Learning Frame Level Attention for Environmental Sound Classification

Zhichao Zhang\textsuperscript{a}, Shugong Xu\textsuperscript{a}, Shunqing Zhang\textsuperscript{a}, Tianhao Qiao\textsuperscript{a} and Shan Cao\textsuperscript{a}

\textsuperscript{a}Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, 200444, China

\textbf{ABSTRACT}

Environmental sound classification (ESC) is a challenging problem due to the complexity of sounds. The classification performance is heavily dependent on the effectiveness of representative feature representations extracted from the environmental sounds. However, ESC often suffers from the semantically irrelevant frames and silent frames. In order to deal with this, we employ a frame-level attention model to focus on the semantically relevant frames and salient frames. Specifically, we first propose a convolutional recurrent neural network to learn spectro-temporal features and temporal correlations. Then, we extend our convolutional RNN model with a frame-level attention mechanism to learn discriminative feature representations for ESC. We investigated the classification performance when using different attention scaling function and applying different layers. Experiments were conducted on ESC-50 and ESC-10 datasets. Experimental results demonstrated the effectiveness of the proposed method and our method achieved the state-of-the-art or competitive classification accuracy with lower computational complexity. We also visualized our attention results and observed that the proposed attention mechanism was able to lead the network to focus on the semantically relevant parts of environmental sounds.

1. Introduction

Environmental sound classification (ESC) has received increasing research attention during recent several years, which is widely applied in surveillance [22], home automation [26], scene analysis [3] and machine hearing [16]. Different from music and speech recognition tasks, ESC has quite limited pre-known knowledge with respect to the temporal and frequency characteristics. In addition, as the environmental sounds usually behaves in a variable fashion, several naive deterministic prediction schemes have been often failed to obtain good performance, which leads ESC to be more challenging in nowadays.

To address this challenge, a variety of signal processing and machine learning techniques have been applied for ESC. For the former, some naive features, e.g. zero-crossing rate or short-time energy, are analyzed via some heuristic backends. With the development of signal processing skills, some dictionary-based methods, such as dictionary learning [7], matrix factorization [4], are successfully applied in ESC [18]. However, this type of schemes often requires tedious feature design process to obtain reasonable accuracy. In addition, some machine learning techniques, including gaussian mixture model (GMM) [9] and support vector machine (SVM) [7] have been widely adopted in ESC. Since these techniques have ability to handle complex high-dimensional features, multiple feature transformation schemes have been applied, such as mel-frequency cepstral coefficient (MFCC), mel-spectrogram features [7], gammatone-spectrogram features [7] and wavelet-based features [10].

In recent years, deep neural networks (DNNs) have shown outstanding performance in feature extraction for ESC. Compared to hand-crafted feature, DNNs have the ability to extract discriminative feature representations from large quantities of training data and generalize well on unseen data. McLoughlin et al. [17] proposed a deep belief network to extract high-level feature representations from magnitude spectrum, which yielded better results than the traditional methods. Piczak [19] first evaluated the potential of convolutional neural network (CNN) in classifying short audio clips of environmental sounds and showed excellent performance on several public datasets. In order to model the sequential dynamics of environmental sound signals, Vu et al. [28] applied a recurrent neural network (RNN) to learn temporal relationships.

However, these schemes are limited to improve classification accuracy since they usually ignore the complex temporal characteristics of environmental sounds. Generally speaking, the temporal structure of environmental sounds can be transient (e.g. gun shot), continuous (e.g. rain) or intermittent (e.g. dog bark), which makes it unfeasible to simply model the temporal variations via the existing techniques like hidden markov model (HMM). In addition, sound clips usually contain many periods of silence in public ESC datasets, with only a few intermittent frames associated with the characteristics of sound classes. Figure 1 shows some examples of log gammatone spectrogram in ESC-50 dataset. We see that some salient and semantically relevant features only distribute in a few frames and the features usually contain silent or noisy frames, which reduces the robustness of model and increase misclassification. To deal with the problems, we explore frame-level attention mechanisms for CNN layers and RNN layers to help the network focus on semantically relevant frames. Attention mechanisms have shown outstanding performance in learning relevant feature representations for sequence data and have been successfully applied to a wide variety of tasks, including speech recognition [6], machine translation [2, 24], document classification [29] and so on. In the field of ESC, several works [11, 13, 23, 15, 32] have...
studied the effectiveness of attention mechanisms and have obtained promising results in several datasets.

