Two-stage Strategy for Small-footprint Wake-up-word Speech Recognition System

Xinya You*, Yajie Zhao†, and Mingyuan Sun‡
*Heilongjiang University
†Columbia University in the City of New York
‡Northeastern University
yxyrihanna123@outlook.com, yz3231@columbia.edu, 20185881@stu.neu.edu.cn

***Abstract*—In this paper, we propose a small-footprint wake-up-word speech recognition (WUWSR) system with two stages to recognize a two-syllable wake-up word. In the first stage, convolution neural network (CNN) is trained to predict the posterior probability of context-dependent state. Thus a wake-up-word is detected according to the confidence score obtained by dynamic programming. In the second stage, we cascade bidirectional long short-term memory network (LSTM), convolutional modules and deep feed-forward network (BLCDNN) successively to verify the detection. The first stage quickly filters out speech without wake-up word, and the second stage refines the detection. In addition, without the intervention of any decoding modules, the proposed system can guarantee low latency. The experimental results demonstrate the effectiveness of this method. Our system, named CNN-BLCDNN, reaches high accuracy and maintains low false alarm rate.**

***Index Terms*—wake-up word speech recognition, human-computer interaction, deep neural network.**

I. Introduction

Systems that can recognize the specific word and initiate voice input are facing increasingly more demands. This kind of systems are supposed to reach high accuracies with low latency and low computational consumption. WUWSR system provides a practical solution to this issue. In an audio stream, WUWSR system retrieves and identifies the wake-up words in the speech, rejects words outside the wake-up-word vocabulary and understands the content.

Researchers contribute considerable efforts to WUWSR system. Compact WUWSR system adopts a template matching algorithm with distance-based scores [1]. [2] treats the consistency of the spatial eigenspace formed by speech source at different frequencies and the resonant curve similarities of the wake-up words as features, then detects wake-up words by a Bayes risk detector.

Recently, keyword spotting-based methods with deep models have been reported: (1). [3], [4], where wake-up words are set as keywords, uses deep feed-forward network (DNN) and CNN to build systems. Without constructing lattice for decoding, these systems can guarantee low latency. Though DNNs predict keywords or sub-keywords directly, they fail to model context influence. Besides, these approaches lack flexibility. Retraining is required when keyword sets alter; (2). A typical exemplar-based keyword spotting method is proposed in [5], where LSTM-based feature extractor converts speech with keywords and testing speech to fixed-length templates and testing feature vectors respectively. Finally, it makes decisions based on the differences between the templates and testing vectors. This method can identify specific speakers. [6] investigates LSTM-based feature extractor by adopting different model units and two dynamic time warping (DTW) approaches. The methods above are based on template matching. Though they have low computational complexities and are easy to be deployed on embedded devices, the accuracy is limited. There are also some methods based on speech recognition framework. [7], [8] reveal the effects of different features and devote them to WUWSR system to promote performance. An end-to-end DNN system, a connectionist temporal classification framework, is transferred to spot wake-up words [9]. Then a refinement step is applied to updating parameters. A DNN-based WUWSR system with two-stage detection is proposed in [10]. This method uses deep acoustic model and a customized decoder to recognize wake-up words. Support vector machine (SVM) works as the subsequent classifier. In paper [11], LSTM is used to model sequential information. Besides, the normalization method and the dynamic search enable the network can detect and recognize multiple wake-up words and be run on real time. [12] modifies the original DTW to generate a gray scale image. A CNN is trained on these images to classify the presence or absence of keyword. Paper [13] combines DTW and CNNs to develop an ASR-free keyword spotter.

There exists massive speech without wake-up words in general environments, which are constantly collected by receivers. It tends to cause a lot of false alarm (FA), which means frequent wrong activations. However, manual reductions of these undesired activations may omit useful speech containing wake-up words. It is necessary to hold high accuracies and low FA rates. Simple systems are hard to meet these requirements, while sophisticated models with high accuracies cause delay easily because of their heavy loads and high computational consumptions.

