

FULL-FREQUENCY DYNAMIC CONVOLUTION: A PHYSICAL FREQUENCY-DEPENDENT CONVOLUTION FOR SOUND EVENT DETECTION

Haobo Yue, Zhicheng Zhang[✉], Da Mu, Yonghao Dang, Jianqin Yin, Jin Tang

Beijing University of Posts and Telecommunications, Beijing, China
{hby, zczhang}@bupt.edu.cn

ABSTRACT

Recently, 2D convolution has been found unqualified in sound event detection (SED). It enforces translation equivariance on sound events along frequency axis, which is not a shift-invariant dimension. To address this issue, dynamic convolution is used to model the frequency dependency of sound events. In this paper, we proposed the first full-dynamic method named *full-frequency dynamic convolution* (FFDConv). FFDConv generates frequency kernels for every frequency band, which is designed directly in the structure for frequency-dependent modeling. It physically furnished 2D convolution with the capability of frequency-dependent modeling. FFDConv outperforms not only the baseline by 6.6% in DESED real validation dataset in terms of PSDS1, but outperforms the other full-dynamic methods. In addition, by visualizing features of sound events, we observed that FFDConv could effectively extract coherent features in specific frequency bands, consistent with the vocal continuity of sound events. This proves that FFDConv has great frequency-dependent perception ability. Code is available at [FFDConv](#).

Index Terms— sound event detection, full-frequency dynamic convolution, frequency-dependent modeling, independent representation spaces, vocal continuity

1. INTRODUCTION

Sound event detection (SED) is one of the subtasks of computational auditory scene analysis (CASA) [1], which helps machines understand the content of an audio scene. Similar to visual object detection [2] and segmentation [3], SED aims to detect sound events and corresponding timestamps (onset and offset), considered as a prior task of automatic speech recognition (ASR) and speaker verification. It has wide applications in information retrieval [4], smart homes [5], and smart cities [6].

SED has achieved great success with the help of deep learning (DL). The general paradigm is that acoustic spectral features are passed through a deep neural network and then transformed into discriminative acoustic representations to distinguish different sound events. Designing an effective feature extractor has become a hot topic in SED, which has

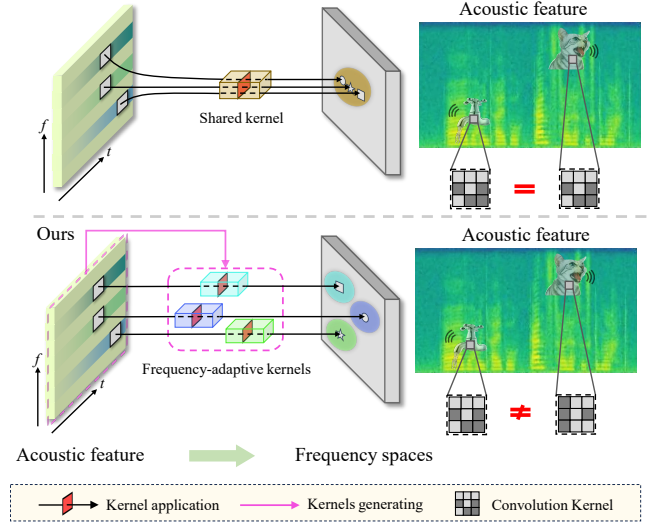


Fig. 1. Illustration of frequency-dependent modeling. Top models time-frequency patterns in the same space with a shared kernel. Bottom models them in several spaces with frequency-adaptive kernels, in which time-frequency patterns specific to sound events can be considered.

been adopting methods qualified in other domains in the past few years. Convolutional neural networks (CNN) from the field of computer vision, such as SENet [7], SKNet [8], and CBAM [9] have been migrated to SED in the spirit of acoustic spectrogram being similar to two-dimensional image data. With the intention that speech and audio are both sound data, Conformer [10] from the field of speech recognition has migrated to SED. However, they all failed to show good performance in SED. Specifically, SENet, SKNet, and CBAM are designed on image data with a clear 2D spatial concept, while audio data is a sequence. Conformer is designed on speech data containing only the speech sound event, meaning time-frequency patterns of speech data are distributed only in a certain fixed frequency band. However, audio data always contains multiple sound events, and so has diverse time-frequency patterns of sound events. All of the above emphasize that DL methods qualified in other domains may not nec-

essarily be compatible with SED.

