IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 1, 2020, accepted August 5, 2020, date of publication August 11, 2020, date of current version August 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015714

Channel Compression: Rethinking Information Redundancy Among Channels in CNN Architecture

JINHUA LIANG, (Student Member, IEEE), TAO ZHANG, (Member, IEEE), AND GUOQING FENG
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
Corresponding author: Tao Zhang (zhangtao@tju.edu.cn)

This work was supported by the Graduate Research Innovation Project of Tianjin under Grant 2019YJSS146.

ABSTRACT Model compression and acceleration are attracting increasing attention due to the demand for embedded devices and mobile applications. Research on efficient convolutional neural networks (CNNs) aims at removing feature redundancy by decomposing or optimizing the convolutional calculation. In this work, feature redundancy is assumed to exist among channels in CNN architectures, which provides some leeway to boost calculation efficiency. Aiming at channel compression, a novel convolutional construction named compact convolution is proposed to embrace the progress in spatial convolution, channel grouping and pooling operation. Specifically, the depth-wise separable convolution and the point-wise interchannel operation are utilized to efficiently extract features. Different from the existing channel compression method which usually introduces considerable learnable weights, the proposed compact convolution can reduce feature redundancy with no extra parameters. With the point-wise interchannel operation, compact convolutions implicitly squeeze the channel dimension of feature maps. To explore the rules on reducing channel redundancy in neural networks, the comparison is made among different point-wise interchannel operations. Moreover, compact convolutions are extended to tackle with multiple tasks, such as acoustic scene classification, sound event detection and image classification. The extensive experiments demonstrate that our compact convolution not only exhibits high effectiveness in several multimedia tasks, but also can be efficiently implemented by benefitting from parallel computation.

INDEX TERMS Acoustic scene classification, convolutional neural networks, image classification, model compression and acceleration, sound event detection.

I. INTRODUCTION

Convolutional neural networks (CNNs) are attracting considerable attention in an increasing array of area, such as computer vision [1]–[3], computational acoustics [4]–[6] and natural language processing [7]–[9]. The general trend is to design deeper and more complicated network architecture to pursue better performance. However, massive resources are required for desired performance, which hinders CNN-based classifiers from the real-time inference in mobile applications. Over the past few decades, various methods have been exploited for model compression and acceleration, including pruning [10]–[13], weight sharing [14], [15], low-rank matrix factorization [16]–[18] and knowledge distillation [19]–[21]. Despite their desirable compression abilities, most of the compression methods typically suffer from two major drawbacks. First, the original complex model is replaced with an approximation one, resulting in the error accumulation. Therefore, fine-tuning is usually necessary for their satisfying performance. Second, various manually chosen parameters (and even a lot of empirical engineering that only experts are competent to deal with) are required in these methods.

To overcome the above drawbacks, several efficient convolution methods are recently developed to design specific convolutional kernels for less parameters and calculations. In 2016, Szegedy *et al.* [22] proposed an asymmetrical convolution where a standard d × d convolution layer is spatially factorized as a sequence of two layers with d × 1 and 1 × d convolutions. Howard *et al.* [23] proposed MobileNet v1 that replaces the standard convolution with the depth-wise

The associate editor coordinating the review of this manuscript and approving it for publication was Seok-Bum Ko.

separable convolution. The work by Zhang *et al.* [24] proposed ShuffleNet, applying group convolution and channel shuffle. Iandola *et al.* [25] proposed SqueezeNet in which 1 × 1 convolutions are utilized to reduce channel numbers and replace a part of 3 × 3 convolutions for less parameters. Although some research [24]–[26] has investigated on reducing the channel number in the current layer to cut down the following convolutional operations, this problem is simply solved by appending 1 × 1 convolutional layer, which introduces extra parameters and considerable interchannel calculations.

In this paper, we found that feature redundancy exists among channels in CNN architecture, i.e., amounts of interchannel information is unimportant or even unnecessary in some cases. Instead of 1 × 1 convolutions, a novel convolutional construction named compact convolution is proposed to implicitly reduce feature redundancy in a nonlearning approach. Specifically, the point-wise operation among channels (the point-wise interchannel operation) is implemented to squeeze the channel dimension of input feature maps. The reason for applying the point-wise operation is threefolds. First, the point-wise operation compresses the interchannel information without extra parameters, directly reducing the cost of computation. Second, the derivation of these point-wise operations can be taken easily, which contributes to the chain rule and training end-to-end networks from scratch. Third, the point-wise operation is well-suited for parallel computation on GPU or other advanced chips. Depth-wise separable convolution is further introduced to decouple spatial feature extraction from interchannel feature extraction. Like other research on efficient convolutional kernels [23]–[25], [27], useful features from feature maps can be extracted with fewer parameters and operations by simply replacing the standard convolution with our compact convolution. In addition, how different types of point-wise operations impact on interchannel feature compression is further investigated. While there is tremendous difference between sounds and images, our compact convolution yields desired performance in multiple tasks, such as acoustic scene classification, sound event detection and image classification. To the best of our knowledge, there is few work to verify the generalization of their models in across multiple media.

Extensive experiments show that compared with general network constructions (such as VGG, Resnet and MobileNets), the network with compact convolutions (hereafter CompactNet) not only greatly reduces computation complexity, but also yields desirable performance. To further illustrate the difference between linear manner and non-linear one, three different point-wise operations are compared. Some guidelines are provided for investigating model compression and accerleration.

The contributions of this work are summarized as follows:

1) A novel convolution named compact convolution is proposed to implicitly reduce feature redundancy in a nonlearning approach. Different from the existing channel compression method which directly utilizes 1 × 1 convolution, the proposed compact convolution adopts the point-wise interchannel operation to squeeze the channel dimension of feature maps with no extra parameters. It turns out that compact convolutions not only cost at least 18 times less computation than standard convolutions in terms of 3 × 3 size, but also yield competitive performance.

