# Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks

Emre Cakir, Toni Heittola, Heikki Huttunen and Tuomas Virtanen

***Abstract*— In this paper, the use of multi label neural networks are proposed for detection of temporally overlapping sound events in realistic environments. Real-life sound recordings typically have many overlapping sound events, making it hard to recognize each event with the standard sound event detection methods. Frame-wise spectral-domain features are used as inputs to train a deep neural network for multi label classification in this work. The model is evaluated with recordings from realistic everyday environments and the obtained overall accuracy is 63.8%. The method is compared against a state-of-the-art method using non-negative matrix factorization as a pre-processing stage and hidden Markov models as a classifier. The proposed method improves the accuracy by 19% percentage points overall.**

## I. Introduction

Sound event is the audio segment that humans would label as a distinctive concept in an acoustic signal [1]. The aim of automatic sound event detection is to recognize the sound events present in a continuous acoustic signal. Monophonic sound event detection deals with the most prominent event at a time instance and polyphonic detection tackles the situations where multiple sound events happen simultaneously. The applications of sound event detection include multimedia indexing [2], scene recognition for mobile robots [3] and surveillance in living environments [4].

The additive nature of sound sources makes it difficult to find the robust features to detect them in polyphonic audio. Conventional classifiers that have been used in speech recognition and monophonic detection are not as successful in polyphonic detection. Monophonic sound event detection systems handle the polyphonic data by detecting only the prominent event, resulting with a loss of information in realistic environments [5]. Polyphonic detection is essential to get high accuracy in complex auditory scenes. State-of-the-art polyphonic detection systems are using Mel-Frequency Cepstral Coefficients (MFCC) to characterize the audio signals and using Hidden Markov Models (HMMs) as classifiers with consecutive passes of the Viterbi algorithm [6]. Recently, non-negative matrix factorization (NMF) was used as a pre-processing step to decompose the audio into streams and detect the most prominent event in each stream at a time [1]. However, the fixed constraint of the NMF on the number of overlapping events reduces its practicality when this number is not known *a priori*. The estimation of the number of overlapping events can be bypassed when using coupled NMF, as shown in [7]. In [8], local spectrogram features were combined with Generalized Hough Transform (GHT) voting system to detect the overlapping sound events. This offers a different path than traditional frame-based features and achieves high accuracy, being evaluated on five different sound events and their combinations.

Polyphonic detection can be formulated as a multi label classification problem. Multi label problem can be addressed by applying single label classification for each of the classes and combining the results. However, the single label encoding of the problem discards the correlation structure between the classes resulting in a weak expressive power [9]. Therefore, multi label classification is necessary to obtain the most of the available information from the real-world data. Some of the applications of multi label classification are used in overlapping sound event recognition [8], scene classification [10] and text categorization [11].

In this paper, we propose to use multi label feed-forward deep neural networks (DNN) for polyphonic sound event detection. In our earlier paper, we have shown that with sufficient numbers of hidden layers, hidden units and training data, DNNs can outperform HMM methods in sound event classification tasks [12]. However, in this paper we extend the work of [12] by encoding the problem as a multi label learning task with no limitations to the number of simultaneous events. the motivation of our work is that DNN can use different sets of its hidden units to model multiple simultaneous events in a given time instance, benefiting from a different nature of nonlinearity than the conventional mixture models [13]. We use spectral domain features to characterize the audio signals and DNNs to learn a mapping between features and sound events. The contribution of this paper is to extend the use of DNNs to the multi label analysis of realistic recordings from everyday environments and model overlapping sound events in a natural way. We also propose a post-processing method to filter the noise in the DNN outputs. The highly realistic and diverse audio material used in this work offers a firm insight on the usability of the method in real-world applications.

The structure of the paper is as follows: the task of the polyphonic sound event detection and the feature extraction process are explained in Section 2. The input-output structure and the architecture of the DNN is described in Section 3. A post-processing method to smoothen the DNN output is explained in Section 4. Section 5 contains the experimental results on the highly realistic material and comparison with the baseline results. In the end, our conclusions on the topic are given in Section 6.

*This work was done with the support of Audio Research Group in Department of Signal Processing, Tampere University of Technology, Finland.

