Zhang and Wu *EURASIP Journal on Audio, Speech, and Music Processing* (2019) 2019:1
https://doi.org/10.1186/s13636-018-0144-6

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Discriminative frequency filter banks learning with neural networks

Teng Zhang* and Ji Wu

Abstract

Filter banks on spectrums play an important role in many audio applications. Traditionally, the filters are linearly distributed on perceptual frequency scale such as Mel scale. To make the output smoother, these filters are often placed so that they overlap with each other. However, fixed-parameter filters are usually in the context of psychoacoustic experiments and selected experimentally. To make filter banks discriminative, the authors use a neural network structure to learn the frequency center, bandwidth, gain, and shape of the filters adaptively when filter banks are used as a feature extractor. This paper investigates several different constraints on discriminative frequency filter banks and the dual spectrum reconstruction problem. Experiments on audio source separation and audio scene classification tasks show performance improvements of the proposed filter banks when compared with traditional fixed-parameter triangular or gaussian filters on Mel scale. The classification errors on LITIS ROUEN dataset and DCASE2016 dataset are reduced by 13.9% and 4.6% relatively.

Keywords: Discriminative frequency filter banks, Networks, Audio scene classification, Audio source separation

1 Introduction

Filter banks have been used for a long time to make time-frequency analysis of audio signals. The most commonly used short-time Fourier transform (STFT) [1] or wavelet transform [2] can decompose audio signals into sub-band components with certain time-frequency locations and resolutions. Filter banks implemented in the time domain [3] are usually shown as Fig. 1. Audio signals are convolved with M frequency-constrained filters, followed by averaging over an n_k-length window. For audio recognition tasks, such as speech recognition [4, 5], automatic speaker verification [6, 7], and audio scene classification [8, 9], the filter banks are used as a front-end feature extractor followed by a back-end classifier. For audio enhancement tasks, such as source separation [10, 11] and speech de-noising [12, 13], a perfect or near perfect reconstruction procedure combined with an up-sampling module and dual filter banks is needed. These fixed-parameter filters are usually in the context of psychoacoustic experiments, which need task-related expertise. To discriminatively learn parameters of filter banks remains a difficult problem.

*Correspondence: zhangteng1887@gmail.com
Department of Electronic Engineering, Tsinghua University, Beijing, China

1.1 Related work

In early pattern recognition studies [14], the input is first converted into some features, which are usually defined empirically by experts and believed to be suitable with recognition targets. Then, a design named discriminative feature extraction (DFE) [4, 15] is proposed to systematically train the overall recognizer in a manner consistent with the minimization of recognition errors. For audio signals, a DFE method with learnable filter banks is first investigated in [16]. In principle, the filter banks are composed of a finite or infinite number of filters. However, this needs careful investigation for the stability of the filters. Besides, the convolution operation in filter banks in the time-domain is time-consuming. Filter banks on FFT-based spectrums [17] have been studied for simplicity, which can be modeled as Eq. 1, where n is the discrete index of different filters, f is the frequency in hertz.

$$w_n(f) = \alpha_n g(c_n(p(f)); s_n(p(f))) \tag{1}$$

Filter banks are parameterized in the frequency domain with the frequency center c_n, bandwidth s_n, gain α_n, shape g, and frequency scale p. The result w_n is a continuous function defined in the frequency domain. When p is a linear function, filter banks are uniformly distributed in the frequency domain. However, there is a strong desire

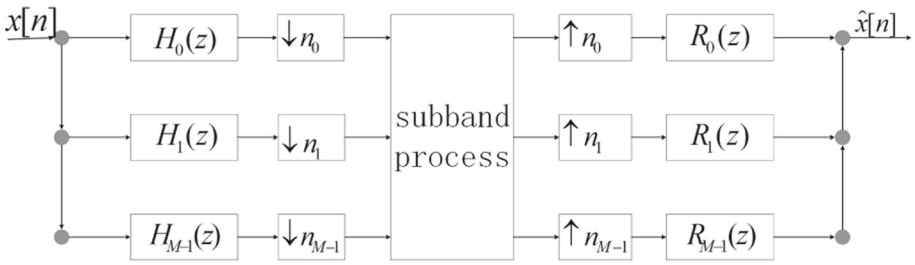


Fig. 1 Non-uniform filter banks. The terms $H_k(z)$ and $R_k(z)$ correspond to the z-transforms of the kth filter and dual filter, i.e., $H_k(z) = z^{-l_k}$ for some integers l_k and $R_k(z) = z^{l_k}$

to analyze audio signals similar to human ears, which means a non-linear function named auditory filter banks [18–20]. Based on psychoacoustics experiments, three non-linear mappings between the frequency and perceptual domain are commonly used, including the Bark scale [21], ERB scale [22], and Mel scale [23]. The parameters α_n, c_n, and s_n in Eq. 1 represent the frequency properties of w_n, which simulate the frequency selectivity in human ears. In [16], g is selected as a gaussian function because of its smoothness and tractability, correspondingly, the Mel filter banks use triangular filters [17]. When g is totally independent and not limited to any specific shape, w_n for each filter can be parameterized as a fully connected mapping from all frequency bins to a value.

Auditory filters of different shapes have been trained discriminatively for robust speech recognition [24]. Filter banks can also be trained discriminatively using Fisher discriminant analysis (FDA) method [25]. In recent years, deep neural networks (DNN) have achieved significant success in the field of audio processing and recognition because of its advantages in discriminative feature extraction. Standard filter banks computed in the time domain have been simulated using unsupervised convolutional restricted Boltzmann machine(ConvRBM) [26]. The speech recognition performance of ConvRBM features is improved compared to the Mel-frequency cepstrum coefficients (MFCCs), and the relative improvements are 5% on TIMIT test set and 7% on WSJ0 database using GMM-HMM systems. Discriminative frequency filter banks can also be learned together with the recognition error using a time-convolutional layer and a temporal pooling layer over the raw waveform [27]. The results in [27] show that the filter size and pooling operation play an important role in the performance improvement, but the temporal convolutional operation is time-consuming.

Filter banks implemented in the frequency domain are also studied with DNNs in recent years. When g in Eq. 1 is parameterized in all frequency bins, and the parameters are restricted to be positive using exponential function exp [28] or sigmoid [29], filter banks with multiple peaks and complicated shape are learned for specific tasks. However, further experiments show that the positive constraint is too weak to learn smooth and robust filter banks. When g in Eq. 1 is restricted to a gaussian shape, the gain, frequency center, and bandwidth in Eq. 1 can be learned using a neural network [30]. The triangular filter shape (commonly used to compute Mel scale features) is not investigated since it is piecewise differentiable and difficult to be incorporated into the scheme of a back-propagation algorithm.

1.2 Contribution of this paper

In this paper, we use a neural network structure to learn the frequency center, bandwidth, gain, and shape of filter banks adaptively, and investigate several different constraints on filter banks and the dual spectrum reconstruction problem.

Filter banks are said to be maximally decimated [3] if the channel decimation rates n_k in Fig. 1 are integers satisfying Eq. 2.

$$1 = \sum_{i=1}^{M-1} \frac{1}{n_i} \tag{2}$$

This condition means that there are more transformed sub-band coefficients per second than the original data points. In this case, the filter banks are overcomplete [31] and a perfect reconstruction from the sub-band coefficients is possible. However, in some scenarios, audio reconstruction from incomplete information is necessary because of the limitation of storage and computing resources, especially when the signals are sampled at a higher rate greater than or equal to 44.1 kHz. Speech reconstruction from MFCCs has been studied by predicting the fundamental frequency and voicing of a frame as intermediation [32–34]. The simplest case is that n_i in Eq. 2 equals to the frame length N, which is equivalent to filter banks implemented in the frequency domain in this paper.

As shown in Eq. 1, when filter banks are parameterized and learned using neural networks, a major concern is the constraint to the shape of its responses in the frequency

range. When the constraint is weak [28, 29], the number of parameters is too large to learn smooth and robust filter banks in some scenarios. When the constraint is a basic shape function and this function is piecewise differentiable such as the triangular shape [30], the model cannot be trained using a back-propagation algorithm.

At the same time, the sub-band processing module in Fig. 1 may introduce distortions, particularly if the sub-bands are not equally processed, in this case, signal reconstruction in the frequency domain is not analytical.

In this paper, the major contributions are summarized as follows:

- *Approximate continuous shape function*: shape constraints play an important role in discriminative frequency filter banks. Few investigations have been conducted to compare different shape constraints, because that commonly used shapes such as triangular shapes are piecewise differentiable. We use steep sigmoid functions and other basic functions to approximate desired shapes. This makes a further study on shape constraints possible.
- *Comparison of different constraints*: in Eq. 1, different selections of trainable parameters can result in different implementations of filter banks. In this paper, we select six different constraints to investigate their applicable condition. When all parameters are constant, we adopt triangular and gaussian shapes whose frequency centers distribute uniformly in the Mel-frequency scale. For weak constraints, we conduct experiments similar to [28, 29]. For strong constraints, both gaussian and triangular constraints are used to train the frequency center, bandwidth, and gain in Eq. 1.
- *Reconstruction from incomplete filter bank coefficients*: in this paper, the amount of filter bank coefficients is much less than original data points, so the reconstruction can be seen as a process of solving overdetermined linear equations. We use a neural network to implement this reconstruction process, and a well-designed regularization method is used to make sure that the filter banks are bounded input bounded output (BIBO-stable).

The paper is organized as follows. Next section briefly describes the Mel-frequency scale used in this paper and introduce the uniformly distributed filter banks with constant parameters as the baseline. Section 3 introduces the analytical and experimental settings of our proposed filter bank learning framework. Then, network structures used in our proposed methods are introduced in Section 4. Section 5 conducts several experiments to show the performance of discriminative frequency filter banks in terms of source separation and audio scene classification tasks. Finally, we conclude our paper and give directions for future work in Section 6.

