Robust Sound Event Classification using Deep Neural Networks

Ian McLoughlin, *Senior Member, IEEE,* Haomin Zhang, Zhipeng Xie, Yan Song, and Wei Xiao

***Abstract*—The automatic recognition of sound events by computers is an important aspect of emerging applications such as automated surveillance, machine hearing and auditory scene understanding. Recent advances in machine learning, as well as in computational models of the human auditory system, have contributed to advances in this increasingly popular research field. Robust sound event classification, the ability to recognise sounds under real-world noisy conditions, is an especially challenging task. Classification methods translated from the speech recognition domain, using features such as mel-frequency cepstral coefficients, have been shown to perform reasonably well for the sound event classification task, although spectrogram-based or auditory image analysis techniques reportedly achieve superior performance in noise. This paper outlines a sound event classification framework that compares auditory image front end features with spectrogram image-based front end features, using support vector machine and deep neural network classifiers. Performance is evaluated on a standard robust classification task in different levels of corrupting noise, and with several system enhancements, and shown to compare very well with current state-of-the-art classification techniques.**

***Index Terms*—Machine hearing, auditory event detection**

I. Introduction

RICHARD F. Lyon, in an IEEE Signal Processing Magazine article of September 2010 [1], outlined the broad research field of machine hearing, in particular advocating a bio-mimetic approach in which machines attempt to model the human hearing apparatus. In fact, he and his group have since published a significant amount of research using this approach [2]–[5]. In general, the published systems perform ear-like front-end auditory analysis, feature extraction, feature size reduction, followed by application of machine learning techniques. The stated application is for the search or query of very large scale audio databases, and thus the efficient representation of auditory features is of great importance in their work. This has led to the use of high performance sparse feature coding techniques allied to suitable machine learning methods. One of the defining features of these methods is a front-end ear-like audio analysis generating features extracted from a stabilised auditory image (SAI) [6].

By contrast to the task of content-based audio retrieval, the current paper is concerned with sound event classification. This is not a retrieval task, but rather one of classification, detection or generalisation. The requirement is that a trained system, when presented with an unknown sound, is capable of correctly identifying the class of that sound. Furthermore, that the techniques should be robust to interfering acoustic noise.

In fact, many researchers have worked on sound event classification over the years, using a myriad of techniques and features. These range from parametric signal processing-based approaches [7]–[9] to automatic speech recognition (ASR) inspired methods [10] which often make use of mel-frequency cepstral coefficients (MFCCs) [11] and similar features. One promising new approach uses time-frequency domain spectrogram image features (SIF), introduced by Jonathan Dennis et. al. [12]–[15]. As with Lyon et. al., Dennis et. al. use biologically inspired front-end processing, novel feature extraction techniques, allied with various back-end classifiers and associated machine learning techniques. Unlike the former approach, the systems introduced by Dennis et. al. are sound event detectors or classifiers. They have been evaluated under real-world conditions including severe levels of degrading acoustic background noise.

In this paper, both SAI [6] and SIF will be evaluated for standard robust sound event classification tasks. The former could loosely be described as sound event classification inspired by the retrieval approaches of Lyon et. al. [3], which we call the Google-SAI system. The latter SIF methods are closer to the work of Dennis [15]. In each case, the front end analysis and feature extraction operations are followed by back-end machine learning methods. We will primarily compare the use of support vector machines (SVM) and deep neural network (DNN) classifiers.

To the best of the authors' knowledge, this paper contributes the first DNN classifier for the time-frequency features of SAI and SIF for sound event detection and classification. It is also the first to apply the Google-SAI feature extraction techniques of Lyon et. al. [3] for sound event detection and classification as opposed to retrieval – and this is evaluated with both SVM and DNN back-end classifiers, using several feature arrangements and scoring refinements. Results will show that the novel method developed from this study, using a DNN classifier with simple de-noising, compares very well to other published techniques on standard classification tasks.

The remainder of this paper is organised as follows. Section II discusses current sound event classification and sound retrieval methods in more detail. Section III details the SAI, SIF features and SVM, DNN classification frameworks which are then evaluated in Section IV. Section V will analyse performance results and explore the effect of changing many

I. McLoughlin, H.-M. Zhang, Z.-P. Xie and Y. Song are with the National Engineering Laboratory of Speech and Language Information Processing, The University of Science and Technology of China, Hefei, PRC.

W.Xiao is with European Research Center, Huawei Technologies Duesseldorf GmbH, Munich, Germany

Manuscript received May 08, 2014; revised May 09, 2014.

system parameters. Section VI will conclude the paper.

II. Current methods

It may be convenient to divide sound event detection methods using two criteria relating to the features and classification algorithms used. Broadly speaking, earlier research in this field tended towards use of relatively simple audio features [16] such as zero crossing rate, frame power, sub-band energy, pitch and so on [8]. Often a very simple heuristic back-end was then used to statically combine the features to reach a decision. To improve on static decision-making, fuzzy classifiers were later introduced. In fact, for a small number of classes, such techniques can be efficient and perform reasonably well. Therefore, most papers using such methods were restricted to evaluating performance with less than 10 sound classes.

More recently, machine learning techniques have increasingly been adopted to classify combined features in ways beyond simple logical heuristics, enabling more sound classes – typically up to 20 – as well as yielding higher performance [17]. With the ability to learn non-obvious relationships between input features and output class, the adoption of machine learning techniques also naturally encouraged the use of more complex input features [18] including MFCCs [7] and perceptual linear prediction (PLP) coefficients [17]. Classification (and recall) techniques for such systems have, in recent years, most commonly involved support vector machines (SVM) [17], [19], Gaussian mixture models (GMMs) [20] or multilayer perceptrons (MLP) [21]. Research in machine hearing is often driven by the success of techniques used for ASR, hence a number of published techniques which make use of MFCC features [11], hidden Markov model toolkit (HTK) and associated back-end classifiers [15].

Meanwhile, another research thread was being built around biologically-inspired models of the human auditory system. Although PLPs, as well as MFCCs, involve non-linear processing designed to model the frequency response of the human ear, the new research was motivated to additionally account for time-domain and strobed temporal integration effects, and perhaps to better model the auditory signal as received by the human brain. For example, Patterson et. al. [22] released the auditory image model (AIM) in 1995, which was used as a basis for the SAI of Walters [6] and for many of the Google-SAI systems developed by Lyon et. al. [2]–[5]. AIM and SAI use was encouraged with the Matlab source code being freely available from the University of Cambridge [23], and a C++ language version of SAI available from Google [24].

While SVM classifiers have a long history in this, and similar research domains, DNN systems [25], [26] are much newer, but have achieved impressive performance for speech-related classification tasks [27]–[29] as well as for acoustic information retrieval [30]. A reasonable hypothesis is that they will similarly perform well for non-speech sound event detection and classification. This paper will thus explore the hypothesis further.

A. SAI with PAMIR

A separate stabilised auditory image (SAI) sequence is formed for each distinct sound recording, and is intended to model the effect of the sound on the physiology of the ear. Initially, a sound segment is analysed to form an AIM [22] representation. The processing steps for this begin with pre-cochear processing [31] which models many of the physiological effects outlined in [32] relating to the outer and middle ear. This is followed by basilar membrane modelling (i.e. cochlear modelling [16]) and neural activity pattern processing, for which several alternative models are available to translate hair cell movement into nerve impulses. For the results presented in this paper, a gammatone model is applied [33], followed by half-wave rectification.

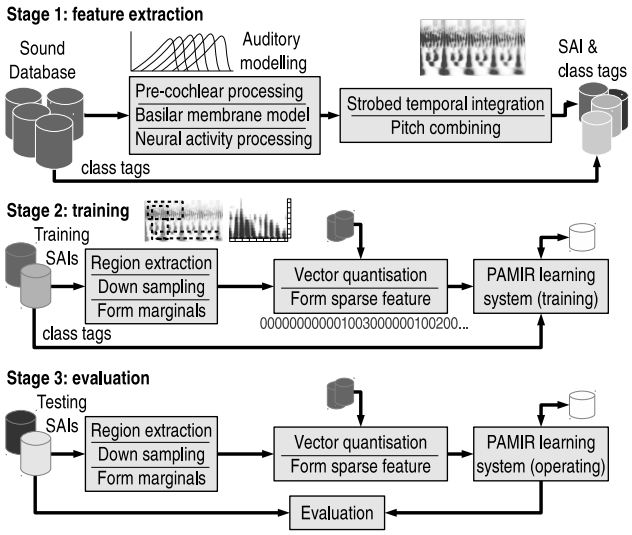


Fig. 1. Block diagram of PAMIR-based recall system using front-end SAI analysis.

The next step to form an SAI is to perform strobed temporal integration on the AIM output, adding another dimension representing delay to the AIM data [6]. In effect, this is modelling the emphasised response of repetitive audio triggers such as fine-grained sound intervals – sounds that may otherwise be inadequately represented by mean responses. The default strobe-finding process performs across a 35 ms long search space [23]. The resulting SAI is a sequence of two dimensional frames. Each frame has dimensions of frequency and delay lag, and is analogous to a short-time spectrogram. Although delay lag, window size and resolution used for the analysis may be configured, the system described in this paper results in SAI frames of dimension 78×561, with one frame produced every 35 ms and each 'pixel' being represented by an 8-bit intensity.

