MULTI-QUARTZNET: MULTI-RESOLUTION CONVOLUTION FOR SPEECH RECOGNITION WITH MULTI-LAYER FEATURE FUSION

Jian Luo, Jianzong Wang*, Ning Cheng, Guilin Jiang, Jing Xiao

Ping An Technology (Shenzhen) Co., Ltd.

ABSTRACT

In this paper, we propose an end-to-end speech recognition network based on Nvidia's previous QuartzNet [1] model. We try to promote the model performance, and design three components: (1) Multi-Resolution Convolution Module, replaces the original 1D time-channel separable convolution with multi-stream convolutions. Each stream has a unique dilated stride on convolutional operations. (2) Channel-Wise Attention Module, calculates the attention weight of each convolutional stream by spatial channel-wise pooling. (3) Multi-Layer Feature Fusion Module, reweights each convolutional block by global multi-layer feature maps. Our experiments demonstrate that Multi-QuartzNet model achieves CER 6.77% on AISHELL-1 data set, which outperforms original QuartzNet and is close to state-of-art result.

Index Terms— speech recognition, multi-resolution, multi-layer feature fusion

1. INTRODUCTION

In the last few years, end-to-end neural networks have achieved remarkable results on automatic speech recognition tasks. Among these models, convolutional neural network architectures have attracted much attention. They are often used in the models combined with recurrent layers, such as CLDNN [2], DeepSpeech [3, 4]. In these works, the CNN layers are designed to reduce the time and spectral variation of the input features, and their outputs are passed to RNN layers for temporal modeling. However, these models are often encountered with speed problems, because RNN layers cannot be trained or inferenced parallelly. Given the evidence that convolutional networks are also suitable on long-range dependency tasks, a lot of fully convolutional approachs were proposed. Wav2Letter [5] proposed an ASR system, which only used a standard 1D convolutional neural network trained by CTC loss. And then, fully convolutional networks [6] were presented. In their works, not only acoustic models but also language models as well as learnable front end were all based on convolutional pipelines.

Inspired by Wav2Letter, Nvidia's team proposed a computationally efficient end-to-end convolutional network named

Jasper [7], which used a stack of 1D-convolution layers, with ReLU and batch normalization. They also found that simple ReLU and batch normalization outperform other activation and normalization. In Jasper, they introduced dense residual connections for training converge and better performance. They then updated their model to QuartzNet [1], replacing traditional 1D-convolution layers by 1D time-channel separable convolutional layers. Time-depth separable convolutions [8, 9, 10] are designed to reduce the number of parameters in traditional convolutions while keeping the receptive field large. The original QuartzNet model has $\mathbb{K} \times \mathbb{C} + \mathbb{C}^2$ parameters, where $\mathbb{K}$ is the kernel size and $\mathbb{C}$ is the channel dimension. By comparison, our proposed multi-resolution separable convolution has $\mathbb{K} \times \mathbb{C} \times \mathbb{S} + \mathbb{C}^2$ parameters, where $\mathbb{S}$ are the stream numbers of multi-resolution. The parameters of our model increase slightly, because stream numbers S are usually quite small (e.g., $\mathbb{S} = 2, 4$), but still dramatically better than traditional convolutions' $\mathbb{K} \times \mathbb{C}^2$ parameters. As shown in experimental section, multi-resolution convolutions get better results in our settings.

Some previous works have tried multi-stream or multi-resolution methods in their models. [11] proposed a multi-stride self-attention mechanism with various strides over neighboring speech frames. [12, 13] used a multi-stream idea combining CNN-based and RNN-based encoders. [14] proposed a multi-scale octave convolution layer to learn robust speech representations efficiently. Inspired from the work [15], we propose multi-resolution separable convolution with channel-wise attention. In their works, they used multi-stream for more effective expressions of speech features to the subsequent self-attention module. However, their model just concatenated all of the stream outputs to one vector, and did not consider the degree of importance of each convolutional stream. In our works, we design a channel-wise attention module, to calculate the attention weight of each convolutional stream automatically. The attention weight module is composed of squeeze and excitation [16, 17] function by pooling the feature maps across each channel dimension.

Another point of our works is that, we try to make connections cross the bottom and up layers. The bottom layers often contain local and detail information, and up layers by contrast usually contain global and general information. We think it is useful to combine them together for better speech expressions.