2 Background

Filter banks are used to model the frequency selectivity of an auditory system in many applications. Traditionally, the design of filter banks is motivated by psychoacoustic experiments, such as the detection of tones in noise maskers [35], or by physiological experiments such as observing the mechanical responses of the cochlea when a sound reaches the ear [36, 37]. The frequency center, bandwidth, and energy gain in the frequency response of filter banks are consistent with the position and vibration patterns in the ear. In the history of auditory filter banks [35], rounded exponential family [38] and gammatone family [39] are the most widely used families. We use the simplest form of these two families, triangular case for the rounded exponential family and gaussian case for the gammatone family, to construct our filter banks in the frequency domain. In this section, we introduce the commonly used Mel-frequency filter banks.

2.1 Mel-frequency scale

The perceptual frequency scale is usually a mapping between the linear frequency domain and the nonlinear perceptual frequency domain. The Mel-frequency scale is the result of a classic psychoacoustical test conducted by Stevens and Volkman [40], which provides the relation between the real frequency and hearing pitch. The conversion from the linear frequency to Mel-scale [41] is as follows, where f is frequency in hertz.

$$\text{Mel}(f) = 1127\log_2\left(1 + \frac{f}{700}\right) \tag{3}$$

2.2 Mel-frequency filter banks

The commonly used MFCC features in the field of speech recognition are computed based on Mel-frequency filter banks. It is a common practice to construct filters distributing uniformly in the Mel-frequency scale, and the bandwidth is often 50% overlapped between neighboring filters.

When the filter shape is restrained using Eq. 4, triangular filter banks are constructed in the Mel-frequency scale. For gaussian filter banks, the bandwidth is 4σ of a gaussian distribution as Eq. 5. These two types of filter banks are the baselines in this paper, respectively named TriFB and GaussFB. Although they are combinations of existing works, we implement our own implementation of these two methods in this paper. In Eqs. 4 and 5, c_n represents the frequency center, s_n represents the bandwidth, mel is the unit of Mel-frequency scale, Tri and Gauss are the triangular and gaussian filter banks defined in the Mel-frequency scale.

$$\mathrm{Tri}(n) = \begin{cases} \frac{2}{s_n}(\mathrm{mel} - c_n) + 1, c_n - \frac{s_n}{2} \le \mathrm{mel} \le c_n \\ \frac{2}{s_n}(c_n - \mathrm{mel}) + 1, c_n \le \mathrm{mel} \le c_n + \frac{s_n}{2} \\ 0, \text{elsewhere} \end{cases} \quad (4)$$

$$\mathrm{Gauss}(n) = \exp\left(-\frac{8(\mathrm{mel} - c_n)^2}{s_n^2}\right) \quad (5)$$

3 Discriminative filter bank learning

For generality, we consider in this section a discriminative filter bank learning framework based on a neural network as shown in Fig. 2.

The input audio signal is first transformed to a sequence of vectors using STFT, the STFT result can be represented as $\boldsymbol{X}_{1...T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. T is determined by the frame shift in STFT, corresponding to the time resolution in the frame theory [42]. The dimension of each vector $\boldsymbol{x}$ can be labeled as N, which is determined by the frame length.

The discriminative frequency filter banks in Fig. 2 can be simplified as linear transformations f_θ, the output of this module can be represented as $\boldsymbol{Y}_{1...T} = \{f_\theta(\boldsymbol{x}_1), f_\theta(\boldsymbol{x}_2), ..., f_\theta(\boldsymbol{x}_T\})$. θ are the parameters of filter banks defined similar to Eq. 1. The dimension of each $\boldsymbol{y}_t = f_\theta(\boldsymbol{x}_t)$ here is equal to M, which is the number of filters.

The back-end application modules in Fig. 2 vary from different applications. For audio scene classification task, they will be deep convolutional neural networks followed by a softmax layer to convert the feature maps to the corresponding categories. However, for audio source separation task, the modules will be composed by a binary gating layer and some spectrogram reconstruction layers. We simplify all these situations and define the back-end application modules as non-linear functions f_β. The filter bank parameters θ can be trained jointly with the back-end parameters β using a back-propagation method in neural networks.

In this framework, filter banks work as a set of weights on a spectrum vector $\boldsymbol{x}_t$ as Eq. 6. Each $\boldsymbol{w}_k$ is a filter with positive values and a bounded range.

$$\boldsymbol{y}_t = f_\theta(\boldsymbol{x}_t) = \left\{\boldsymbol{w}_1^T\boldsymbol{x}_t, \boldsymbol{w}_2^T\boldsymbol{x}_t, ..., \boldsymbol{w}_m^T\boldsymbol{x}_t\right\} \quad (6)$$

In this paper, we consider two types of constraints on filter banks.

- *Shape constraint*: in this case, the amplitude of filter's frequency response is constrained to be a special shape, and only the frequency center, bandwidth, and gain of the filter remain to be trained. The gaussian shape has been investigated in [16, 30]. We will focus on the piecewise differentiable situation such as the triangular shape.
- *Positive constraint*: when all the weights of filters are independent but only constrained to be positive, more complicated filter banks can be learned. Exponential functions such as exp [28] and sigmoid [29] have been used together with a bandwidth constraint for the filters. We investigate two new positive constraints ReLU and square, and discuss their performances associated with the bandwidth constraint.

3.1 Shape constraints of discriminative frequency filter banks

Triangular filters are commonly used to compute Mel-scale filter bank features in many audio applications such as speech recognition. However, when we use a triangular shape described in Eq. 4 to restrict the discriminative frequency filter banks in Fig. 2, the backward propagation

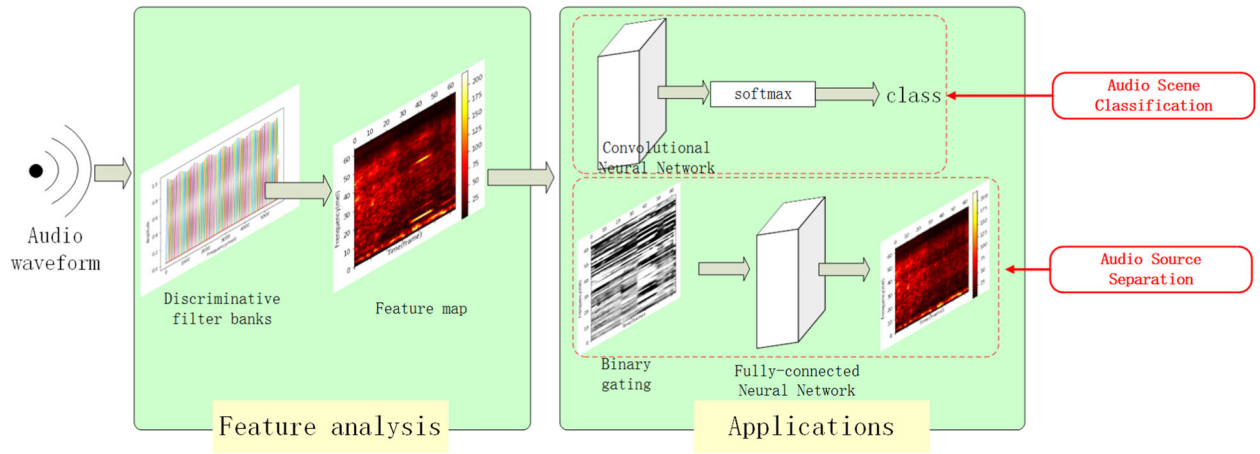


Fig. 2 Discriminative filter bank learning framework. The left part of the framework is the feature analysis procedure including STFT and discriminative filter banks. The right part is the application example of the extracted feature map, such as audio scene classification and audio source separation. Discriminative filter banks in the feature analysis procedure and the back-end application modules are stacked into a deep neural network

process is blocked because of the discontinuous point in the triangular shape.

Instead of using the piecewise continuous form of a triangular shape, we decompose it into piecewise continuous step functions and linear functions as Fig. 3a. We define the piecewise step function as Eq. 7. Then, a mathematical representation of the decomposition can be shown as Eq. 8. α_n is the gain parameter, c_n, s_n, and mel in this formula have been defined in Eq. 4.

$$\mathrm{rec}(x, x_0) = \begin{cases} 1, x > x_0 \\ 0, \text{elsewhere} \end{cases} \tag{7}$$

$$\begin{aligned} f_1(\mathrm{mel}) &= \mathrm{rec}\left(\mathrm{mel}, c_n - \frac{s_n}{2}\right)(1 - \mathrm{rec}(\mathrm{mel}, c_n)) \\ l_1(\mathrm{mel}) &= \frac{2}{s_n}(\mathrm{mel} - c_n) + 1 \\ f_2(\mathrm{mel}) &= \left(1 - \mathrm{rec}\left(\mathrm{mel}, c_n + \frac{s_n}{2}\right)\right)\mathrm{rec}(\mathrm{mel}, c_n) \\ l_2(\mathrm{mel}) &= \frac{2}{s_n}(c_n - \mathrm{mel}) + 1 \\ w_n(\mathrm{mel}) &= \alpha_n(f_1 l_1 + f_2 l_2) \end{aligned} \tag{8}$$

We use a sigmoid function $\mathrm{sig}(x, x_0) = \frac{1}{1+e^{-r_0(x-x_0)}}$ to approximate the step function and get an approximate triangular decomposition as Eq. 9. In this formula, r_0 represents the steep rate of the sigmoid function. Figure 3b is an example when r_0 is 10.

