Convolutional Fusion Network for Monaural Speech Enhancement

Yang Xian[a], Yang Sun[b], Wenwu Wang[c], Syed Mohsen Naqvi[a]

[a]*Intelligent Sensing and Communications Research Group, School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.*
[b]*Big Data Institute, University of Oxford, Oxford, OX3 7LF, U.K.*
[c]*Center for Vision Speech and Signal Processing, Department of Electrical and Electronic Engineering, University of Surrey, Surrey GU2 7XH, U.K.*

Abstract

Convolutional neural network (CNN) based methods, such as the convolutional encoder-decoder network, offer state-of-the-art results in monaural speech enhancement. In the conventional encoder-decoder network, large kernel size is often used to enhance the model capacity, which, however, results in low parameter efficiency. This could be addressed by using group convolution, as in AlexNet, where group convolutions are performed in parallel in each layer, before their outputs are concatenated. However, with the simple concatenation, the inter-channel dependency information may be lost. To address this, the Shuffle network re-arranges the outputs of each group before concatenating them, by taking part of the whole input sequence as the input to each group of convolution. In this work, we propose a new convolutional fusion network (CFN) for monaural speech enhancement by improving model performance, inter-channel dependency, information reuse and parameter efficiency. First, a new group convolutional fusion unit (GCFU) consisting of the standard and depth-wise separable CNN is used to reconstruct the signal. Second, the whole input sequence (full information) is fed simultaneously to two convolution networks in parallel, and their outputs are re-arranged (shuffled) and then concatenated, in order to exploit the inter-channel dependency within the network. Third, the intra skip connection mechanism is used to connect different layers inside the encoder as well as decoder to further improve the model performance. Extensive experiments are performed to show the improved performance of the proposed method as compared with three recent baseline methods.

Keywords: convolutional neural network, model capacity, shuffle, group convolutional fusion unit, depth-wise separable convolution, full information, intra skip connection

Email addresses: Y.xian2@newcastle.ac.uk (Yang Xian), Yang.sun@bdi.ox.ac.uk (Yang Sun), W.wang@surrey.ac.uk (Wenwu Wang), Mohsen.naqvi@newcastle.ac.uk (Syed Mohsen Naqvi)

1. Introduction

Speech enhancement aims to reduce interference from noisy speech mixture and improve the intelligibility and quality of target speech. Monaural speech enhancement is an extreme case, where only single channel noisy speech mixture (e.g. recorded by a single microphone) is available, and both target speech and noise are unknown. This problem is widely found in real-world scenarios such as speech communication, automatic speech recognition, and robotics [1, 2, 3, 4].

Many methods have been proposed to address the problem of monaural speech enhancement. Recently, the deep neural networks (DNN) based methods [5] show great potential in monaural speech enhancement. The DNN based methods can be divided into two categories i.e. mapping-based [6, 7] and masking-based methods [8, 9, 10, 11]. Typically, the time-frequency (T-F) representations of noisy speech mixtures and target speeches are provided to train the DNN which learns the masking or mapping relation between them. Then, the trained DNN is used to estimate the target speech from the noisy speech mixture [12]. In the mapping-based methods, the DNN is used to learn the non-linear mapping between the T-F representation of the target speech and that of the noisy speech mixture. In the masking-based methods, the DNN is used to estimate the T-F mask, which is a matrix of weights indicating the probability of the source being present in the noisy speech mixture at each T-F point. Recent research has suggested that the mapping-based methods perform better than the masking-based methods [13].

Although the vanilla DNN is a powerful model, the inter-dependency between the neighboring temporal frames is not considered explicitly, which may limit its performance in mismatch conditions e.g. speaker-independent or noise-independent cases [14, 15]. To address this problem, the recurrent neural network (RNN) based methods i.e. deep RNN (DRNN) is introduced, where the information from the past frames is exploited along with that of the current frame in the DRNN units [16]. In addition, by using the input, forget and output gates, the long short-term memory (LSTM) is capable of controlling how much past information from longer time frames can be used to update the current frame. The LSTM has been shown to offer advantages over the DNN-based methods for the mismatch conditions [15].

Recently, convolutional neural networks (CNN) have been used for speech enhancement. Motivated by CNN-based image processing methods, in speech enhancement, the T-F rep-

resentation of a noisy speech mixture is taken as an input to the CNN, to estimate the target speech [17]. Inspired by the Inception Network (InceptionNet) [18], the multi-resolution features are also introduced for speech enhancement by using multiple filters with various sizes in each layer of multi-resolution convolutional encoder-decoder (MHCED) network [19]. An autoencoder convolutional neural network (AECNN) is proposed to estimate the target speech [20], by using mean absolute error (MAE) as the cost function. In the phase-and-harmonics-aware speech enhancement network (PHASEN), two streams are used to predict the amplitude and phase, which exploits phase information to boost the performance of amplitude based speech enhancement [21]. Dilated CNN has been used to enlarge the receptive fields and capture the interdependency among different frames. For example, gated residual network (GRN) [13] has been used for speech enhancement with dilated CNN, showing better performance than RNN based methods. In addition, temporal convolutional neural network (TCNN) [22] and Conv-TasNet [23] exploit dilated CNN for time-domain speech enhancement and separation. Another direction is to scale up CNN by improving the parameter efficiency of CNN, using the factorized convolutions and aggressive regularization, such as AlexNet [24], InceptionNets [25, 26], and ShuffleNet [27]. In ShuffleNet, group convolution and channel shuffle are introduced to reduce computational cost while maintaining accuracy [27]. In addition, the depth-wise separable convolution is proposed to replace the standard convolution, which shows advantages in parameter efficiency [28, 29].

The aforementioned methods, however, still have limitations. For instance, although a large kernel size used can enlarge the receptive fields of the model in the conventional convolutional encoder-decoder network, it increases the computational cost [20]. The InceptionNet and MHCED utilize multiple kernels of various sizes to improve the model capacity, and the use of large kernel sizes [19] will likely decrease the parameter efficiency and limit its applicability in resource-limited applications. The AlexNet uses two group convolutions in parallel at each layer, with each group taking half of the input sequence [24]. However, for each group of convolution, only part of the input sequence is used, which may limit each kernel to only obtaining partial information from the full input sequence and potentially degrade the model performance. The channel shuffle is proposed to re-arrange channels of group convolution in ShuffleNet [27], which is helpful in enabling the channels to be related with each other. In addition, the ShuffleNet employs sequential standard convolution and depth-wise convolution

to generate a single feature, which can be further improved by preserving two different feature maps of standard and depthwise convolutions. The AECNN model only employs the skip connections between the encoder and decoder, which feeds the information flow from the encoder layers to their corresponding decoder layers [20]. However, the information flow reuse within the encoder/decoder has not been explored, despite its potential benefit for improving enhancement performance.

In this paper, we propose a new framework, namely, convolutional fusion network (CFN) to mitigate some of these limitations. More specifically, we have following contributions.

First, we propose a convolutional fusion unit consisting of standard convolution and depth-wise separable convolution with smaller kernel size. The weighted outputs from these two convolutions are concatenated as the output of the convolutional fusion unit. The convolutional fusion units are used to build the encoder, instead of using only standard (vanilla) convolution.

Second, we propose a novel decoder with deconvolution, depth-wise separable convolution and upsampling layers to improve the model capacity, which is also capable of reducing the dimension of the encoder output.

Third, channel shuffling is introduced to exploit the inter-channel dependency. More specifically, the full input sequence is fed to standard convolution and depth-wise separable convolution, and their outputs are re-arranged and concatenated to utilize the inter-channel dependency according to the channel order. As a result, both groups of convolution can exploit the information from the full input sequence.

