MTF-CRNN: Multiscale Time-Frequency Convolutional Recurrent Neural Network for Sound Event Detection

KEMING ZHANG, YUANWEN CAI, YUAN REN, RUIDA YE, AND LIANG HE, (Member, IEEE)

1Space Engineering University, Beijing 101416, China
2Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
3Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

Corresponding authors: Liang He (heliang@mail.tsinghua.edu.cn) and Yuan Ren (renyuan_823@aliyun.com)

This work was supported by the National Natural Science Foundation of China under Grant 61403224, Grant 11772001, and Grant U1836219.

ABSTRACT To reduce neural network parameter counts and improve sound event detection performance, we propose a multiscale time-frequency convolutional recurrent neural network (MTF-CRNN) for sound event detection. Our goal is to improve sound event detection performance and recognize target sound events with variable duration and different audio backgrounds with low parameter counts. We exploit four groups of parallel and serial convolutional kernels to learn high-level shift-invariant features from the time and frequency domains of acoustic samples. A two-layer bidirectional gated recurrent unit is used to capture the temporal context from the extracted high-level features. The proposed method is evaluated on two different sound event datasets. Compared to that of the baseline method and other methods, the performance is greatly improved as a single model with low parameter counts without pretraining. On the TUT Rare Sound Events 2017 evaluation dataset, our method achieved an error rate (ER) of 0.09 ± 0.01, which was an improvement of 83% compared with the baseline. On the TAU Spatial Sound Events 2019 evaluation dataset, our system achieved an ER of 0.11 ± 0.01, a relative improvement over the baseline of 61%, and F1 and ER values that are better than those of the development dataset. Compared to the state-of-the-art methods, our proposed network achieves competitive detection performance with only one-fifth of the network parameter counts.

INDEX TERMS Pattern recognition, sound event detection, multiscale learning, time-frequency transform, convolutional recurrent neural network.

I. INTRODUCTION

Sound event detection (SED) recognizes a target sound and detects the onset and offset times in an audio recording. In our everyday lives, sound carries a large amount of information. We can detect audio signals such as gunshots, the cries of babies, falls, and the malfunctioning of a machine through the sound that is given off, as well as endangered animal sounds, which allows us to respond appropriately [1]. In automatic driving, which is currently popular, we can detect whistling alarm sounds and a ring or call for help [2]. SED has always been a research hotspot and has been utilized in many fields, such as transportation anomaly detection, oil and gas pipeline anomaly detection, bridge anomaly detection, seismic wave acoustic detection, and equipment failure monitoring [3], [4].

Traditional sound event detection methods mainly include signal analysis, information entropy [5], statistical analysis, and clustering [6], [7]. Research on SED has shifted from Gaussian mixture models-hidden Markov models (GMM-HMMs) and support vector machines (SVMs) [8] to deep neural networks (DNNs) [9], convolutional neural networks (CNNs), recurrent neural networks (RNNs), and convolutional recurrent neural networks (CRNNs).

Phan proposed a DNN and CNN with weighted and multitask loss functions for audio event detection and focused on improving the loss function of the neural network [10]. Çakir and Virtanen combined the CNN and RNN to detect sound events [11], [12]. Drossos et al. innovatively utilized...
depthwise separable convolutions instead of a CNN and the replacement of the RNNs with dilated convolutions [13]. Stowell et al. reported on the best detection and classification of acoustic scenes and events based on newly created audio datasets and a baseline system [14]. The above methods have obtained good results while overcoming their individual weaknesses. Lim et al. proposed the use of a combination of a 1D convolutional neural network and recurrent neural network (1D-CRNN) to detect a rare sound event [15], and their approach won first place in the Task 2 Challenge on Detection and Classification of Acoustic Scenes and Events DCASE2017 [16]. The best performance is achieved by an ensemble method that combines many models. Compared with the Faster-RCNN [17] and the extended R-FCN [18], Kao et al. introduced a region-based convolutional RNN (R-CRNN) for audio event detection and claimed it was the best single-model method among all methods without using ensembles [19]. Wang et al. proposed a simple recurrent model for detecting rare sound events, which achieved competitive performance [20]. Shen et al. presented a temporal-frequential attention model for sound event detection, which is an innovative pretrained method [21]. Zhang et al. used a multiscale hourglass structure that outperformed previous single-model methods for rare sound event detection [22].

With the increasing depth and complexity of neural networks, it is very important to develop SED systems with high performance, a low parameter count, generality and computational efficiency. We propose the multiscale time-frequency convolutional recurrent neural network (MTF-CRNN) for sound event detection, which uses multiscale time-frequency convolutional kernels to extract audio features from both the time and frequency domains. We utilize a gated recurrent unit (GRU) to obtain context-related short-term and long-term sequence features of sound events. We shared all the PyTorch code for everyone to learn and improve from each other, and our contributions are summarized as follows.

1. Inspired by Inception-v3 [23], we propose a novel multi-scale network structure for SED. The proposed method combines multiple convolution kernels with different sizes to capture useful information in detail from both the fine-grained and coarse-grained features with different time resolutions [24], [25].

2. While being effective and competitive on DCASE2017 and DCASE2019 SED tasks, the proposed MTF-CRNN has significantly less amount of parameters than other networks, which can reduce the computing resource requirements and shorten the inference time.

3. MTF-CRNN is a flexible and adaptive single model framework. One can adjust the number of filter groups to balance between the performance and the number of parameters.

The organizational structure of this article is as follows. Section 2 introduces the related work. Section 3 describes the proposed method. The detailed performance evaluation of the experiments and the results analysis are presented in Section 4. Section 5 draws the conclusion and discusses future work.