LSTM-based WUWSR [11] has high model complexity and poor forward propagation efficiency. Besides, in the first stage of [10], DNN fails to model frequency variations. In the second stage, SVM classifier has limited generalization ability because of the hand-designed features. Most products function normally under the $\geq$ 3-syllable words condition,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

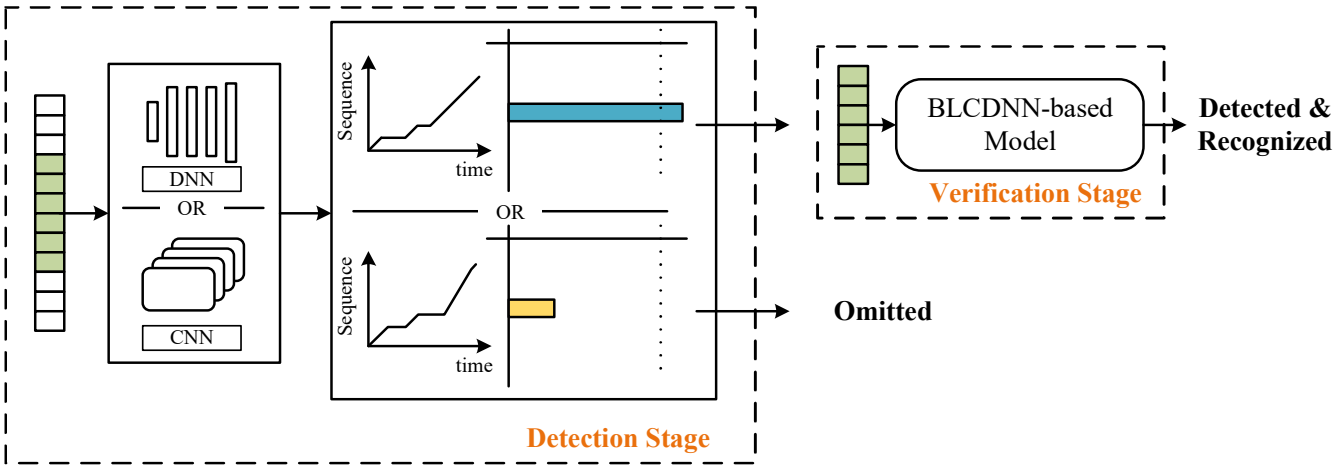

Fig. 1. The detailed architecture and pipeline of two-stage-based WUWSR.

but users tend to pronounce simple orders like 2-syllable wake-up words. Due to the more frequent homophony, the implement of 2-syllable supported systems is far more difficult than $\geq$ 3-syllable circumstances. Thus, we build a novel two-stage WUWSR system that can recognize two-syllable words. The aim of our first stage is to quickly detect the wake-up words by using a lightweight model. In this stage, most speech without wake-up words is screened out. In the second stage, the detected speech is further verified, and we predict whether the speech contains a certain wake-up word or not. This stage can markedly reduce FA while holding high accuracies.

From the statistics collected by smart devices less than 10% of signals collected by devices contain wake-up words. Because most speech filtered out by the first stage contain no wake-up word, the second stage has no need to work in most cases. Thus our strategy can achieve low computational cost while ensuring good performances. The detailed structure of the proposed two-stage strategy-based WUWSR is depicted in Figure 1. Specifically, in the first stage, a CNN is trained to predict the posterior probability of context-dependent state. Then dynamic programming produces confidence score. The wake-up word is recognized when corresponding confidence score exceeds the predefined threshold. Moreover, phonetic knowledge is integrated with model-based classification method to detect wake-up word. The speech segment which passes the first stage will be sent to the next stage. Then BLCDNN is utilized to determine whether this segment is a wake-up word. Otherwise, it will be omitted and the detect process will continue to the next speech segment.

