# Development of an non-speech audio event detection system

Sergey A. Romanov[1], Nikolai A. Kharkovchuk,
Maxim R. Sinelnikov, Mikhail R. Abrash
Saint Petersburg Electrotechnical University "LETI"
St. Petersburg, Russia
[1]saromanov@etu.ru

Vladislav Filinkov
ID R&D Inc.
New York, USA
filinkov@idrnd.net

***Abstract*—This paper describes a process of acoustic event detection system development. It includes sound gathering and preprocessing, detector training and testing. The system can recognize thirteen types of sound events, such as baby crying, a person screaming or asking for help and so on. It has extremely low false positive rate and can be used for automated continuous monitoring.**

***Keywords*—audio event detection; deep learning; audio dataset; data collection**

## I. Introduction

Hearing is one of the most important tools to understand, what is going on beside you. However, unfortunately, some people are deprived of such opportunity because of certain factors. They get used to live in this way, but in some cases, lack of hearing can be life-dangerous. To avoid such situations, special devices are developed, that are specialized for detecting some set of sound events and sending a notification to user if some of this events happened. What is described above is one of the various applications, where sound event detection and classification can be used.

As mentioned before, one of the applications of sound event detection is to help deaf people. That is why special 13 classes were considered: baby cry, cough, dog bark, glass breaking, word “help”, gunshot, moan, scream, siren, sneezing, snoring, toilet flush and water running. Such events as sneezing and cough are important for deaf parents: if their children are ill, they will know it immediately. Class “help” is important for all people. If person needs a help, but he cannot move, he can say or even whisper word “help” that will be detected by special sensors and notification about this event will be sent, for example, to his relatives or his friends. These classes are very important in everyday’s life that it is why we spent a bit amount of time collecting data of these classes.

Sound event tagging and detection gained huge popularity in recent time. Many competitions as DCASE are hold every year to find best solutions for these tasks. With the rise of deep-learning in 2012[1], neural networks, learnt from the massive amount of data, outperformed many classical approaches in signal processing. Neural networks showed astonishing results in a field of computer vision[2], but recently gained popularity in natural language and audio processing. Deep learning is successfully applied to tagging and detection problems and will provide further breakthrough in this area.

As for any machine-learning problem, the appropriate amount of qualified data is a key component for building a high performing model. Therefore, we will pay close attention to the way we collected our data, what sources we used and what difficulties faced.

Also we discuss two approaches that were used for classification and how they can be applied for tagging and detection tasks. In the end we estimate and compare models we used.

## II. Dataset collection

The process of collecting datasets can be divided into three parts: searching ready datasets, gathering new sounds and sounds preparation.

### *A. Data gathering*

There are many datasets for sound event detection, see, for example [3]. Many of them were used accordingly to the classes, interesting in our problem. Basic dataset, used for training, was Audioset [4]. Each file in this dataset is ten seconds length. However, for some of the classes the number of files that Audioset provides is very small, so we had to look for many other datasets and in some cases, even to generate files by ourselves.

The main difficulty was marking data by classes, since many datasets contain fairly long audio files and each of them contains several classes of interest for us. In addition to that, the recordings in the datasets were from different distribution: in some of them events were clean, without extraneous noise, whereas in others sounds were recorded in noisy environment , sometimes this noise completely overlapped with our event, and such sounds is very difficult for neural network to determine correctly. Datasets do not contain two events at the same time, such as: shot in glass, siren with scream, etc.

Class “help” needs to be separately mentioned. This class belongs to speech recognition classes, so we had to search for it separately from other classes. Five hundred records of class “help” were generated artificially with different accents.

978-1-7281-5761-0/20/$31.00 ©2020 IEEE

### B. *Sounds preparation*

The first trial neural network showed necessity to improve dataset: accuracy was low. We listened to all the sounds, deleted strongly noisy recordings and .cut sounds not related to the target event. After that, we get dataset for 13 classes with more than 1000 samples per class. The statistic for our dataset is shown in table 1.

For purposes of comparison with the existing dataset, provided us by “IDR&D” company, we trained our model for 9 classes, exclude water running, toilet flush, snoring, glass breaking and gunshot.

TABLE I. DATASET STATISTIC

| Class | Number of sounds | Total duration |
|---|---|---|
| baby cry | 1112 | 7474.9 |
| cough | 1243 | 6423.5 |
| dog bark | 1171 | 3695.7 |
| glass breaking | 1010 | 1909.3 |
| word “help” | 1000 | 1512.3 |
| gunshot | 2034 | 13401.8 |
| moan | 1000 | 1662.9 |
| scream | 1010 | 3033.8 |
| siren | 1136 | 5210.8 |
| sneezing | 1000 | 6947.2 |
| snoring | 1412 | 12522.9 |
| toilet flush | 1002 | 8303.6 |
| water running | 1001 | 7926.9 |

## III. TRAINING

### A. *Method*

We train out network to classify nine different events. It is multiclass classification, where each sample is assigned to only one label. We provided different approaches for classification: one-to all method, where for each class we train separate binary classifier and treat other classes, different from the original, as noise. Another approach is to train one neural network for classifying between all classes simultaneously. First approach will be applied to classification task, whereas second approach to tagging. This classification model will be very important component in tagging task we are going to discuss lately.

### B. *Feature Extraction*

First we extract features we will need to train our model. For each audio sample we compute fast Fourier transformation with 25ms window and 10ms hop length, at a 16 kHz sample rate. Each sound file now is represented as log-mel spectogram. Such representation gives us huge advantage to use convolutional neural networks, that showed state-of-art performance in many computer vision tasks,, as we treat spectogram to be 1-channel image.