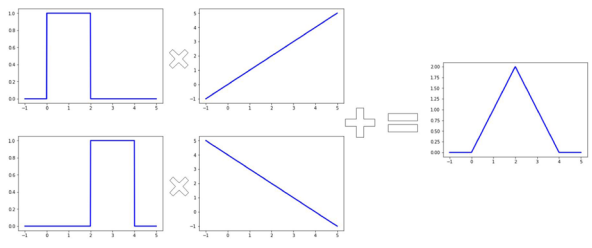
$$\begin{aligned} f_1(\mathrm{mel}) &= \mathrm{sig}\left(\mathrm{mel}, c_n - \frac{s_n}{2}\right)(1 - \mathrm{sig}(\mathrm{mel}, c_n)) \\ l_1(\mathrm{mel}) &= \frac{2}{s_n}(\mathrm{mel} - c_n) + 1 \\ f_2(\mathrm{mel}) &= \left(1 - \mathrm{sig}\left(\mathrm{mel}, c_n + \frac{s_n}{2}\right)\right)\mathrm{sig}(\mathrm{mel}, c_n) \\ l_2(\mathrm{mel}) &= \frac{2}{s_n}(c_n - \mathrm{mel}) + 1 \\ w_n(\mathrm{mel}) &= \alpha_n(f_1 l_1 + f_2 l_2) \end{aligned} \tag{9}$$

The trainable parameters in Eq. 9 are the frequency center c_n, bandwidth s_n, and gain α_n. The goal of the training procedure is to minimize some objective loss ϵ. The derivative of an objective loss given trainable parameters can be calculated by back-propagating error gradients.

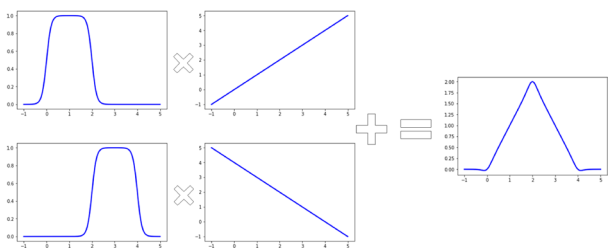
3.2 Positive constraint of discriminative frequency filter banks

Another selection of discriminative frequency filter banks is a set of independent weights $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m\}$. The only constraint is that these weights should be positive to keep their physical meaning of the filters. There are a couple of options to keep them positive:

- *Exponent*: for every parameter w_{ij}, we make it positive by transform it to $v_{ij} = \exp(w_{ij})$[28]. If $w_{ij} \sim N(\mu, \sigma)$, v_{ij} satisfies the log-normal distribution, where the mean of v_{ij} is $e^{\mu+\frac{\sigma^2}{2}}$ and the variance of v_{ij} is $\left(e^{\sigma^2} - 1\right)e^{2\mu+\sigma^2}$.
- *Sigmoid*: for every parameter w_{ij}, we use the sigmoid function $v_{ij} = \frac{1}{1+\exp(-w_{ij})}$[29] to ensure the parameters positive. If $w_{ij} \sim N(\mu, \sigma)$, v_{ij} satisfies a logit-normal distribution, where the moments of v_{ij} is not analytical, but the numerical calculating results have been discussed in [43].
- *ReLU*: for every parameter w_{ij}, we simply make $v_{ij} = 0$, when $w_{ij} < 0$ and $v_{ij} = w_{ij}$, when $w_{ij} \geq 0$. This will lead to a folded normal distribution. When $w_{ij} \sim N(\mu, \sigma)$, the mean of v_{ij} is $\sigma\sqrt{\frac{2}{\pi}}e^{-\frac{\mu^2}{2\sigma^2}} + \mu$ and the variance of v_{ij} is $\mu^2 + \sigma^2 - [\mathrm{mean}(v_{ij})]^2$.
- *Square*: the last option to make the parameters positive is that $v_{ij} = w_{ij}^2$. Then, v_{ij} is a variable satisfying a chi-squared distribution. The mean of v_{ij} is $\sigma^2\left(1 + \mu^2\right)$, and the variance of v_{ij} is $\sigma^4\left(2 + 4\mu^2\right)$.



(a) Triangular-shape decomposition with piecewise continuous step functions and linear functions.



(b) Approximate triangular-shape decomposition with sigmoid functions and linear functions, when steep rate is 10.

Fig. 3 Triangular-shape decomposition. **a** The accurate decomposition using piecewise continuous step functions and linear functions. **b** The approximate decomposition using sigmoid functions and linear functions

Without loss of generality, if we initialize the parameters with a gaussian distribution $w_{ij} \sim N(0, 0.1)$, the moments of the four positive transformations can be calculated as follows:

- Exponent: mean $=$ 1.0, variance $=$ 0.01.
- Sigmoid: mean $=$ 0.5, variance $\approx$ 0.01.
- ReLU: mean $\approx$ 0.08, variance $\approx$ 0.01.
- Square: mean $\approx$ 0.01, variance $\approx$ 0.0002.

In this section, we consider two variants of discriminative frequency filter banks. If the frequency center c_n and bandwidth s_n in Eq. 1 are constant and the filter weights are restrained to be positive, the filter weights are limited in the range of bandwidth. All the above distributions can be good solutions. Another case is that the filter weights are totally independent. In this case, the resulting distributions of the exponent and sigmoid constraints mean that most filter weights are not zero, which violates the physical meaning of filter banks. In order to fulfill the physical meaning, the moments of positive transformations should be around $N(0.1, 0.01)$, which is approximately calculated using the Mel-frequency triangular filter banks defined in Section 2.2. The inverse calculation of these positive transformations shows that when the parameters are initialized $w \sim N(-3.0, 2.0)$, the exponent and sigmoid constraints may result in meaningful distributions.

Thus, when the filter banks are constrained by constant bandwidths and frequency centers, all these positive constraints are suitable. But when the filter weights are totally independent, only ReLU and square constraints are suitable, unless we can perform elaborate initialization for different positive transformations. Our experiments in Section 3.3 demonstrate our conclusion.

3.3 Reconstruction from filter bank coefficients

In the traditional design of filter banks as Fig. 1, the completeness of filter banks is determined by the number of filters M and the channel decimation rate n_k. In our proposal of discriminative frequency filter banks, n_k is equivalent to the frame length N. And in general, M is less than N for the purpose to reduce the computational cost and extracting significant features. In this case, the filter banks are incomplete and hence, the perfect spectral reconstruction from the filter bank coefficients is impossible.

As described before, the spectrum $\boldsymbol{x}_t$ is first transformed to the Mel-frequency scale using a transformation matrix derived from Eq. 3. Then, the filter banks work as a set of weights on it as Eq. 6. Thus, the conversion from spectrum vectors to filter bank coefficients can be represented as Eq. 10. $\boldsymbol{M}$ is the Mel-frequency transition matrix, and $\boldsymbol{F}$ are the discriminative frequency filter banks.

$$\boldsymbol{y}_t = \boldsymbol{x}_t \boldsymbol{M} \boldsymbol{F} \tag{10}$$

The spectrum reconstruction process can be simplified as a reconstruction transformation as Eq. 11. $\boldsymbol{R}$ is the reconstruction matrix, and the parameters in $\boldsymbol{R}$ can be trained jointly with the parameters of filter banks in $\boldsymbol{F}$.

$$\hat{\boldsymbol{x}}_t = \boldsymbol{y}_t \boldsymbol{R} \tag{11}$$

The problem of finding the optimal reconstruction matrix $\boldsymbol{R}$ and filter bank matrix $\boldsymbol{F}$ is equivalent of finding the solution of a linear system [44] as Eq. 12. $\boldsymbol{R}^+$ is the Moore Penrose pseudoinverse [45] of $\boldsymbol{R}$ and has an approximate numerical representation of $\boldsymbol{MF}$. Here, we define the condition number [46] for $\boldsymbol{R}$ as Eq. 13.

$$\boldsymbol{R}\boldsymbol{R}^+ \boldsymbol{x}_t = \hat{\boldsymbol{x}}_t \tag{12}$$

$$\mathrm{cond}(\boldsymbol{R}) = \| \boldsymbol{R} \| \cdot \| \boldsymbol{R}^+ \| \leq \left(\| \boldsymbol{R} \| + \| \boldsymbol{R}^+ \| \right)^2 \tag{13}$$

In Eq. 13, $\mathrm{cond}(\boldsymbol{R})$ means the condition number of $\boldsymbol{R}$ and $\| \cdot \|$ means the Frobenius norm of a matrix.

A large condition number implies that the linear system is ill-conditioned in the sense that small errors in the input can lead to huge errors in the output. So, we modify the reconstruction loss by adding an L2-regularization constraint to keep the linear system stable. This is also known as the bounded-input, bounded-output (BIBO) stability [47].

The L2-regularization for different types of filter banks in Sections 3.1 and 3.2 are discussed respectively as follows.

- *Shape constraint*: for shape constraints in Section 3.1, parameters such as the frequency center c_n and bandwidth s_n, do not contribute to the regularization. Regularization of the gain α_n should be added up across the bandwidth.
- *Positive constraint*: for positive constraints in Section 3.2, all parameters contribute to the regularization. The positive weights v_{ij} should replace the filter bank parameters w_{ij} to calculate the regularization, but the regularization of reconstruction parameters r_{ij} remain unchanged.

3.4 Reconstruction vs classification

For spectrum reconstruction-related tasks as described in Eq. 11, the output size of the reconstruction system is NT, where N is the FFT length, and T is the number of frames. Thus, the number of equations in optimizing the reconstruction matrix $\boldsymbol{R}$ and filter bank matrix $\boldsymbol{F}$ is DNT, where D is the number of audio samples. Meanwhile, for positive constraints, the number of parameters in $\boldsymbol{R}$ and $\boldsymbol{F}$ is about

$2NM$, where M is the number of filter banks. For shape constraints, the number of parameters is about $3M+NM$. M is usually much less than DT, so the reconstruction usually can be seen as a process of solving overdetermined linear equations.

Correspondingly, when the output of filter banks is followed by a classifier, the number of equations in solving the classification task is DC, where C is the number of classes. The number of parameters is $MN + MC$ for positive constraints, and $3M + MC$ for shape constraints. In some small-scale applications, DC is less than MN. The classification is equivalent of solving underdetermined linear equations for positive constraints. Over-fitting is a notorious issue in this scenario. This phenomenon can be seen in Section 5.5.