Jasper [7] designed Dense Residual (DR) [18], making connections between the blocks. Feature Pyramid Networks (FPN) [19, 20] is another popular feature fusion mechanism used in the area of image object detection. FPN adoptes a backbone model, and builds feature pyramid. The feature pyramid sequentially combines two adjacent layers in feature hierarchy with top-down and lateral connections. Recently, Cross-layer Feature Pyramid Network (CFPN) [21] was proposed to improve the progressive fusion between layer-level information of the pyramid network. We adopt the Cross-layer Feature Aggregation (CFA) module of CFPN, aggregated multi-scale features from different convolutional blocks to a global context. The context is used to calculate the attention weight of each block. We design a multi-layer feature fusion module, which combines the re-weighted blocks with local residual connection.

We demonstrate that the proposed Multi-QuartzNet model performs best, when using all of the three components: (1) Multi-Resolution Convolution Module, (2) Channel-Wise Attention Module, and (3) Multi-Layer Feature Fusion Module.

2. MODEL ARCHITECTURE

Our proposed Multi-QuartzNet is designed based on QuartzNet architecture, which is a fully convolutional network trained on CTC loss. The model architecture of Multi-QuartzNet is shown in Figure 1, which has the following structure: 1) The raw acoustic features are first fed into a 1D convolution layer C_1(Conv-BN-ReLU). 2) After that, it is followed by a sequence of multi-resolution blocks, and each block $B_i(i = 1, ..., \mathbb{I})$ is repeated $\mathbb{R}$ times. The block consists of $\mathbb{M}$ times multi-resolution convolution modules. Each convolution module is composed of a multi-resolution convolutional layer (DepthwiseConv-PointwiseConv-BN-ReLU) with stride set $[1, ..., \mathbb{S}]$ and a channel-wise attention module to calculate the attention weight of each stream. 3) Morever, the model introduces a multi-layer feature fusion module, to reweight each convolutional block output. 4) Finally, three additional convolutional layers (C_2, C_3, C_4) are designed to map the features to vocabulary size with CTC loss training.

2.1. Multi-Resolution Convolution

Compared with the original QuartzNet architecture, we replace the 1D time-channel separable convolution with our proposed multi-resolution convolutions. Each convolution has a unique dilated stride but still time-channel separable. More specifically, we define an input feature map $x \in \mathbb{T} \times \mathbb{C}$ of each convolution, where $\mathbb{T}$ is the total frames number and $\mathbb{C}$ is the channel demension. For each stream, the feature x is firstly opearated on a 1D-Depthwise convolutional layer with same kernel size $\mathbb{K}$ but with different stride $\mathbb{S}$. Because the depthwise convolutional layer fixes the channel demension and operates at spatial demension, each stream produces an
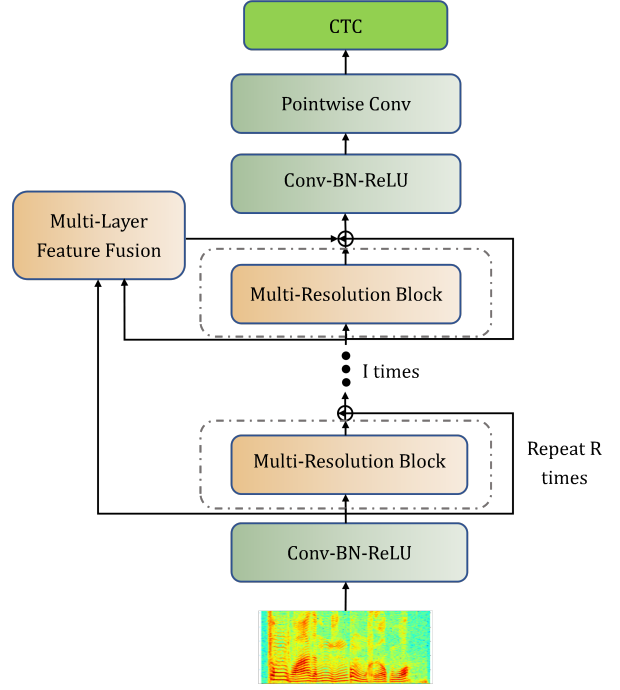

Fig. 1. Multi-QuartzNet Model Architecture

output $x_\mathbb{S}$ with the same shape $\mathbb{T} \times \mathbb{C}$. Then the $x_\mathbb{S}$ is reviewed by the shape of $1 \times \mathbb{TC}$, and is applied by a 1×1 pointwise convolutions across all the channels. And then, batch norm layer and ReLU activation are attached after convolutional layer of each stream. After multi-resolution convolutions, the model sums all the stream outputs by channel dimension,

$$x_{1,\mathbb{S}} = x_1 \oplus x_2... \oplus x_\mathbb{S} \tag{1}$$

where $\oplus$ denotes the add operation. We maintain the same channel dimension to all the streams, so $x_{1,\mathbb{S}}$ has the same shape with input x, which is $x_{1,\mathbb{S}} \in \mathbb{T} \times \mathbb{C}$. The whole multi-resolution convolution module is shown as Figure 2.

