# Audio Events Detection in Noisy Embedded Railway Environments

Tony Marteau, Sitou Afanou, David Sodoyer, Sébastien Ambellouis, Fouzia Boukour

# Audio Events Detection in Noisy Embedded Railway Environments

Tony MARTEAU[1], Sitou AFANOU[1], David SODOYER[2], Sébastien AMBELLOUIS[2], and Fouzia BOUKOUR[2]

[1] SNCF Voyageurs, Centre d'Ingénierie du Matériel, {name.surname}@sncf.fr
[2] COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France, {name.surname}@univ-eiffel.fr

**Abstract.** Ensuring passengers' safety is one of the daily concerns of railway operators. To do this, various image and sound processing techniques have been proposed in the scientific community. Since the beginning of the 2010s, the development of deep learning made it possible to develop these research areas in the railway field included. Thus, this article deals with the audio events detection task (screams, glass breaks, gunshots, sprays) using deep learning techniques. It describes the methodology for designing a deep learning architecture that is both suitable for audio detection and optimised for embedded railway systems. We will describe how we designed from scratch two CRNN (Convolutional Recurrent Neural Network) for the detection task. And since the creation of a large and varied training database is one of the challenges of deep learning, this article also deals with the innovative methodology used to build a database of audio events in the railway environment. Finally, we will show the very promising results obtained during the experimentation in real of the model.

**Keywords:** Audio event detection · Abnormal event · Transport environment · Railway · Deep learning · CRNN

## 1 INTRODUCTION

Surveillance in the railway field is an expensive task. It requires deploying huge resources, both human and material, to ensure the safety of passengers. A whole framework dedicated to this task must be deployed: CCTV cameras and microphones, patrol and surveillance agents, barriers, etc. Nowadays, most of autonomous surveillance systems still require a human operator. With the recent image and signal processing techniques as neural networks (NN) and deep learning (DL), a robust surveillance automation becomes possible. The automation's aim is to help railway operators by reducing security issues by detecting an event very early and allowing the prompt intervention of the railway police.

Developing audio and video algorithms to detect critical events is not a new research action. But with recent NN innovations, this research area has grown

very quickly these last years. Moreover, smart video based event recognition is an active research field but is more difficult inside the railway vehicle due to occlusion issues. In this context, analysing audio environment of a railway has yield promising results in the past [16,25] with classical machine learning techniques. In this paper, we present a work in line with this question: how to detect some critical events by analysing audio environment inside a train ?

We propose to design an event detection system based on NN and DL techniques, that will be based on existing audio equipment's in actual commercial trains. We aim at increasing the capabilities of the actual surveillance systems with automatic detection and identification of some abnormal sounds.

The automatic sound classification and recognition are two active areas of study [22,17,2] and are present in various fields of application as speaker recognition [18], speech emotion classification [24,23], urban sound analysis [19], audio surveillance of roads [9], acoustic scene classification [8], event detection [21] and localisation [4]. It aims at detecting the onset and offset times and labelling for each sound event in an audio sequence.

The prolific research is due to advances of NN and DL that has deeply changed the way to design and use automatic detection systems, for both sound or image stream. In this paper, we study sound classification algorithms to deal with abnormal audio events recognition. As previously cited, several research have already been conducted between 2005 and 2015 : The European research projects BOSS, the french projects SURTRAIN and DÉGIV [16,25]. These works did not use DL and NN because the computing power of computers did not allow us to consider on-board setup. Recently, Laffitte et al. showed the way and presented studies on the automatic detection of screams and shouts in subway train using deep neural networks (DNN) [12,13].

A supervised DNN requires a large amount of data for the training task of the model. Obtaining a database combining both quantity, thousands of data, and quality, for which all the "event to be recognised" are precisely labelled become a complex paradigm. Some audio databases are publicly available in the scientific community like databases from different challenges of Detection and Classification of Acoustic Scenes and Events (DCASE) [1] but the embedded railway environment is not generally considered. This is understandable since it is difficult to collect huge amount of recordings in a train. Indeed, railway is a highly regulated environment, where very strict rules must be followed, in particular concerning fire risks assessment, electromagnetic compatibility, vibrations and personal data protection. Therefore, only certified on-board computers can be used in trains. These computers have limited computing resources for heat dissipation considerations.