4 Model description

As described in Section 3, the discriminative frequency filter banks we proposed here can be integrated into a neural network (NN) structure. The parameters of the models are learned jointly with the target of a specific task. In this section, we introduce two NN-based structures respectively for audio source separation and audio scene classification tasks.

4.1 Audio source separation

In Fig. 4a, the NN structure for audio source separation tasks is divided into three steps. The module of discriminative filter banks is implemented as Eq. 6, which can be denoted as h_1. The reconstruction layer is constructed using a fully connected layer and can be denoted as h_3.

We attempt the audio separation from an audio mixture using a simple masking method [48], which can be represented as a binary masking module in Eq. 14 and denoted as h_2. In Eq. 14, y_{tj} is an element of the feature map $\boldsymbol{Y}$, m_{ji} is a trainable parameter of this layer. The output of this layer is a linear projection modulated by the gates $\boldsymbol{g}_t$. These gates multiply each element of the matrix $\boldsymbol{Y}$ and control the information passed on in the hierarchy. Stacking these three layers on the top of input $\boldsymbol{X}$ gives a representation of the separated clean spectrogram $\hat{\boldsymbol{X}} = h_3 \circ h_2 \circ h_1(\boldsymbol{X})$, the symbol $\circ$ is used here to represent the connection between different layers.

$$\begin{aligned} g_{ti} &= \text{sigmoid}\left(\textstyle\sum_{j=1}^{N} y_{tj} m_{ji}\right) \\ o_{ti} &= y_{ti} g_{ti} \end{aligned} \tag{14}$$

Neural networks are trained on a frame error (FE) minimization criterion, and the corresponding weights are adjusted to minimize the square errors over the whole training dataset. The error of the mapping is given by Eq. 15, where $\boldsymbol{x}_t$ is the targeted clean spectrum, and $\hat{\boldsymbol{x}}_t$ is the corresponding separated representation. As commonly used, L2-regularization is typically chosen to impose a penalty on the complexity of the mapping, which is the λ term in Eq. 15. However, when the layer of discriminative filter banks is implemented with shape constraints, the elements of $\boldsymbol{w}_1$ have definitude physical meanings. Thus, the L2-regularization is operated only on the upper two layers in this model. In this case,

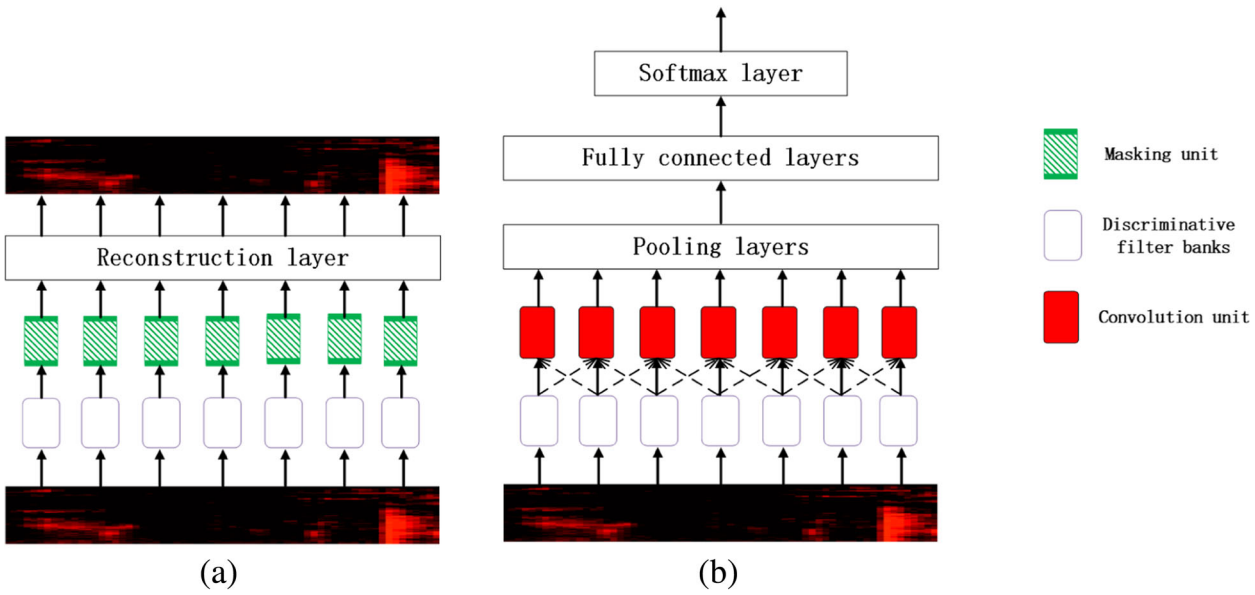


Fig. 4 NN-based structures with proposed methods. **a** is the NN structure for audio source separation tasks. **b** is the NN structure for audio scene classification tasks

the network in Fig. 4a can be optimized by the back-propagation method.

$$\epsilon = \sum_{t=1}^{T} \| \boldsymbol{x}_t - \hat{\boldsymbol{x}}_t \|^2 + \lambda \sum_{l=2}^{3} \| \boldsymbol{w}_l \|^2 \tag{15}$$

4.2 Audio scene classification

In Fig. 4b, a feature extraction structure including the discriminative frequency filter banks is proposed to systematically train the overall recognizer in a manner consistent with the minimization of recognition errors.

The NN structure for audio scene classification tasks can be divided into five steps, where the first layer of discriminative frequency filter banks is implemented using Eq. 6. The convolutional and pooling layers are conducted using the network structure described in [49]. In general, let $\boldsymbol{z}_{i:i+j}$ refer to the concatenation of frames after discriminative filter banks $\boldsymbol{y}_i, \boldsymbol{y}_{i+1}, ...\boldsymbol{y}_{i+j}$. The convolution operation involves a filter $\boldsymbol{w} \in R^h$, which is applied to a window of h frames to produce a new feature. For example, a feature c_i is generated from a window of frames $\boldsymbol{y}_{i:i+h-1}$ by Eq. 16, where $b \in R$ is a bias term and f is a non-linear function. This filter is applied to each possible window of frames to produce a feature map $\boldsymbol{c} = [c_1, c_2, ...c_{T-h+1}]$. Then, a max-overtime pooling operation [50] over the feature map is applied and the maximum value $\hat{c} = \max(\boldsymbol{c})$ is taken as the feature corresponding to this filter. Thus, one feature is extracted using one filter. This model uses multiple filters with varying window sizes to obtain multiple features.

$$c_i = f(\boldsymbol{w} \cdot \boldsymbol{y}_{i:i+h-1} + b) \tag{16}$$

The features extracted from the convolutional and pooling layers are then passed to a fully connected layer and a softmax layer to output the probability distribution over categories. The classification loss of this model is given by Eq. 17, where n is the number of audios, k is the number of categories, $l_{i,j}$ is the category label, and $p_{i,j}$ is the probability distribution produced by the NN structure. In this case, the network in Fig. 4b can be optimized by the back-propagation method.

$$\epsilon = \sum_{i=1}^{n} \sum_{j=1}^{k} l_{i,j} \cdot \log(p_{i,j}) + \lambda \sum_{l=2}^{4} \| \boldsymbol{w}_l \|^2 \tag{17}$$

5 Experiments

To illustrate the properties and performance of the discriminative frequency filter banks proposed in this paper, we conduct three experiments respectively on spectrum reconstruction, audio source separation and audio scene classification tasks. In the first experiment, several groups of comparisons are made on reconstruction errors to verify the assumption and conclusion we proposed in Section 3. Moreover, we have two more experiments to test the applications of the discriminative frequency filter banks to audio source separation and audio scene classification tasks.

5.1 Filter bank settings

All experiments conducted below make a comparison between the discriminative frequency filter banks that can be trained using neural networks and the fixed-parameter filter banks described in Section 2.2. The detailed settings are as follows:

- *TriFB*: frequency centers of the filters distribute uniformly in the Mel-frequency scale, bandwidths are 50% overlapped between neighboring filters, the gain is 1, and the shape is restrained with Eq. 4.
- *GaussFB*: frequency centers of the filters distribute uniformly in the Mel-frequency scale, bandwidths are 4σ of an gaussian distribution as Eq. 5, the gain is 1, and the shape is restrained with Eq. 5.
- *TriFB-DN*: in order to achieve a fair comparison with TriFB, the initialization of the frequency centers, bandwidths, and gain of the filters are the same as TriFB, the shape is restrained with Eq. 9, and the gain and bandwidths are guaranteed to be positive with a square constraint described in Section 3.2.
- *GaussFB-DN*: in order to achieve a fair comparison with GaussFB, the initialization of the frequency centers, bandwidths, and gain of the filters are the same as GaussFB, the shape is restrained with Eq. 5. Other settings are the same as TriFB-DN.
- *BandPosFB-DN*: frequency centers and bandwidths are the same as GaussFB, all parameters are initialized using $N(0, 0.1)$, and are guaranteed to be positive with the square constraint described in Section 3.2. The shape is not restrained.
- *PosFB-DN*: the parameters are initialized using $N(0, 0.1)$ and are guaranteed to be positive with the square constraint described in Section 3.2. There are no constraints for the frequency centers, bandwidths, and shape of the filters.

5.2 Dataset and experimental setup

In this section, we employ three datasets to conduct the experiments. MIR-1K dataset [51] is utilized to implement the spectrum reconstruction and audio source separation experiments. LITIS ROUEN [52] and DCASE2016 [53] datasets are used for audio scene classification experiments.

Details of these datasets are listed as follows:

- *MIR-1K dataset*: this dataset consists of 1000 song clips recorded at a sample rate of 16,000 Hz, with durations ranging from 4 to 13 s. The dataset is then

utilized with four training/testing splits. In each split, 700 examples are randomly selected for training and the others for testing. We use the mean average accuracy over the four splits as the evaluation criterion.

