CROSS-TRIGGERING ISSUE IN AUDIO EVENT DETECTION AND MITIGATION

Huy Phan, Byeonggeun Kim, Vu Nguyen, Andrew Bydlon, Qingming Tang, Chieh-Chi Kao, Chao Wang

Amazon, Cambridge, MA 02142, USA
Amazon Central Machine Learning, Adelaide, SA 5000, Australia

ABSTRACT

Cross-triggering is a critical problem for applications of audio event detection (AED), particularly in low-resource settings. However, not much attention (if not none) has been paid to this problem in the AED research community. In this work, we tackle this problem via a regularization approach. We propose a regularizer, namely mutual exclusivity regularizer, that is able to enforce pairwise exclusivity between two event classes when they do not co-occur. When the regularizer is added to the loss function for network training, in effect, the increase in the score of one event class will result in the decrease of the other and vice versa. To quantify the effectiveness of the proposed regularizer, we developed an AED system based on convolutional neural network (CNN) for the detection of *hand clap* and *door knock*, two transient audio events that share similar spectro-temporal profiles, and conducted experiments on a large-scale real-world dataset (around 274.2 hours). The experimental results show that the proposed approach is able to largely mitigate the cross-triggering issue in various experimental settings. Furthermore, the reduction in cross-triggering, as a result, leads to improvement in the detection performance.

***Index Terms*—** Audio event detection, cross triggering, mutual exclusivity, regularization

1. INTRODUCTION

Cross-triggering issue [1, 2, 3] is common in audio event detection (AED) [4, 5, 6]. It happens in multi-label modeling [7, 8, 9] when the occurrence of one event will not only trigger the detector designated for that event type, but also mistakenly trigger detectors of other events. In general, audio events that have similar spectro-temporal characteristics are more likely to experience this issue. Fig. 1 gives an example on cross-triggering between two detectors for two audio events, *hand clap* and *door knock*, respectively (more details in Section 4). *Hand clap* and *door knock* are transient event types. That is, they both exhibit a sudden increase in signal level at the beginning, followed by an almost immediate decrease in signal level. In addition, their spectral distribution spans almost the entire spectrum of [0, 8] kHz. Thus, cross-triggering is expected. On the left-hand side of the figure, the door knock detector responds to the occurrences of hand claps in the audio recording, producing high detection scores. Similarly, on the right-hand side, the hand clap detector emits high detection scores in response to the occurrences of door knock events. The cross-triggering issue is even worse in resource-constrained AED systems due to design choices, for example, small model footprint with limited modeling capacity, shared parameters, and low-resolution feature input, which are typically constrained by the limited memory and computational resources.

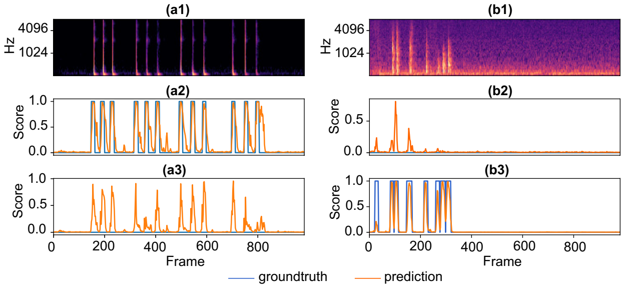

Fig. 1. An example of cross-triggering between *hand clap* and *door knock* detectors. On the left, (a1) shows an audio recording with occurrences of *hand clap* events; (a2) and (a3) show the detection scores produced by the *hand clap* detector and the *door knock* detector, respectively, on the audio recording in (a1) in comparison with the ground truth. On the right, (b1) shows an audio recording with occurrences of *door knock* events; (b2) and (b3) show the detection scores produced by the *hand clap* detector and the *door knock* detector on the audio recording in (b1) in comparison with the ground-truth.