Dynamic convolution network [11] was initially proposed for video prediction. It was designed to generate future frames based on the motion pattern within a particular video. The parameters of the dynamic convolution kernel are always adapted to the input. In SED, different sound events are distributed in different frequency regions, and this frequency dependence is invariant over time. This has motivated some researchers to investigate whether dynamic convolution can improve the capability of 2D convolution in modeling the frequency dependence of sound events. [12] proposed frequency dynamic convolution (FDConv), which found that the time-frequency spectrogram is not translation invariant on frequency dimension like image data. FDConv extracts frequency-adaptive attention weights from input for several pre-initialized convolution kernels. These kernels are then weightedly combined in the number dimension to obtain one convolution kernel. Then, the combined kernel is convoluted with the input in a standard manner. [12, 13], [14] proposed multi-dimensional frequency dynamic convolution (MFDCConv), which extends the frequency-adaptive dynamic properties of convolutional kernels to more dimensions of the kernel space, i.e. in-channels, out-channels, and kernel numbers.

Although FDConv and MFDCConv have achieved great performance, they are essentially the same as basic convolution, which is spatially shared. They belong to semi-dynamic convolution in the field of dynamic convolution. As shown in the upper part of Fig. 1, their perception abilities of different frequency bands are identical. They can only model time-frequency patterns in one representation space, where sound events are not easily recognized from each other. Compared with semi-dynamic convolution, full-dynamic convolution [11, 15–18] attracts more attention recently, which uses a separate network branch to predict a specific filter for each pixel. [18] found this type of dynamic convolution is equivalent to applying attention on unfolded input features, which enables it more effective when modeling complex patterns. Sound events’ time-frequency patterns are highly frequency-dependent, and full-dynamic convolution can model features of spatial pixels with different filters. Full-dynamic convolution may be optimal in dealing with recognizing sound events.

In this paper, we propose a novel method named *full-frequency dynamic convolution* (FFDCConv), which is the first full-dynamic convolution method for SED. As shown in the lower part of Fig. 1, FFDCConv generates frequency-specific kernels, resulting in distinct representation spaces. This design is applied directly in the structure for frequency-dependent modeling. In this way, the 2D convolution is physically furnished with the capability of frequency-dependent modeling, so that the specific time-frequency patterns can be acquired for different sound events. In the end, sound events can be easily recognized from each other in subsequent classification.

The main contributions of this paper are summarized as follows:

- We proposed full-frequency dynamic convolution that can model time-frequency patterns in independent representation spaces. This method will extract more discriminative features of sound events, resulting in effective classification.
- The Proposed method outperforms not only baseline but also pre-existing full dynamic filters method in other domain.
- By visualizing features of sound events, we found the ability to model temporally coherent features is essential to the detection of sound events. And the FFDCConv has this ability.

2. METHODOLOGY

2.1. Full-dynamic convolution

A basic 2D convolution can be denoted as $y = \mathbf{W} * x + \mathbf{b}$, where $x \in \mathbb{R}^{T \times F \times C_{in}}$ and $y \in \mathbb{R}^{T \times F \times C_{out}}$ denote the input feature and output feature; $\mathbf{W} \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$ and $\mathbf{b} \in \mathbb{R}^{C_{out}}$ denote the weight and bias of a basic convolution kernel. In contrast to basis convolution, full-dynamic convolution [11] leverages separate network branches to generate the filters for each pixel. Full-dynamic convolution operation can be written as:

$$\begin{aligned} y &= \text{Concat}(\mathbf{W}_{t,f} * x(t, f)) \\ \mathbf{W}_{t,f} &= G(x, t, f) \end{aligned} \quad (1)$$

where $\mathbf{W}_{t,f}$ denotes weight for the current pixel; The G is the filter generating function; *Concat* here aims to convey that convolution operation of each pixel is independent. For simplicity, the bias term is omitted.