### C. *Architectures*

In both models we used very similar architecture, the only difference is in the number of output units and loss function. We tried different architectures as Mobilenet, Resnet and others, but, as we did not have a huge amount of data, we decided, that it will be reasonable to use pretrained network as feature extraction at first layers, so we chose VGGish model, trained on Audioset by Google. We take activation after last convolutional layer, use global average pooling to reduce dimensionality and pass it to last fully-connected layers with 512 and 128 units respectively. For each fully-connected layer regularization techniques are used: dropout with 0.5 probability of dropping any unit and l2 with labmda parameter equal to 0.001 to avoid overfitting. Regularization is crucial component for this model: it prevents high variance, that occurs if model is trained without regularization. For one-to-all approach we use single output unit with sigmoid activation, showing probability,that current input belongs to labeled class. As for the other approach, the number of output units is equal to the number of classes are used + 1( “ + 1” characterizes noisy class), without nonlinear activation. We tried to use softmax activation on last layer, that is often used for multiclassification, but it showed worse results, so we decided to get rid of it. In the other parts of the network RELU nonlinear activation is used.

In both cases Adam optimizer is used with learning rate starting at 0.001 and decreasing throughout the epochs. The number of epochs, however, is different. For binary classifier it is equal to 15, for multiclass classifier to 50.

The final model, based on VGGish, converges much faster, than other models that we tried. It confirms, that pretrained networks are very powerful tools in deep-learning community now and are very suitable to be used.

Loss function is different for two approaches. For one-to-all method weighted binary-crossentropy is used, as we try to classify one class versus others and our dataset seems to be imbalanced in this situation, so we want to penalize more for false negative rather than for false positive. For multiclass classification mean squared error is used, as we do not use any nonlinear activation the last layer.

Particular difficulty appeared with some rare classes such as:moan, snoring, sneezing. This classes very similar and can be confused. In addition, these classes are very diverse since these parameters are individual for each person and present a certain difficulty for detection and classification.

## IV. RESULTS

We decided to split this section into two parts: classification results and tagging results. In the first part we give evaluation for classification. In the second part we discuss how classification approaches can be applied to tagging task.

### D. *Classification results*

The results we describe here refer to multiclass classification approach. Metric that we are going to rely on is F1-score. Its formula is shown below.

$$\text{F1} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad ,$$

where Precision and Recall are two other popular metrics for classification. Their formulas are shown below.

$$\text{Precision} = \frac{TP}{TP+FP}, \text{Recall} = \frac{TP}{TP+FN}$$

where TP denotes true positive examples, FN denotes false negative examples and FP denotes false positive examples.

Table 2 below shows F1, precision and recall metrics for all nine classes.

| Class | Precision | Recall | F1 | Number of files |
|---|---|---|---|---|
| Baby_cry | 0.829 | 0.797 | 0.813 | 79 |
| Cough | 0.966 | 0.86 | 0.91 | 100 |
| Dog_bark | 0.888 | 0.87 | 0.879 | 100 |
| Glass | 0.875 | 0.955 | 0.913 | 44 |
| Help | 0.931 | 0.931 | 0.931 | 87 |
| Moan | 0.944 | 0.966 | 0.955 | 87 |
| Noise | 0.992 | 1 | 0.996 | 128 |
| Scream | 0.843 | 0.864 | 0.854 | 81 |
| Siren | 1 | 0.964 | 0.982 | 56 |
| Sneeze | 0.844 | 0.92 | 0.88 | 100 |

Table 2 Evaluation

There are some pair of “problematic” classes. By word “problematic” we mean classes that give big number of false positive examples between each other. Pair of “baby cry” and “scream” gives the worst results. It is not miraculous because in some situations it is difficult even for man to differ these two classes. Another pair is “cough” and “sneeze” and they sound pretty similar too. For other classes our network predicts pretty well considering the amount of data we used.

### A. *Tagging results*

For tagging we use our binary classifiers that we trained for each class. We divide our sample on frames and classify each frame separately by all trained classifiers. We take event with highest output probability and compare it with threshold(in out case it is equal to 0.5 but it can be different according to the situation. If the probability is more than the threshold, then we suppose that in this frame certain event happened, otherwise we claim that no event happened. To make sure that the event is correctly classified,we claim that the event happened in this sample only if it happened no less than in five following frames . As we said before, appropriate training of classifiers is a crucial component for getting high performance.

For this task results were evaluated by accuracy metric. On tagging dataset accuracy is equal to 0.88 what is well enough considering “problematic classes” that were discussed earlier.

## V. *Conclusions*

In this paper we discussed classification and tagging tasks we were trying to solve. We described the process of collecting audio dataset and troubles that we faced. Also we showed how classification dataset can be transfered to tagging and detection datasets. In the fourth session we demonstrated results we got and how these results can be interpreted. For training we did not use augmentation, so increasing number of samples in dataset can lead to much better performance. We showed that transfer learning is very powerful approach in deep-learning applications, especially when you have limited amount of data. We did not talk about detection methods so we leave this topic for our next works.

## References

[1] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuoyiin Chang, Tara Sainath. Deep Learning for Audio Signal Processing. In Journal of Selected Topics of Signal Processing, Vol. 13, No. 2, May 2019, pages 206–219.

[2] “Advancements in Image Classification using Convolutional Neural Network” Farhana Sultana, Paramartha Dutta ,Abu Sufian. in press

[3] http://www.cs.tut.fi/~heittolt/datasets

[4] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in Proc. IEEE ICASSP 2017, New Orleans, LA, 2017.