Interspeech 2018
2-6 September 2018, Hyderabad

On Convolutional LSTM Modeling for Joint Wake-Word Detection and Text Dependent Speaker Verification

Rajath Kumar[1], *Vaishnavi Yeruva*[2], *Sriram Ganapathy*[2]

[1]Department of Electrical Engineering, Columbia University, New York, NY
[2]Learning and Extraction of Acoustic Pattern Lab, Indian Institute of Science

rm3497@columbia.edu, vaishnaviy2@gmail.com, sriramg@iisc.ac.in

Abstract

The task of personalized keyword detection system which also performs text dependent speaker verification (TDSV) has received substantial interest recently. Conventional approaches to this task involve the development of the TDSV and wake-up-word detection systems separately. In this paper, we show that TDSV and keyword spotting (KWS) can be jointly modeled using the convolutional long short term memory (CLSTM) model architecture, where an initial convolutional feature map is further processed by a LSTM recurrent network. Given a small amount of training data for developing the CLSTM system, we show that the model provides accurate detection of the presence of the keyword in spoken utterance. For the TDSV task, the MTL model can be well regularized using the CLSTM training examples for personalized wake up task. The experiments are performed for KWS wake up detection and TDSV using the combined speech recordings from Wall Street Journal (WSJ) and LibriSpeech corpus. In these experiments with multiple keywords, we illustrate that the proposed approach of MTL significantly improves the performance of previously proposed neural network based text dependent SV systems. We also experimentally illustrate that the CLSTM model provides significant improvements over previously proposed keyword detection systems as well (average relative improvements of 30% over previous approaches).

Index Terms: Text dependent speaker verification, Key-Word Spotting (KWS), Convolutional Long Short Term Memory (CLSTM) Network, Multi-task learning

1. Introduction

With the rapid outreach of personalized voice activated devices, there is a growing demand for voice technology based authentication consisting of keyword spotting (KWS) (wake-up word detection) combined with text dependent speaker verification (TDSV). In the recent past, several approaches have been proposed for separately modeling the two tasks using deep neural networks (DNNs). For example, keyword spotting using DNN/CNN models with whole word input speech patterns have been explored in [1, 2]. These approaches avoid relying on the traditional approach of using automatic speech recognition (ASR) systems and therefore are particularly useful for new languages and domains where only few words are labeled. In these methods, the network is trained directly to predict the keyword of interest followed by posterior smoothing. The direct modeling also reduces the computational and memory requirements significantly.

For TDSV task, the conventional approach consists of training a Gaussian mixture model - universal background model (GMM-UBM). The UBM is adapted and i-vector representations are derived for each speaker recording [3, 4]. The i-vectors for the enrollment and test speakers are compared using a cosine distance or a probabilistic linear discriminant analysis (PLDA) model [5]. Recently, deep learning methods have been explored for TDSV task where the DNN models are not directly used for classification, but rather as a feature extractor which provides speaker specific embeddings. The models are trained for classifying training speakers and once the network is trained, the embeddings in the hidden layer are extracted for enrollment and test speakers. Typically, a simple classifier using cosine distance is used for scoring in these approaches [6, 7, 8]. With large amounts of per-speaker training data, these neural network based approaches improve over the i-vector based methods [8, 7].

The performance of the DNN based approaches for TDSV rely heavily on the availability of large amounts of background training data. However, in many practical scenarios, the amount of text dependent training data for background speakers would be rather limited. In such a scenario, the GMM-UBM based modeling continues to outperform the DNN based TDSV approaches [9]. This is primarily due to the over-training of the DNN models on the training speakers and the DNN embeddings do not generalize well to new speakers. The most common strategy to overcome this is with pre-training and regularization approaches like dropout [10]. For small amount of background training, even with dropout methods, the DNN based models (with the best choice of architecture and activation functions) provides an equal error rate (EER) which is about twice the GMM-UBM system [9].

In this paper, we propose to develop a TDSV-KWS system using multi-task network employing joint CLSTM and DNN framework where keyword spotting and speaker identification are trained. Multitask learning [11] is parallel learning of correlated tasks using shared features through structure sharing[12] or collaborative backpropagating from one system to another as shown in [13]. The proposed model uses speech spectrogram from a large contextual window (approximately of the duration of the keyword of interest) processed with CNN feature embeddings which are shared for both the KWS and TDSV tasks. In addition, MTL framework also provides a good solution to the KWS task thereby enabling the joint detection of keywords and speaker verification. While multitask learning was previously attempted for speaker verification [14] with large amounts of background training data, the proposed approach performs word detection on unrestricted speech with small amounts of background training data (similar to [9]).

