Acoustic Modeling with Densely Connected Residual Network for Multichannel Speech Recognition

Jian Tang[1], *Yan Song*[1], *LiRong Dai*[1], *Ian McLoughlin*[2]

[1]National Engineering Laboratory for Speech and Language Information Processing University of Science and Technology of China, Hefei, Anhui, P.R.China
[2]School of Computing, University of Kent, Medway, UK

enjtang@mail.ustc.edu.cn, songy@ustc.edu.cn, lrdai@ustc.edu.cn, ivm@kent.ac.uk

Abstract

Motivated by recent advances in computer vision research, this paper proposes a novel acoustic model called Densely Connected Residual Network (DenseRNet) for multichannel speech recognition. This combines the strength of both DenseNet and ResNet. It adopts the basic "building blocks" of ResNet with different convolutional layers, receptive field sizes and growth rates as basic components that are densely connected to form so-called denseR blocks. By concatenating the feature maps of all preceding layers as inputs, DenseRNet can not only strengthen gradient back-propagation for the vanishing-gradient problem, but also exploit multi-resolution feature maps. Preliminary experimental results on CHiME-3 have shown that DenseRNet achieves a word error rate (WER) of 7.58% on beamforming-enhanced speech with six channel real test data by cross entropy criteria training while WER is 10.23% for the official baseline. Besides, additional experimental results are also presented to demonstrate that DenseRNet exhibits the robustness to beamforming-enhanced speech as well as near and far-field speech.

Index Terms: DenseNet, robust acoustic model, ResNet, speech recognition, CHiME-3

1. Introduction

With the advent of deep learning techniques, the performance of automatic speech recognition (ASR) has been significantly improved. However, it is still far from satisfactory in realistic noisy and far-field scenarios. To improve robustness of ASR, microphone arrays are commonly utilized, and multi-channel speech recognition is receiving more and more attention.

Existing multichannel speech recognition system mainly consist of a frontend to improve the robustness to severe signal impairments from noise or reverberation, and a backend for acoustic modeling. Recently, the frontend has become a hot research topic. Most frontend methods rely on a model-based masking of time frequency (TF) bins to estimate signal statistics for steering a corresponding beamformer [1, 2, 3, 4, 5, 6, 7, 8, 9]. Unlike the frontend, backend acoustic modeling has received less attention. In the CHiME-3 challenge, a simple 6 layer DNN network is employed as official baseline. However, in [8], a significant performance improvement was achieved by using a Wide Residual Network (WRN) model.

In this paper, we focus on the backend acoustic modeling, and attempt to find a suitable network architecture for robust ASR. In [10, 11, 12], the multi-resolution cepstral features are demonstrated to improve recognition performance over single resolution one either under the clean or white noise situation. In [13], a WRN model is proposed, which enjoys both the advantages of deeper networks with residual architecture to alleviate the vanishing-gradient problem, and wider network settings to increase the ability to learn different kinds of features. The multichannel speech recognition system with WRN-based backend acoustic model has shown its superiority for robust ASR [8]. More recently, densely connected convolutional networks (DenseNet), which can be seen as an extension of ResNet, achieve state-of-art performance on image recognition [14, 15, 16], Semantic Segmentation [17], and Handwritten Mathematical Expression Recognition [18]. The architecture is constructed from dense blocks and pooling operations, where each dense block is an iterative concatenation of previous feature maps.

Motivated by recent advances in computer vision research, we propose a Densely Connected Residual Network, termed DenseRNet, for backend acoustic modeling in Multichannel ASR. To combine the strength of both DenseNet and ResNet, DenseRNet adopts the "building blocks" of ResNet with different convolutional layers, receptive field sizes and growth rates as basic components to be densely connected to form the so-called denseR blocks. By concatenating the feature maps of all preceding layers as inputs, DenseRNet can not only strengthen gradient back-propagation for vanishing-gradient problem, but also exploit multi-resolution feature maps. Unlike [8], DenseRNet is implemented using a fully convolutional architecture, and no Bi-directional Long Short-Term Memory (BLSTM) layer is used to model temporal sequence and the whole network. To evaluate the effectiveness of DenseRNet, we conducted extensive experiments on the CHiME-3 challenge. The final DenseRNet system can achieve 7.58% in terms of word error rate (WER), which outperforms official baseline (10.23%) by a large margin.