2.2. Overall of proposed method

As is commonly understood, different sound events have different frequency band distributions. For instance, catcall, which is sharp, shrill, and high-pitched, is often heard in the high-frequency range; running water, which is low, soft, and soothing, is often heard in the low-frequency range. Based on this, we explore designing a new convolution for SED, which can capture the distribution of frequency bands and model time-frequency patterns of sound events in different frequency representation spaces.

Inspired by full dynamic convolution [18], we designed the full-frequency dynamic convolution (FFDCConv) for SED. Overall, as shown in Fig. 2, FFDCConv employs a separate branch to predict kernels for each frequency band, in which the content of kernels is based on input feature. In the kernel-generating branch, there are two sub-branches: the spatial

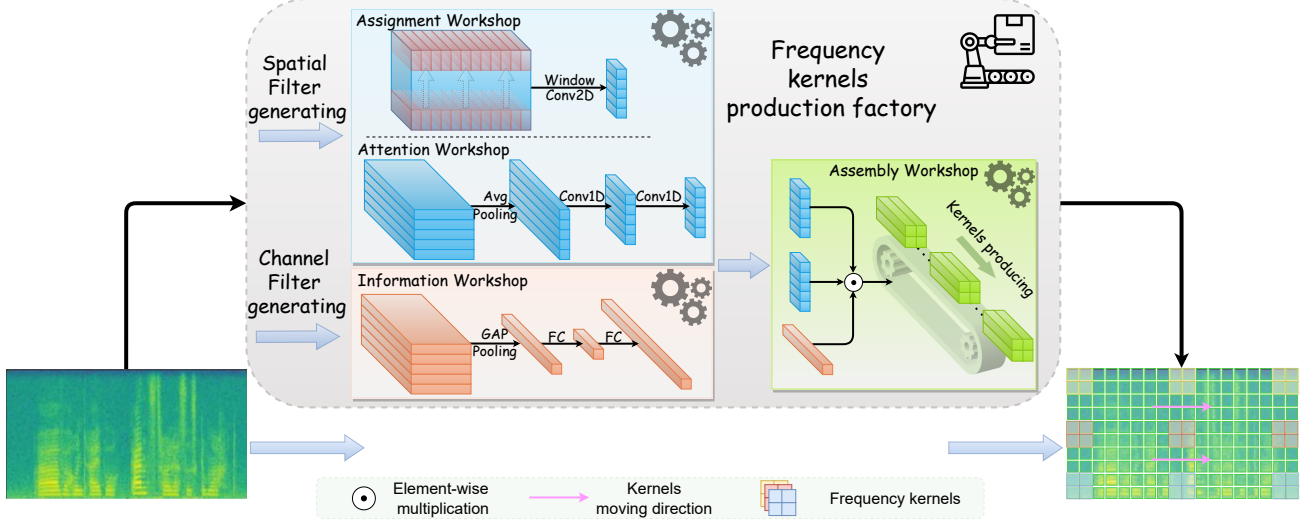


Fig. 2. Illustration of full-frequency dynamic convolution. In general, the factory produces frequency-dependent kernels from acoustic feature, and then kernels are convoluted with input along the time axis. In the factory, there are two workshops aiming to produce spatial filters and channel filters, respectively. And they are integrated in the assembly workshop.

filter-generating branch for the spatial space of kernels and the channel filter-generating branch for the channel space of kernels. After spatial and channel filters are obtained, they are combined and then convoluted with the input feature. Note that similarly, full-temporal dynamic convolution (FTDConv) predicts kernels for each temporal frame, and kernels are convoluted with input along the frequency axis.

2.3. Full-frequency dynamic convolution

Unlike the previous semi-dynamic convolution, FFDConv is designed directly in the structure for frequency-dependent modeling. It models the feature along the frequency axis in different representation spaces. Mathematically, FFDConv can be written as:

$$y = \text{Concat}(\mathbf{W}_f * x(f), \dim = f) \quad (2)$$

$$\mathbf{W}_f = G_s(x, f) \odot G_c(x, f)$$

where \mathbf{W}_f is the content-adaptive kernel for the f^{th} frequency band; $x(f) \in \mathbb{R}^T$ is the f^{th} frequency band of input feature; G_s and G_c are the spatial and channel filter-generating function; \odot denotes the elemental dot product operator. For clarity, *Concat* here aims to convey that \mathbf{W}_f is convoluted with input along the time axis.