In this paper, we proposed an attention mechanism based convolutional RNN architecture (ACRNN) in order to focus on semantically relevant frames and produce discriminative features for ESC. In our proposed technique, softmax is used as scaling function to generate attention weights [32]. However, the softmax based attention will be forced to focus on a few frames with large weights, which is not ideal for some types of environmental sounds, especially for continuous signals. It is also shown that the softmax based attention does not work well for CNN layers. Therefore, we investigated a sigmoid based attention for CNN layers and compared the classification accuracy with the softmax based one. In addition, this type of method usually requires a large amount of training data and the current public ESC datasets are quite limited. Therefore, we applied an efficient data augmentation scheme named mixup for ESC, which was originally proposed to handle image-related tasks [30]. We evaluated our proposed method on the ESC-10 and ESC-50 datasets and our model achieved the state-of-the-art or competitive performance in terms of classification accuracy. Furthermore, we visualized the learned attention results in order to give a better understanding of how the proposed frame-level attention helps recognize different environmental sounds. The main contributions of this paper are summarized as follows.

- To deal with silent frames and semantically irrelevant frames, we employ an attention model to automatically focus on the semantically relevant frames and produce discriminative features for ESC. We explore both the performance of frame-level attention mechanism for CNN layers and RNN layers. We investigate the selection of attention scaling function and visualize the attention results to have a better understanding how frame-level attention works.

- To analyze temporal relations, we propose a novel convolutional RNN model which first uses CNN to extract high level feature representations and then inputs the features to bidirectional GRUs for temporal summarization. We combine the convolutional RNN and attention model in a unified architecture.

- To further improve classification performance, we apply a data augmentation pipeline which is applied directly to audio spectrogram of ACRNN via mixup and helps the network learn useful features.

The rest of this paper is organized as follows. Recent related works of ESC are introduced in Section 2. Section 3 provides detailed description about the proposed methods, including feature extraction, network architecture, frame-level attention mechanism and data augmentation. Section 4 provides the experimental settings and results on the ESC-10 and ESC-50 datasets. Finally, Section 5 concludes the paper.

2. Related Work

In this section we introduce the recent deep learning methods for environmental sound classification.

The 2-D CNNs are originally used to analyze spectrogram-like features. Piczak [19] first proposed to apply a 2-D CNN to learn the log mel spectrogram features and obtained a significant improvement than KNN, SVM and random forest. Since log mel spectrogram could be considered as a 2-D image, several well-known image recognition networks, including AlexNet and GoogLeNet, have been adopted to classify the input spectrogram features for ESC [5]. More recently, Zhang et al. [31] compared the performance between log mel spectrogram and log gammatone spectrogram features for ESC and reported that the log gammatone spectrogram based system performed better in terms of classification accuracy.

Some researchers also proposed to learn the representations directly from 1-D raw waveform data. Dai et al. [8] proposed to use 1-D convolution and pooling to learn classification model which showed competitive accuracy with log mel spectrogram based methods, however, required more convolutional layers (up to 34 layers). Tokozume et al. [25] proposed an end-to-end network named EnvNet, which first used 1-D convolution to learn 2-D feature maps and then classified via 2-D convolution and pooling. In [1], Aytar et al. proposed to learn rich sound features from a large amount of unlabeled videos and then adapted the learned sound features to target datasets, where the sound recognition network utilized 1-D convolution.

Recently, attention mechanisms have been incorporated to improve classification performance in the field of ESC.
[13, 23, 15, 11, 32]. Guo et al. [11] proposed to apply attention to a convolutional LSTM network, where the attention is calculated via a weighted sum up of the output of LSTM layers along time dimension. Closed to [11], Wang et al. [13] proposed to stack multiple attention network to get more powerful representations. In addition, Li et al. [15] proposed a temporal attention mechanism for convolutional layers to enhance the representative ability of CNN, which was calculated from input spectrogram and re-weighted the CNN feature maps via dot-product operation along the time dimension. In this paper, we investigate the frame-level attention mechanism for CNN layers and RNN layers and provide a visualization results of our attention to give a better understanding how frame-level attention helps recognize different environmental sounds.

