A NEW APPROACH FOR SPEECH KEYWORD SPOTTING IN NOISY ENVIRONMENT

Peiwen Ye and Hancong Duan

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

ABSTRACT

Keyword Spotting works to detect wake-up keywords in a continuous voice stream, which is widely used in products such as mobile devices and smart home. Recently, DNNs dominate keyword spotting and dramatically improve performance. However, few researchers concerned about noise in speech keyword recognition. Thus, we propose an architecture for the detection under noisy scenario. Our framework combines attention mechanism and residual structure based on the CNN backbone. In addition, we use separable convolution to reduce the number of model's parameters, which makes it applicable in the embedded devices. Noises from various scenes are utilized for data augmentation to boost performance. The proposed method achieves an accuracy of 94.93% on the noisy test set based on the Google Speech Commands dataset. We also compare performance between the proposed method and RNN-based algorithm, and prove our model achieve higher accuracy with fewer parameters.

KEYWORDS

Keyword Spotting, Noise robustness, Command Recognition, Attention Mechanism, Data augmentation.]

1. INTRODUCTION

Deep learning algorithms have entered a challenging new phase. In various cognitive tasks, including image classification [1] and speech recognition [2], the accuracy has surpassed humans. More and more artificial intelligent products appeared in our daily life, and people's productivity has been greatly improved. Among them, voice control further liberates people's hands and eyes, familiar voice products on the market include smart speakers such as Baidu Xiaodu, Xiaomi Xiaoai, Ali Tmall Genie, and voice assistants such as Apple Siri and Microsoft Cortana [3]. The devices wait for the wake-up word and activates the voice interaction program accordingly when the wake-up word occurs.

The release of the Google Speech Commands dataset provides a common basis for KWS system evaluation and it is allowed for scientific research [4]. With the publishing of the dataset, Warden also provided a baseline model based on the convolution architecture [5], with an accuracy rate of 85.4%. Hidden Markov Models (HMM), a traditional keyword recognition method, which model keywords and backgrounds together [6][7][8]. Recently, more significant architectures are applied to KWS tasks. Deep neural networks (DNN), starting with the fully-connected networks, have been shown to perform efficiently [9]. A major disadvantage of DNN-based KWS algorithms is that they cannot effectively model local temporal and spectral correlations in speech features. CNN overcomes this shortcoming by treating the input time-domain and spectral-domain features as images and perform two-dimensional convolution operations on them [10][11]. Recent studies have shown that RNN-based keyword recognition using LSTM cells can exploit longer temporal contexts with gating mechanisms and internal states [12][13][14].

Most of current KWS researches are based on clean data, and ignore the diversity of usage scenarios. We directly test their model on a noisy test set and found that the performance was not satisfactory. At the same time, it also needs to meet the requirements of low memory and low power consumption. We summarize our contribution as follows:

- We train with a large amount of noisy data to simulate a wide variety of noisy environments, to improve the noise robustness of the model.
- To further improve the accuracy of the model, inspired by natural language process, we propose an architecture, combining attention module with residual mechanism, and we use depth separable convolution replace the ordinary convolution to reduce the resource occupation.
- We also apply attention mechanism to RNN and CNN backbones with different parameter size to compare the performance.

2. Related Work

2.1. Keyword Spotting

Keyword spotting is used to detect the preset words from a continuous speech, and is usually deployed in devices that require low power consumption and small memory, such as smart speakers and smart watches. Considering about the function, in addition to detecting target words, the classifier also needs to distinguish between silence and words or sounds that are not in the target list. The pipeline is as Figure 1.