Lastly, we apply an intra skip connection mechanism inside the encoder and decoder. With intra skip connection, the ability of reusing information flow within the encoder and decoder is refined.

The remainder of the paper is organized as follows. Section 2 provides a statement of monaural speech enhancement problem. Section 3 presents the proposed CFN method. The experimental settings and results are discussed in Section 4. Section 5 draws the conclusions.

2. Problem Statement

In monaural speech enhancement, the aim is to recover the clean speech from a noisy speech mixture, written as:

$$y(m) = s(m) + n(m) \tag{1}$$

where $y(m)$ denotes the noisy speech mixture which is recorded using a single microphone, $s(m)$ and $n(m)$ represent the clean speech signal and noise at discrete time m, respectively. By using the short-time Fourier Transform (STFT), the spectrum of noisy speech mixture at time frame t and frequency bin f is represented as:

$$Y_{t,f} = S_{t,f} + N_{t,f} \tag{2}$$

where $S_{t,f}$ and $N_{t,f}$ are the STFT of the clean speech signal and noise, respectively. For convenience, the indices $t \in [1, T]$ and $f \in [1, F]$ are omitted hereafter, unless specified. The neural network model is trained to learn the mapping relation G_θ between the magnitude spectra of the clean speech signal $|S|$ and the noisy speech mixture $|Y|$, where G_θ is parametrized by θ. The mapping function is estimated by optimizing the loss function as:

$$Loss = \min_\theta \frac{1}{TF} \sum_1^T \sum_1^F |(G_\theta(|Y|) - |S|)|$$

$$= \min_\theta \frac{1}{TF} \sum_1^T \sum_1^F |(|\hat{S}| - |S|)| \tag{3}$$

where $|\hat{S}|$ is the magnitude spectrum of the estimated target speech, which is combined with the phase information of the noisy mixture to reconstruct the target speech.

3. System Description

3.1. Proposed Network Architecture

The proposed CFN is a convolutional encoder-decoder structure with multiple skip connections for monaural speech enhancement. The details of the proposed CFN are shown in Fig. 1. The proposed CFN takes the magnitude spectrum of the noisy mixture as input, and outputs the magnitude spectrum of estimated target speech. The estimated target speech is reconstructed using the estimated magnitude of the target speech and the phase information of the noisy speech mixture. The encoder has multiple layers of group convolutional fusion units (GCFU), and each unit includes standard convolution and depth-wise separable convolution. The number of output channels of the GCFU is increased from 16 to 128 in the encoder. The encoder is applied to reduce the dimension of input sequence by using the strides in GCFU.

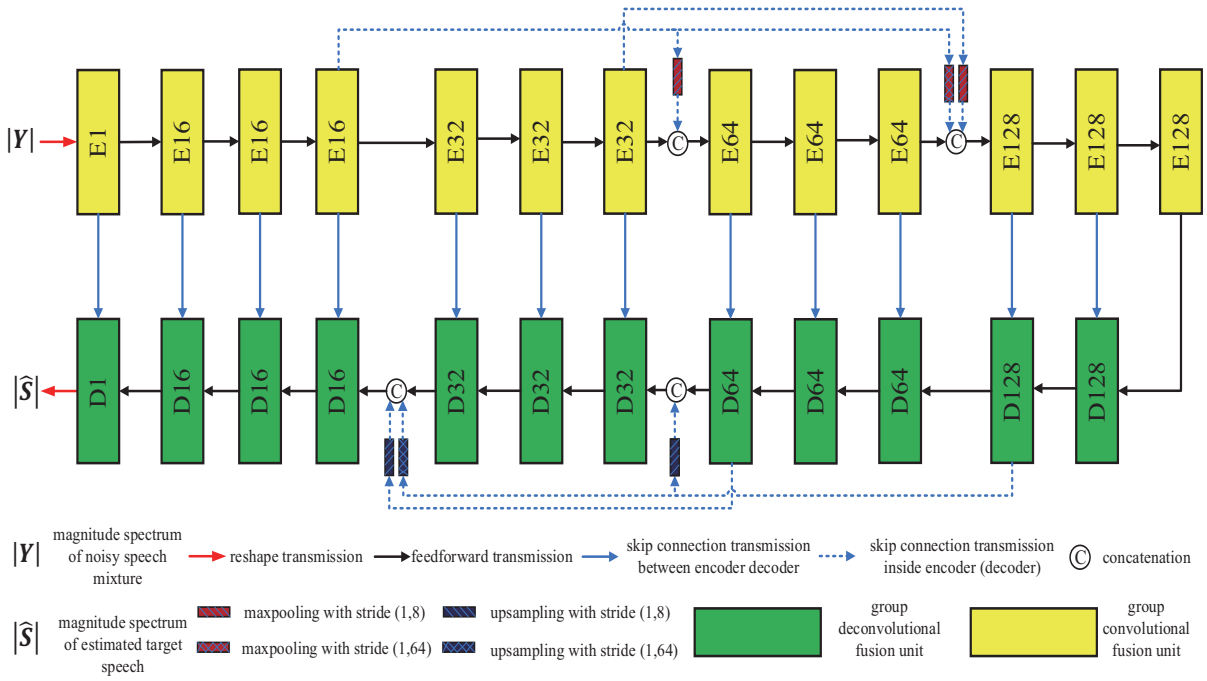

Figure 1: The architecture diagram of the proposed CFN. The components and their functions are listed at the bottom of Fig. 1. For example, $E64$ represents GCFU with 64 output channels in the encoder, and $D64$ represents GDFU in the decoder. The encoder is on the top of the figure, and the decoder is on the bottom of the figure. The kernel sizes of standard convolution and deconvolution are set to (1,3), and their strides sizes are set to (1,2). The kernel size of depth-wise separable convolution is set to (3,3), and stride size is (1,1). The pooling layer with stride size (1,2) is used to reduce the dimension of the depth-wise separable convolution output in GCFU, and upsampling layer with size (1,2) is employed to increase its dimension in GDFU. The last layer of the encoder uses stride size (1,1) for convolution and depth-wise separable convolution, and pooling with stride size (1,1) for depth-wise separable convolution.

The decoder has a mirror structure with the encoder, and each group deconvolutional fusion unit (GDFU) consists of standard deconvolution and depth-wise separable deconvolution. The decoder is used to recover the dimension of the encoder output and generate the final output. In addition, multiple types of skip connections are applied to improve feature reuse. More specifically, the GDFU is connected with the output from the corresponding symmetric GCFU by skip connection. Furthermore, the skip connections are used to connect different group convolutional/deconvolutional fusion units inside the encoder/decoder.

3.2. Group Convolutional Fusion Units

The proposed CFN employs GCFU including two different convolutions, i.e. standard convolution and depth-wise separable convolution. The outputs of the two convolutions are concatenated together. The proposed GCFU is shown in Fig. 2. For every GCFU, we set the input matrix $\mathbf{X}$ with height H, width W, and channel M, i.e. $\mathbf{X} \in \mathbb{R}^{H \times W \times M}$.

A standard 2D-convolutional layer can be characterized by an input $\mathbf{X}$, and a bank of

filters $\mathbf{F}$. More specifically, for the filter $\mathbf{F} \in \mathbb{R}^{K\times L\times M\times N}$, N represents the number of filters i.e. the number of output channels of the standard 2D-convolutional layer. The operation of the standard 2D-convolutional layer is:

$$\mathbf{C}_{(k,l,n)} = \sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{m=1}^{M}\mathbf{F}_{(i,j,m,n)}\mathbf{X}_{(k+i-1,l+j-1,m)} \tag{4}$$

The output of the standard 2D-convolutional layer is $\mathbf{C} \in \mathbb{R}^{H\times W\times N}$. Also, we can use the stride sizes to control the output size of $\mathbf{C}$. The batch-normalization and activation function LeakyReLU [30] are followed to generate the 2D-convolution output.