Mitigation of the cross triggering issue is critical for many AED applications [5], for example smart home [10, 11], surveillance [12], health care [13, 14], where false alarms adversely affect users' experience. However, this issue has remained largely uncharted in the literature. Sajjadi *et al.* [2] proposed an unsupervised regularization term to enforce exclusivity between all possible label pairs, assuming that an example is only associated with one class (i.e, single-label modeling). This is, therefore, not suitable for multi-label modeling with co-occurrences of multiple classes. Inspired by [2], we propose a regularization method to tackle the cross-triggering issue. Different from [2], the proposed regularizer is formulated for multi-label modeling, but it should be generalize for both single-label and multi-label settings (we will show that [2] is a special case of our formulation). The main idea is to design a regularizer to enforce mutual exclusivity between any two events whenever they are not co-occurring. Due to the effect of the regularization, the increase of one event's detection score will lead to the decrease of the other event's detection score and vice versa. As a result, the cross triggering issue is mitigated.

We implemented an AED system based on convolutional neural network (CNN) for two events, *hand clap* and *door knock*, under the constraints of low-resource setting and conducted experiments on a large-scale internal dataset. The experimental results reveal strong presence of the cross-triggering issue and the proposed mutual exclusivity regularizer is able to substantially suppress the cross triggering rate, leading to improvement of the detection performance.

2. MUTUAL EXCLUSIVITY VIA REGULARIZATION

2.1. Mutual exclusivity regularizer (MER)

Without loss of generality, let $\mathcal{K}=\{1,2,\ldots,K\}$ denotes the label set of K events. Let $\mathbf{x}$ denote an audio example and $\mathbf{y}$ denotes its ground-truth. In general, any number of audio events can co-occur in $\mathbf{x}$ (i.e., polyphonic). $\mathbf{y}$ is, therefore, a multi-hot encoding vector of size K. Let $\mathcal{P}$ and $\mathcal{N}$ be the set of positive and negative labels associated with the example $\mathbf{x}$, respectively, where $\mathcal{K}=\mathcal{P}\bigcup\mathcal{N}$ and $\mathcal{P}\bigcap\mathcal{N}=\emptyset$. We can express mutual exclusivity between any positive-negative label pair as a Boolean function using disjunction of conjunctions:

$$f(\mathbf{y}) = \bigvee_{j\in\mathcal{P}}\left(\bigwedge_{k\in\mathcal{N}} \neg y_k \wedge y_j\right). \tag{1}$$

In order to optimize (1) with gradient descent, we need to approximate f with a differentiable function. To that end, we replace $\bigvee$ by $\sum$, and $\bigwedge$ by $\prod$. We also approximate the boolean operation $\neg y_k$ by $1-y_k$. We can relax and replace the binary vector $\mathbf{y}$ with the output of a multi-label network $\hat{\mathbf{y}} = F_{\mathbf{\Phi}}(\mathbf{x})$ whose elements are continuous value between 0 and 1. Here, $F_{\mathbf{\Phi}}$ denotes the multi-label network with the parameters denoted as $\mathbf{\Phi}$. We then can re-write (1) as

$$\begin{aligned} f(\mathbf{y}|\mathbf{x},\mathbf{\Phi}) &\approx -\sum_{j\in\mathcal{P}}\Big(\prod_{k\in\mathcal{N}} \hat{y}_j(1-\hat{y}_k)\Big) & (2)\\ &\approx -\sum_{j\in\mathcal{P}}\Big(\hat{y}_j \prod_{k\in\mathcal{N}}(1-\hat{y}_k)\Big). & (3)\end{aligned}$$

The term on the right-hand side of (3) defines the regularizer which can be optimized to enforce exclusivity between any positive-negative label pairs (i.e., they do not co-occur). We denote this mutual exclusivity regularizer as $\mathcal{R}_{me}$. The regularizer reduces to the one proposed in [2] when $|\mathcal{P}| = 1$ and $|\mathcal{N}| = K-1$, meaning there is only one positive class associated with the example $\mathbf{x}$.

Instead of optimize (3), we can optimize:

$$\begin{aligned} \mathcal{R}_{me} &= -\sum\nolimits_{j\in\mathcal{P}}\big(\hat{y}_j \sum\nolimits_{k\in\mathcal{N}}\log(1-\hat{y}_k)\big) & (4)\\ &= -\sum\nolimits_{j\in\mathcal{P}}\sum\nolimits_{k\in\mathcal{N}}\hat{y}_j\log(1-\hat{y}_k). & (5)\end{aligned}$$

Please note that both (3) and (5) are minimized when $\hat{y}_j = 1, \forall j\in\mathcal{P}$ and $\hat{y}=0, \forall k\in\mathcal{N}$. This regularizer will be weighted by a hyper-parameter λ and added to the loss function for network training.