- *LITIS ROUEN dataset*: this is the largest publicly available dataset for ASC to the best of our knowledge. The dataset contains about 1500 min of audio scene recordings belonging to 19 classes. Each audio recording is divided into 30-s examples without overlapping, thus obtaining 3026 examples in total. The sampling frequency of the audio is 22,050 Hz. The dataset is provided with 20 training/testing splits. In each split, 80% of the examples are kept for training and the other 20% for testing. We use the mean average accuracy over the 20 splits as the evaluation criterion.
- *DCASE2016 dataset*: the dataset is released as task 1 of the DCASE2016 challenge. We use the development data in this paper. The development data contains about 585 min of audio scene recordings belonging to 15 classes. Each audio recording is divided into 30-s examples without overlapping, thus obtaining 1170 examples in total. The sampling frequency of the audio is 44,100 Hz. The dataset is divided into fourfolds. Our experiments obey this setting, and the average performance will be reported.

In all experiments, the audio signal is first transformed using STFT with the frame length of 1024 and the frame shift of 10 ms, so the size of audio spectrums is 513×128. The mini-batch size is set to be 50, and the learning rate is initialized with 0.001.

In our audio source separation experiments, the number of discriminative filters is set to be 64, other parameters are set as described in Section 4.1. When the spectrum reconstruction is needed, the regularization coefficient is set to be 0.0001. Training is done using the Adam [54] update method and is stopped after 500 training epochs.

In our audio scene classification experiments, the number of discriminative filters is also set to be 64. For both LITIS ROUEN and DCASE2016 datasets, we use rectified linear units; the window sizes of convolutional layers are $64 \times 2 \times 64$, $64 \times 3 \times 64$, and $64 \times 4 \times 64$, and the fully connected layers are $196 \times 128 \times 19(15)$. For DCASE2016 dataset, we use the dropout rate of 0.5. Training is done using the Adam update method and is stopped after 100 training epochs.

5.3 Properties of discriminative frequency filter banks

In this experiment, we analyze the properties of the discriminative frequency filter banks using the clean music audios in MIR-1K dataset. The binary gating layer in Fig. 4a is left out for simplicity. To quantify the performance of our method, we evaluate the reconstruction performance using the metric of signal to distortion ratios (SDR). In Eq. 18, $\hat{x}$ is the reconstructed signal and x is the source signal.

$$\mathrm{SDR}(x, \hat{x}) = 10\log_{10}\left(\frac{||x||^2}{||x - \hat{x}||^2}\right) \tag{18}$$

Table 1 shows the reconstruction SDR under different positive constraints. In order to exclude the influence of filter numbers, these experiments are configured with $M = 32$ and $M = 64$, respectively. The consistent results in Table 1 demonstrate that exponent, sigmoid, ReLU, and square positive constraints show similar performances when parameters are constrained by fixed frequency center and bandwidth, but ReLU and square positive constraints perform much better than exponent and sigmoid constraints when parameters are totally independent and initialized with $N(0, 0.1)$. As we have discussed in Section 3.2, in this case, ReLU and square constraints can result in a similar parameter distribution with the traditional Mel-frequency triangular filter banks, but exponent and sigmoid constraints will result in an entirely different distribution, which violates the physical meaning of the filter banks. However, when the initialization for exponent and sigmoid constraints are finely designed to be $N(-3.0, 2.0)$, the results improve a lot for totally independent situations. Taken together, the performance of ReLU and square positive constraints are more stable, and their performances are similar, so we can select the square constraint in follow-up experiments because of its differentiability.

For audio scene classification tasks, we use DCASE2016 dataset to examine the rationality of our selection. Table 2 is the classification performance on the validation part of DCASE2016 dataset. The NN structure is implemented

Table 1 Reconstruction SDR under different positive constraints in decibel

Initialization	Constraint	M = 32		M = 64	
		F-B	T-I	F-B	T-I
N(0, 0.1)	Exponent	8.57	6.72	13.10	6.74
	Sigmoid	8.54	8.89	12.84	9.43
	ReLU	8.45	14.44	12.84	18.54
	Square	8.57	14.44	12.84	18.24
N(− 3.0, 2.0)	Exponent	9.02	14.44	13.25	17.70
	Sigmoid	8.36	14.31	12.84	17.97

F-B represents the parameters with fixed frequency center and bandwidth; *T-I* represents the totally independent parameters; *M* represents the number of filters; *N* means Gaussian distribution

Table 2 Audio scene classification performance under different positive constraints

Initialization	Constraint	F-B		T-I	
		Accuracy	MCC	Accuracy	MCC
$N(0, 0.1)$	Exponent	77.21	75.85	71.20	69.48
	Sigmoid	77.87	76.60	70.75	68.97
	ReLU	77.37	76.05	76.92	75.59
	Square	77.89	76.63	75.96	74.62
$N(-3.0, 2.0)$	Exponent	77.97	76.70	73.52	72.04
	Sigmoid	77.16	75.82	71.44	69.81

F-B represents the parameters with fixed frequency center and bandwidth; *T-I* represents the totally independent parameters; *N* means Gaussian distribution

as Fig. 4b, and the training process is stopped after 180 epochs. Accuracy and Matthews correlation coefficient (MCC) are employed to make the comparison. The results are consistent with Table 1. For all these positive constraints and initialization schemes, the classification performances are similar when parameters are constrained by fixed frequency center and bandwidth. However, when parameters are totally independent, ReLU and square positive constraints are more stable. If the parameters are initialized with $N(0, 0.1)$, it is difficult to converge to an optimal solution for exponent and sigmoid constraints. Therefore, our selection of square positive constraint also works for audio scene classification tasks.

Table 3 is the reconstruction SDRs with and without regularization. The results in the last two columns show the performance improvement by adding proper L2-regularization constraint as described in Section 3.3. Comparing with TriFB and GaussFB, the results of the four discriminative frequency filter bank models improve a lot. $\boldsymbol{MF}$ is the Moore Penrose pseudoinverse of $\boldsymbol{R}$ in Eq. 12; thus, $\boldsymbol{R}$ is the dual matrix determined by $\boldsymbol{F}$. Thus, the L2-regularization constraint in Eq. 13 comes down to $\| \boldsymbol{R} \|$ or $\| \boldsymbol{F} \|$. In TriFB and GaussFB, $\boldsymbol{F}$ is fixed experimentally, $\| \boldsymbol{F} \|$ is constant, so the regularization constraint makes no difference. The results in the first two columns show the performances of different filter bank methods. Totally independent parameters with only positive constraints get the best result, gaussian and triangular shape constraints follow closely. Triangular shape constraint performs a little better than gaussian constraint. Fixed-bandwidth parameters with positive constraint make no obvious improvement in contrast with traditional TriFB and GaussFB.

Table 3 Reconstruction SDR with/without regularization in decibel

Method	M = 32, R = T	M = 64, R = T	M = 64, R = F
TriFB	8.45	13.01	12.92
GaussFB	8.12	12.44	12.44
TriFB-DN	10.32	14.69	13.28
GaussFB-DN	9.55	12.92	12.22
BandPosFB-DN	8.57	12.84	12.15
PosFB-DN	14.44	18.24	17.21

M represents the number of filters, *R* represents the regularization option, *T* represents true, and *F* represents false

A direct perspective of the six types of filter banks can be seen in Fig. 5. Comparing with TriFB in Fig. 5a, the filter banks of TriFB-DN in Fig. 5c show great difference along the Mel axis. The frequency centers and bandwidths in TriFB-DN distribute relatively regular at low frequencies, but out of order at high frequencies. Comparing with GaussFB in Fig. 5b, the bandwidths of GaussFB-DN in Fig. 5d are less overlapped between neighboring filters. The filter banks of BandPosFB-DN come to be multimodal in the fixed bandwidth. The results in Table 3 show that the frequency center and bandwidth are more important than the shape in music reconstruction tasks. As we have discussed in Section 3.4, the reconstruction tasks usually can be seen as a process of solving overdetermined linear equations, which means that the more parameters the better. Result for PosFB-DN demonstrates this assumption, PosFB-DN has much more parameters than other methods, thus get a much better reconstruction result.

Finally, in this experiment, in order to compare the learned frequency centers and traditional auditory scales, we have shown several frequency center plots in Fig. 6. In Fig. 6a, frequency centers learned in the audio separation task on MIR-1K dataset are compared with the Mel scale. We have also compared frequency centers learned in the audio classification task on both DCASE2016 and LITIS ROUEN datasets with the Mel scale in Fig. 6b, c. For DCASE2016 dataset as shown in Fig. 6b, learned frequency centers coincide well with the Mel scale. The frequency centers almost keep the initial value; this may be due to the lack of data. In Fig. 6a, we can see that the changes of frequency centers can only be observed in high-frequency regions, which means that the learned frequency centers tend to give a different representation of high-frequency components in audio separation tasks. This result is consistent with our experiments in Section 5.4. However, the frequency centers in Fig. 6c change only in relatively low-frequency regions. This observation shows the difference between separation and classification tasks.