One of the innovations made by Walters [6] was in discovering that each SAI frame can be reasonably well represented by its marginals, meaning a concatenated vector comprising the mean-of-rows and mean-of-columns. We will later present, in Section IV-B, experimental results from using Minkowsky summation instead of a simple mean, as well as the effect of concatenating variance information to the representative vector. Additionally, we have investigated histogram equalisation, vector normalisation and subtractive de-noising of this representative vector. The large dimensionality reduction implicit in reducing each SAI frame to a vector of marginals is

important when considering large scale audio retrieval, but is less important to small and medium scale audio classification applications. A second innovation from Walters is that further dimensionality reduction is possible by replacing the whole SAI by multi-resolution regions from the image [6]. This is accomplished by dividing the SAI into a series of different sized wide and short rectangles (which have narrow frequency span but wide delay lag span) as well as tall narrow rectangles (which are located around a narrow region of delay lag, but have a wide frequency span). In practice, the various rectangles are down-sampled to match the size of the smallest one before each is represented by its marginals as mentioned above. The location, number and size of rectangles is configurable. 49 rectangles of size 16×32 were used in [2], with results from many other configuration reported in [6]. The authors of the current paper have also investigated the effect of rectangle size, number of rectangles and resolution, discussed in Sections III-A and IV-B.

To further improve efficiency, the large-scale systems [4] perform vector quantisation (VQ) [16] or matching pursuit [34] on each rectangle, and represent the output as a sparse code. In this paper, both VQ and non-VQ results will be presented. Where VQ is used, performance tends to increase with codebook size, saturating at a size of around 512. A size of 256 was used in [4].

The final aspect of the Google-SAI systems is the machine learning algorithm. All published systems appear to make use of the passive aggressive model for image retrieval (PAMIR) [35]. The stated motivation is the prior availability of PAMIR, coupled with its demonstrated good performance at the desired recall task (in fact, PAMIR had performed well for MFCC features in [11], although tests by the authors of the current paper – not reported here – on a large *Freesound* database suggest that k-NN is able to outperform PAMIR when using SAI features).

The basic structure discussed here will be expanded in Section III to form an audio classification system which will then be evaluated against similar methods. In this paper, although we explore many permutations of the SAI front end, the sharp difference to prior work is the use of SVM and, in particular, DNN classifiers (as opposed to the shallow sparse coding technique with PAMIR used in Google-SAI).

B. Dennis SIF with SVM

The SIF feature used by Dennis et. al. [14] begins with either a linear or log scaled spectrogram which is then normalised before being represented using a pseudo colourmap (i.e. three-way thresholding is applied). The colourmap image is then decomposed into orthogonal primary colour components, each of which emphasise a particular intensity region of the spectrogram image. The three primary images are then divided into regions – in [12], 9×9 blocks are used, each of which are represented in turn by their second and third central moments. The feature vector is thus 9×9×3×2=486 dimensional, which is classified using one-against-one linear SVM [12], [15]. The current paper adopts the same standard classifier evaluation task as Dennis [15] (including background noise conditions), thus allowing a direct comparison of performance among many different methods. However the major differences are that (i) we will implement an SIF feature directly from a downsampled spectrogram, without division into blocks or representation by central moments and (ii) we will apply deep learning techniques to the classification problem. One significant point is that the new SIF feature used with DNN incorporate additional temporal context information that appears to be advantageous for classification in noise.

C. Performance criteria

For machine hearing, a number of performance criteria are possible depending upon the target application. For the PAMIR-based systems [5], recall performance is computed rather than classification per. se. Furthermore, the systems are evaluated by ranking the output candidate tags and identifying the proportion of top-k results achieved – where a correct tag lying within the highest k scoring outputs is considered to be a correct result. Performance curves may be plotted for a range of k that is typically between 1 and 20 [11].

Having a classification target, the current paper will adopt the evaluation method, experimental dataset, and scoring methods of Dennis et. al. [14]. These will allow a direct comparison between several techniques from different authors which have been evaluated under the same conditions (this will be presented in Section V-A). The use of a defined dataset, from the Real Word Computing Partnership (RWCP) [36] allows for reproducible comparisons. The dataset and classification task is described fully in Section IV.

III. The Proposed Framework

In this paper, we will first investigate the use of Google-style SAI features with a back-end SVM classifier, and use this baseline to evaluate the effect of several modifications to the feature extraction and representation process. Next, the SVM classifier in the best performing system is replaced with a DNN back-end. Finally, the DNN classification performance is evaluated with a number of different feature representations that are derived from an SIF. The performance evaluation will be described in Section IV. First, the following subsections will describe the basic building blocks used, namely the two types of features (SAI and SIF) and two types of classifier (SVM and DNN).

A. SAI features

The SAI features used in this work are derived as defined in [4] using AIM-C [24], extracted over 35 ms windows to yield a real-valued matrix for each analysis frame. By default, the non-linear frequency resolution is 78 and the time lag resolution is 561. In [4], multiple rectangular regions were extracted from each SAI according to a 'start-small and then double' heuristic outlined in [6], starting with an initial window of size $D = 16$ by $B = 32$. In total $R = 49$ rectangular regions were extracted and then downsampled to match the $D \times B$ size of the initial window, and the marginals of each region computed as discussed in Section II-A to yield a representative feature vector of size $D + B$, i.e. $16 + 32 = 48$ per SAI frame.

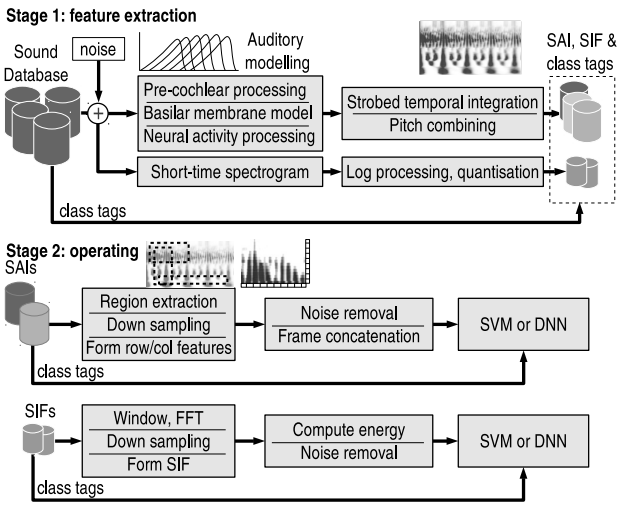


Fig. 2. Block diagram of audio event classification system using front-end SAI and SIF analysis.

For the Google-SAI PAMIR-based recall systems [4], the features from each rectangular region are vector quantized using a separate codebook of size 256 for each feature. A sparse representation is then constructed by concatenating the 49 VQ output vectors, forming a super-vector of length $12,544$. In practice, this has a sparsity of 99.6%. Since there are multiple analysis frames per sound file, and sound files have different length, the sparse vectors from all analysis frames within the sound file are then summed to yield a slightly less-sparse representative feature vector.

A block diagram of the original PAMIR system was shown in Fig. 1. When operated and evaluated as a classifier, the system has three distinct phases of operation, which are namely the extraction of features from all sounds which are partitioned into training and testing datasets. Next, the former sounds are used to train PAMIR. Finally, the remaining sounds are used for evaluation. Cross-validation ensures that the particular allocation of files into training and testing regions does not impact results score. The standard training process is explained in [6].

Using the same feature vectors with an SVM classifier, we noted that performance was significantly better when using real-valued marginals from all rectangular regions to represent a single analysis window, instead of performing VQ and presenting a sparse vector. Thus, the classifier was trained with data from many analysis windows for each sound file, having a feature vector dimension of $R(B+D)$, with no pre-processing of sound files. A block diagram of the SAI extraction and feature vector formation can be seen in Fig. 2.

Neither the original PAMIR nor the proposed SVM classifier feature vectors appear able to adequately represent long-duration time-domain variations within a sound file (i.e. longer than the 35 ms delay lag analysis). Thus a modification of the SVM classifier feature vector will be proposed and evaluated later. This is to cluster and concatenate consecutive features into a longer vector which includes time-domain context, T. Section IV-B3 will briefly explore this and demonstrate a performance gain from the additional context. However the resulting vector becomes very large, with dimension $T.R(B+D)$. Fortunately, further experimentation reveals that the benefit of additional temporal information is greater than the benefit gained by using multiple rectangular windows to represent the SAI. Thus, a more efficient solution which provides very good performance is to simply downsample the entire SAI to size $D \times B$, represent this using marginals (i.e. $D+B$) and then concatenate with neighbouring marginals to incorporate context, giving a size of $T(B+D)$ – which is even more efficient than the original feature vector since $T << R$.

In experiments reported in Section V, the SVM classifier will be evaluated for static input features (i.e. without context, size $R(B+D)$, as well as for features including the context from T concatenated neighbouring frames (i.e. dimension $T(B+D)$). Note that the spectrogram features described in the next subsection will also be formed into lower dimension feature vectors in a similar way.

