Audio Event Detection and classification using extended R-FCN Approach

Kaiwu Wang, Liping Yang, Bin Yang

Key Laboratory of Optoelectronic Technology and Systems(Chongqing University), Ministry of Education,
ChongQing University, China
{amsturdy, yanglp, Braun}@cqu.edu.cn

ABSTRACT

In this study, we present a new audio event detection and classification approach based on R-FCN—a state-of-the-art fully convolutional network framework for visual object detection. Spectrogram features of audio signals are used as the input of the approach. The proposed approach consists of two stages like R-FCN network. In the first stage, we detect whether there are audio events by sliding convolutional kernel in time axis, and then proposals which possibly contain audio events are generated by RPN (Region Proposal Networks). In the second stage, time and frequency domain information are integrated to classify these proposals and refine their boundaries. Our approach can output the positions of audio events directly which can input a two-dimensional representation of arbitrary length sound without any size regularization.

Index Terms—*audio event detection, Convolutional Neural Network, spectrogram feature*

1. INTRODUCTION

Intelligent surveillance system is becoming increasingly ubiquitous in our living environment. At present, most of the video-based surveillance system is lack of robustness and reliability in practical application. For example, video-based surveillance doesn't work well in some specific scenarios, such as the night or cloudy circumstances. Audio surveillance has been alone or in combination with video surveillance to solve this problem. The work in [1] described a framework for scene analysis in a typical surveillance scenario through integrating audio and visual information. Generally, audio stream is much less onerous than video stream and the audio devices are more inexpensive. Audio events detection has been one of the important components to intelligent surveillance of security.

Unlike the static or changing slowly backgrounds in video surveillance backgrounds, there may be some impulsive sounds in audio backgrounds. Moreover, the audio signal is more versatile when audio events are superimposed on one or more backgrounds with different signal-to-noise ratio (SNR). Thus, it is a challenge work to detect audio events correctly from an audio segment.

Early works mainly concentrated on extracting different types of hand-crafted features, and training effective classifiers for recognition with traditional machine learning algorithms. The most typical classifier is Support Vector Machines (SVM). For instance, a method aimed at recognizing environmental sounds for surveillance and security applications is presented in [2], which applies one-class support vector machines together with a sophisticated measure. Another approach is to utilize a Gaussian Mixture Model (GMM) for sounds recognition. The proposed approach in [3] first classifies a given audio frame into vocal and non-vocal events, and then performs further classification into normal and target events using GMM. The non-stationary (time-frequency) techniques are applied to sound classification and produce a good result. In [4], Jonathan Dennis et al. extract image feature from the SPD image—a novel two-dimensional representation that characterizes the spectral power distribution over time in each frequency sub-band. In [5], [6], features are extracted from the spectrogram image of sound signals for automatic sound recognition. More recently, methods based on Deep Neural Networks (DNNs) have achieved good performance for sound event classification and detection. In [12], the authors outline a sound event classification framework that compares auditory image front end features with spectrogram image-based front end features, using deep neural network classifiers. The work in [13] employs an ensemble learning framework—a stack of ensemble classifiers named multi-resolution stacking (MRS). A concatenation of lower building blocks' predictions and the expansion of the raw acoustic feature is fed into each classifier in MRS. The lower building blocks describe a base classifier in MRS, named boosted deep neural network. In [14], the authors record a database of sounds occurring in subway trains in real conditions of exploitation and use DNNs to classify the sounds into screams, shouts and other categories. In these audio events detection methods, Deep Neural Networks (DNNs) mostly work as a classifier which achieve better performance than other traditional classifiers, but can't directly detect audio events in real-time when the audio events occur.

In this paper, we present a new audio event detection and classification approach by extending the R-FCN framework [11] which is a fully convolutional neural network used in visual object detection. Inspired by the framework, the proposed approach also consists of three parts (not include the extracting of feature maps). The first part is RPN Network [10] for generating a list of proposals which possibly contain audio events. However, unlike the proposals with different widths and heights in [11], the proposals in our extended R-FCN framework have the same height and just vary in width. We do this just because the sound is one dimension signal. The second part refines the boundaries of above proposals. The third part classifies above proposals into audio events or backgrounds. In the second and third parts, we not only use the information in time domain but also use the information in different frequency ranges for classification and boundary regression. Spectrogram features of audio signals are