2) Some guidelines on replacing learnable parameters and complex operations in convolutional layers are summarized. This facilitates further investigation on feature dimension reduction in CNNs.

3) The proposed convolution can be easily applied in general CNN architectures, by replacing the current convolutions with our compact convolutions. Moreover, the compact convolution can extract either audio or visual features to solve multimedia problems.

The reminder of the paper is organized as follows. Section II provides a brief survey of related work. Section III first presents the proposed compact convolution, and then applies it into several popular CNN architectures. In Section IV, extensive experiments are conducted to evaluate CompactNets. Finally, several conclusions and possible future works are given in Section V.

II. RELATED WORK

A. 1 × 1 CONVOLUTION

1 × 1 convolution was first proposed by Lin *et al.* [28] as a universal function approximator for feature extraction on the local patches. They found that 1 × 1 convolution not only has great capability in modeling various distributions of latent concepts, but also facilitates the learnable interactions of cross-channel information. Sequent work in [29], [30] utilized 1 × 1 convolution for tuning the number of feature maps in CNN architecture. However, 1 × 1 convolution involves considerable parameters and operations. This work applies the point-wise interchannel operation to reduce the dimension of feature maps.

B. MAXOUT FUNCTION

In 2013, Goodfellow *et al.* [31] proposed a maxout construction that performs a max pooling across multiple affine feature maps. It turned out that the maxout construction results in a piecewise linear function which is capable of modeling any convex function. Wu *et al.* [32] proposed the Max-Feature-Map (MFM) layer as a variation of maxout activation to suppress low-activation neurons in each layer. Rather than a better function approximator, this paper focuses on the efficient approaches for reducing the interchannel redundancy, and compressing the dimension of feature maps in a larger range. Moreover, besides the max pooling, two more operations are investigated and further integrated into the proposed convolutional layer.

C. DEPTH-WISE SEPARABLE CONVOLUTION

Howard *et al.* [23] proposed MobileNets v1 which took the idea of the depth-wise separable convolution and achieved

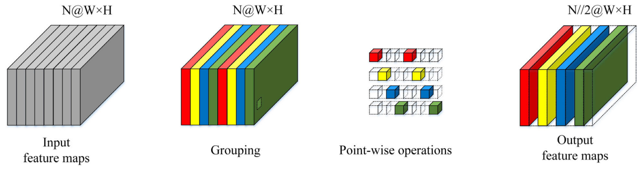


FIGURE 1. Point-wise interchannel operation over feature maps. The compact factor *C* is set to 2 in this figure. Thus, the input feature maps are first grouped and then point-wise operations are implemented over the 2 feature maps in each group.

preferable results on small models. Depth-wise separable convolution consists of a depth-wise convolution for spatially filtering and a point-wise convolution (1 × 1 convolution) for exchanging information among channels. By replacing standard convolutions with depth-wise separable convolutions, the optimized network costs about 9 times less computation than the standard convolution at the cost of a small reduction in accuracy. Inspired by the depth-wise separable convolution, the compact convolution decouples spatial feature extraction from interchannel feature extraction. Moreover, the point-wise interchannel operation is introduced between the depth-wise convolution and 1 × 1 convolution. Thus, the efficiency of convolution is further improved.

III. PROPOSED METHOD

A. THE POINT-WISE INTERCHANNEL OPERATION

As shown in Fig. 1, the point-wise operation is implemented on the feature maps across channels. The input feature maps are firstly divided into groups. And a new feature map is extracted point by point over C feature maps in each group. Therefore, the parameter C can be deemed as a hyper-parameter for adjusting the ratio of channel compression. As C gets larger, the resulting construction becomes more compact.

The input feature maps and the output feature maps of the point-wise interchannel operation are denoted as $I \in F^{N\times W\times H}$ and $O \in F^{N'\times W\times H}$, where N and N' are the channel numbers of input feature maps and output feature maps, W and H are the width and height of the feature maps respectively. Each pixel on the output feature maps is independently calculated with the values in the identical position across channels. Thus, the point-wise interchannel operation of the position (w, h) $(0 \le w < W, 0 \le h < H)$ is defined as

$$O_{w,h}(n) = T_{k=0}^{C-1}\left(I_{w,h}(n + \frac{N}{C}k)\right), \quad n \in [0, N') \tag{1}$$

Here $T(*)$ represents the point-wise operations across channels in the same group. The adopted point-wise operation can be divided into non-linear and linear manners. The non-linear manner which combines C feature maps and outputs element-wise maximum one is defined as:

$$O_{w,h}(n) = max_{k=0}^{C-1}\left(I_{w,h}(n + \frac{N}{C}k)\right), \quad n \in [0, N') \tag{2}$$

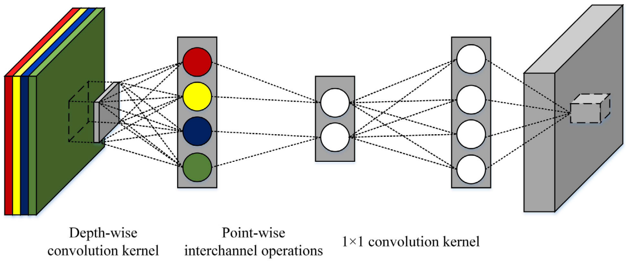

Depth-wise convolution kernel
Point-wise interchannel operations
1×1 convolution kernel

FIGURE 2. Illustration of compact convolutions. Input feature maps go through depth-wise convolutions, point-wise interchannel operations and 1 × 1 convolutions. The compact factor *C* is set to 2 in this figure. Different colors denote different channels of feature maps.

