# Filterbank Learning for Deep Neural Network Based Polyphonic Sound Event Detection

Emre Cakir, Ezgi Can Ozan, Tuomas Virtanen

***Abstract*— Deep learning techniques such as deep feedforward neural networks and deep convolutional neural networks have recently been shown to improve the performance in sound event detection compared to traditional methods such as Gaussian mixture models. One of the key factors of this improvement is the capability of deep architectures to automatically learn higher levels of acoustic features in each layer. In this work, we aim to combine the feature learning capabilities of deep architectures with the empirical knowledge of human perception. We use the first layer of a deep neural network to learn a mapping from a high-resolution magnitude spectrum to smaller amount of frequency bands, which effectively learns a filterbank for the sound event detection task. We initialize the first hidden layer weights to match with the perceptually motivated mel filterbank magnitude response. We also integrate this initialization scheme with context windowing by using an appropriately constrained deep convolutional neural network. The proposed method does not only result with better detection accuracy, but also provides insight on the frequencies deemed essential for better discrimination of given sound events.**

## I. Introduction

A sound event is an audio segment that can be labeled as a distinctive concept in an audio signal. Some examples of sound events can be given as dog bark, car horn, footsteps etc. Automatic detection of sound events has recently drawn a surging interest especially in accordance with the invention of modern machine learning techniques. Sound event detection has applications in smart homes [1], scene recognition for mobile robots [2] and surveillance in living environments [3], [4].

An audio analysis system can be divided into two stages as *sound representation* and *classification* of consecutive time frames. In order to extract the relevant information, the audio signals are often transformed into a higher level representation. This representation is obtained by the *acoustic features*. In audio related classification and detection tasks such as speech recognition, music classification, acoustic scene classification and sound event detection, there are certain conventional, hand-crafted acoustic features that have proved to perform in a satisfactory level. These features are commonly extracted from the time-frequency representations of the audio signals, *i.e.*, spectrograms. Some of the conventional acoustic features can be listed as spectral energy, zero-crossing rate [5], (log) mel band energy [6] and mel-frequency cepstral coefficients (MFCC) [7]. The spectrogram of an audio signal can also be regarded as an image, which makes it possible to apply and extend hand-crafted image feature extraction methods on sound event detection. Consequently, local spectrogram features [8], histogram of gradients [9], [10] and Gabor filterbank features [4] have been proposed recently for sound event detection.

The evidence from auditory psychophysics proves that humans perceive sound signals along a nonlinear scale in frequency domain [11]. Human ear is more sensitive to the changes in the lower frequencies than in the higher frequencies. Consequently, acoustic features are often chosen by using nonlinear scales (log frequency scale, mel scale, bark scale etc.) to give better correspondence with human perception. Mel scale is a perceptual scale in which the pitches are adjusted by the listeners so that the successive pitches are perceived to have equal distance among the scale [12]. In sound related classification tasks, acoustic features that utilize the mel scale have been found to provide robust sound representation and have become standard features. Mel band energy and especially MFCCs can be given as examples for such features [6], [7]. On the other hand, the empirical nature of mel scale leads to several hand-crafted mel formulas [12], [13], [14], [15], [16], all of which provide only an approximation to the real human perception. There are also certain views that mel scale needs to be readdressed due to biased experiments, namely the original experiments did not take into account the hysteresis, *i.e.*, human perception varying when the frequencies are listened in the ascending order than in descending order [17]. To sum up, mel scale does not provide a perfect fit for human perception. On the other hand, the goal in polyphonic sound event detection is not modeling perfectly the human perception, but to obtain high detection accuracy. Nevertheless these two goals are closely related to each other, and mel scale representation can be a good starting point for acoustic features in a sound event detection task.

For the classification stage, the traditional approaches in sound event detection and other audio related classification tasks have been Gaussian mixture modeling (GMM) [4], Hidden Markov modeling (HMM) [18] and Support Vector Machines (SVM) [19]. Recently, deep neural networks (DNN), which are neural networks with multiple hidden layers, have been shown to give better results in sound event detection [6], [20], [21], [22], [23].

Deep learning architectures have opened an alternative path to hand-crafted data representation methods. In a deep architecture, with each hidden layer, the input is transformed to a higher level representation which is to be classified at the

*The research leading to these results has received funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND. Computing resources of *Taito* supercluster from Finnish Grid Infrastructure has been used in this work.

output layer. The layer parameters such as weights and biases are generally initialized with small random values. While backpropagating the error derivatives during the training stage, these weights and biases are updated and higher level features are obtained to present a better model for the target output, *i.e.*, features are *learned* through the input data. Due to their high expressional capability, deep classifiers do not rely on the high level hand-crafted representations of the *raw* data as their input and they are able to express the highly nonlinear relationship between the raw data and the target output. This led to a trend of using raw data (pixel values for an image [24] or magnitude/power spectrum for audio [25], [26], [27] or even raw audio [28]) as input and deep learning methods as the classifier in machine learning tasks.

