



# Feature extraction strategies in deep learning based acoustic event detection

Miquel Espi, Masakiyo Fujimoto, Keisuke Kinoshita, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT Corporation, Japan

{espi.miquel, fujimoto.masakiyo, kinoshita.k, nakatani.tomohiro}@lab.ntt.co.jp

## Abstract

Non-speech acoustic events are significantly different between them, and usually require access to detail rich features. That is why directly modeling a real spectrogram can provide a significant advantage, instead of using predefined features that usually compress and downsample detail as typically done in speech recognition. This paper focuses on the importance of feature extraction for deep learning based acoustic event detection, and more specifically on exploiting local spectro-temporal features of sounds. We do this in two ways: (1) outside the model, using multiple resolution spectrogram simultaneously based on the fact that there is a time-frequency detail trade-off that depends on the resolution with which a spectrogram is computed (e.g. ‘steps’ would require a finer time resolution, while sounds that span many frequencies require finer frequency detail); and (2), with a model that implicitly exploits locality, convolutional neural networks, which are a state-of-the-art 2D feature extraction model. An experimental evaluation shows that the presented approaches outperform state-of-the-art deep learning baseline with a noticeable gain in the CNN case, and provides insights regarding CNN-based spectrogram characterization.

**Index Terms:** acoustic event detection, spectro-temporal locality, multi-resolution, convolutional neural networks

## 1. Introduction

In conversation scene understanding, research typically tends to concentrate on automatic speech recognition (ASR) as it is considered the most informative component of the acoustic scene. That being so, non-speech acoustic signals provide cues that make us aware of the environment, and we use that information to achieve a complete understanding of each and every situation we face. Typically, we actively or passively neglect mentioning certain concepts that can be inferred from our location, the actions we are performing, or things that are happening around us. For instance, if we are watching a sports game, most of our spontaneous speech utterances are very likely to be related to sports, and ASR could benefit from having such topic knowledge in advance [1]. By hearing a door opening we usually assume that somebody has left or entered the room. Having access to such information in an automated manner can enhance the performance of ASR, diarization, or source separation technologies [2]. Acoustic event detection (AED) deals with detecting and classifying non-speech acoustic signals and the goal is to convert a continuous acoustic signal into a sequence of event labels with associated start and end times. Applications range from rich transcription in speech communication [3, 4] and scene understanding [5, 6], to providing a source of information for speech enhancement and ASR.

We have already seen the potential of directly modeling a real spectrogram to achieve robust AED [7, 8], thus providing the classifier with high-resolution observations. This con-

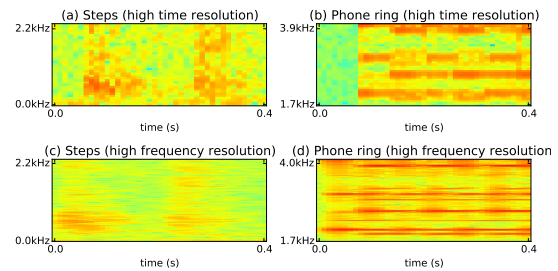


Figure 1: Magnified log power spectrogram regions for ‘steps’ and ‘phone ring’ for high-time resolution (10 ms frame length), and high frequency resolution (90 ms).

trasts with traditional approaches in which the classifier receives predefined acoustic features (e.g. MFCC, or Mel-filter banks) [9, 10]. These usually compress and neglect details that we might need. In this way, as in many sparse-analysis based blind source separation applications [11, 12], the real spectrogram is much sparser, and provides greater detail in time-frequency space. In this way, our work in [8] successfully applies deep learning to that acoustic event detection based on spectrogram patches. Here, a spectrogram patch is considered as a group of consecutive spectrum frames which contain enough detail to model complex temporal and spectral structures.