The gradient of Eq. (2) takes the following form:

$$\frac{\partial O_{w,h}(n)}{\partial I_{w,h}(j)} = \begin{cases} 1, & \arg\max\limits_{0\le j<C}\left(n + \frac{N}{C}j\right) \\ 0, & otherwise \end{cases} \tag{3}$$

Likewise, the linear manner is defined as:

$$O_{w,h}(n) = \frac{1}{m}\sum_{k=0}^{C-1} I_{w,h}(n + \frac{N}{C}k), \quad n \in [0, N') \tag{4}$$

Here m is set to 1 when the sum method is applied, otherwise set to C. The gradient of Eq. (4) can be written as follows:

$$\frac{\partial O_{w,h}(n)}{\partial I_{w,h}(j)} = \frac{1}{m} \tag{5}$$

Because the point-wise operation can be simultaneously processed in different groups, it is well-suited for parallel computation on the modern processors. Compared with 1 × 1 convolution performing weighted linear recombination across all the input feature maps, each output feature map produced by the point-wise operation is calculated from the local information of the grouped input feature maps with no extra learnable weights. Thus, the point-wise interchannel operation is capable of reducing considerable parameters and computation resources.

B. COMPACT CONVOLUTION

Taking advantages of the depth-wise separable convolution and the point-wise interchannel operation, a novel compact convolution layer is proposed for the efficient network. The proposed compact convolution is illustrated in Fig.2. Depth-wise convolution is operated over each input feature map to extract spatial features. The following point-wise interchannel operation squeezes the channel dimension of feature maps extracted by depth-wise convolutions, and preserves their major information. Finally, 1 × 1 convolution is applied for the exchange of information among channels. As one can see, there is a bottleneck construction inside the compact convolution. The bottleneck construction leaves the 1 × 1 layer with smaller input/output dimensions, which is beneficial to reduce the cost of computation. Compared with other bottleneck constructions [30] designed with 1 × 1 convolution, the proposed compact convolution reduces the channel dimension with less calculation and no extra learnable weights.

A standard convolution layer takes a $W_{in} \times H_{in} \times C_{in}$ feature map F as input. Here W_{in} and H_{in} are the spatial width and height of the input feature map, C_{in} is the number of input channels. And a $W_{out} \times H_{out} \times C_{out}$ feature map G is produced by a standard convolution, where W_{out} and H_{out} are the width and height of the output feature map and C_{out} is the number of output channels. The standard convolutional layer is parameterized by convolution kernel sized $K \times K \times C_{in} \times C_{out}$ where K is the spatial dimension of the kernel assumed to be square, C_{in} and C_{out} are numbers of input and output channels as defined previously.

Based on [33], the complexity of networks is evaluated with FLOPs, i.e. the number of floating-point multiply-add operations. Assume that F denotes FLOPs of the standard convolution. It can be computed as:

$$F = 2C_{in}K^2 H_{out} W_{out} C_{out} \tag{6}$$

Likewise, F' represents FLOPs of the compact convolution. Through the depth-wise convolution, the point-wise interchannel operation and 1×1 convolution, F' is calculated as:

$$F' = \left(2K^2 + \frac{m}{C} + \frac{2C_{out}}{C}\right) C_{in} H_{out} W_{out} \tag{7}$$

where m is set to $C - 1$ when the maximum and the sum methods are imposed, otherwise set to C. Then the compression rate α of F' over F is obtained as:

$$\alpha(F', F) = \frac{1}{C_{out}} + \frac{m}{2CK^2 C_{out}} + \frac{1}{CK^2} \tag{8}$$

Since the standard convolution sized 3×3 is the most frequently-used construction in CNN architecture, the kernel size of the compact convolution is set to 3 in the experiments. It turns out that when the number of filters is large, FLOPs of the compact convolution sized 3×3 are approximately 18 times less than FLOPs of the standard counterpart. With the increase of the compact factor C, the ratio gets even higher. A further discussion is demonstrated in Sect. V.

C. APPLICATION IN NETWORK ARCHITECTURES

Since the compact convolution is a "sparse" version of the standard convolution, it can be embedded into general network architectures by simply replacing standard convolutions with compact convolutions. In this work, three different networks with compact convolutions are proposed as follows.

VGG-like. Following the design principle of VGG net [34], a block consisting of two-layer 3×3 convolutions is imposed as a basic building block. Considering the limitation of dataset size, an eight-layer stacked convolutional model is adopted in the proposed VGG-like networks. The standard convolution is utilized as the first two convolutional layers, and compact convolutions are imposed as the other six convolutional layers. All the convolutional layers are followed by batch normalization [35] and ReLU non-linear activation [36].

ResNet-like. The "bottleneck" design is adopted in our proposed networks, which has been demonstrated desired performance in [30]. Different from [30], it is unnecessary to append 1×1 convolution following 3×3 convolution in the bottleneck block, because our compact convolution itself includes a 1×1 convolution.

MobileNet-like. To build MobileNet-like networks, depth-wise convolutions and point-wise convolutions are replaced with compact convolutions. In [23], the input channel number of a given depth-wise separable convolution with width multiplier α is reduced from C_{in} to αC_{in}. Likewise, its output channel number is reduced to αC_{out}. Therefore, the model complexity with width multiplier α decreases by roughly α^2. Since our compact convolution adjusts the number of channels through the point-wise interchannel operation, the width multiplier of the depth-wise convolution is fixed to 1 so as not to interfere with the experiments.

