Convolutional Time Delay Neural Network for Khmer Automatic Speech Recognition

Nalin Srun, Sotheara Leang, Ye Kyaw, Sethserey Sam

Convolutional Time Delay Neural Network for Khmer Automatic Speech Recognition

Author version

Nalin Srun[1], **Sotheara Leang**[1,2], **Ye Kyaw Thu**[2], **Sethserey Sam**[2]

[1] Institute of Digital Research and Innovation, CADT, Phnom Penh, Cambodia

[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

Abstract

Convolutional Neural Networks have been proven to successfully capture spatial aspects of the speech signal and eliminate spectral variations across speakers for Automatic Speech Recognition. In this study, we investigate the Convolutional Neural Network with Time Delay Neural Network for an acoustic model to deal with large vocabulary continuous speech recognition for Khmer. Our idea is to use Convolutional Neural Networks to extract local features of the speech signal, whereas Time Delay Neural Networks capture long temporal correlations between acoustic events. The experimental results show that the suggested network outperforms the Time Delay Neural Network and achieves an average relative improvement of 14% across test sets.

***K*eywords** Khmer ASR, Time Delay Neural Network, Convolutional Neural Network, Low-resource Language

1 Introduction

Google Assistant, Siri, Alexa, and Cortana are transforming the way users interact with their devices, homes, and automobiles. These voice assistants simplify our lives, and many individuals become dependent on them. A key aspect of that smart merchandise is Automatic Speech Recognition (ASR). It permits you to talk into your computer or tool, and it translates your voice into textual information automatically. At present, a variety of approaches including conventional models [1, 2], hybrid models [3, 4] and end-to-end neural models [5] are carried out in almost ASR systems to achieve the state of the art speech recognition accuracy. The rapid advancement of ASR is heavily dependent on the massive amounts of audio and annotated transcripts. Recognizing the majority of languages having a rich resource gives a very mature performance [6] when compared to low-resource languages, which have so far achieved limited accuracy [7].

In a recent study [8], Khmer ASR was studied with Deep Neural Network (DNN) versus Gaussian Mixture Model (GMM), and the result showed that the DNN outperformed by approximately 3.65% and 1.10% in the open test and close test, respectively. However, when working in a language with limited resources, DNN results in overfitting [9]. Furthermore, the DNN can only acquire a limited amount of contextual information from the speech, limiting its ability to deal with long-range correlations in the speech signal [10, 11]. In recent years, various novel networks have been studied, and it has been discovered that Time Delay Neural Network (TDNN) is one of the potential neural networks for modeling the long temporal information of the speech [12, 13]. The TDNN is a stacked architecture that employs a modular and progressive design to build a bigger network from the sub-components. In the lower layers, it learns the narrow context of the input features, whereas in the top layers, it learns the larger context. Furthermore, it is quicker to train TDNN and requires much less training data compared to Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) [14]. Speech can be regarded as an image because the spectral properties of speech consist of two dimensions, such as time and frequency. As a result, several researches on Convolutional Neural Network (CNN) has been conducted for speech recognition [15]. The three critical components of CNN, including local receptive field, weight sharing, and sub-sampling, are known to be capable of minimizing the variance of translation, scaling, and distortion of the speech signal [16]. Therefore, CNN has additional advantages for speech processing when dealing with diverse pronunciations for different speakers. In this research, we propose to investigate the benefits of combining CNN and TDNN in dealing with large vocabulary continuous speech recognition for Khmer. Our goal is to utilize the CNN as an extra function to the TDNN to learn local features of the speech signal.

This research is structured as follows: the proposed method is given in Section 2. Section 3 introduces the experiments including the datasets, language model, lexicon, and baseline model. The results and discussions are presented in Section 4. Finally, Section 5 presents the conclusion and future work.

2 Proposed Method

In this investigation, the CNN-TDNN of Kaldi's Mini-LibriSpeech recipe [17] was adopted. The network consisted of 6 convolutional neural network blocks, followed by 9 delay neural network blocks, and a fully connected block (Fig. 1). It took the 40-dimensional Mel-frequency Cepstral Coefficients (MFCCs) and 100-dimensional i-vector as the input features to estimate 2,976 states of the triphone Hidden Markov Model (HMM). The network contains 4.7 million parameters and can gather ±30 contextual inputs from a particular acoustic frame.