In [8], spectrogram patches are modeled using deep neural networks (DNN) [13], which are feed-forward networks with multiple fully-connected hidden layers that have been pre-trained using restricted Boltzmann machines (RBM) [14]. The input layer is pre-trained as a Gaussian-binary RBM (i.e. Gaussian visible units and binary hidden units), and the following hidden layers are pre-trained as a binary-binary RBM. Finally, the model is fine-tuned using the back-propagation algorithm to estimate HMM state posteriors as is typically in many speech recognition studies. Having a fully-connected input layer means that each of the hidden nodes in this first layer learn representations of entire spectrogram patches. We call this “global” characterization of the spectrogram in this paper. While this shows to perform well, information that shows up when looking closely to the spectrogram (e.g. how stationary or transient a sound is) is dismissed. We refer to this as “local” characterization of the spectrogram, and explore two feature extraction strategies within the deep learning paradigm that consider it. First, we look at an approach that considers “local” characteristics outside the model, and that uses multiple resolution spectrograms simultaneously to obtain several output labels streams that are then merged. And second, we consider an approach that implicitly exploits “locality” in the model, convolutional neural networks (CNN) [15], which is a state-of-the-art 2D feature extraction model in deep learning.

## 2. Exploiting spectro-temporal locality

As we mentioned above, spectrogram-input DNNs learn features that “globally” describe the spectrogram patches. This

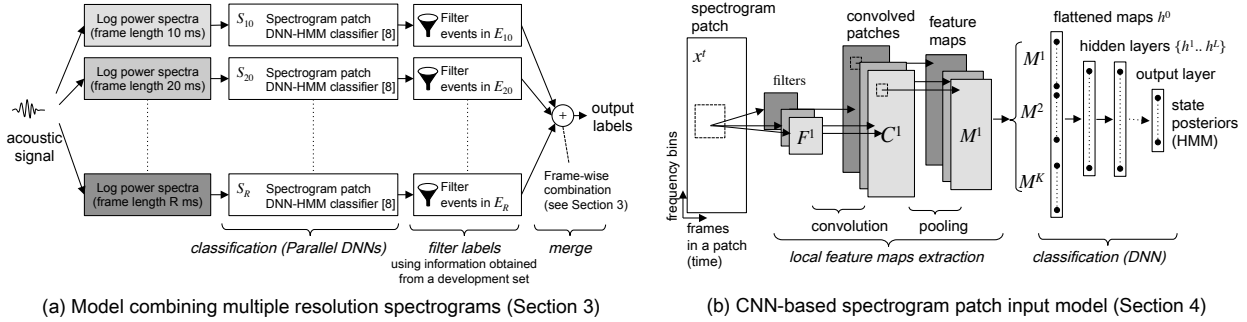


Figure 2: Two deep learning based approaches to AED that exploit spectro-temporal locality: explicitly by combining multiple recognition streams from different spectro-temporal resolutions (a), and implicitly using CNNs which implicitly exploit locality in the input.

makes sense since spectral shapes contained in acoustic events are less variable than speech, for instance. The hidden nodes in the first layer are fully connected to the input, and therefore they model entire spectrogram patch shapes. Whenever these shapes appear in an input patch certain hidden nodes are or are not activated, and this is propagated through the neural network until it reaches the output layer. This completely ignores local spectro-temporal properties of sounds such as characterizing sounds as being rather stationary, transient, and so on. While these local properties are not necessarily meaningful in all cases, we can assume they contribute to a better recognition of acoustic events. We approach our goal of exploiting spectro-temporal locality in two ways: outside the model, by providing the model with augmented input in the form of multiple spectrograms for different resolutions of the same sound samples; and inside the model, with a model that exploits spectro-temporal locality implicitly.