Unlike the standard 2D-convolution, the depth-wise separable convolution has two steps: depth-wise convolution i.e. a spatial convolution performed independently over every input channel, and the point-wise convolution i.e. a standard convolution, which projects every channel's output of the depth-wise convolution to a new channel space. Mathematically, we can spilt the filter $\mathbf{F}$ into two filters, the depth-wise filter $\mathbf{D} \in \mathbb{R}^{K\times L\times 1\times 1}$, and point-wise filter $\mathbf{P} \in \mathbb{R}^{1\times 1\times M\times N}$.

$$\begin{aligned}\mathbf{S}_{(k,l,n)} &= \sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{m=1}^{M}\mathbf{F}_{(i,j,m,n)}\mathbf{X}_{(k+i-1,l+j-1,m)} \\ &= \sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{m=1}^{M}\mathbf{D}_{(i,j,m)}\mathbf{P}_{(m,n)}\mathbf{X}_{(k+i-1,l+j-1,m)} \\ &= \sum_{m=1}^{M}\mathbf{P}_{(m,n)}\left(\sum_{i=1}^{K}\sum_{j=1}^{L}\mathbf{D}_{(i,j,m)}\mathbf{X}_{(k-i,l-j,m)}\right)\end{aligned} \tag{5}$$

The output of the depth-wise separable convolutional layer is $\mathbf{S} \in \mathbb{R}^{H\times W\times N}$. Similarly, the batch-normalization and activation function LeakyReLU [30] are followed after depth-wise separable convolutional layers. In addition, the max pooling operation is used to down-sample the output of the depth-wise separable convolution. Different from the conventional residual structure that sums two output convolutions [13], the convolutional fusion is realized by concatenating the weighted outputs of the two convolutions:

$$\mathbf{B} = [\alpha_1\mathbf{C}, \alpha_2\mathbf{S}] \tag{6}$$

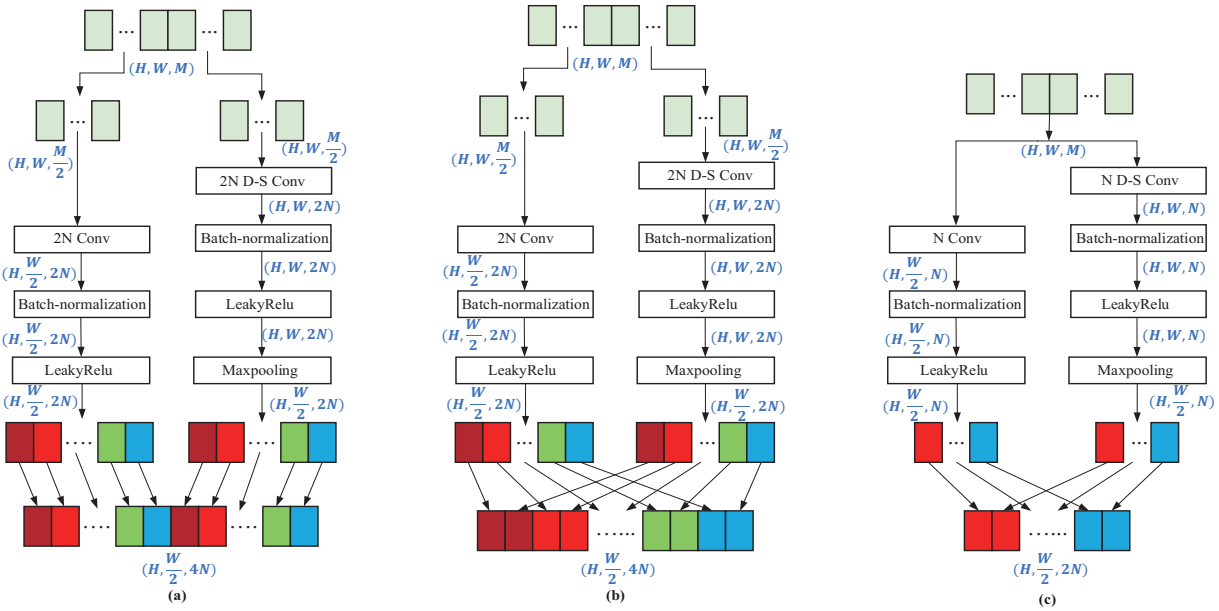

Figure 2: (a) Group convolution that consists of the standard convolution and depth-wise separable convolution. $2N$ represents the number of output channels of convolutional layers. The standard convolution and depth-wise separable convolution take half input sequence as input, and each of them generates output with $2N$ channels. Then outputs of convolution and depth-wise separable convolution are concatenated directly. (b) Group convolution with channel shuffle. The input sequence is divided into two parts based on channel index, where the first part of the input sequence is feed to the standard convolution, and the second part of the input sequence is fed to the depth-wise separable convolution layer. Their outputs are re-arranged and concatenated using the channel shuffle, and final output has $4N$ channels. (c) Proposed GCFU with channel shuffle. The full input sequence (full information) is fed to the N channel standard convolution and N channel depth-wise separable convolution. Their outputs are re-arranged and concatenated using the channel shuffle, and final output has dimension $2N$.

where α_1 and α_2 represent the weight parameters of the standard and depth-wise separable convolutions, respectively. They can assign weights to two kinds of convolutions.

3.3. Channel Shuffle

Group convolution is motivated by the original idea of AlexNet [24], where two convolution filter groups are employed in parallel in each layer to improve network efficiency. In Fig. 2(a), the input sequence is distributed to parallel convolutions. For the next layer with similar structure, the output of a particular channel is only related to a small fraction of the input channels, and information flow between channels is limited [27]. Therefore, we introduce channel shuffle to re-arrange group channels in the proposed CFN model, which entangles the outputs of the two kinds of convolutions. In addition, the input and output of this layer will be related. In Fig. 2(b), the input sequence is divided into two parts based on channel index, where the first part is fed to the standard convolution, and the second part is fed to the depth-wise separable convolution. As a result, neither the standard convolution nor depth-

wise separable convolution has utilized the full input sequence, which may limit their model performance due to the use of partial input. To address this problem, in our proposed GCFU, we design a new structure to exploit the full sequence and channel shuffle, as shown in Fig. 2(c). The full input sequence is fed to both standard convolution and depth-wise separable convolution. They are employed to generate different feature maps for the full input sequence. The outputs of standard convolution and depth-wise separable convolution are re-arranged and concatenated according to their channel numbers.

Both the standard convolution and depth-wise separable convolution have N output channels, and they can be represented as $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2..., \mathbf{C}_N]$ and $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_N]$. The full information channel shuffle of GCFU is:

$$\mathbf{B}_s = [\alpha_1\mathbf{C}_1, \alpha_2\mathbf{S}_1, ..., \alpha_1\mathbf{C}_N, \alpha_2\mathbf{S}_N] \tag{7}$$

where $\mathbf{B}_s$ represents the channel shuffled group convolution. By using the full information channel shuffle, the output of the standard convolution and depth-wise convolution are fully related, and the next layers can obtain the shuffled information flow.

3.4. Group Deconvolutional Fusion Units

For convolutional encoder-decoder, the decoder is exploited to map the low dimension encoder output to a higher dimension that is equal to the original dimension of the input sequence. The standard decoder uses deconvolutional layers to up-sample the encoder output. However, there is no depth-wise separable deconvolution structure. To address this issue, we propose group deconvolutional fusion unit (GDFU) to up-sample the encoded feature map, and generate the input to the next GDFU layer. The GDFU architecture is shown in Fig. 3. The lower dimensional feature map is fed to deconvolutional and depth-wise separable convolutional layers respectively, to generate the dense feature maps. Batch-normalization and LeakeyRelu [30] steps are followed. Inspired by the work [31], which uses the transferred pooling layers and standard convolutional layers to build the convolutional decoder, we use an upsampling layer to up-sample feature map of the depth-wise separable convolutional layer. Finally, channel shuffle is exploited to re-arrange outputs of the two streams as shown in Fig. 3.