D. ANALYSIS IN THE TRAINING STAGE

Different types of point-wise operations make various impacts among channels on both inference and backpropagation stage. In Eq. (4) and Eq. (5), except for weights, the sum and the average methods process the feature maps among channels in the same way. Therefore, the point-wise operation can be divided into linear and non-linear manner according to the interchannel processing. Empirically, the linear manner is prone to preserve the major information among local channels, while the non-linear one tends to extract prominent features among local channels. The accuracy and cross-entropy loss of three different point-wise interchannel operations on the DCASE 2019 dataset are shown in Fig. 3. The convergence of the max method is slower than the convergence of the other two methods on both the training dataset and the validation dataset. In addition, it can be seen that the curves resulted by the sum and the average methods are similar, because both of them compress the information among channels in the linear manner.

IV. GENERALIZATION IN MULTIMEDIA

To assess its capacity of generalization in cross media, the proposed networks are applied to tackle with three different tasks, including acoustic scene classification (ASC), sound event detection (SED) and image classification (IC). ASC and SED take 2-D time-frequency spectrograms as inputs to CNN classifier while IC directly utilizes images as inputs. "Acoustic scene" here is referred as a mixture of background noise and sound events associated with a specific audio scenario. So compared with SED, ASC tends to make the discrimination with more abstract and global features.

ASC aims at enabling devices to recognize the specific audio environment from a recording or an on-line stream. To solve this problem, the proposed networks are trained and evaluated on the development dataset of TAU Urban Acoustic Scenes 2019 [37] in DCASE 2019 task 1. The dataset contains

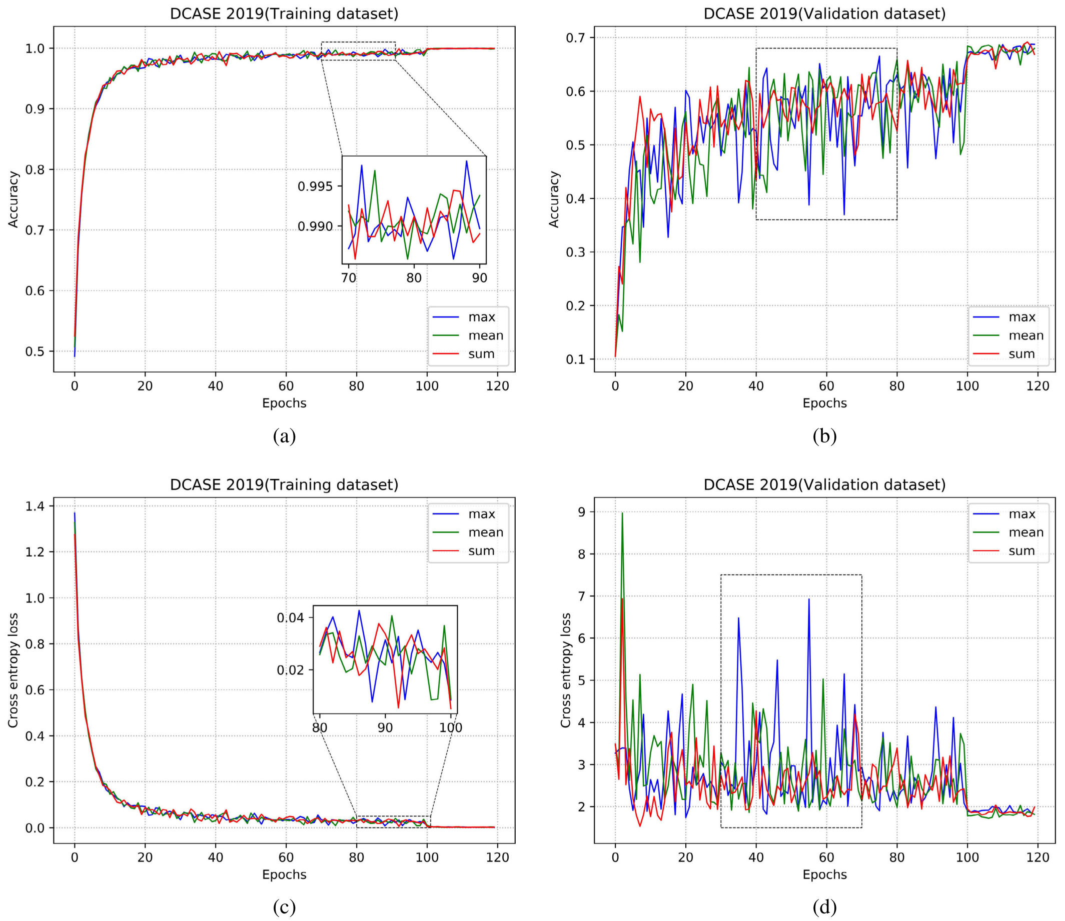


FIGURE 3. Accuracy and cross-entropy loss of three different point-wise interchannel operations on the DCASE 2019 dataset. (a) Training accuracy (b) Validation accuracy (c) Training loss (d) Validation loss. Some details of curves are highlighted with dashed rectangular boxes.

several acoustic scenes and various locations for each scene. The original recordings sampled with 44.1kHz are segmented into 10-second clips. The dataset consists of 10 scene classes, including airport, shopping mall, metro station, street pedestrian, public square, street traffic, tram, bus, metro and park.

To facilitate the proposed models training, the raw waves with binaural channels are firstly downmixed to mono. Then the log-scaled mel-spectrograms are extracted from each audio wave with hamming widow size of 1724 samples (corresponding to 0.04s), overlap of 50%, and 128 mel bands. Therefore, a feature map with a size of 128 × 512 is generated for each audio waves. The features are finally normalized with z-scores, and fed into the proposed models.

SED aims to detect and classify events that occur in different environments. To solve this problem, the proposed networks are trained and evaluated on UrbanSound8K [38]. The dataset contains 8732 labeled sound clips of urban sounds from 10 classes, including air conditioner, car horn children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. Different from DCASE 2019, the length of clips is varying from 0s to 4s. The pre-processing on SED for training is similar with the one on ASC, except zero padding is adopted to unify the length of raw wave.