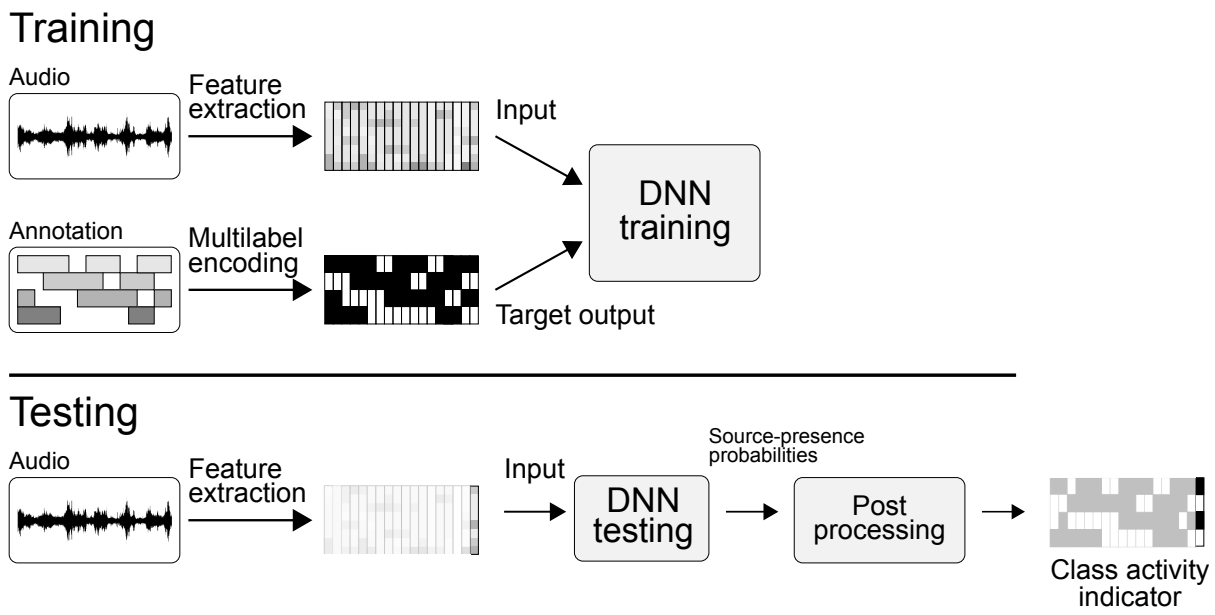


Fig. 1: Framework of the training and testing procedure for the proposed system.

## II. Sound Event Detection

The main objective of the proposed method is to temporally locate the sound events in a recording collected from a realistic auditory scene and give each event a label from a set of possible labels. The framework of the proposed sound event detection method is shown in Figure 1.

Auditory scenes are composed of multiple sound events occurring at the same time. Detecting the events separately from a realistic auditory scene leads to a multi label classification problem. Figure 2 illustrates the polyphonic nature of sound events in realistic environments.

As a pre-processing step for the feature extraction, the recordings are amplitude normalized, divided into frames and Hamming window with 50 ms duration and 50% overlap is applied. The spectral domain features (*e.g.* Mel-band and log Mel-band energies) and cepstral domain features (*e.g.* MFCCs) are extracted from the short time frames of the audio signal. For each time frame, a feature vector $\mathbf{x}_t$ is obtained, where $t$ is the frame index. Each feature vector corresponds to a learning instance for the neural network.

In order to extract the dynamic property information of the signal, a frame concatenation method is used. The feature vectors that are extracted from the adjacent time frames are concatenated together to form a single training instance. The resulting feature vector has a dimension of $|\mathbf{x}| = (2 \times N_{\text{adj}} + 1) \times N_{\text{f}}$ where $N_{\text{adj}}$ is the number of adjacent frames concatenated with the original frame and $N_{\text{f}}$ is the number of features extracted from the short time frame. This method is often called *context windowing* and has been used in many other studies as well [12], [14], [15].

For each frame, target output vector $\mathbf{y}_t$ includes the multi label encoding of the audio events present in the frame. Each sound event is assigned to a class which is encoded as a single binary variable. The events present in a frame are annotated with 1 and the rest is 0. An illustrative example of annotation can be found in Figure 2, where the target output vector $\mathbf{y}$ for frame $t$ is $\mathbf{y}_t = [1\ 0\ 1]$. The number of possible classes is known in advance and therefore the length of the output vector is fixed, but the number of active events in a frame is not known *a priori*.