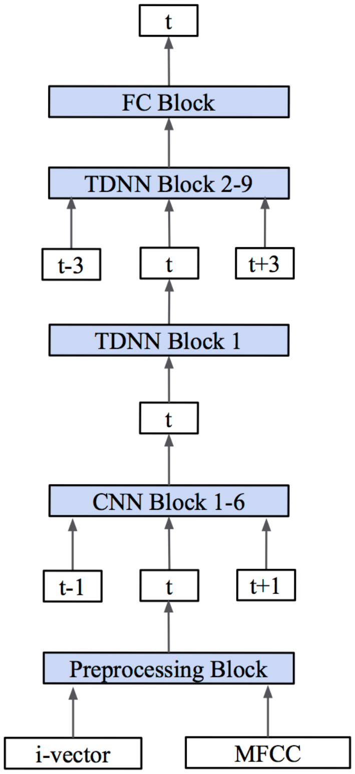

Figure 1: Adopted architecture for CNN-TDNN

2.1 Feature Extraction

The speech signals were first split into short-time frames of a length of 25ms with an overlap of 10ms. A Hamming window, a pre-emphasis factor of 0.97, a cepstral liftering coefficient of 22 and cepstral mean normalization (CMN) were used. Finally, 40-dimensional high resolution MFCCs were computed and 100-dimensional i-vectors were extracted on top of PCA-reduced spliced-MFCC features for speaker adaptation.

2.2 Preprocessing Block

The input features were preprocessed before being fed to the network as shown in Fig. 2. First, mel-filterbanks were computed from 40-dimensional MFCCs through inverse discrete cosine function (MFCCs are more compressible, thus it is preferred to dump them to disk rather than mel-filterbanks). Second, batch normalization and SpecAugment were applied on the mel-filterbanks. In Kaldi, the SpecAugment is implemented as a neural layer that does time and frequency masking on-the-fly during the training. Third, 100-dimensional i-vectors were projected to 200-dimensional vectors through a linear transformation, followed by a batch normalization. Finally, 40-dimensional mel-filterbanks and 200-dimensional i-vectors were combined to produce 40x6 input features.

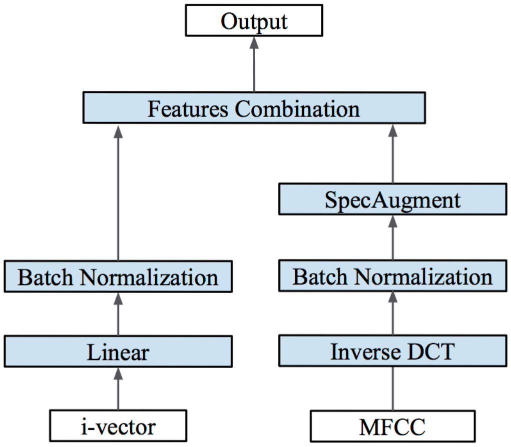

Figure 2: Preprocessing block

2.3 Convolutional Neural Network Block

Our network employed 6 CNN blocks, each with a convolutional layer followed by a rectified linear unit (ReLU) activation function and a batch normalization, to learn local spatial information and to eliminate variations in inter-speaker spectra (Fig. 3). Each convolutional layer took three consecutive outputs from the preceding layer of time steps (t-1, t, t+1), allowing the CNN blocks to gather information from 12 neighboring frames. During each convolution operation, 3x3 filters with a strip size of 1 and a zero-padding size of 1 were employed. The first three convolutional layers use 48 filters; the next two layers use 64 filters; and the last layer uses 128 filters. Layers 3, 5, and 6 use a subsampling size of 2 to lower the height dimension of the input feature. The final outputs (1x5x128) were transformed into 640-dimensional vectors and utilized as inputs to TDNN sequence blocks.

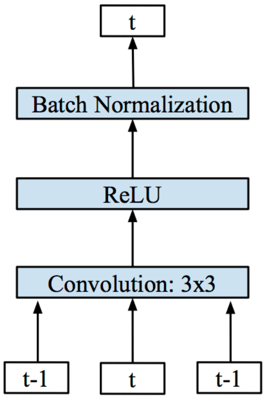

Figure 3: Convolutional neural network block

2.4 Time Delay Neural Network Block

Our network was made up of 9 TDNN blocks, and thus, it can extract information from the long temporal context of the inputs. Each block begins with a TDNN layer, followed by a ReLU activation function and a batch normalization function (Fig. 4). The Factored Time Delay Neural Network (TDNNF) was used in the network. It uses Single Value Decomposition (SVD) to improve matrix multiplication, resulting in fewer parameters and much less computation. The first TDNNF layer has a dimension of 768 while SVD has a dimension of 192. Time stride was not applied in this layer. The subsequent TDNNF layers consist of 768 dimensions and 96 dimensions for SVD. Each layer took 6 consecutive outputs from ±3 contexts and concatenated them with the current output from the previous layer. Therefore, the final output from the TDNN blocks can capture information from the previous and the prior 24 inputs. RestNet connections were applied in TDNN layers to capture information from the input from the previous layer. Except for the first layer, the output from each layer was summed with the output from the previous layer with a scaling factor of 0.66.