IC is a classical problem in computer vision. Aiming at evaluating the performance of our models on IC, CIFAR 10 is utilized for further experiments in Sect. V. CIFAR 10 contains 60000 32×32 color images from 10 non-overlapping classes in the dataset, including airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Without much preprocessing, only normalization is applied for better convergence. The proposed networks are trained on 50000 samples, and validated on 10000 samples.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

The proposed CompactNets with the sum, the max and the average methods are referred as CompactNet-S, CompactNet-M and CompactNet-A, respectively. Since compact convolution is applicable to most of the common network architectures, the proposed CompactNets are built in the same constructions as three different comparison networks, including VGG-8, ResNet and MobileNets. In addition, XVGG-8 is designed by replacing compact convolutions with depth-wise separable convolutions in order to evaluate the performance of the VGG-like CompactNet. To evaluate the efficiency of our CompactNets, some efficient convolutional neural

TABLE 1. Comparison of several models over parameters, complexity computations and speed on two platforms. The results are grouped by different network architectures. The complexity of our CompactNets with the maximum, sum (left) and the average (right) methods are given independently. The speed on CPU and GPU is evaluated with single thread. The best results are highlighted in bold.

Model	Num. of params.	Complexity (MFLOPs)	Speed on CPU (Samples/sec.)	Speed on GPU (Samples/sec.)
VGG-8	4,697,034	20233.6	1.70	6.62
XVGG-8	580,362	6521.4	3.21	7.41
VGG-like CompactNet (C=2)	329,034	5661.7/5663.9	3.04	7.81
VGG-like CompactNet (C=4)	200,010	5231.8/5232.9	3.26	7.94
VGG-like CompactNet (C=8)	135,498	5016.9/5017.4	**3.32**	**8.13**
ResNet	23,601,930	9535.0	3.10	5.59
ResNet-like CompactNet (C=2)	9,803,722	3860.2/3861.0	4.00	6.21
ResNet-like CompactNet (C=4)	8,546,250	3341.9/3342.3	4.55	6.33
ResNet-like CompactNet (C=8)	7,917,514	3082.7/3082.9	**4.88**	**6.37**
1.0 MobileNet v1	3,238,538	1362.6	9.52	62.50
0.5 MobileNet v1	834,378	354.1	11.36	71.43
0.25 MobileNet v1	220,970	95.3	**20.83**	**76.92**
MobileNet-like CompactNet (C=2)	1,668,746	709.5/710.8	10.31	66.67
MobileNet-like CompactNet (C=4)	883,850	382.9/383.6	12.35	71.43
MobileNet-like CompactNet (C=8)	491,402	219.7/220.0	14.08	**76.92**

TABLE 2. Comparison of several efficient convolutional neural networks over parameters, complexity computations and speed on two platforms. The best results are highlighted in bold.

Model	Num. of params.	Complexity (MFLOPs)	Speed on CPU (Samples/sec.)	Speed on GPU (Samples/sec.)
1.0 MobileNet v1	3,238,538	1362.6	9.52	62.50
1.0 MobileNet v2	2,288,458	700.4	6.99	38.46
ShuffileNet v1 2× (g=3)	3,631,450	1221.2	6.71	40.00
ShuffileNet v2 2×	5,407,316	1504.1	6.99	58.82
MobileNet-like CompactNet (C=2)	1,668,746	709.5/710.8	**10.31**	**66.67**

TABLE 3. Accuracy of various models on DCASE 2019. The best results are highlighted in bold.

Model	Complexity (MFLOPs)	Accuracy (%)
VGG-8	20233.6	**69.96±0.31**
XVGG-8	6521.4	68.56±1.06
VGG-like CompactNet-A (c=2)	5663.9	68.30±0.75
VGG-like CompactNet-S (c=2)	5661.7	68.66±1.44
VGG-like CompactNet-M (c=2)	5661.7	68.88±0.92
ResNet	9535.0	66.44±1.12
ResNet-like CompactNet-A (c=2)	3861.0	68.78±0.40
ResNet-like CompactNet-S (c=2)	3860.2	**69.00±0.79**
ResNet-like CompactNet-M (c=2)	3860.2	68.04±0.67
1.0 MobileNet v1	1362.6	67.86±0.63
MobileNet-like CompactNet-A (c=2)	710.8	67.92±0.31
MobileNet-like CompactNet-S (c=2)	709.5	67.92±1.23
MobileNet-like CompactNet-M (c=2)	709.5	**68.08±1.03**

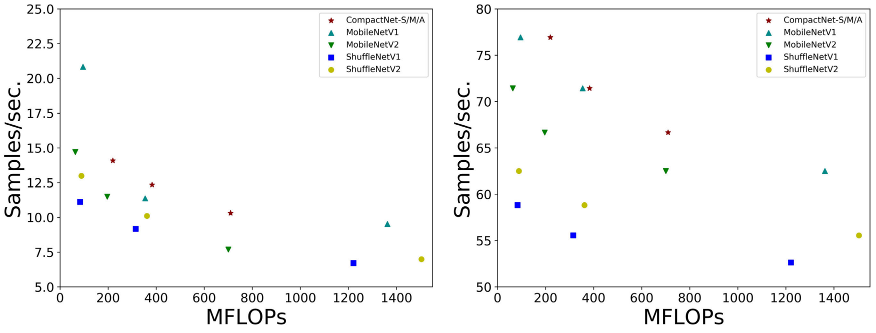
networks (MobileNet v2, ShuffleNet v1 and Shufflenet v2) are built for comparison. Since nothing but convolutional layers changed in the following comparison experiments, only FLOPs of convolutions and our point-wise interchannel operations are taken into account. The above networks are trained by minimizing the cross-entropy loss with Adam optimizer. The learning rate, and batch size are set to 0.001 and 32 respectively.

