ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414639

MULTISTREAM CNN FOR ROBUST ACOUSTIC MODELING

Kyu J. Han[1], *Jing Pan*[1], *Venkata Krishna Naveen Tadala*[2], *Tao Ma*[1] *and Dan Povey*[3]

[1]ASAPP, Mountain View, CA, USA
[2]Sensory, Portland, OR, USA
[3]Xiaomi Inc., Beijing, China

ABSTRACT

This paper proposes *multistream CNN*, a novel neural network architecture for robust acoustic modeling in speech recognition tasks. The proposed architecture processes input speech with diverse temporal resolutions by applying different dilation rates to convolutional neural networks across multiple streams to achieve the robustness. The dilation rates are selected from the multiples of a sub-sampling rate of 3 frames. Each stream stacks TDNN-F layers (a variant of 1D CNN), and output embedding vectors from the streams are concatenated then projected to the final layer. We validate the effectiveness of the proposed multistream CNN architecture by showing consistent improvements against Kaldi's best TDNN-F model across various data sets. Multistream CNN improves the WER of the test-other set in the LibriSpeech corpus by 12% (relative). On custom data from ASAPP's production ASR system for a contact center, it records a relative WER improvement of 11% for customer channel audio to prove its robustness to data in the wild. In terms of real-time factor, multistream CNN outperforms the baseline TDNN-F by 15%, which also suggests its practicality on production systems. When combined with self-attentive SRU LM rescoring, multistream CNN contributes for ASAPP to achieve the best WER of 1.75% on test-clean in LibriSpeech.

Index Terms: Multistream CNN, robust acoustic modeling, speech recognition

1. INTRODUCTION

Automatic speech recognition (ASR) with processing speech inputs in multiple streams, namely *multistream ASR*, has long been researched mostly for robust speech recognition tasks in noisy environments since the earlier works such as [1, 2, 3]. The multistream ASR framework was proposed based on the analysis of human perception and decoding of speech, where acoustic signals enter into the cochlea and are broken into multiple frequency bands such that the information in each band can be processed in parallel in the human brain [4]. This approach worked reasonably well in the form of multi-band ASR where band-limited noises dominate signal corruption [5, 6]. Later, further development was made in regards with multistream ASR in the areas of spectrum modulation and multi-resolution based feature processing [7, 8, 9] and stream fusion or combination [10, 11, 12, 13].

With the advent of deep learning, multistream ASR research shifts its focus on deep neural network (DNN) architectures where multiple streams of encoders process embedding vectors in parallel. Although some forms of artificial neural networks like multilayer perceptron (MLP) [14] had already been utilized in the literature for multistream ASR [3, 13], they were shallow and their usage was limited to fusing posterior outputs from a classifier in each stream. The recent DNN architectures for multistream ASR instead perform more unified functions, not only processing information in parallel but combining the information streams to classify all at once. In [15, 16] a multistream ASR architecture was simplified into one neural network where a binary switch was randomly applied to each feature stream when concatenating the multistream features as the neural network input. In decoding, a tree search algorithm was utilized to find the best stream combination. In [17], a stream attention mechanism inspired by the hierarchical attention network [18] was proposed to multi-encoder neural networks that can accommodate diverse viewpoints when processing embedding vectors. This multi-encoder architecture was successful in data sets recorded with multiple microphones [19, 20]. As multi-head self-attention [21] became widely employed, multistream self-attention architectures were also investigated in [22, 23].

This paper presents *multistream CNN* (as illustrated in Figure 1) as a novel neural network architecture for robust speech recognition. The proposed architecture processes input speech with diverse temporal resolutions by having stream-specific dilation rates to convolutional neural networks (CNNs) across multiple streams to achieve the robustness. In each stream we stack TDNN-F[1], a variant of 1D-CNN. The dilation rate for the TDNN-F layers in each stream is chosen from multiples of the default sub-sampling

Codes are available at https://github.com/asappresearch/multistream-cnn.

[1] TDNN-F stands for factorized time-delay neural network [24]. The convolution matrix in TDNN-F is decomposed into two factors with the orthonormal constraint, followed by a skip connection, batch normalization and a dropout layer.

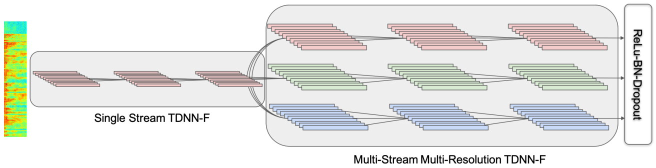

Fig. 1: Schematic diagram of the proposed multistream CNN architecture.

rate (3 frames) for model training and decoding. The choice of multiples of 3 for the dilation rates can offer a seamless integration with the training and decoding process. Output embedding vectors from the streams are concatenated then projected to the final layer.