FFDConv employs a separate branch to generate convolution kernels for each frequency band, in which there are two sub-branches: spatial filter-generating branch and channel filter-generating branch. The spatial filter-generating module is designed to predict the spatial content of dynamic kernels, and the channel-generating module is designed to predict the channel content of dynamic kernels. For efficiency, the dy-

namic filters are decoupled into spatial and channel ones, following [18].

Spatial filter generating. As illustrated in Fig. 3, we use a standard Conv2D to compress the time dimension of input and map channel dimension from C to K^2 , whose kernel weight $W \in \mathbb{R}^{C \times K^2 \times T \times W}$, where W is the window size of the kernel in the frequency dimension. It moves along the frequency axis when convoluted with input. In this way, not only are the adjacent frequency components considered, but information along the time axis is aggregated. Then, the spatial filter of FFDConv is obtained, which assigns $K \times K$ spatial weight to every frequency kernel and is highly related to the input. Note that full dynamic convolution [18] assigns $K \times K$ spatial weight to every pixel. Consequently, FFDConv can model features from different frequency bands of the input in independent representation spaces.

Considering these representation spaces may be far apart from each other, we employ an attention module following [12] to limit individual differences between them so as not to be too large. Finally, the spatial filter is passed through a Filter-Norm module following [18], avoiding the gradient vanishing/exploding during training.

Channel filter generating. As illustrated in Fig. 3, the channel filter generating module is similar to the SE block [7]. It compresses the time and frequency feature of input by applying an average pooling and maps the channel dimension from C to CK^2 by two fully connected (FC) layers. Between two fully connected layers, the ReLU activation function is applied to introduce non-linearity. After input is passed through this module, the channel filter of FFDConv is obtained, which assigns C channel weight to each spatial location of the frequency kernel. It should be noted that the channel filter for

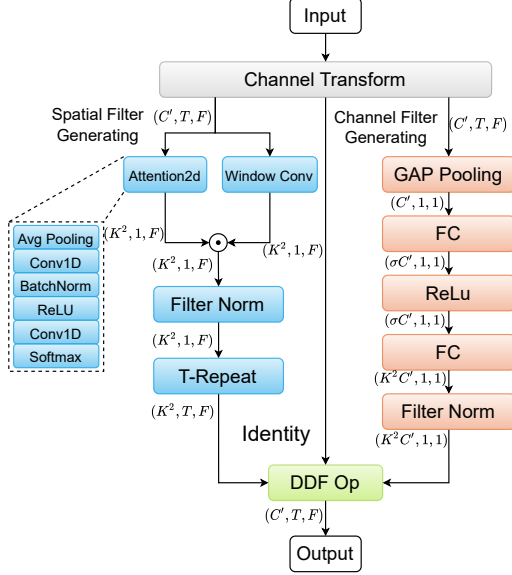


Fig. 3. Details of the FFDCov

F frequency kernels is the same. In the end, the channel filter is also passed through the Filter-Norm [18]. The spatial and channel filters are mixed by dot product, and the full frequency kernels are obtained. We then use them to model time-frequency patterns of input features.

2.4. FFDCov block

Considering frequency kernels of FFDCov don't have output channel dimension, we design an FFDCov block that contains the channel mapping. As illustrated in Fig. 3, firstly, the channel dimension of input is mapped from C_{in} to C_{out} after passing through the channel transformation module. Then, based on the input feature, the spatial and channel filters are obtained by passing through the spatial and channel filter generating module. Full-frequency dynamic kernels are obtained by mixing the spatial and channel filters. Finally, the kernels are convoluted with input along the time axis.

In the actual algorithm, following [18], spatial filters, channel filters, and input are sent to DDF operation to get the output, which is implemented in CUDA, alleviating any need to save intermediate multiplied filters during network training and inference. Note that the DDF op needs $H \times W$ spatial filters. We repeat the $1 \times F$ spatial filters to $T \times F$ so that the kernel's weights are the same along the time axis when convoluted with input in f^{th} frequency band.