All the experiments are implemented in python. Besides, experiments are conducted on the computer with Intel®Xeon(R) CPU E5-2650 v4 2.20 GHz and Nvidia RTX 2080Ti GPU. The proposed models are evaluated with Tensorflow.

B. ALGORITHMIC COMPLEXITY

The parameters, complexity and speed of different models are listed in Table 1. The FLOPs of compact convolutions with the max, the sum and the average methods are given independently. For better observation, the results are grouped by different network architectures. Except the MobileNet-like networks on CPU, CompactNets ($C = 8$) are fastest on both CPU and GPU among the networks in the identical structures. Specifically, the speed of VGG-like CompactNet ($C = 8$) is 1.95× and 1.23× more than its VGG-8 counterpart on CPU and GPU respectively. In addition, the speed of ResNet-like CompactNet ($C = 8$) is 1.57× and 1.14× more than its ResNet counterpart on CPU and GPU respectively. It turns out that 0.25 MobileNet v1 is faster than Mobile-like CompactNets. This is because the complexity of 0.25 MobileNet v1 is merely a half of MobileNet-like CompactNet ($C = 8$) complexity. The non-linearity reduction of parameters and FLOPs are caused by the other unchanged convolutions in the networks, such as the first two standard convolutions in the VGG-like networks. Similarly, there are merely a few significant changes in complexity among the three proposed ResNet-like CompactNets, because only one 1 × 1 convolution at the end gets compacted while the other 1 × 1 convolutions have no change.

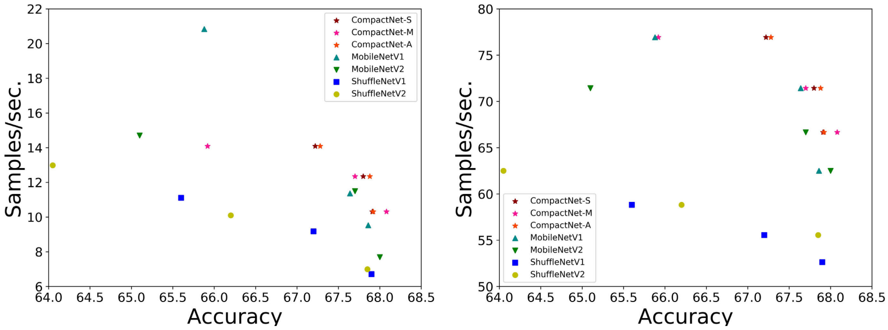
Table 2 lists parameters, computation complexity and speed of several efficient convolutional neural networks. The speed of mobileNet-like CompactNet ($C = 2$) is the fastest on both CPU and GPU. Calculation complexity vs. speed on two different platforms is shown in Fig. 4. Our proposed CompactNets are on the top right region under both cases. It is worthy to note that the indirect metric (complexity) is inconsistent with the direct one (speed), e.g. the difference between CompactNet ($C = 2$) and 1.0 MobileNet v2. This result conforms to the finding in [26]: Besides FLOPs, Memory access cost (MAC) and optimized operation on specific platforms should be also taken into consideration.



(a)

(b)

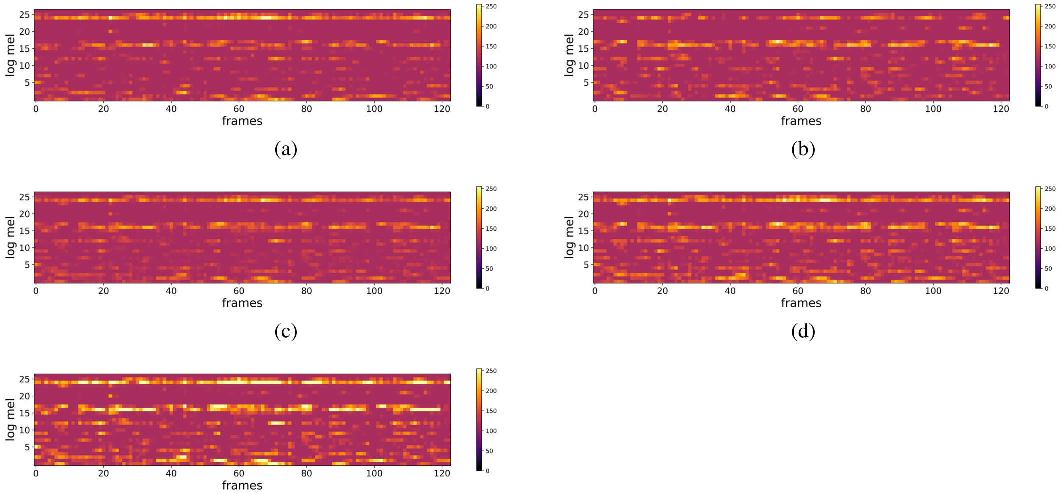
FIGURE 4. Calculation complexity vs. the speed on two different platforms. (a) On CPU (b) On GPU.



(a)

(b)

FIGURE 5. Accuracy on DCASE 2019 vs. the speed on two different platforms. (a) On CPU (b) On GPU.



(a)

(b)

(c)

(d)

(e)

FIGURE 6. Internal feature maps in CNN architecture. (a), (b) are the two input feature maps in the same group. (c), (d), (e) are the compact results with the average, the max and the sum methods respectively. The horizontal axis corresponds to the temporal frames limited to 10s, and the vertical axis corresponds to logarithmic mel-frequency bands. The color in the spectrograms reflects the energy intensity.