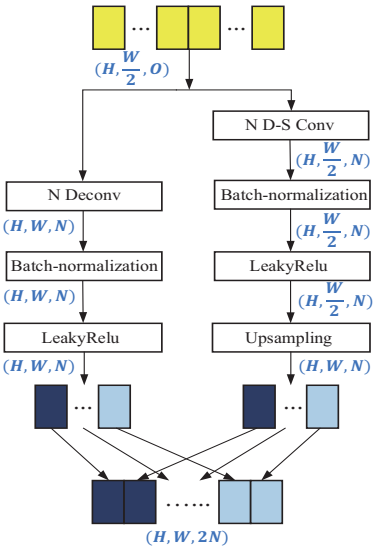

Figure 3: Structure diagram of the proposed GDFU with channel shuffle. The full input sequence is fed to the standard deconvolution and depth-wise separable convolution. N Deconv represents N channel standard deconvolution, and N D-S Conv denotes the N channel depth-wise separable convolution.

3.5. Skip Connection inside Encoder or Decoder

The input sequence is processed with many layers in convolutional encoder-decoder. Some information may be lost due to the variations in the dimension of feature representation of the signal [20]. To address this issue, the skip connections between the encoder and decoder are introduced to improve the feature reuse. This is achieved by connecting the encoder layers with their corresponding decoder layers. Nevertheless, the feature reuse within the encoder/decoder has not been explored, despite its potential benefits in boosting the enhancement performance. On the other hand, densely connecting all the layers inside the encoder/decoder will substantially increase the computational cost. Here, we propose block dense connections to facilitate the feature reuse inside the encoder/decoder, as shown in Fig. 1, where the encoder layers with the same number of output channels are set as a block, e.g. block-16, block-32, block-64 and block-128. For instance, the output of block-16 is fed to the other blocks (block-32, block-64 and block-128) of the encoder. Since GCFU has stride size $(1, 2)$ in the encoder layers, the output sizes of different layers are varied, and as a result, the output of block-16 cannot be directly concatenated with the output of the other blocks. Therefore, we design a new mechanism to down-sample the features of block-16 using a max-pooling layer with stride size (1, 8). Then, the down-sampled output is concatenated with the output block-32, the concatenated representation is fed to block-64. Similarly, we develop other skip connections within the encoder, as shown in Fig. 1. On the contrary, the layers are up-sampled in the

decoder to match the size of the skip connections.

4. Experiments

4.1. Data and Setup

We use clean utterances from TIMIT [32] and IEEE [33] corpora, together with the environmental noises from the NOISEX-92 [34] and Non-Speech Sounds [35] datasets to build training and testing datasets. The TIMIT database consists of 6300 utterances, spoken by 630 male and female speakers, and the IEEE corpus contains 720 utterances spoken by a male speaker. The Non-Speech Sounds dataset contains 100 environmental noises. For speaker-independent case, We randomly select 1500 utterances from TIMIT and IEEE corpora as the training utterances. In addition, we choose 100 utterances from TIMIT corpus as the testing utterances, and the speakers of testing utterances are different from the speakers of training utterances, which represents a speaker-independent case. The training and testing utterances are mixed with the Babble, Artillery, Airplane, Factory, Tank, and White noises from NOISEX-92 [34] dataset. The noises' names indicate their recording environments, and they are four minutes long. The training and testing datasets are generated with three signal-to-noise ratio (SNR) levels i.e. -5dB, 0dB and 5dB. Furthermore, we also evaluate our proposed method with speaker- and noise-independent cases. These experiments aim to assess the performance of the proposed method under challenging mismatch conditions. We randomly select 200 utterances from the TIMIT corpus [32] as the training utterances, and 100 utterances from the TIMIT corpus as the testing utterances, and the speakers of testing utterances are different from the speakers of training utterances. Three types of unseen noise i.e. Water, Wind and Pink noises, are chosen as the testing noises from the Non-Speech Sounds [35] and NOISEX-92 [34] datasets. We use 108 noises in training, which consist of 98 noises from Non-Speech Sounds dataset and 10 noises from NOISEX-92 dataset. The training noises are mixed with the training speeches in SNR levels of -5dB, 0dB and 5dB. Similarly, the testing speeches are mixed with three unseen environmental noises at SNR levels of -5dB, 0dB and 5dB to generate the testing dataset. In total, we use about 60 hours ($1500\times6\times3\times2.5\div3600+200\times100\times3\times2.5\div3600=60.42$) noisy speech mixtures to train our model, and about 2 hours noisy speech mixtures to test our model.

In additional experiments, we compare the proposed CFN methods with ShuffleNet [27],

Conv-TasNet [23] and TCNN [22] methods. For training dataset, we used 15660 noisy mixtures for each SNR level (-5dB, 0dB, 5dB). These mixtures were generated by mixing 580 clean utterances from TIMIT [32] and VCTK [36] corpora with 27 noises from NOISEX-92 [34] and Non-speech Sound [35] databases. The testing dataset includes 240 noisy mixtures with 6 noises for each SNR level. In total, 46980 noisy mixtures are exploited to train the proposed CFN and baseline methods, and 720 noisy mixtures are used to test baseline and proposed methods. The testing speakers are unseen in the training set.

The parameters of the CFN are shown in Fig. 1. The signals in the training and testing datasets are re-sampled at 16 kHz. The magnitude spectrum of these signals is obtained using Short Time Fourier Transform (STFT) with Hanning window of 512 samples and 50% overlap between the neighboring windows, and then log-compressed. The MAE is used as the cost function for the baselines (discussed in the next sub-section) and the proposed CFN methods. The Adam optimization algorithm with 0.0001 initial learning rate [37] is employed. The best models are selected. For quantitative evaluation, short-time objective intelligibility (STOI) [38] and perceptual evaluation of speech quality (PESQ) [39] are used to measure the enhancement performance. The STOI indicates the intelligibility quality of the estimated target speech, which ranges in (0, 1), and the PESQ shows the perceptual quality of the estimated speech which ranges in (0, 4.5). The higher value of the measurements indicates better enhancement performance.

4.2. Baseline Methods

We use three state-of-the-art methods as the baselines, and they are, respectively, the DNN in [7], GRN in [13], and AECNN in [20]. DNN is a fundamental method in deep learning, and the GRN and AECNN show advantages over RNN. DNN has four hidden layers, and each hidden layer has 1024 units. Also, the dropout with a rate of 0.2 is used in DNN to reduce the over-fitting [40]. The output layer of DNN has the same number of units as the length of the input sequence. The GRN model is a 62-layered deep fully connected convolutional model with residual connections. The stacked convolutional layers use gated convolution with an increased dilated ratio. The dilated convolution offers larger receptive fields, which enables each kernel to filter out information on longer-term of the sequence than standard convolution. The Sigmoid activation function follows the dilated convolution to build a gate mechanism to control the information flow in GRN. Finally, the prediction module takes the information

flow from the stacked dilated convolution layers by the feed-forward and skip connections, and generates the magnitude spectrum of the estimated target speech.

