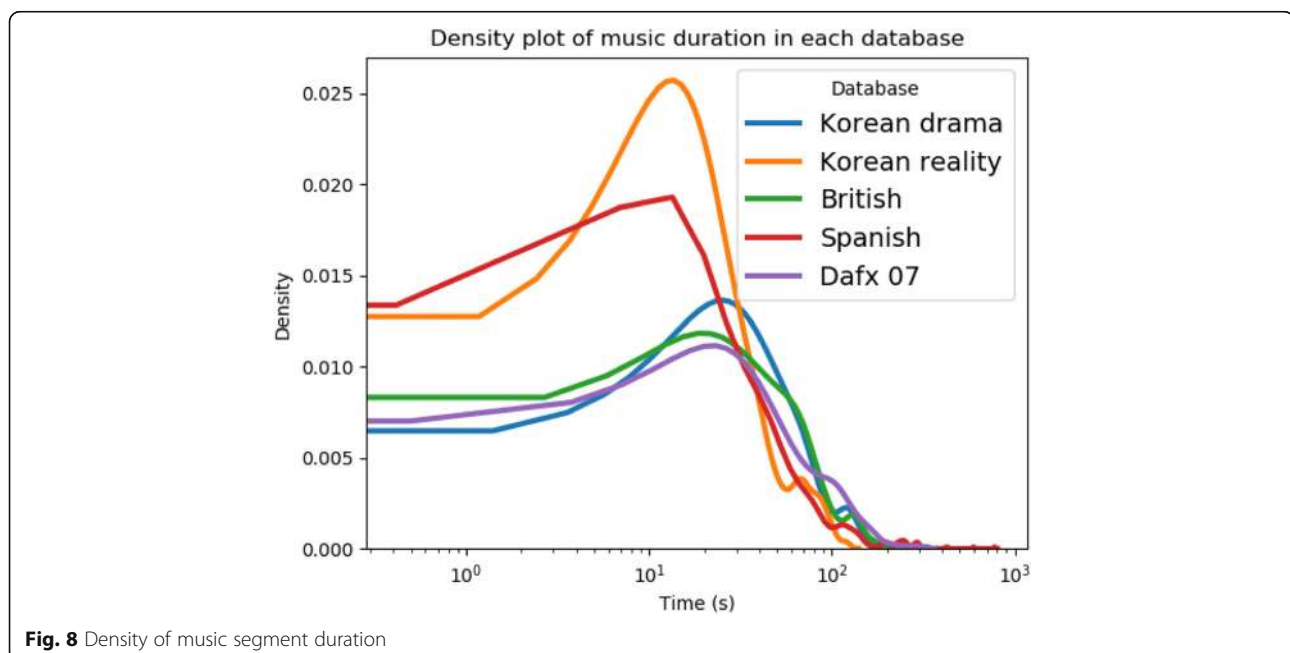
Label	Korean drama (dev)	Korean reality	British broadcast	Spanish broadcast	MIREX 2015	Dafx 07
Pure music	35.2	19.8	26.9	29.5	74.3	43.0
Music + noise	15.4					
Music + speech	15.7	56.5	20.0	20.5	2.1	
Music + speech + noise	1.0					
Total music	67.5	76.3	53.0	50.0	76.5	43.0
Total non-music	32.4	23.6	46.9	50.0	23.5	57.0
Total duration (h)	2.9	1.6	7.9	12	5.24	9