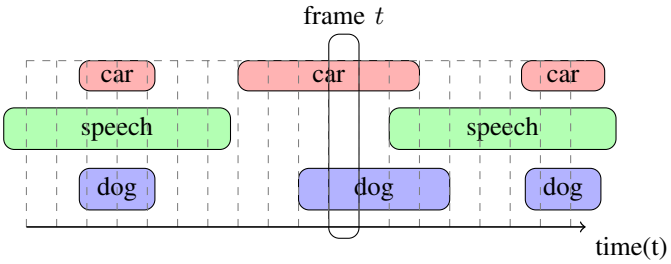


Fig. 2: Overlapping sound events in a recording from a realistic environment. Frame $t$ represents the short time frame from the recording where only car and dog barking events are present.

## III. Multi Label Neural Network

Feed-forward neural networks with multiple hidden layers, *i.e.*, deep neural networks (DNN) are used for multi label classification. Deep architectures build a hierarchy among the features. In each layer, higher level features are extracted implicitly by the composition of lower level features. This automatic structure eases the work of learning highly non-linear functions mapping the input to the output directly from data, therefore reducing the need to find human-crafted intermediate representations [16].

DNNs are composed of an input layer, multiple layers of hidden units with nonlinear activation functions and an output layer. The input vector $\mathbf{x}_t$ consists of the spectral features

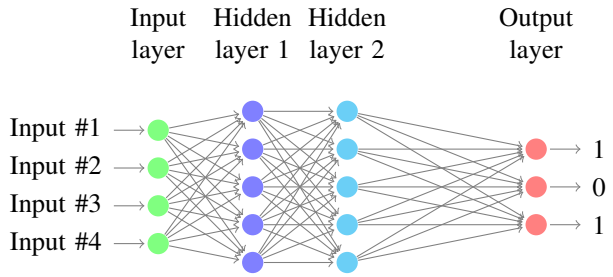


Fig. 3: Symbolic representation of the NN topology with two hidden layers and multi label outputs.

extracted from frame $t$. For simplicity, frame index $t$ will be omitted during the feed-forward algorithm description.

The output vector $\mathbf{h}^k \in \mathbb{R}^M$ for layer $k$ with $M$ units is calculated from the weighted sum of the outputs for the previous layer $\mathbf{h}^{k-1} \in \mathbb{R}^D$. Starting with $\mathbf{h}^0 = \mathbf{x}$ and

$$\mathbf{g}^k = \mathbf{W}^k \mathbf{h}^{k-1} + \mathbf{b}^k, \; 1 \leq k < L \tag{1}$$

$$\mathbf{h}^k = f(\mathbf{g}^k) \tag{2}$$

where $\mathbf{W}^k \in \mathbb{R}^{D \times M}$ is the weight matrix between $(k-1)^{\text{th}}$ layer with $D$ units and $k^{\text{th}}$ layer with $M$ units, $\mathbf{b}^k \in \mathbb{R}^M$ is the bias vector for the $k^{\text{th}}$ layer, $L$ is the number of layers and $f(\cdot)$ is the nonlinear activation function applied element-wise. Output $\mathbf{h}^L \in \mathbb{R}^N$ is used as the source presence prediction vector $\hat{\mathbf{y}} = \mathbf{h}^L$, where $\hat{\mathbf{y}}(i)$ is the source presence prediction for the event $i \in [1, 2, ..N]$ and $N$ is the number of sound events. During the training stage, $\hat{\mathbf{y}}$ is involved in calculating the cost function, explained below in detail. During the testing stage, $\hat{\mathbf{y}}$ is binarized with a threshold to get the binary detection vector $\mathbf{z}_t$. An illustration of a small-scale DNN with binarized output is presented in Figure 3.

Maxout function [17] for hidden layers and logistic sigmoid function (bounded between 0 and 1) for output layer are used as activation functions in DNN structure. Maxout is a piecewise linear activation function which can be seen as a generalization of rectified linear units [18]. Maxout calculates the maximum of a set of $R$ affine projections of the input. In mathematical terms, given $\mathbf{g}^k = [g^k(0), g^k(1), ..., g^k(j), ...g^k(M \times R - 1)]$, for non-overlapping pools of size $R$, maxout function implements

$$\mathbf{h}^k(i) = \max_{r=0}^{R-1} \mathbf{g}^k(j + r) \quad \text{where } j = i \cdot R \tag{3}$$

where $\mathbf{h}^k \in \mathbb{R}^M$, $\mathbf{g}^k \in \mathbb{R}^{M \times R}$ for layer $k$ with $M$ units and $R$ is the number of affine feature mappings. Hidden units with maxout functions at each layer are divided into non-overlapping pieces and each piece generates a single activation via the max pooling operation, as illustrated in Figure 4. Unlike conventional optimization functions, maxout is not bounded, it is easier to optimize and does not suffer from vanishing gradients problem by sparsifying the gradients [14].