The AECNN is an 18-layered convolutional encoder-decoder structure. The convolutional encoder is exploited to reduce the dimension of the input magnitude spectrum by using the convolutional layers with strides sized 2. The deconvolutional decoder has a mirror structure with the convolutional encoder, which is employed to recover the dimension of the output of the convolutional encoder to the original dimension i.e same as the input noisy speech magnitude spectrum. The number of output channels of the convolutional encoder is increased from 64 to 256, but the number of channels of the convolution decoder is reduced from 256 to 64, and the output layer of the AECNN has one channel. The layers of the encoder are connected with layers of the decoder that have the same number of the output channels by skip connections. The MAE between the magnitude spectrum of noisy speech and that of the estimated target speech is employed in AECNN. Magnitude spectrum of 257 units are fed into the baseline methods and the proposed CFN, and they output the magnitude of estimated target speech. The same training and testing datasets are employed for the baselines and the proposed method. The number of parameters for the baseline methods are respectively, DNN (5.5 Million), GRN (2.5 Million), AECNN (6.4 Million), and the number of parameters of the proposed CFN is 3.5 Million.

In additional experiments, we introduce three baseline methods. They are ShuffleNet [27], TCNN [22] and Conv-TasNet [23]. The ShuffleNet was originally proposed for computer vision problems, which was built by 15 stacking convolutional shuffle units. We have modified the structure of ShufffleNet to fit the length of the speech signal. The TCNN is a 40-layered convolutional encoder-decoder model, operated in time domain. They are connected using dilated convolutional layers to enlarge the receptive fields. The Conv-TasNet is a convolutional encoder-decoder model, which includes three modules: encoder, separation, and decoder module. The separation module estimates the mask to separate the target speech. The Conv-TasNet is trained and tested in time domain. The number of parameters of these baseline methods are, respectively, ShuffleNet (4.0 Million), TCNN (5.1 Million), and Conv-TasNet (5.1 Million).

4.3. Experimental Results for Seen Noises

Tables 1 and 2 provide comparisons among the proposed CFN and the baseline methods in terms of STOI and PESQ for speaker-independent case with seen Babble, Artillery, Airplane, Factory, Tank, and White noises. The DNN offers, on average, STOI = 75.39% and PESQ = 2.14, which provides the lowest improvement over the noisy speech mixture across all compared methods. The results show that the DNN provides limited enhancement performance for speaker-independent speech enhancement, where the speakers in the test set are different

Table 1: Speech enhancement performance comparison in terms of STOI and PESQ for speaker-independent case with Babble, Artillery, Airplane noises. *Italic* text is the proposed method. **Bold** number indicates the best performance.

Measures	STOI(%)											
Noises	Babble				Artillery				Airplane			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	54.62	63.82	72.47	63.64	65.34	73.02	79.66	72.67	54.57	64.13	73.42	64.04
DNN	68.89	72.98	79.69	73.85	74.62	79.29	82.92	78.94	68.18	74.85	80.48	74.50
GRN	69.76	76.89	81.42	76.02	77.80	82.31	85.10	81.74	72.70	78.60	83.10	78.13
AECNN	72.01	77.78	82.51	77.43	79.62	83.68	86.59	83.30	73.87	77.99	84.43	78.76
CFN	**75.67**	**80.33**	**83.85**	**79.95**	**81.81**	**85.88**	**87.43**	**85.04**	**77.55**	**82.15**	**85.56**	**81.77**
Measures	PESQ											
Noises	Babble				Artillery				Airplane			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	1.37	1.63	1.92	1.64	1.66	1.94	2.18	1.93	1.36	1.59	1.87	1.61
DNN	1.77	2.08	2.34	2.06	2.10	2.33	2.54	2.32	1.78	2.12	2.37	2.09
GRN	1.86	2.16	2.42	2.15	2.20	2.47	2.70	2.46	1.93	2.25	2.55	2.24
AECNN	1.92	2.19	2.45	2.19	2.32	2.55	2.75	2.54	2.03	2.32	2.57	2.31
CFN	**2.16**	**2.41**	**2.62**	**2.40**	**2.49**	**2.68**	**2.86**	**2.68**	**2.24**	**2.51**	**2.73**	**2.49**

Table 2: Speech enhancement performance comparison in terms of STOI and PESQ for speaker-independent case with Factory, Tank, White noises. *Italic* text is the proposed method. **Bold** number indicates the best performance.

Measures	STOI(%)											
Noises	Factory				Tank				White			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	54.17	63.57	74.07	63.94	72.24	76.17	79.54	75.98	53.26	62.56	72.42	62.75
DNN	62.00	69.76	76.09	69.28	80.34	82.67	83.14	82.05	67.28	74.10	79.90	73.76
GRN	68.06	74.98	80.42	74.49	81.37	83.86	85.42	83.55	72.32	78.36	82.50	77.73
AECNN	69.72	75.77	81.72	75.74	83.57	84.67	86.17	84.48	73.11	79.34	83.00	78.48
CFN	**71.61**	**78.19**	**86.20**	**78.67**	**84.64**	**86.26**	**87.31**	**86.07**	**76.29**	**81.01**	**85.02**	**80.77**
Measures	PESQ											
Noises	Factory				Tank				White			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	1.33	1.61	1.88	1.61	1.76	1.99	2.21	1.99	1.13	1.31	1.57	1.34
DNN	1.67	1.97	2.22	1.95	2.37	2.57	2.60	2.51	1.72	2.05	1.95	1.91
GRN	1.78	2.09	2.35	2.07	2.43	2.60	2.74	2.59	1.81	2.17	2.38	2.12
AECNN	1.80	2.10	2.40	2.10	2.58	2.76	2.83	2.72	1.95	2.25	2.40	2.20
CFN	**1.98**	**2.24**	**2.63**	**2.28**	**2.72**	**2.86**	**2.99**	**2.86**	**2.20**	**2.44**	**2.64**	**2.63**

Table 3: The p-value of the t-test at 5% Significance Level, and comparison of proposed method with the baseline methods for speaker-independent case. H_0 denotes the null hypothesis, and (+) indicates the improvement of two pairs is statistically significant at the 95% confidence level.

Measures	STOI		PESQ	
	p-value	H_0	p-value	H_0
Noisy	1.63E-10	(+)	1.29E-14	(+)
DNN	1.75E-10	(+)	5.34E-12	(+)
GRN	4.33E-10	(+)	1.85E-12	(+)
AECNN	1.21E-07	(+)	5.53E-12	(+)

from those in the training set, compared with other methods, including the proposed method.

The GRN provides, on average, STOI = 78.69% and PESQ = 2.27. The GRN offers further improvements over the DNN methods. The GRN utilizes the dilated convolutional layers to enlarge the receptive fields, which means one kernel (filter) can take information from a larger region and generate the output. Therefore, the temporal information from the long-term frames is captured. The convolutional layer with Sigmoid is employed to build the gate mechanism to control the information flow in GRN. Besides, residual learning is employed by using the skip connections among the different layers of GRN. By joint using these strategies, the GRN offers a better enhancement performance in terms of STOI and PESQ than DNN in the speaker-independent speech enhancement.

The AECNN provides, on average, STOI = 79.75% and PESQ = 2.34, which outperforms GRN and DNN methods, which is consistent with the finding in [20]. The AECNN employs a speech encoder-decoder structure to estimate the magnitude spectrum of target speech. The convolutional encoder takes the magnitude spectrum of the noisy speech mixture as input, which generates an output of lower dimension. The convolutional decoder is utilized to recover the dimension of the encoder output. In addition, MAE between the magnitude spectra of the estimated target speech and the original target speech is used as the cost function. The experimental results show that the AECNN i.e. convolutional encoder-decoder is an advanced method over DNN and GRN methods.