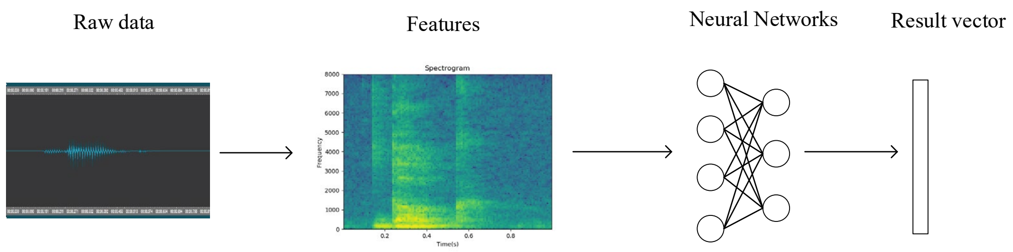


Figure 1. Keyword spotting overview

Zhang et al. compared and analyzed several network structures [15] for quiet speech command recognition. Among them, DNNs consist of a stack of fully connected layers and nonlinear activation layers. Each fully connected layer is followed by a ReLU activation functions. And then output is probabilities of the k keywords generated by a linear layer and a softmax layer. DNN has less operations/inference and hence suit well to systems that have limited computing capability, but the accuracy is not very impressive. RNN has shown excellent performance in many sequence tasks, especially speech recognition, language modeling and translation [16], etc. RNNs not only exploit the temporal relationship between input signals, but also utilize a “gating” mechanism to capture long-term dependencies. RNN cells can be LSTM cells or Gated Recurrent Unit (GRU) cells [17][18]. Since weights are reused across all T time steps, RNNs tend to have a smaller number of parameters compared to DNNs. However, since RNNs are time-series dependent, it cannot be computed in parallel. CNNs treat the input temporal and spectral domain features as images and performs two-dimensional convolution operations on them. Convolutional layers are usually followed by batch normalization [19], ReLU activation functions, and sometimes max or average pooling layers for the purpose of reducing the dimensionality of features. During inference, the batch-normalized parameters can be collapsed into the weights of the convolutional layers. To reduce parameters and speed up training, a linear low-rank layer is added between the convolution and dense layers. For better performance, CRNN, combination of CNN and RNN, exploits local temporal/spatial correlation using convolution layers and obtain global temporal dependencies in the speech features with recurrent layers [20]. Most of the later studies are based on the above-mentioned architectures.

2.2. Attention mechanism

The target of the attention mechanism is to selectively filter a small amount of important information from a large amount of information, focus on key information with limited attention. It ignores most unimportant information to save resources, and obtains the most effective information efficiently and quickly [21]. Attention was firstly introduced to deal with the problem that the frame generated by encoder-decoder is too long and information lost in the seq2seq task [22]. As shown in Figure 2, the results obtained from cells are concatenated and fed to the attention layer. In simple terms, attention gives weights to words according to their contribution.

Since the encoder encode input data of any length into a fixed-length vector, when inputting a very long data, it will cause information loss. Attention solves this problem by encoding the input data of any length into a vector sequence instead of a vector. The attention module can not only be applied to the encoder-decoder model, but is a general idea and does not depend on a specific framework. In the field of speech recognition, the main contribution of the attention mechanism is to align the output characters with the input speech signal.

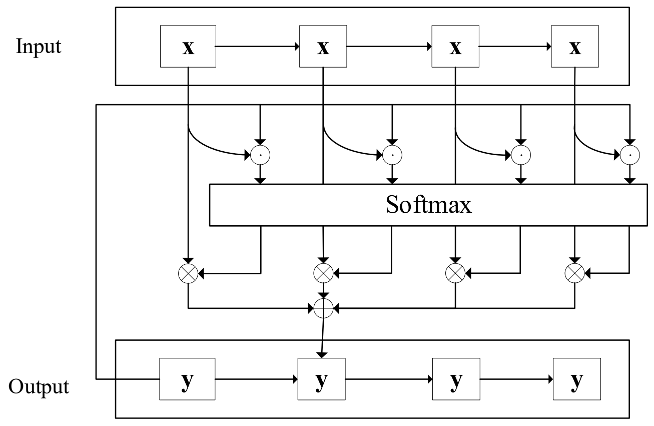


Figure 2. Attention mechanism in encoder-decoder architecture

3. METHOD

Based on CNN, we propose an architecture that incorporates attention mechanism and residual mechanism on CNN backbone.

3.1. Backbone

Most of machine learning is shallow model (such as GMM, HMM)[23][24][25], which means weak nonlinear transformation ability, so it is not enough to describe the complex high-dimensional features of speech. Since the input of GMM is a single frame, the influence of co-pronunciation is ignored, so we use the spliced frame as the input of the neural network to model the observation probability. Both traditional HMM and RNN can model time series. Since HMM is a shallow model and RNN has the problem of gradient disappearance, a long short-term memory network (Long short-term memory, LSTM) is used to model time series [26]. Long short-term memory, a special kind of RNN, mainly used to solve the problem of gradient disappearance and gradient explosion when input a long sequence. Simply put, compared to ordinary RNNs, LSTM can perform better in longer sequences.