In our recent work [6], we found that for the given polyphonic environmental sound recordings, the optimal multi-label detection method was using mel band energies as acoustic features and DNN with two hidden layers. DNNs with mel band energy features outperformed the traditional MFCC feature with GMM-HMM classifier method by a huge margin. The ability of DNNs to use subsets of their hidden units to detect several events simultaneously makes them a suitable choice for polyphonic sound event detection tasks, where multiple events are occuring simultaneously [6], [25]. DNNs also tend to perform better with lower level acoustic features (such as mel band energies) compared to traditional higher level features (such as MFCCs) [29]. This can be explained with the fact that MFCCs are obtained by applying discrete cosine transform (DCT) on mel band energy features to decorrelate these features, but DNNs are already very powerful in modeling data correlation and do not necessarily require a DCT step [30].

In this work, we propose to integrate the empirically obtained human perception information with feature learning capabilities of the deep architectures to learn a new filterbank. This ad-hoc filterbank is initialized with a human perception based method and tuned with deep learning techniques. We use the high-resolution magnitude spectrum of an audio signal (which can be regarded as low level representation) as the input for deep feedforward and deep convolutional neural networks (CNN). Instead of initializing all the weights and biases with small random values, we initialize the first hidden layer weights of the network as the coefficients of a human-perception based filterbank magnitude response, namely mel filterbank. During training, the first hidden layer weights, *i.e.*, the filterbank magnitude response is updated to provide better discrimination over the target sound events. The trained network does not only provide better detection accuracy over completely randomly initialized networks, but also, depending on the updates made on the first hidden layer weights during training, it gives an idea of which frequency bins are deemed more essential for discrimination in a given sound event detection task.

The organization of this paper is as follows. Polyphonic sound event detection task is explained in detail in Section II. The proposed mel scale filterbank and mel scale weight initialization technique for deep neural networks is proposed in Section III. Acoustic data and network settings used in the work is explained and an evaluation of the proposed methods is presented in Section IV. Finally, our conclusions on the subject is presented in Section V.

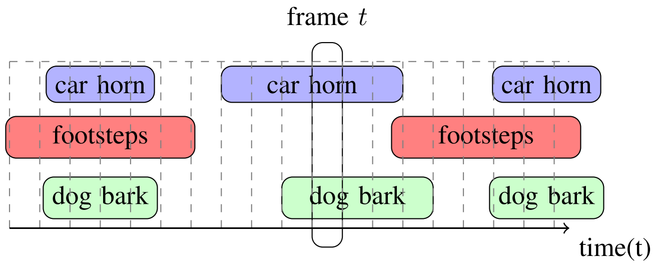


Fig. 1: Overlapping sound events in a recording from a realistic environment. Frame $t$ represents the short time frame from the part of the audio signal where only car horn and dog bark events are present.

## II. Sound Event Detection

The aim of sound event detection is to temporally locate the sound events and assign a class label to each event present in an audio signal. In a real-life situation, there can be multiple sound events present simultaneously, *i.e.*, the audio signal can be polyphonic at a certain time. An example of this is illustrated in Figure 1, where an audio signal is labeled with the sound events happening in time domain. Sound event detection can be formulated as a multi-label classification problem in which the temporal locations of sound events are obtained by doing binary classification of the activity of each sound event class over consecutive time frames.

In the sound representation stage, the audio signal $x(t)$ is first divided into $N$ time frames by using *e.g.* Hamming window of length 40 ms and 50% overlap. Then, the magnitude spectrum matrix $\mathbf{X}$ for the audio signal $x(t)$ is obtained by taking the absolute value of the Fast Fourier transform (FFT) of the short time frames. The traditional approach is to compute further higher level representations from $\mathbf{X}_{m,n}$, where $m$ and $n$ are the frequency bin and frame indices. Instead, we add another hidden layer of the deep architecture and initialize that layer's weights with mel filterbank magnitude response to automatically learn the higher level representations. This technique is explained in detail in Section III.