2. Review of DenseNet and ResNet

In this section, we will briefly review ResNet and DenseNet architectures.

2.1. DenseNet: Densely connected convolutional network

DenseNet is composed of multiple dense blocks. Each block can be further divided into several densely connected convolution layers (see Fig. 1a). Each layer is defined as a basic component in a dense block, which contains composite functions of BN, rectifier non-linearity (ReLU) activation, convolution and dropout. Specifically, let $H_l()$ be a non-linear transformation of the l-th layer. This receives the feature maps of all preceding layers, denoted by $x_0, x_1, ..., x_{l-1}$, as input

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]) \quad (1)$$

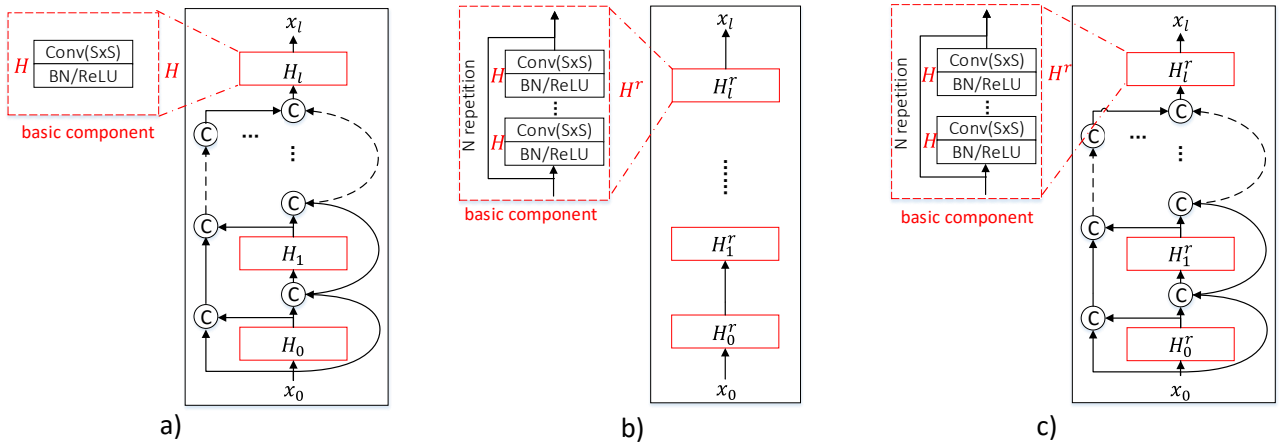

Figure 1: *Illustration of a) dense block, b) residual block and c) denseR block architectures.*

where [...] refers to the concatenation of all preceding layers.

According to [16], each H_l takes $k \times (l-1) + k_0$ input feature maps and produces k-dimensional output, where k_0 is the dimension of block input x_0, and k is referred to as the *growth rate*. To prevent the block growing too wide and to improve parameter efficiency, the growth rate k is practically limited to a small integer ($k = 12$). In addition, this k is fixed for each dense block.

Like standard CNN architectures such as VGGNet [19], the dense blocks can be connected into a network. A transition layer is further inserted between two adjacent dense blocks to change the feature map sizes. Such transition layers are composed of a 1×1 convolution followed by a 2×2 pooling operation.

2.2. ResNet: Residual Network

Just as in DenseNet, ResNet consists of several residual blocks. According to [20], each block is composed of multiple "building blocks". These "building blocks" are defined as basic components in ResNet, which contain several convolution layers with a "short connection", as shown in Fig. 1b. Specially, the output x_l of the l-th component can be expressed as

$$x_l = F_l(x_{l-1}) + x_{l-1} \triangleq H_l^r(x_{l-1}) \quad (2)$$

where x_{l-1} is the input feature map, F is a composite of 2 or 3 non-linear transformations H, and H_l^r is a residual transformation that sums the identity mapping of the input to the output. As shown in eq.(2), H_l^r allows for the reuse of features and permits the gradient to flow directly to earlier layers.