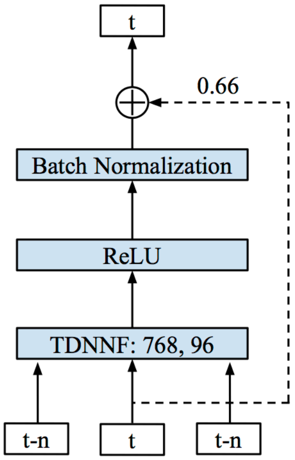

Figure 4: Time delay neural network block. The number n denotes the size of adjacent inputs to be combined with the current input t.

2.5 Training Recipe

A context-independent (CI) model was first trained on 13-dimensional MFCCs with their first and second-order derivatives. MFCCs were computed using 23 Mel-filterbanks on overlapping frames of 25ms with 10ms shift. A Hamming window, a pre-emphasis factor of 0.97, a cepstral liftering coefficient of 22 and CMN were used. The first context-dependent (CD) was trained over 40 iterations using 2.5k states and 25k Gaussian distributions. Second CD training using 5k states and 50k Gaussians was performed by concatenating ±4 contexts of 13-dimensional MFCCs and projecting into a 40-dimensional subspace using linear discriminant analysis (LDA), and the features were decorrelated using maximum likelihood linear transform (MLLT). Speaker-adaptive training (SAT) was performed with the same configuration of 5k states and 50k Gaussians to normalize the features using feature space maximum likelihood linear regression (fMLLR). Note that the configurations of CD models are based on [8].

Three data augmentations were used to expand the amount of the training data and provide greater signal variability in speech. First, speed perturbation with factors of 0.9, 1.0, and 1.1 was carried out, producing 3 times the training data. Second, volume perturbation was used to make the acoustic model resilient to volume fluctuations by using random factors sampled from a uniform distribution of [1/8, 2]. Finally, SpecAugment [18] was conducted on the fly in the neural layer of the network by masking the frequency channels of at most 50% of the spectrum and masking the time steps of the frames of at most 20%.

The 40-dimensional high-resolution MFCCs were extracted from the augmented training data, and half of the data was used to train the i-vector extractor [19], which was used to compute the 100-dimensional i-vectors from speech signals. Finally, CNN-TDNN was trained on the aligned training data from the SAT model across 20 epochs with starting and final learning rates of 0.002 and 0.0002, respectively, using lattice-free maximum mutual information (LF-MMI) [20].

3 Experiments

3.1 Datasets

In this study, the training set includes portions of the Basic Travel Expression Corpus (BTEC) [21] consisting of 47 speakers and 48,369 utterances. It is a speech corpus that was produced from the previous work [8] and designed to cover the expressions in the traveling domain. More data was collected from the internet (online websites and YouTube) and combined into the datasets to examine the performance of Khmer ASR in several domains. Finally, the new speech corpus, which is mostly in natural speech and background noise, and contains 119 speakers, 73,660 utterances, and around 153 hours, was used during the learning phase. Furthermore, we proposed to investigate the performance of CNN-TDNN in five domains, including News, Law, Health, Daily life and General. The specification of the training and testing sets are given in Table 1.

3.2 Language Model

The language model is an essential component of the ASR system. By estimating the likelihood of word sequences, it is able to model how words are put together to form sentences. It can predict which words will follow particular words and with what probability. A 3-grams language model was built using SRILM [22]. A Witten-Bell Smoothing technique was used to prevent assigning zero probability to the unknown words and a pruning factor of 1e-8 was applied to reduce the size of the language model. Using an in-house toolkit [23], the text of each dataset was segmented into morphemes and subjected to punctuation removal and numbers to text conversion. Table 2 shows specification of the training and testing datasets as well as their perplexities.