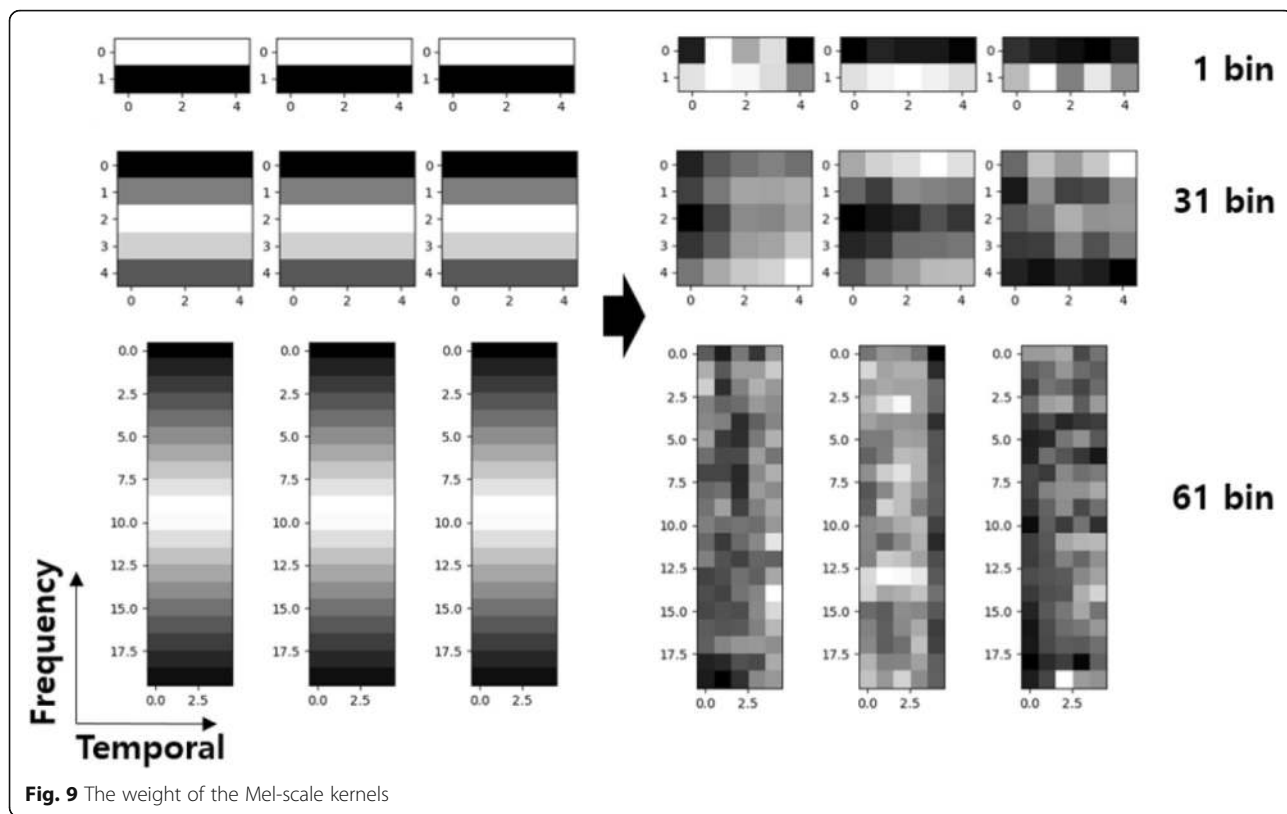
music signal is lower than that of speech, which is similar to broadcast data.

We collected the broadcast data in 3 languages, i.e., British English, Spanish, and Korean, for the test of music detection. The broadcast data in British English and Spanish included various genres, such as drama, documentary, news, and kids. They had 8 and 12 h, respectively. The Korean broadcast data consisted of 3 h of drama and 1 h and 30 min of reality show. They also included metadata that manually tagged the music segment. We used Praat [27] to create metadata. The metadata includes speech and music start and end times. Tagging of music segment and speech segment was performed independently. Accordingly, these data include background music and even include music whose power of music is smaller than the power of speech or noise. In particular, the metadata of Korean drama data also includes noise segments. We used the Korean drama data as a development data set for model selection and parameter adjustment.

We also used two public data for evaluation. The first was the MIREX 2015 data for music/speech detection [3]. The second was Seyerlehner's dafx 07 data for music detection in television productions [18]. The length of the MIREX 2015 data was 5 h and included classical, folk, ethnic, and country music. We note that the data also contained a very small amount of overlapping music and speech. The dafx 07 data was 9 h and includes talk show, music show, documentary, news, soap opera, parliament, and cooking show.

Table 2 shows the ratio of music and non-music of the data we collected and the public data. Since most of the data is tagged only for music and speech segments, it is not possible to display the mixing ratio for the noise. However, since the noise segment is tagged in the Korean drama data, the mixing ratio of the noise is also displayed. The dafx 07 data is tagged only in the music segment. In the case of the British, Spanish, and dafx 07 data, the ratio of music is low, unlike other data, because it contains a genre with a





small proportion of music (news, talk show, and so on), which can be expected to be more difficult data for music detection. Figure 8 shows the density of the music segment duration. The peak of the density distribution is around 20 s, which means that more than half the music segment duration is longer than 20 s. A very long music segment is even longer than 700 s. These long segments of music make it difficult to detect music.