According to [20], He *et al.* followed the design rules of VGGNet [19], in which the width of each residual block (or the number of channels) started from 64 in the first residual block, and then increased by a factor of 2 for the remaining blocks.

In [8], Heymann *et al.* applied WRN as a backend acoustic model for multichannel speech recognition, and achieved a state-of-the-art performance. Compared with ResNet, DenseNet has several compelling advantages: in addition to the advantage of alleviating the vanishing-gradient problem, DenseNet can further strengthen feature propagation and exploit multi-resolution feature maps. All the above reported works motivate us to combine the strength of both DenseNet and ResNet for more powerful backend acoustic modeling. In the next section, the proposed DenseRNet will be detailed.

3. DenseRNet: Densely connected Residual Network

DenseRNet takes the similar hierarchical architecture of DenseNet and ResNet, which consists of multiple denseR blocks (see Fig. 1c). In this section, we first describe the structure of the denseR block, followed by the introduction of transition layers that will be inserted between denseR blocks. Then the DenseRNet-based backend acoustic model is introduced and finally, we will discuss the parameter settings for this model.

3.1. denseR block

The denseR block is composed of several basic components (*i.e.,* "building blocks" of ResNet). The basic components in a block are densely connected. Specifically, the l-th component receives the output of all preceding components, denoted as $x_0, x_1, ...x_{l-1}$, the output can be expressed as

$$x_l = H_l^r([x_0, ..., x_{l-1}]) \quad (3)$$

where the denseR block introduces residual transformation H_l^r same as the eq.(2). As shown in Fig.1, we can see that the proposed denseR block architecture combines both dense and residual block structures. In summary, the denseR block takes the dense connection structure of the basic components, which aim to combine the advantages of both DenseNet and ResNet. In our proposed DenseRNet system, two additional layers will be used to improving the computational efficiency.

3.2. Bottleneck layer

Just as in [16], an additional bottleneck layer, that is a 1×1 convolution, can be introduced before each basic component. The bottleneck layer can further reduce the number of input feature maps $[x_0, ..., x_{l-1}]$ and thus improve the computational efficiency. In practice, we set this number to be same as the growth rate k.

3.3. Transition layer

The transition layer is inserted into two adjacent denseR blocks to construct the DenseRNet. For speech recognition, the transition layer is designed as a composite of a 1×1 convolution layer and a pooling operation. The transition layer can further improve the model compactness by reducing the number of feature maps before feeding into the next denseR block.

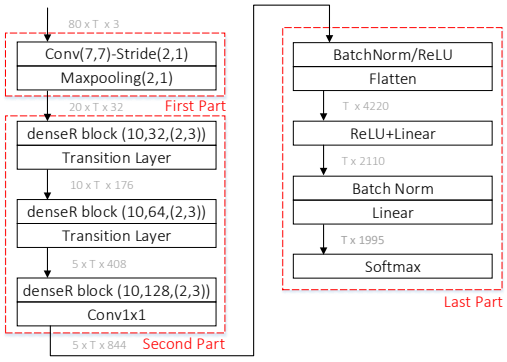

Figure 2: *The architecture of the DenseRNet-based backend acoustic model. The annotations in gray indicate the dimension of the tensors where B is the mini-batch size and T is the number of frames of the largest utterance within the batch.*

3.4. DenseRNet-based backend acoustic model

The architecture of the DenseRNet-based backend acoustic model is shown in Fig. 2.

The input to DenseRNet is a $D \times T \times C$ feature map, where D denotes the dimension of the features, T is the number of frames, and C are the channels. In all experiments, the 80 dimensional mean-normalized log-mel filterbank features are extracted from a given utterance. With T frames in the utterance, the input feature map dimension is thus $80 \times T$. In addition, the delta and delta-delta of the input are further exploited. The final input to DenseRNet is a $80 \times T \times 3$ feature map.