The present paper addresses two problems. the first problem is dealing with the build of a railway synthetic database dedicated to abnormal audio events detection. This database has been built by mixing sound patterns and real embedded railway background sounds. The second problem is focusing on the design of convolutional and recurrent neural network for abnormal events detection trained from this database. In the introduction section we present some stud-

ies on audio classification and detection. The second section present an original database dedicated to the abnormal events detection in railway environment. Two architectures of convolutional and recurrent neural network are presented in the third section. The following sections are dedicated to the experiments description and detection results respectively. Finally the last section presents the conclusions.

## 2 A railway database for abnormal events detection

An embedded railway environment is a very specific place where new acoustic constraints have to be considered. This acoustic environment is very noisy and not stationary. It is a mixture of many acoustic sources emitted from mechanical, electrical and electronic sub-systems working simultaneously and also emitted by passengers. In this context we propose to build a dedicated database by mixing railway background and abnormal event sounds. Both are presented in the following sections and are followed by a description of the mixing method we use.

### 2.1 Railway background sounds

Railway background sounds have been recorded during technical rolling on board of multiple (suburbans, regional and high speed) SNCF train to create variability and make our system less specific. The mobile capture equipment have been placed in the middle and at the tail of the train. Six hours of background sounds have been recorded. The audio signal has been recorded on a single 32 bits channel and has been sampled at 44.1kHz. These background sounds are a mixture of sounds of the engines, sounds of friction of the wheels on the rails, sounds of air conditioner, commercial audio messages etc. These railway background signal is clearly a polyphonic and not stationary background signal. s caisses ?

### 2.2 Abnormal events and additional sounds

Four types of abnormal sounds events to detect have been chosen: gunshots, screams, glass breaks and sprays. The samples of these class sounds are extracted from Freesound website [10] in order to check the audio content all the samples have been listen before incorporating them into the final dataset. These abnormal sound are recorded on a 32 bits mono channel signal and sampled at 44.1kHz.

In a commercial train, other operation sounds as buzzers, door opening/closing, passengers conversations etc. appear. Because, these sounds are not recorded during technical rolling, we added all these additional sounds from an other railway audio dataset.

The duration distributions of abnormal events and additional sounds are presented in table 1.

| | Number | Total | Min | Mean | Max |
|---|---|---|---|---|---|
| Gunshot | 358 | 404.6s | 0.13s | 1.13s | 2.68s |
| Scream | 339 | 335.8s | 0.28s | 1.14s | 1.99s |
| Glass break | 175 | 237.9s | 0.38s | 1.35 | 2.98s |
| Spray | 310 | 253.6s | 0.13s | 0.81s | 1.92s |
| Add. sounds | 459 | 910.8s | 0.53s | 1.98s | 2.0s |

Table 1: Duration distribution of the abnormal events sequences and additional sounds.

### 2.3 Database samples generation process

Here we detail how we mix the background, abnormal and additional sounds to generate one audio sequence of our new database. The duration of each generated sound sequence is 10 seconds and below is the workflow we follow to process each sound sequence in the dataset:

1. Selection of a background sound randomly in the background dataset. The gain of the audio sample is selected randomly between 0 and -10 dB to create variability without introduce audio saturation.
2. Selection of 0 up to 3 abnormal events to detect. The temporal localisation is fixed randomly within the 10 seconds of background. Overlapping of samples is allowed and a random gain between -5 and -15 dB is applied.
3. Choice randomly of the presence or not of an other abnormal events (repeated for the three other abnormal sounds). These other events can occur when dedicated events take place. In order to make the system more reliable against these others events, necessary for a correct identification, the detector learn to consider all the other sounds as patterns not to be detected. The temporal localisation is randomly set in the 10 seconds background. Overlapping of samples allowed and a random gain between -5 and -15 dB is applied.
4. Choice randomly the presence or not of an other normal events. The temporal localisation and the gain are choosen randomly in same interval conditions.