C. EVALUATION ON ASC

Table 3 shows the accuracy of different models to handle ASC task on DCASE 2019. The proposed CompactNet-S and CompactNet-M yield the best results among ResNet-like models and MobileNet-like models respectively. It can be seen that VGG-8 outperforms the CompactNets by 1.3%. However, taking the model complexity into consideration, the proposed models are still competitive. Compared with XVGG-8 and MobileNet v1 consisting of separable convolution, CompactNet-M still outperforms them by 0.32% and 0.22% respectively. This indicates that the point-wise interchannel operation can not only squeeze the channel

TABLE 4. Comparison of several models over complexity computations and accuracy in two different tasks.

Model	Sound Event Detection (SED)		Image Classification (IC)	
	Complexity (MFLOPs)	acc. (%)	Complexity (MFLOPs)	acc. (%)
VGG-8	7943.3	84.34±1.41	15549.2	87.58±0.16
XVGG-8	2560.2	81.42±1.90	5050.8	85.18±0.23
VGG-like CompactNet-S (C=2)	2222.7	82.74±1.03	4392.5	84.12±0.13
VGG-like CompactNet-M (C=2)	2222.7	83.34±1.43	4392.5	83.20±0.30
VGG-like CompactNet-S (C=4)	2053.9	82.20±1.94	4063.4	82.76±0.27
VGG-like CompactNet-M (C=4)	2053.9	82.06±1.22	4063,4	80.80±0.28
VGG-like CompactNet-S (C=8)	1969.5	80.94±1.35	3898,8	80.64±0.13
VGG-like CompactNet-M (C=8)	1969.5	80.06±1.39	3898,8	77.10±0.39
ResNet	3743.2	77.82±0.88	7614.9	72.72±5.03
ResNet-like CompactNet-S (C=2)	1515.4	79.40±1.04	3270.2	74.20±1.41
ResNet-like CompactNet-M (C=2)	1515.4	78.08±0.38	3270.2	73.70±0.83
ResNet-like CompactNet-S (C=4)	1311.9	79.18±1.06	2873,3	73.90±1.26
ResNet-like CompactNet-M (C=4)	1311.9	78.02±0.94	2873.3	72.68±0.67
ResNet-like CompactNet-S (C=8)	1210.2	79.02±1.40	2674.9	73.72±0.46
ResNet-like CompactNet-M (C=8)	1210.2	77.84±1.67	2674.9	71.34±1.15
1.0 MobileNet v1	534.9	77.50±1.06	1050.5	78.00±0.37
MobileNet-like CompactNet-S (C=2)	278.5	78.42±1.09	550.4	76.80±0.34
MobileNet-like CompactNet-M (C=2)	278.5	78.46±0.68	550.4	75.00±0.32
MobileNet-like CompactNet-S (C=4)	150.3	77.88±1.25	300.4	74.22±0.54
MobileNet-like CompactNet-M (C=4)	150.3	77.30±0.67	300.4	71.82±0.80
0.5 MobileNet v1	139.0	75.58±1.28	274.8	73.36±0.74
MobileNet-like CompactNet-S (C=8)	86.3	77.58±2.46	175.4	70.14±1.15
MobileNet-like CompactNet-M (C=8)	86.3	75.38±2.06	175.4	68.12±0.54
0.25 MobileNet v1	37.4	73.30±1.95	74.8	66.92±0.91

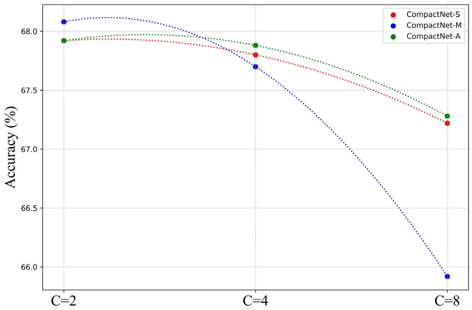


FIGURE 7. Accuracy variations of CompactNets with three different point-wise interchannel operations.

dimension of input feature maps but also filter the useful information which helps further feature extraction. Note that the CompactNets surpass the ResNet by a large margin, because the ResNet is overfitting due to the limitation of dataset. Thus, the proposed compact convolution is capable of avoiding overfitting by reducing the number of learnable weights.

The accuracy on DCASE 2019 vs. the speed on two different platforms is demonstrated in Fig. 5. Our proposed CompactNets are on the top right region under both cases. It turns out that the performance of comparison networks, such as ShuffleNet v1 and ShuffleNet v2, deteriorates rapidly along with the decrease of the scale factor. In contrast, when the compact factor C increases, the variation of our CompactNet accuracy is small. This indicates that the point-wise interchannel operation can not only squeeze the channel dimension of feature maps, but also retain the useful information in features.

D. COMPARISON BETWEEN LINEAR AND NON-LINEAR MANNERS

Fig. 6 illustrates the internal feature maps resulting in the three different point-wise interchannel operations. The max method is clearer than the average method in the detailed information. This indicates that the max method can extract the iconic features from inputs while the average method tends to keep the major information of feature maps. In addition, the distribution of feature with the average method is identical to the one with the sum method. This phenomenon accords to the analysis in Sect. III *A* and *B*.

In Fig. 7, the accuracy variations of CompactNets with three different point-wise interchannel operations are illustrated. With the increase of compact factor C, the performance of CompactNet-S is always consistent with the performance of CompactNet-A. Combining the analysis in Sect. III *A* and *D*, we can summarize several guidelines:

G1) The average method and the sum method among channels work in the same way. By taking FLOPs of these two methods into consideration, the average operation can be replaced with the sum operation to squeeze the channel dimension of input feature maps.

G2) The nonlinear operation is relatively hard to converge, and it tends to yield desirable performance with small compact factors. The maximum method extracts the maximum value within a group and discards the remaining ones. As the compact factor C gets large, this nonlinear mapping loses a large amount of characteristic information, which leads to a rapid deterioration in performance.