The first part of the DenseRNet is an initial convolution layer, which comprises 32 convolutions of size 7×7 with stride 2×1 followed by max-pooling with size of 2×1.

The second part of the network is composed of three denseR blocks. Following the terminology of DenseNet, each denseR block may be configured with four parameters $(L, k, (N, S))$, where L is the number of basic components, k the growth rate. (N, S) is related to the basic component: where N is the number of convolution layers and S is the kernel size. We will discuss the setting of those parameters in section 3.5. L is empirically set to 10 and the growth rate k of the three denseR blocks are set to 32, 64, 128 respectively. As mentioned, a transition layer is inserted between adjacent denseR blocks for improving the model compactness and a 1×1 convolutional layer is inserted before connecting to the final layers. The final layers consist of two fully-connected layers with batch normalization and ReLU activations. The final outputs are the posterior probabilities for the context-dependent states for each frame.

3.5. Discussion

In this section, we will focus on discussing how to set the parameters of denseR blocks, *i.e.,* the growth rate k and the number of convolution layers N.

Growth rate k. As described in section 2, k is fixed to a small integer in DenseNet, which aims to improve model compactness However, each basic component produces the k-dimensional output. This setting may not be optimal. Furthermore, k is fixed for each dense block. In ResNet, the width of each residual block increases by a factor of 2.

Number of convolution layers N. Parameter N is related to the basic components. In DenseNet, the basic component is a convolution layer, *i.e.,* N=1. While in ResNet, the basic component is a "building block" containing multiple convolution layers, *e.g.,* N=2 or 3. It is unclear what is optimal setting of N.

Table 1: *The performance comparison of DenseRNet with different configurations in terms of real word error rate(WER) in real test set (in%). The DenseRNet is configured with parameters $(L, k, (N, S))$, where L is the number of basic components, and k the growth rate. (N, S) is related to the basic component: where N is the number of convolution layers and S is kernel size.*

Model	L	k	(N, S)	#Para(MB)	WER
M1	22	(24 24 24)	(1,3)	7.81	11.7
M2	23	(16 32 64)	(1,3)	13.5	11.2
M3	10	(32 64 128)	(1,3)	11.9	11.6
M4	10	(32 64 128)	(2,3)	13.8	7.90
M5	10	(32 64 128)	(1,5)	16.1	8.39
M6	10	(32 64 128)	(3,3)	15.8	7.58

Based on the above discussion, we will study the following questions for the multichannel speech recognition task:
Q1. How to set the growth rate k?
Q2. Whether it is necessary to fix the growth rate k.
Q3. How to set the number of convolution layers: N?

4. Experimental evaluation

To evaluate the effectiveness of DenseRNet-based backend acoustic model, extensive experiments are conducted on CHiME-3 dataset [21]. For fair comparison, all the frontend processing is obtained by using six channel Generalized Eigenvalue (GEV) beamformer, and the backends are each trained on all six channels noisy utterances [9].

4.1. Implementation

Our implementation for CHiME-3 follows the structure as shown in Fig. 2. The input to DenseRNet is described in section 3.4. We adopt batch normalization before each convolution and activation, following [8] and initialize the weights as in [22]. Dropout [23] with a probability of 0.5 is added across the layer except for the input and output layers. To optimize the model, we use ADAM [24] with learning rate 8×10^{-4}, and frame-level cross entropy (CE) criterion is adopted as the objection function. The remaining experimental settings are similar to [8]. We use the Keras library in all experiments [25]

4.2. Evaluation on different parameter settings

In the following we evaluate the DenseRNet configured with different parameter settings of CHiME-3. The parameters to be evaluated includes: the growth rate k, the number of basic components L, and (N, S) is the parameters of basic component.

The experimental results are shown in Table 1. we also list the model size in terms of million bytes(MB) in the table. From Table. 1, we can see that when the growth rate k is fixed to 24, the WER is 11.7% as (M1). While in M2, the increasing growth rate k is used, the WER is reduced to 11.2%. This may answer question Q2. It is not necessary to fix growth rate k.