The labels are generated at the same time than the integration of the samples of the events to be detected. The labeling is a One Hot encoding labeling: each sequence associated to an event to detect has its own label tensor. Each component of this tensor is initialized to 0, except for the frames when a event to detect occurs where the element is set to 1. The length of this label vector is equal to the length of output model. For example, the Figure 1 shows an overview of the spectrogram and its label. In this sequence a 2-seconds scream sample is inserted at the 7.5th second of the background file, the start timestamp will be 7.5 and the end timestamp will be 9.5.

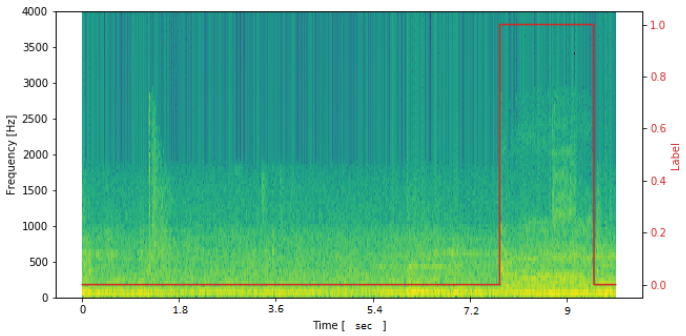



Fig. 1: Spectogram $10 \log_{10} |x(f,t)|$ of an audio sequence of 5511 frames (10 sec) with the label vector associeted in red line.

## 3 Abnormal event detection

The algorithm has to label each abnormal event that appears during one 10 seconds sequence. All the sequence is consider as the input of our CRNN. We adopt a One-vs-All (OvA) strategy to predict the labels. One CRNN is viewed as a binary classifier designed and trained for one event to detect: scream, gun shot, glass break and spray. We can take into account the polyphonic detection problem ie. the cases where several events appear at the same time. Finally by using dedicated "less complex" networks, we can expect faster detection by using multi-processors capacity of the computer.

Each CRNN consists in extracting the feature map of each sequence (convolution layers) and to analyze the temporal coherence of the frequency activity (recurrent layers). Finally, it computes one event activity probability for every frames of the sequence. The final detection is done by applying a threshold $\sigma = 0.5$.

### 3.1 Model architecture

We propose two models based on the Convolutional Recurrent Neural Network (CRNN) developed in the recent papers [14,6,3,5]. These studies show that the combination of convolutional and recurrent layers allows to jointly capture the invariance in frequency domain and to model short and long term temporal dependencies.

The first model consists in the following layers : two convolutional, two Gated Recurrent Unit (GRU) [7] and three fully connected (FC) layers. GRU is prefered to Long Short Term Memory (LSTM) to reduce the number of parameters and avoid the vanishing gradient problem. In many works in audio events detection, MEL coefficients (MFCCs), are generally used as input features [13]. In [6], the input of the network is a log time-frequency representation of the data in

MEL band energies over frames. In our work, we use directly the magnitudes of the spectrum [11] and let the first layers optimise the extraction of higher level parameters. $N$ spectra are computed on each 10 seconds sequence of the database.