We structure this paper as follows. In Section 2, we detail training and evaluation data, and share experimental setups for ablation discussions in Section 3, where we explain our proposal of multistream CNN and analyze the impact of a few design choices in the proposed architecture on the LibriSpeech data. In Section 4, we discuss the performances of single- and multistream CNNs on custom data from ASAPP's production ASR system for a contact center in terms of both WER and RTF. In section 5, we conclude this work with summaries and comments on future directions.

2. DATA AND EXPERIMENTAL SETUPS

2.1. Data

The LibriSpeech corpus [25] is a collection of approximately 1,000hr read speech (16kHz) from audio books. Each dev/test category (clean and other) contains around 5hrs of audio. This corpus also provides n-gram LMs trained on 800M token texts.

The Switchboard-1 Release 2 (LDC97S62) and Fisher English Training Part 1 and 2 (LDC2004S13, LDC2004T19, LDC2005S13, LDC2005T19) corpora total 2,000hrs of 8kHz telephony speech. We use the HUB5 eval2000 data (LDC2002S09, LDC2002T43) for evaluation.

We collect roughly 500hrs of 8kHz audio from our production ASR system. An eval set is 10hr of audio collection with a balanced distribution between agent and customer channel recordings.

2.2. Experimental Setups

For LibriSpeech, neural networks for acoustic modeling are trained on the 960hr training set with the LF-MMI objective [26]. The learning rates are decayed from 10^{-3} to 10^{-5} over the span of 6 epochs. The minibatch size is 64. We use the n-gram LMs provided by the LibriSpeech corpus for the 1st pass decoding and 2nd pass rescoring.

Regarding model training with SWBD/Fisher, we leverage the default Kaldi recipe[2] [27]. We train models on the 2,000hr training data. For neural network AMs, we exponentially decay the learning rates from 10^{-3} to 10^{-4} during 6 epochs. The minibatch size is 128. The default n-gram LMs produced by the recipe are used for decoding.

We fine-tune the SWBD/Fisher model with the ASAPP custom data for further evaluation on data in the wild. We adjust the learning rate decay schedule, starting from 10^{-5} to 10^{-7} for 6 epochs with the minibatch size of 128. The PocoLM toolkit[3] is used to train a 4-gram LM for the 1st-pass decoding in the evaluation.

3. MULTISTREAM CNN

As illustrated in Figure 1, the proposed multistream CNN architecture branches multiple streams after processing given input speech frames with 5 CNN layers in a single stream where CNNs could be TDNN-F or 2D-CNN (in case of applying SpecAugment [28]). After being branched out, a stack of 17 TDNN-F layers in each stream process the output of the single-streamed CNNs with a unique dilation rate. Consider an embedding vector $\mathbf{x}_i$ comes out of the single-streamed CNN layers at a given time step of i. An embedding vector $\mathbf{y}_i^m$ from a stream m having gone through the stack of TDNN-F layers with a dilation rate r_m can be written as below:

$$\mathbf{y}_i^m = \textit{Stacked-TDNN-F}_m\left(\mathbf{x}_i; [-r_m, r_m]\right), \tag{1}$$

where $[-r_m, r_m]$ means a 3×1 kernel with the dilation rate r_m for each TDNN-F layer. It is crucial to choose r_m from the multiples of a sub-sampling rate used for model training and decoding. (In our case, we choose the multiples of 3 frames.) Output embedding vectors from all the streams are concatenated and followed by ReLu, batch normalization and a dropout layer;

$$\mathbf{z}_i = \textit{Dropout}\left(\textit{BN}\left(\textit{ReLu}\left(\textit{Concat}\left(\mathbf{y}_i^1, \mathbf{y}_i^2, \ldots, \mathbf{y}_i^M\right)\right)\right)\right), \tag{2}$$

which is projected to the output layer via a couple of fully connected layers.

In next subsections, we analyze the effect of our design choices in the proposed multistream CNN architecture using the LibriSpeech dev and test sets. Unless specified, the complexity of all the models compared in the analysis is around 20M parameters for a fair comparison. The baseline model is a 17-layer (single-stream) TDNN-F in the recipe[4] for LibriSpeech of the Kaldi toolkit.