From M3, we reduce the L to 10, and find that the performance slightly degrades to 11.6%. This may due to the fact that the model with configuration of smaller L, (*i.e.,* L=23 vs.

Table 2: *Comparison of various Multichannel Systems based on CE criterion. The individual abbreviations mean: "Kaldi": baseline back-end, "DenseRNet": DenseRNet with same configurations as M4, "DenseNet": DenseNet with configurations* $\{L = 132, k = 24\}$ *[16], "ResNet": remove the input concatenate operation on the basis of "DenseRNet". Besides, WER(R/S) indicates word error rate of real and simulation test data, respectively.*

Back-end	Param(M)	WER(R/S)
Kaldi	30.2	10.2/9.62
WRBN	18.5	9.16/8.45
DenseNet	10.7	9.28/9.45
ResNet	20.2	8.23/8.77
DenseRNet	13.8	7.90/8.10

L=10), the receptive filed size is smaller. In this case, it indicates that the network with larger receptive field size may be better, and it's useless to give a large value to k (Q1).

We further evaluate the effect of N, S, the parameters of basic component, *i.e.,* the number of convolution layers and S is kernel size. From M4, when we increase the N to 2, and find that the WER significantly reduces from 11.6% to 7.9%. This may also attribute to the increasing of the receptive field size. To further identify it, we conduct the experiment with configuration (N, S)=(1, 5). The WER slightly degrades from 7.9% to 8.39% which means that two convolution layers (M5) can achieve 0.49% absolute reduce than the comparable receptive field (M4). This gives the facts that more convolution layer in the basic component is help for performance improvement (Q3). Since receptive field size is same in M4 and M5, it demonstrates that the increasing depth of network may have the similar effect as having larger receptive field size. The best performance is achieved with (N, S)=(3, 3), the WER of 7.58% has been achieved in M6.

4.3. Performance comparison with different models

We conduct the experiments to compare the proposed DenseRNet with other backend acoustic models, including 6-layer DNN of the official baseline, WRBN [8], ResNet, and DenseNet and DenseRNet. Except for the official baseline performance, we implement the model ourself using Keras. The experimental results may be different with the literature, mainly due to various frontend processing. For fair comparison, we configure the model to have the similar receptive field size, as shown in Table.2 From Table.2, we can see that WER of DenseRNet achieves the best performance, outperforming the official baseline with a large margin.

4.4. Experiments on robustness of DenseRNet

In this experiment, we evaluates the robustness of DenseRNet. Three kinds of speech,(*i.e.,* CH5, Enh, CH0) in CHiME-3 test corpus is used for evaluation [21]. The results are reported in Table 2. Firstly, we can find that performance of DenseRNet is superior to the 6-layer DNN model for all evaluations.. For different evaluation conditions, DenseRNet can achieve the similar performance on Enh (6.46%) and CH0 (7.90%). And DenseRNet is more robust than baseline to beamforming-enhanced speech as well as near, far-field one.

In Figure 3, we further analyze the robustness of DenseRNet by analyzing the figure of the mean feature maps, given

Table 3: *Compare 3 input feature (CH5,Enh,CH0) on DenseRNet, DNN (official baseline), BLSTM separately. The individual abbreviations mean: "CH5", "Enh", "CH0" represent the 5-th noisy far-field, the beamforming-enhanced and near-field utterances, respectively. "Real" and "Simu" indicate word error rate of real and simulation test data, respectively*

Model	input	Real	Simu
DenseRNet	CH5	14.1	9.73
	Enh	7.90	8.10
	CH0	6.46	4.29
DNN	CH5	32.2	20.9
	Enh	10.2	9.69
	CH0	8.09	5.10

one utterance in CHiME-3 real test set. From the first column, which shows three different input, *i.e.,* CH5, Enh, and CH0. They are clearly different. From the second column to last one in the figure, which corresponds to the mean of output feature maps from 2^{rd} denseR block, we can see that they tend to have the similar activations. This demonstrate a certain robustness to beamforming-enhanced speech as well as near, far-field one.