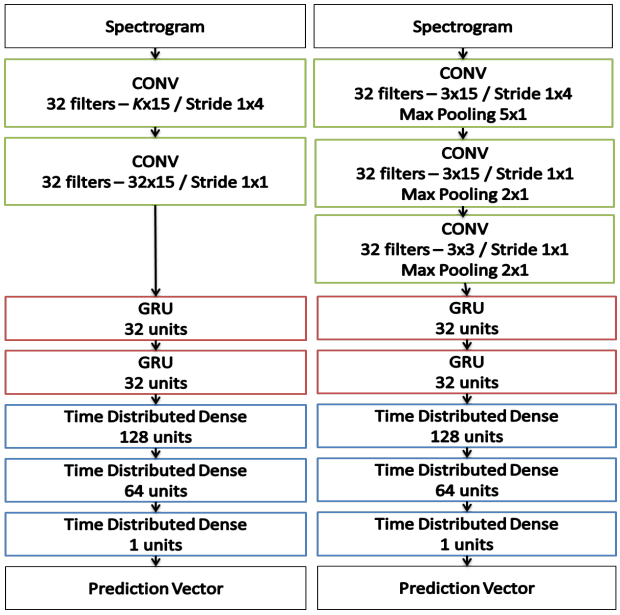
The basic version of this first model (CRNN 1) is defined as follows (fig. 2a):

- **1 convolutional layer**. It is composed of 32 filters that use $k \times 15$ kernel. Here $k$ is the total number of filters in the time-frequency representation. We use a stride of 4 samples to reduce the dimensionality of the resulting feature map. We use a stride rather than pooling to obtain better computational performance [20]. The convolutional layer is activated by a ReLu function.
- **1 convolutional layer**. This second convolutional layers is composed of 32 filters that use $32 \times 15$ kernel, a stride of 1 and is activated by a ReLu function.
- **2 layers of GRU**. GRUs are used to extract temporal information from the feature map output of the second convolutional layer. Each GRU layer has 32 units.
- **3 FC layers**. The 2 first layers are respectively 128 and 64 neurons layers activated through a ReLU function. The last one is a single neuron layer activated through a Sigmoid function. These layers gradually reduce the size of the output and are distributed over the time. The last layer is computing the activity probability for each class.

The second model (CRNN 2) is inspired by the network architecture of [3]. For this configuration, the convolution operation of the first and the second layer does not integrate all the frequencies as before i.e. it uses a $3 \times 15$ kernel, 3 along the frequencies range as in [3]. A third convolutional layer is added and a max-pooling operation is used after each convolutional layer. More precisely, the basic version of this second model (CRNN 2) is defined as follows (fig. 2b):

- **1 convolutional layer**. The layer is composed of 32 filters that use $3 \times 15$ kernel. As in CRNN 1 we use a stride of 4 samples to reduce the dimensionality of the feature map. A max-pooling of $5 \times 1$ is applied to reduce the dimension output along the frequencies range.
- **1 convolutional layer**. The layer is composed of 32 filters with a $3 \times 15$ kernel and a stride of 1. It is followed by a $2 \times 1$ max-pooling.
- **1 convolutional layer**. The layer is composed of 32 filters with a $3 \times 3$ kernel, a stride of 1 and is followed by a $2 \times 1$ max-pooling.
- **2 layers of GRU**. GRUs are used to extract temporal information from the feature-map of the third convolution. Each layer has 32 units.
- **3 FC layers**. The 2 first layers are activated through a ReLU function, and the last one through a Sigmoid function. These layers gradually reduce the size of the output and are distributed over the time as in CRNN 1. The last layer is computing the activity probability for each class.

For both network, the last FC layer is not providing one probability for all frames. Because of the use of a 4 samples stride for the first layer of each network, the length of the output vector is divided by 4 i.e. it is equal to $N/4$.




(a) CRNN 1 (b) CRNN 2

Fig. 2: The two basic CRNN architectures.

## 4 Evaluation

The experiments consist firstly in evaluating the influence of the units number of the GRU and the structure of the convolutional layers for CRNN 1 and CRNN2. Secondly, we compare the performances of both architectures. For this both study the evaluations have been realised with the synthetic mixture described in section 2.3. Finally, in the last run, we present a preliminary result for the portability and the feasibility of detection in real conditions of the railway environment.