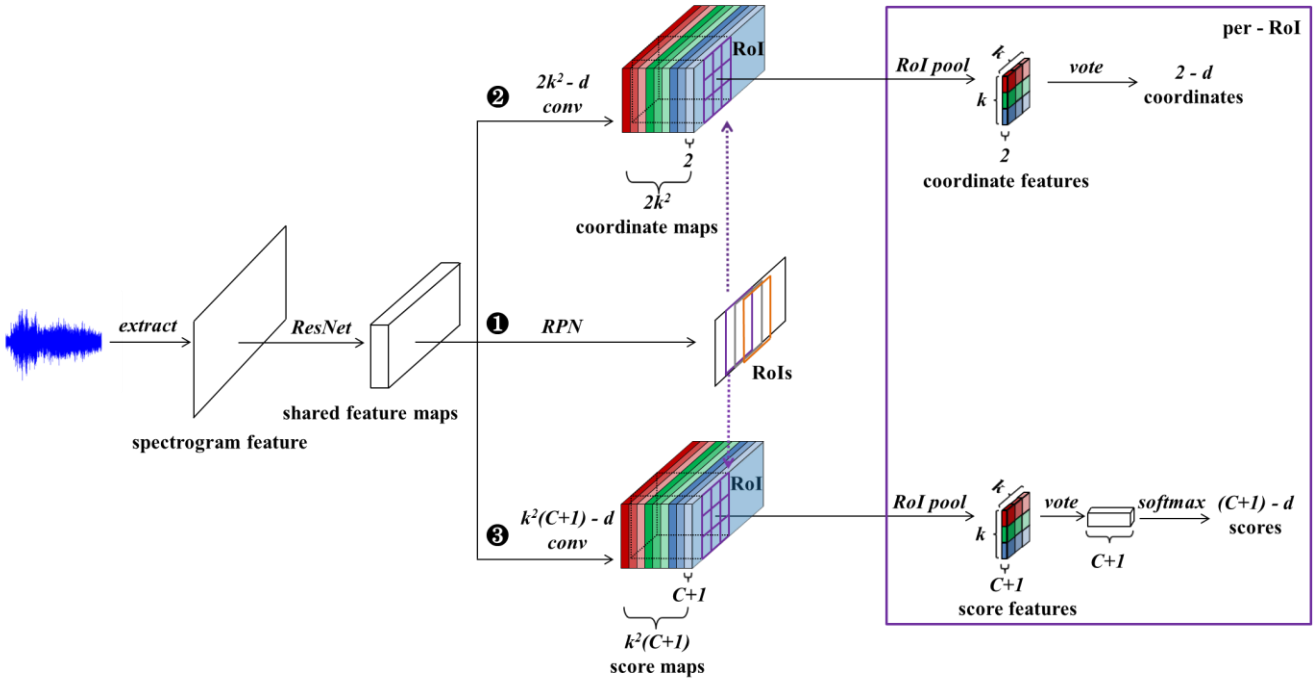

Figure 1: Overview of the system architecture for audio detection and classification

fed into the architecture as inputs. Based on the capacity of ResNet network [8] for extracting features and solving the gradient degradation problem when training a deep neural network, feature maps are produced by the modified ResNet50 network fed with these spectrogram feature. These feature maps are shared by above three parts.

The rest of this paper is organized as follows. Section 2 presents the proposed deep learning approach for audio detection and classification. Section 3 discusses our experiments and results. Section 4 concludes this work.

2. PROPOSED APPROACH

Our approach follows the deep learning framework of R-FCN which is a fully convolutional neural network for visual object detection [11]. We improve the Region Proposal Network (RPN) to make the framework suitable for audio detection. To make the overall system more simply and generate real-time result, we substitute ResNet50 for ResNet101, and modify ResNet50 network. The overall system architecture essentially consists of two stages through three parts in Figure 1. Firstly, we detect the approximate position (proposal) of audio event and don't care the audio event class by our first part. So, this can be treated as a binary classification problem. Secondly, we classify the proposal into audio events or background and refine the boundaries in our second and third parts. These proposals are also called as regions of interest (RoIs).

2.1. The Modified ResNet50 Network

Based on the capacity of ResNet50 [8] network for extracting features and solving the degradation problem when training a deep neural network, we extract the shared feature maps from spectrogram feature by the modified ResNet50 network. Experiments show that the performance will be reduced because of the reduction of feature maps' resolution. So, we remove the last *fc* layer, pooling layer and three building blocks in ResNet50. The modified ResNet50 begins with a 7×7 convolutional layer which has 64 kernels and a stride of 2, then follows a 3×3 max pooling layer with a stride of 2, the rest of the modified ResNet50 are series of building blocks. The last convolutional layer has 1024 kernels, so the generated shared feature maps has 1024 channels.