3. Methods

In this section, we introduce the proposed method for ESC. Firstly, we describe how to generate log gammatone spectrogram (Log-GTs) features from environmental sounds. Then, we introduce the architecture of ACRNN which combines convolutional RNN and a frame-level attention mechanism. The architecture of proposed ACRNN is shown in Figure 2. And we will give a detailed description about the architecture of convolutional RNN and the attention mechanism, respectively. Finally, the data augmentation methods are introduced.

3.1. Feature Extraction and Preprocessing

Given a signal, we first use short-time Fourier transform (STFT) with hamming window size of 23 ms (1024 samples at 44.1kHz) and 50% overlap to extract the energy spectrogram. Then, we apply a 128-band gammatone filter bank [27] to the energy spectrogram and the resulting spectrogram is converted into logarithmic scale. In order to make efficient use of limited data, the spectrogram is split into 128 frames (approximately 1.5s in length) with 50% overlap. The delta information of the original spectrogram is calculated, which is the first temporal derivative of the static spectrogram. Afterwards, we concatenate the static log gammatone spectrogram and its delta information to a 3-D feature representation $X \in \mathbb{R}^{128 \times 128 \times 2}$ (Log-GTs) as the input of the network.

3.2. Architecture of Convolutional RNN

In this section, we introduce the proposed convolutional RNN which is used to analyze Log-GTs for ESC. We first use CNN to learn high level feature representations from the Log-GTs. Then, the CNN-learned features are fed into bidirectional gated recurrent unit (Bi-GRU) layers which are used to learn the temporal correlation information. Finally, these features are fed into a fully connected layer with a softmax function to output the probability distribution of different classes. In this paper, the convolutional RNN is comprised of eight convolutional layers ($l_1$-$l_8$) and two Bi-GRU layers ($l_9$-$l_{10}$). The architecture and parameters of network are as follows:

- $l_1$-$l_2$: The first two stacked convolutional layers use 32 filters with a receptive field of (3, 5) and stride of (1, 1). This is followed by a max-pooling with a (4, 3) stride to reduce the dimensions of feature maps. ReLU activation function is used.
- $l_3$-$l_4$: The next two convolutional layers use 64 filters with a receptive field of (3, 1) and stride of (1, 1), and is used to learn local patterns along the frequency dimension. This is followed by a max-pooling with a (4, 1) stride. ReLU activation function is used.
- $l_5$-$l_6$: The following pair of convolutional layers uses 128 filters with a receptive field of (1, 5) and stride of (1, 1), and is used to learn local patterns along the time
Frame-level attention for (a) CNN layers and (b) RNN layers. For CNN layers, we use frame-level attention to obtain attention map, which is multiplied in frame-wise of CNN features, resulting the attention weighted features. For RNN layers, we utilize frame-level attention to obtain attention weights, which is multiplied in frame-wise of input features. Then, we aggregate these attention weighted representations to form a feature vector, which can be seen as a high-level representation of a sound like *dog bark*.

In order to reduce the frequency dimension to one. Then, we use a scaling function to form a normalized attention map $A \in \mathbb{R}^{1 \times T \times 1}$, which holds the frame-level attention weights for CNN features. The process can be expressed as,

$$A = \sigma(\text{AvgPool}(\text{Conv}^{3 \times 3}(M)))$$

Where $\sigma$ denotes scaling function. With attention map $A$, the attention weighted CNN features are obtained as,

$$M' = M \cdot A$$

The attention is applied by multiplying the attention vector $A$ to each feature vector of $M$ along frequency dimension and channel dimension.

### 3.3.2. Attention for RNN Layers:

As shown in Figure 3(b), we first feed the Bi-GRU output $h_t = [\tilde{h}_1, \tilde{h}_t]$ through a one-layer MLP to obtain a hidden representation of $h_t$, then we calculate the normalized importance weights $\beta_t$ as,

$$\beta_t = \frac{\exp(W \ast h_t)}{\sum_{t=1}^{T} \exp(W \ast h_t)}$$

Where $W$ is the weight matrix of the MLP. After that, we compute the feature vector $v$ through a weighted sum of the frame-level convolutional RNN features based on the
weights $\beta_i$ as,

$$v = \sum_{i=1}^{T} \beta_i h_i \quad (4)$$

The feature vector $v$ is forwarded into the fully connected layer for final classification.