Given the differences between acoustic events in terms of time and frequency resolution, we can assume that spectrogram-input AED systems are dependent on the resolution with which the spectrogram was computed. Fig. 1 shows a magnified spectrogram region of two acoustic events, ‘steps’ and a ‘phone ringing’, with high time resolution (top), and high frequency resolution (bottom). Observing the high time resolution spectrogram (Fig. 1 a and b) one is capable of recognizing without major efforts onsets, transient sounds, and low energy signals. This is not the case with high frequency resolution (Fig. 1 b and c) but we have more detailed access in the frequency axis. This trade-off between time and frequency resolution arises from the fact that the frame length influences the shape of time-frequency bins in a spectrogram, and this shape influences the amount of detail on each axis at the local scale. Section 3 describes a simple approach that looks at different spectrogram resolutions of the same input in parallel to obtain a combined output.

Section 4 describes how we incorporate the use of CNNs, which we have already seen used in acoustic signal processing applications [16, 17, 18], in addition to computer vision. CNNs provide the means to extract local features from the spectrogram itself. What makes CNNs a perfect candidate is the way in which the convolution of relatively small-sized filters over a spectrogram patch is able to learn the local feature maps (convolution is only done with adjacent bins in time and frequency, i.e. local). CNNs are closely related to the concept of feature extraction, and model not just the input as a whole, but also independent local features in an integrative manner. The entire model is then globally built by jointly training the convolutional and DNN architectures as a whole using back-propagation (see Fig. 2.b). CNNs provide an excellent way of learning convolution filters that extract salient local filters from 2D inputs. While, in images, these account for figure corners, edges, and so

on, such filters are also meaningful when the inputs are spectrogram patches. Finding local features that highlight continuity in time, continuity in frequency, or other more sophisticated local patterns, allows the model to unfold a single spectrogram into many local feature maps and classify them.

### 3. Combining multiple spectral resolution

A set of single resolution acoustic event classifiers whose parallel output is then combined using rules that have been previously learned using a development dataset. Single resolution classifiers  $\{S_{10}, \dots, S_r, \dots, S_R\}$  are trained separately for the same AED task and the same output labels each for a different spectral resolution  $r$  (e.g.  $r = 10$  ms frame length) [8]. The combination scheme is learned using a development dataset, obtaining label sets  $E_r$  for each of the single-resolution classifiers  $S_r$ , which contain the set of event labels that perform best for each resolution  $r$ . For instance, if the event “door knock” returned its best frame-score using  $S_{40}$  (40 ms frame length spectrogram input classifier) then the labels “door knock” will be contained in  $E_{40}$ . Recognition proceeds as follows:

1. Compute log power spectrograms for each resolution.
2. Single resolution DNN classifiers  $S_r$  receive corresponding input and then output a sequence of labels.
3. These output sequences of labels are then filtered using their corresponding optimal sets of labels  $E_r$ . E.g. if the output sequence for  $S_{10}$  contains “steps” and “phone”, but “phone” is not in  $E_{10}$ , then the output after filtering will replace “phone” with “silence” labels.
4. Finally, for each frame, labels are merged resulting in frames that can contain multiple events as in real life. E.g. After filtering, if for a given frame we have multiple “silence” labels, and a “door knock”, the output will be “door knock”; but if we have multiple “silence” labels, a “door knock” and an “applause”, the final output will contain the last two.

Fig. 2.a is a diagram describing this architecture. This is a rather simple approach, but we could draw conclusions from it.

### 4. Spectrogram patch input CNN model

CNNs exploit time-frequency local correlation by enforcing local connectivity patterns between neurons of adjacent layers. The input hidden units to the DNN part of the model ( $C^k$ , and  $M^k$  after pooling) are connected to a locally limited subset of units in the input spectrogram patch, which are contiguous in time and frequency. Each sparse filter  $F^k$  is additionally replicated across the entire input patch forming a feature map, which shares the same parametrization (i.e. weights and bias).