Stochastic gradient descent (SGD) algorithm is used as the learning algorithm for the DNN. Training cost function for the neural network is selected as Kullback-Leibler (KL) divergence, as it is able to characterize the general accuracy of the class membership probabilities [19]. KL divergence is calculated as

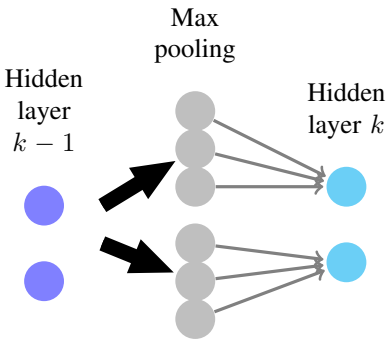


Fig. 4: Maxout activation function with $R = 3$ feature mappings.

$$\begin{aligned} KL(\mathbf{y}_t || \hat{\mathbf{y}}_t) = \sum_{i=1}^{N} & \mathbf{y}_t(i) \log \mathbf{y}_t(i) \\ & - \mathbf{y}_t(i) \log \hat{\mathbf{y}}_t(i) \\ & + (1 - \mathbf{y}_t(i)) \log (1 - \mathbf{y}_t(i)) \\ & - (1 - \mathbf{y}_t(i)) \log (1 - \hat{\mathbf{y}}_t(i)) \end{aligned} \tag{4}$$

where $\mathbf{y}_t(i)$ is the target output for the $i^{th}$ event, $\hat{\mathbf{y}}_t(i)$ is the source presence prediction obtained from the output layer for the $i^{th}$ event and $N$ is the total number of event classes. For binary $\mathbf{y}_t$, as in our case, some terms in (4) drop out and the resulting KL divergence is

$$\begin{aligned} KL(\mathbf{y}_t || \hat{\mathbf{y}}_t) = \sum_{i=1}^{N} & -\mathbf{y}_t(i) \log \hat{\mathbf{y}}_t(i) \\ & -(1 - \mathbf{y}_t(i)) \log (1 - \hat{\mathbf{y}}_t(i)) \end{aligned} \tag{5}$$

The DNN parameters such as number of hidden units, learning rate, initial weight bias etc. are selected by a grid search over the parameter values. The instances are processed over mini batches of size 50. The most successful topology for this task is found to be DNN with 2 hidden layers with 800 units each.

## IV. Post-processing

Environmental sound events naturally take at least a few seconds, once they are initiated. When we experimented with environmental audio, we noticed some abrupt changes between consecutive frames in the detection probabilities for some of the events. Our reasoning to this is as follows. The audio is processed in very short time frames and the events may contain intermittent periods. The annotation of the audio material is done with a rather coarse time resolution, since a human annotator would miss these less (if any) active frames in the events and do the annotation for larger chunks of frames. Although these frames are erroneously annotated with some of the labels, they do not have the spectral characteristics of the labels associated with them. Moreover,

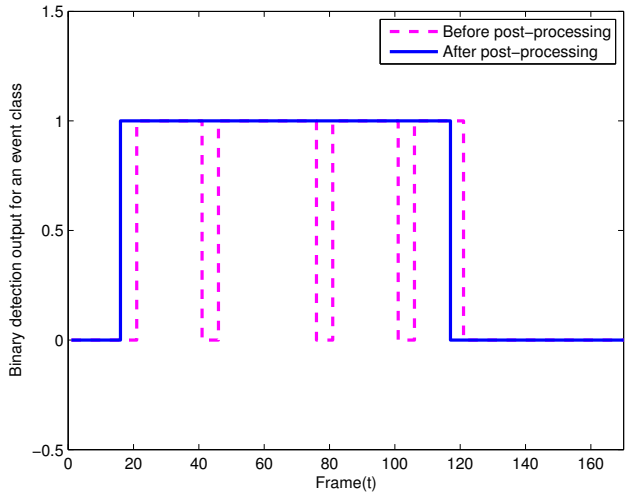


Fig. 5: Median filtering based post-processing method.

a greater problem occurs if the DNN *learns* these instances as belonging to a specific class with rather noise-like spectrum, such as the sound of rain or the wind on trees.

This causes some undesired intermittent behavior and *noise* in the DNN detection probabilities.