The proposed CFN method offers, on average, STOI = 82.20% and PESQ = 2.52, which provides 2.45% STOI improvement and 0.17 PESQ improvement over the AECNN, GRN and DNN methods. These results prove that the CFN shows advantages in processing speaker-independent speech enhancement. Meanwhile, the CFN uses fewer parameters, thus offering a higher parameter efficiency. The reason will be discussed in the next subsection.

To further evaluate whether the improvement in terms of STOI and PESQ is statistically

Table 4: Speech enhancement performance comparison in terms of STOI and PESQ for speaker- and noise-independent cases with Water, Wind and Pink noises. *Italic* text is the proposed method. **Bold** number indicates the best performance.

Measures	STOI(%)											
Noises	Water				Wind				Pink			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	56.10	67.18	77.22	66.83	71.50	77.37	82.35	77.07	53.75	63.17	72.67	63.20
DNN	72.36	78.08	82.95	77.80	74.75	80.68	84.70	80.04	60.91	66.64	74.97	67.51
GRN	74.85	80.00	86.04	80.29	76.55	82.24	87.13	82.01	63.79	70.97	76.37	70.37
AECNN	76.76	81.78	87.09	81.88	78.72	84.55	87.68	83.65	64.97	71.45	78.65	71.69
CFN	**79.58**	**84.12**	**88.22**	**83.98**	**82.57**	**86.32**	**88.90**	**85.93**	**69.33**	**75.60**	**82.02**	**75.65**
Measures	PESQ											
Noises	Water				Wind				Pink			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	1.22	1.36	1.60	1.40	1.54	1.75	1.95	1.75	1.38	1.44	1.75	1.52
DNN	1.77	1,97	2.15	1.96	1.76	1.99	2.18	1.98	1.56	1.79	2.06	1.80
GRN	1.79	2.00	2.28	2.02	1.82	2.02	2.24	2.02	1.66	1.90	2.18	1.91
AECNN	1.86	2.05	2.32	2.07	1.89	2.09	2.27	2.08	1.79	2.02	2.26	2.02
CFN	**2.03**	**2.22**	**2.44**	**2.23**	**2.04**	**2.27**	**2.46**	**2.26**	**1.92**	**2.20**	**2.45**	**2.19**

significant, we compare the performance of the proposed CFN with baseline methods and noisy speech mixture using t-test at a significant level of 0.05 in Table 3. The t-test is performed following statistical analysis in [41]. When p-values smaller than 0.05, it means there is statistical significant difference between the results of the two groups. We observed all p-values are smaller than 0.05, and all H_0 are +, which confirms that the improvements by the proposed CFN over the baselines are statistically significant.

4.4. Experimental Results for Unseen Noises

Table 4 provides comparisons among the proposed CFN and baseline methods in terms of STOI and PESQ for speaker- and noise-independent cases with unseen Water, Wind, Pink noises. Similarly, experimental results of speaker- and noise-independent cases show similar trends of enhancement performances. The DNN offers the worst enhancement performance in terms of STOI and PESQ, which demonstrates the limitation of DNN in processing challenging speech enhancement problems. The GRN provides improvements over the DNN method. Also, the AECNN outperforms DNN and GRN methods, which yields, on average, STOI = 79.07% and PESQ = 2.06.

The proposed CFN method yields the best enhancement performance, on average, STOI = 81.85% and PESQ = 2.25. Several contributions are exploited to boost enhancement performance. The CFN uses standard convolution and depth-wise separable convolution to produce different feature maps, reinforcing the model capacity of the proposed CFN method. More-

Table 5: The p-value of the t-test at 5% Significance Level, and comparison of the proposed method with the baseline methods for speaker- and noise-independent cases. H_0 denotes the null hypothesis, and (+) indicates the improvement of two pairs is statistically significant at the 95% confidence level.

Measures	STOI		PESQ	
	p-value	H_0	p-value	H_0
Noisy	7.03E-5	(+)	1.04E-06	(+)
DNN	1.21E-06	(+)	2.68E-07	(+)
GRN	2.24E-05	(+)	8.63E-08	(+)
AECNN	1.45E-04	(+)	5.55E-8	(+)

over, a novel decoder that consists of deconvolution and depth-wise separable convolution is employed to up-sample the encoder output. In addition, full information channel shuffle structure is designed to reduce the number of parameters and exploit the relations across the channels. Also, two types of skip connections are introduced to enhance the feature reuse, especially intra skip connections within the encoder/decoder, which makes proceeding layers of the encoder receive more information from previous layers of the encoder/decoder. With the contributions above, the CFN shows advantages over the DNN, GRN and AECNN for noise- and speaker-independent cases. In addition, the results of the t-test in Table 5 demonstrate that the proposed CFN method yields statistically significant improvements over the baseline methods.

4.5. Additional Experiments

We perform the additional experiments to compare the proposed CFN with ShuffleNet, TCNN and Conv-TasNet methods. The results are shown in Table 6. All methods offer improvements over the noisy speech mixtures, which show that they are feasible to address the speech enhancement problem. The ShuffleNet provides the lowest improvements over the other methods. The TCNN provides improvements over the ShuffleNet in terms of STOI and PESQ. The TCNN exploits the encoder to generate a low dimensional representation of the input. The temporal convolutional module used the dilated convolution to enlarge the

Table 6: Speech enhancement performance comparison in terms of STOI and PESQ. *Italic* text is the proposed method. **Bold** number indicates the best performance.

Noises	STOI				PESQ			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	59.21	68.41	76.94	68.19	1.28	1.59	1.71	1.53
ShuffleNet	68.24	75.75	81.30	75.10	1.56	1.89	2.11	1.85
TCNN	71.46	78.75	83.48	77.84	1.87	2.04	2.27	2.06
Conv-TasNet	71.02	80.10	85.11	78.74	1.61	1.96	2.29	1.95
CFN	**75.71**	**81.19**	**85.29**	**80.73**	**1.94**	**2.17**	**2.37**	**2.16**

receptive fields. Then, the decoder is used to reconstruct the enhanced frame. Furthermore, the Conv-TasNet offers improvements over the TCNN and ShuffleNet in terms of STOI, which is also operated in the time-domain. The Conv-TasNet uses the linear encoder to generate the representation of speech, which is fed to separation module to produce the mask. Finally, the weighted encoder output is converted to the speech using the decoder. The proposed CFN offers the most significant improvements over the baseline methods.

4.6. Ablation Analysis and Scalar Parameters

We perform the ablation analysis in Table 7 to show the contribution of every component in the proposed CFN. Full denotes the results of the proposed CFN method using all the components. No SC denotes deleting the standard convolution from the proposed method. No D-SC represents ablating the depth-wise separable convolution. No GDFU Decoder represents replacing the GDFU decoder by the standard deconvolutional decoder. No FICS represents the proposed method without using the full information channel shuffle. No CS denotes removing channel shuffle in the proposed method. No ISCED denotes removing the intra-skip connections of encoder/decoder.

Table 7: Ablation analysis in terms of STOI, PESQ and number of parameters.

Measures	STOI	PESQ	No. of Parameters(Million)
Full	70.18	1.73	3.5
No SC	65.89	1.57	1.2
No D-SC	66.12	1.55	0.6
No GDFU Decoder	68.31	1.64	1.8
No FICS	70.36	1.71	6.6
No CS	69.54	1.72	3.5
No ISCED	69.31	1.70	3.1

We observed that the standard convolution yields the most improvements in terms of STOI and PESQ, which proves the standard convolution has a better model capacity over the depth-wise separable convolution in the proposed CFN. Meanwhile, the depth-wise separable convolution has similar importance when compared with standard convolution. However, we observed that there are about 4% STOI and 0.2 PESQ performance decrease when using the standard convolution or depth-wise separable convolution. These results confirm the standard convolution and depth-wise separable convolution are limited in processing mismatch speech enhancement, but the proposed CFN is capable to provide a better model capacity for speech enhancement. In addition, we replace the GDFU decoder by standard decoder, which seems to result in lower STOI and PESQ scores. These results show that the proposed GDFU offers

better performance when compared with the standard convolutional decoder. The number of channels used in the method No FICS has been doubled as compared with that in the proposed FCN, as shown in Fig. 2 (a). As a result, the proposed FCN offers a reduction of 3.1 million parameters, compared with the method No FICS. In addition, the proposed FCN slightly boosts the PESQ performance and maintains the STOI performance. Therefore, the proposed full information channel shuffle helps improve the parameter efficiency while maintaining the enhancement performance.