The task in the classification stage is to model the nonlinear relationship between the magnitude spectrum $\mathbf{X}$ and the target outputs $\mathbf{Y} \in \mathbb{R}^{K \times N}$ as

$$\mathbf{X} \rightarrow \alpha(\cdot) \rightarrow \mathbf{Y} \qquad (1)$$

where $\alpha$ is the classifier model, $\mathbf{Y}$ is the binary matrix encoding the present events in frame $n$ and $K$ is the pre-determined number of events. If the event $k$ is present in a frame $n$, $\mathbf{Y}_{k,n}$ is set to 1 and otherwise 0. In multi-label sound event detection, the pre-determined events can occur

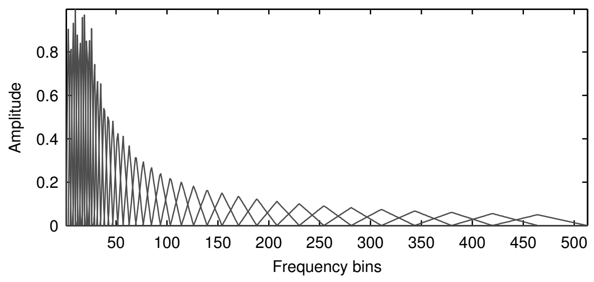


Fig. 2: Mel filterbank magnitude response.

in any different combination at a time. So, a vector of length $K$ is required to encode the target outputs $\mathbf{Y} \in \mathbb{R}^{K \times N}$.

The classifier model $\alpha$ is trained using annotated material. The start and end times of each event in the audio signal $x(t)$ is annotated and used to obtain the binary target outputs $\mathbf{Y}$. The annotation procedure is illustrated in Figure 1 and explained in more detail in Section IV-A.

## III. Method

In this work, we combine the following findings:

- Mel scale approximates the human hearing perceptually better than a linear frequency scale [12],
- New improvements on DNN architectures and learning are needed to push the features even further back to the raw levels of acoustic measurements [30].

We use magnitude spectrum of short time frames as input for a DNN and initialize the first hidden layer weights of a DNN with mel scale filterbank magnitude response. This is done to integrate the empirical human perception knowledge in the automatic feature learning. In order to introduce the temporal information encoded in the consecutive time frames, we use appropriately constrained CNN with context window input. We obtain the binary detection output by regarding the network outputs as posterior probabilities and thresholding these probabilities with a constant $\phi$.

### A. *Mel scale filterbank*

Mel scale is a perceptually motivated scale that is designed so that the intervals consist of equal perceptual pitch increments. The reference point is often chosen as setting 1000 mels to 1 kHz. The relation between mel scale frequency $mel$ and frequency $f$ (Hz) is commonly defined with the following formula [15]:

$$mel = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2}$$

First, the lowest and highest edges of the frequency range $[0, \frac{Fs}{2}]$ are converted into mel frequencies according to (2), where $Fs$ is the sampling rate. Then, the mel range is equally spaced into $B$ mel bands. Mel scale filterbank magnitude response $\mathbf{F}_{b,m}$ is designed according to this mel scale, where $b$ and $m$ are the mel filter and frequency bin indices, respectively. The magnitude response of the filterbank is approximated using triangular band-pass filters. For each mel filter, the triangular band-pass filter magnitude response

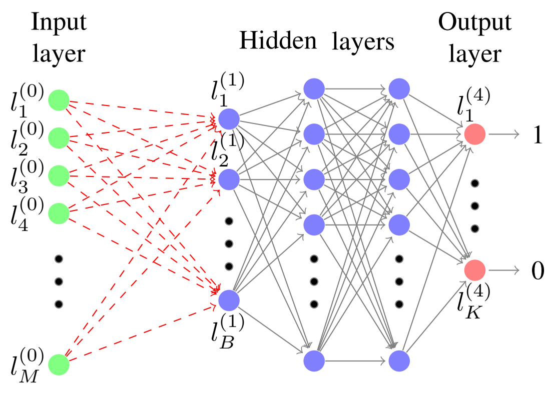


Fig. 3: Symbolic representation of the DNN topology with mel scale weight initialization. The dashed red weight connections between $l^{(0)}$ and $l^{(1)}$ are initialized with mel filterbank magnitude response $\mathbf{F}_{b,m}$ where $b \in [1, ...B]$ and $m \in [1, ...M]$.

$\mathbf{F}_{b,m}$ is selected in the range $[0, 1]$, where the response is 0 outside a band, and from band start rises linearly to peak 1, from which it linearly decays to 0 at band end. The slope is larger at the lower frequencies where the mel bands contain fewer frequency bins, and vice versa. Then, the magnitude response for each filter is scaled so that the magnitude sum for each filter is approximately constant. In this work, the resource code in [31] is used to obtain the mel scale filterbank magnitude response explained above and visualized in Figure 2.

Mel band magnitudes $\mathbf{E}$ are computed from the magnitude spectrum $\mathbf{X}$ and mel scale filterbank magnitude response $\mathbf{F}$ as

$$\mathbf{E}_{b,n} = \sum_{m=1}^{M} \mathbf{F}_{b,m} \mathbf{X}_{m,n} \tag{3}$$

where $M$ is the half of the number of FFT bins in the magnitude spectrum. In this work, 1024 point DFT is used, therefore $M = 1024/2 + 1 = 513$. Negative frequency terms of $\mathbf{X}$ are discarded because the FFT is computed for real input and therefore $\mathbf{X}$ is Hermitian-symmetric, where the negative frequency terms consist of complex conjugate values of the positive frequency terms.