3.2 Baseline systems

We implemented the baseline systems for performance comparison with our proposed algorithm. The first baseline system had the same CNN structure as the structure of the proposed algorithm, but did not include the *melCL*. Other baseline systems had a RNN structure, which was composed of the bidirectional GRU (gated recurrent unit) [28] or LSTM (long short-term memory) layers [29]. The RNN structure had 2 layers with 1024 node, the hyperbolic tangent activation function, and backpropagation through time (BPTT) of 101 (the same as the temporal dimension of the CNN input). Each baseline system implemented two versions, using the log power coefficients (spectrogram) of STFT with 512 FFT points or Mel-spectrograms as inputs of CNN or RNN.

3.3 Model selection

We had the model (baseline and proposed system) learn with many epochs (over 50 epochs). However, it is well known that as the model is repeatedly learned, the model overfits the training data. In addition to our system, overfitting is a more critical problem, because the domains of training data (mixed data) and test data (broadcast data) are very different. For this reason, we needed to select the optimal model. We saved the model every 5k iterations (1k for RNN). We evaluated the saved models using the development datasets and selected the best performance model. We repeated the overall experiment three times and confirmed the consistency of the experimental results.

4 Results

We first visualized the weights of the learned Mel-scale kernels. Figure 9 shows the initial weights and the final weights of the Mel-scale kernels. Each row corresponds to the bins #1, #31, and #61 of the Mel-scale filter, the left is the initial weight, and the right is the weight after learning. Each bin has three kernels, depending on the number of feature maps. We did not find any special patterns in the learned kernel, but we found that the learned kernel is different from the Mel-scale filter,

Table 3 Performance of the proposed and baseline systems

Test data	Model type	<i>F</i> -score (%)	Precision (%)	Recall (%)
Korean drama (dev)	Spectrogram + CNN with <i>melCL</i> (proposed)	95.9	95.9	96.0
	Spectrogram + CNN	92.2	94.0	90.5
	Mel-spectrogram + CNN	94.2	95.7	92.8
	Spectrogram + bi-GRU	88.0	87.0	89.0
	Mel-spectrogram + bi-GRU	93.4	91.9	95.0
	Mel-spectrogram + bi-LSTM	90.6	90.1	91.1
Korean reality	Spectrogram + CNN with <i>melCL</i> (proposed)	94.7	93.0	96.4
	Spectrogram + CNN	90.7	91.4	89.9
	Mel-spectrogram + CNN	93.5	91.1	95.9
	spectrogram + bi-GRU	90.6	84.9	97.2
	Mel-spectrogram + bi-GRU	92.3	88.5	87.8
	Mel-spectrogram + bi-LSTM	92.6	87.5	98.4
British 8 h	Spectrogram + CNN with <i>melCL</i> (proposed)	86.5	85.3	87.8
	Spectrogram + CNN	83.5	79.8	87.5
	Mel-spectrogram + CNN	86.8	83.3	90.5
	Spectrogram + bi-GRU	75.0	65.7	87.4
	Mel-spectrogram + bi-GRU	78.5	67.8	93.1
	Mel-spectrogram + bi-LSTM	80.5	72.5	90.5
Spanish 12 h	Spectrogram + CNN with <i>melCL</i> (proposed)	88.9	84.7	93.4
	Spectrogram + CNN	86.6	80.0	94.4
	Mel-spectrogram + CNN	80.9	70.6	94.6
	Spectrogram + bi-GRU	75.3	63.8	92.0
	Mel-spectrogram + bi-GRU	74.1	61.5	93.2
	Mel-spectrogram + bi-LSTM	75.6	63.4	93.6
MIREX 2015	Spectrogram + CNN with <i>melCL</i> (proposed)	95.3	99.4	91.6
	Spectrogram + CNN	93.8	98.8	89.3
	Mel-spectrogram + CNN	92.5	93.8	91.2
	Spectrogram + bi-GRU	92.8	94.9	90.8
	Mel-spectrogram + bi-GRU	94.3	92.3	96.4
	Mel-spectrogram + bi-LSTM	95.3	94.1	92.7
Dafx 07	Spectrogram + CNN with <i>melCL</i> (proposed)	84.9	84.0	85.9
	Spectrogram + CNN	84.4	77.7	92.3
	Mel-spectrogram + CNN	80.1	69.2	95.1
	Spectrogram + bi-GRU	68.4	57.5	84.5
	Mel-spectrogram + bi-GRU	69.0	53.3	98.0
	Mel-spectrogram + bi-LSTM	70.6	55.4	97.3

which indicates that Mel-scale filters are not always the best solution.