II. RELATED WORK

A. MULTISCALE TIME-FREQUENCY TRANSFORM

Multiscale techniques are popular in many fields, where they are used to solve problems that have intrinsic features in multiple scales. By using multiscale filters, similar to those used to form a corresponding virtual sensor, the network is transformed into a multisensor. The multiscale filter is a time-frequency local transform that can extract information of features from signals effectively and efficiently [26]–[28]. We apply the multiscale technique to extract intrinsic features from both time-domain and frequency-domain features of the sound event.

B. CRNN

The convolutional recurrent neural network (CRNN) is a combination of a CNN and RNN that has the advantages of both the CNN and RNN. The CNN includes the convolution layer, pooling layer and activation function, which are able to extract higher-level features that are invariant to local temporal and spectral variations. RNNs are very convenient for extracting sequence features and have become one of the most important neural networks [29]. A CRNN has some unique advantages over traditional neural network models, especially for sequence-like objects, because there is no restriction on the length of the class sequence object [30].

III. PROPOSED METHOD

Multiscale approaches have recently made great progress in object detection and image and audio recognition fields [25]. We simplify the network parameters, expand the multiscale method, and propose the multiscale time-frequency convolutional recurrent neural network (MTF-CRNN) to detect sound events. The MTF-CRNN is a single model without pre-training that utilizes four groups of parallel CNNs with different time-frequency convolutional kernels to extract more useful information from the input log Mel spectrum. The MTF-CRNN improves the time-frequency feature extraction capability with low network parameter counts [31]–[33]. Our method is used to detect various types of rare audio events with different durations and event-to-background ratios (EBRs). Moreover, we can extend our method to detect polyphonic sound events with few data [34].

A. INPUT AUDIO FEATURES

We use the log Mel spectrum as the input raw feature for the model. The spectrum operates on audio signals sampled at 44.1 kHz. The feature extraction consists of pre-emphasizing, frame segmentation, windowing, the fast Fourier transform (FFT), Mel filtering, and logarithmic computing. The pre-emphasized factor is 0.97, and a Hamming window with a length of 40 ms and a shift of 20 ms is applied. The FFT is performed to obtain the amplitude spectrum. Then, we apply 128 Mel filter banks and logarithmic
comparing to obtain the log Mel spectrum [15], [20]. Finally, we obtain time-frequency features with a size of $(T \times 128)$ for each sound clip, where $T$ is the number of frames.

B. NETWORK ARCHITECTURE

To enhance the robustness of the network and its ability to perceive all types of sound events, the SED method based on multiscale time-frequency is proposed. It is mainly composed of a multiscale time-frequency convolutional neural network (MTF-CNN), a GRU, and a fully connected output layer. The MTF-CNN focuses on the temporal and frequential properties from the input audio features to extract meaningful information. Inspired by Inception-v3, which employs a 42-layer deep network to classify images, MTF-CNN is simple by using four groups of three-layer convolution kernels to detect sound events. The GRU has been proven to be a powerful method for identifying the context sequential information. The FC outputs the probability values of the presence of the target sound event for each time frame. Figure 1 displays the architecture of our network.

1) MTF-CNN

The MTF-CNN is inspired by Inception-v3, which replaces $n \times n$ convolution by a $1 \times n$ convolution followed by a $n \times 1$ convolution, and the computational cost saving increases dramatically [23]. We utilize four groups of parallel convolutional neural networks to extract more features from acoustic samples. Even the same type of event may have different durations. The components and frequency distributions of the sound events are different as well. Joint convolution of the time domain and frequency domain is adopted to enhance the detection effectiveness. The frequency characteristics are obtained by convolution in the frequency domain. The time-domain features are obtained by convolution in the time domain. The first group of parallel CNN layers is realized by a $1 \times 1$ convolutional layer followed by batch normalization (BN) [35] and rectified linear units (ReLUs) [36] for activation. The other three groups are composed of three CNN layers, which include $1 \times 1$, $1 \times n$ ($n = 3, 5, 7$) for obtaining the frequency-domain features, and $n$ ($n = 3, 5, 7$) for obtaining the time-domain features. Batch normalization (BN) and rectified linear units (ReLUs) are applied after each CNN layer. The last filter size of the convolutional layer is set to 32. We concatenate the outputs of four groups of parallel CNNs together from 1 dimension. Finally, max-pooling with the size of 128 is applied to extract the representative value. The MTF-CNN network parameter counts are shown in Table 1, where $T$ is the number of frames. According to the experiments and referring to other papers [11], 32 filters are applied for each group of parallel CNNs, which is also convenient for the output of 128 features. To introduce more parameter counts to improve the sound event detection ability, we use 64 filters to obtain more detailed features in the hidden layer, which achieves better detection than that using 32 filters, as confirmed by our experiments.

| Name    | Filter Size | Filter Number | Pad   | Activation       | Output Size                  |
|---------|-------------|---------------|-------|------------------|------------------------------|
| CNN1    | $1 \times 1$ | 32            | (0,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN2-1  | $1 \times 1$ | 32            | (0,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN2-2  | $1 \times 3$ | 64            | (0,1) | BN+ReLU          | $64 \times T \times 128$     |
| CNN2-3  | $3 \times 1$ | 32            | (1,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN3-1  | $1 \times 1$ | 32            | (0,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN3-2  | $1 \times 5$ | 64            | (0,2) | BN+ReLU          | $64 \times T \times 128$     |
| CNN3-3  | $5 \times 1$ | 32            | (2,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN4-1  | $1 \times 1$ | 32            | (0,0) | BN+ReLU          | $32 \times T \times 128$     |
| CNN4-2  | $1 \times 7$ | 64            | (0,3) | BN+ReLU          | $64 \times T \times 128$     |
| CNN4-3  | $7 \times 1$ | 32            | (3,0) | BN+ReLU          | $32 \times T \times 128$     |
2) GATED RECURRENT UNIT
The GRU is a common RNN that can effectively capture the context characteristics of a time series [37]. The GRU consists of a reset gate and an update gate. The reset gate determines how to combine new input information with the information stored in the memory. The update gate defines the amount of previous memory saved to the current time step, saves information in long-term sequences, and is not cleared over time or removed because it is irrelevant to the prediction [38].