2.2. The Improved RPN Network

For audio events detection, the improved RPN network outputs the start time, the length of proposals with scores estimating the probability that these proposals belong to audio events or background. We model this process followed a fully convolutional network [10]. The shared feature maps produced by the modified ResNet50 are fed into RPN network.

A strip window slides over the shared feature maps to generate m base regions for proposals at each position on the time axis of feature map. These m base regions are different in length but start at the same time of audio signal corresponded to the current position of sliding window. The sliding window has a size of $n \times 1$, n is the height of the shared feature maps. At each position of sliding window, 256-dimensional vector representing these m base regions simultaneously is generated, which is used to produce scores and coordinates of proposals based on these m base regions. The coordinates represent the difference of start time and length between a base region and proposal. The scores estimate the probability that these m proposals belong to audio

events or background. So, *2m* scores and *2m* coordinates are generated for each position of sliding window. In another word, we generate *m* base regions at every few frames of audio signal, and extract 256-d feature vector from front part of these *m* base regions to represent all *m* base regions simultaneously. Then, the 256-d feature vector is used for generating scores and refining the position.

Taking inspiration from the character of convolutional layer, the improved RPN network is implemented by three normal convolutional layers. The 256-d feature vector is produced by a $n \times 1$ convolutional layer which has 256 kernels and works as a sliding window. Then the 256-d vector is simultaneously fed into two sibling 1×1 convolutional layers—a classification layer (*cls* layer) and a regression layer (*reg* layer). The *cls* layer and *reg* layer have *2m* convolutional kernels respectively. The *cls* layer outputs *2* scores that estimate the probability that the proposal belongs to audio events or background. So the *cls* layer has *2m* outputs. The *reg* layer has *2m* outputs representing the difference between base regions and proposals in start time and length. The key ideal of RPN network for audio detection is illustrated in the Figure 2. We use *m* lines with different lengths represent the *m* base regions in the Figure 2.

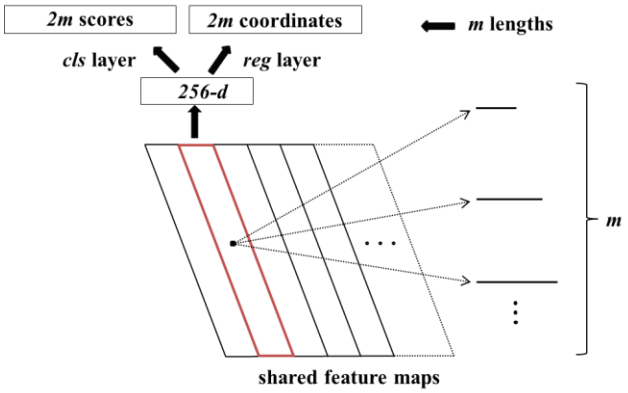

Figure 2: Key ideal of RPN network for audio detection

We use two 2-d vectors (B_s, B_l) and (G_s, G_l) denote the position of a base region and ground truth respectively. We are more concerned with the time that audio event occurs, so the 2-d position vector denotes the start and length of an audio event. We need to define an operation *F* to produce position vector of proposal from a base region:

$$(P_s, P_l) = F(B_s, B_l) \quad (1)$$

Here, (P_s, P_l) denote the position vector of a proposal. When the base region near to the ground truth, we take shift and scale transformation into consideration from a base region. The operation *F* can be simply defined as:

$$P_s = B_l * t_s + B_s, \qquad P_l = B_l * exp(t_l) \quad (2)$$

t_s ,t_l is the shift factor and scale factor need to be learned, and they are the *reg* layer's output for a base region. So, the corresponding labels are defined as:

$$t_s^* = (G_s - B_s)/B_l, \qquad t_l^* = \log(G_l / B_l) \quad (3)$$

When training the improved RPN network, our loss function for audio detection followed the multi-task loss in [10] which can solve the classfication and coordinates refining simultaneously, defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{2}\sum_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{2}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (4)$$

Here, *i* is the index of base regions, P_i represents the probability that base region *i* is predicted as an audio event or background. We set the balance weight λ=1. The classification loss L_{cls} is log loss over two classes (audio event or background). The ground-truth label P_i^* is 1 if the base region *i* is an audio event, otherwise is 0. This means the regression loss L_{reg} (smooth L_1 loss function, defined in [9]) is activated only for the base region near to ground truth. t_i is a two-dimensional vector (t_s ,t_l) for base region *i*, which is produced by *reg* layer. t_i^* is a two-dimensional vector (t_s^* ,t_l^*) for base region *i*.