3. EXPERIMENT

3.1. Dataset and experiment setup

All experiments are conducted on the dataset of Task 4 in the DCASE 2022. The training set consists of three types

Table 1. SED performance comparison between models using different dynamic convolution on the validation set.

Model	PSDS1 \uparrow	PSDS2 \uparrow	CB-F1 \uparrow	IB-F1 \uparrow
Baseline [19]	0.370	0.579	0.469	0.714
DDFConv [18]	0.387	0.624	0.467	0.720
FTDConv	0.395	0.651	0.495	0.740
FFDConv	0.436	0.685	0.526	0.751

of data: weakly labeled data (1578 clips), synthetic strongly labeled data (10000 clips), and unlabeled in-domain data (14412 clips). The real validation set (1168 clips) is used for evaluation. The input acoustic feature is the log Mel spectrogram extracted from 10-second-long audio data with a sampling rate of 16 kHz. The feature configuration is the same as [13], in which the input feature has 626 frames and 128 mel frequency bands.

The baseline model is the CRNN architecture [19], which consists of 7 layers of conv blocks and 2 layers of Bi-GRU. Attention pooling module is added at the last FC layer for joint training of weakly labeled data, and mean teacher (MT) [20] is applied for consistency training with unlabeled data for semi-supervised learning. Data augmentations such as MixUp [21], time masking [22], frame-shift, and Filter-Augment [23] are used. The data augmentation parameters are identical to [12].

Poly-phonetic sound event detection scores (PSDS), collar-based F1 score (EB-F1), intersection-based F1 score (IB-F1) are used to evaluate the model performance. Median filters with fixed time length are used for post-processing, and sound events have different thresholds from each other to obtain hard predictions for calculating EB-F1. The metrics hyperparameters are identical to [12]. The model is trained using the Adam optimizer with a maximum learning rate of 0.001, and ramp-up is used for the first 80 epochs.

3.2. Full-frequency dynamic convolution on SED

We compared the performances of baseline with full dynamic convolution methods, including decoupled dynamic convolution (DDFConv) [18], full-temporal dynamic convolution (FTDConv), and full-frequency dynamic convolution (FFDConv). For full dynamic convolution methods, dynamic convolution layers replaced all convolution layers except the first layer from the baseline model [19].

The results are shown in Table 1. Three types of full dynamic convolution can all outperform the baseline, which proves full dynamic convolution qualifies in SED. In addition, it can be seen that the effects of three types of convolution are in increasing order. First, FTDConv and FFDConv employ content-adaptive temporal or frequency kernels, which can be viewed as giving prior knowledge to SED compared with DDFConv. Second, FFDConv outperforms FTD-

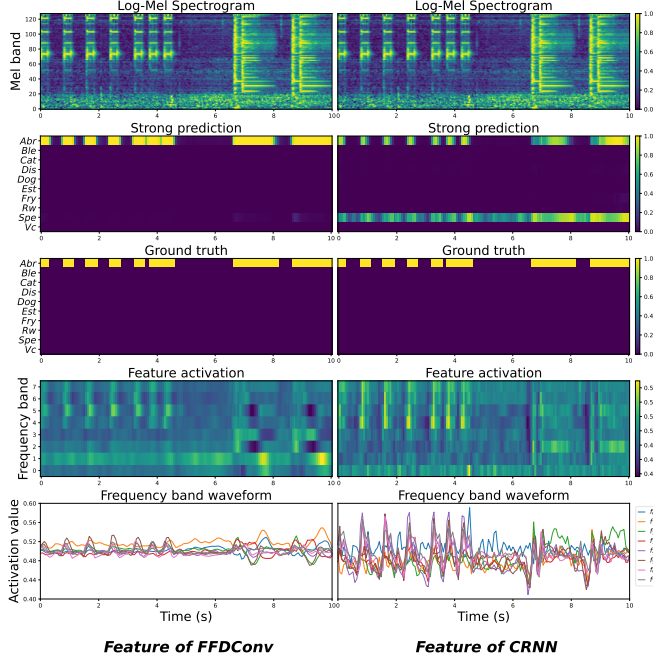


Fig. 4. Feature comparison of FFDCConv and CRNN. Features activation of the 5th Conv block are shown in the 4th row. The trends of frequency band features over time are shown in the 5th row. Note that y-axis labels of strong prediction are abbreviations of the sound event categories. For example, Abr stands for Alarm bell ringing.