### 3.4. Data Augmentation

Limited data easily leads model towards overfitting. In this paper, we use time stretch with a factor randomly selected from [0.8, 1.3] and pitch shift with a factor randomly selected from [-3.5, 3.5] to increase raw training data size. In addition, an efficient mixup [30] augmentation method is used to construct virtual training data and extend the training distribution. In mixup, the training feature-target pair $(\hat{x}, \hat{y})$ is generated by mixing two feature-target samples, which is determined by

$$\begin{align*}
\hat{x} &= \lambda x_i + (1-\lambda)x_j \\
\hat{y} &= \lambda y_i + (1-\lambda)y_j
\end{align*} \quad (5)$$

where $x_i$ and $x_j$ are two features randomly selected from the original training Log-GTs, $y_i$ and $y_j$ are their one-hot labels. The mixing factor $\lambda$ is determined by a hyper-parameter $\alpha$ and $\lambda \sim \text{Beta}(\alpha, \alpha)$. The training targets used for the mixed samples are produced with the same proportion. The mixing between features and corresponding one-hot labels encourages the model to learn linear interpolated characteristics in-between training examples, which can reduce the amount of undesirable oscillations when we predict samples outside the training examples [30]. For example, suppose that there is a dog bark Log-GTs and a rain Log-GTs, whose one-hot labels are $y_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0]$ and $y_2 = [0, 1, 0, 0, 0, 0, 0, 0, 0]$ respectively. If we mix the dog bark Log-GTs and the rain Log-GTs with $\lambda$ equal to 0.7, the training target for the mixed sample should be set as $\hat{y} = [0.7, 0.3, 0, 0, 0, 0, 0, 0, 0]$.

### 4. Experiments and Results

#### 4.1. Experiment Setup

To evaluate the performance of our proposed method, we carry out experiments on two publicly available datasets: ESC-50 and ESC-10 [20]. ESC-50 is a collection of 2000 environmental recordings containing 50 classes in 5 major categories, including animals, natural soundscapes and water sounds, human non-speech sounds, interior/domestic sounds, and exterior/urban noises. All audio samples are 5 seconds in duration with a 44.1 kHz sampling rate. ESC-10 is a subset of 10 classes (400 samples) selected from the ESC-50 dataset (dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, fire crackling).

In this paper, we use a sampling rate of 44.1 kHz for all samples in order to use rich high-frequency information. For training, all models optimize cross-entropy loss using mini-batch stochastic gradient descent with Nesterov momentum of 0.9. Each batch consists of 64 segments randomly selected from the training set without repetition. All models are trained for 300 epochs by beginning with an initial learning rate of 0.01, and then divided the learning rate by 10 every 100 epochs. We initialize the network weights to zero mean Gaussian noise with a standard deviation of 0.05. In the test phase, we evaluate the whole sample prediction with the highest average prediction probability of each segment. Both the training and testing features are normalized by the global mean and standard deviation of the training set. All models are trained using Keras library with TensorFlow backend on a Nvidia P100 GPU with 12GB memory.

#### 4.2. Experiment Results

We compare our model with existing methods reported as PiczakCNN [19], SoundNet [1], WaveMsNet [33], EnvNet-v2 [25] and Multi-Stream CNN [15]. According to [19], PiczakCNN consists of two convolutional layers and three fully connected layers. The input features of CNN are generated by combining log mel spectrogram and its delta information. We refer PiczakCNN as a baseline method.

The results are summarized in Table 1. We see that ACRNN outperforms PiczakCNN and obtained an absolute improvement of 13.2% and 21.2% on ESC-10 and ESC-50 datasets, respectively. Then, we compare our model with several state-of-the-art methods: SoundNet8 [1], WaveMsNet [33], EnvNet-v2 [25] and Multi-Stream CNN [15]. We observe that on both ESC-10 and ESC-50 datasets, ACRNN obtains the highest classification accuracy. Note that WaveMsNet [33] and Multi-Stream CNN [15] achieve the same classification accuracy as ACRNN on ESC-10 but using feature fusion (raw data and spectrogram features), whereas ACRNN only utilizes spectrogram features.

| Model                  | ESC-10  | ESC-50  |
|------------------------|---------|---------|
| KNN [21]               | 66.7%   | 32.3%   |
| SVM [21]               | 67.5%   | 39.6%   |
| Random Forest [21]     | 72.7%   | 44.3%   |
| AlexNet [5]            | 78.4%   | 78.7%   |
| Google Net [5]         | 63.2%   | 67.8%   |
| PiczakCNN [19]         | 80.5%   | 64.9%   |
| SoundNet [1]           | 92.1%   | 74.2%   |
| WaveMsNet [33]         | 93.7%   | 79.1%   |
| EnvNet-v2 [25]         | 91.4%   | 84.9%   |
| ProCNN [14]            | 92.1%   | 82.8%   |
| Multi-Stream CNN [15]  | 93.7%   | 83.5%   |
| ACRNN                  | 93.7%   | 86.1%   |