5.4 Audio source separation

In this experiment, we investigate the application of discriminative frequency filter banks in audio source

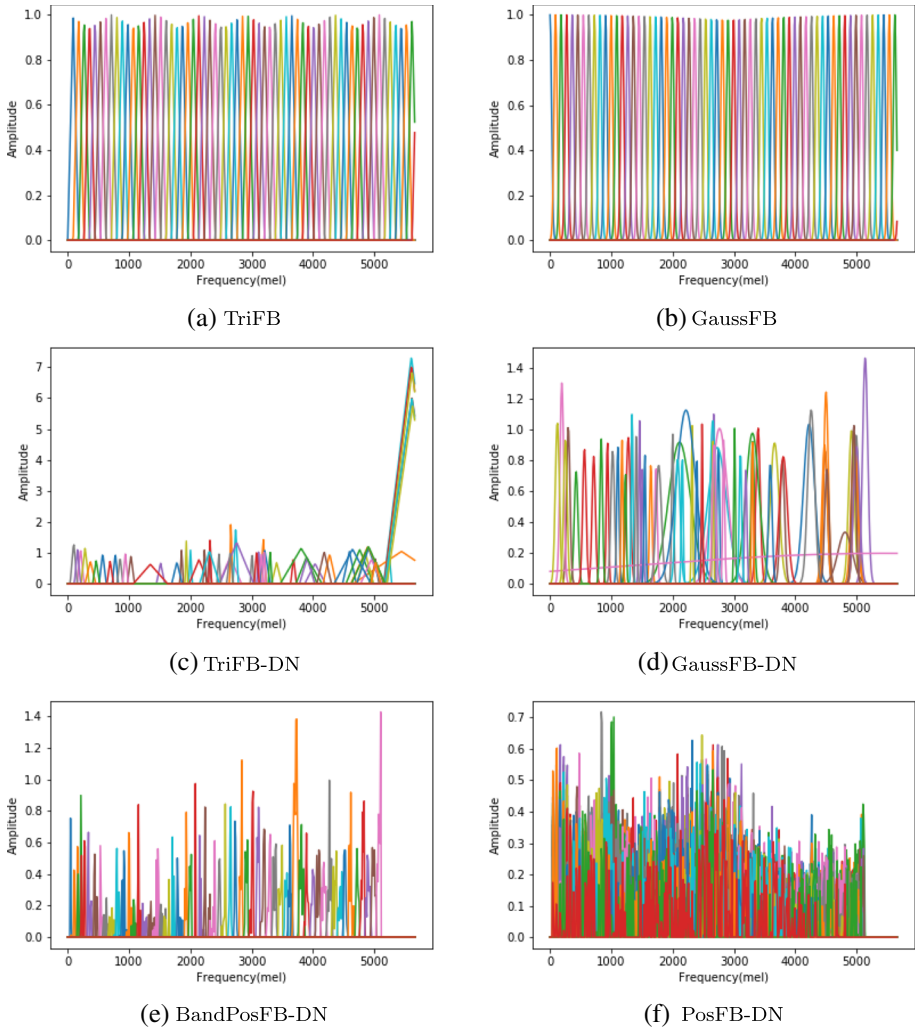


Fig. 5 Shape of different filter banks. **a**, **b** The traditional fixed-parameter filter banks. **c**–**f** The discriminative frequency filter banks we proposed and learned in the simple music spectrum reconstruction task

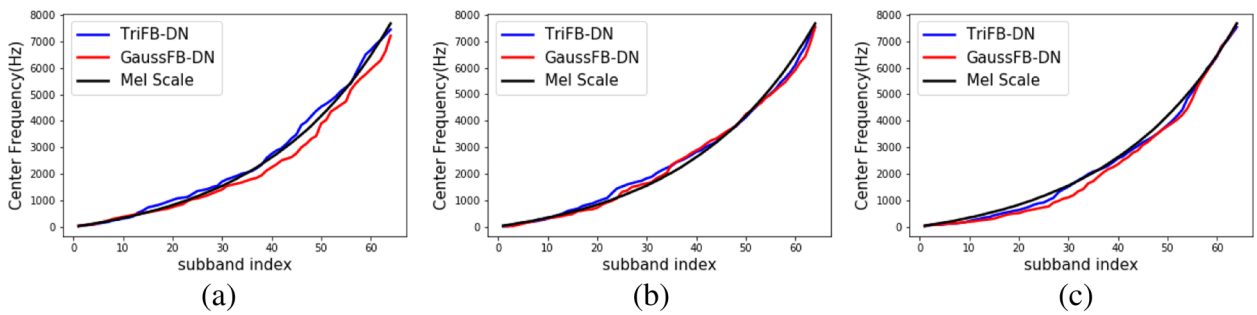


Fig. 6 Comparison of frequency centers learned from network with the Mel scale. **a** Audio separation task on MIR-1K dataset. **b** Audio classification task on DCASE2016 dataset. **c** Audio classification task on LITIS ROUEN dataset

Table 4 Reconstruction SDR of audio source separation in decibel. M/V represents the energy ratio between music and voice

M/V	0.1	1	10
TriFB	4.47	8.30	12.01
GaussFB	4.85	8.39	12.22
TriFB-DN	5.19	8.51	12.92
GaussFB-DN	5.13	8.45	13.01
BandPosFB-DN	5.33	8.39	12.84
PosFB-DN	5.70	9.14	16.99

separation tasks using the MIR-1K dataset. We attempt the music separation from a vocal and music mixture using Fig. 4a.

Table 4 shows the reconstruction SDR in the music separation task. In order to achieve a fair comparison between different filter bank methods, we mix the vocal and music tracks under various conditions, where the energy ratio between music and voice takes 0.1, 1, and 10 respectively. The results of discriminative frequency filter banks in Table 4 show consistent performance improvements in comparison with TriFB and GaussFB. As an example, when we use the PosFB-DN method, and the energy ratio between music and voice is 1, the reconstruction SDR is improved by 0.75 dB compared to GaussFB. When the energy ratio is 0.1, which means that the voice is much louder than music, BandPosFB-DN performs better than TriFB-DN and GaussFB-DN, because the relatively independent parameters can limit the voice amplitude effectively. However, when music is louder, the flexible frequency center and bandwidth in TriFB-DN and GaussFB-DN give better separation results than BandPosFB-DN. In keeping with Table 3, TriFB-DN performs a little better than GaussFB-DN when voice is louder, but the advantage is much smaller than the results in Table 3.

Figure 7 shows the clean music spectrum (**a**), mixed spectrum (**b**), and separated spectrums (**c**–**h**) when the energy ratio is 1. For this example, the separated spectrum can be discussed in the following aspects. In high-frequency regions, TriFB-DN, GaussFB-DN, and PosFB-DN perform much better than the others, which is consistent with Fig. 6a. For these three types of discriminative frequency filter banks, the shape and positive constraints allow the filter banks to learn a more precise representation of high-frequency components. While for fixed-bandwidth methods such as TriFB, GaussFB, and BandPosFB-DN, the representations of high-frequency components are confused. In low-frequency regions, TriFB and GaussFB tend to result in a smooth energy distribution, thus give better performance for spectrum reconstruction.

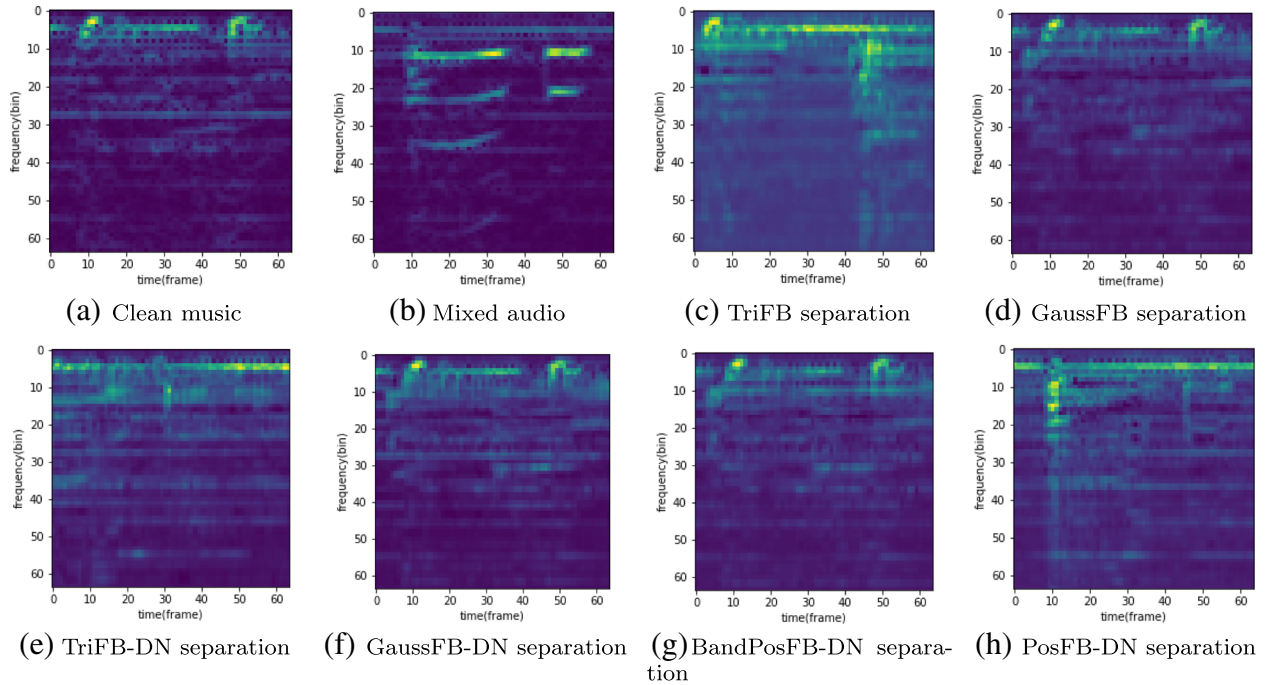


Fig. 7 Reconstructed spectrums of audio source separation tasks. The clean music spectrum in **a** is randomly selected from the dataset. **b** The corresponding music and vocal mixture. **c**–**h** The reconstructed music spectrums from the mixture spectrums using different filter bank methods

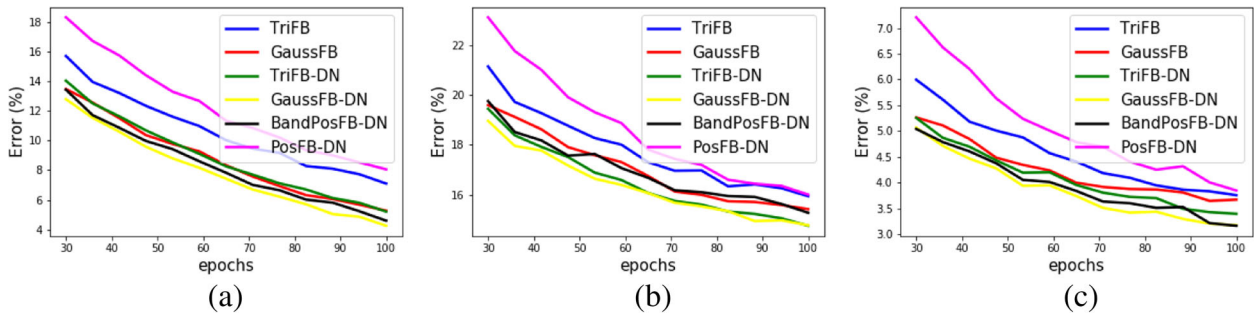


Fig. 8 Training and validation curves on LITIS ROUEN dataset. **a** 1 s clip classification error on training set. **b** 1 s clip classification error on validation set. **c** 30 s audio classification error on validation set

5.5 Audio scene classification (ASC)

When filter banks are used as a feature extractor, the filter banks proposed in this paper can extract more salient features. In this section, we apply the discriminative frequency filter banks to the ASC task. The NN structure is implemented as Fig. 4b. We employ LITIS ROUEN and DCASE2016 datasets in our experiments.