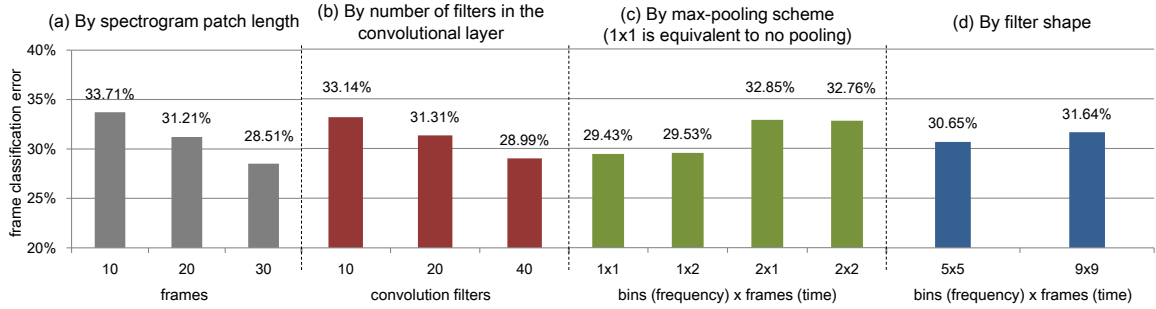


Figure 3: Averaged frame-score errors by setup parameter as described in subsection 5.1: spectrogram patch length, number of convolutional filters to be trained, max-pooling scheme, and filter shape.

Convolved patches  $C^k$  are obtained by convolving a given spectrogram patch input  $x^t$  with a linear filter  $F^k$  of shape  $S \times S$ , adding a bias term  $b^k$ , and applying a non-linear activation,

$$C_{ij}^k = \tanh \left( \sum_{m=1}^S \sum_{n=1}^S \left( F_{mn}^k x_{(i+m-\lfloor \frac{S}{2} \rfloor), (j+n-\lfloor \frac{S}{2} \rfloor)}^t \right) + b^k \right) \quad (1)$$

Then we apply max-pooling according to a pooling scheme of shape  $P_1 \times P_2$  that we chose to obtain the feature map  $M^k$ ,

$$M_{ij}^k = \max \left( C_{(iP_1:(i+1)P_1), (jP_2:(j+1)P_2)}^k \right) \quad (2)$$

where  $P_1$  and  $P_2$  refer to pooling along frequency and time, respectively (e.g. a 1x1 pooling scheme is equivalent to no pooling). The pooling stage has no parameters, and therefore there is no learning either. Replicating the convolution units this way allows features to be detected regardless of their position in time or frequency. This directly relates to the fact that we are not learning event-dependent features, but rather useful local filters that reveal more independent aspects of sounds.

The rest of the CNN architecture consists of fully connected layers of hidden nodes (i.e. a vector of activation nodes  $h^l$ ) with sigmoid activations as in regular deep neural networks. For the first hidden layer  $h^1$ , the input layer  $h^0$  will be a flattened concatenation of all the feature maps  $\{M^1 \dots M^K\}$ .

## 5. Experiments and results

AED experiments were performed as part of the acoustic event recognition task in CHIL2007 [3], a database of seminar recordings in which 12 non-speech event classes appear besides speech: applause (ap), spoon/cup jingle (cl), chair moving (cm), cough (co), door slam (ds), key jingle (kj), door knock (kn), keyboard typing (kt), laugh (la), phone ring (pr), paper wrapping (pw), and steps (st). For CNN, a log power spectrogram was computed using 10 ms frames with a 10 ms shift, while for multi-resolution DNN the frame lengths ranged between 10 ms and 60 ms with a 10 ms shift. All models and experiments were implemented using the Theano library [19].

Both DNN and CNN models are trained to estimate HMM posterior states, similarly to how it is done in ASR. The HMM topology consists of one state per acoustic event, and an ergodic architecture in which all states have a self transition and then equal transitions to all other states. This results in a model similar to a GMM-HMM but where the GMM is replaced with a neural network. Then the final sequence of events is obtained by applying Viterbi decoding (see [8] for further details).

### 5.1. Setup parameters

We considered two parameters for the approach combining multiple spectrogram resolutions:

- Frame length (resolution): 10, 20, 30, 40, 50, and 60 ms, with 129, 257, 257, 513, 513, 513 bins respectively.