2.2. Implementation in practice

Given the ground-truth multi-hot encoding vector $\mathbf{y}\in\{0,1\}^K$ and the network output $\hat{\mathbf{y}}\in[0,1]^K$ of the input $\mathbf{x}$, we will first compute two outer products $\mathbf{Y}_\otimes$ and $\hat{\mathbf{Y}}_\otimes$:

$$\begin{aligned} \mathbf{Y}_\otimes &= \mathbf{y}(\mathbf{1}-\mathbf{y})^\mathsf{T}, & (6)\\ \hat{\mathbf{Y}}_\otimes &= \hat{\mathbf{y}}\log(\mathbf{1}-\hat{\mathbf{y}})^\mathsf{T}, & (7)\end{aligned}$$

where $\mathbf{1}$ denotes the vector of all 1s. While $\hat{\mathbf{Y}}_\otimes$ reflects the computation of the regularizer in (5), $\mathbf{Y}_\otimes$ is used as the masking matrix, where $\mathbf{y}_\otimes(j,k) = 0$ if both classes j and k are present in the example $\mathbf{x}$ (i.e., co-occurring events) and $\mathbf{y}_\otimes(j,k) = 1$ otherwise. The regularizer can then be computed by element-wise product of $\mathbf{Y}_\otimes$ and $\hat{\mathbf{Y}}_\otimes$, followed by taking grand sum of the resulting matrix:

$$\mathcal{R} = \mathbf{1}(\mathbf{Y}_\otimes \odot \hat{\mathbf{Y}}_\otimes)\mathbf{1}^\mathsf{T}. \tag{8}$$

In practice, network training with a mini-batch of data will add an extra dimension (i.e., batch dimension) to $\mathbf{y}$ and $\hat{\mathbf{y}}$. However, the core idea for the implementation remains the same. In `Pytorch`, it can be as convenient as

```
import torch as t
Y = t.einsum('ki,kj->kij', y, 1-y)
Yh = t.einsum('ki,kj->kij', yh, t.log(1-yh))
R = -t.sum(t.mul(Yh, Y))
```

And so can `Tensorflow` implementation:

```
import tensorflow as tf
Y = tf.einsum('ki,kj->kij', y, 1-y)
Yh = tf.einsum('ki,kj->kij', yh, tf.log(1-yh))
R = -tf.reduce_sum(tf.math.multiply(Yh, Y))
```

3. AED SYSTEM

In this section, we will describe the AED system developed for detection of *hand clap* and *door knock*. We deliberately make different design choices for the CNN architecture, including small number of input features, small network architecture with shared layers to take into account the limited memory and computational resources.

3.1. Input

Given an audio signal, sampled at 16 kHz, we divide it into 25 ms frames with 10 ms overlap. We then extract $F = 20$ Log Mel-Filterbank Energy (LFBE) features to represent each frame. The network ingests a segment $\mathbf{x}$ of $T = 11$ audio frames as input, i.e., $\mathbf{x}\in\mathbb{R}^{T\times F}$.

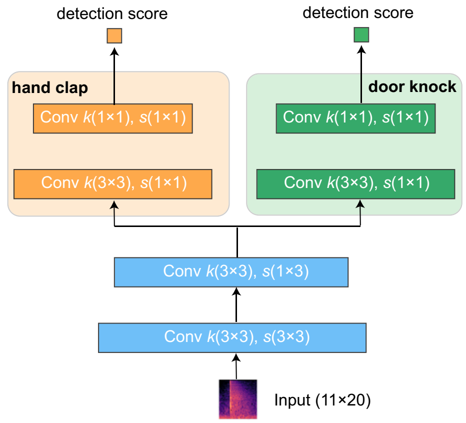

Fig. 2. The CNN for *hand clap* and *door knock* detection.