### 4.1 Features extraction description

The input features of the networks is the module of the complex-valued spectra computed on a $T$ seconds audio signal $x(t)$. The spectra $x(f, n)$ are calculated by a Fast Fourier Transform using a sliding Hamming windows of 200 samples and a 60% overlap: $f$ and $n$ denote respectively the frequency and the frame index. Finally, the input of the network is a matrix composed of $N$ magnitude vectors $|x(f, n)|$. In our experiments, $T = 10s$, the sampling rate is $f_s = 44.1kHz$ and the input is $101 \times 5511$ matrix composed of 5511 vectors of $k = 101$ frequencies.

### 4.2 Evaluation procedure in synthetic mixtures case

We evaluate both CRNN architectures by modifying their parameters.

In a first step, we study the influence of the number of units per recurrent layer: 0 (simple convolutional network), 32 (the basic CRNN described previoulsly) and 64 units per recurrent layers. For these three cases, the number of convolution filters is fixed to 64. This step is realised for the architecture CRNN 1.

In a second step, we test the influence of the filters number per convolutional layers for two configurations: 32 filters (the basic CRNN described previoulsy) and 64 filters. In both cases, the units number of GRU layers is fixed to 32. This step is realised for the architecture CRNN 1 and CRNN 2.

CRNN is trained and tested independently for each event using dedicated built synthetic database. In total, we generated for each event 11000 10 seconds sequences: 7000 sequences for training, 2000 for validation, and 2000 for testing. This corresponds to 30 hours of sounds for each class.

For learning phase, we use a 0.01 learning rate and the Adam optimizer for binary crossentropy loss function. Early stopping is triggered after twenty iterations without loss improvement on validation database. A batch normalisation, a 0.2 dropout and a layer normalisation are applied on each convolutional or recurrent layer during the training phase.

The test phase consist in presenting the 2000 sequences of 5511 spectra to compare the $N$ predictions with the $N$ truth labels for each sequence. With a sequence length of 5511, the length of output vector of the CRNN is equal to $N = 1375$. The evaluation is made by computing *accuracy*, *precision*, *recall* rates and $F1score$ [15] for the $2000 \times N = 2750000$ predictions. The rates are calculated as:

$$accuracy = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad \text{(1a)} \qquad F1score = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad \text{(1b)}$$

$$precision = \frac{\text{TP}}{\text{TP + FP}} \quad \text{(2a)} \qquad recall = \frac{\text{TP}}{\text{TP+FN}} \quad \text{(2b)}$$

where TP, FP, TN and FN are respectively the number of true positive, false positive, true negative and false negative predictions.

### 4.3 Detection description in real environment

To check the performance of our CRNN architectures under the real conditions of the railway environment, we carried out tests at the SNCF Technicentre des Ardoines, in Vitry sur Seine. There, we were able to access an Ile-de-France region suburban train (Z2N train) operating on lines C and D. The train used for the test was stationary for maintenance reasons. The train, however, had its engine running, and many trains were near which produced a huge amount of noise. We set up the system inside the train and used only one IP camera. When the train is set ON, the system launched automatically with the onboard computer. It assigned itself a fixed IP address and started its autotests. The system was able to connect to the IP camera and start reading the audio stream

from the IP camera. Some abnormal events were played using a speaker placed at different locations in the train:

1. Speaker at 2m from the microphone, and we play 3 different samples
2. Speaker at 4m from the microphone, and we play 3 different samples
3. Speaker at 6m from the microphone, and we play 3 different samples

The audio stream is captured and stored in a FIFO memory of 10 sec refreshed every 0.5 seconds. The detection tests was performed using the models learned with the synthetic mixture presented in 2.3 (without new learning phase). In this real context the evaluation consisted in checking the detection of the corresponding event by monitoring logs files of the system. For these experiments only two critical abnormal events have been tested: the scream and the gunshot events.

### 4.4 Results in synthetic environment

The table 2 and the table 3 present the results of the experimentation plan described in the section 4.2.

For both tables $Target$ and $BG$ refer respectively to the event class and the background.

The first conclusion is that both architectures yield good results in our railway environment with an accuracy over 90%.