- Spectrogram patch length: 10, 20, and 30 frames.

CNNs add more hyper-parameters to the typical DNN model, and therefore, we have designed a broad set of experiments to learn how these parameters affect the performance with the following settings:

- Spectrogram patch length: 10, 20, and 30 frames.
- Filter shape ( $S \times S$ ): 5x5 and 9x9 filter (bins x frames).
- Number of filters ( $K$ ): 10, 20, and 40 filters.
- Pooling ( $P_1 \times P_2$ ): 1x1 (no pooling), 2x1 (frequency pooling), 1x2 (time pooling), and 2x2 (both axes).

CNN models have one convolutional layer and two hidden fully connected layers with 512 nodes each, while the DNN-only models have a first hidden layer with 1024 nodes to deal with the input and two hidden layers with 512 hidden nodes each. Note that for the multi-resolution approach, a single frame can contain multiple labels. In this case, for any given frame, if the labels contained in the recognizer output completely match those in the ground truth, the frame will be considered correct; any other case will cause the frame to be considered incorrect.

### 5.2. Multi-resolution DNN Results

Looking at event-wise results with the best performing spectrogram patch length (20 frames) as shown in Table 1, the first conclusion is that the best performing resolution overall is not the best resolution for each and every acoustic event class separately. In general, and consistent with previous assumptions, certain low energy events such as ‘keyboard typing’ are better tracked with short frame resolutions, whereas long frames perform better for a ‘door slam’ (50 ms). This also occurs with ‘applause’ (40 ms), with a very similar structure in the frequency domain. On the other hand, with events such as ‘chair move,’ switching the frame length seems to have almost no effect on performance. The performance of other sounds such as ‘laugh’ varies with no strong trend.

Table 2 shows two metrics: frame classification error, as in Fig. 3; and AED-accuracy which is the event-wise f-measure between precision and recall. Here, the overall results here confirmed that even a simple combination approach provided a significant error reduction over the best performing single resolution. Thus, pointing out the relevance of spectral resolution.

### 5.3. CNN model results

The CNN results are shown in Fig. 3, and further summarized for ease of comparison in Table 2. At first sight, the first conclusion is that we are able to obtain better performance than with any DNN-only model (Table 2). The best performance was obtained with the the longest spectrogram patch configuration (Fig. 3.a). In terms of filter shapes, filters covering smaller regions provide better performance on average (Fig. 3.d) but the actual best score came from a wide filter size (Table 2). As for

Table 1: Resolution-event results (frame-score error %) for the best performing DNN spectrogram patch size (20 fr./patch).

AE	Frame length (resolution)					
	10ms	20ms	30ms	40ms	50ms	60ms
ap	23.60%	34.40%	33.77%	<b>17.33%</b>	30.09%	27.14%
cl	28.15%	<b>15.95%</b>	31.82%	37.73%	37.59%	29.35%
cm	62.40%	64.28%	66.24%	69.01%	<b>55.99%</b>	76.37%
co	<b>63.02%</b>	72.17%	72.50%	82.90%	78.41%	70.02%
ds	70.29%	83.08%	81.24%	88.37%	<b>61.25%</b>	78.33%
kj	87.10%	88.53%	85.36%	87.29%	<b>82.88%</b>	86.33%
kn	50.33%	72.91%	62.96%	<b>33.10%</b>	55.42%	76.44%
kt	<b>61.62%</b>	72.02%	73.01%	72.70%	67.38%	71.48%
la	<b>86.32%</b>	87.85%	87.51%	89.21%	89.09%	88.51%
pr	46.41%	44.01%	48.64%	<b>39.74%</b>	44.56%	47.17%
pw	17.65%	17.71%	12.71%	<b>7.97%</b>	17.40%	11.84%
st	45.14%	52.84%	48.16%	53.56%	<b>36.72%</b>	51.61%
all	<b>30.19%</b>	32.82%	32.65%	31.90%	31.66%	32.65%

Table 2: AED evaluation results with the ‘test’ set.