We use a two-layer bidirectional GRU recurrent network [39] containing 64 hidden units to extract sequence information. In the training process, a discard rate of 20% is adopted to prevent overfitting. Next, the features of the GRU layer are fed to the full connection layer, and batch normalization is then carried out. A ReLU is applied as an activation function.

3) FULLY CONNECTED LAYER
The output features from the GRU are fed into a fully connected layer (FC) that consists of two layers. The input layer contains 128 units. A ReLU activation function is followed. The output layer contains N units, where N is the number of the categories of target audio events to be detected. A final sigmoid activation function is used to output the target audio event probability for each time frame.

A postprocessing method that includes a sliding window, median filtering, and finding continuous regions is applied in the testing stage. To find the presence of a sound event, the threshold value (0.6/0.5/0.4 for ‘baby crying’/‘glass breaking’/‘gun shot’) according to the experiment is used. The starting position of the longest continuous region is set as the onset, and the ending position is set as the offset.

IV. PERFORMANCE EVALUATION
To evaluate the proposed method, we performed experiments on the TUT Rare Sound Events 2017 dataset [40] and DCASE2019 TAU Spatial Sound Events 2019 dataset [41], which are often used for sound event detection.

A. EVALUATION METRIC
We use the standard sound event detection segment-based metrics, F1 and error rate (ER) to evaluate our method. We compare the system output and reference in one-second segments [42]. The F1 is calculated as shown in Eq.(1).

\[
F1 = \frac{2 \sum_{t=1}^{T} TP(t)}{(2 \sum_{t=1}^{T} TP(t) + \sum_{t=1}^{T} FP(t) + \sum_{t=1}^{T} FN(t))}
\]

where \(TP(t)\) (true positives) is the total number of sound events that are active in both the reference and system output for the t-th one-second segment. The number of \(FP(t)\) (false positives) denotes the number of sound events that are active in the prediction but are inactive in the reference. \(FN(t)\) (false negatives) is the number of sound events inactive in the predictions but active in the reference.

The error rate \((ER)\) is calculated by integrating the number of errors in terms substitutions \((S)\), deletions \((D)\), and insertions \((I)\) over the total number of segments \(t\), with \(N(t)\) as the number of sound events labeled as active in the reference in segment \(t\). \(ER\) is expressed by Eq.(2), where \(S(t)\) is the number of reference events for which a correct event was not output, \(D(t)\) is the number of reference events that were not correctly identified, and \(I(t)\) is the number of events in the system output that are not correct.

\[
ER = \frac{(\sum_{t=1}^{T} S(t) + \sum_{t=1}^{T} D(t) + \sum_{t=1}^{T} I(t))}{\sum_{t=1}^{T} N(t)}
\]

B. PROCEDURE AND CONFIGURATION
The test system was implemented using PyTorch. The network was trained using the Adam algorithm for gradient-based optimization [43]. The log Mel spectral features were fed into a four-dimensional matrix. Training was performed for a maximum of 300 epochs using a learning rate of 0.001. We utilize early stopping criteria, and a patience of 10 epochs. We split 20% of the training subset as the validation subset. After each epoch of training, we use the validation subset to evaluate the performance and save the model with the smallest cost function as the best model. If the cost function does not decrease within 10 consecutive epochs, we stop the training. We run the procedure five times, and provide the mean and the standard deviation of these experiments. An implementation of our method can be found at https://github.com/zhang201882/MTF-CRNN.

C. DATASET
The IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) is jointly organized by Tampere University of Technology (TUT), Carnegie Mellon University (CMU), and Institut national de recherche en informatique et automatique (Inria). We evaluate the proposed method on the DCASE2017 TUT Rare Sound Events 2017 datasets [40] and the DCASE2019 TAU Spatial Sound Events 2019-Ambisonic datasets [41].

1) TUT RARE SOUND EVENTS 2017
TUT Rare Sound Events 2017 is used for DCASE2017 task2, which consists of isolated rare sound events for each target class and recordings of everyday acoustic scenes to serve as background. The background recordings that originate from the TUT Acoustic Scenes 2016 development dataset [44] are from 15 different acoustic scenes. The rare sound events are as follows: baby crying (BC), glass breaking (GB), and gun shot (GS) [40]. TUT Rare Sound Events 2017 consists of two datasets: a development dataset and an evaluation dataset.

Development dataset: The development dataset consists of the training subset and the testing subset (T2-dev).

Training subset: We used the development dataset to generate the training subset, which obtained 15000 audio clips (5000 per event class). The duration of the target sound events varied from 0.26 to 5.1 s.

Testing subset (T2-dev): We used the development dataset to generate the testing subset with 1500 audio clips.
Evaluation dataset: We applied the evaluation dataset to synthesize the evaluation subset (T2-eval) with 1500 audio clips (500 per event class). T2-eval data were synthesized using the original audio material that was not included in the development dataset. The duration of the target sound events varied from 0.25 to 4.82 s.

The examples of BC, GB and GS sound events in the time domain are shown in Fig. 2 and the TF domain is shown in Fig. 3. It is very clear that the three classes of target sound events have different spectrograms.

The mixtures were synthesized by summing the background samples with the corresponding isolated rare sound events that are from the DCASE2017 Task2-TUT Rare Sound Events 2017 dataset and were randomly selected. The synthesized controlling parameters include the event-to-background ratio (EBR) of −6, 0, and 6 dB and the randomly selected position for the target sound and event occurrence probability of 0.5 (there may be one or no events per mixture). Under the condition of different EBRs, the sound event waveforms are obviously different. As shown in Fig. 4, there are obvious distinctions between the three classes of sound events GS (EBR = 0 dB), GB (EBR = 6 dB) and BC (EBR = −6 dB) with different EBRs. Moreover, when the EBR is equal to zero or 6 dB, the events and background show a significant difference, but when the EBR is −6 dB, distinguishing the sound events and background is hard. Our goal is to recognize and detect the target sound events that have different lengths and the different EBRs.