3.3 Lexicon

The lexicon is the pronunciation dictionary that connects sub-word units (phonemes) to words. The acoustic model uses it to learn mapping from sequence of phonemes to words. The corpus from [24] was used to build a grapheme-to-phoneme (G2P) model using Phonetisaurus G2P [25]. The dataset, which included 57 phonemes and 34k words, was created using the International Phonetic Alphabet (IPA) standard. Finally, a new lexicon was constructed using the vocabulary of the language model, which consists of around 100k words.

3.4 Baseline Model

In this study, we proposed using TDNN as the baseline model. The suggested network design is given in Fig. 5. It is made up of 15 TDNN blocks and is capable of capturing 30 adjacent contexts from a particular input feature. The network is similar to CNN-TDNN except for the first six TDNN blocks. Our idea is to investigate the performance of the acoustic model when CNNs are utilized in the network while maintaining the same configuration. Each TDNN accepts 3 successive input features (t-1, t, t+1), and thus, the TDNN blocks can obtain information from 6 neighboring input features.

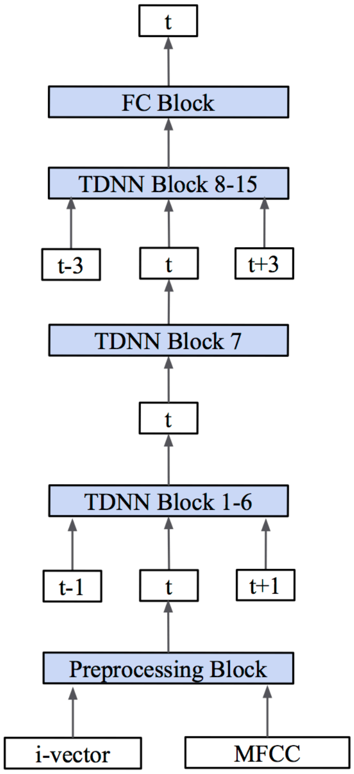

Figure 5: Proposed architecture for baseline model

3.5 Evaluation Method

Word Error Rate (WER) was used as the metric to evaluate the accuracy of TDNN and CNN-TDNN. It is a common measure for evaluating the performance of an ASR system. The following is the formula:

$$WER = \frac{S + D + I}{S + D + C} \tag{1}$$

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and C is the number of correct words. WER is between zero and one. The lower the WER, the better performance of the acoustic model.

4 Results and Discussions

Table 3 compares the performance of CNN-TDNN with TDNN across the five test sets. TDNN performs

Table 1: Dataset Preparation

Dataset	Domain	Hour	Utterance	Speaker
Train Set	Multi-domain	∼ 153	73,660	57 males, 62 females
Test Set	News	∼ 4	970	2 males, 3 females
	Law	∼ 3	1,035	2 males, 3 females
	Health	∼ 4	1,399	2 males, 3 females
	Daily Life	∼ 5	1,840	2 males, 3 females
	General	∼ 4	1,474	2 males, 3 females

Table 2: Dataset Preparation for Language Model

Dataset	Domain	Vocabulary	Sentence	Average Word	Perplexity
Train Set	Multi-domain	$\sim 100K$	$\sim 3M$	53	036.42
Test Set	News	$\sim 3K$	$\sim 3K$	28	090.38
	Law	$\sim 2K$	$\sim 3K$	20	158.29
	Health	$\sim 3K$	$\sim 3K$	20	158.29
	Daily Life	$\sim 3K$	$\sim 4K$	24	238.94
	General	$\sim 3K$	$\sim 3K$	22	169.29

best in the News, with a WER of 11.53%, followed by 17.57% in the Health and Daily Life, and a WER of roughly 19% in the Law and General. CNN-TDNN, on the other hand, outperforms by a significant margin of 14.4% on average throughout the test sets. It achieves the best WER of 11.53% in the News, followed by 14.83% in the Daily Life and around 15.7% in the Health and General. According to the experimental results, introducing CNNs into the network significantly improves performance when compared to TDNN with the same configuration and capability in capturing long temporal dependency of 30 adjacent contexts. Thus, CNNs enable the network to efficiently learn representation of the speech signal.

In addition, error analysis was conducted to examine the inaccuracy of the model in generating the hypothesis by using SCLITE [26]. We discovered that the majority of the problems were due to phonetic and encoding issues. A phonetic mistake occurs when the model estimates a word that sounds the same as another word but has a different meaning (homophones). When the model generates the same word but with a different encoding, this is referred to as an encoding error. We contend that these issues are caused by the quality of the corpus used to build the lexicon and the language model. Tables 4 and 5 show the most common examples of phonetic and encoding errors, respectively.