The multi-resolution convolution module enables the network to capture multi-scale information. With small dilated stride (e.g., $\mathbb{S} = 1$), the stream will more focus on the local information like spectral of phoneme. With large dilated stride (e.g., $\mathbb{S} = 4$), the stream will pay more attention to the global information like background noise. By contrast, traditional convolution just looks around the local information which is restricted by kernel size $\mathbb{K}$.

2.2. Channel-Wise Attention Module

As Figure 2 depicts, we design a channel-wise attention module in multi-resolution convolution. The attention module uses squeeze-and-excitation function to calculate the attention
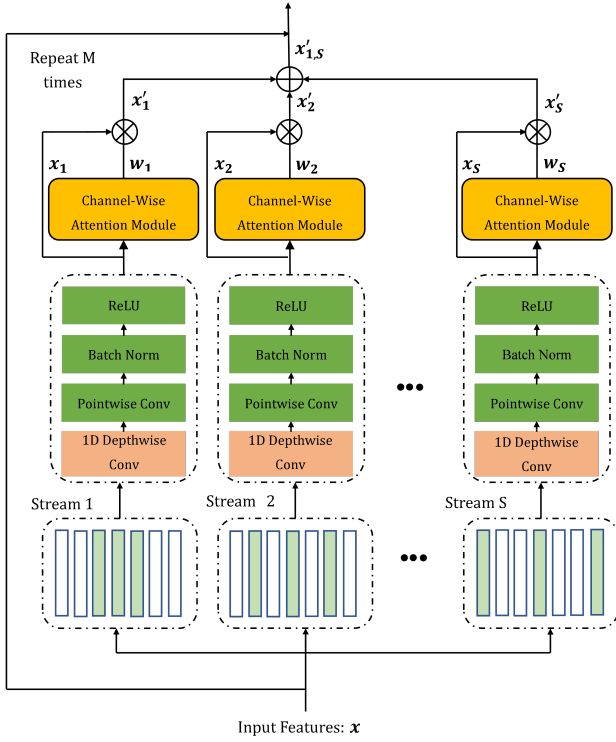
Fig. 2. Multi-Resolution Convolution Module

Fig. 3. Multi-Layer Feature Fusion

weight of each channel. The convolutional outputs $x_{\mathbb{S}}$ are then reweighted by multiplying the attention weight across each channel dimension. Inspired by SENet [16], the channel-wise attention module firstly squeezes spatial information into a channel-wise vector by average and maximum pooling. Formally, the average pooling vector $x_{\mathbb{S}}^{avg} \in \mathbb{C}$ and max pooling vector $x_{\mathbb{S}}^{max} \in \mathbb{C}$, are calculated by:

$$x_{\mathbb{S}}^{avg} = F^{avg}(x_{\mathbb{S}}) = \frac{1}{\mathbb{T}} \sum_{t=1}^{\mathbb{T}} x_{\mathbb{S}}(t) \tag{2}$$

$$x_{\mathbb{S}}^{max} = F^{max}(x_{\mathbb{S}}) = \max_{t=1}^{\mathbb{T}} x_{\mathbb{S}}(t) \tag{3}$$