3.2. Network architecture

The network architecture is illustrated in Fig. 2. It features two 2D convolutional layers shared between the two events with the common kernel size of 3×3. The stride is configured to 2×2 and 1×3 and the number of kernels is set to 24 and 32 for the first and second layer, resulting in feature maps of size $5 \times 9 \times 24$ and $3 \times 3 \times 32$, respectively.

After the two shared convolutional layers, the network is split into two branches, one for each event, with similar architecture to learn event-specific features for classification. Each branch has two convolutional layers. The first convolutional layer is designed to have 40 kernels, the kernel size of 3×3, and the stride of 1×1, producing a feature map of size $1 \times 1 \times 40$. The second convolutional layer is a 1×1 convolutional layer with a single kernel to reduce the previous layer's multi-channel feature map into a single output.

All the convolutional layers in the CNN are followed by batch normalization [15] and ReLU activation [16], except for the last 1×1 convolutional layer in the two branches whose activation is sigmoid for classification purpose. In general, since *hand clap* and *door knock* may happen at the same time, sigmoid activation is used to cope with multi-label classification. The network has 30,698 parameters in total.

3.3. Loss function

Given the groundtruth multi-hot encoding vector $\mathbf{y} \in \{0, 1\}^K$ and the network output $\hat{\mathbf{y}} \in [0, 1]^K$, $K = 2$ in this case, the network is trained to minimize the following loss function

$$\mathcal{L} = BCE(\mathbf{y}, \hat{\mathbf{y}}) + \lambda \mathcal{R}_{me}(\mathbf{y}, \hat{\mathbf{y}}), \tag{9}$$

where λ is a hyper-parameter trading-off between the binary cross-entropy loss (BCE) and the mutual exclusivity regularizer $\mathcal{R}_{me}$. Furthermore, we revise the BCE loss to penalize false positives and false negatives differently:

$$BCE(\mathbf{y}, \hat{\mathbf{y}}) = \sum_k (\beta_{FP} y_k \log \hat{y}_k + \beta_{FN}(1-y_k)\log(1-\hat{y}_k)) \, , \tag{10}$$

Table 1. Summary of the experimental dataset.

Partition	#recording (background)	#recording (w/ hand clap)	#recording (w/ door knock)	#hand clap event	#door knock event
Training	44,457	8,050	4062	60,706	62,659
Valid.	13,490	967	462	8,310	6,431
Test	17,433	929	485	6,431	7,155
Total	83,743	9,946	5,009	75,447	76,245

where β_{FP} and β_{FN} are the penalizing weights for false positives and false negatives, respectively.

3.4. Post-processing

At inference time, the trained network is applied on a test audio recording in sliding window fashion with a shift of 2 frames, producing two detection scores every 20 ms, one for *hand clap* and one for *door knock*. A median filter of size M is used to smooth the detection scores before applying a threshold τ to determine the detected events.

4. EXPERIMENTS

4.1. Data

We conducted our experiments on an Amazon internal de-identified dataset. The dataset consists of 98,698 recordings of 10s audio (approximately 274.2 hours), out of which 9,946 have the presence of *hand claps*, 5,009 have the presence of *door knocks*, and 83,743 containing rich background noises and a variety of environmental sounds like speech, appliance beeping, smoke alarm, dog barking, etc. The dataset was partitioned into training, validation, and test set which are summarized in Table 1. *hand clap* and *door knock* events were strongly labeled, i.e., each event was labeled with their identity, onset time, and offset time.

4.2. Parameters

The network was trained for 30 epochs with a batch size of 512. During training, the network was validated on the validation set and the snapshot obtaining the best performance on the validation set was retained for testing. Regarding the parameters associated with the BCE loss function in (10), we fixed $\beta_{FP} = 1.0$ and $\beta_{FN} = 2.0$. For post-processing, the median filter size M was set to 3.

Using the case with the trade-off hyper-parameter $\lambda = 0.0$ (i.e., without regularization effect, see (9)) as the baseline, we examined how λ affects the cross triggering rate by varying its value in $\{0.01, 0.1, 1.0, 5.0, 10.0\}$. We also varied the detection threshold τ in $\{0.5, 0.55, \ldots, 0.95\}$ to examine how the detection performance varies with respect to it.