[2] https://github.com/kaldi-asr/kaldi/tree/master/egs/fisher_swbd/s5

[3] https://github.com/danpovey/pocolm

[4] https://github.com/kaldi-asr/kaldi/egs/librispeech/s5/local/chain/run_tdnn.sh

Table 1: LibriSpeech WERs (%) by multistream CNNs with various combinations of dilation rates across streams. d: embedding dimension for TDNN-F.

System	d	dev		test	
		clean	other	clean	other
Baseline	1,536	3.26	8.86	3.68	8.92
Multistream CNN					
1-2	768	3.36	8.86	3.80	8.91
1-2-3	512	3.33	8.81	3.84	8.93
1-2-3-4-5	307	3.36	8.62	3.67	8.76
1-2- · · · -6-7	219	3.27	**8.39**	3.65	8.85
1-2- · · · -8-9	170	3.27	8.45	**3.60**	**8.63**
1-3-6	512	3.27	8.58	3.67	8.71
3-6-9	512	3.24	8.30	3.59	8.70
6-9-12	512	**3.17**	**8.25**	**3.54**	**8.41**
1-3-6-9-12	307	3.29	8.26	3.57	8.78
3-6-9-12-15	307	3.22	8.30	3.58	8.52

3.1. Multiple Streams w/ Dilation Rates

Table 1 compares multistream CNN models against the baseline as we increase the number of streams with various dilation rates. For example, `1-2-3-4-5` indicates that the dilation rates of 1, 2, 3, 4, and 5 are used to TDNN-F layers over the total 5 streams respectively in a multistream CNN model. We adjust the dimension of embedding vectors for the TDNN-Fs to keep the model complexity around 20M parameters for a fair comparison. From the upper half of the table, it is evident that the proposed multistream CNN architecture improves the WERs of the 'other' data sets more noticeably as we increase the number of streams up to 9 by the increment of 1. We don't report model performances with more streams since we observed no improvement after 9 streams. This could have resulted from combined reasons, such as too small embedding dimension for TDNN-F over too many streams. In the lower half of the table, we apply the multiples of the sub-sampling rate (i.e., 3 frames). The results prove careful selection of dilation rates would further improve WER even with smaller numbers of streams. The choice of the multiples of 3 for TDNN-F layers seems to be better streamlined with the training and decoding process where input speech frames are sub-sampled every 3 frames.

We find the best setup from the `6-9-12` configuration, which, juxtaposed with the baseline, shows a relative WER improvements of 6.9% and 5.7% on dev-other and test-other, respectively.

Table 2: LibriSpeech WERs (%) by multistream CNNs in larger size. N: model complexity in # of parameters. d: embedding dimension for TDNN-F.

System	N	d	dev		test	
			clean	other	clean	other
Baseline	20.7M	1,536	3.26	8.86	3.68	8.92
Multistream CNN						
6-9-12	20.6M	512	3.17	8.25	3.54	8.41
6-9-12	73.2M	1,536	**3.07**	8.10	**3.40**	**8.32**
6-9-12	93.9M	1,536	3.09	**7.98**	3.52	**8.32**

3.2. Larger Networks

Table 2 contrasts the WERs of the multistream CNN models with the same `6-9-12` configuration, but with different model complexity. The 73M parameter model has 3 times larger embedding dimension for TDNN-F (1,536 versus 512), while the 94M parameter model has 7 more TDNN-F layers in each stream. As observed in the table, the larger-sized multistream CNN models reached lower WERs, but the improvement from the 20M parameter model seems marginal considering much longer training times. In real-world applications, especially for cases where online inference is critical, the 20M parameter model must be a reasonable choice.

3.3. Toward State-of-the-Art

In this section, we optimize the multistream CNN model with the `6-9-12` configuration (20M parameter) with SpecAugment and neural network based LMs toward competitive state-of-the-art results in LibriSpeech.

Since its introduction in [28], the SpecAugment data augmentation method of masking random time-frequency bands from input spectrograms has been wildly adopted by both hybrid and end-to-end ASR systems. SpecAugment is known to prevent neural network models from being overfit thus enable them to become more robust to unseen testing data. To apply this method on top of the proposed multistream CNN architecture, we replace the first 5 layers of TDNN-F of the model (corresponding to the Single Stream TDNN-F part in Figure 1) with 5 layers of 2D-CNN to better accommodate log-mel spectrograms. We use 3 × 3 kernels for the 2D-CNN layers with a filter size of 256 except for the first layer with the filter size of 128. Every other layer we apply frequency band sub-sampling with the rate of 2.