We calculated the frame-by-frame *F*-score, precision, and recall to verify the music detection performance. Table 3 shows the music detection performance of the proposed algorithm and baseline systems. Here, the number of bins for the *melCL* and Mel-spectrogram was 64, and the temporal dimension of the input of all systems was 101. The number of parameters to be studied

were about 29.47 million in the proposed system, 114.66 million for the first baseline system (using spectrogram + CNN), 29.46 million for the second baseline system (using Mel-spectrogram + CNN), 20.46 million for the third baseline system (using spectrogram + bi-GRU), 19.28 million for the fourth baseline system (using Mel-spectrogram + bi-GRU), and about 25.71 million for the fifth baseline system (using Mel-spectrogram + bi-LSTM), respectively. The *F*-score in the table shows that the

Table 4 Performance of the proposed algorithm by genre

Test data	Genre	<i>F</i> -score (%)	Precision (%)	Recall (%)
British 8 h	Documentary	81.6	75.5	88.8
	Drama	78.4	82.5	74.7
	Kids	96.1	95.5	96.7
	Reality	86.0	86.5	85.5
	Show	86.9	86.2	87.5
Spanish 12 h	Documentary	79.6	72.8	88.0
	Drama	88.9	82.3	96.5
	Kids	95.6	98.6	92.7
	News	22.8	13.1	89.8
	Show	92.3	91.5	93.1

proposed system was better than the baseline systems for all the data sets, except the British 8-h data set. The highest *F*-score in the British 8-h data set was 86.8% in the second baseline system, which is very similar to the 86.5% in the proposed system. The proposed system showed stable values of more than 84% in both precision and recall of all data sets. This means that the proposed system achieved a stable performance regardless of the type of data. The *F*-score of the proposed system showed the highest value of 95.9% on the Korea drama data. In the baseline systems, the Mel-spectrogram and spectrogram have different performance, depending on the data.

We measured the performance by genre to investigate the cause of performance degradation of the British and Spanish data among our collected data. Table 4 shows

the performance by genre of the British and Spanish data. Both data have low detection performance in drama and documentary. In particular, the news genre of the Spanish data showed very low precision, which is expected to be very small in the news genre. Interestingly, both data have the highest performance for the kids genre, and we assume that this is because the kids genre uses a lot of music and universal music, compared to other genres.

Next, we compared the proposed method with open source programs. We used open source of music/speech discrimination [16] and audio segmentation toolkit [15]. The two open-source sources are different from the data and learning methods we use for learning. Because both open sources do not allow overlap of music and speech, there is a limitation in performance comparison with our system. However, we performed this comparison experiment to verify that the performance of our algorithm was reliable. In order to compare the performance fairly, we measured the detection performance of the music segment without speech.

Table 5 shows the music detection performance of our proposed algorithm and other algorithms. The performance of the music segment without speech is indicated by “recall_nosp.” Because unlike our algorithm, the other algorithms [15, 16] do not allow overlap, we mainly compared the performance of precision and recall_nosp. The algorithm with the highest precision varied with the test data. This fact seems to result from the difference in the algorithm structure and the influence of the learning data (especially music). Nevertheless, our algorithm showed