### B. *DNN with mel scale weight initialization*

As the input for the DNN, we use the first half of the magnitude spectrum, which is a lower level representation compared to traditional acoustic features such as MFCCs. As in [6], DNN structure consists of two hidden layers before the output layer. The difference with the architecture in [6] is that we add another hidden layer $l^{(1)}$ of $B$ hidden units at the front part of the DNN architecture, *i.e.*, between the input layer and the first hidden layer. For a given input vector $\mathbf{x}$ (where $\mathbf{x} = \mathbf{X}_{:,n}$, time frame index $n$ is omitted for simplicity), the input layer $l^{(0)}$ is set to $\mathbf{x}$ and the hidden neuron outputs $\mathbf{z}_i^{(1)}$ for the layer $l^{(1)}$ are computed as

$$\mathbf{z}_i^{(1)} = \theta(\sum_{j=1}^{M} \mathbf{W}_{i,j}^{(1)} \mathbf{x}_j) \qquad (i = 1, 2, ...B) \tag{4}$$

where $\mathbf{W}^{(1)} \in \mathbb{R}^{B \times M}$ are the hidden neuron weights and $\theta$ is the activation function. The output layer outputs $\mathbf{z}^{(4)} \in [0, 1]$ are treated as the posterior probability $\tilde{y}$ for each event class. Selecting $\theta$ as rectified linear unit will result in the same structure in computing mel band magnitudes as in Eq. 3 and computing feedforward neural network outputs as in Eq. 4. Instead of initializing the weights $\mathbf{W}^{(1)}$ with small random values, we use mel filterbank magnitude response $\mathbf{F}_{b,m}$, so that this layer will learn a filterbank that provides better discrimination of the given sound events. The process is illustrated in Figure 3. For the output activations, logistic sigmoid function is used to obtain multi-label outputs.

Mel scale filterbank magnitude response $\mathbf{F}_{b,m}$ is a sparse matrix due to the fact that mel bands span only a few frequency bins in the linear scale, especially for the lower frequencies. On the other hand, randomly initialized weights of the rest of the architecture, *i.e.*, weights for layers $l^{(2)}$, $l^{(3)}$ and $l^{(4)}$ are sampled from a uniform distribution $U(-\Delta, \Delta)$, where $\Delta$ is calculated from the randomized initialization method proposed in [32]. The random weights drawn from a uniform distribution $U(-\Delta, \Delta)$ have zero mean by definition and their variance can be calculated as $\sigma^2 = \frac{1}{12}(-\Delta - \Delta)^2$. Using $\mathbf{F}_{b,m}$ directly to initialize the weights $\mathbf{W}^{(1)}$ would result in a mismatch in terms of mean and variance with the rest of the architecture, which may slow down the learning. Therefore, the mean of $\mathbf{F}_{b,m}$ coefficients is subtracted and the variance is matched with the uniformly distributed weights before initialization. In order to avoid some neurons having the same initial weights as zero and introducing an undesired symmetry to initialization after mean and variance matching, the zero weights of $\mathbf{W}^{(1)}$ are replaced with small random values. This initial mel filterbank response is illustrated in Figure 6(a).

### C. DNN with context windowing

Context windowing, *i.e.*, expanding the input instance from the features of a single frame to a neighbourhood of frames, has been shown to improve the performance significantly in sound event detection with a DNN classifier [29]. In context windowing for DNN, the input $\mathbf{X}_{:,n}$ is concatenated with $\frac{C-1}{2}$ preceding and $\frac{C-1}{2}$ succeeding frame features to form context window $\hat{\mathbf{X}}_{:,n}$ as

$$\hat{\mathbf{X}}_{:,n}^{T} = \left[ \mathbf{X}_{:,n-\frac{C-1}{2}}^{T} \cdots \mathbf{X}_{:,n+\frac{C-1}{2}}^{T} \right] \tag{5}$$

where superscript $T$ represents the the transpose.

While using DNNs, context windowing can be done by concatenating input frame features in single dimension. However, this approach can be deemed inefficient, as the information that some of the features actually come from consecutive time frames is discarded while presenting the input to DNN. In addition, when the number of input features is high, the number of parameters to be learned between the input and the first hidden layers increase accordingly. Moreover, combining context windowing with mel scale weight initialization in a DNN architecture would result in separate learned filterbanks for each frame, unless a weight sharing technique is used. Therefore, in this work, context windowing for DNN input instances is only used in the baseline mel band magnitude feature experiments. The experiments involving mel scale weight initialization and context windowing are conducted with CNN architectures.