Table 1: Comparison of ACRNN and existing methods. We perform 5-fold cross validation (CV) by using the official fold settings. The average results of CV are recorded.
by ACRNN for ESC-50 dataset. We see that most classes achieve higher accuracy than 80% (32/40). Particularly, church bells obtains a 100% recognition rate. However, we observe that only 52.5% (21/40) helicopter samples are correctly recognized with 17.5% (7/40) samples misclassified as airplane. We attribute this mistakes to the similar characteristics between the two environmental sounds.

In order to further demonstrate the efficiency of the proposed method, we compare the computational complexity of our proposed approach and PiczakCNN [19] that is usually referred as a baseline of neural network methods for ESC tasks. We reproduced the network architecture of PiczakCNN according to the description provided by [19]. Table 2 shows the number of model parameters and floating point operations (FLOPs) of the proposed method and PiczakCNN. From Table 2, we can see that the total parameters and FLOPs of ACRNN are significantly lower than PiczakCNN, while ACRNN outperforms PiczakCNN in terms of classification accuracy. In addition, we also compare the computational complexity of the convolutional RNN (without attention) and ACRNN (with attention at $l_{10}$). The result shows that the model parameters of the two models are almost same with 3.81 million and the FLOPs of ACRNN is only 0.01 million more than convolutional RNN, which indicates that with the computational complexity almost unchanged, the proposed method can effectively improve the classification accuracy.

### 4.3. Effects of attention mechanism

To investigate the effect of the attention mechanism, we compare the results of proposed convolutional RNN with and without the attention mechanism. In Table 3, the results show that the attention mechanism delivers a significantly improved accuracy even when we use a data augmentation scheme. In addition, the data augmentation boosts an improvement of 2.0% and 4.8% on ESC-10 and ESC-50 datasets, respectively.

---

**Table 2**

Computational complexity of PiczakCNN and proposed ACRNN. *Params* denotes the number of model parameters and *FLOPs* denotes the number of floating point operations.

| Model                  | Params (M) | FLOPs (M) |
|------------------------|------------|-----------|
| PiczakCNN [19]         | 31.53      | 63.27     |
| convolutional RNN (Ours) | 3.81      | 9.17      |
| ACRNN (Ours)           | 3.81       | 9.18      |

---

**Figure 4:** Confusion matrix of ACRNN with an average classification accuracy 86.1% on ESC-50 dataset.
Table 3
Classification accuracy of proposed convolutional RNN with and without the attention mechanism. ‘augment’ denotes a combination of time stretch, pitch shift and mixup.

| Model Settings | ESC-10 | ESC-50 |
|----------------|--------|--------|
| no attention   | 93.0%  | 84.6%  |
| attention at $l_2$ (softmax) | 93.5%  | 85.2%  |
| attention at $l_2$ (sigmoid) | 93.5%  | 85.6%  |
| attention at $l_4$ (softmax) | 92.7%  | 83.8%  |
| attention at $l_4$ (sigmoid) | 93.5%  | 85.0%  |
| attention at $l_6$ (softmax) | 92.7%  | 84.4%  |
| attention at $l_6$ (sigmoid) | 93.0%  | 84.8%  |
| attention at $l_8$ (softmax) | 92.5%  | 84.9%  |
| attention at $l_8$ (sigmoid) | 93.2%  | 85.0%  |
| attention at $l_{10}$ | 93.7%  | 86.1%  |

Table 4
Classification accuracy of applying the attention mechanism to the output of different layers of the proposed convolutional RNN and using different scaling functions.

| Model Settings | ESC-10 | ESC-50 |
|----------------|--------|--------|
| no attention   | 93.0%  | 84.6%  |
| attention at $l_2$ (softmax) | 93.5%  | 85.2%  |
| attention at $l_2$ (sigmoid) | 93.5%  | 85.6%  |
| attention at $l_4$ (softmax) | 92.7%  | 83.8%  |
| attention at $l_4$ (sigmoid) | 93.5%  | 85.0%  |
| attention at $l_6$ (softmax) | 92.7%  | 84.4%  |
| attention at $l_6$ (sigmoid) | 93.0%  | 84.8%  |
| attention at $l_8$ (softmax) | 92.5%  | 84.9%  |
| attention at $l_8$ (sigmoid) | 93.2%  | 85.0%  |
| attention at $l_{10}$ | 93.7%  | 86.1%  |