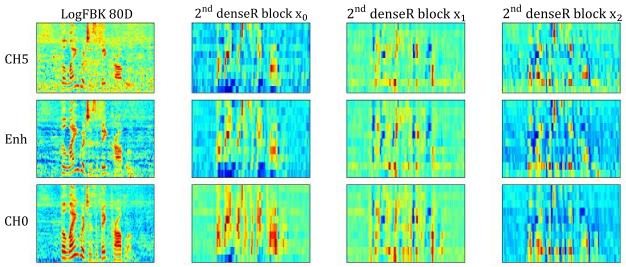

Figure 3: *The means of input feature map in the* 2^{rd} *denseR block in the real test set for three kinds of input, 'CH5', 'Enh' and 'CH0'.*

5. Conclusions

In this paper, a novel architecture, termed DenseRNet, is proposed. DenseRNet takes the similar hierarchical architecture as DenseNet and ResNet, consists of multiple denseR blocks To combine the strength of both DenseNet and ResNet, DenseRNet adopts "building block" of ResNet as the basic component. The basic components is densely connected in denseR block. DenseRNet can not only strengthen gradients back-propagation for vanishing-gradient problem, but also exploit multi-resolution feature maps. To evaluate the effectiveness of DenseRNet, we conducted experiments on CHiME-3 corpus with different convolutional layers, receptive field sizes and growth rates. We achieved a WER 7.58% using DenseRNet-based acoustic model on the beamforming-enhanced speech with the six channel real test data, outperforming the official baseline, (WER) 10.23%. Additional experimental results are also presented to demonstrate that DenseRNet exhibits the robustness to beamforming-enhanced speech as well as near, far-field one

6. ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China(Grant No:2017YFB1002202) and the Key Science

and Technology Project of Anhui Province (Grant No. 17030901005), and National Natural Science Foundation of China grant No.U1613211.

7. References

[1] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 436–443.

[2] X. Xiao, C. Xu, Z. Zhang, S. Zhao, S. Sun, S. Watanabe, L. Wang, L. Xie, D. L. Jones, E. S. Chng *et al.*, "A study of learning based beamforming methods for speech recognition," in *Proc. CHiME 2016 Workshop*, 2016, pp. 26–31.

[3] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.

[4] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and H. Li, "On time-frequency mask estimation for mvdr beamforming with application in robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3246–3250.

[5] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 196–200.

[6] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "Beamnet: End-to-end training of a beamformer-supported multi-channel asr system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5325–5329.

[7] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, "Optimizing neural-network supported acoustic beamforming by algorithmic differentiation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 171–175.

[8] L. D. Jahn Heymann and R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," in *CHiME 2016 workshop*, 2016.

[9] X. Zhang, Z.-Q. Wang, and D. Wang, "A speech enhancement algorithm by iterating single-and multi-microphone processing and its application to robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 276–280.

[10] S. Vaseghi, N. Harte, and B. Milner, "Multi-resolution phonetic/segmental features and models for hmm-based speech recognition," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1263–1266.

[11] P. McCourt, S. Vaseght, and N. Harte, "Multi-resolution cepstral features for phoneme recognition across speech sub-bands," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1. IEEE, 1998, pp. 557–560.

[12] P. McMahon, N. Harte, S. Vaseghi, and P. McCourt, "Discriminative spectral-temporal multiresolution features for speech recognition," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 581–584.

[13] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[14] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense convolutional networks for efficient prediction," *arXiv preprint arXiv:1703.09844*, 2017.

[15] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger, "Memory-efficient implementation of densenets," *arXiv preprint arXiv:1707.06990*, 2017.

[16] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, no. 2, 2017, p. 3.

[17] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1175–1183.

[18] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," *arXiv preprint arXiv:1801.03530*, 2018.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chimespeech separation and recognition challenge: Dataset, task and baselines," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 504–511.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] F. Chollet *et al.*, "Keras," 2015.