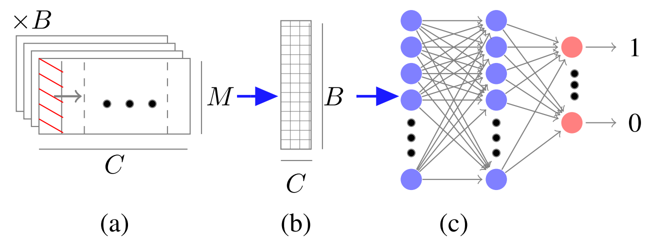


Fig. 4: An illustration of the CNN topology with mel scale weight initialization. Cascaded windows represent the $B$ feature maps and the crossed section represents the convolutional filter shape $M \times 1$.

### D. CNN with mel scale weight initialization and context windowing

CNNs can be regarded as the extensions of DNNs that can also learn the spatial and/or temporal information encoded in the multi-dimensional input features. CNNs have three properties that can be used to explain their difference with DNNs: local receptive field, weight sharing and pooling. Local receptive field means that each hidden neuron in a CNN layer is not connected to all the neurons in the previous layer, but only a local region of the input neurons to the convolutional layer. Multiple local fields with the same shape can be defined to extract different features in each region by working parallel to each other, and the weights and biases of each of these local fields define a feature map. The local region is shifted through all the input neurons and the same local region weights and biases are applied through the whole input space, *i.e.*, the weights are shared throughout the input. Finally, a pooling layer is often added after the convolutional layer to subsample the output and reduce the translational variations on the feature maps.

In this work, we aim to benefit from the temporal information encoded in the consecutive time frames (by using context windowing) and at the same time keep the filterbank structure in the first hidden layer of the deep architecture. In order to combine context windowing with mel scale weight initialization in a deep architecture, the weights in the first hidden layer are initialized with mel filterbank magnitude response and shared over the input frame features in the context window. This structure essentially becomes a constrained version of CNN using context windows as input.

The context window $\hat{\mathcal{X}} \in \mathbb{R}^{M\times C\times N}$ is formed as

$$\hat{\mathcal{X}}_n = \begin{bmatrix} \mathbf{X}_{1,n-\frac{C-1}{2}} & \mathbf{X}_{1,n-\frac{C+1}{2}} & \cdots & \mathbf{X}_{1,n} & \cdots & \mathbf{X}_{1,n+\frac{C-1}{2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_{M,n-\frac{C-1}{2}} & \mathbf{X}_{M,n-\frac{C+1}{2}} & \cdots & \mathbf{X}_{M,n}\cdots & \cdots & \mathbf{X}_{M,n+\frac{C-1}{2}} \end{bmatrix} \tag{6}$$

where $\hat{\mathcal{X}}_n = \hat{\mathcal{X}}_{:,:,n}$.

After context windowing, each input instance for CNN can be regarded as the spectrogram for $C$ frames, as shown in Figure 4(a). The target output for each input instance $\hat{\mathcal{X}}_n$ will be the same as the target output for the center frame of the context window, *i.e.*, $\hat{\mathbf{Y}}_{:,n} = \mathbf{Y}_{:,n}$. The number of feature maps in the convolutional layer is fixed to $B$ and for each feature map, the convolutional filter shape is fixed to $M \times 1$. Given the context window $\hat{\mathbf{X}}$ (where $\mathbf{X} = \mathcal{X}_{:,:,n}$, time frame index $n$ is omitted for simplicity), the output of the convolutional layer $\mathbf{Z} \in \mathbb{R}^{C\times B}$ is calculated as

$$\mathbf{Z}_{c,b} = \theta(\sum_{m=1}^{M} \mathbf{W}_{b,m}\hat{\mathbf{X}}_{m,c}) \qquad (c = 1, 2, ...C) \tag{7}$$

where $\mathbf{W}_{b,:}$ represents the weights for the $b^{th}$ feature map. The complete CNN topology corresponds to a single convolutional layer without pooling connected to two fully connected hidden layers before the output layer as illustrated in Figure 4(c).

We implement the mel scale weight initialization in the way that we introduce $B$ feature maps with dimensions $M \times 1$, and initialize the weights $\mathbf{W}_{b,m}$ with mel scale filterbank magnitude response $\mathbf{F}_{b,m}$. This way, each feature map is initialized with the coefficients of the corresponding mel band and filterbank learning is introduced to the deep architecture in the convolutional layer. As the name suggests, each *feature map* learns a different higher level *feature* and the learning includes making updates on the standard mel filterbank magnitude response.