2) TAU SPATIAL SOUND EVENTS 2019
We evaluate the proposed method on the DCASE2019 Task 3 dataset, the TAU Spatial Sound Events 2019-Ambisonic (FOA), which provides four-channel first-order ambisonic recordings. The dataset consists of a development dataset (T3-dev) and an evaluation dataset (T3-eval). The development dataset consists of 400, 60-second long recordings sampled at 48000 Hz and divided into four cross-validation splits of 100 recordings each. The evaluation dataset consists of 100 60-second recordings. The recordings in each split comprise 10 recordings that have either polyphonic sound (up to two) or no temporally overlapping sound events; they were synthesized using the spatial room impulse response from the five locations convolved with 11 classes sound event clips [41].

D. EXPERIMENTS ON TUT RARE SOUND EVENTS 2017
1) BASELINE
The baseline system provided a comparison point for participants as they developed their systems for the DCASE2017 Challenge. The baseline was based on the multilayer perception architecture (MLP) and used log Mel-band energies as features. The feature vector was calculated in frames of 40 ms with a 50% overlap; furthermore,
40 Mel bands covering the frequency range 0 to 22050 Hz were used. The feature vector was constructed using a 5-frame context, resulting in a feature vector length of 200. The MLP consists of two dense layers of 50 hidden units, each with a 20% dropout. The network is trained using the Adam algorithm for gradient-based optimization, and training is performed for a maximum of 200 epochs using a learning rate of 0.001 [44].
2) TEST RESULTS

a: PERFORMANCE COMPARISON WITH THE SINGLE-SCALE TIME-FREQUENCY CRNN

To investigate the performance of different single-scale models, we experimented with four different single-scale time-frequency CRNN (STF-CRNN). Table 2 shows the results of the performance comparison of the MT-CRNN with some STF-CRNN on the TUT Rare Sound Events 2017 development dataset (T2-dev), where single-scale time-frequency (STF) indicates only one group of time-frequency CNNs is used in the model. Four different STF-CRNNS were tested, namely, the 1×1 CRNN, the 3×3 CRNN, the 5×5 CRNN and the 7×7 CRNN. From the results, we can see that the 3×3 CRNN performs better than the other STF-CRNNS; the reason might be because its scale is appropriate for the dataset. We know from the results that the MT-CRNN is the best model. The average F1 was 5% higher than that of the best STF-CRNN and 24% higher than that of the worst STF-CRNN. At the same time, the average ER was 38% lower than that of the best STF-CRNN and 78% lower than that of the worst STF-CRNN.

| Method     | Metric | BC     | GB     | GS     | Average |
|------------|--------|--------|--------|--------|---------|
| 1×1CRNN    | F1     | 0.703±0.06 | 0.879±0.07 | 0.691±0.04 | 0.758±0.06 |
|            | ER     | 0.46±0.02 | 0.13±0.03 | 0.61±0.11 | 0.41±0.05 |
| 3×3CRNN    | F1     | 0.884±0.05 | 0.922±0.06 | 0.879±0.03 | 0.895±0.05 |
|            | ER     | 0.16±0.03 | 0.06±0.02 | 0.21±0.03 | 0.14±0.03 |
| 5×5CRNN    | F1     | 0.881±0.04 | 0.914±0.05 | 0.791±0.05 | 0.862±0.05 |
|            | ER     | 0.18±0.02 | 0.08±0.01 | 0.32±0.06 | 0.19±0.03 |
| 7×7CRNN    | F1     | 0.871±0.04 | 0.906±0.06 | 0.782±0.05 | 0.833±0.05 |
|            | ER     | 0.21±0.03 | 0.09±0.02 | 0.37±0.04 | 0.22±0.03 |
| Proposed   | F1     | 0.918±0.02 | 0.976±0.01 | 0.933±0.02 | 0.942±0.02 |
|            | ER     | 0.13±0.02 | 0.04±0.01 | 0.11±0.01 | 0.09±0.01 |

b: PERFORMANCE COMPARISON OF DIFFERENT MULTISCALE METHODS

To obtain the best multiscale model, we carried out experiments on four different multiscale models including the MT-CRNN.

(1) Multiscale frequency-domain convolutional recurrent neural network (MF-CRNN): The MF-CRNN only uses a multiscale CNN with a kernel of 1×1 and 1×n (n = 3, 5, 7), which extracts the frequency domain features using the multiscale CNN.

(2) Multiscale time-domain convolutional recurrent neural network (MT-CRNN): The MT-CRNN only uses a multiscale CNN with a kernel of 1×1 and n×1 (n = 3, 5, 7), which extracts the time-domain features using a multiscale CNN.

(3) Simultaneous convolution of the multiscale time-frequency CRNN (SMTF-CRNN): The SMTF-CRNN is similar to the MTF-CRNN in the net architecture. Four sets of multiscale time-frequency CNNs are used, but a convolution kernel of n×n (n = 1, 3, 5, 7) is used to extract the time-frequency features simultaneously.