G3) The linear operation is relatively easy to converge, and it tends to outperform other methods in the case of large compact factors. In contrast to maximum method, average and sum methods preserve most of the information which facilitates model compression with a large compact factor.

These three guidelines can not only help researchers utilize CompactNets, but also expose the roles of different operations in CNNs.

E. EXTEND TO OTHER TASKS

Based on **G1**, only the sum method and the max method, corresponding to linear manner and non-linear one respectively, are discussed in this subsection.

Table 4 lists the computation complexity and the accuracy in two different tasks. It turns out that our proposed CompactNets produce satisfying results in SED and IC. In SED, our CompactNet-S ($C = 2$) and CompactNet-M ($C = 2$) surpass the competing models among ResNet-like models and MobileNet-like models by 1.58% and 0.96% respectively. Compared with XVGG-8 that consists of separable convolutions, CompactNet-M ($C = 2$) still conducts higher accuracy by 1.92%. In IC, CompactNet-S ($C = 2$) outperforms ResNet by about 1.48%. It is worthy to note that XVGG-8 and 1.0 MobileNet v1 yield better results than CompactNets by 1.06% and 1.2%. This is because the number of samples in CIFAR 10 is large, and each sample sized 32×32 is easy to learn. Therefore, the input feature maps have less leeway to be squeezed.

VI. CONCLUSION

In this paper, a novel convolutional construction was proposed for implicitly reducing feature redundancy, where the point-wise interchannel operation was adopted to squeeze the channel number of feature maps. The depth-wise separable convolution and the point-wise interchannel operation were integrated to speed up calculations and retain a satisfying performance. Unlike traditional methods for dimensional reduction in CNN which introduce considerable learnable weights, our compact convolution has the capacity to squeeze the channel dimension of feature maps with no extra parameters. Moreover, we showed the generalization capacity of models to handle three different tasks, including acoustic scene classification, sound event detection and image classification. Extensive experimental results demonstrated that the proposed method can not only cut down the run time on CPU and GPU but also produce promising performance.

In future, we will investigate proper alternatives to the current convolutional construction with less complexity, and applications to other general multimedia tasks.

REFERENCES

[1] S. Xie and H. Hu, "Facial expression recognition using hierarchical features with deep comprehensive multipatches aggregation convolutional neural networks," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 211–220, Jan. 2019.

[2] H. Cao, H. Liu, E. Song, G. Ma, X. Xu, R. Jin, T. Liu, and C.-C. Hung, "Multi-branch ensemble learning architecture based on 3D CNN for false positive reduction in lung nodule detection," *IEEE Access*, vol. 7, pp. 67380–67391, 2019.

[3] C. Li and B. Yang, "Adaptive weighted CNN features integration for correlation filter tracking," *IEEE Access*, vol. 7, pp. 76416–76427, 2019.

[4] J. Xie, K. Hu, M. Zhu, J. Yu, and Q. Zhu, "Investigation of different CNN-based models for improved bird sound classification," *IEEE Access*, vol. 7, pp. 175353–175361, 2019.

[5] T. Zhang, J. Liang, and B. Ding, "Acoustic scene classification using deep CNN with fine-resolution feature," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113067. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419307845

[6] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 131–135.

[7] Y. Jin, C. Luo, W. Guo, J. Xie, D. Wu, and R. Wang, "Text classification based on conditional reflection," *IEEE Access*, vol. 7, pp. 76712–76719, 2019.

[8] L. Ma, Z. Lu, L. Shang, and H. Li, "Multimodal convolutional neural networks for matching image and sentence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2623–2631.

[9] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416305929

[10] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," 2016, *arXiv:1608.08710*. [Online]. Available: https://arxiv.org/abs/1608.08710

[11] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1389–1397.

[12] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2755–2763.

[13] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," 2018, *arXiv:1802.00124*. [Online]. Available: https://arxiv.org/abs/1802.00124

[14] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 2285–2294.

[15] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 94–106, Jan. 2015.

[16] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6655–6659.

[17] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–13.

[18] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. De Freitas, "Predicting parameters in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2148–2156.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[20] P. Luo, Z. Zhu, Z. Liu, X. Wang, and X. Tang, "Face model compression by distilling knowledge from neurons," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3560–3566.

[21] A. K. Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3438–3446.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[26] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 116–131.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[28] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, vol. 18, 2013, pp. III–1319.

[32] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[33] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*. [Online]. Available: http://arxiv.org/abs/1611.06440

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[36] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[37] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," 2018, *arXiv:1807.09840*. [Online]. Available: http://arxiv.org/abs/1807.09840

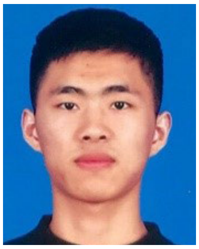
[38] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Int. Conf. Multimedia - MM*, 2014, pp. 1041–1044.

JINHUA LIANG (Student Member, IEEE) received the B.E. degree from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2018. He is currently pursuing the M.S. degree with the School of Electrical and Information Engineering, Tianjin University.

His research interests include acoustic scene classification, sound event detection, and model compression and accelerating.

TAO ZHANG (Member, IEEE) received the M.S. degree from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2001, and the Ph.D. degree from Tianjin University, in 2004.

He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. He is also with the Texas Instruments DSP Joint Laboratory, Tianjin University. His current interests include image, video, and acoustic signal processing, auditory model, speech enhancement, and hardware/software partitioning.



GUOQING FENG received the B.E. degree from the College of Vocation and Technology, Hebei Normal University, in 2019. He is currently pursuing the M.E. degree with the School of Electrical and Information Engineering, Tianjin University.

His research interests include acoustic signal processing and machine learning.