We employ multiple stages of LM rescoring in order to obtain the minimum WERs on the test sets in LibriSpeech. The LMs are trained on normalized texts where typos are corrected as well as spelling consistencies between British and American English are addressed. The initial decoding is based on the decoding graph constructed from the multi-

Table 3: State-of-the-art performances on Librispeech using multistream CNN and neural network based LMs including self-attentive SRU LM.

Setup	dev		test	
	clean	other	clean	other
Multistream CNN	2.62	6.78	2.80	7.06
+TDNN-LSTM LM	2.14	5.82	2.34	6.04
+Self-Attentive SRU	1.55	4.22	**1.75**	4.46

stream CNN AM and a 3-gram LM, resulting in the initial hypotheses in a lattice format. Lattice rescoring is done with a larger sized 4-gram LM, followed by a second-pass lattice rescoring with a TDNN-LSTM language model [29]. In the final rescoring stage, we use an interpolated self-attentive SRU LM [30]. We linearly interpolate two self-attentive SRU models, one of which is trained on word pieces using byte-pair encoding (BPE) and the other is trained at a word level. After the interpolation, we re-rank the N-best hypotheses from the lattices rescored by the TDNN-LSTM LM in the previous stage. In our experiments, we empirically keep N at 100.

In Table 3, we tabulate the performances of the three LM rescoring stages for multistream CNN where the 4-gram LM rescores the first pass decoding results (first line) and the other two neural network based LMs further rescore the n-gram LM rescored results. The 1.75% WER on test-clean is, to the best of the authors' knowledge, the lowest reported in the literature, without extra data (e.g., Libri-Light) taken into consideration for either AM or LM training.

4. MULTISTREAM CNN IN THE WILD

In this section, we manifest the feasibility of the proposed multistream CNN architecture in real-world scenarios. We use our custom training data (500hrs) collected from ASAPP's contact center ASR system to fine-tune the seed models (baseline TDNN-F and multistream CNN) trained on the SWBD/Fisher corpora mentioned in Section 2.1. The seed model performances on the HUB5 eval2000 data consisting of the SWBD and CH (i.e., CallHome) portions are presented in Table 4. A noteworthy observation in the table is that the proposed model architecture continues to excel the baseline model in more challenging data. This is further highlighted in Table 5 where the two fine-tuned models (baseline and multistream CNN) are evaluated on the ASAPP custom eval set of 10hrs also described in Section 2.1. The relative WER improvement (WERR) of 11.4% on the customer channel recordings[5] declares the robustness of the proposed multistream CNN model architecture in the wild. In addition, the relative real-time factor (RTF) improvement of 15.1% against the baseline TDNN-F model shows the practicality of the proposed model architecture in real-world applications, especially where online inference is necessary.

Table 4: HUB5 eval2000 WERs (%) by telephony seed models. SWBD: Switchboard, CH: CallHome in HUB5 eval2000.

	Baseline		Multistream CNN	
	SWBD	CH	SWBD	CH
WER (%)	8.7	16.2	9.0	15.6

Table 5: Relative performance improvements by multistream CNN on ASAPP's custom data for conversational speech over telephony channels, against the baseline TDNN-F model. The absolute performances are not disclosed. RTF: real time factor.

	WERR (%)		Relative
	Agent	Customer	RTF Imp.
Multistrem CNN	8.8	11.4	15.1

5. CONCLUSIONS

In this paper, we proposed a novel neural network architecture, namely multistream CNN, for robust speech recognition. The reasoning behind the proposal was that diversity in temporal resolution across multiple streams would enhance the overall robustness in acoustic modeling. We empirically showed that it would further improve the benefit of having such diversity in temporal resolution to choose dilation rates for TDNN-F layers across multiple streams form the multiples of 3 frames (i.e., sub-sampling rate). We tested multistream CNN models on various data sets including ASAPP's custom data collected from a contact center ASR system to demonstrate the robustness and practicality of the proposed model architecture.

Multistream CNN seems promising to be utilized in a number of ASR applications and frameworks. We plan to continue to improve this multistream model architecture to further enhance our production ASR systems.

[5] Compared to agent channel audio, customer channel audio are by far challenging for ASR systems due to noisier acoustic environments, non-native/accented speech, multiple talkers, etc.

6. REFERENCES

[1] Herve Bourlard, Stephane Dupont, Hynek Hermansky, and Nathaniel Morgan, "Towards subband-based speech recognition," in *EUSIPCO*, 1996, pp. 1–4.

[2] Herve Bourlard and Stephane Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *ICSLP*, 1996, pp. 426–429.

[3] Hynek Hermansky, Sangita Tibrewala, and Misha Pavel, “Towards ASR on partially corrupted speech,” in *ICSLP*, 1996, pp. 462–465.