Compared with CNN-based baseline, CNN-BLCDNN achieves an 83.44% relative FA rate decrease under the same operating point, which indicates that BLCDNN can effectively reduce FA. The performance of the overall system is measured by equal error rate (EER). For EER, compared with the baseline, our CNN achieves a 15.94% relative improvement. By combining CNN and BLCDNN, the best WUWSR system can achieve a 69.93% relative improvement over DNN-based system and a 31.40% relative improvement over DNN-BLCDNN.

The rest of this paper is organized as below. Section II introduces the model and modules utilized in the first stage. Section III introduces the model used in the second stage. Data preparation and experimental setup are described in Section IV, while the results of our experiment are presented in Section V. Finally, Section VI gives conclusions and discussions.

II. CNN-BASED DETECTION STAGE

The first stage can be divided into three sub-modules: feature extraction module, deep neural network module and dynamic programming search module: Feature extraction module (in Section IV-B) converts original signals into features; deep neural network module receives features and outputs the posterior probability of each frame; dynamic programming search module converts posterior probabilities to confidence score. A wake-up word is obtained when the confidence score exceeds a threshold.

A. Deep neural network module

The deep neural network works as conventional acoustic models. We firstly train a GMM-HMM system, and the training labels are context-depend HMM states.

In this stage, we are more interested in rapidly selecting speech that contains a wake-up word from abundant voice input from the environment. Besides, our model needs to be small-footprint and has low-computing cost. Therefore, CNN is applied to the acoustic model in the first stage.

Convolutional layers are good at modeling frequency variations [14]. Some acoustic variations can be effectively normalized and the resultant feature representation may be immune to speaker variations, colored background and channel noises. Besides, filters that work on local frequency region can efficiently represent local structures and properly describe their combinations, which contribute to classifications. Since the output nodes display the tied-triphone states, the number

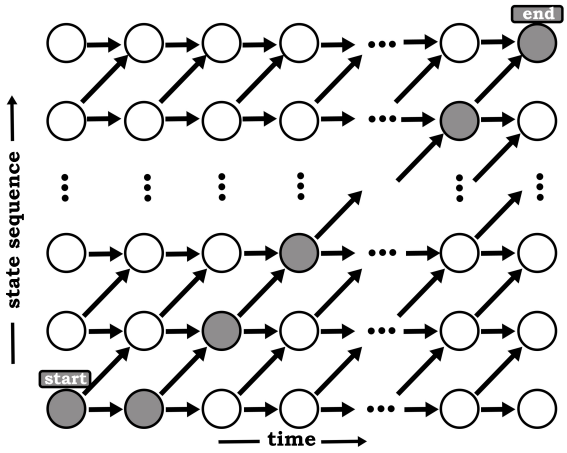

Fig. 2. Diagram of dynamic programming search

is far more than the number of hidden nodes, which means the output layer has more parameters than others. In order to further compress the model while maintaining performance, low-rank matrix factorization [15] is applied to the output layer.

Our CNN occupies 5 convolutional layers and 3 fully-connected (FC) layers. In (inChannel, outChannel, kernelW, kernelH) format, the convolutional layers line up by (1, 4, 5, 5)-, (4, 4, 3, 3)-, (4, 8, 5, 5)-, (8, 8, 3, 3)- and (8, 8, 1, 1)-convolution layer from shallow to deep, and have no pooling and share 1 stride. The FC layers have 3 layers with 512, 128, 5191 nodes respectively. ReLU works as the activation units, and layernorm [16] is adopted to further promote performance.

B. Dynamic programming search module

Dynamic programming search (DPS) module is applied to converting posterior probability into confidence score. A wake-up word is recognized if its confidence score exceeds the predefined threshold.