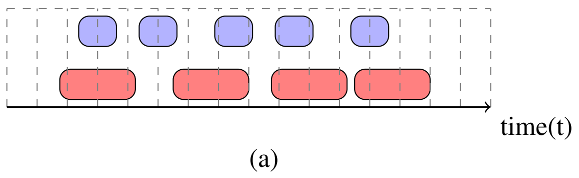
### E. Decision Thresholding

In a neural network based sound event detection system, logistic sigmoid output of the network can be treated as the posterior probability $\tilde{y}$ for each event class for each time frame, and the posterior probabilities of the sound events produced by a deep learning classifier should be converted into a binary detection format. In a real-life sound event detection task, the number of positive detections in a frame $n$ is not constant and it is in the range of $[0, K]$. In order to obtain multiple binary detection outputs in a single frame, posterior probabilities for each sound event is thresholded with a constant $\phi$ (in this work, $\phi = 0.5$ is used to set an unbiased probability threshold).

## IV. Evaluation

### A. Acoustic Data

We evaluate the performance of the proposed method with synthetic material that consists of sequences of occasionally overlapping sound events. There are 9 sound event classes that have been investigated in this work: alarm clock, baby crying, cat meowing, dog barking, door bell, footsteps, glass shattering, music and speech. The samples for these classes except music and speech are collected from *stockmusic* [1]. The total length of these samples is around 4270 seconds, with an average length of 17 seconds per sample. In addition, the speech samples are the clean samples from the training set (all speakers) of the 2nd CHiME challenge [33] and the music samples are 15-20 second randomly selected excerpts from music tracks with various genres [34]. The samples for each class are distributed randomly as 60 % in training set, 20 % in validation set and 20% in test set. In order to simulate the polyphony in real-life recordings, the collected samples are mixed in 15 minute mixtures in the following way. Among the samples collected from *stockmusic*, the samples having the same track are grouped together in either training, validation or test sets to avoid that samples recorded in the same environment would be easily recognized by the factors other than their sound characteristics. For each mixture, randomly selected samples from the event classes are added to the mixture by introducing a random time delay, adding a sample to the mixture and repeating this until the end of the mixture. By doing this for each class, a mixture with time-varying polyphony is obtained. The polyphony is roughly controlled with the range of the random time delay, as illustrated in Figure 5. The ground truth activity for each mixture is obtained by automatically annotating (labeling) the frames in the mixture where the sample from the corresponding event class is used. In order to avoid mislabeling of possible silent segments in the beginning and the end of the samples, mean RMS energy of the sample is calculated in short frames, and then, compared to this mean RMS energy, the frames in the beginning and end of the sample without enough RMS energy are left unlabeled. A total of 50 mixtures obtained this way for the whole dataset. Among the annotated frames, the polyphony percentage of the test set is given in Table I.



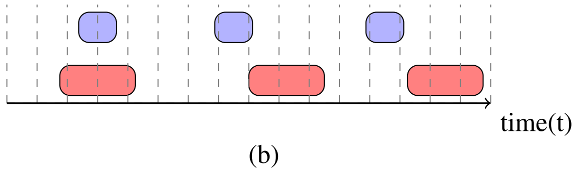


Fig. 5: The polyphony of the mixture is controlled with the length of the time delay in between samples: (a) high polyphony, (b) low polyphony.

[1]http://www.stockmusic.com

TABLE I: Polyphony level versus data amount percentage for the annotated frames in the test set.

| Polyphony | 1 | 2 | 3 |
|---|---|---|---|
| Percentage | 90.9 | 8.5 | 0.5 |

## B. Settings

In order to obtain the magnitude spectrum, the mixtures are divided into 40 ms time frames with 50% overlap. Hamming window is applied for each frame to reduce the boundary effect. Magnitude spectrum for each time frame is calculated by taking the absolute value of 1024-point FFT. The baseline model uses 40 mel band energies and two hidden layers for the DNN [6]. Consequently, our DNN method uses a fully-connected hidden layer with 40 neurons and for CNN, this hidden layer is replaced with convolutional layer with 40 feature maps. For both DNN and CNN methods, the first hidden layer is followed by 2 hidden layers of 200 neurons. During the training with stochastic gradient descent algorithm, learning rate is set to 0.1 and Nesterov accelerated momentum with value 0.2 is used. These network hyper-parameters are selected according to grid search over the validation set. Constant thresholding of the network outputs with $\phi = 0.5$ is used to get the binary outputs. Each experiment has been repeated ten times and the results are presented as the average accuracy over the experiments. This is done to take into account the accuracy changes due to random initialization of the network parameters. The experiments are conducted based on Keras neural networks library [35].

## C. Evaluation Metric

The evaluation metric for this work is frame-wise multi-label F1 score. Correct, wrong and missed detections of events are obtained for each input instance by comparing the estimated binary outputs to the target outputs. Then, precision and recall is calculated from the total number of correct, wrong and missed detections of events throughout the whole test set. F1 score is calculated as the harmonic mean of precision and recall.