In the data preprocessing step, we first divide a 30-s example into 1-s clips with 50% overlap. Then each clip is processed as Fig. 2 for feature extraction. The classification results of all these clips will be averaged to get an ensemble result for the 30 s example.

Training and validation curves on LITIS ROUEN dataset are shown in Fig. 8. All these methods are stopped after 100 training epochs. In Fig. 8a and b, 1-s clip classification errors on the training and validation set are compared between different methods. We can see that GaussFB-DN performs better than GaussFB along all training epochs, so is TriFB-DN and TriFB. The performance of BandPosFB-DN is almost the same as GaussFB on the validation set. The poor performance of PosFB-DN may be due to the difficulty to learn so many parameters using this dataset. We have also compared 30 s audio classification errors on the validation set in Fig. 8c. The results are almost exactly the same as Fig. 8b, except that BandPosFB-DN becomes one of the best performing methods.

Table 5 is the performance comparison of LITIS ROUEN dataset after 100 training epochs. Evaluation criteria such as accuracy, F-measure and MCC are employed to make the comparison. CNN-Gam [9] is the best performing single-feature model to the best of our knowledge. However, owning to elaborate implementation of the sub-band processing module and classification module in Fig. 2, our baseline model with traditional TriFB and GaussFB perform much better than it. Among these four types of filter banks, shape constrained GaussFB-DN and fixed-bandwidth constrained BandPosFB-DN get the best classification performance, BandPosFB-DN reduces the classification error by relatively 13.9%. While the positive constrained PosFB-DN make no difference in comparison with TriFB and GaussFB.

Training and validation curves on DCASE2016 dataset are shown in Fig. 9. After 100 training epochs, all these methods encounter the overfitting problem. This observation is different from Fig. 8. Table 6 is the performance comparison after 100 training epochs. In order to achieve a fair comparison, we use the same NN structure on both DCASE2016 and LITIS ROUEN datasets, including the hyper-parameters. In keeping with the results in Table 5, TriFB-DN, GaussFB-DN, and BandPosFB-DN get better classification performances as well. The performance of PosFB-DN gets much worse. In comparison with reconstruction related tasks, classification tasks have fewer output dimensions, so when parameters are not constrained by specific shapes, the number of parameters is too large to converge to a stable and smooth classification model.

Table 5 Performance comparison on LITIS ROUEN dataset

Method	Accuracy	F-measure	MCC	Error
TriFB	96.24	96.19	96.01	3.76
GaussFB	96.33	96.44	96.11	3.67
TriFB-DN	96.61	96.50	96.39	3.39
GaussFB-DN	96.83	96.71	96.63	3.17
BandPosFB-DN	96.84	96.71	96.64	3.16
PosFB-DN	96.15	96.04	95.91	3.85
CNN-Gam [9]	95.8	95.8	–	4.2
CNN-MFCC [9]	94.0	93.7	–	6.0
CNN-Log [9]	95.1	95.0	–	4.9
RNN-Gam [8]	96.4	96.6	–	3.6
RNN-MFCC [8]	95.4	95.8	–	4.6
RNN-Log [8]	95.9	96.2	–	4.1

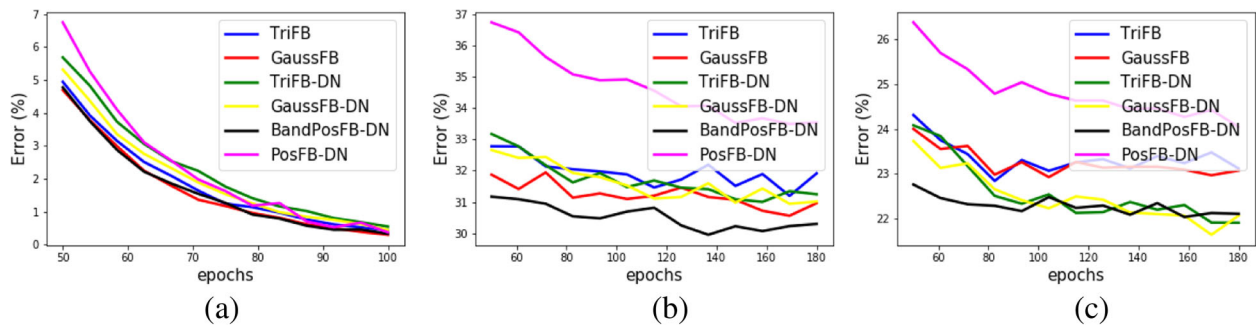


Fig. 9 Training and validation curves on DCASE2016 dataset. **a** 1-s clip classification error on training set. **b** 1-s clip classification error on validation set. **c** 30 s audio classification error on validation set

We also investigate the classification result when we use less than 30 s audios. Figure 10 is the classification error on the two datasets when audios extend from 1 s to 30 s. With long audios, we expect to extract more information by accumulating more statistics. As a result, for DCASE2016 dataset, GaussFB-DN can obtain an accuracy of 75.2% at 15 s, which is better than TriFB at 30 s.

6 Conclusion

The construction of discriminative frequency filter banks that can be learned by neural networks has been presented in this paper. The filter banks are implemented on FFT-based spectrums and can be constrained under different conditions to express different aspects of physical meanings. For shape-related constraints, a piecewise differentiable triangular shape is approximated using several differentiable basic functions. For positive constraints, ReLU and square constraints are proposed to fulfill the demand for the probability distribution of weights. Then, a spectrum reconstruction method from incomplete filter bank coefficients is implemented using neural networks. A well-designed regularization strategy is also studied to guarantee the filter banks to be BIBO-stable. Overall, this paper provides a practical and complete framework to learn discriminative frequency filter banks for different tasks.

The discriminative frequency filter banks proposed in this paper are compared with traditional fixed-parameter filter banks using several experiments. The results show performance improvements for both music reconstruction and audio classification tasks. However, not all variants of discriminative frequency filter banks are suitable for all situations. In our experiments, positive constrained filter banks perform best on music reconstruction tasks, and shape constrained filter banks obtain the best results on ASC tasks.

Discriminative frequency filter banks on FFT-based spectrums have the ability to get adaptive resolution on the frequency domain. To achieve adaptive resolution on the time domain, the future work will include introducing temporal information into filter banks, for example, the filter banks may span several frames. We will also perform cross-domain experiments to learn filter banks on one dataset and use it for classification tasks on another dataset to see if the generalized filter banks can be learned as done in [55].

Table 6 Performance comparison on DCASE2016 dataset

Method	Accuracy	F-measure	MCC	Error
TriFB	76.88	76.08	75.55	23.12
GaussFB	77.31	76.55	76.01	22.69
TriFB-DN	78.09	77.39	76.83	21.91
GaussFB-DN	78.36	77.44	77.10	21.64
BandPosFB-DN	77.89	77.07	76.63	22.11
PosFB-DN	75.96	74.93	74.62	24.04
MFCC-GMM [53]	72.5	–	–	27.5
Mel-DNN [56]	76.4	–	–	23.6
Mel-CNN [57]	76.0	–	–	24.0

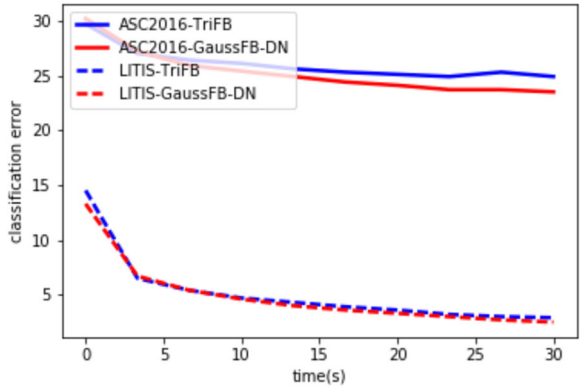


Fig. 10 Early classification error

Acknowledgments
Not applicable.

Funding
This work was partly funded by National Natural Science Foundation of China (Grant No. 61571266).

Availability of data and materials
The datasets analysed during the current study are available in the MIR-1K repository, http://sites.google.com/site/unvoicedsoundseparation/mir-1k/, LITIS ROUEN repository, https://sites.google.com/site/alainrakotomamonjy/home/audio-scene, and DCASE2016 repository, http://www.cs.tut.fi/sgn/arg/dcase2016/download.