Under normal circumstances, speech recognition is based on the speech spectrum after time-frequency analysis, and the speech time-frequency spectrum has structural characteristics. With regard to improve the speech recognition performance, it is necessary to overcome the various diversity faced by the speech signal, including the diversity of the speakers and the diversity of the environment, and so on. A convolutional neural network carries out translation-invariant convolution in time and space respectively. Thanks to the idea of applying convolutional neural network to the acoustic model of speech recognition, the invariance of convolution can be used to surmount the diversity of the speech signal. From this perspective, it can be considered that the time-frequency spectrum obtained by the analysis of the entire speech signal is treated as an image, and the deep convolutional network widely used in the image is introduced to recognize it. The earliest widely used neural network DNN is a standard feedforward neural network composed of some fully connected layers and nonlinear activation layers. The input of DNN is a flattened feature matrix, and the network structure contains a stack of d hidden layers, and each layer has n neurons. One of the main shortcomings of the DNN-based KWS algorithm is that it cannot effectively simulate the local time and spectral correlations in speech features. CNN makes use of this correlation of the time domain and spectral domain to process the features into an image, and perform a two-dimensional convolution operation on it. The convolutional layer is usually followed by batch normalization, ReLU activation functions and optional max/average pooling layers [27], which reduce the dimensionality of features. In the inference process, the batch normalized parameters can be folded into the weight of the convolutional layer. In some cases, in order to reduce parameters and speed up training, a linear low-rank layer is added between the convolutional layer and the dense layer. It is just a fully connected layer without nonlinear activation. We use LSTM and CNN as the backbone networks to study speech keyword recognition in noisy environments.

3.2. Attention Mechanism

The attention mechanism identifies key features in the data by using the weights of the new attention layer. Attention module includes spatial attention and channel attention. The spatial attention shows attention to different positions on the feature map, similarly, the channel attention shows attention to different data channels. For speech data, that means, the attention to time domain and frequency domain respectively.

Channel attention module passes the input feature map through global max pooling and global average pooling based on width and height respectively, and then Multilayer Perceptron (MLP). The outputs of the MLP are concatenated based on element-wise, and then the sigmoid activation operation is performed to generate the final channel attention feature map. Channel attention module makes use of the elementwise multiplication of the channel attention feature map and the input feature map to generate the input features required by the spatial attention module. The feature map is compressed in the spatial dimension to obtain a one-dimensional vector for next operate. When compressing in the spatial dimension, average pooling and max pooling are both considered. Average pooling and max pooling can be used to aggregate spatial information of feature maps, which will be fed to a shared network to compress the spatial dimension of input feature maps. Finally, element-wise summation is performed to produce a channel attention map.

The spatial attention module uses the feature map generated by channel attention module. The algorithm firstly carries out a channel-based global max pooling and global average pooling, and then concatenates the results based on the channel. Next, it reduces the dimension to 1 by a convolution operation. By using a sigmoid activation function, the spatial attention feature is generated. Finally, we obtain the final generated feature by multiplying the input feature of the module to the spatial attention feature. The spatial attention mechanism compresses the channel, and performs average pooling and max pooling respectively in the channel dimension. The operation of max pooling is to extract the maximum value on the channel, and the number is the result of multiplying the height by the width; the operation of average pooling is to extract the average value on the channel, and the number is the same as the previous one. Finally, the feature maps of 1-channel are merged and obtain a 2-channel feature map.

The spatial attention ignores the information in the channel domain, and treats the image features in each channel equally. This limits the spatial transformation method only be used in the original image feature extraction stage, and it is not very interpretable if it is applied to other layers of the neural network. Similarly, the channel attention directly uses global average pooling in a channel, ignoring the local information in each channel.