Table 3 shows the performance of the different multiscale methods on the T2-dev and T2-eval datasets. SMTF-CRNN achieved the best ER in GB event detection on the T2-eval dataset, which showed that the 1×n and n×1 architecture does not exceed n×n in all audio event detections. A n×n convolutional layer can be replaced by two layers of 1×n and n×1 kernels to reduce the parameter counts, increase the nonlinearity, and improve the performance. However, applying the transformation directly to the kernels results in significant information loss [23]. The MTF-CRNN method achieved the best results for all other event detections on the development dataset and the evaluation dataset on the whole. As is well known, various sound events have different spectral characteristics along different time frames. The MTF-CRNN is better than other models, as shown in Table 3, because it extracts not only the important characteristics of the frequency domain but also the audio features of different time-domain frames. This result is the same as with the Inception-v3 architecture, and using a n×1 convolution followed by a 1×n convolution is slightly better than a n×n convolution. Compared with the MT-CRNN and MF-CRNN, our recommended method (MTF-CRNN) increases parameter counts by 16%, and achieves average ER reduction of at least 57% and 45% on the T2-dev and T2-eval datasets, respectively.

c: PERFORMANCE COMPARISON OF DIFFERENT GROUPS OF MULTISCALE CONVOLUTIONAL KERNELS

In Table 4 and Table 5, we list the F1 and ER of the method that uses 2-6 groups of multiscale convolutional kernels (MCK) on the T2-dev and T2-eval datasets for three target sound events, respectively. The Parameters of the 2-6 groups of MCK with different network architecture are shown in Table 6. In the table, the method 2(3,4,5,6) groups of MCK means that different models use 2-6 groups of multiscale convolutional kernels. From the results, we can see that with the increase in the number of groups, the performance improved, but when the number of groups ascended to a certain extent, the performance began to decline. We propose that the MTF-CRNN that utilized four groups of multiscale convolutional kernels achieved the best results for detecting BC, GB and GS. The following 4 groups of MCK by default is the MTF-CRNN method that we recommended.

d: PERFORMANCE COMPARISON OF OTHER METHODS

Table 7 shows the F1 and ER of the MTF-CRNN and other SED methods on the T2-dev and T2-eval datasets. We compare MTF-CRNN with the following methods:

- Baseline: The baseline method is based on a multiplayer perceptron architecture (MLP) and uses log mel-band energies as features [44].
- DNN/CNN: The DNN/CNN coupled CNNs and DNNs with novel weighted and multi-task loss function for audio event detection [10].
TABLE 3. The performance comparison of different multiscale methods on the T2-dev dataset.

| Method     | parameter counts | Metric | BC   | GB   | T2-dev | Average | BC   | GB   | T2-eval | Average |
|------------|------------------|--------|------|------|--------|---------|------|------|---------|---------|
| MF-CRNN    | 197K             | F1     | 0.856±0.04 | 0.902±0.06 | 0.805±0.04 | 0.854±0.05 | 0.852±0.04 | 0.862±0.06 | 0.65±0.04 | 0.788±0.05 |
|            |                  | ER     | 0.21±0.02 | 0.11±0.03 | 0.32±0.03 | 0.21±0.03 | 0.29±0.02 | 0.19±0.03 | 0.46±0.04 | 0.31±0.03 |
| MT-CRNN    | 197K             | F1     | 0.854±0.06 | 0.906±0.05 | 0.873±0.05 | 0.887±0.05 | 0.873±0.03 | 0.845±0.06 | 0.668±0.04 | 0.795±0.04 |
|            |                  | ER     | 0.2±0.03 | 0.09±0.01 | 0.34±0.05 | 0.21±0.03 | 0.22±0.02 | 0.22±0.03 | 0.54±0.04 | 0.32±0.03 |
| SMTF-CRNN  | 336K             | F1     | 0.83±0.06 | 0.93±0.05 | 0.803±0.06 | 0.857±0.06 | 0.869±0.03 | 0.92±0.05 | 0.79±0.04 | 0.86±0.04 |
|            |                  | ER     | 0.24±0.02 | 0.04±0.01 | 0.3±0.03 | 0.19±0.02 | 0.23±0.03 | 0.07±0.02 | 0.37±0.03 | 0.22±0.03 |
| Recommended| 228K             | F1     | 0.918±0.02 | 0.976±0.01 | 0.933±0.02 | 0.942±0.02 | 0.897±0.03 | 0.951±0.02 | 0.839±0.02 | 0.892±0.02 |
|            |                  | ER     | 0.13±0.02 | 0.04±0.01 | 0.11±0.01 | 0.09±0.01 | 0.15±0.01 | 0.08±0.02 | 0.28±0.01 | 0.17±0.01 |

TABLE 4. The performance comparison of different groups of MCK on the T2-dev dataset.

| Method     | Parameter counts | Metric | BC   | GB   | T2-dev | Average | BC   | GB   | T2-eval | Average |
|------------|------------------|--------|------|------|--------|---------|------|------|---------|---------|
| 2 groups of MCK |                    | F1     | 0.865±0.06 | 0.910±0.07 | 0.838±0.05 | 0.871±0.06 | 0.19±0.03 | 0.06±0.02 | 0.26±0.03 | 0.17±0.03 |
| 3 groups of MCK |                    | F1     | 0.904±0.03 | 0.926±0.05 | 0.888±0.02 | 0.906±0.03 | 0.16±0.02 | 0.06±0.01 | 0.21±0.03 | 0.14±0.02 |
| 4 groups of MCK (Recommended) |                | F1     | 0.918±0.02 | 0.976±0.01 | 0.933±0.02 | 0.942±0.02 | 0.13±0.02 | 0.04±0.01 | 0.11±0.01 | 0.09±0.01 |
| 5 groups of MCK |                    | F1     | 0.911±0.03 | 0.921±0.07 | 0.891±0.03 | 0.908±0.04 | 0.11±0.01 | 0.05±0.02 | 0.15±0.03 | 0.11±0.02 |
| 6 groups of MCK |                    | F1     | 0.881±0.03 | 0.932±0.06 | 0.871±0.03 | 0.895±0.04 | 0.19±0.01 | 0.05±0.03 | 0.18±0.03 | 0.14±0.02 |

TABLE 5. The performance comparison of different groups of MCK on the T2-eval dataset.