We evaluate the TDSV task using the proposed MTL framework and compare the performance with previously proposed neural network architectures [8, 15]. Similarly, the KWS wake up task is compared against word based independent KWS systems [1, 2]. In these experiments, the CLSTM approach provides good improvements for KWS task. The CLSTM with

10.21437/Interspeech.2018-1759

MTL setting also shows the best results for the TDSV task. In both these tasks, the proposed model achieves significant improvements in equal error rate (EER) compared to the previously proposed neural network methods.

The rest of the paper is organized as follows. The baseline neural network based systems for KWS and TDSV are discussed in Sec. 2. This is followed by detailed explanation of the proposed multitask architecture in Sec. 3. The experimental setup with information about data used for training and testing as well as the performance measure is given in Sec. 4. The results of various experiments are provided in Sec. 5 along with a detailed analysis.

2. Baseline Systems for KWS and TDSV

2.1. Keyword Spotting

Recently, an approach for keyword spotting using whole word modeling was proposed using feedforward deep neural networks (DNNs) [1]. Here, a DNN is trained directly to predict the keyword of interest, which is followed by posterior smoothing. Unlike the conventional ASR based methods, the keyword search in this case is restricted to the window of features used in the DNN and posterior smoothing. In such a manner, the sequential search is simplified leading to low complexity keyword detection system. This approach has been advanced with the use of convolutional neural networks (CNNs)[16] which replace the DNNs.

Although, there are several approaches to KWS [1, 2, 17, 18], we focus on Keyword/Filler neural network method [1, 2] where only the speech signal corresponding to the considered keyword is given the true label and everything else is considered as false or filler. The network is trained to discriminate between the two classes. We have implemented the feedforward [1], convolutional [2], recurrent networks- Long Short Term Memory (LSTM) and Bidirectional-LSTM (BLSTM) [19] and convolutional LSTM (CLSTM) models [20].

In CLSTM model, the convolutional layer discussed in Sec. 3 generates correlated feature maps. These feature maps are then time distributed so as to retain the temporal structure of the layer (unlike the flattening operation). Thus, we obtain a feature map which accumulates all the frequencies in their respective time steps which is then given as input to the LSTM layer.

2.2. Text Dependent Speaker Verification

2.2.1. d-vector Approach

Motivated largely by the success of the d-vector approach proposed in [8], there have been several approaches reported over the recent years [9, 7, 15, 14]. In this framework, a neural network is trained using audio data and its respective speaker labels. However, the inference is drawn from the last layer before softmax. During verification, the enrollment utterances are averaged at the last layer inferences over the audio frames to obtain a single vector, termed as the d-vector. The d-vector representing the enrollment speaker is compared with the d-vector from the test utterance. Various NN architectures like feedforward architecture [8, 9], convolutional network [15] and recurrent models have been proposed for TDSV. In our experiments using these architectures for TDSV (reported in Sec. 5), the recurrent architectures performed the poorest on small dataset amongst other investigated architectures which are similar to the findings from [9]. Since our dataset is quite small, the network is prone to over-fitting. We attempt to overcome this effect

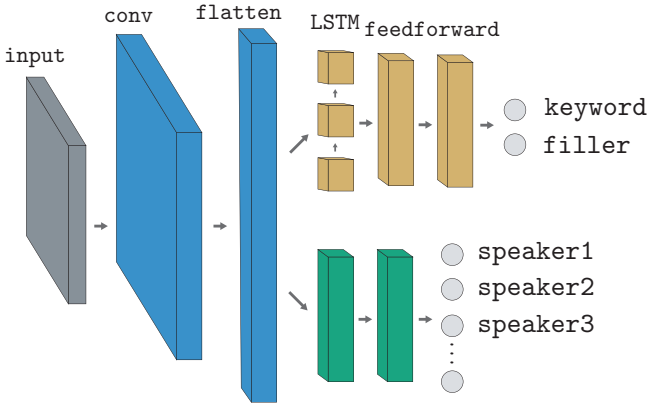

Figure 1: *Multitask model of Text Dependent Speaker Verification and Keyword Spotting. Here, convolutional (*`conv`*) is shared across both the tasks represented in blue.*

with dropout training (factor of 0.4 after the first layer and 0.2 in higher layers) in the feed forward network.

