SOUND EVENT DETECTION IN REAL-LIFE AUDIO

Dmitrii Ubskii, Alexei Pugachev

ITMO University
Chair of Speech Information Systems, ul. Krasutskogo, 4
St. Petersburg, 196084, Russia
{ubskiy, pugachev}@speechpro.com

ABSTRACT

In this paper, an acoustic event detection system is proposed. This system uses fusion of several classifiers (GMM, DNN, LSTM) using another classifier (DNN) in attempt to achieve better results.

The proposed system yields F1 score of up to 21% for indoors subset of the provided data and up to 44% for outdoors subset.

Index Terms— acoustic event detection, neural networks, long short-term memory

1. INTRODUCTION

Acoustic event detection is an application of pattern recognition and machine learning in which an audio signal is mapped to corresponding sound events present in the auditory scene. Automatic audio event detection is utilized in a host of applications, including surveillance, speech detection and audio segmentation. This task is also particularly challenging because it involves multi-label classification.

Most conventional approaches to multi-label classification involve a set of one-vs.-rest binary classifiers, one for each label, results of which are then combined, or problem transformation to a single-label classification over the power set of original classes [1]. These approaches do not scale well with increasing number of classes, however, as each new class significantly increases training time and memory requirements.

In this contribution, a system is proposed which uses regression to calculate scores for each possible class for a sample. Proposed solution to the problem of scaling is using a single classifier for all classes that outputs prediction scores for each class. Addition of a new class in that case entails only minor adjustments of the system.

2. SYSTEM OVERVIEW

This section explains approach to the sound event detection used in this system. The approach is based largely on paper by Emre Cakir et al. [2] which uses a deep neural network for similar task.

All neural networks are implemented using Computational Network Toolkit by Microsoft Research [4].

2.1. Feature extraction

As a pre-processing step for the feature extraction, the recordings are divided into frames with 40 ms duration and 50% overlap. The MFCCs are then computed, as well as differential (Δ) and acceleration ($\Delta\Delta$) coefficients. Zeroth MFCC coefficient is discarded.

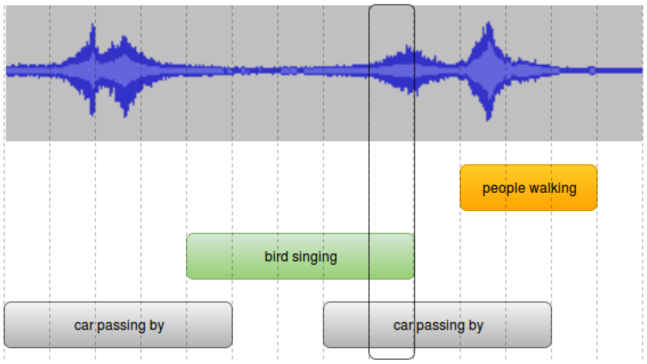

Figure 1: Overlapping sound events in a recording. Highlighted is an example frame in which only *bird singing* and *car passing by* events are present.

For each time frame a target output vector is obtained. Each element of the vector is a binary variable encoding whether the event is present in a given time frame. An illustrative example of such encoding is presented in Figure 1, where the target output vector for highlighted frame is [0 1 1].

2.2. Neural networks

In order to achieve better results, outputs of several classifiers are combined and fed to another classifier. The classifiers used in this case are baseline GMM and two neural networks with different architectures described further.

Both neural networks described in this section use Kullback-Leibler (KL) divergence [3] as their loss function, which for binary outputs is calculated as

$$\begin{aligned} KL(\mathbf{y}_t||\hat{\mathbf{y}}_t) = \sum_{i=1}^{N} \mathbf{y}_t(i)\log\hat{\mathbf{y}}_t(i) \\ + (1-\mathbf{y}_t(i))\log(1-\hat{\mathbf{y}}_t(i)), \end{aligned} \tag{1}$$

where $\mathbf{y}_t(i)$ is the target output for i^{th} event, $\hat{\mathbf{y}}_t(i)$ is the prediction for i^{th} event, and N is the total number of event classes.

KL divergence can be seen as an approximation of segment-based error rate (described in section 4) with first element of the sum representing deletions, and the second element representing insertions.

Both networks were trained using 4-fold cross-validation as used in the baseline system.

2.2.1. *DNN*

The first subsystem trained is a deep neural network with two hidden layers, 800 rectified linear neurons each, a bottleneck layer with 80 rectified linear neurons for feature extraction, and a sigmoid output layer. Frame concatenation method is used in order to provide context for the network, 15 frames before and after the target frame.

2.2.2. *LSTM*

Second network is a long short-term memory recurrent neural network with a single bottleneck sigmoid layer for feature extraction and a sigmoid output layer. This network tries to classify each run of 30 frames, outputting a prediction vector for the last frame of the run.

2.3. Fusion DNN

Outputs of bottleneck layers of two networks described previously are concatenated with outputs of GMMs and fed into the final DNN. This DNN uses KL divergence as its loss function as well.

This final network has three hidden layers, 800 rectified linear neurons each, and a sigmoid output layer. The outputs are then cut off using threshold determined at training phase, such that all values above or equal to the threshold are considered to be 1, and the rest are 0.

3. POST-PROCESSING

Many events contain intermittent periods that do not possess the same cepstral characteristics the rest of the frames with such label do, e.g. pauses betwen steps. The annotation of the audio material is done with a coarse time resolution, and since the system uses very short time frames, these pauses cause some abrupt changes in the output predictions of the system.

In order to smoothen the outputs in the testing stage, a median filter is applied to the results of the fusion DNN. Filtering is done using 11-frame window centered at the frame that is being processed.

4. EVALUATION

Evaluation considers two segment-based metrics: error rate (ER) and F1 score (F1). These metrics use segments of one second length to compare the ground truth and the system output.

ER is calculated based on the total number of insertions (I), deletions (D) and substitutions (S):

$$ER = \frac{\sum S(k) + \sum D(k) + \sum I(k)}{\sum N(k)} \tag{2}$$

F1 is calculated based on the total number of false positives (FP), false negatives (FN) and true positives (TP):

$$F1 = \frac{2 \cdot \sum TP(k)}{2 \cdot \sum TP(k) + \sum FP(k) + \sum FN(k)} \tag{3}$$

Evaluation of the system on the data as per 4-fold approach produced F1 of up to 21% on the indoors subset (*home*) and up to 44% on outdoors subset (*residential_area*). ER, however, is in the range of 0.9–1.0 for *residential_area* and 2.0–3.0 for *home*.

5. CONCLUSION

The system proposed in this paper outperforms baseline system by 5% on the indoors subset and 12.5% on the outdoors subset of the provided data when scored using F1. The system also scales considerably better with addition of new classes.

6. REFERENCES

[1] G. Tsoumakas, I. Katakis, “Multi-Label Classification: An Overview,” in *Int J Data Warehousing and Mining*, 2007, pp. 1–13.

[2] E. Cakir, T. Heittola, H. Huttunen, T. Virtanen, “Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks,” in *IJCNN*, 2015, pp. 1–7.

[3] S. Kullback and R.A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, 1951, pp. 79–86.

[4] A. Agarwal et al., “An Introduction to Computational Networks and the Computational Network Toolkit”, *Microsoft Technical Report MSR-TR-2014-112*, 2014.