| Method     | Parameter counts | Metric | BC   | GB   | T2-eval | Average | BC   | GB   | T2-eval | Average |
|------------|------------------|--------|------|------|---------|---------|------|------|---------|---------|
| 2 groups of MCK |                    | F1     | 0.832±0.06 | 0.904±0.05 | 0.802±0.04 | 0.846±0.05 | 0.25±0.03 | 0.11±0.02 | 0.31±0.02 | 0.22±0.02 |
| 3 groups of MCK |                    | F1     | 0.865±0.04 | 0.889±0.06 | 0.803±0.05 | 0.853±0.05 | 0.23±0.04 | 0.11±0.01 | 0.32±0.04 | 0.22±0.03 |
| 4 groups of MCK (Recommended) |               | F1     | 0.897±0.03 | 0.951±0.02 | 0.839±0.02 | 0.892±0.02 | 0.15±0.01 | 0.08±0.02 | 0.28±0.01 | 0.17±0.01 |
| 5 groups of MCK |                    | F1     | 0.848±0.05 | 0.921±0.05 | 0.808±0.06 | 0.859±0.05 | 0.23±0.03 | 0.07±0.02 | 0.29±0.05 | 0.20±0.03 |
| 6 groups of MCK |                    | F1     | 0.827±0.04 | 0.897±0.06 | 0.785±0.03 | 0.836±0.04 | 0.29±0.02 | 0.11±0.03 | 0.35±0.02 | 0.25±0.02 |

- **R-FCN**: A new audio event detection and classification approach based on R-FCN—a state-of-the-art fully convolutional network framework for visual object detection [18].
- **CRNN**: Çakir et al. combined the CNN and RNN for rare sound event detection [11].
- **R-CRNN**: The R-CRNN is a Region-based CRNN to detect audio event and achieved the best performing single-model method among all methods without using ensembles on DCASE2017 dataset at that time [19].

Compared with other methods in Table 7, the proposed MTF-CRNN achieved the best results on the whole. The proposed method achieved an average F1 of 0.942±0.02, which is better than the baseline [44] by 30%, and an average ER of 0.09±0.01 which obtained an improvement of 83% compared with the baseline [44] on the T2-dev dataset.

On the T2-eval dataset, the MTF-CRNN obtained a F1 of 0.892±0.02, which surpassed the baseline by 39%, and an ER of 0.17±0.01 which outperformed the baseline by a relative 73%. We believe that the reason for providing better performance is related to the fact that MTF-CNN achieves time-frequency domain discriminative features simultaneously by using four groups of parallel convolution kernels with appropriate sizes and that the GRU extracts the global and local sequence information. Competitive performance also depends on the low number of network parameters compared to that of other methods, which makes the neural network easy to train and reduces overfitting.

Compared with other state-of-the-art methods, the performance of our model is competitive. Table 8 and Table 9 show the F1 and ER of the 1D-CRNN [15], TFA [21], MTFA [22], and MTF-CRNN on the T2-dev and T2-eval datasets for three target sound events, respectively. The 1D-CRNN [15] applies a 1-dimensional convolution layer, batch normalization (BN) and a pooling layer, followed by RNN layers and a fully connected layer. The 1D-CRNN achieved the best result by changing the set of mixtures.
and different timesteps. The number of parameters of 1D-CRNN ensemble model is about 2 to 6 times that of one single model 294K, which is far greater than that of our proposed MTF-CRNN. Compared with 1D-CRNN ensemble 2-6 relatively high-performance single models for each specific rare audio event, our method is a general simple single model applicable to all sound events, and can be conveniently applied to detection and recognition of all sound events.

TFA [21] presented a temporal-frequential attention model for sound event detection, which is an innovative pretrained method with about 260K parameter counts for 3×3, 5×5, and 7×7 convolution kernels that are used. MTFA [22] used a multiscale hourglass structure that outperformed previous single-model methods for rare sound event detection, with a performance that depends on the quantity of annotated data. Our model performs worse than other state-of-the-art methods but has the fewest parameter counts. As shown in Table 10, the parameter counts of our recommended model are compared with different models. In Table 10, the number 2(3,4,5,6) indicates different models using 2-6 groups of multiscale convolutional kernels (MCK). From the results, we can see that with the increase in the number of groups, the count of parameters increases correspondingly. Compared with 3 groups of MCK results in Table 4, Table 5, and Table 10, our recommended configuration (4 groups of MCK) increases the number of parameters by 31%, and achieves average ER reduction of 38% and 23% on the T2-dev and T2-eval datasets, respectively. Although MT-CRNN, MF-CRNN, and 2-3 groups of MCK has fewer parameter counts, less running time and relatively faster speed, as long as it does not put forward explicit requirements for running time in mobile, portable or real-time, the model with lower ER and higher F1 should be selected as far as possible, the 4 groups of MCK is our recommended model. The parameter counts of the proposed MTF-CRNN using four groups of multiscale convolutional kernels is approximately 228K, which is only one-fifth that of the MTFA parameter counts. The number of parameters of our method is far less than that of the R-CRNN and MTFA and less than that of TFA and the number of corresponding parameter counts of the CRNN [11] for detecting BC and GB. We use the general Python function numel() to calculate model parameter counts.

The parameter reduction is not a simple linear relationship with running time, but it is affected by many factors. We can’t accurately determine the comparison of their runtime according to the parameters of the proposed model and those of other methods. According to the proposed method and different parameter setups, we can see the influence of different parameter changes on runtime. A model with low parameter counts could be much faster than another one [13], [45], [46]. The comparison of the parameter counts, training and inference times of 2-5 groups MCK models are shown in Table 11. Compared with 2 groups of MCK, our recommended configuration (4 groups of MCK) increases the number of parameters by 62%, and achieves 33% inference times increased on the T2-dev dataset (in 4 vs. 3 minutes). Compared with 5 groups of MCK, our recommended configuration (4 groups of MCK) reduces the number of parameters by 8%, and achieves 12% training time increased on the T2-dev dataset (in 2381 vs. 2697 minutes). The proposed MTF-CRNN has
Table 11. The comparison of the parameter counts and running times of different models.

| Model                     | Parameter counts | Training times(min) | Inference times(min) |
|---------------------------|------------------|---------------------|----------------------|
| 2 groups of MCK           | 141K             | 1557                | 3                    |
| 3 groups of MCK           | 174K             | 1666                | 4                    |
| 4 groups of MCK           | 228K             | 2381                | 4                    |
| (Recommended)             |                  |                     |                      |
| 5 groups of MCK           | 249K             | 2697                | 4                    |

significantly less amount of parameters than other networks, which can reduce the computing resource requirements and shorten the training and inference times in a certain extent.