4.3. Evaluation metrics

We used F1-score as the overall detection performance metric here. Note that false accept rate (FAR) and true positive rate (TPR) could also be used alternatively [5] for the purpose.

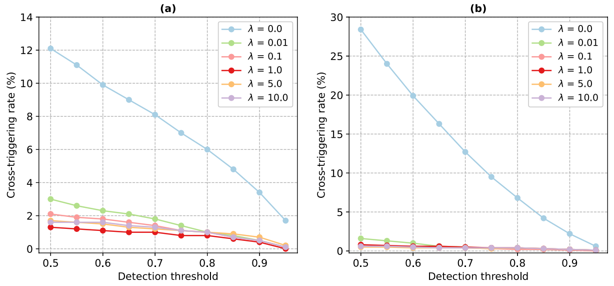

Fig. 3. Reduction in cross-triggering rate resulted with different values of λ in comparison to the baseline ($\lambda = 0$). (a) *hand clap* and (b) *door knock*. Note that cross-triggering rate on y-axis is relative.

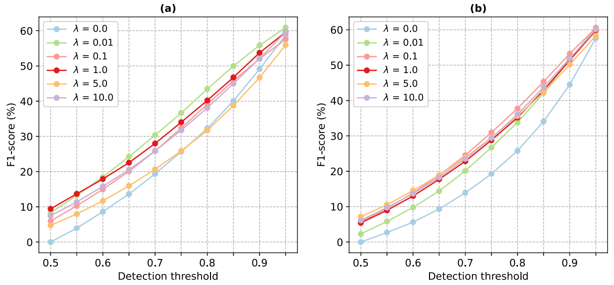

Fig. 4. F1-score improvement obtained with different values of λ in comparison to the baseline ($\lambda = 0$). (a) *hand clap* and (b) *door knock*. Note that F1-score on y-axis is relative.

In addition, we computed the cross-triggering rate to quantify the severity of cross-triggering issue. The cross-triggering rate of a detector is defined as ratio between the number of events from the other class mistakenly detected by the detector and the total number of detected events.

4.4. Experimental results

Fig. 3 shows the cross-triggering rate induced by the two detectors with different values of λ. Overall, it can be observed from the baseline that the cross-triggering rate is naturally mitigated with the increase of the detection threshold. The mutual exclusivity regularizer is able to reduce the cross-triggering rate even more. The reduction is consistently observed on both hand clap and door knock. For example, at the lowest detection threshold $\tau = 0.5$, the regularizer with $\lambda = 10.0$ suppresses the hand clap detector's cross-triggering down to 1.6% from 12.1% of the baseline, and the door knock detector's cross-triggering rate down to 0.6% from 28.4%. At the highest detection threshold $\tau = 0.95$, the hand clap detector's cross-triggering rate is reduced to 0.1% from 1.7% of the baseline and the door knock detector's cross-triggering rate is reduced to 0.1% from 0.6% of the baseline.

To illustrate the effect of regularization with different values of λ, we visualize in Fig. 5 the detection scores produced by the one detector on an audio recording with event occurrences of the other class. As expected for both cases, the increasing effect of regularization with increasing value of λ helps to gradually diminish the detection score caused by the event occurrences of the other class.