## D. Overall Results

DNN and CNN architectures with different features, first hidden layer weight initialization methods and context window lengths has been experimented and the accuracy results are presented in Table II. Regardless of the context window length $C$, deep architectures trained with magnitude spectrum features outperform the mel band magnitude features. This can be seen as a promising step on using lower level features as input and automatically learning the higher level features through deep learning in sound related tasks. Moreover, CNNs whose first layer weights are initialized with mel filterbank magnitude response generally outperform the randomly initialized CNNs.

In addition to accuracy benefit, mel filterbank weight initialization also has an *understandability* benefit. Randomly initialized weights have an implicit final structure since

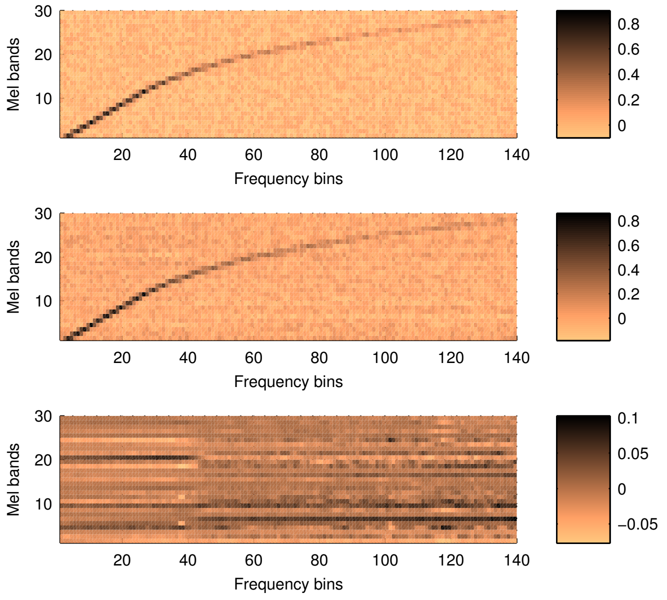


Fig. 6: (a): initial mel filterbank response, (b): final mel filterbank response, (c): difference between (a) and (b).

they map a very complex nonlinear function starting from random coefficients for this function. On the other hand, mel initialized CNNs provide understandable weights in the first hidden layer, where one can deduct the essential frequency bins for the particular sound event detection task. The initial and final mel filterbank responses for $C = 11$ is visualized in Figure 6. The final filterbank clearly preserves the nonlinear mel scale structure. Moreover, when we look at the difference between initial and final mel filterbank responses as given in Figure 6(c), we notice that the frequency bins in some mel bands are emphasized by decreasing the weights of the frequency bins in the same filter, but outside the mel band. This points to the fact that the frequencies in the given mel band play an important role in the detection of the sound events.

Consistent with our observations in [29], introducing time information encoded in successive frames, *i.e.*, using context windows rather than individual frames, provides great boost in accuracy. The accuracy nevertheless converges at around $C = 11$, which corresponds to 240 ms of audio signal with the given settings. Using the appropriately constrained CNN allows us to introduce context windowing to mel filterbank weight initialization. Consequently, the best detection accuracy (marked with bold in Table II) is obtained when these two methods are combined.

## E. Class-wise Results

The detection accuracy for each sound event class is presented in Table III. Context window length for each method in Table III is selected as $C = 11$, for which the accuracy is highest as presented in Table II. In sound event detection, the input usually consists of unstructured sounds, which makes it hard to both model the sound events and interpret the accuracy of the proposed methods for each sound event

TABLE II: Detection accuracy as a function of acoustic feature, network architecture and context window length $C$. *MEL*: mel band energy, *FFT*: magnitude spectrum, *random init*: complete random initialization of network weights, *mel init*: mel filterbank weight initialization in the first hidden layer weights, $\mu$ *and* $\sigma$ *match*: mean and variance matching.

| Feature | Architecture | C=1 | C=3 | C=5 | C=7 | C=9 | C=11 | C=13 |
|---|---|---|---|---|---|---|---|---|
| MEL | DNN random init | 69.3 | 75.3 | 77.1 | 77.9 | 77.9 | 78.9 | 78.9 |
| FFT | CNN random init | 71.4 | 76.1 | 78.2 | 79.4 | 80.6 | 81.2 | 80.7 |
| FFT | CNN mel init | 71.5 | 75.8 | 78.5 | 80.2 | 81.2 | 80.8 | 80.8 |
| FFT | CNN mel init ($\mu$ and $\sigma$ match) | 71.3 | 76.2 | 78.4 | 79.7 | 80.9 | **81.8** | 81.3 |

TABLE III: Detection accuracy as a function of acoustic feature & network architecture and sound event classes for context window length $C = 11$.