On the one hand, regarding the impact of the recurrent layers (table 2), we can observe that the performance decrease without recurrent layers for all rates. It confirms that we need to take into account the temporal evolution of spectral patterns extracted by the convolutional Layers. The GRU layers do not benefit to the spray events that has really complex spectral-temporal structure. For the three other classes, recurrent layers improve target recall: 12% for Scream and up to 92% for Gunshot. On other hand, the number of units in recurrent layers does not influence significantly the quality detection.

The table 3 presents the effect of the number of filters on the performance. It is difficult to highlight a major improvement w.r.t the number of filters used. Nevertheless, for CRNN 1, 32 or 64 filters yield quite similar rates and for CRNN 2, increase the number of filters seems to severely decrease performance for all events except for scream.

### 4.5 Results in real environment

The results in Table 4 present the number of detected events for scream and Gunshot. In general manner, all events are correctly detected. However, it appears clearly that the detection rate depends on the distance between the source and the microphone. The sensibility effect is reduced by increasing the number of microphones. In this case, the microphones have to be distributed in the railway vehicle insuring that the distance between passengers and one microphone is less that 6m.

| Event | Config | Accuracy | F1 Score | | Precision | | Recall | |
|---|---|---|---|---|---|---|---|---|
| | | | Target | BG | Target | BG | Target | BG |
| Scream | 0 GRU | 0.90 | 0.82 | 0.93 | 0.92 | 0.90 | 0.73 | 0.97 |
| | 32 GRU | 0.93 | 0.87 | 0.95 | 0.92 | 0.93 | 0.82 | 0.97 |
| | 64 GRU | 0.93 | 0.88 | 0.95 | 0.93 | 0.93 | 0.83 | 0.97 |
| Gunshot | 0 GRU | 0.80 | 0.54 | 0.87 | 0.82 | 0.80 | 0.40 | 0.96 |
| | 32 GRU | 0.91 | 0.83 | 0.94 | 0.89 | 0.91 | 0.77 | 0.96 |
| | 64 GRU | 0.90 | 0.82 | 0.93 | 0.89 | 0.90 | 0.75 | 0.96 |
| Spray | 0 GRU | 0.95 | 0.87 | 0.97 | 0.92 | 0.95 | 0.83 | 0.98 |
| | 32 GRU | 0.95 | 0.88 | 0.97 | 0.96 | 0.95 | 0.81 | 0.99 |
| | 64 GRU | 0.96 | 0.90 | 0.97 | 0.94 | 0.96 | 0.86 | 0.99 |
| Glass | 0 GRU | 0.91 | 0.71 | 0.95 | 0.86 | 0.92 | 0.60 | 0.98 |
| | 32 GRU | 0.95 | 0.86 | 0.97 | 0.91 | 0.96 | 0.82 | 0.98 |
| | 64 GRU | 0.95 | 0.86 | 0.97 | 0.92 | 0.96 | 0.81 | 0.99 |

Table 2: Detection performances for the architecture CRNN 1 in function the absence or the complexity of GRU layers. "0 GRU" stands for no GRU.