In this work, wake-up word is firstly converted to state sequence using GMM-HMM alignment. Different wake-up words can be represented by different state sequences. Secondly, state sequences are used to build DPS graphs. For convenience, we describe the following computation under single-wake-up-word case. Thirdly, DPS module processes speech in real time, thus the graph is built in real time. The posterior probability for each frame is appended to DPS graph in chronological order. A DPS graph is shown in Figure 2. The ordinate of the graph represents the state sequence in order and the abscissa represents the frame chronologically. In this experiment, wake-up word is two-syllable, mostly covering 15-40 frames. Therefore, the graph is created from the start frame forwarding to 15-40 frames, with a total of 25 graphs constructed synchronously for one time. For each involved graph, we operate dynamic programming to get confidence score. Finally, from the first frame, we just follow the previous step to iterate to the last frames.

In dynamic programming, starting from the beginning of the graph, the state can only be specified to the next state in order or stay in the original state. Finally, the state must reach the end of the graph. The set of gray dots in Figure 2 indicates the path with the highest confidence score in the graph. The recursion formula of dynamic programming is:

$$CS_{i,s} = MAX\{CS_{i-1,s}, CS_{i-1,s-1}\} + p_{i,s}, \tag{1}$$

where i is time, s is state index. $CS_{i,s}$ is the highest confidence score from beginning to (i, s). $p_{i,s}$ is the poster possibility of state s in frame i. The confidence score of this graph is:

$$CS_{end} = CS_{end}/FrameNum, \tag{2}$$

where $FrameNum$ is the total number of frames used to build the graph, and CS_{end} represents the highest confidence score from beginning to end point. If the confidence score exceeds the threshold, the search is stopped and wake-up word is considered to be detected. At the same time, the start and end frames of wake-up word are also recorded. Otherwise, it lasts until the end frame. In practice, for one wake-up word, there might be alternative pronunciations in different accents, which should also be considered. We address this issue by constructing multiple graphs that represent the alternative pronunciations.

The advantages of dynamic programming search are obvious. Our method can run in real time and the content of wake-up word can be recognized at the same time because of the acoustic information which is brought by state sequence. Besides, we can easily replace wake-up word by changing the state sequence when building the graph. By building several search graphs, multiple wake-up words can be recognized at one time. In this process, there is no need to re-train CNN. We only need to add new BLCDNN in the second stage, which will be described in detail in section III.

DTW [17], [18], [19] and DPS are all based on dynamic programming. But the difference is obvious. DTW is used to measure the similarity of two templates with different lengths. Each node in the constructed graph represents the distance between the two templates. Besides, DTW has three search directions. The proposed DPS measures the probability that a wake-up word whether located in all sequential speech segments or not. Each node in the constructed graph represents posterior probability, which means the probability that the current frame belongs to that state. As depicted in Figure 2, DPS is operated on time sequences, thus there are only two search directions.

III. BLCDNN-BASED VERIFICATION STAGE

In this section, an utterance-level BLCDNN-based verification model is conducted to further determine whether there is a wake-up word in the speech. The architecture is diagrammed in Figure 3. Only speech that is judged as the host of wake-up word in the first stage can be sent to the second stage, where the speech is cut into segment according to the start and end point, and the segment is fed into the verification model.

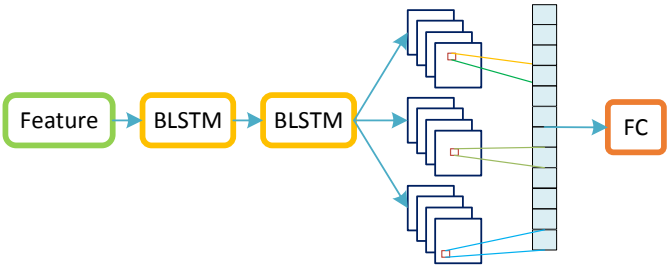

Fig. 3. BLCDNN-based verification model.