The DNN classifier [37], when evaluated using SAI features, simply replaces the SVM classifier, with the same input features. This will be discussed further in section III-D.

The SAI features can be visualised in Fig. 3, showing images for two different sounds, for two different background noises, and two images of sound plus noise at 0dB SNR.

B. SIF features

The initial spectrogram comprises a stack of fast Fourier transform (FFT) magnitude spectra. Given a length N sound vector s, a spectral line f is obtained from highly overlapped and windowed frames of length w_s sample. For current frame F, spectral line f_F is thus obtained as follows:

$$s_F(n) = s(F.\delta + n).w(n) \qquad for\ n = 0 \ldots (w_s - 1) \quad (1)$$

$$f_F(k) = |\sum_{n=0}^{w_s-1} s_F(n) e^{\frac{-j2\pi nk}{w_s}}| \qquad for\ k = 1 \ldots (w_s/2 - 1) \quad (2)$$

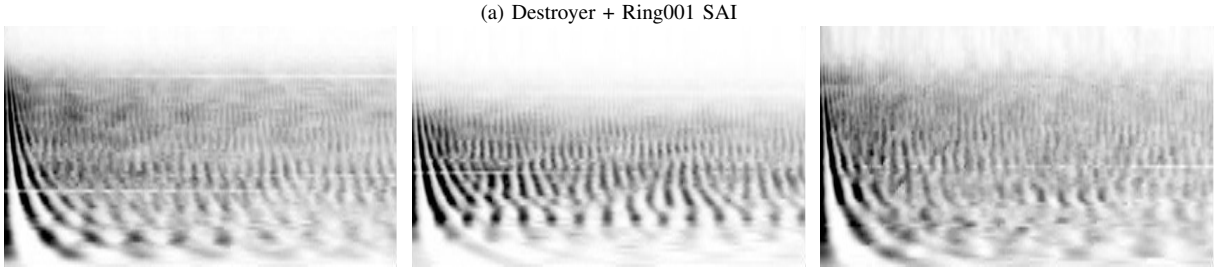
where δ is the sample advance between analysis frames and $w(n)$ defines an N-point Hamming window. Down sampling is performed to match the B bin frequency resolution of the SAI-based method by averaging over $B' = \lfloor w_s/2B \rfloor$ samples. The resulting spectra are stacked to form an overlapped spectrogram ($\mathcal{S}$).

$$\mathcal{S}(l,m) = \frac{1}{B'} \cdot \sum_{n=B'.l}^{B'.(l+1)} f_{F-m}(n) \qquad for\ l = 0 \ldots B/\delta \quad (3)$$

In practice, the spectrogram $\mathcal{S}$ contains a history of up to D consecutive spectral lines (i.e. $m = 0 \ldots D-1$) which are concatenated to populate a $(B.D+1)$ dimension feature vector V which is augmented by a scalar energy metric. Feature vector $\mathbf{v}$ comprises elements $v(i) = \mathcal{S}(\lfloor i/B \rfloor, i - B.\lfloor i/B \rfloor))$ for $i = 0 \ldots (B.D-1)$ with the energy metric defined as,

$$v(B.D) = \sum_{l=0}^{D-1} \sum_{m=0}^{B-1} \mathcal{S}(l,m) \quad (4)$$

(a) Destroyer + Ring001 SAI



(b) Factory Floor 1+Drum004 SAI

Fig. 3. Example stabilised auditory images of, from left to right: the 0dB SNR combined signals, the noise-free named sound from the RWCP database, and the corresponding slice of the named NOISEX-92 background noise. The same sounds are shown using the SIF representation in Fig. 4.

This scalar energy metric is designed to capture information regarding frame energy, on the basis that very low energy frames are likely to be less discriminative to sound classification than higher energy frames. In fact, our tests reveal that the use of just a single energy metric leads to between 10 and 20% classification performance gain in noisy conditions. This value is also investigated as a scaling for the DNN frame output classification later. The feature vector $\mathbf{v}$, with a dimensionality of only $(B.D+1)$ constitutes the DNN initial layer input, and thus defines its input layer size.

A simple approach to de-noising (DN) is also investigated to mitigate the noise added to the SIF test features (training, by contrast, is always performed using noise-free sounds). Each file in the test data set, corrupted by additive noise, is represented by multiple overlapped analysis frames of downsampled spectrogram information, with each frame generating a length B spectral vector. To perform de-noising, the minimum quantity in each of the B frequency bins is computed across the entire sound file, with each minimum value subsequently subtracted from every spectral vector prior to forming the feature matrix. De-noising proceeds from $\mathcal{S}$ in eqn. (3). The de-noised spectrogram $\mathcal{S}_{dn}$ is thus,

$$\mathcal{S}_{dn}(l,m) = \mathcal{S}(l,m) - \min_{l}(\mathcal{S}(l,m)) \tag{5}$$

for $m = 0 \dots (B-1)$. The initial $B.D$ elements of the final feature vector $\mathbf{v}$, are then formed from $\mathcal{S}_{dn}$, rather than $\mathcal{S}$, however the energy metric $v(B.D)$ is computed from original spectrogram data as usual, as in eqn. (4).

The SIF features (without energy) can be seen in Fig. 4, for two sounds, two background noises and the combination of each. The sounds, noises and combined vectors are identical to those used to produce the SAI images in Fig. 3.

C. SVM classifier

Given a length V input feature vector $\mathbf{v} = [v_1, v_2, ..., v_V]^\top$, with $\mathbf{v} \in R^V$ and corresponding vector of K classes, $\mathbf{y} = [y_1, y_2, ..., y_K]^\top$, with $\mathbf{y} \in \{1, -1\}^K$, SVM with linear kernel, solves the primal optimisation of the normal vector to the hyperplane, w;

$$\min_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{V}\xi_i \tag{6}$$

where $c > 0$ is a regularisation constant and we use slack variables ξ to define an acceptable tolerance;

$$y_i(\mathbf{w}^T\psi(\mathbf{v}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1 \dots V \tag{7}$$

given that $\psi(\mathbf{v}_i)$ maps $\mathbf{v}_i$ into a higher dimensional space. Since $\mathbf{w}$ typically has high dimensionality [38], we usually solve the related problem,

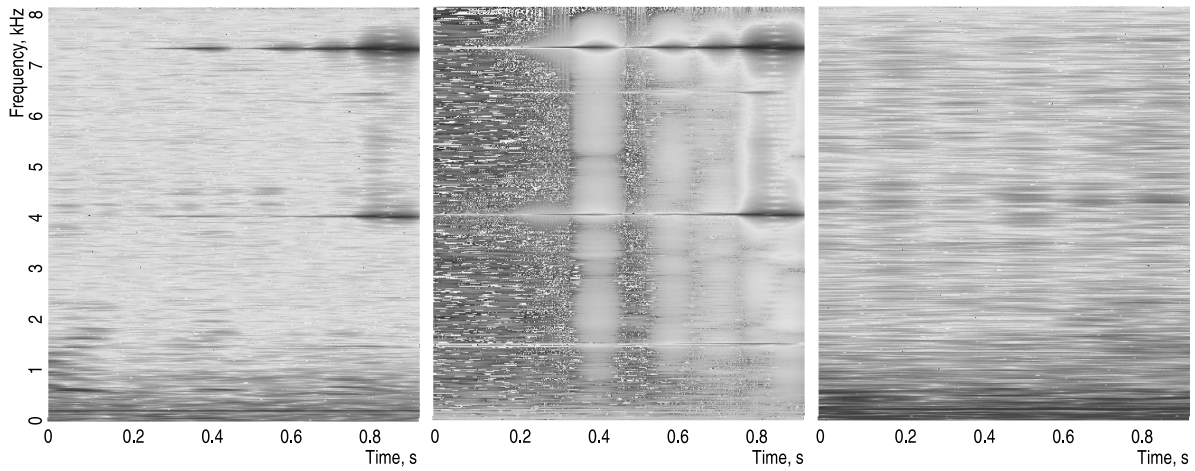
$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha + \mathbf{e}^T\alpha \tag{8}$$

where $\mathbf{e} = [1, \dots V]^T$ is a vector of all ones, Q is a $V \times V$ positive semi-definite matrix such that $Q_{ij} \equiv y_i y_j K(\mathbf{v}_i, \mathbf{v}_j)$ and $K(\mathbf{v}_i, \mathbf{v}_j) \equiv \psi(\mathbf{v}_i)^T\psi(\mathbf{v}_j)$ is the kernel function. Eqn. 8 is subject to $\mathbf{y}^T\alpha = 0, \quad 0 \leq \alpha_i \leq c, \quad i = 1 \dots V$.