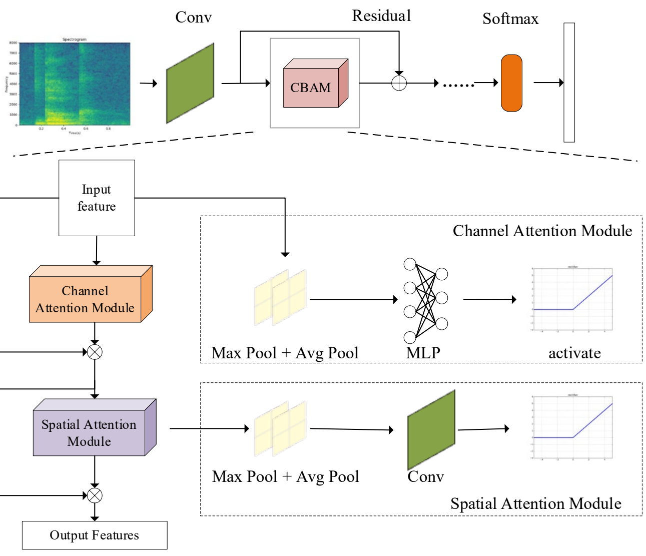


Figure 3. Model architecture with attention mechanism

For the above two kinds of attention, the spatial attention ignores the information in the channel domain, and treats the image features in each channel equally. This limits the spatial transformation method to the original image feature extraction stage, and it is not very interpretable if it is applied to other layers of the neural network. Similarly, the channel attention directly uses global average pooling in a channel, ignoring the local information in each channel. The model architecture with attention mechanism is in Figure 3. Combining channel attention and spatial attention can not only save parameters and computing power, but also perform better. we combine above two attention mechanism and residual module. Residual structure was firstly proposed by He et al. to eliminate the problem of difficulty in training and accuracy degradation with excessive depth networks [31]. The proposal of residual network is a milestone event in the history of CNN images. At ILSVRC and COCO 2015, Deep residual network (ResNet) achieved 5 firsts, and once again refreshed the history of the CNN model on ImageNet. So far, it has become a classic structure in the field of computer vision. The residual mechanism concatenates past feature and transformed features to maintain information, considering the similarity between speech keyword recognition and image recognition, we applied the residual method to our attention module, we concatenate the input features and output feature of the attention module as the input for the next operation. Therefore, not only the mask is added according to the information of the current network layer, but also the information of the previous layer is passed down. This can effectively prevent the problem of information loss due to the deepening of network layers.

3.3. Depthwise Separable Convolution

For the purpose of further reducing the power consumption of the neural network and adapt to embedded wearable devices, separable convolution is used instead of ordinary convolution, mainly using depthwise separable convolution (DSC), which is the first to appear in a doctoral dissertation called "Rigid-motion scattering for image classification" [28]. DSC is composed of Depthwise Convolution and Pointwise Convolution. Each channel of the Depthwise Convolution input feature map uses a convolution kernel, and then the outputs of all convolution kernels are spliced to get its final output, as shown in Figure 4. Because the quantity of output channels of the convolution is equal to the number of convolution kernels, and only one convolution kernel is used for each channel in Depthwise Separable Convolution, the quantity of output channels of a single channel after the convolution operation is also 1. Then if the channels' number of the input feature map is N, a single convolution kernel is used for each of the N channels to obtain N feature maps with 1 channel. Then, the N feature maps are spliced in order to obtain an output feature map with channel N. This operation enormously cuts down the number of parameters of the convolution kernel. The depthwise separable convolution module is shown in Figure 4.

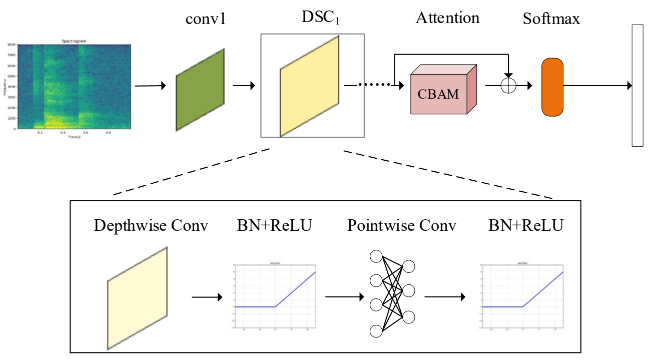


Figure 4. Depthwise separable convolution

4. Experiment

The audio signal is pre-processed into a discrete matrix vector by the acoustic feature-extract module, which is used as the input feature of our keyword detection model. We extract 40-dimensional feature vectors within a 25-milliseconds frame, with a frame offset of 10 milliseconds. The neural network model is used to process the input MFCC features. After the classification model, a fixed-length vector, which has the same size with the categories, is generated, and the output value is exactly the probability of the data belonging to each category. The pipeline of the entire model is in the Figure 1.