We aim to use BLSTM to model the dependencies of the speech and CNN to extract discriminative features that are useful for the classification task. In convolutional layer, to enable the network to extract complementary features and enrich the representation, we learn several different filters simultaneously. Convolutional filters with multiple sizes capture valuable features from different scales, which benefit a lot to robust classification. The feature maps produced by the convolution layer are forwarded to the pooling layer. 1-max pooling is employed on each feature map to keep the most dominant feature only. The dominant features are then concatenated to form a feature vector to be fed to final layer. This step transforms the variable-length, high-dimensional vector into a fixed-length form. Finally, a FC layer maps it to two output nodes. One of the nodes indicates that the segment is related to the wake-up word and the other node represents the segment without wake-up word.

BLCDNN model consists of 2 BLSTM layers, 1 convolutional layer and 1 FC layer. Each BLSTM layer has 128 units. In (kernelW, kernelH) format, the convolutional layer has 3 different filter sizes that are (5, 5), (3, 3) and (1, 1) both with 4 output channels and 1 stride. Final results can be obtained by analyzing the output of BLCDNN.

IV. Experimental Setup

A. Data preparation

Extensive experiments have been performed to demonstrate the effectiveness of the proposed small-footprint WUWSR system. Here, we select one two-syllable wake-up word, which is '{ruò}-{qí}' in Mandarin Chinese Pinyin. 12 DPS graphs are constructed, which represent different pronunciations of the wake-up word. For CNN acoustic model in the first stage, close-talk dataset contains 3000-hour speech in training set and 10-hour speech in development set.

In the second stage, training set contains 559400 utterances that contain wake-up word and 445072 negative samples. Development set has 20089 utterances, which has 10000 positive samples. We evaluate performance in one test set, which contains 10451 positive samples and 100000 negative samples. All the utterances utilized in the second stage and test set are collected from our circular array with eight microphones. Utterances are recorded in rooms with variant sizes, and the ratio of the utterances of male to female is almost 6:4. The distance between speaker's mouth and microphone array varies from 1m to 5m. The signal-to-noise ratio (SNR) ranges between 10dB and 20dB. In our study, the development set is used to choose the model. The experimental results are all evaluated on the test set.

B. Training preparation

The multi-channel speech collected by microphones is enhanced by a minimum variance distortionless response beamformer [20], and single-channel speech segment is divided into frames. In the first stage, DNN and CNN are conducted as frame level. For each frame, acoustic features of DNN are generated based on 29-dimensional log-mel filterbank features along with their first and second order derivatives. 5 past frames and 5 future frames are appended to current frame, constituting a total of 957-dimensional feature vector. For CNN, acoustic features are generated based on 29-dimensional log-mel filterbank features. Similarly, 5 past frames and 5 future frames are appended to current frame, constituting a total of 319-dimensional feature vector. The alignments are generated by a well-trained GMM-HMM system with 5191 senones. In the second stage, BLCDNN processes the input in utterance level. The input is 29-dimensional log-mel filterbank features. The features are extracted by Kaldi [21] and the models are all trained on Tensorflow [22]. Adam algorithm [23] is utilized for training where the base learning rate is 0.001.

V. Results

A. Baseline systems

First, a DNN-based WUWSR baseline system that works in the first stage is built. The baseline represented as DNN-1stage uses a network with 4 hidden layers with 512 nodes and 1 hidden layer with 128 nodes. ReLU non-linearity and layernorm [16] are applied to each hidden layer. Another baseline integrates DNN-1stage and an additional BLCDNN, which is represented as DNN-BLCDNN.

Results are presented in three aspects. The first one is to evaluate FA rate and accuracy under a fixed operating point of the 1-st stage. In this experiment, the operating point is randomly selected to make the system in the first stage reaches 98% accuracy, thus the accuracy and FA rate of the first stage can be obtained. The commercial operation points are referred to customer needs. The whole system is measured by the change in accuracy and FA rate. The second evaluation metric is EER, which represents the point at which false rejection rate and FA rate are equal. Obviously, EER measures the overall performance of the system. The Lower EER indicates the better performance of the WUWSR system. The third one is receiver operating characteristic (ROC) curve, the better performance can be obtained if the ROC curve is closer to the upper left corner.