2.3. The classification and refining of proposals

After the first stage, we select 6000 proposals which are most likely to be audio events from the translations of all base regions. We do this by checking their probability to be audio event because the first stage is a binary problem. In the second stage, $k^2(C+1)$-channel score maps are produced by a convolutional layer which has $k^2(C+1)$ kernels and $2k^2$-channel coordinate maps are produced by a convolutional layer which has $2k^2$ kernels as seen in Figure 1. *C* is the categories of audio events (+1 for background), *k* equals 3, but which is a hyper parameter related to the parameter of the following *RoI pooling* layer. Given the 6000 proposals (RoIs), each RoI is divided into $k \times k$ parts, and $k \times k$ score / coordinate features are produced through *RoI pooling* layer from score / coordinate maps respectively. The details of the *RoI pooling* layer defined in [11]. After the *RoI pooling* layer, a vote principle (implemented by an average pooling layer) is applied to every channel of scores / coordinates features. Then the *2*-d coordinates are generated, and the *(C+1)*-dimensional scores for each category are generated after a *softmax* layer.

In the score or coordinate maps, the three red, green, blue maps represent the high, middle, low frequency components respectively, and each of these three maps in every color (red, green, blue) maps represents sequentially one third segment of a proposal in time axis correspond to the color depth, the deepest color score map represents first segment of a proposal and the lightest color score map represents the last segment. Each component of score / coordinate features is from average pooling on only one of $k \times k$ score / coordinate maps correspond to their same color. This can be understood that not only components at different time periods but also the different frequency components at same time periods have different response to the classifying or refining of a RoI. For example, there are three RoIs which have same size and one third overlap as showed in Figure 3. However, the same one third part has different response to the classifying or refining of these three RoIs showed by the lighter of every color. Moreover, different frequency components have different response to the classifying or refining of each RoI showed by the different color (red, green, blue). What's more, there are always $k \times k$ score / coordinate features regardless of the size of RoI. In another word, the sound with arbitrary length can be processed.

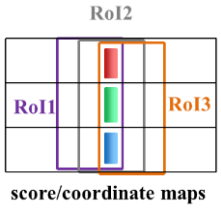

Figure 3: Illustration of the *RoI pooling* layer's function

3. EXPERIMENTS

We perform experiments on the dataset of IEEE DCASE Challenge 2017 Task 2, which consists of isolated sound events for three target class (babycry, glassbreak, gunshot) and recordings of everyday acoustic scenes to serve as background. The background audio material consists of recordings from 15 different audio scenes, and is part of TUT Acoustic Scenes 2016 dataset [7]. We regularize audio signals to same sample frequency of 44.1 KHz and synthesize our training data according to the event-to-background ratio (EBR, -6.0, 0, 6.0). Our training data has 15000 mixtures (5000 per target class, each mixture only contains a target class event). Then we generate the grayscale spectrogram through short-time Fast Fourier transform. A hamming window with a length of 1024 samples and an overlap of 441 samples is applied. The generated spectrogram feature has a height of 512. We evaluate our approach on the development dataset. The mainly evaluation metrics are Error Rate (ER) and F-Score (F) calculated using event-based onset-only condition with a collar of 500*ms* [15].

Table 1: The results for each class on development dataset

	Class	*F* (%)	*ER*	*DR*	*IR*
Baseline	babycry	69.5	0.73	0.18	0.55
	glassbreak	88.1	0.23	0.17	0.06
	gunshot	51.2	0.85	0.56	0.29
	All	70.0	0.60	0.30	0.30
Our approach	babycry	97.2	0.06	0.03	0.03
	glassbreak	94.6	0.10	0.09	0.00
	gunshot	81.4	0.32	0.28	0.04
	All	91.4	0.16	0.13	0.02

Table 1 shows the result on Event-based overall metrics for each class on development dataset compared with Baseline system for DCASE Challenge 2017 Task 2. *DR*, *IR* are the Deletion Rate, Insertion Rate respectively [15]. The implementation of Baseline is based on a multilayer perceptron architecture (MLP) and uses log mel-band energies as features. The features are calculated in frames of 40 ms with a 50% overlap, using 40 mel bands covering the frequency range 0 to 22050 Hz. The feature vector was constructed using a 5-frame context, resulting in a feature vector length of 200. The MLP consists of two dense layers of 50 hidden units each, with 20% dropout. For each of the target classes, there is a separate binary classifier with one output neuron with sigmoid activation, indicating the activity of the target class. Our approach has an *ER* value of 0.16, F-Scores of 91.4% for all classes, which outperforms the Baseline system (*ER*: 0.60, *F*: 70.0%). The *ER* values of babycry and gunshot are improved by 0.67 and 0.53 respectively. It shows great improvement for babycry and gunshot.