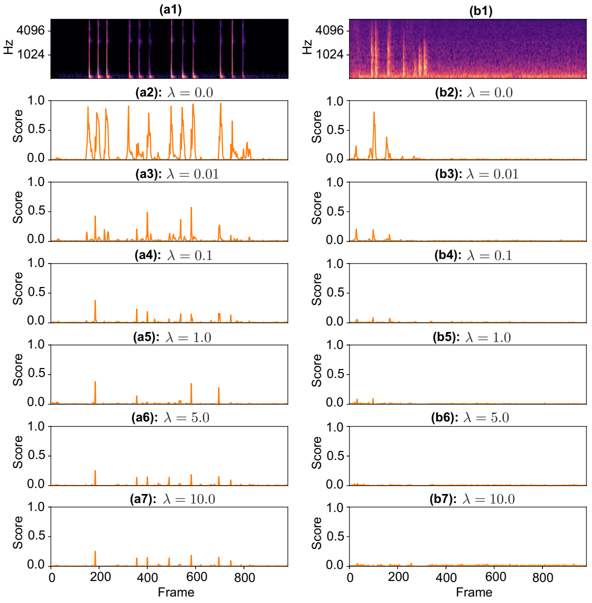

Fig. 5. The effect of λ on cross-triggering mitigation. On the left, (a1) shows an audio recording with occurrences of *hand clap* events; (a2)-(a7) show the *door knock* detector's detection scores produced on the audio recording in (a1) with different values of λ. On the right, (b1) shows an audio recording with occurrences of *door knock* events; (b2)-(b7) show the *hand clap* detector's detection scores produced on the audio recording in (b1) with different values of λ.

At the same time, using the mutual exclusivity regularizer also results in improved detection performance. We show in Fig. 4 the F1-scores obtained by the two detectors with different values of λ. From the figure, a consistent pattern is seen for the to detectors that F1-score is improved with the use of the regularizer. Since cross-triggers constitute a part of false positives, the reduction of cross-triggers is expectedly translated into the reduction in the number of false positives and improves the precision, and then, F1-score as a result.

5. CONCLUSION

We presented in this work a regularization approach to mitigate the cross-triggering issue in AED systems. The employment of the proposed mutual exclusivity regularizer in network training in effect enforces two event classes to be mutual exclusive, i.e., the increase in the detection score of one class will cause the detection score of the other class to decrease. The experimental results on the large-scale real-world dataset for *hand clap* and *door knock* detection showed that using the proposed regularizer not only helps to largely diminish the cross-triggering rate but also leads to improvement of the detection performance. These results are consistent across all the studied experimental settings.

6. REFERENCES

[1] Andreas Panteli, Jonas Teuwen, Hugo M. Horlings, and Efstratios Gavves, "Sparse-shot learning with exclusive cross-entropy for extremely many localisations," in *Proc. ICCV*, 2021, pp. 2793–2803.

[2] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen, "Mutual exclusivity loss for semi-supervised deep learning," in *Proc. ICIP*, 2016, pp. 1908–1912.

[3] Ming-Kun Xie and Sheng-Jun Huang, "Partial multi-label learning with noisy label identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3676–3687, 2021.

[4] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D. Plumbley, "Real-world acoustic event detection," *IEEE Signal Process. Mag.*, vol. 38, no. 5, pp. 67–83, 2021.

[5] Çagdas Bilen, Giacomo Ferroni, Francesco Tuveri, Juan Azcarreta, and Sacha Krstulovic, "A framework for the robust evaluation of sound event detection," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, pp. 61–65, IEEE.

[6] Huy Phan, Oliver Y Chén, Philipp Koch, Lam Pham, Ian McLoughlin, Alfred Mertins, and Maarten De Vos, "Unifying isolated and overlapping audio event detection with multi-label multi-task convolutional recurrent neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 51–55.

[7] Grigorios Tsoumakas and Ioannis Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[8] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 5, no. 6, pp. 1291–1303, 2017.

[9] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi, "General multi-label image classification with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16478–16488.

[10] Anastasios Vafeiadis, Konstantinos Votis, Dimitrios Giakoumis, Dimitrios Tzovaras, Liming Chen, and Raouf Hamzaoui, "Audio content analysis for unobtrusive event detection in smart homes," *Engineering Applications of Artificial Intelligence*, vol. 89, no. 103226, 2020.

[11] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon, "Sound event detection in synthetic domestic environments," in *Proc. ICASSP*, 2020, pp. 86–90.

[12] Rosa Maria Alsina-Pagès, Joan Navarro, Francesc Alías, and Marcos Hervás, "homesound: Real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring," *Sensors*, vol. 17, no. 4, pp. 854, 2017.

[13] L. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. McLoughlin, "CNN-MoE based framework for classification of respiratory anomalies and lung disease detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2938–2947, 2021.

[14] Athanasios Lykartsis, Markus Hädrich, and Stefan Weinzierl, "Sound event detection in synthetic domestic environments," in *Proc. EUSIPCO*, 2021, pp. 980–984.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.