Authors' contributions
TZ designed the core methodology of the study, carried out the implementation and experiments, and drafted the manuscript. JW participated in the study and helped to draft the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 1 April 2018 Accepted: 29 November 2018
Published online: 03 January 2019

References

1. J. Allen, Short term spectral analysis, synthesis, and modification by discrete fourier transform. IEEE Trans. Acoust. Speech Signal Process. **25**(3), 235–238 (1977)
2. I. Daubechies, The wavelet transform, time-frequency localization and signal analysis. IEEE Trans. Inf. Theory. **36**(5), 961–1005 (1990)
3. S. Akkarakaran, P. Vaidyanathan, in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference On, vol 3*. New results and open problems on nonuniform filter-banks (IEEE, Piscataway, 1999), pp. 1501–1504
4. A. Biem, S. Katagiri, B.-H. Juang, in *Neural Networks for Processing [1993] III. Proceedings of the 1993 IEEE-SP Workshop*. Discriminative feature extraction for speech recognition (IEEE, Piscataway, 1993), pp. 392–401
5. Á. de la Torre, A. M. Peinado, A. J. Rubio, V. E. Sánchez, J. E. Diaz, An application of minimum classification error to feature space transformations for speech recognition. Speech Comm. **20**(3-4), 273–290 (1996)
6. N. Chen, Y. Qian, H. Dinkel, B. Chen, K. Yu, in *INTERSPEECH*. Robust deep feature for spoofing detection—the sjtu system for asvspoof 2015 challenge (International Speech Communication Association (ISCA), Dresden, 2015), pp. 2097–2101
7. Y. Qian, N. Chen, K. Yu, Deep features for automatic spoofing detection. Speech Comm. **85**, 43–52 (2016)
8. H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, A. Mertins, Audio scene classification with deep recurrent neural networks. arXiv preprint arXiv:1703.04770 (2017)
9. H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, A. Mertins, Improved audio scene classification based on label-tree embeddings and convolutional neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(6), 1278–1290 (2017)
10. B. Gao, W. Woo, L. Khor, Cochleagram-based audio pattern separation using two-dimensional non-negative matrix factorization with automatic sparsity adaptation. J. Acoust. Soc. Am. **135**(3), 1171–1185 (2014)
11. J. Le Roux, E. Vincent, Consistent wiener filtering for audio source separation. IEEE Signal Process Lett. **20**(3), 217–220 (2013)
12. P. Majdak, P. Balazs, W. Kreuzer, M. Dörfler, in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference On*. A time-frequency method for increasing the signal-to-noise ratio in system identification with exponential sweeps (IEEE, Piscataway, 2011), pp. 3812–3815
13. D. L. Donoho, De-noising by soft-thresholding. IEEE Trans. Inf. Theory. **41**(3), 613–627 (1995)
14. R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification*. (Wiley, New York, 1973)
15. A. Biem, S. Katagiri, in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference On, vol 2*. Feature extraction based on minimum classification error/generalized probabilistic descent method (IEEE, Piscataway, 1993), pp. 275–278
16. A. Biem, S. Katagiri, E. McDermott, B.-H. Juang, An application of discriminative feature extraction to filter-bank-based speech recognition. IEEE Trans. Speech Audio Process. **9**(2), 96–110 (2001)
17. S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoustics Speech Signal Process. **28**(4), 357–366 (1980)
18. V. Hohmann, Frequency analysis and synthesis using a gammatone filterbank. Acta Acustica U. Acustica. **88**(3), 433–442 (2002)
19. T. Irino, R. D. Patterson, A dynamic compressive gammachirp auditory filterbank. IEEE Trans. Audio Speech Lang. Process. **14**(6), 2222–2232 (2006)
20. E. A. Lopez-Poveda, R. Meddis, A human nonlinear cochlear filterbank. J. Acoust. Soc. Am. **110**(6), 3107–3118 (2001)
21. E. Zwicker, E. Terhardt, Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. J. Acoust. Soc. Am. **68**(5), 1523–1525 (1980)
22. B. R. Glasberg, B. C. Moore, Derivation of auditory filter shapes from notched-noise data. Hear. Res. **47**(1), 103–138 (1990)
23. R. P. Lippmann, Speech recognition by machines and humans. Speech Commun. **22**(1), 1–15 (1997)
24. B. Mak, Y.-C. Tam, R. Hsiao, in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference On, vol 2*. Discriminative training of auditory filters of different shapes for robust speech recognition (IEEE, Piscataway, 2003), p. 45
25. T. Kobayashi, J. Ye, in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference On*. Discriminatively learned filter bank for acoustic features (IEEE, Piscataway, 2016), pp. 649–653
26. H. B. Sailor, H. A. Patil, in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference On*. Filterbank learning using convolutional restricted boltzmann machine for speech recognition (IEEE, Piscataway, 2016), pp. 5895–5899
27. T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, O. Vinyals, in *INTERSPEECH*. Learning the speech front-end with raw waveform cldnns (International Speech Communication Association (ISCA), Dresden, 2015), pp. 2097–2101
28. T. N. Sainath, B. Kingsbury, A.-R. Mohamed, B. Ramabhadran, in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop On*. Learning filter banks within a deep neural network framework (IEEE, Piscataway, 2013), pp. 297–302
29. H. Yu, Z.-H. Tan, Y. Zhang, Z. Ma, J. Guo, Dnn filter bank cepstral coefficients for spoofing detection. IEEE Access. **5**, 4779–4787 (2017)
30. H. Seki, K. Yamamoto, S. Nakagawa, in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference On*. A deep neural network integrated with filterbank learning for speech recognition (IEEE, Piscataway, 2017), pp. 5480–5484
31. S. Strahl, A. Mertins, Analysis and design of gammatone signal models. J. Acoust. Soc. Am. **126**(5), 2379–2389 (2009)
32. B. Milner, X. Shao, Prediction of fundamental frequency and voicing from mel-frequency cepstral coefficients for unconstrained speech reconstruction. IEEE Trans. Audio Speech Lang. Process. **15**(1), 24–33 (2007)
33. D. Chazan, R. Hoory, G. Cohen, M. Zibulski, in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference On, vol 3*. Speech reconstruction from mel frequency cepstral coefficients and pitch frequency (IEEE, Piscataway, 2000), pp. 1299–1302
34. B. Milner, X. Shao, in *ICSLP*. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model (International Speech Communication Association (ISCA), Denver, 2002), pp. 2421–2424
35. R. F. Lyon, A. G. Katsiamis, E. M. Drakakis, in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium On*. History and future of auditory filter models (IEEE, Piscataway, 2010), pp. 3809–3812

36. T. Necciari, N. Holighaus, P. Balazs, Z. Prusa, A perceptually motivated filter bank with perfect reconstruction for audio signal processing. arXiv preprint arXiv:1601.06652 (2016)
37. W. A. Yost, R. R. Fay, *Auditory perception of sound sources, vol 29*. (Springer Science & Business Media, Berlin, 2007)
38. S. Rosen, R. J. Baker, A. Darling, Auditory filter nonlinearity at 2 khz in normal hearing listeners. J. Acoust. Soc. Am. **103**(5), 2539–2550 (1998)
39. R. Patterson, I. Nimmo-Smith, J. Holdsworth, P. Rice, in *a Meeting of the IOC Speech Group on Auditory Modelling at RSRE, vol 2*. An efficient auditory filterbank based on the gammatone function, (1987)
40. S. S. Stevens, J. Volkmann, The relation of pitch to frequency: A revised scale. Am. J. Psychol. **53**(3), 329–353 (1940)
41. S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al., The htk book. Camb. Univ. Eng. Dept. **3**, 175 (2002)
42. P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, G. Velasco, Theory, implementation and applications of nonstationary gabor frames. J. Comput. Appl. Math. **236**(6), 1481–1496 (2011)
43. P. Frederic, F. Lad, Two moments of the logitnormal distribution. Commun. Stat.–Simul. Comput.®. **37**(7), 1263–1269 (2008)
44. M. James, The generalised inverse. Math. Gaz. **62**(420), 109–114 (1978)
45. A. Ben-Israel, T. N. Greville, *Generalized Inverses: Theory and Applications, vol 15*. (Springer Science & Business Media, Berlin, 2003)
46. R. Hagen, S. Roch, B. Silbermann, *C*-algebras and Numerical Analysis*. (CRC Press, Boca Raton, 2000)
47. P. Varaiya, R. Liu, Bounded-input bounded-output stability of nonlinear time-varying differential systems. SIAM J. Control. **4**(4), 698–704 (1966)
48. X. Zhao, Y. Shao, D. Wang, Casa-based robust speaker identification. IEEE Trans. Audio Speech Lang. Process. **20**(5), 1608–1616 (2012)
49. Y. Kim, Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
50. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(Aug), 2493–2537 (2011)
51. H. Chao-Ling, J. Shing, R. Jang, MIR Database (2010). http://sites.google.com/site/unvoicedsoundseparation/mir-1k/. Accessed 8 Dec 2018
52. A. Rakotomamonjy, G. Gasso, Histogram of gradients of time-frequency representations for audio scene classification. IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP). **23**(1), 142–153 (2015)
53. A. Mesaros, T. Heittola, T. Virtanen, in *Signal Processing Conference (EUSIPCO), 2016 24th European*. Tut database for acoustic scene classification and sound event detection (IEEE, Piscataway, 2016), pp. 1128–1132
54. D. Kingma, J. Ba, Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
55. H. B. Sailor, H. A. Patil, Novel unsupervised auditory filterbank learning using convolutional rbm for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(12), 2341–2353 (2016)
56. Q. Kong, I. Sobieraj, W. Wang, M. Plumbley, Deep neural network baseline for dcase challenge 2016. Tampere University of Technology, Department of Signal Processing. Proceedings of DCASE 2016 (2016)
57. D. Battaglino, L. Lepauloux, N. Evans, F. Mougins, F. Biot, Acoustic scene classification using convolutional neural networks. DCASE2016 Challenge, Tech. Rep. Tampere University of Technology, Department of Signal Processing (2016)