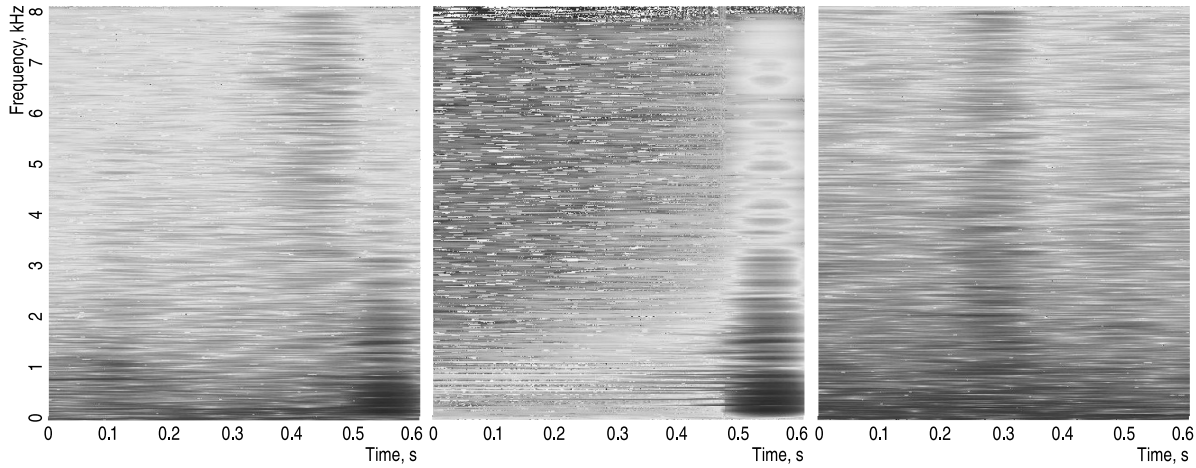
After solving eqn. (8), using the primal-dual relationship, the optimal $\mathbf{w}$ satisfies, $\mathbf{w} = \sum_{i=1}^{V} y_i\alpha_i\psi(\mathbf{x}_i)$. and the decision function is the sign of $\mathbf{w}^T\psi(\mathbf{v}_i) + b$ from eqn. 7 which can be computed simply using,

$$sgn\left(\sum_{i=1}^{V} y_i\alpha_i K(\mathbf{v}_1, \mathbf{v}) + b\right) \tag{9}$$

Using LIBSVM [38], several kernels $K(\mathbf{v}_i, \mathbf{v}_j)$ were tested, namely linear $\mathbf{v}_i^T\mathbf{v}_j$, third order polynomial $(\gamma\mathbf{v}_i^T\mathbf{v}_j)^3$, radial basis $e^{-\gamma||\mathbf{v}_i - \mathbf{v}_j||^2}$ and sigmoid, $\tanh(\gamma\mathbf{v}_i^T\mathbf{v}_j)$. All numerical results in this paper are given for a linear kernel, since this performed best for almost all experiments. The regularisation constant $c = 32$ was found to be very insensitive over a large range and we set $\gamma = 0.03$, which is close to the default (i.e. $1/N$ or 0.02) but, as estimated by the LIBSVM toolkit, resulted in slightly improved performance. In all cases,



(a) Destroyer + Ring001 SIF



(b) Factory + Drum004 SIF

Fig. 4. Example spectrograms of, from left to right: the 0dB combined signals, the noise-free named sound from the RWCP database, and the corresponding slice of the named NOISEX-92 background noise. The same sounds are shown using the SAI representation in Fig. 3.

system parameters were constant between classes (i.e. globally fixed). Given that LIBSVM uses the one-against-one multi-class method, $K(K-1)/2$ binary models were required to represent all classes. Since input scaling is important, the SVM input feature vector was mapped to the required input range $MIN \ldots MAX$ prior to training and testing,

$$v(i) = \frac{(MAX - MIN).(u(i) - \min(\mathbf{u}))}{\{(\max(\mathbf{u}) - \min(\mathbf{u})) - MIN\}} \quad (10)$$

for $i = 1 \ldots V$, where $u(i)$ denotes the ith element of unscaled input vector $\mathbf{u}$. Likewise $v(i)$ is the ith element of scaled feature vector $\mathbf{v}$. In this implementation, $MIN = -1$ and $MAX = +1$.

D. DNN classifier

An L-layer DNN classifier is constructed with the output layer in a one-of-K configuration (i.e. K classes), and the input layer fed with the feature vectors. The DNN is constructed from individual pre-trained RBM pairs, each of which comprise V visible and H hidden stochastic nodes, $\mathbf{v} = [v_1, v_2, ..., v_V]^\top$, and $\mathbf{h} = [h_1, h_2, ..., h_H]^\top$. Two different RBM structures are used in this paper. Intermediate and final layers are Bernoulli-Bernoulli, whereas the DNN input layer is formed from a Gaussian-Bernoulli RBM. In the former, nodes are assumed to be binary (i.e. $\mathbf{v}_{bb} \in \{0,1\}^V$ and $\mathbf{h}_{bb} \in \{0,1\}^H$), and the energy function of the state $E_{bb}(\mathbf{v}, \mathbf{h})$ is therefore:

$$E_{bb}(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ji} - \sum_{i=1}^{V} v_i b_i^v - \sum_{j=1}^{H} h_j b_j^h \quad (11)$$

w_{ji} represents the weight between the ith visible unit and the jth hidden unit and b_i^v and b_j^h are respective real-valued biases. Bernoulli-Bernoulli RBM model parameters are $\theta_{bb} = \{\mathbf{W}, \mathbf{b^h}, \mathbf{b^v}\}$, with weight matrix $\mathbf{W} = \{w_{ij}\}_{V \times H}$ and biases $\mathbf{b^h} = [b_1^h, b_2^h, ..., b_H^h]^\top$ and $\mathbf{b^v} = [b_1^v, b_2^v, ..., b_V^v]^\top$.

The Gaussian-Bernoulli RBM visible nodes are real (i.e. $\mathbf{v}_{gb} \in R^V$), while the hidden nodes are binary (i.e. $\mathbf{h}_{gb} \in \{0,1\}^H$). Thus, the energy function becomes:

$$E_{gb}(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} \frac{v_i}{\sigma_i} h_j w_{ji} + \sum_{i=1}^{V} \frac{(v_i - b_i^v)^2}{2\sigma_i^2} - \sum_{j=1}^{H} h_j b_j^h \quad (12)$$

Every visible unit v_i adds a parabolic offset to the energy function, governed by σ_i. Gaussian-Bernoulli RBM model parameters thus contain an extra term, $\theta_{gb} = \{\mathbf{W}, \mathbf{b^h}, \mathbf{b^v}, \sigma^2\}$, with variance parameter σ_i^2 pre-determined rather than learnt from training data.

Given an energy function $E(\mathbf{v}, \mathbf{h})$ defined as in either eqn. (11) or eqn. (12), the joint probability associated with configuration $(\mathbf{v}, \mathbf{h})$ is defined as,

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z} e^{\{-E(\mathbf{v}, \mathbf{h}; \theta)\}} \tag{13}$$

where Z is a partition function, $Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{\{-E(\mathbf{v}, \mathbf{h}; \theta)\}}$.

1) Pre-training: Given a training set, RBM model parameters θ can be estimated by maximum likelihood learning using the contrastive divergence (CD) algorithm [25]. This runs through a limited number of steps in a Gibbs Markov chain to update hidden nodes $\mathbf{h}$ given visible nodes $\mathbf{v}$ and then update $\mathbf{v}$ given the previously updated $\mathbf{h}$. The input layer is trained first (i.e. the layer 1 $\mathbf{v}_{gb}$ input is the feature vector $\mathbf{v}$ from Section III-B). After training, the inferred states of its hidden units $\mathbf{h_1}$ become the visible data for training the next RBM visible units $\mathbf{v_2}$. The process repeats to produce multiple trained layers of RBMs. Once complete, the RBMs are stacked to produce the DNN, as shown in Fig. 5.

2) Fine-tuning: A size K softmax output labelling layer is then added to the pre-trained stack of RBMs [37]. The function of the layer is to convert a number of Bernoulli distributed units in the final layer, $\mathbf{h_L}$, into a multinomial distribution through the following softmax function,

$$p(k|\mathbf{h_L}; \theta_{\mathbf{L}}) = \frac{\phi(k, \theta_L)}{\sum_{p=1}^{K} \phi(p, \theta_L)} \qquad for \quad k = 1 \ldots K \tag{14}$$

where θ_L represents the model parameters for the entire DNN, $\phi(k, \theta_L) = e^{\{\sum_{i=1}^{H} w_{ki} h_i + b_k\}}$, and $p(k|\mathbf{h}; \theta_L)$ denotes the probability of the input being classified into the k-th class.

Back propagation (BP) is then used to train the stacked network, including the softmax class layer, based on minimising the cross entropy error between the true class label, c and the class predicted by the softmax layer. The cross-entropy cost function, $\mathcal{C}$, is easily computed as $-\sum_{k=1}^{K} c_k \log p(k|\mathbf{h}; \theta_L)$.

Both the dimensions and number of hidden layers in the DNN are explored in Section IV-B6 to obtain a trade-off between performance and size. During training, dropout (proportion of weights fixed during training batches to prevent over-training) was maintained at 0.1, and mini-batch training size was set to 100, both being common default parameters. In all cases, the DNNs were pre-trained and fine-tuned exclusively with noise-free sound features, and used 1000 training epochs. Note that the winning label from the DNN softmax output layer will be post-processed to yield an overall classification result (described later in Section V-C).