4.4. Where and how to apply attention
In this section, we first investigate the classification performance when applying frame-level attention mechanism to the different layers of CNN and RNN. Specifically, we conduct experiments about applying frame-level attention mechanism to $l_2$, $l_4$, $l_6$, $l_8$ and $l_{10}$ layers of our convolutional RNN network. As shown in Table 4, our model obtains the highest classification accuracy and boosts an absolutely improvement of 0.7% and 1.5% when applying the attention mechanism at $l_{10}$ on both ESC-10 and ESC-50 datasets, respectively. For CNN layers, applying the attention mechanism at $l_2$ layer can obtain better classification accuracy than others. In addition, we observe that applying the attention mechanism at lower-layers ($l_2$ and $l_4$) obtain better performance than applying it at higher-level layers ($l_6$ and $l_8$). We argue that lower-level features usually keep the basic and useful characteristics of environmental sounds and the attention mechanism can help preserve them.

In addition, we evaluate the effect of scaling function on classification accuracy. Specifically, we select softmax and sigmoid function as scaling function and compare their classification accuracy. From Table 4 we can see that when we select softmax as scaling function, our model only obtains a slight improvement with the attention mechanism applied at $l_2$ and $l_8$ on ESC-50 dataset and $l_2$ on ESC-10 dataset. When applying attention mechanism at other CNN layers, the classification accuracy decreased. However, when we select sigmoid as scaling function, the proposed frame-level attention can always improve the classification accuracy on ESC-10 and ESC-50 datasets.

4.5. Visualization of attention
To have a better understanding how the proposed frame-level attention helps recognize different environmental sounds, we visualize the attention results for different sound classes (dog bark, crying baby, clock tick and rain). From Figure 5, we can see that the proposed frame-level attention mechanism is capable of focusing on the important temporal events while reducing the impact of background noise. For example, when our model tries to predict a dog bark, our frame-level attention will assign more weights on the semantically relevant frames, while de-weighting the semantically irrelevant or silent ones.

5. Conclusion
In this paper, we proposed an attention mechanism based convolutional recurrent neural network (ACRNN) for ESC. We investigated the frame-level attention mechanism for CNN layers and RNN layers. In addition, we compared the computational complexity of our proposed approach and existing work. Experimental results on ESC-10 and ESC-50 datasets demonstrated the effectiveness of the proposed method and achieved state-of-the-art or competitive classification accuracy with low computational complexity. We also compared the classification accuracy when applying different layers, including CNN layers and RNN layers. For CNN layers, we explored the effects of using different attention scaling function on classification accuracy. The experimental results showed that applying attention for RNN layers obtained highest accuracy and sigmoid worked better for generating attention weights than softmax when applying attention at CNN layers. Finally, we visualized the learned attention results to have a better understanding how attention works. While the proposed method achieves the promising results, the robustness to noise of the proposed method is not quantified in this paper. Therefore, in the future work, we will evaluate the robustness to noise of the proposed method with different types of noise with different levels of SNR.

References
[1] Aytar, Y., Vondrick, C., Torralba, A., 2016. Soundnet: Learning Sound Representations from Unlabeled Video, in: Proc. Int. Conf. Neural Inf. Process. Syst., pp. 892–900.

[2] Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473.

[3] Barchiesi, D., Giannoulis, D., Stowell, D., Plumbley, M.D., 2015. Acoustic Scene Classification: Classifying Environments from the Sounds They Produce. IEEE Signal Process. Magazine 32, 16–34.

[4] Bisot, V., Serizel, R., Essid, S., Richard, G., 2017. Feature Learning with Matrix Factorization Applied to Acoustic Scene Classification. IEEE/ACM Trans. Audio, Speech, Language Process. 25, 1216–1229.
Figure 5: Visualization of frame-level attention results with sigmoid as scaling function at $l_2$ layer. The first row represents the log gammatone spectrogram of dog bark, crying baby, clock tick and rain, the second row indicates the learned frame-level attention weights.