B. CNN-based detection stage

In this section, we explore the impact of applying CNN to WUWSR system, and CNN-1stage is constructed. From Table I, we can quantitatively assess the system performance.

TABLE I
Comparison of WUWSR systems with different structures.

System	Parameters (million)	Fixed operating point		EER (%)	EER Relative Improvement (%)
		Accuracy (%)	FA rate (%)		
DNN-1stage	2.28	98.00	3.82	2.76	–
CNN-1stage	2.30	98.00	3.08	2.32	15.94
DNN-BLCDNN	$2.28 + 1.06$	97.23	0.81	1.21	56.16
CNN-BLCDNN	$2.30 + 1.06$	97.36	0.51	0.83	69.93

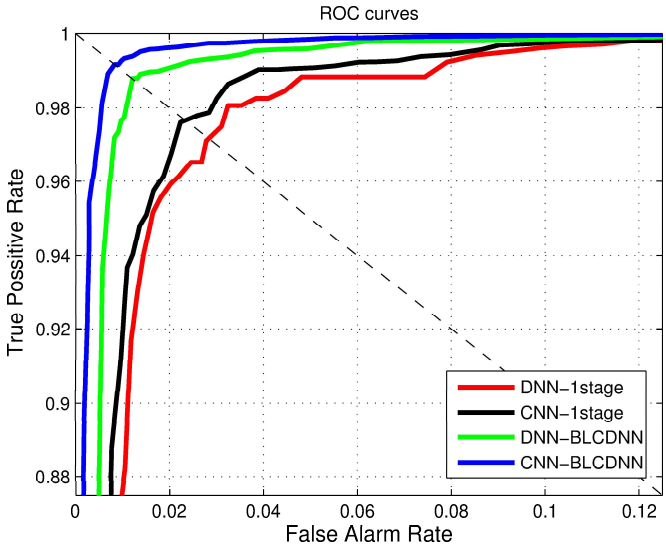

Fig. 4. ROC curves of different WUWSR systems

With the help of CNN, CNN-1stage achieves improvement in FA rate with the same accuracy. EER decreases significantly. Compared with DNN-1stage, the EER of CNN-1stage decreases from 2.76% to 2.32%, with a relative improvement of 15.94%. The experimental results show that CNN can improve the performance of WUWSR.

C. Two-stage-based WUWSR system

In this section, two-stage-based WUWSR systems, DNN-BLCDNN and CNN-BLCDNN, are constructed. The operating points are the same as the first stage. By using BLCDNN, the accuracy keeps stable, but FA rate decreases dramatically. Compared with CNN-1stage, the FA rate of CNN-BLCDNN decreases from 3.08% to 0.51%, improved 83.44% relatively. This shows that BLCDNN can be used to effectively reduce FA while maintaining performance. The system performance can be evaluated quantitatively by EER. The results are all summarized in Table I. Compared with DNN-1stage, DNN-BLCDNN achieves a 47.84% relative improvement, from 2.32% to 1.21%. Compared with CNN-1stage, CNN-BLCDNN achieves a 64.22% relative improvement, from 2.32% to 0.83%. In conclusion, compared with DNN-1stage, CNN-BLCDNN achieves a 69.93% relative improvement, from 2.76% to 0.83%. Compared with DNN-BLCDNN, CNN-BLCDNN achieves a 31.40% relative improvement, from 1.21% to 0.83%. The experimental results show that, by using CNN and two-stage strategy, WUWSR system can be effectively improved.

ROC curves are depicted in Figure 4. It is obvious that CNN-BLCDNN is closer to the upper left corner than others. It indicates that under the same false alarm, CNN-BLCDNN has higher true positive rate. The points of EER in Figure 4 are the crossover points of dotted line and ROC curves.