4.1. Datasets and Input Pre-processing

We used authoritative Google Speech Commands dataset as the benchmark dataset. Google Speech Commands dataset version 1 has 65K utterances from different speakers, each of which is 1 second long. Each utterance belongs to one of 30 categories, including common words such as "Go", "Stop", "Left", and "Down". The dataset's sampling rate is 16kHz. In our experiment, we considered to recognize 10 words and additional tags "unknown" and "silent". Besides, we also try to recognize all 35 words. For the both tasks and each architecture, we measured the top-1 classification accuracy.

We use feature extraction module to obtain MFCC features, which will be fed to the neural network [29]. In addition to the traditional signal processing steps such as pre-enhancement, framing, windowing, Fourier transformation, and Mel filtering, some other data enhancement methods are introduced, including randomly shifting the voice signal on the time axis, randomly enhancing the power spectrum, and appropriately doing some time masking and frequency masking [30]. Time domain masking means randomly selecting 0-50 consecutive frames, and set all their mel-filter groups value to 0, and the frequency domain mask similarly is to randomly select 1 to 30 continuous dimensions from 40 mel-filter groups, and set their values to 0, as shown in Figure 5. For each mini-batch, 1/3 only has time domain masking, 1/3 only has frequency domain masking, and the others have all masking. These data augmentation means can enrich the diversity of data effectively.

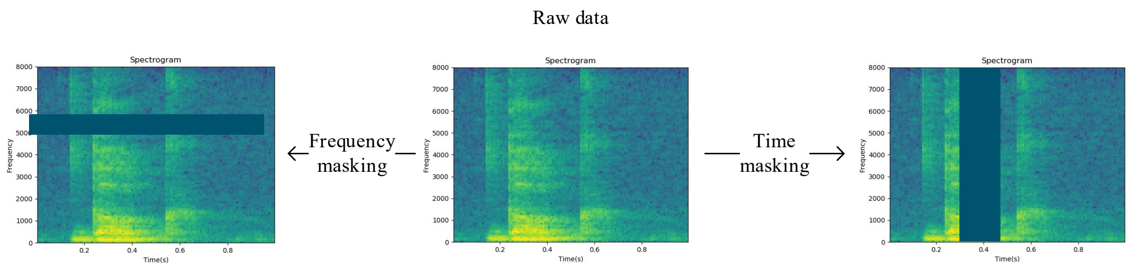


Figure 5. Data augmentation

In addition to the data augmentation measures described above, a variety of noisy data in various scenes with different signal-to-noise ratio is added to the training data to enhance the performance of the model in a noisy environment. Noise comes from scenes including kitchen, living, washing, field, park, river, hallway, meeting, office, resto, station, squares, traffic, car and metro and so on. Each category contains 16 pieces of noise data, and the length of each noise wav is 300s [31]. The format of noise data is also mono 16KHz. By detecting recognition accuracy on noisy data, the results proved that the noise mixed training method has greatly advantages.

4.2. Model Training

We used the tensorflow framework to train the model. The hyperparameters used are shown in the Table 1.For Data processing, we set the time window length as 25ms, and time window stride as 10ms. Extract MFCC features using 40 mel filters.

Table 1. Hyperparameters used in our experiment

Data processing	
Time window length	25ms
Time window stride	10ms
Regularization	
Wight decay	10^{-5}
Dropout	0

training	
Batch size	100
Weight initialization	Xavier
Optimizer	Adam
Training steps	3000
Schedule	exponential
Learning rate	0.001

For training, the mini batch size is 100. We use the Adam optimizer to train with an exponential decay of the learning rate for all architecture, and the learning rate decays by 0.1 after every 10,000 steps start from 0.001. The model ends training after two attenuations. The cross-entropy loss function is used to measure the loss between labels and predicted values. We also use dropout and regularization to prevent overfitting, and the L2 weight decay is 10-5.

4.3. Results

We add additional synthetic keyword samples for each keyword category, including clean and low SNR, based on 65K Google Speech Command Dataset utterances. The noise comes from airport, babble, car, exhibition, restaurant, street, subway and train. For each category, there are 900~1500 new utterances, both for clean and noisy. We shuffle the original datasets and our additional samples to generate training, validating and testing data randomly.

The increased number of samples is shown in Table 2. The "original" means sample size of Google, and the "total" is the amount of the original data plus the added data, including clean and noisy, and the "ratio of noise" means the ratio of the added noisy data to the total.