The channel shuffle re-arranges the outputs of two convolutional layers, which makes them related and improves enhancement performance. The layers of CFN may not well reconstruct the input sequence, with the intra skip connections of encoder or decoder, more information from previous layers is exploited due to the feature reuse in the proposed CFN model. However, the performance improvements by the intra skip connections are relatively small as compared with those by convolutional/decovolutional fusion units and channel shuffle.

Table 8: Scale Parameters Analysis in terms of STOI and PESQ

Scale Parameters	STOI	PESQ
Unprocessed	50.76	1.19
$\alpha_1 = \alpha_2 = 1$ (CFN)	70.18	1.73
$\alpha_1 = \alpha_2 = 0.5$	69.77	1.71
$\alpha_1 = \alpha_2 = 2$	69.97	1.73
$\alpha_1 = \alpha_2 = 1.5$	69.89	1.71
$\alpha_1 = 0.5, \alpha_2 = 1$	69.62	1.71
$\alpha_1 = 1, \alpha_2 = 0.5$	70.14	1.72

We evaluate the performance of the proposed CFN method on speech enhancement tasks as a function of varied scale parameters. Table 8 shows the enhancement performance in terms of STOI and PESQ, which is used to demonstrate the performance differences caused by adjusting α_1 and α_2. From this table, we can observe that the best performance is obtained by using $\alpha_1 = \alpha_2 = 1$, as employed in the proposed CFN method, which drops for other values of the scale parameters.

Fig. 4 shows the spectra of target speech, noisy mixture and enhanced speech of different methods. DNN, GRN, AECNN and the proposed CFN remove most of the noise from the noisy mixture, while some noise in the low frequency region remains in the enhanced speech by DNN, GRN and AECNN. The enhanced spectrum of CFN appears to be most similar to that of target speech, which further confirms its improved performance.

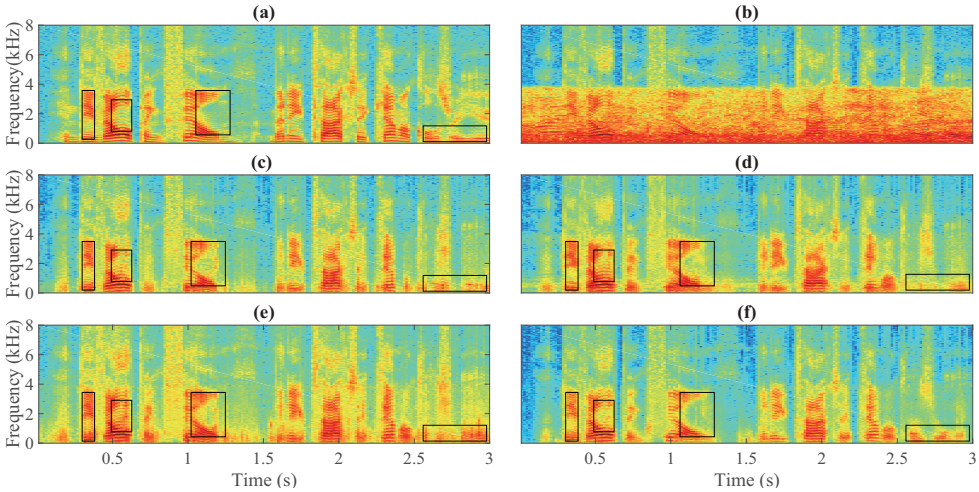

Figure 4: Spectra of different signals: (a) target speech, (b) noisy speech mixture, (c) enhanced speech by DNN, (d) enhanced speech by GRN, (e) enhanced speech by AECNN, (f) enhanced speech by CFN.

4.7. Time Domain and T-F Domain Compassion

We have also performed experiments to compare the performance difference of AECNN in time-domain and time-frequency (T-F) domain, as shown in Table 9. The AECNN in the T-F domain performs better than that in the time domain in terms of PESQ. These results show the advantage of applying AECNN in the T-F domain. The proposed CFN offers improvements over AECNN-T and AECNN-TF in terms of STOI and PESQ.

Table 9: Speech enhancement performance comparison between the time domain and T-F domain. *Italic* text is the proposed method. **Bold** number indicates the best performance.

Noises	STOI				PESQ			
SNR / Methods	-5dB	0dB	5dB	Avg.	-5dB	0dB	5dB	Avg.
Noisy Mixture	61.44	70.14	78.27	69.95	1.42	1.57	1.77	1.59
AECNN-T	72.18	79.33	84.33	78.61	1.58	1.90	2.22	1.90
AECNN-TF	72.78	79.01	83.82	78.54	1.79	2.01	2.22	2.01
CFN	**75.78**	**81.38**	**85.34**	**80.83**	**1.89**	**2.15**	**2.31**	**2.11**

4.8. Depth Multiplier of Depth-wise Separable Convolution

We also perform experiments to analyze the effects of depth multiplier in depth-wise separable convolution. The depth multiplier represents the number of depth-wise convolution output channels for each input channel. These experiments aim to find the balance between speech enhancement performance and depth multiplier i.e parameter efficiency. The experimental results are shown in Fig. 5. With the increase in D value, speech enhancement performance in terms of STOI and PESQ is improved. However, a larger number of parameters is needed, which means it will need more computational resource. $D = 1$ offers, on

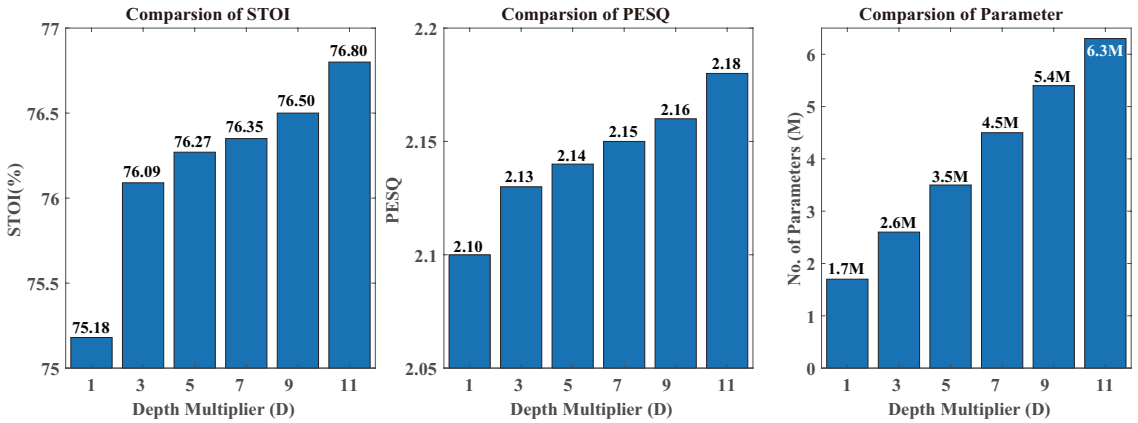

Figure 5: The STOI performance, PESQ performance and the number of parameters with different depth multipliers. The depth multiplier D is changed from 1 to 11 with a step of 2.

average, STOI = 75.18% and PESQ = 2.10, which provides the lowest enhancement performance but requires the smallest number of parameters around 1.7 Million. However, when we set $D = 11$, it provides, on average, STOI=76.83% and PESQ = 2.17, which offers the highest enhancement performance but needs more parameters around 6.3 Million. When D is increased to 3, significant improvements are observed in terms of STOI and PESQ. If D larger than 5, the improvements of STOI and PESQ become stable. For example, $D = 5$ offers on average STOI=76.27%, and $D = 7$ offers STOI= 76.35%. In summary, considering the number of parameters, memory size and enhancement performance, we select $D = 5$ in the proposed CFN model.