System	Frame-error	AED-acc
Best single resolution DNN model [8] 20 frames/patch, 10ms frames	30.19%	54.82%
Multi-resolution DNN models		
10 frames patch	28.19%	56.95%
20 frames patch	<b>27.45%</b>	<b>57.03%</b>
30 frames patch	29.84%	54.01%
Best performing CNN models		
No pooling, 9x9 filter (40), 30 frames patch	<b>23.58%</b>	<b>61.38%</b>
1x2 pooling, 9x9 filter (20), 30 frames patch	24.79%	60.85%
1x2 pooling, 5x5 filter (20), 30 frames patch	24.89%	60.85%

pooling, the general conclusion is that the effects are different for pooling along frequency and pooling along time (Fig. 3.c), but none truly contributes to improve performance.

## 6. Discussion

The main clear result was that CNNs outperformed multi-resolution DNNs. This is a largely expected result since CNNs feature a richer ensemble of feature extraction configurations, and more opportunities for learning among the hidden layers. CNNs assume there is an order in the input, as opposed to the fully connected layers in DNN, in which the location of the data in the input is irrelevant. This makes a difference as regards image recognition and, as we have learned here, it also makes a difference in spectrogram-based AED. The next major observation is that a combination of multiple feature extraction procedures provides better results than the single best feature extraction procedure. This, of course, is relatively unsurprising, as a combination of classifiers (or feature extractors) is always best if the information provided by each one is complementary.

Conceptually, both approaches operate in different directions. Multi-resolution analysis assumes that salient local features arise from observing the spectrum at different time-frequency resolutions. The CNN model exploits smaller salient structures that appear in the spectrogram, and describes more general properties (e.g. continuity in time or frequency, fluctuating patterns). The two analyses are not exclusive, but rather complementary, and should be investigated in the future.

### 6.1. CNN parameters

Although it is hard to draw strong conclusions with respect to pooling, the experiments reveal that there is no gain in pooling. Pooling along frequency degrades the performance significantly, while pooling in time does not seem to have either positive or negative effects. Experiments with larger datasets might reveal a trend in this matter in the future, but we have not yet seen a gain. We argue that while max pooling makes sense when downsampling images, this is not so straight forward for downsampling of acoustic spectrograms. In the future

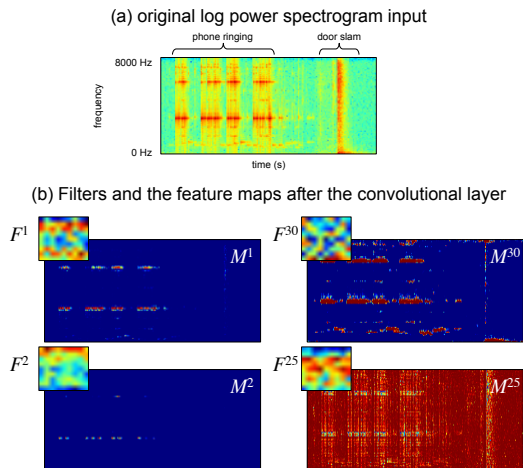


Figure 4: Selected filters and convolutional layer output of a CNN with 40 9x9-filters 30 frames input.

more appropriate downsampling strategies similar to filter-bank or MFCC-like functions might provide better results, and this is a direction worth exploring since CNN-based image recognition benefits from the inclusion a pooling stage.

As for the CNN filters, looking at the feature maps after convolution and pooling (Fig. 4) reveals some interesting insights about what these filters are learning. For instance, map  $M_{40}$  shows how  $F_{40}$  focuses on short time components, and  $M_1$  and  $M_{25}$  focus on more stationary and harmonic components. Interestingly enough,  $M_2$  shows that  $F_2$  has learned a filter that highlights components that fluctuate in time and frequency.