Table 2. Data size in our experiments

Original(K)	Total(K)	Ratio of noisy data(%)
65	147	27.9

We test the performance of depthwise separable convolution. Since this method replaces the convolution operation of the CNN model, we directly add the attention module based on the CNN model for comparison. The CNN model contains two convolutional layers, one linear layer and two fully connected layers. The structure is shown in Table 3.

Table 3. Architecture of CNN_attention

Input	Layer	c	f	s
B×49×10×1	Conv	28	(10,4)	(1,1)
B×40×7×28	Conv	30	(10,4)	(2,1)
B×16×4×30	flatten	-	-	-
B×1920	Linear	-	-	-
B×16	Fc	-	-	-
B×128	Fc	-	-	-

Most scholars like to test models in 12 categories with additional silence and non-crucial words when using Google Speech Command dataset. Therefore, we use LSTM and CNN as the backbone network to test the recognition accuracy of the model on the noisy test data for comparison. At the same time, in order to test the performance of the model on a larger data set, the experiment also involved all 35 classes of the dataset. The following Table 4 displays the results of the model for 12 classes.

Table 4. Noisy testing dataset accuracy of 12-classes.

Backbone	**V1**	**V2**	**V3**
LSTM	94.03%	94.58%	94.72%
CNN	94.48%	94.66%	94.93%

Among them, v1 means testing the original model in noisy test data, as a comparison, v2 mixed training data with noise based on v1, v3 introduces the attention module based on v2. The following Table 5 shows the performance of the model for 35 classes.

Table 5. Noisy testing dataset accuracy of 35-classes.

Backbone	**V1**	**V2**	**V3**
LSTM	92.60%	92.64%	93.12%
CNN	93.49%	93.56%	93.79%

V1~v3 are the same meaning as the above Table 4. It can be seen from the above two tables that the model performs well on both datasets. The method of mixing noise with training data promotes. And the attention module also performs well.

We take a small CNN as an example, Table 3 display the input for each layer, and "c" is channel for convolution, "f" represents the size of filter, s means the stride for x and y respectively. The total parameter size is about 69.9KB. We test three different scales of CNN to compare accuracy and memory footprint. The results are as Table 6.

Table 6. Performance of different convolution

Network	**Params(K)**	**Accuracy(%)**
CNN+attention	69.9	91.72
	179.7	92.08
	504.3	93.16
+DSC	24.4	94.48
	144.1	94.66
	498.3	94.93

Our model greatly reduces the amounts of parameters through separable convolution. We obtain better performance than normal CNN with much smaller parameter size. Separable convolution effectively reduces the number of parameters by decomposing convolution kernel and drops the correlation of convolution operations between channels, and has little impact on the calculation results.

5. CONCLUSIONS

This article introduces the application of LSTM and CNN in the speech keyword recognition system. Our works are achieved with the latest release of Google Speech Command dataset, which provides a general benchmark for this task. In past studies, most scholars published their results on clean test data. However, in fact we are surrounded by a variety of noisy environments. Our research shows that adding attention module to the model is conducive to the task of keyword spotting. At the same time, according to the noise test results, it is necessary to add additive noise with different signal-to-noise ratios in different scenarios to the training data for joint training.

We also considered the lightweight method of separable convolution. In theory, depthwise separable convolution requires less computation. However, since the computational intensity of deep convolution (the ratio of FLOPs to memory access) is too low, it is difficult to effectively use hardware, so it is difficult to effectively implement deep separable convolution in practice. Therefore, in the future, I will focus on effective ways to reduce the computing consumption of the model, meantime explore the model structure to further improve the recognition ability of the model.