| Event | Config | | Acc. | F1 Score | | Precision | | Recall | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Target | BG | Target | BG | Target | BG |
| Scream | CRNN1 | 32 filters | 0.92 | 0.87 | 0.95 | 0.89 | 0.94 | 0.85 | 0.96 |
| | | 64 filters | 0.93 | 0.87 | 0.95 | 0.92 | 0.93 | 0.82 | 0.97 |
| | CRNN2 | 32 filters | 0.90 | 0.81 | 0.93 | 0.94 | 0.89 | 0.72 | 0.98 |
| | | 64 filters | 0.91 | 0.83 | 0.94 | 0.92 | 0.91 | 0.76 | 0.97 |
| Gunshot | CRNN1 | 32 filters | 0.90 | 0.81 | 0.93 | 0.91 | 0.90 | 0.73 | 0.97 |
| | | 64 filters | 0.91 | 0.83 | 0.94 | 0.89 | 0.91 | 0.77 | 0.96 |
| | CRNN2 | 32 filters | 0.89 | 0.80 | 0.93 | 0.87 | 0.90 | 0.74 | 0.96 |
| | | 64 filters | 0.85 | 0.71 | 0.90 | 0.79 | 0.86 | 0.64 | 0.93 |
| Spray | CRNN1 | 32 filters | 0.95 | 0.89 | 0.97 | 0.93 | 0.96 | 0.85 | 0.98 |
| | | 64 filters | 0.95 | 0.88 | 0.97 | 0.96 | 0.95 | 0.81 | 0.99 |
| | CRNN2 | 32 filters | 0.96 | 0.91 | 0.98 | 0.94 | 0.97 | 0.89 | 0.98 |
| | | 64 filters | 0.87 | 0.61 | 0.93 | 0.96 | 0.87 | 0.45 | 0.99 |
| Glass | CRNN1 | 32 filters | 0.94 | 0.81 | 0.96 | 0.86 | 0.95 | 0.76 | 0.97 |
| | | 64 filters | 0.95 | 0.86 | 0.97 | 0.91 | 0.96 | 0.82 | 0.98 |
| | CRNN2 | 32 filters | 0.96 | 0.88 | 0.97 | 0.89 | 0.97 | 0.86 | 0.98 |
| | | 64 filters | 0.82 | 0.62 | 0.88 | 0.50 | 0.96 | 0.83 | 0.82 |

Table 3: Detection performances in function the complexity of convolutional layers for architecture CRNN 1 and CRNN 2.

| Distance | Scream | Gunshot |
|---|---|---|
| 2m | 3/3 | 3/3 |
| 4m | 3/3 | 3/3 |
| 6m | 3/3 | 2/3 |

Table 4: Event detection results in real environment for three distances between events and microphone.

## 5 CONCLUSIONS

In this paper, we present a new railway audio database and two CRNN architectures designed for abnormal audio event detection. Our evaluation show that using a kernel shape of the same size as the number of frequency bands (CRNN 1) yield better rates. As in [6,3], the detection results show that catching the temporal structure of the spectrum improves the performance rates. Increasing the number of filters has a weak impact on the detection performance only for CRNN 1. The complexity of the CRNN 1 and the number of parameters are lower than for CRNN 2. It seems to be a quite promising embedded solution for real railway conditions.

## 6 Acknowledgement

We would like to thank Helmi REBAI and Martin OLIVIER for strongly contributing to the advancement of this study.

## References

1. http://dcase.community/challenge2019/
2. Abeßer, J.: A review of deep learning based methods for acoustic scene classification. Applied Sciences **10**(6) (2020)
3. Adavanne, S., Pertilä, P., Virtanen, T.: Sound event detection using spatial features and convolutional recurrent neural network. In: IEEE Int. Conf. on Acoust., Speech and Signal Process. pp. 771–775. New Orleans, LA, USA (Mar, 5-9 2017)
4. Adavanne, S., Politis, A., Nikunen, J., Virtanen, T.: Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. IEEE J. of Sel. Topics in Signal Process. **13**(1), 34–48 (2019)
5. Adavanne, S., Parascandolo, G., Pertilä, P., Heittola, T., Virtanen, T.: Sound event detection in multichannel audio using spatial and harmonic features. In: Detection and Classification of Acoust. Scenes and Events Workshop. Budapest, Hungary (Sept, 3 2016)
6. Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T.: Convolutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Trans. on Audio, Speech, and Lang. Process. **25**(6), 1291–1303 (Jun 2017)
7. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. pp. 103–111. Association for Computational Linguistics, Doha, Qatar (Oct, 25 2014)
8. Drossos, K., Magron, P., Virtanen, T.: Unsupervised adversarial domain adaptation based on the wasserstein distance for acoustic scene classification. In: IEEE Workshop on Appl. of Signal Process. to Audio and Acoust. pp. 259–263. New Paltz, NY, USA (Oct, 20-23 2019)
9. Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., Vento, M.: Audio surveillance of roads: A system for detecting anomalous sounds. IEEE Trans. on Intell. transp. Syst. **17**(1), 279–288 (2016)