Conv, which can prove that time-frequency patterns of sound events are highly frequency-dependent, and this dependency is time-invariant. Moreover, FFDCConv models acoustic features with different kernels along the frequency axis, which can be thought to be frequency components modeled in different representation spaces. As if components of the feature are split into different frequency spaces and then reassembled. This is consistent with the characteristics of sound events.

3.3. Fine-grained modeling study

To explore FFDCConv’s ability to understand acoustic spectral information at a fine-grained level. We visualized feature of the middle layer. More visualizations can be found in the supplementary material.

The visualization results are shown in Fig. 4. Comparing the features of FFDCConv and CRNN, we can see that most of the time-frequency patterns modeled by CRNN are temporally isolated and disjoint. In contrast, FFDCConv’s patterns and their neighbors are in a whole, thereby forming a distinct time-frequency representation. Moreover, this phenomenon can also be found in trends of frequency band features over time. The waveforms of FFDCConv are smoother than CRNN. Specifically, the duration of peak and trough is longer in FFDCConv’s waveform, which results from the feature being

Table 2. Comparison of different window size, W .

Model	Atten	W	PSDS1	PSDS2
FFDCConv	✗	3	0.421	0.650
	✓	1	0.421	0.659
	✓	3	0.436	0.685
	✓	5	0.423	0.656
	✓	7	0.432	0.666

mostly coherent over time. There are more pulses in the resting state of CRNN’s waveforms, which are in a disorganized state. Besides, the distributions of frequency band features are consistent with alarm_bell_ring’s spectrogram in FFDCConv’s waveforms. The values of the low-frequency band features are smaller than those of the middle and high-frequency bands when the alarm bell rings. However, the differences between frequency bands in CRNN are ambiguous. As for the model’s prediction, the CRNN’s isolated features directly lead to the incoherent output compared with ground truth, which proves that the feature’s coherence over time is essential. It’s interesting that the low-frequency white noise of the sound clip is filtered by FFDCConv, but CRNN tagged it as a speech. This has to do with that dynamic convolution concentrates more on high-frequency texture information, and white noise in the spectrogram lacks clear contour information.

Actually, most SED models are trained in a frame-based supervised way, which always leads to the feature and output being discrete over time. However, FFDCConv can alleviate this by frequency-dependent modeling, which models different patterns for frequency bands, leading to a distinct representation of a sound event. This modeling way is like an attention mechanism in which the distribution of frequency band information of the spectrogram is maintained. Besides, the convolution kernel for a frequency band is shared in all frames, which produces temporally coherent representations. This is consistent with both the continuity of the sound waveform and the vocal continuity of sound events.

3.4. Ablation study

We compared the performance of different window sizes of the build kernel when generating spatial filters. Note that the size of the spatial filter K is set to 3.

The results are shown in Table 2. With constraints of the attention module, FFDCConv can get better performance. This proves that before attention, spatial filters of different frequency spaces may have a large distance from each other. The performance of FFDCConv is the best when window size is set to 3. This is because the adjacent frequency components are considered compared to size 1 when generating the spatial filter, and size 5 may suffer from overfitting. In addition, it’s interesting that the performance recovers when the window size is set to 7. This may have to do with the fact that

dynamic convolutions are relatively unstable.

4. CONCLUSIONS

In this paper, we proposed full-frequency dynamic convolution, the first full-dynamic method for SED. Full-frequency dynamic convolution is designed to model time-frequency patterns in different frequency spaces. This design in structure physically furnished 2D convolution with capability of frequency-dependent modeling. Experiments on the DESED show that full-frequency dynamic convolution is superior to not only baseline but also other full-dynamic convolutions, which proves FFDCnv qualifies in SED. In addition, by visualizing features of sound events, we found that FFDCnv can extract temporally coherent features in specific frequency bands, which is consistent with the vocal continuity of sound events. This proves that FFDCnv has great frequency-dependent perception ability. In the future, we aim to explore new methods to model vocal continuity of sound events.