IV. Evaluation and Design

This section begins by discussing and describing the performance evaluation used in this paper, before outlining a number of experiments that were conducted to explore various parameters in the systems prior to final system design. Finally, the structures, sizes and parameters of the evaluation systems will be presented.

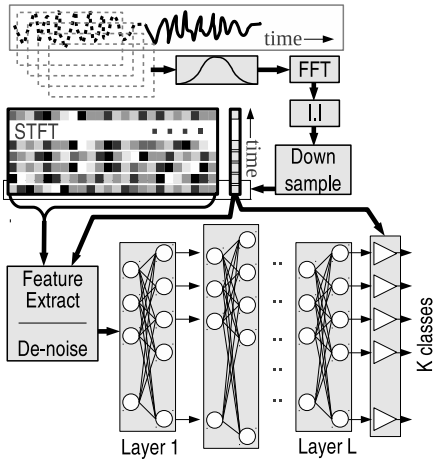


Fig. 5. Diagram showing detail of SIF formation and extraction of DNN feature vector.

A. The evaluation task

The evaluation task used in this paper is identical to that reported by Dennis et.al. in [14] and [15]. The advantage of using a standard evaluation is that it is repeatable by others, and eases the comparison of results with other published techniques that make use of the same evaluation method. In addition, the common availability of both the sound and noise databases is a significant advantage. A total of 50 sound classes are chosen from the Real World Computing Partnership (RWCP) Sound Scene Database in Real Acoustic Environments [36] following the selection criteria in [14]. In the RWCP database, every class contains 80 recordings, and contains a single example sound per recording. The sounds were captured with high SNR and have both lead-in and lead-out silence sections. As in [14], the training data set comprises 50 randomly-selected files from each class. The remaining 30 files from each class are set aside for evaluation. Therefore, a total of 2500 files are available for training and 1500 per testing run. All evaluations apart from the multi-condition tests use classifiers that are trained with exclusively clean sounds, with no pre-processing or noise removal applied. In all cases, evaluation is performed separately for both clean sounds, as well as sounds corrupted by additive noise. The noise-corrupted tests use four background noise environments selected from the NOISEX-92 database (again, we confine the selection to those used in [14], namely "Destroyer Control Room", "Speech Babble", "Factory Floor 1" and "Jet Cockpit 1"). These environments were chosen by Dennis [15] as realistic examples of non-stationary noise with predominantly low-frequency components. During evaluation under noisy conditions, noise is added to the test data set at levels of 20, 10 and 0 dB SNR. For each file in the test data set, one of the four NOISEX-92 recordings is randomly selected, a random starting point identified within the noise file, and then sample-wise added, at the given SNR to the sound file. SNR is calculated over the entire noise and sound file in each case.

The multi-condition evaluations train the system with a variety of clean and noise-corrupted sounds, again exactly following the evaluation method of Dennis [12]. "Speech

Babble" is chosen for use in the evaluation, while training data comprises a random selection of clean sounds and noise-corrupted sounds using the remaining three noises at SNRs of 20dB and 10dB as well as clean sounds. The intention is that this approach allows the trained systems to be less sensitive to the effects of noise. Training using a larger selection of noise types might be expected to further reduce noise sensitivity, although this would in turn require a larger data and would not comply with the standard evaluation methodology.

B. System Design

Both the feature extraction methods as well as the classifiers are highly tunable, with a very large degree of freedom in terms of system design and parameter choices. Many experiments were therefore conducted to evaluate performance related to parameter choice. While these are not exhaustive searches of all parameters, the following subsections discuss design choices and present experimental results relating to system design which may be valuable to other researchers. The final system designs used for evaluation will be given in Section IV-C.

1) VQ: The Google-SAI system makes use of VQ to form a sparse representation of the input data for the PAMIR classifier, reporting results for a 256 entry codebook [4]. When testing on standard tests using RWCP data (Section IV-A), we found that performance scaled nearly linearly with codebook sizes from 128 to 1024 (i.e. top-5 accuracy for power-of-two codebook sizes between 32 and 1024 was 81%, 81%, 88%, 90%, 92%, 95%). However, in all tested cases, under many evaluation conditions, the removal of the VQ and sparse coding stages resulted in higher SVM classification accuracy. This is understandable since the initial use of both VQ and sparse coding in [2] was motivated for reasons of computational efficiency, rather than classification performance. Since the current paper is primarily concerned with robust sound event classification performance, neither VQ nor sparse coding will be employed in the final evaluations.

2) Region selection: The Google-SAI system used $R = 49$ rectangular regions extracted from each SAI, with the regions selected according to a 'start-small and then double' heuristic, clipped to the SAI window extents, as outlined in [6]. As mentioned in Section III-A the initial window had dimensions of $D = 16$ and $B = 32$. We used the same heuristic and sizes and evaluated system performance with differing number of windows $R = 1 \ldots 147$. In general, a slight performance improvement over the baseline system was found to be achievable by using $R = 54$ windows but with decreasing gains as further windows were added. In addition, a slightly smaller initial window size of $D = 16$ and $B = 24$ was found to perform well. For example, noise-free classification accuracy in a baseline SVM classifier with SAI input features for $R = 54$ windows was 93.40%, whereas it only rose to 93.60% for $R = 120$ windows, at a significant additional computational and memory cost (corresponding 0dB SNR noise figures are 6.47% and 6.67%). Furthermore, the DNN classifier, discussed below, performed better by representing the entire auditory image by a single 16×24 sized down-sampled window (with time domain context – see below).

3) Connected frames and context: Various techniques in ASR exploit longer duration context in the front-end feature vector (e.g. Shifted Delta Cepstrum [39]) to improve performance. Similar approaches could reasonably be expected to have greater importance for the current application since it lacks an equivalent to the back-end language model used in ASR. We therefore explored two different approaches to incorporating temporal context of T windows. The first used a method similar to the Google-SAI system by computing the mean of input feature vectors (in our case, across T windows, rather than over the entire variable-length file as in the Google-SAI system). However there are now multiple feature vectors representing each file. The second method was to concatenate T features (i.e. feature dimension becomes $T(B + D)$) to form a larger feature vector, and again there are many feature vectors representing each file. Having been trained on individual feature vectors, the SVM classifiers in each case produce one classification output or 'vote' per testing context, with a sound file being classified based upon the output class which receives the most votes. Note that another method of combining scaled classifier outputs will be evaluated in Section V-C.

On the standard RWCP evaluation task for noise-free sounds (Section IV-A), performance tended to improve with increasing context length from $2 \ldots 10$. Classifying on context shows modest gains of up to 1.2% with a context size of 8, decreasing thereafter, over static baseline performance of 89.53%. However the higher dimensionality feature vector of the second method achieved greater improvements of 0.8%, 1.6%, 3.0%, 3.9%, 6.7%, 6.6%, 6.6%, 7.0%. 7.1% (as context increased from $2 \ldots 10$). The implication was clear that temporal information is under-represented in the static feature vectors from individual frames, and thus a context size of $T = 10$ was chosen for the SAI features. In Section V, results will compare the use of context ($T = 10, R = 1$) with no context ($T = 1, R = 54$) for an SVM classifier, as well as explore the context system performance with a DNN classifier.

4) Alternative computation of marginals: Even a naïve summation of SAI region marginals works well in practice, as demonstrated by the Google-SAI recall system [4]. However this clearly ignores second order statistics; classifying 16 mid-grey level samples would be equivalent to 8 black and 8 white samples, which would in turn rate a striped SAI region as being equivalent to one of unvarying greyness (refer to Fig. 3 to see the 'stripes' visible in the SAI plots). However our experiments revealed that increasing the SAI-derived feature vector size by incorporating a variance statistic does not meaningfully influence the results in an $R = 54$ region SVM classifier evaluation. However replacing the naïve summation with a Minkowski sum [40] (which gives preference to the 'louder' elements, for vectors A and B, $sum(A, B) = a + b : a \in A, b \in B$) traded a very slight 0.8% reduction in noise-free SVM accuracy against a similar improvement for noisy cases. Since the gain was not significant, the evaluation results reported in Section V maintain the use of naïve marginal computation.

5) Effect of acoustic noise type: Although the standard evaluation task (Section IV) uses noise from the NOISEX-92 database, we also compared this against performance with

TABLE I
COMPARISON OF PERFORMANCE WITH AWGN AND NOISEX-92 NOISE

System	clean	20dB	10dB	0dB	mean
AWGN	93.40%	47.40%	26.80%	12.13%	44.93%
NOISEX-92	93.40%	59.27%	26.87%	6.47%	46.50%

AWGN-corrupted sounds at the same SNR levels. The results, reported in Table I, using an SVM baseline classifier, indicate that, although the 10dB performance is similar, the NOISEX-92 task is more 'difficult' overall than classification in AWGN.