In order to filter this noise and smoothen the outputs in the testing stage, a median filtering based post-processing approach is implemented. The source presence predictions $\hat{\mathbf{y}}_t$ are obtained from the output layer of the DNN and then binarized by using a certain threshold value to give the binary estimation vector $\hat{\mathbf{z}}_t$. For each frame, the post-processed output $\hat{\mathbf{z}}_t$ is obtained by taking the median of the binary outputs in a 10-frame window as

$$\hat{\mathbf{z}}_t = \begin{cases} 1, \text{median}(\mathbf{z}_{(t-9):t}) = 1 \\ 0, \text{otherwise} \end{cases} \tag{6}$$

The method is continued by sliding this 10-frame window by 1 when every new frame is processed through the DNN. The effect of the median filtering on the detection outputs is illustrated in Figure 5.

## V. Evaluation

The proposed method is evaluated on realistic recordings from everyday contexts and compared with the baseline system. In addition, three different features are experimented individually: Mel-band energies, log Mel-band energies and MFCCs. The accuracy for various polyphony levels and the effect of post-processing is also investigated.

### A. Acoustic Data

The evaluation sound database contains recordings from various everyday environments. The same database has been previously used in different experiments on sound event detection [1], [5], [6]. It consists of a total of 103 recordings and each of them are 10 to 30 minutes long. The total duration of the recordings is 1133 minutes. Recordings were done using 44.1 kHz sampling rate and 24-bit resolution.

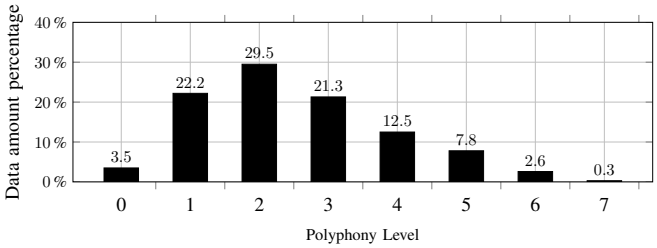


Fig. 6: The percentage of the amount of the sound material as a function of the polyphony level.

The recordings are collected from 10 different contexts: basketball match, beach, public bus, car, hallway, office, restaurant, shop, street and stadium. For each context, 8 to 14 recordings are present.

There are 61 different event classes categorized in this database. The start and end times of the events are manually annotated from the recordings. Some of the events included in the database are "brakes squeaking", "cheering", "referee whistle" etc. In each context, 9 to 16 events are present. Some of the events can be found in multiple contexts (e.g. "speech") and some of the events are context specific (e.g. "ball hitting the floor"). The total duration of each event in the database can be found in Figure 7. This database is a valuable source considering the lack of publicly available environmental polyphonic sound databases in the field. Figure 6 illustrates the amount of frames with different polyphony levels in the whole database.

### B. Evaluation Procedure

As the evaluation metric, F1 score is calculated inside non-overlapping one-second blocks. If an event

- is detected in one of the instances inside a block and it is also present in the same block of the annotated data, that event is regarded as correctly detected.
- is *not* detected in any of the instances inside a block but it is present in the same block of the annotated data, that event is regarded as missed.
- is detected in one of the instances inside a block but it is *not* present in the same block of the annotated data, that event is regarded as false alarm.

For each one-second block, the number of correct, missed and false alarm events are accumulated. Precision and recall are calculated according to these variables as

$$precision = \frac{correct}{correct + false\ alarm} \tag{7}$$

$$recall = \frac{correct}{correct + missed} \tag{8}$$

For each block, these two metrics are combined as their harmonic mean, *F1 score*, which can be formulated as

$$F1\ score = \frac{2 \times precision \times recall}{precision + recall} \tag{9}$$

The results are presented by taking the average F1 scores of the one second blocks which correspond to the specific