Here squeezed vector $x_{\mathbb{S}}^{avg}$ and $x_{\mathbb{S}}^{max}$ are calculated separately by each channel. Next, the excitation function takes the squeezed vector into a gating mechanism network, which is composed of two fully connected layers with a sigmoid activation:

$$\begin{aligned} w_{\mathbb{S}} = F^{ex}(x_{\mathbb{S}}^{avg}, x_{\mathbb{S}}^{max}) &= \sigma(g(x_{\mathbb{S}}^{avg}, x_{\mathbb{S}}^{max})) \\ &= \sigma(w_2\delta(w_1 x_{\mathbb{S}}^{avg}) \oplus w_2\delta(w_1 x_{\mathbb{S}}^{max})) \end{aligned} \tag{4}$$

where δ refers to the ReLU activation, and σ denotes the sigmoid function. $w_1 \in \mathbb{C}/\mathbb{D} \times \mathbb{C}$ and $w_2 \in \mathbb{C} \times \mathbb{C}/\mathbb{D}$, are the
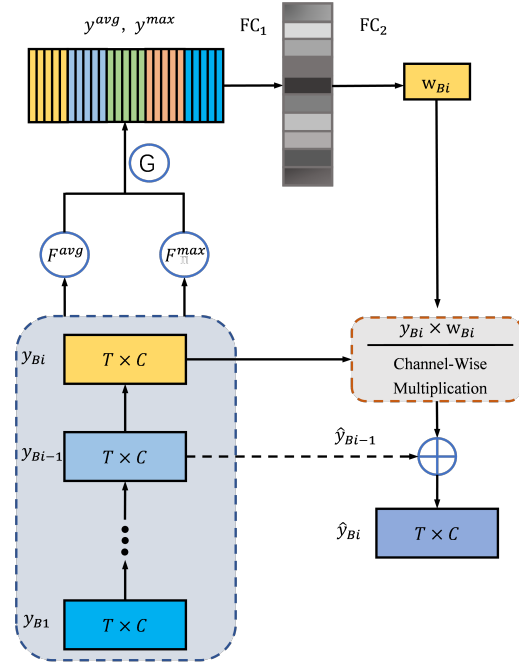
gating network weights. $\mathbb{D}$ is the dimension reduction ratio. So output $x_{\mathbb{S}}'$ of each convolution are reweighted by channel-wise product of $x_{\mathbb{S}}$ and $w_{\mathbb{S}}$. After that, all of the reweighted streams are summed together like Eq. 1:

$$x_{\mathbb{S}}' = F^{scale}(x_{\mathbb{S}}, w_{\mathbb{S}}) = x_{\mathbb{S}} * w_{\mathbb{S}} \tag{5}$$

$$x_{1,\mathbb{S}}' = x_1' \oplus x_2'... \oplus x_{\mathbb{S}}' \tag{6}$$

The channel-wise attention module gives attention values to each stream, making the network excite and suppress the stream referring to whole spatial information.

2.3. Multi-Layer Feature Fusion

In deep convolutional networks, residual connections are necessary for training converge. DenseNet [18] and DenseRNet [22] are the components of Jasper. In DenseNet, each block output was residually connected to the input of all following blocks. However, DenseNet just adds all of the residual connections together, regardless of the importance degree of each connection. In this paper, we introduce a multi-layer feature fusion module, to reweight each convolutional block by global multi-layer feature maps.

The multi-layer feature fusion module is shown as Figure 3, which is inspired by the Cross-layer Feature Aggregation Module(CFA) of Cross-layer FPN [21]. Formally, each convolutional block output is noted as $y_{Bi} \in \mathbb{T} \times \mathbb{C}$. Here we assume the channel dimension of each block is the same for convenience. The feature fusion module firstly applies a global pooling at each block, and then concatenates all the pooling vectors to a global context y^{avg} and y^{max}:

$$y^{avg} = G_{i=1}^{\mathbb{I}}(F^{avg}(y_{Bi})) \qquad (7)$$

$$y^{max} = G_{i=1}^{\mathbb{I}}(F^{max}(y_{Bi})) \qquad (8)$$

where G is the concatenated operation across dimensions of all blocks, and $\mathbb{I}$ is the blocks number. Then we design another excitation function to compute multi-layer feature fusion weights by leveraging the context y^{avg} and y^{max}:

$$\begin{aligned} w_{Bi} = F^{ex}(y^{avg}, y^{max}) &= \sigma(g(y^{avg}, y^{max})) \\ &= \sigma(\hat{w}_2\delta(\hat{w}_1 y^{avg}) \oplus \hat{w}_2\delta(\hat{w}_1 y^{max})) \end{aligned} \qquad (9)$$

where $\hat{w}_1 \in \hat{\mathbb{C}}/\hat{\mathbb{D}} \times \hat{\mathbb{C}}$ and $\hat{w}_2 \in \hat{\mathbb{C}} \times \hat{\mathbb{C}}/\hat{\mathbb{D}}$, are the multi-layer gating weights. $\hat{\mathbb{C}} = \mathbb{C} \times \mathbb{I}$, and $\hat{\mathbb{D}}$ is the multi-layer reduction ratio. Then each block ouput y_{Bi} is reweighted by w_{Bi}, and add the local residual connection $\hat{y}_{Bi-1}$ to get the final block output $\hat{y}_{Bi}$:

$$\hat{y}_{Bi} = F^{scale}(y_{Bi}, w_{Bi}) \oplus \hat{y}_{Bi-1} \qquad (10)$$

3. EXPERIMENTS

3.1. Data Set

Our experimental works are implemented by comparing the performance of our models with original QuartzNet on Mandarin speech recognition tasks. We use AISHELL-1 [23] data set, a publicly available Mandarin speech corpus. AISHELL-1 contains 178 hours recording audios (16 kHz, 16 bit). The corpus is divided into training, development and testing sets. Character Error Rates (CER) are evaluated on testing set.

3.2. Experimental Setup

We train both small Multi-QuartzNet5x3 and large Multi-QuartzNet15x5 models on AISHELL-1 training set. The parameters of our models are listed on table 1 and table 2. The input features are 64-dimension mfcc coefficients with 20ms frame length and 10ms frame shift. Data augmentation ($\pm 10\%$ speed perturbation, SpecAugment) are used in the training procedure. The stream numbers of multi-resolution convolution are set to $\mathbb{S} = 4$ for Multi-QuartzNet5x3, and $\mathbb{S} = 2$ for Multi-QuartzNet15x5 because of GPU memory limit. Reduction ratio $\mathbb{D}$ and $\hat{\mathbb{D}}$ maintain 16 for all of the experiments. All of the models are trained for 400 epochs with batch size 32 for each GPU. Novograd [24] optimizer is used with learning rate of 0.01 and weight decay of 0.0001. We also apply

Table 1. Small Multi-QuartzNet5x3 Model Configuration

Block	$\mathbb{R}$	$\mathbb{M}$	$\mathbb{K}$	$\mathbb{C}$	Stride Set
C_1	1	1	33	256	[1]
B_1	1	3	63	512	[1, 2, 3, 4]
B_2	1	3	63	512	[1, 2, 3, 4]
B_3	1	3	75	512	[1, 2, 3, 4]
B_4	1	3	75	512	[1, 2, 3, 4]
B_5	1	3	75	512	[1, 2, 3, 4]
C_2	1	1	87	512	[1]
C_3	1	1	1	1024	[1]
C_4	1	1	1	$\|labels\|$	[1]

Table 2. Large Multi-QuartzNet15x5 Model Configuration

Block	$\mathbb{R}$	$\mathbb{M}$	$\mathbb{K}$	$\mathbb{C}$	Stride Set
C_1	1	1	33	256	[1]
B_1	3	5	33	256	[1, 3]
B_2	3	5	39	256	[1, 3]
B_3	3	5	51	512	[1, 3]
B_4	3	5	63	512	[1, 3]
B_5	3	5	75	512	[1, 3]
C_2	1	1	87	512	[1]
C_3	1	1	1	1024	[1]
C_4	1	1	1	$\|labels\|$	[1]

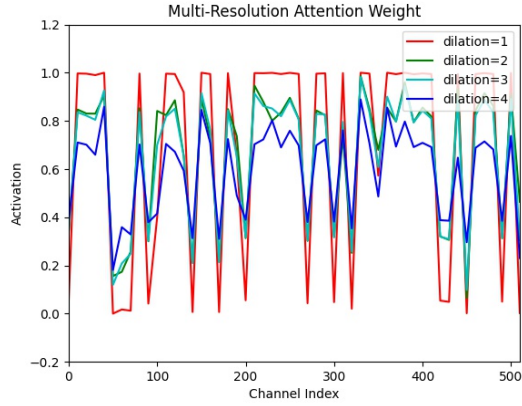the CosineAnnealing [25] training policy with 8000 warmup steps.