5. REFERENCES

- [1] J. Rouat, “Computational Auditory Scene Analysis: Principles, Algorithms, and Applications,” *IEEE TNN*, vol. 19, no. 1, pp. 199–199, Jan. 2008.
- [2] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye, “Object Detection in 20 Years: A Survey,” *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, March. 2023.
- [3] Jiaying Sun, Yujie Li, Huimin Lu, Tohru Kamiya, and Seiichi Serikawa, “Deep Learning for Visual Segmentation: A Review,” in *COMPSAC*, July. 2020, pp. 1256–1260.
- [4] Qin Jin, Peter Schulam, Shourabh Rawat, Susanne Burger, Duo Ding, and Florian Metze, “Event-based video retrieval using audio,” in *INTERSPEECH*, Sept. 2012, pp. 2085–2088.
- [5] Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer, “Monitoring Activities of Daily Living in Smart Homes: Understanding human behavior,” *IEEE SPM*, vol. 33, no. 2, pp. 81–94, March. 2016.
- [6] Juan Pablo Bello, Charlie Mydlarz, and Justin Salamon, *Sound analysis in smart cities*, pp. 373–397, Springer International Publishing, Sept. 2017.
- [7] Jie Hu, Li Shen, and Gang Sun, “Squeeze-and-Excitation Networks,” in *CVPR*, Jun. 2018, pp. 7132–7141.
- [8] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang, “Selective Kernel Networks,” in *CVPR*, Jun. 2019, pp. 510–519.
- [9] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, “CBAM: Convolutional Block Attention Module,” in *ECCV*, Sept. 2018, pp. 3–19.
- [10] Tong Na and Qinyi Zhang, “Convolutional network with conformer for semi-supervised sound event detection,” Tech. Rep., DCASE2021 Challenge, 2021.
- [11] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool, “Dynamic filter networks,” in *NEURIPS*, Dec. 2016.
- [12] Hyeonuk Nam, Seong-Hu Kim, Byeong-Yun Ko, and Yong-Hwa Park, “Frequency Dynamic Convolution: Frequency-Adaptive Pattern Recognition for Sound Event Detection,” in *INTERSPEECH*, Sept. 2022, pp. 2763–2767.
- [13] Chao Li, Aojun Zhou, and Anbang Yao, “Omni-Dimensional Dynamic Convolution,” in *ICLR*, Apr. 2022.
- [14] Shengchang Xiao, Xueshuai Zhang, and Pengyuan Zhang, “Multi-Dimensional Frequency Dynamic Convolution with Confident Mean Teacher for Sound Event Detection,” in *ICASSP*, Jun. 2023, pp. 1–5.
- [15] Julio Zamora Esquivel, Adan Cruz Vargas, Paulo Lopez Meyer, and Omesh Tickoo, “Adaptive Convolutional Kernels,” in *ICCV Workshops*, Oct. 2019, pp. 0–0.
- [16] Zhi Tian, Chunhua Shen, and Hao Chen, “Conditional Convolutions for Instance Segmentation,” in *ECCV*, Aug. 2020, pp. 282–298.
- [17] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen, “SOLOv2: Dynamic and Fast Instance Segmentation,” in *NEURIPS*, Dec. 2020, pp. 17721–17732.
- [18] Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang, “Decoupled Dynamic Filter Networks,” in *CVPR*, Jun. 2021, pp. 6647–6656.
- [19] Emre Çakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, “Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection,” *IEEE/ACM TASLP*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.
- [20] Antti Tarvainen and Harri Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *NEURIPS*, Dec. 2017.
- [21] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” *ICLR*, Oct. 2018.
- [22] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *INTERSPEECH*, Sept. 2019, pp. 2613–2617.
- [23] Hyeonuk Nam, Seong-Hu Kim, and Yong-Hwa Park, “Filteraugment: An Acoustic Environmental Data Augmentation Method,” in *ICASSP*, May. 2022, pp. 4308–4312.