3. System Architecture

3.1. Feature Extraction

We use acoustic features f, which are described by normalized 32 dimensional log filter bank energy computed every 10 ms over a window of 25 ms. For each keyword chosen, a window size, t is considered based on the histogram spread of the keyword duration in the training data. Using the window size t, a training example is constructed with a symmetric window of acoustic features having $\frac{t}{2}$ frames on either side of every speech feature vector. The center frame of this window is then associated with the corresponding label that indicates the presence or absence of the keyword in the chosen t frame window of acoustic features. We call this a positive example when the keyword is part of the t frame window and a negative example when it is not. During the training, a portion of the examples that lie in the boundary region of the chosen keyword are omitted. At the end of this process, we get $n \times (t \times f)$ features, where n is the number of training examples.

3.2. Multitask Network Architecture

Text dependent speaker verification benefit by phonetic features from (keyword spotting model), thus structure sharing is the preferred multitask approach. Particularly, we share the lower level representations of the network across the tasks as shown in Fig. 1. The convolutional layer performs a two-dimensional non-linear filtering of the input speech spectrogram which tends to reduce the spectral variance present in speech utterances and allows the modeling of local spectro-temporal correlations.

The first layer is a convolutional layer consisting of a convolution and max-pooling operation. The convolution operation is achieved by weight sharing across the entire training sample [16]. We use 128 filters and ReLU activations. Here, each training sample is given as a matrix $(t \times f)$, on which 2-D convolution is operated without zero-padding using a kernel size, $(m \times n)$. This outputs k feature maps each of size $(x \times y)$

		train				test (SPK)				test (KWS)	
Keyword	t	nb spk	nb utt	avg utt	min-max	nb spk	nb utt	avg utt	min-max	+ class	- class
government	75	84	834	9.9	6-29	94	419	4.5	4-5	1296	11664
company	71	144	1541	10.7	8-20	185	995	5.4	4-7	1901	17109
hundred	59	177	1776	10.0	8-14	347	1814	5.2	4-7	1901	17109
nineteen	79	144	1516	10.5	8-20	129	668	5.2	4-7	984	8856
thousand	77	136	1217	8.9	6-22	208	914	4.4	4-5	1901	17109
morning	69	132	976	7.4	6-15	279	1201	4.3	4-5	1901	17109
business	81	116	776	6.7	5-16	99	396	4.0	4-4	1901	17109

Table 1: *Specifications of the sub-dataset created using the combined recordings of Wall Street Journal (WSJ) and Librispeech. Here, t - window size, nb spk - number of speakers; nb utt - number of utterances; avg utt - average utterance; min-max - minimum and maximum number of utterances occurred per speaker.*

where x is $(t - m + 1)$ and y is $(f - n + 1)$. We use non overlapping frequency dominant pooling with kernel size (1×4). This choice of max pooling kernel was motivated by [2]. After the pooling layer, the resultant output will be of dimension,

$$(\frac{x-p}{s} + 1) \times (\frac{y-q}{v} + 1) \quad (1)$$

This is flattened and passed on to the feedforward layers for speaker mapping as shown in Fig. 1. The resultant CNN feature map of dimensions given in Eq. 1 is also processed as temporal vector sequence along the x dimension using a long short term memory (LSTM) network recurrent architecture [20] on the keyword branch. Thus, the CLSTM model is capable of modeling the local spectro-temporal correlations of the speech spectrogram (using the early CNN layers) as well as the long term dependencies in the speech utterance (using the LSTM layers). While this model has shown promise for speech recognition [21], this paper illustrates the first attempt using CLSTM models for joint TDSV and KWS task.

3.3. Posterior Handling

3.3.1. Keyword Spotting

The raw posteriors are taken from the softmax layer across the entire sequence in observation. Smoothing is applied over a window of size 10. This is done so as to eliminate spurious frames. The max value across the entire sequence is used when making the decision. The metrics used in this work are at the word level and not frame level (similar to the previous approaches in [1, 2]).

3.3.2. Speaker Verification

When evaluating speaker verification task, the raw posteriors are taken from the last hidden layer as described in [8]. The posteriors are $L2$ normalized and averaged across the frames of an utterance to obtain the d-vector. During enrollment, the final d-vector representing the speaker is derived by averaging the d-vector across the enrollment utterances. While testing, cosine distance is used to compare the d-vector obtained from the test utterance and the claimed speaker's d-vector. Other classifiers such as PLDA can also be employed instead of cosine distance metric as shown in [14]. In our experiments, only 3 enrollment utterances were used for computing the d-vector of the speaker.