During inference, a standard beam search algorithm similar to [3] is used with character-level 4-gram langauge model. The langauge model is trained on the transcripts of training set. Formally, $p(l|x)$ is noted as the output of Multi-QuartzNet model, $p_{lm}(l)$ is the output of language model, and $wc(l)$ is the word counts of prediction character sequence l. We attempt to find a sequence l that maximizes the combined probability:

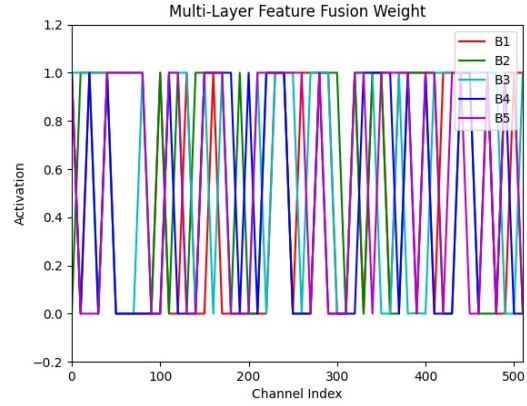$$Q(l) = log(p(l|x)) + \alpha log(p_{lm}(l)) + \beta wc(l) \qquad (11)$$

We maximize this objective $Q(l)$ by beam search algorithm, and beam size $= 200$. We also fine-tune the hyperparameter $\alpha = 1.8$ and $\beta = 3.5$ on the development set.

3.3. Results

We study the effects of each component on our proposed Multi-QuartzNet network. As illustrated in table 3, for both 5x3 and 15x5 models, the performances are greatly improved with multi-resolution convolution than original QuartzNet model. With channel-wise attention module, the CER results can get a further reduction. Finally, multi-layer feature fusion enables the model to interact cross multi-layer, resulting in the best accuracy in our experiments. Our proposed Multi-Quartznet network is composed of three above components. In addition, we can see that large Multi-Quartznet15x5 model is better than 5x3 model although the Aishell-1 data set is relatively small.

(a) Multi-Resolution Attention Weight

(b) Multi-Layer Feature Fusion Weight

Fig. 4. Channel-Wise Weight Analysis

Table 3. The Effect of Each Component on Multi-QuartzNet, CER(%)

Model	Size	LM	Test
QuartzNet	5×3	4-gram	8.55
	15×5	4-gram	**7.18**
Multi-Resolution Convolution	5×3	4-gram	7.79
	15×5	4-gram	**6.90**
+Channel-Wise Attention	5×3	4-gram	7.62
	15×5	4-gram	**6.84**
+Multi-Layer Feature Fusion	5×3	4-gram	7.28
	15×5	4-gram	**6.77**

Table 4. Comparsion of Multi-QuartzNet with Other Models

Model	CER(%)
LAS [26]	10.56
RNN-T [27]	11.82
SA-T [27]	9.30
Sync-Transformer [28]	8.91
Combiner [29]	8.87
LFMMI [30]	7.62
LDS-REG [31]	10.56
ESPnet Transformer [32]	6.70
MTH-MoChA [33]	7.68
Multi-QuartzNet(ours)	**6.77**

We also compare our model with other published models in table 4. Our best Multi-Quartznet15x5 model achieves CER 6.77% on AISHELL-1 testing set, which outperforms most of the listed models and is close to ESPnet Transformer [32].

3.4. Analysis

To verify the attention and feature fusion module, we show channel-wise activations of last convolutional block B_5 on various dilations in Fig 4a and fusion weights on different blocks in Fig 4b. We observe that for small dilation $\mathbb{S} = 1$, the activations tend to be closed 0 or saturated 1. While for big dilation $\mathbb{S} = 4$, the activations are more stable, and closer to the middle region 0.5. It demonstrates that large dilation has large reception of field, and captures more stable long-context information. While for multi-layer fusion weights, they show switch characteristics, turning on or off some channels of the outputs.

4. CONCLUSION

In this paper, we propose a fully-convolutional network named Multi-QuartzNet for speech recognition. The network introduces multi-resolution convolution into the original QuartzNet. Morever, we design a channel-wise attention module and a multi-layer feature fusion module. The attention module calculates the attentions of each stream, and the fusion module computes the weights of each block. The experiments show that Multi-QuartzNet outperforms the original QuartzNet, and is close to state-of-the-art performance on AISHELL-1 data set. Future works include exploring model performances on other languages like english and experiments on larger corpus.

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP*, 2020.

[2] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *ICASSP*, 2015.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," in *Computer Science*, 2014.

[4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, and Zhenyao Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Computer Science*, 2015.

[5] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," in *Computer Science*, 2016.

[6] Neil Zeghidour, Qiantong Xu, Vitaliy Liptchinsky, Nicolas Usunier, Gabriel Synnaeve, and Ronan Collobert, "Fully convolutional speech recognition," in *arXiv:1812.06864*, 2018.