Phonetic error examples:

Scores: (#C #S #D #I) 20 1 0 0
REF: ដោយសារ តែ មុខ របរ ហ្នឹង មាន ការ រីក ចម្រើន ខ្លាំង នាង ខ្ញុំ ក៏ ពង្រីក សាខា ទៅ លក់ នៅ ក្នុង ខណ្ឌ សែន (Due to the rapid growth of this business, I expanded the branch to sell in Sen district)
HYP: ដោយសារ តែ មុខ របរ និង មាន ការ រីក ចម្រើន ខ្លាំង នាង ខ្ញុំ ក៏ ពង្រីក សាខា ទៅ លក់ នៅ ក្នុង ខណ្ឌ សែន (Due to the business and rapid growth, I expanded the branch to sell in Sen district)

Scores: (#C #S #D #I) 19 1 0 0
REF: កាត់ បន្ថយ នូវ ការ ជួប ជុំ គ្នា ដែល បច្ចុប្បន្ន យើង កំពុង តែ ជួប នូវ បញ្ហា ជម្ងឺ ឆ្លង កូវីដ ដប់ ប្រាំបួន (Reducing the gatherings when we are currently facing with Covid 19)
HYP: កាត់ បន្ថយ នូវ ការ ជួប ជុំ គ្នា ដែរ បច្ចុប្បន្ន យើង កំពុង តែ ជួប នូវ បញ្ហា ជម្ងឺ ឆ្លង កូវីដ ដប់ ប្រាំបួន (Reducing the gatherings as well when we are currently facing with Covid 19)

The first example of phonetic errors, the demonstrative adjective "ហ្នឹង" (“this” in English), was predicted as the conjunction "និង" (“and” in English) when they

Table 3: Experimental results in WER(%) on each test set

Model	**News**	**Law**	**Health**	**Daily Life**	**General**
TDNN	11.53	19.18	17.57	17.57	19.17
CNN-TDNN	09.94 (↓13%)	16.02 (↓16%)	15.70 (↓10%)	14.83 (↓15%)	15.68 (↓18%)

Table 4: Phonetic Error

REF ⇒ HYP	**Freq**
ហ្នឹង ⇒ និង	233
អ្នក⇒នាក់	38
ដែល⇒ដែរ	28

are both homophones. Similarly, the particle "ដែល" of adverb "បច្ចុប្បន្ន" in second example, was predicted as adverb "ដែរ" and these two words have the same pronunciation.

Table 5: Encoding Error

REF ⇒ HYP	**Freq**
សម្ដេច⇒សម្តេច	32
ឧត្ដម⇒ឧត្តម	30

Encoding error examples:

Scores: (#C #S #D #I) 13 1 0 0
REF: ការ សង្គ្រោះ ជាតិ ជា ប្រវត្តិសាស្ត្រ របស់ សម្ដេច អគ្គ មហា សេនា បតី តេជោ ហ៊ុន សែន (National salvation is the history of Samdech Akka Moha Sena Padei Techo Hun Sen)
(សម្ដេច ⇒ 179F 1798 17D2 178A 17C1 1785)
HYP: ការ សង្គ្រោះ ជាតិ ជា ប្រវត្តិសាស្ត្រ របស់ សម្តេច អគ្គ មហា សេនា បតី តេជោ ហ៊ុន សែន
(សម្តេច ⇒ 179F 1798 17D2 178F 17C1 1785)

Scores: (#C #S #D #I) 9 1 0 0
REF: រដ្ឋ មន្ត្រី ក្រសួង ការពារ ជាតិ រួម ជាមួយ នឹង ឯក ឧត្ដម (Minister of National Defense together with His Excellency)
(ឧត្ដម ⇒ 17A7 178F 17D2 178A 1798)
HYP: រដ្ឋ មន្ត្រី ក្រសួង ការពារ ជាតិ រួម ជាមួយ នឹង ឯក ឧត្តម
(ឧត្តម ⇒ 17A7 178F 17D2 178F 1798)

In encoding error instances, coeng "ដ" (178A) was predicted as coeng "ត" (178F). As a result, the word "ឧត្តម", and "សម្តេច" were synthesized as "ឧ + ត + ្ + ត + ម" and "ស + ម + ្ + ត + េ + ច" instead of "ឧ + ត + ្ + ដ + ម" and "ស + ម + ្ + ដ + េ + ច", respectively.