6) DNN size and structure: The DNN classifier, described in Section III-D, must support an input feature dimension of $T(B+D) = 10(24+16) = 400$ for the SAI features with context. For the SIF features, input feature dimension is $(B.D)+1 = (30 \times 24)+1 = 721$. In both cases, the number of output layers, $K = 50$, determined by the number of sound classes in the standard RWCP evaluation. The internal network layer dimensions were initially set to follow Hinton et. al. [28] for his DNN examples, before a step-wise search (minimum resolution 10) of hidden layer widths of between 100 and 300 was performed (as well as experiments for 500, 1000, 2000 nodes), while constraining internal layers to be of equal size. Given $L = 2$ layers, for the given evaluation on SIF features on noise-free sounds, performance was found to increase up to 210 hidden nodes (92.35% accuracy). Increasing the number of hidden nodes to 250 slightly reduced performance to 92.10%. Increasing further saw accuracies of 92.52% for 300 nodes, 92.77% for 500 nodes, 92.63% for 1000 nodes and 92.70% for 2000 nodes (all at the cost of increased computation time – the latter two investigations requiring several days of GPU time). A brief investigation was also made into the effects of depth. With inner layer size set to 210 nodes, five-layer performance (i.e. an additional hidden layer of dimension 210) was 92.22% and six-layer performance was 92.13%. The investigations thus showed only marginal changes in performance as depth increased beyond two hidden layers, and beyond 210 hidden nodes, and thus the baseline SIF feature DNN classifier structure was set to 721–210–210–50 An equivalent layer size investigation was performed for the SAI features, revealing optimal performance with hidden layers of 200 nodes. Thus the baseline SAI feature DNN classifier structure was set to 400–200–200–50.

C. Final Structure

The final structure of the systems used for evaluation in the following section are shown in Table II. The context refers to the number of connected frames presented within a single feature vector, as described in Section IV-B3, whereas the Time and Frequency resolutions shown are the final downsampled sizes used to represent a single analysis frame, which has been created using the given time domain window size.

V. RESULTS AND DISCUSSION

This section will first present the performance of other reported systems, before evaluating the performance of the proposed robust sound event classification systems for both SVM and DNN classifiers, using several arrangements of SAI and SIF features.

TABLE II
FINAL SYSTEM PARAMETERS FOR EVALUATION

Classifier	SVM		DNN	
Features	SAI	SAI	SAI	SIF
Context T (frames)	1	10	10	30
Freq. resolution B	24	24	24	24
Time resolution D	16	16	16	1
Window	35ms	35ms	35ms	128ms
Regions R	54	1	1	1
Feature dimension	2160	400	400	721
Hidden layers L	N/A	N/A	2	2
Hidden nodes H	N/A	N/A	200	210

A. Comparison with other systems

A significant advantage of choosing a standard evaluation task (Section IV-A), allows comparison against other systems [12]. Table III reveals the performance some of the many systems evaluated by Dennis [15]. These include hidden Markov models (HMM) with MFCC features, also the same features used with an SVM classifier. Both exhibit good performance in noise-free conditions, but degrade significantly in noise. The latter classifier was also evaluated with an ETSI Advanced Front End (ETSI-AFE) toolkit enhancement [41], which uses noise removal techniques to significantly improve performance in noisy conditions. The MPEG-7 method uses a set of 57 features per frame, reduced to a dimensionality of 12 through principal component analysis (PCA) [42], and then augmented with difference and acceleration features. These are used in conjunction with a 5 state HMM having 6 Gaussian mixtures. The Gabor method used a feature-finding single-layer perceptron network to select the best 36 features [15]. This yielded the highest noise-free performance of all tested systems.

Gammatone cepstral coefficients were extracted by 36 gammatone filters in the GTCC system, then reduced to 12 dimensions using PCA before being augmented in the same way as the MPEG-7 method. The MP+MFCC system used matching pursuit (MP) [43] to find the top five Gabor bases from a decomposition of the signal window, yielding four mean and variance features from the Gabor bases scale and frequency parameters. These were concatenated with MFCC features, before being augmented with deltas and accelerations to form the final feature vector. Finally, Dennis developed a SIF extraction method ('Dennis SIF') as described in Section II-B which is shown in Table III to improve performance in noise.

B. SVM and DNN performance with SAI features

The SAI features, computed as in Section III-A, are evaluated with the SVM and DNN classifiers of Sections III-C and III-D respectively. All parameters are as specified in Section IV-C. Table IV reports the classification accuracy for various degrees of corrupting noise. All systems perform well for

TABLE III
CLASSIFICATION ACCURACY FOR SEVERAL STATE-OF-THE-ART SOUND EVENT DETECTION METHODS (RESULTS COURTESY OF [15])

System	clean	20dB	10dB	0dB	mean
MFCC-HMM	99.4%	71.9%	42.3%	15.7%	57.4%
MFCC-SVM	98.5%	28.1%	7.0%	2.7%	34.1%
ETSI-AFE	99.1%	89.4%	71.7%	35.4%	73.9%
MPEG-7	97.9%	25.4%	8.5%	2.8%	33.6%
Gabor	99.8%	41.9%	10.8%	3.5%	39.0%
GTCC	99.5%	46.6%	13.4%	3.8%	40.8%
MP+MFCC	99.4%	78.4%	45.4%	10.5%	58.4%
Dennis SIF	91.1%	91.1%	90.7%	80.0%	88.5%

TABLE IV
CLASSIFICATION ACCURACY FOR SAI FEATURES USING SVM AND DNN CLASSIFIERS

System	clean	20dB	10dB	0dB	mean
SAI, static, SVM	93.40%	59.27%	26.87%	06.47%	46.50%
SAI $T = 10$, SVM	94.33%	75.60%	41.73%	09.73%	55.35%
SAI $T = 10$, DNN	96.20%	77.40%	49.80%	19.13%	60.63%

classification of noise-free sounds but degrade sharply with additive noise.

The incorporation of SAI context is shown to improve the mean SVM classification score by 19%, with the greatest performance improvement being for the 10dB noise condition. However, exactly the same features classified using the DNN, achieves an additional mean improvement of 10%, with by far the greatest contribution occurring for the 0dB noise condition. It appears that, while the DNN yields only moderate benefits for noise-free sound classification, it works particularly well in high levels of noise. Thus the results highlight the noise-robust discriminative capabilities of the DNN.

C. DNN performance with SIF features

Since Dennis et. al. achieved good results with SIF-like features [12], as reported in Section V-A, the DNN classifier was next trained and evaluated with SIF input, extracted as outlined in Section III-D. The DNN size is 721-210-210-50 and incorporates a $T = 10$ input context, with all other parameters as specified in Section IV-C.

Table V presents results from a number of systems. The baseline is a straightforward DNN classifier with a $\delta = 16$ sample step between spectrogram windows, and no denoising. The overall classification from testing one sound file is the maximum class score from the mean of all classification results (since there are multiple classification contexts per file).

The baseline DNN achieves 98.07% noise-free accuracy, but only 31.20% for the 0dB noise condition. Compared to the results in Table III, the DNN performance is positioned between Dennis's SIF result [15] and the others in the table: Noise-free accuracy is within 1.8% of the highest score, yet the 0dB accuracy ranks third, and is relatively good compared to all but the Dennis result.

Moving down Table V, **voting** (denoted as -v) classifies a sound file based on votes from the individual context winning class outputs. This improves low-noise performance, but degrades the 0dB result. **e-scaled** (denoted as -e) weights the votes from individual classification contexts by the context energy $v(B.D)$, in eqn. (4). The idea being that quieter (low-energy) regions of the sound file are likely to contribute less discriminative capability than higher energy regions, and therefore receive a reduced voting weight. It can be seen that this trades off around 2.5% noise-free performance to purchase a significant improvement in noise-corrupted performance.

TABLE V
CLASSIFICATION ACCURACY FOR OVERLAPPED SIF FEATURES USING DNN CLASSIFIER ON 16-STEP OVERLAPPING FRAMES

System	clean	20dB	10dB	0dB	mean
baseline	98.07%	85.07%	67.53%	31.20%	70.47%
DNN-v	98.07%	87.27%	70.67%	28.07%	71.02%
DNN-e	95.87%	93.73%	86.40%	45.80%	80.45%
DNN-DN	96.73%	94.60%	90.27%	76.47%	89.52%
DNN-DN-v	98.87%	95.33%	92.40%	78.87%	91.37%
DNN-DN-e	96.00%	94.37%	93.53%	85.13%	92.26%

TABLE VI
CLASSIFICATION ACCURACY FOR DE-NOISED OVERLAPPED SIF FEATURES USING DNN CLASSIFIER

System	clean	20dB	10dB	0dB	mean
DNN-δ25-v	98.13%	95.07%	89.00%	73.47%	88.92%
DNN-δ25-e	96.47%	93.60%	89.67%	78.87%	89.65%
DNN-δ16-v	**98.87%**	95.33%	92.40%	78.87%	91.37%
DNN-δ16-e	96.00%	94.37%	93.53%	85.13%	92.26%
DNN-δ8-v	98.67%	95.80%	92.60%	79.40%	91.62%
DNN-δ8-e	96.20%	95.80%	**94.13%**	**85.47%**	**92.90%**
DNN-δ4-v	98.20%	**96.13%**	90.73%	71.73%	89.20%
DNN-δ4-e	95.53%	95.07%	92.13%	82.67%	91.35%

In Section V-A, it was noted that the addition of ETSE-AFE de-noising technique to the MFCC-HMM system was able to significantly improve performance in noise. Therefore a simple de-noising technique was developed for the SIF features used with the DNN classifier, as shown in eqn. (5). When this is applied to the baseline system, the result (listed as **DNN-DN** in Table V), is again a trade off with slightly reduced noise-free accuracy but 0dB performance.