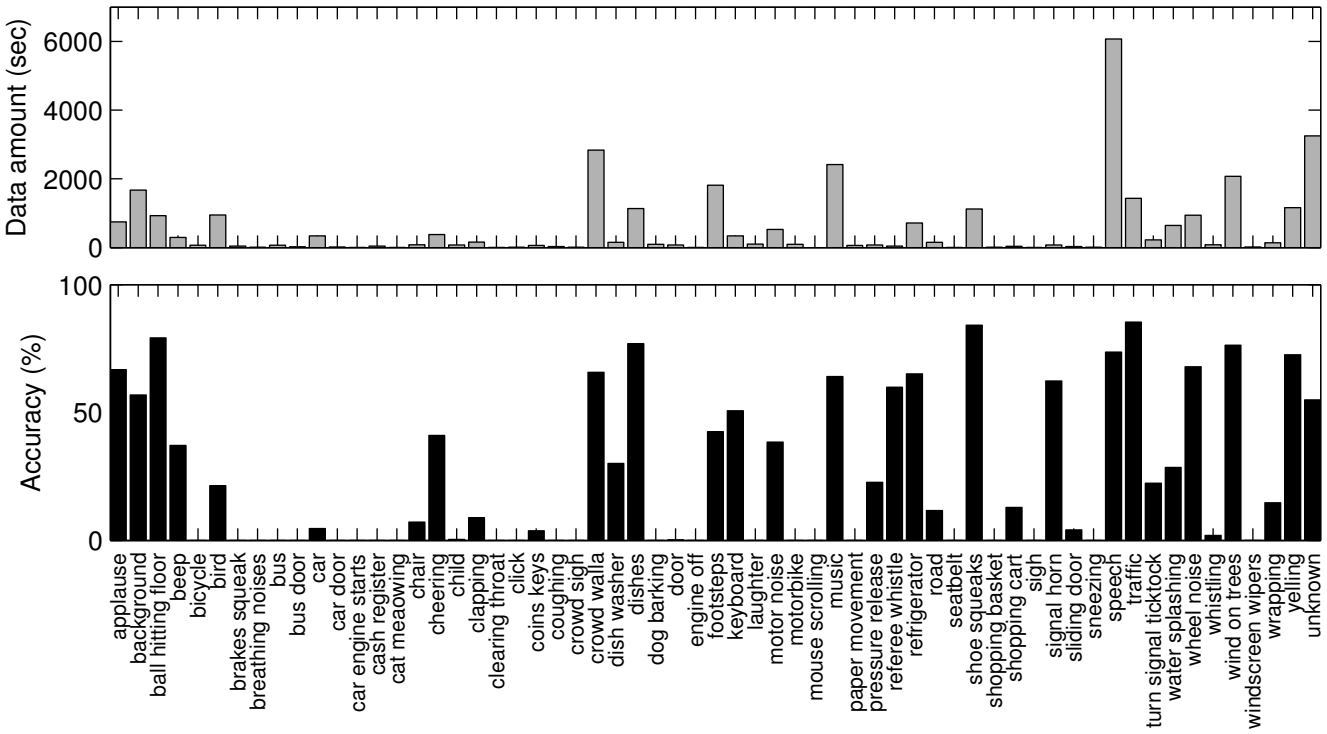


Fig. 7: The amount of annotated data (in seconds) and the accuracy for each sound event.

concept (context, polyphony level etc.) F1 score is referred as the *accuracy* throughout the rest of the paper.

Calculating the accuracy in one-second blocks is plausible for three different reasons. Firstly, the aim of the sound event detection is to detect an event with certainty when it happens, rather than finding the exact start and end time of the event with very high precision. Secondly, monitoring the outputs in every second and calculating the accuracy gives a reasonable time resolution without losing crucial information. Lastly, as noted in Section IV, the annotations have rather coarse time resolution. Therefore, in some cases they do not exactly match the time frames that they are annotated with, but they are nevertheless found in a one-second range. One-second block evaluation helps to compensate these minor mismatches in the annotations.

### C. Results

The proposed system is evaluated with stratified five-fold cross-validation. Once the features are extracted from all the recordings in the database, the feature data set is divided into five non-overlapping folds and one fold is used in the development stage for determining parameters of the DNN. The results from the other folds are averaged and presented. The grid search range and the final selected value for some essential DNN parameters during the development stage are presented in Table I. Log Mel-band energies are used as features in all the experiments, except the varying feature experiment given in Table II.

Our system, DNN with 2 hidden layers of 800 units each, log Mel-band energy features with 5-frame context

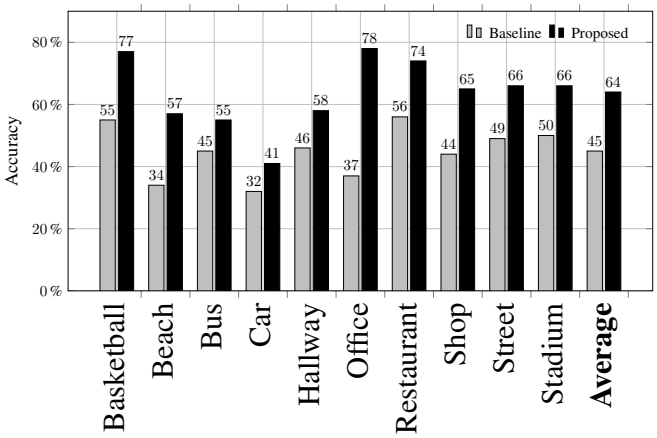


Fig. 8: Context-wise detection accuracies for proposed system with a comparison to the baseline system.