5 Conclusion

In this study, we reported the state-of-the-art for dealing with Khmer ASR using CNN and TDNN. The results reveal that integrating the CNNs into the network of TDNN significantly improves the performance of the model across all test sets. In future work, a new acoustic modeling approach will be investigated to improve the overall performance of the model in terms of accuracy and decoding speed. Furthermore, due to the constrained variety of datasets in comparison to different well-resourced languages, a larger dataset is desired for future development of Khmer ASR. More data needs to be acquired and used for various data augmentation methods in order to examine more environmental factors. Khmer encoding issues and discrepancies will also be studied to eliminate encoding errors while building a corpus, and it is widely assumed that different misspelling errors are no longer a problem, as long as there is a huge amount of correct data for speech and text corpora.

Acknowledgment

We would like to express our gratitude to CADT students, particularly Mr. Rithborey Chorn and Mr. Samnang Pov, for their support in collecting and preparing the speech corpus.

References

[1] Lawrence R Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] Andreas Stolcke, Luciana Ferrer, Sachin Kajarekar, Elizabeth Shriberg, and Anand Venkataraman, “Mllr transforms as features in speaker recognition,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[3] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, “A fast learning algorithm for deep

belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] Abdel-rahman Mohamed, George Dahl, Geoffrey Hinton, et al., "Deep belief networks for phone recognition," in *Nips workshop on deep learning for speech recognition and related applications*, 2009, vol. 1, p. 39.

[5] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

[6] Magnus Stenman, "Automatic speech recognition an evaluation of google speech," 2015.

[7] Laurent Besacier, V-B Le, Christian Boitet, and Vincent Berment, "Asr and translation for under-resourced languages," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. IEEE, 2006, vol. 5, pp. V–V.

[8] Kak Soky, Vichet Chea, and Sethserey Sam, "Khmer automatic speech recognition system based on dnn models," in *First Regional Conf. on Optical character recognition and Natural language processing for ASEAN Languages (ONA), Phnom Penh, Cambodia*, 2017.

[9] Xinwei Li, Yue Pan, Matthew Gibson, and Puming Zhan, "Dnn online adaptation for automatic speech recognition," *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2018*, pp. 46–53, 2018.

[10] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong, "Lstm time and frequency recurrence for automatic speech recognition," in *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*. IEEE, 2015, pp. 187–191.

[11] Dong Yu and Jinyu Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of automatica sinica*, vol. 4, no. 3, pp. 396–409, 2017.

[12] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

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

[14] Apeksha Shewalkar, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, 2019.

[15] Muhammadjon Musaev, Ilyos Khujayorov, and Mannon Ochilov, "Image approach to speech recognition on cnn," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*, 2019, pp. 1–6.

[16] Du Guiming, Wang Xia, Wang Guangyan, Zhang Yan, and Li Dan, "Speech recognition based on convolutional neural networks," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2016, pp. 708–711.

[17] "Kaldi mini-librispeech recipe," `https://github.com/kaldi-asr/kaldi/blob/master/egs/mini_librispeech/s5/local/chain/tuning/run_cnn_tdnn_1b.sh`.

[18] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[19] "Kaldi ivector recipe," `https://github.com/kaldi-asr/kaldi/blob/master/egs/mini_librispeech/s5/local/nnet3/run_ivector_common.sh`.

[20] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, 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.

[21] Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto, "Creating corpora for speech-to-speech translation," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[22] "The sri language modeling toolkit," `http://www.speech.sri.com/projects/srilm`.

[23] Vichet Chea, Ye Kyaw Thu, Chenchen Ding, Masao Utiyama, Andrew Finch, and Eiichiro Sumita, "Khmer word segmentation using conditional random fields," *Khmer Natural Language Processing*, pp. 62–69, 2015.

[24] Kak Soky, Xugang Lu, Peng Shen, Hiroaki Kato, Hisashi Kawai, Chuon Vanna, and Vichet Chea, "Building wfst based grapheme to phoneme conversion for khmer," *Proc. Khmer Natural Language Processing (KNLP)*, 2016.

[25] "Phonetisaurus g2p," `https://github.com/AdolfVonKleist/Phonetisaurus`.

[26] "Sctk, the nist scoring toolkit," `https://github.com/usnistgov/SCTK`.