Table 5 Performance of the proposed and other discrimination algorithms

Test data	Algorithm	<i>F</i> -score (%)	Precision (%)	Recall (%)	Recall_nosp (%)
Korean drama	Proposed	95.9	95.9	96.0	96.8
	Tsipas et al. [16]	77.9	97.0	65.1	74.6
	Doukhan et al. [15]	79.9	95.4	68.7	80.3
Korean reality	Proposed	94.7	93.0	96.4	97.5
	Tsipas et al. [16]	66.6	96.2	51.0	67.6
	Doukhan et al. [15]	68.0	96.4	52.5	75.6
British 8 h	Proposed	86.5	85.3	87.8	92.2
	Tsipas et al. [16]	67.8	83.2	57.3	72.2
	Doukhan et al. [15]	67.4	80.6	57.9	86.7
Spanish 12 h	Proposed	88.9	84.7	93.4	96.8
	Tsipas et al. [16]	71.1	91.8	58.0	76.4
	Doukhan et al. [15]	73.1	92.9	60.2	86.5
MIREX 2015	Proposed	95.3	99.4	91.6	92.5
	Tsipas et al. [16]	96.4	99.3	93.7	96.1
	Doukhan et al. [15]	95.4	99.3	91.9	94.1
Dafx 07	Proposed	87.6	88.3	87.0	87.0
	Tsipas et al. [16]	70.6	88.7	58.7	58.7
	Doukhan et al. [15]	65.7	87.9	52.4	52.4

the highest recall in most of these data (except MIREX 2015). We think that the reason is because we used music with noise as training data. We guess that the recall of our algorithm is somewhat low because of the large proportion of pure music in MIREX 2015 data.

5 Conclusion

In this paper, we propose a new method of music detection in broadcast content by using a convolutional layer with a Mel-scale kernel. Our proposed system is the CNN model with *melCL*, which is trained by mixed data with music, speech, and noise. To verify the performance, we developed a baseline system, collected the broadcast data in various languages of British English, Spanish, and Korean, and performed various music detection experiments. As a result, the proposed method showed better performance than the baseline system. For the Korean drama data set, it showed an *F*-score of 95.9%. In addition, the proposed method showed a higher performance than the other methods of music detection that used open sources. We also submitted our algorithm to the MIREX 2018 Challenge [4] and achieved the second-best result in the music detection task. However, we found that sometimes there are missed detections of music (percussive and traditional music) and false detection of noise (bells ringing). We will study robust music detection and music/non-music separation based on the proposed method. We also plan to extend our research to speech/non-speech detection.

Abbreviations

BIC: Bayesian information criterion; BPTT: Backpropagation through time; CFA: Continuous frequency activation; CNN: Convolutional neural network; DNN: Deep neural network; GLR: Generalized likelihood ratio; GMM: Gaussian mixture model; GRU: Gated recurrent unit; HMM: Hidden Markov model; KL: Kullback Leibler; LSTM: Long short-term memory; melCL: Convolutional layer with Mel-scale kernel; MFCC: Mel-frequency cepstral coefficient; MLP: Multilayer perceptron; ReLU: Rectified linear unit; SSLMs: Self-similarity lag matrices; STFT: Short time Fourier transform; SVM: Support vector machine; ZCA: Zero-phase component analysis; ZCR: Zero crossing rate

Acknowledgements

This research project was supported by the Ministry of Culture, Sports and Tourism (MCST) and from Korea Copyright Commission in 2018. [2018-micro-9500, Intelligent Micro-Identification Technology for Music and Video Monitoring].

Authors' contributions

BYJ and WHH carried out the numerical experiments and drafted the manuscript. BYJ, WHH, JHK, and OWK participated in the design of the study and helped to draft the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Electronics Engineering, Chungbuk National University, Cheongju, South Korea. ²ETRI (Electronics and Telecommunications Research Institute), Daejeon, South Korea.