REFERENCES

[1] Chen Y, Li J, Xiao H, et al. Dual path networks [J]. arXiv preprint arXiv:1707.01629, 2017.
[2] Xiong W, Wu L, Alleva F, et al. The Microsoft 2017 conversational speech recognition system[C]//2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018: 5934-5938.
[3] McGraw I, Prabhavalkar R, Alvarez R, et al. Personalized speech recognition on mobile devices[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5955-5959.
[4] Warden P. Speech commands: A dataset for limited-vocabulary speech recognition [J]. arXiv preprint arXiv:1804.03209, 2018.
[5] Chen G, Parada C, Heigold G. Small-footprint keyword spotting using deep neural networks[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 4087-4091.
[6] Richard C Rose and Douglas B Paul, "A hidden markov model based keyword recognition system" in Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on. IEEE, 1990, pp. 129–132.
[7] Rohlicek J R. Continuous HMM for Speaker Independent Word Spotthing[J]. Proc. ICASSP, May. 1994, 1994.
[8] Wilpon J G, Miller L G, Modi P. Improvements and applications for key word recognition using hidden Markov modeling techniques[C]//[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1991: 309-312.
[9] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in Acoustics, speech and signal processing (icassp), 2014 ieee international conference on. IEEE, 2014, pp. 4087–4091
[10] Sainath T, Parada C. Convolutional neural networks for small-footprint keyword spotting[J]. 2015.
[11] Chen X, Yin S, Song D, et al. Small-footprint keyword spotting with graph convolutional network[C]//2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019: 539-546.
[12] Wollmer M, Eyben F, Keshet J, et al. Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks[C]//2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009: 3949-3952.
[13] Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in Spoken Language Technology Workshop (SLT), 2016 IEEE.IEEE, 2016, pp. 474–480.
[14] Chai S, Zhang W Q, Lv C, et al. An End-to-End Model Based on Multiple Neural Networks with Data Augmentation for Keyword Spotting [J]. International Journal of Asian Language Processing, 2020, 30(02): 2050006.
[15] Zhang Y, Suda N, Lai L, et al. Hello edge: Keyword spotting on microcontrollers [J]. arXiv preprint arXiv:1711.07128, 2017.
[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

[17] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. Journal of machine learning research, 3(Aug):115–143, 2002.

[18] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures[C]//International conference on machine learning. PMLR, 2015: 2342-2350.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456, 2015.

[20] Sercan O Arik, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates. Convolutional recurrent neural networks for small-footprint keyword spotting. arXiv preprint arXiv:1703.05390, 2017.

[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[22] Shi B, Yang M, Wang X, et al. Aster: An attentional scene text recognizer with flexible rectification [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(9): 2035-2048.

[23] Liu Y, He L, Zhang W Q, et al. Investigation of Frame Alignments for GMM-based Digit-prompted Speaker Verification[C]//2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018: 1467-1472.

[24] Zimmermann M, Ghazi M M, Ekenel H K, et al. Visual speech recognition using PCA networks and LSTMs in a tandem GMM-HMM system[C]//Asian conference on computer vision. Springer, Cham, 2016: 264-276.

[25] Nankaku Y, Sumiya K, Yoshimura T, et al. Neural Sequence-to-Sequence Speech Synthesis Using a Hidden Semi-Markov Model Based Structured Attention Mechanism [J]. arXiv preprint arXiv:2108.13985, 2021.

[26] Wilkinson N, Niesler T. A Hybrid CNN-BiLSTM Voice Activity Detector[C]//ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 6803-6807.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778..

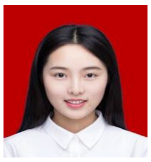
[28] Sifre L, Mallat P S. Rigid-Motion Scattering For Image Classification Author[J]. English. Supervisor: Prof. Stéphane Mallat. Ph. D. Thesis. Ecole Polytechnique, 2014.

[29] Huizen R R, Kurniati F T. Feature extraction with mel scale separation method on noise audio recordings[J]. arXiv preprint arXiv:2112.14930, 2021.

[30] Park D S, Chan W, Zhang Y, et al. Specaugment: A simple data augmentation method for automatic speech recognition[J]. arXiv preprint arXiv:1904.08779, 2019.

[31] Büchel J, Zendrikov D, Solinas S, et al. Supervised training of spiking neural networks for robust deployment on mixed-signal neuromorphic processors[J]. arXiv preprint arXiv:2102.06408, 2021.

AUTHORS



Peiwen Ye Received the B.S. degree in computer science (2015) from Southwest University. Now she is seeking a master's degree in computer science at the University of Electronic Science and Technology of China, her research direction is deep learning and speech recognition.

Hancong Duan received the B.S. degree in computer science from Southwest Jiaotong University in 1995, the M.E. degree in computer architecture in 2005, and the Ph.D. degree in computer system architecture from UESTC in 2007. Currently he is a professor of computer science at UESTC. His current research interests include Deep Learning, Large-Scale P2P Content Delivery Network, Distributed Storage and Cloud Computing.