TABLE I: Grid search range for essential DNN parameters and the final values used in the experiments.

| Parameter | Range | Final |
|---|---|---|
| Learning rate | 0.001 - 1.0 | 0.02 |
| # hidden layers | 2 - 5 | 2 |
| # hidden units (in each layer) | 100 -1000 | 800 |
| Initial weight range | $\pm$0.001 - $\pm$0.05 | $\pm$0.001 |
| Context window length | 1 - 13 | 5 |

TABLE II: Overall detection accuracies with different input features before and after post-processing (PP).

| Feature | Before PP | After PP |
|---|---|---|
| MFCCs | 53.5 | 56.8 |
| Mel-band energies | 56.0 | 60.4 |
| Log Mel-band energies | 57.2 | 61.7 |

window, is evaluated against the baseline system [1], which is the state-of-the-art method for the polyphonic detection. The baseline method consists of decomposing the audio into different streams by non-negative matrix factorization. For each audio stream, sound event detection is done using MFCCs as features and HMM as a classifier. In our experiments, we observed that DNNs do not necessarily require this kind of sound source separation based pre-processing to determine how many events are active in a time instance. As illustrated in Figure 8, the proposed system outperforms the baseline method by a huge margin. Depending of the context, proposed method offers an increase in accuracy between 9-39% among different contexts and 19% units average increase. Due to the natural diversity of each context, the variance of the accuracy between contexts is quite high.

The relationship between the amount of annotated data and the accuracy for each sound event is illustrated in Figure 7. Differing from other experiments, the accuracy is calculated for each single sound event and therefore represents the single label accuracy. There is a clear correlation between the amount of data and the accuracy for each event. This brings the fact that DNNs require large training databases to learn the mappings between the features and the sound events. This also shows that there is still room for improvement in accuracy once the audio database is expanded. On a related note, we also investigated using shorter frame lengths and/or higher overlap in order to create more instances for the DNN learning. However, this increased the detrimental effect of the erroneous annotations without providing a significant boost on the accuracy. Nevertheless, the effect of the frame length is not investigated exhaustively in this work and therefore out of the scope of this paper.

The overall accuracy of the model trained with different features are presented in Table II. Mel-band energies and log Mel-band energies are calculated in 40 Mel-bands and the number of static MFCC coefficients are chosen to be 16, a standard value in event detection methods. The topology of DNN is kept fixed (except the number of inputs) while using different features in order to make a valid comparison. There is a slight increase in accuracy for Mel-band and log Mel-band energies over MFCCs. This can be explained with the loss of information caused by selecting the first few coefficients after the Discrete Cosine Transform (DCT) [13]. Another point would be that the sum of the MFCCs of different sound sources are not equal to the MFCCs of the mixture of these sources.

The detection predictions $\hat{\mathbf{y}}_t$ are binarized with various thresholds and the accuracy for each polyphony level (*i.e.*, the number of simultaneously active sound events) is given in Table III. For the majority of the levels, the accuracy takes its highest value around the threshold 0.5, which suits with the default guessing of a threshold for a prediction between 0 and 1. The accuracy is higher for high threshold values in the low polyphony levels. This can be explained by the fact that the prominent sound events have very high prediction value, *i.e.*, probability in lower polyphony levels and using high threshold effectively clears the non-present sound events with lower probability. On the other hand, the accuracy is higher for low threshold values in the high polyphony levels. Since the activation function for the output layer of the DNN is *logistic sigmoid*, the detection probabilities are bounded between 0 and 1. However, the sum of the predictions for each sound event is not bounded at all and this sum increases when the polyphony level is increased. When two events with similar spectra are simultaneously active, they share a lower probability compared to the case that only one of them is active. The detection probabilities are distributed over a higher number of sound events in high polyphony levels. Therefore, a lower threshold is required in order to detect multiple sound events.