The result on Event-based overall metrics for each EBR on development dataset is shown in Table 2. F-Scores and *ER* for all EBRs (not includes the mixture which doesn't contain target event) are 91.7% and 0.15. The difference of *ER* value between each *EBR* is lower than that of Baseline system, which proves our approach is more robust to noise. Our approach has lower *IR* value for each class and *EBR*, which indicates our approach is more reliable.

Table 2: The results for each EBR on development dataset

	Class	*F* (%)	*ER*	*DR*	*IR*
Baseline	-6.0	60.0	0.72	0.45	0.27
	0	77.6	0.43	0.24	0.19
	6.0	82.7	0.34	0.20	0.14
	All	73.5	0.50	0.30	0.20
Our approach	-6.0	86.8	0.23	0.20	0.02
	0	93.3	0.13	0.12	0.01
	6	95.2	0.09	0.08	0.01
	All	91.7	0.15	0.13	0.01

On the evaluation dataset, F-score and ER on segment-based metrics are 0.3173 and 82.0%, respectively. Our approach generally produced better result because of the two stage strategy. After the first stage, there will be a global summary for an audio event compared with Baseline's strategy. The 5-frame context of Baseline system only has a local summary for an audio event. At the same time, the using of deeper network and the integration of time and frequency domain information is another reason for the better performance.

4. CONCLUSION

We present an audio event detection and classification approach which can directly output the positions of audio events from an audio signals with arbitrary length. We improved RPN network for generating proposals which possibly contain audio events. Time and frequency domain information are fully utilized to classify these proposals and refine their boundaries.

5. REFERENCES

[1] Cristani M, Bicego M, Murino V, et al. Audio-Visual Event Recognition in Surveillance Video Sequences[J]. IEEE Transactions on Multimedia, 2007, 9(2): 257-267.

[2] Rabaoui A, Davy M, Rossignol S, et al. Using One-Class SVMs and Wavelets for Audio Surveillance[J]. IEEE Transactions on Information Forensics and Security, 2008, 3(4): 763-775.

[3] Atrey P K, Maddage N C, Kankanhalli M S, et al. Audio Based Event Detection for Multimedia Surveillance[C]. international conference on acoustics, speech, and signal processing, 2006: 813-816.

[4] Dennis J, Tran H D, Chng E S, et al. Image Feature Representation of the Subband Power Distribution for Robust Sound Event Classification[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2013, 21(2): 367-377.

[5] Sharan R V, Moir T J. Noise robust audio surveillance using reduced spectrogram image feature and one-against-all SVM[J]. Neurocomputing, 2015: 90-99.

[6] Sharan R V, Moir T J. Subband Time-Frequency Image Texture Features for Robust Audio Surveillance[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2605-2615.

[7] Mesaros A, Heittola T, Virtanen T. TUT database for acoustic scene classification and sound event detection[C]// European Signal Processing Conference. 2016:1128-1132.

[8] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. computer vision and pattern recognition, 2015: 770-778.

[9] Girshick R. Fast R-CNN[C]. international conference on computer vision, 2015: 1440-1448.

[10] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015: 1-1.

[11] Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[C]. neural information processing systems, 2016: 379-387.

[12] Mcloughlin I V, Zhang H, Xie Z, et al. Robust sound event classification using deep neural networks[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2015, 23(3): 540-552.

[13] Zhang X, Wang D. Boosting contextual information for deep neural network based voice activity detection[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2016, 24(2): 252-264.

[14] Laffitte P, Sodoyer D, Tatkeu C, et al. Deep neural networks for automatic detection of screams and shouted speech in subway trains[C]. international conference on acoustics, speech, and signal processing, 2016: 6460-6464.

[15] Mesaros A, Heittola T, Virtanen T. Metrics for polyphonic sound event detection[J]. Applied Sciences, 2016, 6(6): 162.

[16] Waibel A, Hanazawa T, Hinton G, et al. Phoneme recognition using time-delay neural networks[J]. IEEE transactions on acoustics, speech, and signal processing, 1989, 37(3): 328-339.