3.3.3. Performance Metrics

The performance of both the tasks are evaluated by plotting a receiver operating characteristic(ROC) curve. Here, the false reject rate (FRR) is computed per false alarm rate (FAR) by varying thresholds and an equal error rate (EER) is obtained. The Area under the Curve (AUC) obtained by plotting True Alarm Rate (TAR) against FAR is also tabulated.

4. Experimental Setup

4.1. Data

We use a Kaldi recipe on the Librispeech corpus [22] to build a deep belief network (DBN) - DNN ASR system. After the ASR model training, the training data is forced aligned to generate the ground truth labels for our task. The training and validation data together consists of 1000 hours of speech sampled at 16 kHz and contains 292367 sentences spoken by 2484 speakers. The WSJ (WSJ0 and WSJ1) corpus containing 39923 sentences (test and train data combined) spoken by 381 speakers is also force aligned with the Librispeech trained ASR model to generate labels. We note that all the WSJ data in our systems use the clean WSJ corpus ($wv1$ microphone). Pooling the two corpora, seven keywords $-$ *business, company, government, hundred, morning, nineteen and thousand* were selected and a sub-dataset containing recordings of the each keyword considered was formed. The specifications of each of the sub-dataset is tabulated in Table. 1. Here, test data mentioned for speaker verification task is a combination of both enrollment and testing utterances and is a held-out set. The enrollment utterances were randomly selected per speaker from this held-out test dataset. For keyword spotting wake up task, sentences without the keyword are also added to the test data also during evaluation (to measure the false alarm rates accurately). However, the train data tabulated in Table 1 remains the same across all experiments. For TDSV, only the speech segments of the keyword are picked from the train data, while the entire dataset is considered for KWS.

4.2. Training

We use a fixed batch size of 128 and a stochastic gradient descent with momentum algorithm for the optimization task. The method of bold driver learning rate parameter with exponential decay [23] is adapted in the following manner - if the accuracy on a validation set decreases after an epoch, then the weights of the previous epoch are restored and the learning rate is halved. The training process is stopped when there is no positive increase in the accuracy even after reducing the learning rate thrice. The initial learning rate for all experiments are kept at 0.02. Cross entropy as loss function is utilized.

In multitask network, the gradients in shared layer are propagated as follows,

$$\frac{\partial L_{spk}}{\theta_{shared}} + \frac{\partial L_{kws}}{\theta_{shared}} \quad (2)$$

	DNN [8]		CNN [15]		LSTM [7]		BLSTM		CLSTM		CLSTM-MTL		I-VEC	
Keyword	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC
government	18.9	89.6	13.9	93.8	25.8	82.4	22.9	84.4	16.3	91.0	**13.2**	**93.8**	10.0	96.3
company	14.0	93.7	9.0	96.7	14.4	93.1	15.4	92.3	12.4	95.0	**8.7**	**97.1**	7.7	97.5
hundred	18.8	89.7	10.5	96.0	18.8	89.5	18.7	89.5	12.4	94.6	**9.5**	**96.6**	9.2	96.6
nineteen	13.1	94.4	7.5	97.8	12.5	94.6	13.4	94.1	8.2	97.1	**6.2**	**98.5**	3.1	99.5
thousand	14.1	93.7	9.9	96.2	19.5	88.4	20.7	88.1	11.7	95.1	**9.0**	**96.7**	7.2	97.5
morning	19.1	88.7	15.5	92.5	24.2	83.6	25.2	82.8	18.2	89.8	**14.2**	**93.6**	11.8	95.3
business	17.4	91.5	10.2	96.4	15.1	92.5	16.2	91.7	11.9	95.2	**9.5**	**97.0**	12.9	93.7
Average	16.5	91.6	10.9	95.6	18.6	89.2	18.9	89.0	13.0	94.0	**10.0**	**96.2**	8.8	96.6

Table 2: *Text Dependent Speaker Verification results for various neural network architectures and i-vector system trained on short utterances and limited number of training samples. The highlighted values are the best amongst neural network system.*

	DNN [1]		CNN [2]		LSTM [17]		BLSTM		CLSTM		CLSTM-MTL	
Keyword	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC
government	9.3	96.9	6.0	98.7	7.5	96.3	6.9	97.1	**3.3**	**98.9**	8.5	96.3
company	6.5	98.1	5.4	98.5	5.7	96.8	7.6	96.2	**4.3**	**98.1**	9.0	95.6
hundred	13.8	93.6	10.4	94.2	11.6	94.2	11.4	95.2	**7.2**	**97.3**	12.7	93.3
nineteen	9.6	96.6	7.9	97.8	7.8	96.6	7.7	96.3	**5.6**	**98.1**	7.1	97.1
thousand	7.5	98.0	5.9	97.7	5.5	97.8	5.7	98.0	**4.3**	**98.8**	6.4	97.5
morning	7.6	97.8	4.6	99.0	4.7	98.2	6.9	97.1	**4.1**	**99.3**	6.0	97.5
business	7.0	98.1	5.7	98.9	4.5	98.3	4.8	98.4	**3.3**	**98.9**	6.9	97.5
Average	8.7	97.0	6.6	96.7	6.8	96.9	7.3	96.9	**4.6**	**98.5**	8.1	96.4