DNN-DN-v and **DNN-DN-e** combine both techniques discussed above (voting or e-scaling and simple de-noising). Overall results are excellent, achieving mean accuracies of 91.37% and 92.26% respectively; better than any other reported techniques evaluated on the same standard tests.

D. Exploring DNN performance and overlap

Table VI further explores the performance of the DNN classifier with SIF input, with either straightforward voting or e-scaling of the output classifications, while adjusting the step size between analysis windows. Altering δ in eqn. (1) has the effect of changing the time resolution of the SIF. From the results listed, which show the best performance for each level of noise in bold text, it can be seen firstly that $\delta = 16$ performs best for noise-free classification, whereas a smaller step size is preferred for classification of noise-corrupted sounds.

This tendency is far easier visualised in Fig. 6 which plots the main results discussed in this section in terms of error rate (rather than accuracy) for different features and noise levels.

Three trends are distinguishable from the plot, which shows generally improving results from left to right. Firstly, that major performance improvements from left to right are in the

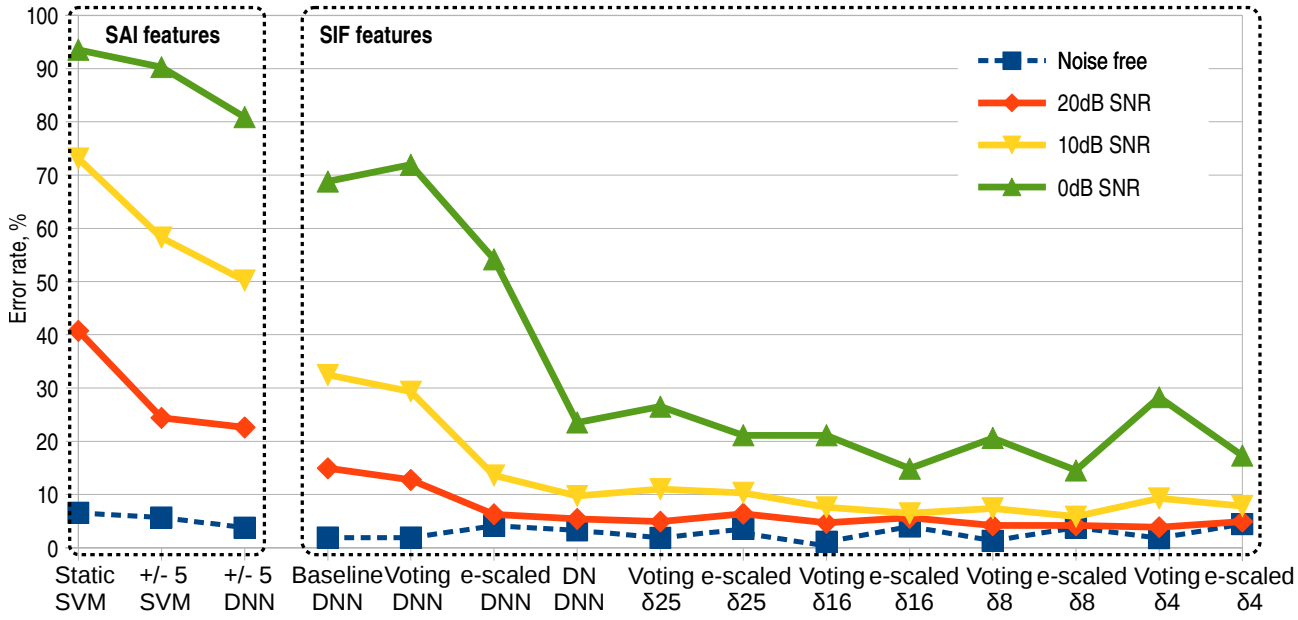


Fig. 6. Results of various system configuration in clean and noisy conditions, in terms of error rate.

TABLE VII
PERFORMANCE OF MFCC FEATURES WITH HMM, SVM AND DNN (THE HMM AND SVM RESULTS ARE COURTESY OF [15]).

Noise	clean	20dB	10dB	0dB	mean
MFCC-SVM	98.5%	28.1%	7.0%	2.7%	34.1%
MFCC-DNN	89.5%	48.8%	29.9%	11.2%	44.9%

higher noise cases, and predominantly contributed by the de-noising process. Secondly that the gap between performance curves at the left of the graph is almost linear, whereas at the right hand side it is exponential, meaning that the effect of noise in the 0dB case is becoming intractable given the tested techniques. Thirdly, there is an interesting undulation in results for the 8 most right hand systems. Grouping the 0 and 10dB results together as 'high noise' cases, and the noise-free and 20dB results as 'low noise' cases, we can see that minima in the former coincide with maxima in the latter. This clearly illustrates that the voting systems favour low noise, whereas energy scaled systems favour high noise conditions.

To further highlight the ability of the DNN compared to SVM, Table VII reproduces the MFCC-SVM results of Table III and compares them to the performance achieved by a DNN. The input features were 12 MFCC coefficients including DC, with delta and delta-delta and a context size of ± 5, as in the main results above. It is evident that DNN performance in clean conditions was reduced compared to SVM, but improved significantly in the presence of background noise. Most importantly, these MFCC results are bettered by all systems using SAI and SIF features. We thus argue that using the more representative input features of MFCC, the classification power of DNN slightly exceeds that of SVM. However given richer input features (being those derived from SAI and SIF respectively), the discriminative abilities of the DNN appear more able to extract meaningful classification relationships.

TABLE VIII
MULTI-CONDITION (MC) CLASSIFICATION ACCURACY COMPARED TO MISMATCHED CLASSIFICATION PERFORMANCE (PRESENTED IN ITALICS) FOR SEVERAL SYSTEMS.

Noise	clean	20dB	10dB	0dB	mean
SAI-DNN	96.20%	77.40%	49.80%	19.13%	60.63%
SAI-DNN-MC	63.67%	62.73%	57.87%	26.80%	52.77%
SIF-DNN-DN-v	98.87%	95.33%	92.40%	78.87%	91.37%
SIF-DNN-DN-e	96.00%	94.37%	93.53%	85.13%	92.26%
SIF-DNN-MC-v	96.90%	96.90%	93.20%	80.40%	91.85%
SIF-DNN-MC-e	94.70%	95.80%	92.10%	87.70%	92.58%
SIF-SVM-MC	91.13%	91.10%	90.71%	80.95%	88.55%

E. Multi-condition noise tests

Multi-condition (MC) results are presented in Table VIII, where classification is evaluated for sounds corrupted with NOISEX-92 "Speech Babble", on systems trained with "Destroyer Control Room", "Factory Floor 1" and "Jet Cockpit 1" at two noise levels, and compared to the non-MC mismatched results reported previously (shown in italics). For SAI features, MC training significantly improves the robustness to noisy conditions, but at the expense of a considerable reduction in classification accuracy for low noise conditions. Both reported systems have identical DNN structures and context size of $T = 10$ and used only a voting approach since the feature lacks energy informations.

SIF results are then reported for both e-scaled and voting mechanisms. For the SIF features, MC training yields a very slight performance improvement over the denoised SIF-DNN systems, mainly under high noise conditions as would be expected, and again at the expense of some low noise performance. Clearly SIF outperforms SAI in all tests, and at all noise levels, whether MC training is used or not.

Multi-condition testing was also used by Dennis et. al. [15] to evaluate his SIF method with SVM classifier, as shown at the bottom of Table VIII, achieving an average accuracy

of 88.55%. It should also be noted that they proposed, in [14], a more complex sub-band power distribution (SPD) image feature with a *k*NN classifier and a powerful de-noising strategy: Assuming a clean sample of the background noise is available, their method computes a noise mask which is then applied to the combined sound plus noise signal prior to classification. Impressive results are achieved, averaging 95.95% accuracy. The technique would be a good choice where noise is known or is able to be estimated accurately prior to presentation of the noise-corrupted sound.

VI. Conclusion and Future Work

This paper has proposed a DNN-based robust sound classification system, and evaluated performance using a standard database and assessment task. Starting from the state-of-the-art Google-SAI PAMIR recall method, the same features were evaluated with an SVM classifier. Additional context information was found to improve performance, thus this was incorporated along with adjustments to SAI window regions, and evaluated with both SVM and DNN classifiers. Performance under noise-free conditions was good, but degraded rapidly with increasing levels of noise. Multi-condition training was shown to be able to mitigate much of the performance loss in high noise conditions, but at the expense of a considerable reduction in classification accuracy of clean sounds.