D. Model size and efficiency

As can be seen from Table I, employing CNN brings performance improvement while a very small increase in overall parameters. Besides, low-rank matrix factorization is applied to compressing the model. These make the system stay in small-footprint. In commercial situations, a mass of speech and noise without wake-up words is constantly received by the smart device. In the first stage, a simple and rapid detection is applied to screening out speech without wake-up words. Only a little speech is fed into the second stage for further verification. Thus BLCDNN is omitted customarily. This strategy greatly reduces the calculations and delays, which also achieves high accuracy and low FA.

VI. Conclusions

In this paper, we propose a small-footprint WUWSR system with two-stage strategy. The first stage quickly detect the wake-up word by using a lightweight model. If the segment is detected as the wake-up word, the second stage is used to verify the detection. The proposed method successfully processes two-syllable-based wake-up word, and the phonetic knowledge of the wake-up word is also acquired by using dynamic programming search. Experimental results show the effectiveness of the method. Our system successfully recognizes two-syllable word while maintaining high accuracy and low FA. From the view of EER, our system, CNN-BLCDNN, achieves a 69.93% relative improvement over DNN-1stage and a 31.40% relative improvement over DNN-BLCDNN. Our method is easy to implement and can be applied in practice. Besides, no need for updating or adjusting hand-designed features, our system can achieve satisfied performance in a varied environment just by increasing the training data of the second stage.

References

[1] A. H. Xing, T. Li, J. L. Pan, and Y. H. Yan, "Compact wake-up word speech recognition on embedded platforms," in *Applied Mechanics and Materials*, vol. 596, 2014, pp. 402–405.

[2] J.-S. Hu, M.-T. Lee, and T.-C. Wang, "Wake-up-word detection for robots using spatial eigenspace consistency and resonant curve similarity," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3901–3906.

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

[4] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *INTERSPEECH*, 2015, pp. 1478–1482.

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

[6] J. Hou, L. Xie, and Z. Fu, "Investigating neural network based query-by-example keyword spotting approach for personalized wake-up word detection in Mandarin Chinese," in *IEEE International Symposium on Chinese Spoken Language Processing*, 2016, pp. 1–5.

[7] C.-T. Shih, "Investigation of prosodic features for wake-up-word speech recognition task," Ph.D. dissertation, Florida Institute of Technology, 2009.

[8] V. Këpuska and T. Klein, "A novel wake-up-word speech recognition system, wake-up-word recognition task, technology and evaluation," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e2772–e2789, 2009.

[9] S. Zhang, W. Liu, and Y. Qin, "Wake-up-word spotting using end-to-end deep neural network system," in *IEEE International Conference on Pattern Recognition*, 2016, pp. 2878–2883.

[10] F. Ge and Y. Yan, "Deep neural network based wake-up-word speech recognition with two-stage detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2761–2765.

[11] C. Li, L. Zhu, S. Xu, P. Gao, and B. Xu, "Recurrent neural network based small-footprint wake-up-word speech recognition system with a score calibration method," in *2018 24th International Conference on Pattern Recognition*. IEEE, 2018, pp. 3222–3227.

[12] R. Shankar, C. Vikram, and S. M. Prasanna, "Spoken keyword detection using joint dtw-cnn." in *Interspeech*, 2018, pp. 117–121.

[13] R. Menon, H. Kamper, J. Quinn, and T. Niesler, "Fast asr-free and almost zero-resource keyword spotting using dtw and cnns for humanitarian monitoring," *Interspeech*, pp. 2608–2612, 2018.

[14] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4277–4280.

[15] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6655–6659.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[17] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

[18] O. Ghitza and M. M. Sondhi, "Hidden Markov models with templates as non-stationary states: an application to speech recognition," *Computer Speech & Language*, vol. 7, no. 2, pp. 101–120, 1993.

[19] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-based continuous speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1377–1390, 2007.

[20] J. Bitzer and K. U. Simmer, "Superdirective microphone arrays," in *Microphone arrays*, 2001, pp. 19–38.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

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