Table 3: *Keyword spotting results for various neural network architecture*

The error from the text dependent SV network is backpropagated only for the keyword frames while both keyword and non keyword error are backpropagated in the keyword branch. To accommodate this weight imbalance, only during multitask training the batch size is increased in proportion to keyword/filler ratio such that TDSV branch sees approximately 128 samples of keyword during each iteration which is the batch size of all the baseline models in consideration.

5. Results

For completeness, an i-vector based TDSV is also implemented in this paper. The i-vector features are derived using a 512 mixture component GMM-UBM. This is followed by a total variability matrix model for dimensionality reduction to 256 dimensions.

Note that, to have a fair comparison all the models are designed with approximately similar parameters. In the multitask, each branch has the same parameter as it's respective baseline. The results for various NN architectures for TDSV is reported in Table 2. Amongst the neural network architectures for speaker verification the feed-forward and recurrent architectures perform the poorest. The convolutional neural network model [15] is the most effective at capturing speaker specific features. While the stand-alone CLSTM model performs poorly, sharing the lower level phonetic features has proven to be beneficial and performs better than the CNN model. The i-vector model shown in the last column of the Table further improves over the neural network architectures. This also validates previous observations made along the same lines comparing neural network architectures with i-vector models for small amounts of training data [9].

In the case of KWS wake-up word detection, recurrent architectures show significant improvements over the feed-forward model. The CLSTM model provides the best KWS accuracy among various NN models considered. The average relative improvements of 30% is achieved for the proposed CLSTM model compared to the previous models based on LSTM framework [17]. Contrary to the results of speaker verification model, The MTL framework does not improve the CLSTM architecture. This may be attributed to the fact that preserving speaker information may be diluting the goal of the KWS task which attempts to derive the keywords irrespective of the target speaker.

In summary,

- The recurrent LSTM architectures are most suitable for phonetic and word classification while the convolutional architectures are suitable for both speaker and phonetic features.
- Convolutional front-end feature maps combined with recurrent architectures is suitable for learning shared features (speaker and phonetic).
- The MTL framework in combination with the CLSTM model provides significant benefits for speaker verification where the knowledge of phonetic information helps in speaker clustering. However, the speaker information in MTL is not beneficial for KWS task.
- The neural network approaches with small amounts of speaker training data do not perform as well as the i-vector features in TDSV task.

6. Conclusion

Concluding, we have investigated various neural network architectures for text dependent speaker verification and keyword spotting and proposed a multitask architecture with a convolutional front-end. We have also demonstrated the effectiveness of learning shared feature representations of phonetic and speaker features for the speaker verification task.

7. Acknowledgements

The authors would like to thank Amith Manchanda and Venkata Prajwal for constructive discussions. The authors would also like to thank Dhanush Bekal and Harish Haresamudram for their help with i-vectors.

8. References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[2] T. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015.

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[5] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[6] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[7] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[8] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[9] G. Bhattacharya, J. Alam, T. Stafylakis, and P. Kenny, "Deep neural network based text-dependent speaker recognition: Preliminary results."

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

[11] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[12] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7304–7308.

[13] Z. Tang, L. Li, D. Wang, and R. C. Vipperla, "Collaborative joint training with multi-task recurrent model for speech and speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[14] N. C. Y. Q. K. Yu, "Multi-task learning for text-dependent speaker verification," 2015.

[15] Y.-h. Chen, I. Lopez-Moreno, T. N. Sainath, M. Visontai, R. Alvarez, and C. Parada, "Locally-connected and convolutional neural networks for small footprint speaker recognition." in *INTERSPEECH*, 2015, pp. 1136–1140.

[16] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[17] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5236–5240.

[18] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 421–426.

[19] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv preprint arXiv:1703.05390*, 2017.

[20] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[21] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[23] R. Battiti, "Accelerated backpropagation learning: Two optimization methods," *Complex systems*, vol. 3, no. 4, pp. 331–342, 1989.