Comparing the performance obtained using different numbers of filters (Fig. 3.b) we can intuitively reach the conclusion that the more parameters we have, the more local features we can learn, therefore more filters (40) usually means better performance. That being said, we can already achieve fairly good performances with 10 filters without pooling. A careful observation of the filters after training reveals that as we increase the number of parameters (number of filters) some of these filters seem to receive less training and remain largely random. This might also be in part due to the small amount of data.

### 6.2. Other aspects

It should be noted that the CNN models in this work have not been pre-trained in contrast to the unsupervised pre-training step in DNN-only models, resulting in lower computational costs. Furthermore, the addition of many more parameters to train in the convolutional layer usually means additional computational training cost, however, local convolution weights and biases are shared for each of the filters, and this considerably reduces the number of parameters that we must train.

## 7. Conclusions

This paper considered the importance of exploiting spectro-temporal locality in feature extraction for acoustic event detection using deep learning models. We accomplished this by exploring two paths: multi-resolution spectrogram patch input DNN models, and CNN feature extraction of time-frequency local maps. Both approaches outperform the single-resolution baseline, with a noticeable gain in the CNN case. However, we consider that both approaches are complementary since they focus on different aspects of local characterization. In the future we intend to explore a combination of these two approaches, attempt to understand how CNN parameters interact with the performance, and consider layer-wise pre-training for the CNN.

## 8. References

- [1] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 499–513, 2012.
- [2] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard, "Informed source separation: source coding meets source separation," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. 257–260.
- [3] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. Chu, A. Tyagi, J. Casas, J. Turmo, L. Cristoforetti, F. Tobia, A. Pnevmatikakis, V. Mylonakis, F. Talantzis, S. Burger, R. Stiefelhagen, K. Bernardin, and C. Rochet, "The CHIL audiovisual corpus for lecture and meeting analysis inside smart rooms," *Language Resources and Evaluation*, vol. 41, no. 3-4, pp. 389–407, 2007.
- [4] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, 2013, pp. 1–4.
- [5] K. Imoto, S. Shimauchi, H. Uematsu, and H. Ohmuro, "User activity estimation method based on probabilistic generative model of acoustic event sequence with user activity and its subordinate categories," in *INTERSPEECH'2013*, 2013, pp. 2609–2613.
- [6] C. Canton-Ferrer, T. Butko, C. Segura, X. Giro, C. Nadeu, J. Hernandez, and J. Casas, "Audiovisual event detection towards scene understanding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2009, pp. 81–88.
- [7] X. Lu, Y. Tsao, and S. Matsuda, "Sparse representation based on a bag of spectral exemplars for acoustic event detection," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2014, pp. 6399–6403.
- [8] M. Espi, Fujimoto, Y. Kubo, M., and T. Nakatani, "Spectrogram patch based acoustic event detection and classification in overlapping speech scenarios," in *HSCMA*, 2014, pp. 117–121.
- [9] X. Zhuang, X. Zhou, M. Hasegawa-Johnson, and T. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.
- [10] M. Espi, M. Fujimoto, D. Saito, N. Ono, and S. Sagayama, "A tandem connectionist model using combination of multi-scale spectro-temporal features for acoustic event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4293–4296.
- [11] S. Araki, T. Nakatani, and H. Sawada, "Simultaneous clustering of mixing and spectral model parameters for blind sparse source separation," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 5–8.
- [12] T. Nakatani and S. Araki, "Single channel source separation based on sparse source observation model with harmonic constraint," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 13–16.
- [13] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [14] G. Hinton, "A practical guide to training restricted Boltzmann machines," in *Technical report 2010-003*. Machine Learning Group – University of Toronto, 2010, pp. 1–10.
- [15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [16] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *2013 12th International Conference on Document Analysis and Recognition*, vol. 2. IEEE Computer Society, 2003, pp. 958–958.
- [17] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 2519–2523.
- [18] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *EUSIPCO*, 2014.
- [19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.