E. EXPERIMENTS ON TAU SPATIAL SOUND EVENTS 2019

1) BASELINE

Sound event detection is the first step of the DCASE2019 Task3. The baseline system provided a comparison point for participants as they developed their systems for the DCASE2019 Task3 Challenge. The baseline was based on the CRNN and used the magnitude of the short-time Fourier transform as the input feature. The CRNN consists of multiple layers of a 2D CNN and a bidirectional GRU. A binary cross-entropy loss is performed for a maximum of 1000 epochs. The network is trained using the Adam optimizer with default parameters. Early stopping is used to control the network from overfitting [41], [47].

2) TEST RESULTS

The performance comparison between MTF-CRNN and other methods on the T3-dev and the T3-eval datasets (the TAU Spatial Sound Events 2019-Ambisonic dataset) is presented in Table 12. The methods include:

- **Baseline**: The baseline method employing a CRNN is used to generate benchmark scores on the DCASE2019 Task3 dataset [41].
- **CE-CRNN**: Kapka et al. ensemble four CRNN single models which run in a consecutive manner to recover all possible informations about occurring events. The method achieves first place on the DCASE2019 Task3 challenge [48].
- **Two-stage**: A two-stage polyphonic sound event detection and localization method which obtained second place on the DCASE2019 Task3 challenge [49].
- **MTFA**: MTF-CRNN achieved better performance than other methods with more parameters and does not achieve good performance on datasets with few data. The MTFA is a relatively complex method with more parameters and does not achieve good performance on datasets with few data. The MTFA and MTF-CRNN achieved better performance than other methods on the DCASE2019 dataset, which shows that the multiscale time-frequency method is good at rare sound event detection.

We proposed a multiscale time-frequency convolutional recurrent neural network (MTF-CRNN) for sound event detection that achieved competitive performance with low parameter counts. We evaluated our model on both the DCASE2017 Task2 dataset and DCASE2019 Task3 dataset. The results showed that MTF-CRNN has the ability to adapt to different datasets. T2-dev and T2-eval are based on the DCASE2017 task2 dataset, which were synthesized with the duration of the target sound events varying from 0.24 to 5.1 s and the event-to-background ratio (EBR) of −6 dB, 0 dB, and 6 dB. The experiments prove that MTF-CRNN can detect rare sound events with variable dura-

Table 12. The performance comparison of different methods on the T3-dev and T3-eval datasets.

| Method        | Parameter counts | T3-dev ER | T3-dev F1 | T3-eval ER | T3-eval F1 |
|---------------|------------------|-----------|-----------|------------|------------|
| Baseline[41]  | *                | 0.34      | 0.799     | 0.28       | 0.854      |
| CE-CRNN[48]   | 2M               | 0.14      | 0.893     | 0.08       | 0.947      |
| Two-stage[49] | 6M               | 0.13      | 0.938     | 0.08       | 0.955      |
| MTFA[22]      | 1221K            | 0.36±0.03 | 0.792±0.01| 0.23±0.03  | 0.874±0.02 |
| Proposed      | 228K             | 0.17±0.01 | 0.899±0.01| 0.11±0.01  | 0.936±0.01 |

V. DISCUSSION AND CONCLUSION

Through the experiment, we proved that the performance of the MTF-CRNN is based on a single model with lower parameter counts than the CRNN [11], R-CRNN [19] and MTFA [22]. We first compared the performance of the MTF-CRNN to some single-scale methods on the T2-dev dataset. Second, we achieved the best performance compared with the method that applied the multiscale frequency-domain convolution (MF-CRNN), the multiscale time-domain convolution (MT-CRNN) and multiscale simultaneous convolution of the multiscale time-frequency CRNN (SMTF-CRNN) on both the T2-dev and T2-eval datasets. Third, we performed 2-6 groups of multiscale convolution on the T2-dev and T2-eval datasets, and our proposed MTF-CRNN is the best on average. Our performance is worse than that of MTFA [22] on the DCASE2017 Task2 dataset, but our method is superior on the DCASE2019 Task3 dataset. The reason may be that the MTFA is a relatively complex method with more parameters and does not achieve good performance on datasets with few data. The MTFA and MTF-CRNN achieved better performance than other methods on the DCASE2017 dataset, which shows that the multiscale time-frequency method is good at rare sound event detection.

We proposed a multiscale time-frequency convolutional recurrent neural network (MTF-CRNN) for sound event detection that achieved competitive performance with low parameter counts. We evaluated our model on both the DCASE2017 Task2 dataset and DCASE2019 Task3 dataset. The results showed that MTF-CRNN has the ability to adapt to different datasets. T2-dev and T2-eval are based on the DCASE2017 task2 dataset, which were synthesized with the duration of the target sound events varying from 0.24 to 5.1 s and the event-to-background ratio (EBR) of −6 dB, 0 dB, and 6 dB. The experiments prove that MTF-CRNN can detect rare sound events with variable dura-
tions and different audio backgrounds. Through the experiment, we showed that the MTF-CRNN is better than the MT-CRNN, MF-CRNN and SMTF-CRNN. Compared to that of other methods, the performance has been greatly improved as a single model with a low number of parameters. On the DCASE2019 Task3 dataset, our method’s performance is better than that of the MTFA and the baseline method [41]. The MTF-CRNN has better performance on the T3-eval than on the T3-dev dataset and exhibits good generalization capabilities of the systems. In the future, we will continue to test our method with more datasets, attempt more combinations of scale convolutional kernel functions, strive to improve the method’s performance, and translate our method into engineering practice. All of our code has been uploaded to GitHub, where it can be freely downloaded at https://github.com/zhang201882/MTF-CRNN.