| Feature & Architecture | alarm clock | baby cry | cat meow | dog bark | door bell | footsteps | glass shatter | music | speech |
|---|---|---|---|---|---|---|---|---|---|
| MEL & DNN random init | 61.0 | 77.0 | 79.1 | 69.1 | 66.0 | 76.0 | 75.0 | 89.6 | 88.0 |
| FFT & CNN random init | 70.8 | 76.9 | 78.8 | 74.8 | 64.5 | 78.9 | 81.3 | 90.5 | 89.1 |
| FFT & CNN mel init | 65.0 | 77.6 | 82.2 | 74.4 | 57.7 | 77.7 | 81.7 | 90.0 | 89.3 |
| FFT & CNN mel init ($\mu$&$\sigma$ match) | 74.3 | 76.5 | 81.8 | 74.9 | 69.6 | 75.9 | 82.0 | 90.3 | 88.8 |

separately. However, for the *more* structured classes such as alarm clock and door bell (which often consist of multiple sinusoids recorded simultaneously), mel filterbank weight initialization provides a significant boost in accuracy. For the classes such as baby crying, footsteps, music and speech the accuracy difference between the methods seems negligible.

## V. Conclusions

In this work, we aim to combine the automatic feature learning capabilities of the deep learning architectures with the empirically obtained human perception knowledge. Instead of using the traditional, perceptually motivated mel band magnitudes as input features, we use magnitude spectrum features and initialize the first hidden layer weights with mel filterbank magnitude response, which essentially corresponds to computing the mel band magnitudes through the first hidden layer outputs. During learning process, the filterbank is updated to provide better discrimination over sound events. The proposed method does not only improve the detection accuracy over randomly initialized networks, but also provide *interpretable* weights in the first hidden layer, which can be used to deduct the important frequency bins in the detection of the given sound events.

## References

[1] D. Hollosi, J. Schröder, S. Goetze, and J. Appell, "Voice activity detection driven acoustic event classification for monitoring in smart homes," in *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*. IEEE, 2010, pp. 1–5.

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

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

[4] J. Geiger and K. Helwani, "Improving event detection for audio surveillance using Gabor filterbank features," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2015.

[5] J. T. Geiger, B. Schuller, and G. Rigoll, "Large-scale audio feature extraction and SVM for acoustic scene classification," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2013)*, 2013, pp. 1–4.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multilabel deep neural networks," in *Int. Joint Conf. on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[7] J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[8] J. Dennis, H. Tran, and E. 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] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[10] V. Bisot, S. Essid, and G. Richard, "HOG and subband power distribution image features for acoustic scene classification," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2015.

[11] J. B. Allen, "Cochlear modeling," *IEEE ASSP Magazine*, vol. 2, no. 1, pp. 3–29, 1985.

[12] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[13] S. S. Stevens and J. Volkmann, "The relation of pitch to frequency: A revised scale," *The American Journal of Psychology*, pp. 329–353, 1940.

[14] J. Makhoul and L. Cosell, "LPCW: An LPC vocoder with linear predictive spectral warping," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1976, pp. 466–469.

[15] D. O'shaughnessy, *Speech communication: human and machine*. Universities press, 1987.

[16] D. D. Greenwood, "A cochlear frequency-position function for several species 29 years later," *The Journal of the Acoustical Society of America*, vol. 87, no. 6, pp. 2592–2605, 1990.

[17] D. Greenwood, "The mel scale's disqualifying bias and a consistency of pitch-difference equisections in 1956 with equal cochlear distances and equal frequency ratios," *Hearing research*, vol. 103, no. 1-2, pp. 199–224, 1997.

[18] 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.

[19] D. Sadlier, N. E. O'Connor *et al.*, "Event detection in field sports video using audio-visual features and a support vector machine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225–1233, 2005.

[20] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[21] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2015, pp. 559–563.

[22] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic

events using deep neural networks," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2014.

[23] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–12, 2015.

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

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

[26] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[27] T. N. Sainath, B. Kingsbury, A. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 297–302.

[28] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2015.

[29] E. Cakir, "Multilabel sound event classification with neural networks," Master's thesis, Tampere University of Technology, Finland, 2014.

[30] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, and J. Williams, "Recent advances in deep learning for speech research at Microsoft," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8604–8608.

[31] D. P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005, online web resource. [Online]. Available: http://www.ee.columbia.edu/ dpwe/resources/matlab/rastamat/

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[33] E. Vincent *et al.*, "The second chimespeech separation and recognition challenge: Datasets, tasks and baselines," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 126–130.

[34] T. Heittola, "Automatic classification of music signals," Master's thesis, Department of Information Technology, Tampere University Of Technology, 2004.

[35] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.