[7] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan Cohen, Huyen Nguyen, and Ravi Gadde, "Jasper: An end-to-end convolutional neural acoustic model," in *Interspeech*, 2019.

[8] Awni Hannun, Ann Lee, Qiantong Xu, and Ronan Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," in *Interspeech*, 2019.

[9] Sudhakar Kumawat and Shanmuganathan Raman, "Depthwise-stft based separable convolutional neural networks," in *ICASSP*, 2020.

[10] Elahe Rahimian, Soheil Zabihi, Seyed Farokh Atashzar, A. Asif, and Arash Mohammadi, "Xceptiontime: A novel deep architecture based on depthwise separable convolutions for hand gesture classification," in *arXiv:1911.03803*, 2019.

[11] Kyu J.Han, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou, "Multi-stride self-attention for speech recognition," in *Interspeech*, 2019.

[12] Ruizhi Li, Xiaofei Wang, Sri Harish Mallidi, Shinji Watanabe, Takaaki Hori, and Hynek Hermansky, "Multi-stream end-to-end speech recognition," in *arXiv:1906.08041*, 2019.

[13] Ruizhi Li, Xiaofei Wang, Sri Harish Mallidi, Takaaki Hori, Shinji Watanabe, and Hynek Hermansky, "Multi-encoder multi-resolution framework for end-to-end speech recognition," in *arXiv:1811.04897*, 2018.

[14] Joanna Rownicka, Peter Bell, and Steve Renals, "Multi-scale octave convolutions for robust speech recognition," in *arXiv:1910.14443*, 2019.

[15] Kyu J. Han, Ramon Prieto, Kaixing Wu, and Tao Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," in *arXiv:1910.00716*, 2019.

[16] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu, "Squeeze-and-excitation networks," in *arXiv:1709.01507*, 2017.

[17] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *arXiv:1807.06521*, 2018.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *arXiv:1608.06993*, 2016.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *arXiv:1612.03144*, 2016.

[20] Chaoxu Guo, Bin Fan, Qian Zhang, Shiming Xiang, and Chunhong Pan, "Augfpn: Improving multi-scale feature learning for object detection," in *arXiv:1912.05384*, 2019.

[21] Zun Li, Congyan Lang, Junhao Liew, Qibin Hou, Yidong Li, and Jiashi Feng, "Cross-layer feature pyramid network for salient object detection," in *arXiv:2002.10864*, 2020.

[22] Jian Tang, Yan Song, Lirong Dai, and Ian Mcloughlin, "Acoustic modeling with densely connected residual network for multichannel speech recognition," in *Interspeech*, 2018.

[23] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *arXiv:1709.05522*, 2017.

[24] Boris Ginsburg, Patrice Castonguay, Oleksii Hrinchuk, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Huyen Nguyen, Yang Zhang, and Jonathan M. Cohen, "Stochastic gradient methods with layer-wise adaptive moments for training of deep networks," in *arXiv:1905.11286*, 2019.

[25] Ilya Loshchilov and Frank Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *arXiv:1608.03983*, 2016.

[26] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie, "Component fusion: Learning replaceable language model component for end-to-end speech recognition system," in *ICASSP*, 2019.

[27] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, and Zhengqi Wen, "Self-attention transducers for end-to-end speech recognition," in *Interspeech*, 2019.

[28] Zhengkun Tian, Jiangyan Yi, Ye Bai, Jianhua Tao, Shuai Zhang, and Zhengqi Wen, "Synchronous transformers for end-to-end speech recognition," in *ICASSP*, 2020.

[29] Jiawei Wu, Chenyan Xiong, Tobias Schnabel, Yizhe Zhang, William Yang Wang, and Paul Bennett, "Combiner: Inductively learning tree structured attention in transformer," in *ICLR*, 2020.

[30] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *ICSDA*, 2017.

[31] Sining Sun, Pengcheng Guo, Lei Xie, and Mei-Yuh Hwang, "Adversarial regularization for attention based end-to-end robust speech recognition," in *TASLP*, 2019.

[32] Shigeki Karita, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Yalta, and Ryuichi Yamamoto, "A comparative study on transformer vs rnn in speech applications," in *ASRU*, 2019.

[33] Baiji Liu, Songjun Cao, Sining Sun, Weibin Zhang, and Long Ma, "Multi-head monotonic chunkwise attention for online speech recognition," in *arXiv:2005.00205*, 2020.