Received: 31 October 2018 Accepted: 5 June 2019

Published online: 26 June 2019

References

1. B. Taras, C. Nadeu, Audio segmentation of broadcast news in the Albayzin-2010 evaluation: overview, results, and discussion. *EURASIP J. Audio Speech Music Proc.* **2011**(1), 1–10 (2011)
2. D. Castán, D. Tavarez, P. Lopez-Otero, J. Franco-Pedroso, H. Delgado, E. Navas, L. Docio-Fernández, D. Ramos, J. Serrano, A. Ortega, E. Lleida, Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains. *EURASIP J. Audio Speech Music Proc.* **2015**(33), 1–9 (2015)
3. Mirex 2015: music/speech classification and detection. http://www.music-ir.org/mirex/wiki/2015:Music/Speech_Classification_and_Detection Accessed 12 Sept 2018
4. Mirex 2018: music and/or speech detection. http://www.music-ir.org/mirex/wiki/2018:Music_and/or_Speech_Detection Accessed 4 Sept 2018 Accessed 12 Sept 2018
5. Schulz, H., & Fonollosa, J. A. (2009). A Catalan Broadcast Conversational Speech Database. *Proc. I Joint SIGIL/Microsoft Workshop on Speech and Language Technologies for Iberian Languages*
6. S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Proc. IEEE Trans. Acoust. Speech Signal Process.* **28**(4), 357–366 (1980)
7. N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification. *Proc IEEE Trans. Audio Speech Lang. Process.* **19**(4), 788–798 (2011)
8. Gtzan music speech dataset. http://marsyasweb.appspot.com/download/data_sets/ Accessed 12 Sept 2018
9. Marolt, M. (2015). Music/Speech Classification and Detection Submission for MIREX 2015. <https://www.music-ir.org/mirex/abstracts/2015/MM3.pdf>
10. Melendez-Catalan, B., Molina, E., & Gomez, E. (2018). Music and/or speech detection MIREX 2018 submission. <https://www.music-ir.org/mirex/abstracts/2018/MMG.pdf>
11. T. Theodorou, I. Mporas, N. Fakotakis, An overview of automatic audio segmentation. *Int J Inform Technol Comput Sci (IJITCS)* **6**(11), 1–9 (2014)
12. Grill, T., & Schluter, J. (2015). Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. *Proc. European Signal Processing Conference (EUSIPCO)*, pp. 1296–1300
13. Doukhan, D., & Carrive, J. (2017). Investigating the use of semi-supervised convolutional neural network models for speech/music classification and segmentation. *Proc. The Ninth International Conference on Advances in Multimedia (MMEDIA)*
14. A. Coates, A.Y. Ng, *Learning Feature Representations with k-means. Neural Networks: Tricks of the Trade* (2012), pp. 561–580
15. D. Doukhan, J. Carrive, F. Vallet, A. Larcher, S. Meignier, in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. An open-source speaker gender detection framework for monitoring gender equality (2018), pp. 5214–5218
16. N. Tsipas, L. Vrysis, C. Dimoulas, G. Papanikolaou, Efficient audio-driven multimedia indexing through similarity-based speech/music discrimination. *Multimedia Tools and Applications* **76**(24), 25603–25621 (2017) <https://github.com/nicktgr15/similarity-based-speech-music-discrimination> Accessed 12 Sept 2018
17. Tsipas, N., Vrysis, L., Dimoulas, C., & Papanikolaou, G. (2015). Mirex 2015: Methods for speech/music detection and classification. *Proc. Music information retrieval evaluation eXchange (MIREX)*
18. Seyerlehner, K., Pohle, T., Schedl, M., & Widmer, G. (2007). Automatic music detection in television productions. *Proc. the 10th International Conference on Digital Audio Effects (DAFx07)*
19. Example of music in television productions of the Austrian national broadcasting corporation, <http://www.cp.jku.at/people/seyerlehner/md.html> Accessed 12 Sept 2018
20. Wieser, E., Husinsky, M., & Seidl, M. (2014). Speech/music discrimination in a large database of radio broadcasts from the wild. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2134–2138
21. Choi, K., Fazekas, G., & Sandler, M. (2016). Automatic Tagging Using Deep Convolutional Neural Networks. *arXiv preprint arXiv:1606.00298*
22. A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, T. Weyde, *Singing Voice Separation with Deep U-Net Convolutional Networks* (2017)

23. Labrosa music-speech corpus. <http://www.ee.columbia.edu/~dpwe/sounds/musp/> Accessed 12 Sept 2018
24. G.R. Arce, *Nonlinear Signal Processing: A Statistical Approach* (Wiley, 2005)
25. Snyder, D., Chen, G., & Povey, D. (2015). Musan: A Music, Speech, and Noise Corpus. Arxiv preprint arXiv:1510.08484
26. K.J. Piczak, *ESC: Dataset for Environmental Sound Classification*. Proc. ACM International Conference on Multimedia (2015), pp. 1015–1018
27. P. Boersma, Praat, a system for doing phonetics by computer. *Glott International*, 5 (2002)
28. K. Cho, B. Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, in *Proc. Empirical Methods in Natural Language Processing (EMNLP)*. Learning phrase representations using RNN encoder-decoder for statistical machine translation (2014), pp. 1724–1734
29. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput* 9(8), 1735–1780 (1997)

6 Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