TABLE III: F1 scores for various binarizing thresholds and polyphony levels for the proposed system after post-processing.

| | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** |
|---|---|---|---|---|---|---|---|---|
| **1** | 44.6 | 51.8 | 57.6 | 62.5 | 66.0 | 66.9 | 64.5 | 56.3 |
| **2** | 51.8 | 56.8 | 59.7 | 61.1 | 61.1 | 58.9 | 54.2 | 44.4 |
| **3** | 55.7 | 59.3 | 60.9 | 61.3 | 59.8 | 56.6 | 50.4 | 39.1 |
| **4** | 60.3 | 63.3 | 64.5 | 64.5 | 63.2 | 60.1 | 54.6 | 42.8 |
| **5** | 64.5 | 65.5 | 65.2 | 64.5 | 62.5 | 59.3 | 53.3 | 40.7 |

The detection accuracy as a function of the polyphony level is given in Figure 9. Binarizing threshold value is fixed at 0.5 for all polyphony levels. The effect of median filtering-based post-processing is clearly visible, especially for lower polyphony levels. Post-processing offers a great boost on the accuracy for lower polyphony levels, *i.e.*, when less events are simultaneously active. As explained in Section IV, post-processing compensates the DNN's tendency to map the frames with low activity, which are found in low polyphony levels, to the non-present sound events. These frames hardly ever appear in very high polyphony levels, hence the effectiveness of the post-processing diminishes. The mapping of low activity frames with non-present events also explains the decreased accuracy in lower polyphony levels before the post-processing.

## VI. Conclusions

In this paper, using multi label DNNs for polyphonic sound event detection in realistic environments was proposed. Multi label DNN classification with median filtering-based post-processing was observed to be able to detect overlapping sound events with high accuracy. The proposed method outperforms the baseline method by 19%. Spectral domain features from short time frames of audio material

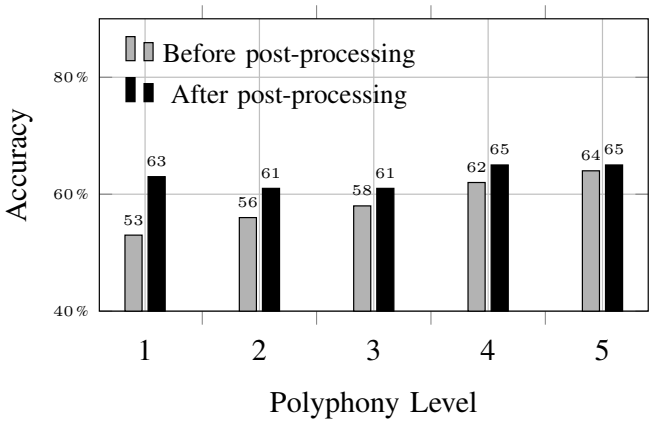


Fig. 9: Detection accuracy vs. polyphony level for the proposed system before and after the post-processing.

were extracted and used as input for the DNN. A post-processing method is proposed which increases the detection accuracy, especially when the number of simultaneously active events in a frame is lower than 4. It is also observed that using a higher binarizing threshold for low polyphony levels provide a better detection accuracy and *vice versa*. For future work, implementing better post-processing methods to handle the noise in the DNN output is planned. Training method extensions such as momentum and weight decay can also be implemented. Investigating more informative features for higher accuracy and robustness is also possible. Another future work direction would be to do the multi label DNN classification for each context separately, which requires a significant amount of data for each context and the reason why we choose the context independent approach in the first place.

## REFERENCES


[1] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 8677–8681.

[2] D. Zhang and D. Ellis, "Detecting sound events in basketball video archive," *Dept. Electronic Eng., Columbia Univ., New York*, 2001.

[3] S. Chu, S. Narayanan, C. Kuo, and M.J. Mataric, "Where am I? scene recognition for mobile robots using audio features," in *IEEE Int. Conf. Multimedia and Expo (ICME)*. IEEE, 2006, pp. 885–888.

[4] A. Harma, M.F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *IEEE Int. Conf. Multimedia and Expo (ICME)*. IEEE, 2005, pp. 4–pp.

[5] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2010, pp. 1267–1271.

[6] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1, 2013.

[7] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2013, pp. 1–4.

[8] J. Dennis, H.D. Tran, and E.S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[9] M. Zhang and Z. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[10] M.R. Boutell, J. Luo, X. Shen, and C.M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[11] A. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Workshop on Text Learning*, 1999, pp. 1–7.

[12] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2014.

[13] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T.N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] Pawel Swietojanski, Jinyu Li, and Jui-Ting Huang, "Investigation of maxout networks for speech recognition," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 7649–7653.

[15] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 1562–1566.

[16] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[17] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2013, pp. 1319–1327.

[18] G.E. Dahl, T.N. Sainath, and G.E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8609–8613.

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