[4] John Allen, “How do humans process and recognize speech?,” *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 2, no. 4, pp. 567–577, 1994.

[5] Herve Bourlard and Stephane Dupont, “Subband-based speech recognition,” in *ICASSP*, 1997, pp. 1251–1254.

[6] Sangita Tibrewala and Hynek Hermansky, “Sub-band based recognition of noisy speech,” in *ICASSP*, 1997, pp. 1255–1258.

[7] Hynek Hermansky and Sanjita Sharma, “Temporal patterns (TRAPS) in ASR noisy speech,” in *ICASSP*, 1999, pp. 289–292.

[8] Hynek Hermansky and Sanjita Sharma, “Multi-resolution RASTA filtering for TANDEM-based ASR,” in *ICSLP*, 2005, pp. 361–364.

[9] Zoltan Tuske, Ralf Schluter, and Hermann Ney, “Acoustic modeling of speech waveform based on multi-resolution neural network signal processing,” in *ICASSP*, 2018, pp. 4859–4863.

[10] Shigeki Okawa, Enrico Bocchieri, and Alexandros Potamianos, “Multi-band speech recognition in noisy environments,” in *ICASSP*, 1998, pp. 641–644.

[11] Andrew Morris, Astrid Hagen, Hervé Glotin, and Hervé Bourlard, “Multi-stream adaptive evidence combination for noise robust ASR,” *Speech Communication*, vol. 34, no. 1-2, pp. 25–40, 2001.

[12] Nima Mesgarani, Samuel Thomas, and Hynek Hermansky, “Adaptive stream fusion in multistream recognition of speech,” in *Interspeech*, 2011, pp. 2329–2332.

[13] Sri Harish Mallidi, Tetsuji Ogawa, Karel Vesely, Phani S. Nidadavolu, and Hynek Hermansky, “Autoencoder based multi-stream combination for noise robust speech recognition,” in *Interspeech*, 2015, pp. 3551–3555.

[14] Herve Bourlard and Nelson Morgan, “Connectionist speech recognition: A hybrid approach,” *The Springer International Series in Engineering and Computer Science*, vol. 247, no. 1, 1994.

[15] Sri Harish Mallidi and Hynek Hermansky, “Novel neural network based fusion for multistream ASR,” in *ICASPP*, 2016, pp. 5680–5684.

[16] Sri Harish Mallidi and Hynek Hermansky, “A framework for practical multistream ASR,” in *Interspeech*, 2016, pp. 3474–3478.

[17] 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,” 2018, [Online]. Available: https://arxiv.org/abs/1811.04897.

[18] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, “Hierarchical attention networks for document classification,” in *NAACL/NLT*, 2016, pp. 1480–1489.

[19] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, “The third ‘CHIME’ speech separation and recognition challenge: Analysis and outcome,” *Comp. Speech Lang.*, vol. 46, pp. 605–626, 2017.

[20] Maurizio Omologo Mirco Ravanelli, Piergiorgio Svaizer, “Realistic multi-microphone data simulation for distant speech recognition,” in *Interspeech*, 2016, pp. 2786–2790.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *NeurIPs*, 2017, pp. 5998–6008.

[22] Kyu J. Han, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou, “Multi-stride self-attention for speech recognition,” in *Interspeech*, 2019, pp. 2788–2792.

[23] Kyu J. Han, Ramon Prieto, and Tao Ma, “State-of-the-art speech recognition using multi-stream self-attention with dilated 1D convolution,” in *ASRU*, 2019, pp. 54–61.

[24] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur, “Semi-orthogonal low-rank matrix factorization for deep neural networks,” in *Interspeech*, 2018, pp. 3743–3747.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “LibrSspeech: An ASR corpus based on public domain audio books,” in *ICASSP*, 2015, pp. 5206–5210.

[26] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, “Purely sequence-trained neural networks for ASR based on lattice-free MMI,” in *Interspeech*, 2016, pp. 2751–2755.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, “The Kaldi speech recognition toolkit,” in *ASRU*, 2011.

[28] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiua, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019, pp. 2613–2617.

[29] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, “Recurrent neural network language model adaptation for conversational speech recognition,” in *Interspeech*, 2018, pp. 3373–3377.

[30] Jing Pan, Joshua Shapiro, Jeremy Wohlwend, Kyu J. Han, Tao Lei, and Tao Ma, “ASAPP-ASR: Multistream CNN and self-attentive SRU for SOTA speech recognition,” in *Interspeech*, 2020, pp. 16–20.