REFERENCES

[1] X. Wu, H. Gong, P. Chen, Z. Zhi, and Y. Xu, “Intelligent household surveillance robot,” in Proc. IEEE Int. Conf. Robot. Biomimetics, Feb. 2009, pp. 1734–1739.

[2] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, “Audio surveillance of roads: A system for detecting anomalous sounds,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 1, pp. 279–288, Jan. 2016.

[3] T. Y. Zhu, Q. Wang, and M. D. Gao, “Application of outlier mining technology in traffic incident detection,” Comput. Sci. Explor., vol. 8, pp. 111–120, 2014.

[4] J. Schroder, S. Goetzte, V. Gretzmanncher, and J. Annemuller, “Automatic acoustic siren detection in traffic noise by part-based models,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., May 2013, pp. 493–497.

[5] X. F. Li, C. S. Yang, and S. C. Ding, “Clustering of real-time traffic incident detection in urban tunnels based on information entropy,” Comput. Technol. Develop., vol. 23, pp. 212–215, 2013.

[6] X. F. Zhang, Z. G. Guo, and S. Liu, “Self-similarity clustering event detection based on triggers guidance,” Comput. Sci., vol. 37, pp. 212–220, 2010.

[7] S. Li, Y. Gu, Y. Luo, J. Chambers, and W. Wang, “Enhanced streaming based subspace clustering applied to acoustic scene data clustering,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 11–15.

[8] A. Temko and C. Nadeu, “Classification of acoustic events using SVM-based clustering schemes,” Pattern Recognit., vol. 39, no. 4, pp. 682–694, Apr. 2006.

[9] J. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, “Robust sound event classification using deep neural networks,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 23, no. 3, pp. 540–552, Mar. 2015.

[10] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, “Weighted and multi-task loss for rare audio event detection,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Munich, Germany, Apr. 2018, pp. 336–340. [Online]. Available: http://dcase.community/challenge2017/task-rare-sound-event-detection-results#technical-reports.

[11] E. Cakir and T. Virtanen, “Convolutional recurrent neural networks for rare sound event detection,” in Proc. Detection Classification Acoustic Scenes Events (DCASE) Challenge, Munich, Germany, Nov. 2017. [Online]. Available: http://dcase.community/challenge2017/task-rare-sound-event-detection-results#technical-reports.

[12] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

[13] K. Drossos, I. S. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, “Sound event detection with depthwise separable and dilated convolutions,” 2020, arXiv:2002.00476. [Online]. Available: http://arxiv.org/abs/2002.00476.

[14] D. Stowell, D. Giannouls, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events,” IEEE Trans. Multimedia, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.
G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, “Minimal gated unit for recurrent neural networks,” Int. J. Autom. Comput., vol. 13, no. 3, pp. 226–234, Jun. 2016.

A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in Proc. 24th Eur. Signal Process. Conf. (EUSIPCO), Aug. 2016, pp. 1128–1132.

S. Adavanne, A. Politis, and T. Virtanen, “A multi-room reverberant dataset for sound event localization and detection,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), 2019. [Online]. Available: http://dcase.community/challenge2019/task-sound-event-localization-and-detection-results#technical-reports

Y.-Cao, T. Iqbal, Q. Kong, M. Galindo, W. Wang, and M. Plumbley, “Two-stage sound event localization and detection using intensity vector and generalized cross-correlation,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE) Challenge, 2019. [Online]. Available: http://dcase.community/challenge2019/task-sound-event-localization-and-detection-results#technical-reports

DCASE2019. Accessed: Jun. 30, 2019. [Online]. Available: http://dcase.community/challenge2019/task-sound-event-localization-and-detection-results

K. Zhang et al.: MTF-CRNN: Multiscale Time-Frequency Convolutional Recurrent Neural Network for Sound Event Detection

YUANWEN CAI received the B.S. and M.S. degrees in electrical engineering from the National University of Defense Technology, Changsha, China, in 1988 and 1990, respectively, and the Ph.D. degree in control science and technology from Beihang University, Beijing, China, in 2005. He is currently a Professor with the Department of Astronautics Science and Technology, Space Engineering University, Beijing. His current research interests include aerospace launch and measurement technology.

YUAN REN (Member, IEEE) received the Ph.D. degree in instrument science and technology from Beihang University, Beijing, China, in 2012. From 2016 to 2017, he was a Visiting Scholar with Duke University, Durham, NC, USA, working with co-advisor Prof. E. H. Dowell. He is currently an Associate Professor with Space Engineering University, Beijing, China. He has authored or coauthored more than 60 articles as a corresponding author and one monograph. He has been granted more than 30 Chinese invention patents as the first or the second inventor. His current research interests include spacecraft guidance, navigation and control technologies, and novel quantum sensing and control technology.

RUIDA YE received the B.S. degree from Hubei Polytechnic University, Huangshi, China. He is currently pursuing the M.S. degree in aerospace engineering with Space Engineering University, Beijing, China. His research interests include deep learning, acoustic event detection, and machine fault detection.

LIANG HE (Member, IEEE) received the B.S. degree in communication engineering from the Civil Aviation University of China, Tianjin, China, in 2004, the M.S. degree in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2006, and the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2011. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. His research interests are in the area of speech signal processing and artificial intelligence, primarily in speaker recognition, spoken language recognition, and acoustic event detection.

KEMING ZHANG received the B.S. degree in communication engineering from Information Engineering University, Zhengzhou, China, in 1997. He is currently pursuing the Ph.D. degree with Space Engineering University. He is a Senior Engineer specializing in communication and artificial intelligence.