Subsequently, a novel low-resolution overlapped spectrogram image feature was developed and evaluated with the DNN classifier. Several variants of the system were then proposed and evaluated, including a simple de-noising method as well as post-processing of context-by-context classification outputs across a single sound. Multi-condition training, where the DNN is trained with noise-corrupted samples, was found to improve classification performance for high noise conditions, achieving an average accuracy of 92.58%. In general, the task of classifying sounds in high levels of noise is found to be extremely challenging. Results reported here and elsewhere indicate a trade-off between performance in noise-free and in high noise conditions. Systems performing best in clean conditions are seldom able to cope with high levels of noise. Conversely, the best-performing systems in high levels of noise will often sacrifice some performance in clean conditions. This highlights the importance of testing such systems in realistic, noisy conditions or in developing adaptive systems. Finally, it should be noted that there are a large number of tunable parameters related to the front end feature-extraction process, the DNN classifier, and the classification post-processor. Not all parameters and combinations have been fully explored in this paper.

Acknowledgment

This work is supported by the Huawei Innovation Research Program under Machine Hearing and Perception Project Contract No. YB2012120147, and by the Fundamental Research Funds for the Central Universities, China, under grant no. WK2100000002. Yan Song is supported by the Natural Science Foundation of China (NSFC, Grant No. 61172158)

References

[1] R. F. Lyon, "Machine hearing: an emerging field," *IEEE SIgnal Processing Magazine*, vol. 42, pp. 1414–1416, 2010.

[2] R. F. Lyon, M. Rehn, S. Bengio, T. C. Walters, and G. Chechik, "Sound retrieval and ranking using sparse auditory representations," *Neural computation*, vol. 22, no. 9, pp. 2390–2416, 2010.

[3] R. F. Lyon, M. Rehn, T. Walters, S. Bengio, and G. Chechik, "Audio classification for information retrieval using sparse features," Mar. 2010, uS Patent App. 12/722,437.

[4] R. F. Lyon, J. Ponte, and G. Chechik, "Sparse coding of auditory features for machine hearing in interference," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5876–5879.

[5] R. F. Lyon, "Machine hearing: Audio analysis by emulation of human hearing," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. viii–viii.

[6] T. C. Walters, "Auditory-based processing of communication sounds," Ph.D. dissertation, University of Cambridge, 2011.

[7] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 209–215, 2003.

[8] C.-C. Lin, S.-H. Chen, T.-K. Truong, and Y. Chang, "Audio classification and categorization based on wavelets and support vector machine," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 644–651, 2005.

[9] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Workshop on machine listening in Multisource Environments*, 2011, pp. 36–40.

[10] L.-H. Cai, L. Lu, A. Hanjalic, and H.-J. Zhang, "A flexible framework for key audio effects detection and auditory context inference," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 1026–1039, 2006.

[11] G. Chechik, E. Ie, M. Rehn, S. Bengio, and R. F. Lyon, "Large scale content-based audio retrieval from text queries," in *ACM International Conference on Multimedia Information Retrieval (MIR)*, 2008.

[12] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," *Signal Processing Letters, IEEE*, vol. 18, no. 2, pp. 130–133, 2011.

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

[14] ——, "Image feature representation of the subband power distribution for robust sound event classification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 367–377, 2013.

[15] J. W. Dennis, "Sound event recognition in unstructured environments using spectrogram image processing," Ph.D. dissertation, Nanyang Technological University, Singapore, 2014.

[16] I. V. McLoughlin, *Applied Speech and Audio Processing*. Cambridge University Press, 2009.

[17] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro, "Non-speech audio event detection," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 1973–1976.

[18] A. Plinge, R. Grzeszick, and G. A. Fink, "A bag-of-features approach to acoustic event detection," in *Acoustics, Speech and Signal Processing, 2014. ICASSP 2014 Proceedings. 2014 IEEE International Conference on*. IEEE, 2014, pp. 3732–3736.

[19] H. Phan, M. Maa, R. Mazur, and A. Mertins, "Acoustic event detection and localization with regression forests," in *15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014)*, Singapore, September 2014.

[20] P. K. Atrey, M. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[21] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "On the use of audio events for improving video scene segmentation," in *Analysis, Retrieval and Delivery of Multimedia Content*, ser. Lecture Notes in Electrical Engineering, N. Adami, A. Cavallaro, R. Leonardi, and P. Migliorati, Eds. Springer New York, 2013, vol. 158, pp. 3–19. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-3831-1_1

[22] R. D. Patterson, M. H. Allerhand, and C. Giguere, "Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform," *The Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1890–1894, 1995.
[23] S. Bleeck, T. Ives, and R. D. Patterson, "Aim-mat: the auditory image model in matlab," *Acta Acustica United with Acustica*, vol. 90, no. 4, pp. 781–787, 2004.
[24] T. C. Walters, "Aim-c, a c++ implementation of the auditory image model," 2014. [Online]. Available: http://code.google.com/p/aimc/
[25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[26] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.
[27] R. Gupta, K. Audhkhasi, S. Lee, and S. Narayanan, "Paralinguistic event detection from speech using probabilistic time-series smoothing and masking," *Proc. of Interspeech, Lyon, France*, pp. 173–177, 2013.
[28] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Mag*, vol. 29, no. 6, pp. 82–97, 2012.
[29] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans Audio Speech Lang Processing*, vol. 20, no. 1, pp. 7–13, 2012.
[30] Z. Huang, C. Weng, K. Li, Y.-C. Cheng, and C.-H. Lee, "Deep learning vector quantization for acoustic information retreival," in *Acoustics, Speech and Signal Processing, 2014. ICASSP 2014 Proceedings. 2014 IEEE International Conference on*. IEEE, 2014, pp. 1364–1368.
[31] B. R. Glasberg and B. C. Moore, "A model of loudness applicable to time-varying sounds," *Journal of the Audio Engineering Society*, vol. 50, no. 5, pp. 331–342, 2002.
[32] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Academic Press, 1992.
[33] B. C. Moore and R. D. Patterson, *Auditory frequency selectivity*. Plenum Press, 1986.
[34] F. Bergeaud and S. G. Mallat, "Processing images and sounds with matching pursuits," in *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*. International Society for Optics and Photonics, 1995, pp. 2–13.
[35] D. Grangier and S. Bengio, "A discriminative kernel-based approach to rank images from text queries," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 8, pp. 1371–1384, 2008.
[36] S. Nakamura, K. Hiyane, F. Asano, T. Yamada, and T. Endo, "Data collection in real acoustical environments for sound scene understanding and hands-free speech recognition," in *EUROSPEECH*, 1999, pp. 2255–2258.
[37] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, Technical University of Denmark, 2012.
[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[39] W.-Q. Zhang, L. He, Y. Deng, J. Liu, and M. T. Johnson, "Time–frequency cepstral features and heteroscedastic linear discriminant analysis for language recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 266–276, 2011.
[40] R. Schneider, *Convex bodies: the Brunn–Minkowski theory*. Cambridge University Press, 2013, vol. 151.
[41] A. Sorin and T. Ramabadran, "Extended advanced front end algorithm description, version 1.1," *ETSI STQ Aurora DSR Working Group, Tech. Rep. ES*, vol. 202, p. 212, 2003.
[42] M. Casey, "Mpeg-7 sound-recognition tools," *IEEE Transactions on circuits and Systems for video Technology*, vol. 11, no. 6, pp. 737–747, 2001.
[43] S. Chu, S. Narayanan, and C.-C. Kuo, "Environmental sound recognition with time–frequency audio features," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 1142–1158, 2009.

Ian McLoughlin (M'94, SM'04) has worked in industrial R&D and academia in five countries and three continents. He became a Chartered Engineer (UK) in 1998, and a Fellow of the IET in 2013. He is a Professor in the National Engineering Laboratory of Speech and Language Information Processing at the University of Science and Technology of China (USTC). His PhD in EEE was granted by the University of Birmingham, UK in 1997.

Haomin Zhang Haomin Zhang received the B.E. degree from the Department of Electronic Engineering & Information Science at the University of Science and Technology of China, Hefei, China in 2013. He is currently pursuing his M.E.E. degree in the same university. His current research interests include sound event detection and advanced machine learning techniques.

Zhipeng Xie Zhipeng Xie received the B.E. degree in Communications Engineering from Anhui University, Hefei, China in 2013. He is currently working toward his M.E.E. degree in the Department of Electronic Engineering & Information Science at the University of Science and Technology of China, Hefei, China. His current research interests include sound event detection and machine hearing based on audio representations.

Yan Song Yan Song received his B.Sc degree in Electronic Engineering from the University of Electronic Science and Technology of China in 1994. He received an M.Sc and Ph.D degree from the Department of Electronic Engineering and Information Science at the University of Science and Technology of China in 1997 and 2006. He has been a lecturer in Department of Electronic Engineering and Information Science of University of Science and Technology of China since 2000. Currently he is working in the National Engineering Laboratory for Speech and Language Information Processing. His research interests include Multimedia information processing, automatic language identification, speaker diarization and image classification.

Wei Xiao Wei Xiao got his B.Sc and MPhil degrees both from Central South University, Changsha, P.R. China. He joined Huawei Technologies in 2006, and now is a senior research engineer in Huawei Technologies Duesseldorf GmbH, Munich, Germany. His research includes auditory modeling, quality assessment of speech and audio, quality enhancement. He has over 20 granted or pending patents, and is also a delegate to ITU-T SG12 in charge of activities related to quality assessment of speech and audio, and serves as a member of the Huawei MPEG Expert Group.