5. Conclusions

We have presented a novel convolutional model, named convolutional fusion network (CFN), to address the monaural speech enhancement problem. Speech enhancement was considered as a sequence-to-sequence problem by the CFN, where the magnitude spectrum of the noisy speech mixture is taken as the input, for estimating the magnitude spectrum of the target speech. The proposed CFN model improves the model capacity, inter-channel dependency, parameter efficiency and feature reuse. With the proposed group convolutional fusion units, the standard convolution and depth-wise separable convolution were used to reinforce the model capacity of CFN. Then, the novel decoder allowed the CFN to take the advantages of two different convolutions. This has been confirmed by the experimental results that the group convolutional model had better model capacity than standard convolution. The channel

shuffle structure is introduce to exploit the information about inter-channel dependency. In addition, utilizing skip connections inside the encoder and decoder can promote feature reuse and improve the performance.

Interesting aspects for future study include the use of adaptive weights in the depth-wise separable and vanilla convolutions for each time-frequency point, and the use of spatial and contextual information under the current framework.

References

[1] D. L. Wang, Deep learning reinvents the hearing aid, IEEE Spectrum March Issue (2017) 32–37.

[2] P. C. Loizou, Speech Enhancement : Theory and Practice, CRC Press, 2013.

[3] S. M. Naqvi, M. Yu, J. A. Chambers, A multimodal approach to blind source separation of moving sources, IEEE Journal of Selected Topics in Signal Processing 4 (2010) 895–910.

[4] Y. Sun, Y. Xian, W. W. Wang, S. M. Naqvi, Monaural source separation in complex domain with long short-term memory neural network, IEEE Journal of Selected Topics in Signal Processing 13 (2019) 359–369.

[5] Y. X. Wang, D. L. Wang, Towards scaling up classification-based speech separation, IEEE Transactions on Audio, Speech, and Language Processing 21 (2013) 1381–1390.

[6] K. Han, Y. X. Wang, D. L. Wang, W. S. Woods, I. Merks, T. Zhang, Learning spectral mapping for speech dereverberation and denoising, IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (2015) 189–198.

[7] Y. Xu, J. Du, L. R. Dai, C. H. Lee, A regression approach to speech enhancement based on deep neural networks, IEEE Transactions on Audio, Speech, and Language Processing 23 (2015) 7–19.

[8] Y. Jiang, D. L. Wang, R. Liu, Z. Feng, Binaural classification for reverberant speech segregation using deep neural networks, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (2014) 2112–2121.

[9] D. L. Wang, Time-frequency masking for speech separation and its potential for hearing aid design, Trends in Amplication 12 (2008) 332–351.

[10] Y. X. Wang, A. Narayanan, D. L. Wang, On training targets for supervised speech separation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (2014) 1849–1858.

[11] X. L. Zhang, D. L. Wang, A deep ensemble learning method for monaural speech separation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 24 (2016) 967 – 977.

[12] Y. Xu, J. Du, L. R. Dai, C. H. Lee, An experimental study on speech enhancement based on deep neural networks, IEEE Signal Processing Letters 21 (2014) 65–68.

[13] K. Tan, D. L. Wang, Gated residual networks with dilated convolutions for monaural speech enhancement, IEEE/ACM Transactions on Audio, Speech, and Language Processing 27 (2019) 189–198.

[14] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Comput, 1997.

[15] J. T. Chen, D. L. Wang, Long short-term memory for speaker generalization in supervised speech separation, The Journal of the Acoustical Society of America 141 (2017) 4705–4714.

[16] P. S. Huang, M. Kim, M. H. Johnson, P. Smaragdis, Joint optimization of masks and deep recurrent neural networks for monaural source separation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (2015) 2136–2147.

[17] S. R. Park, J. Lee, A fully convolutional neural network for speech enhancement, Annual Conference of the International Speech Communication Association (INTERSPEECH) (2017) 1993–1997.

[18] C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proc. of Computer Vision and Pattern Recognition (CVPR) (2015).

[19] E. M. Grais, D. Ward, M. D. Plumbley, Raw multi-channel audio source separation using multi- resolution convolutional auto-encoders, Proc. of European Signal Processing Conference (EUSIPCO) (2018).

[20] A. Pandey, D. L. Wang, A new framework for cnn-based speech enhancement in the time domain, IEEE/ACM Transactions on Audio, Speech, and Language Processing 27 (2019) 1179–1188.

[21] D. C. Yin, C. Luo, Z. W. Xiong, W. J. Zeng, Phasen: A phase-and-harmonics-aware speech enhancement network, AAAI Conference on Artificial Intelligence (2020).

[22] A. Pandey, D. L. Wang, Tcnn: Temporal convolutional neural network for real-time speech enhancement in the time domain, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019).

[23] L. Yi, N. Mesgarani, Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation, IEEE/ACM transactions on audio, speech, and language processing 27 (2019) 1256–1266.

[24] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, Z. W. J. Shlens, Rethinking the inception architecture for computer vision, Proc. of Computer Vision and Pattern Recognition (CVPR) (2016).

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, Proc. of Association for the Advancement of Artificial Intelligence (AAAI) conference (2016).

[27] X. Y. Zhang, X. Y. Zhou, M. X. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018).

[28] C. François, Xception: Deep learning with depthwise separable convolutions, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).

[29] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).

[30] A. L. Mass, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, International Conference on Machine Learning (ICML) (2013).

[31] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 2481–2495.

[32] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, Darpa timit acoustic phonetic continuous speech corpus cdrom (1993).

[33] IEEE recommended practice for speech quality measurements, IEEE Transaction on Audio Electroacoust (1969) 225–246.

[34] A. Varga, H. Steeneken, Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems, Speech Commun (1993) 247–251.

[35] G. N. Hu, D. L. Wang, A tandem algorithm for pitch estimation and voiced speech segregation., IEEE Transactions on Audio, Speech, and Language Processing 18 (2010) 2067–2079.

[36] C. Veaux, J. Yamagishi, S. King, The voice bank corpus: Design, collection and data analysis of a large regional accent speech database, International Conference Oriental COCOSDA held jointly with Conference on Asian Spoken Language Research and Evaluation (2013).

[37] D. P. Kinqma, J. L. Ba, Adam: A method for stochastic optimization, International Conference for Learning Representations (ICLR) (2015).

[38] C. H. Taal, R. C. Hendriks, R. Heusdens, J. Jensen, An algorithm for intelligibility prediction of time frequency weighted noisy speech, IEEE Transactions on Audio, Speech, and Language Processing 19 (2011) 2125–2136.

[39] Y. Hu, P. C. Loizou, Evaluation of objective quality measures for speech enhancement, IEEE Transactions on Audio, Speech and Language Processing 16 (2008) 229–238.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overftting, Journal of Machine Learning Research (2014) 1929–1958.

[41] A. Adeel, M. Gogate, A. Hussain, W. M. Whitmer, Lip-reading driven deep learning approach for speech enhancement, IEEE Transactions on Emerging Topics in Computational Intelligence (2018) 1–10.