10. Font, F., Roma, G., Serra, X.: Freesound technical demo. In: ACM Int. Conf. on Multimedia. pp. 411–412. Barcelona, Spain (October, 21 2013)
11. Huzaifah, M.: Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. CoRR **abs/1706.07156** (2017)
12. Laffitte, P., Sodoyer, D., Tatkeu, C., Girin, L.: Deep neural networks for automatic detection of screams and shouted speech in subway trains. In: IEEE Int. Conf. on Acoust., Speech and Signal Process. pp. 6460–6464. Shanghai, China (Mar, 20-25 2016)
13. Laffitte, P., Wang, Y., Sodoyer, D., Girin, L.: Assessing the performances of different neural network architectures for the detection of screams and shouts in public transportation. Expert Syst. with Appl. **117**, 29 – 41 (2019)
14. Lim, H., Park, J., Lee, K., Han, Y.: Rare sound event detection using 1D convolutional recurrent neural networks. In: Detection and Classification of Acoust. Scenes and Events Workshop. Munich, Germany (Nov, 16 2017)
15. Mesaros, A., Heittola, T., Virtanen, T.: Metricsfor polyphonic sound event detection. Applied Sciences **6**(6), 162 (2016)
16. Pham, Q.C., Lapeyronnie, A., Baudry, C., Lucat, L., Sayd, P., Ambellouis, S., Sodoyer, D., Flancquart, A., Barcelo, A.C., Heer, F., Ganansia, F., Delcourt, V.: Audio-video surveillance system for public transportation. In: 2nd Int. Conf. on Image Process. Theory, Tools and Appl. Paris, France (Jul, 7-10 2010). https://doi.org/10.1109/ipta.2010.5586783
17. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S., Sainath, T.: Deep learning for audio signal processing. IEEE J. of Sel. Topics in Signal Process. **13**(2), 206–219 (2019)
18. Ravanelli, M., Bengio, Y.: Speaker recognition from raw waveform with sincnet. In: IEEE Spoken Lang. Technol. Workshop. pp. 1021–1028. Athens, Greece (Dec, 18-21 2018)
19. Salamon, J., Bello, J.P., Farnsworth, A., Kelling, S.: Fusing shallow and deep learning for bioacoustic bird species classification. In: IEEE Int. Conf. on Acoust., Speech and Signal Process. pp. 141–145. New Orleans, LA, USA (Mar, 5-9 2017)
20. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: The all convolutional net. In: 3rd Int. Conf. on Learning Representations. San Diego, CA, USA (May 7-9 2015)
21. Turpault, N., Serizel, R., Salamon, J., Shah, A.P.: Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In: Detection and Classification of Acoust. Scenes and Events Workshop. pp. 253–257. New York University, NY, USA (October 2019)
22. Virtanen, T., Plumbley, M.D., Ellis, D. (eds.): Computational Analysis of Sound Scenes and Events. Springer Int. Publishing, 1 edn. (2018)
23. Xie, Y., Liang, R., Liang, Z., Huang, C., Zou, C., Schuller, B.: Speech emotion classification using attention-based lstm. IEEE/ACM Trans. on Audio, Speech, and Lang. Process. **27**(11), 1675–1685 (2019)
24. Zhang, Z., Coutinho, E., Deng, J., Schuller, B.: Cooperative learning and its application to emotion recognition from speech. IEEE/ACM Trans. on Audio, Speech, and Lang. Process. **23**(1), 115–126 (2015)
25. Zouaoui, R., Audigier, R., Ambellouis, S., Capman, F., Benhadda, H., Joudrier, S., Sodoyer, D., Lamarque, T.: Embedded security syst. for multi-modal surveillance in a railway carriage. In: Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XI and Optical Materials and Biomaterials in Security and Defence Syst. Technol. XII. SPIE, Toulouse, France (Oct, 21 2015)