AN EVALUATION OF SOME FACTORS AFFECTING ACCURACY OF THE VIETNAMESE KEYWORD SPOTTING SYSTEM

Nguyen Huu Binh, Nguyen Quoc Cuong, Tran Thi Anh Xuan*

Abstract: *Keyword spotting (KWS) is one of the important systems on speech applications, such as data mining, call routing, call center, customer-controlled smartphone, smart home systems with voice control, etc. With the goals of researching some factors affecting the Vietnamese Keyword spotting system, we study the combination architecture of CNN (Convolutional Neural Networks)-RNN (Recurrent Neural Networks) on both clean and noise environments with 2 distance speaker cases: 1m and 2m. The obtained results show that the noise trained models are better performance than clean trained models in any (clean or noise) testing environment. The results in this far-field experiment suggest to us how to choose the suitable distance of the recording microphones to the speaker so that there is no redundancy of data with the contexts considered to be the same.*

Keywords: Keyword spotting; Speech recognition; Far-field distance; Convolutional neural networks; Recurrent neural networks.

1. INTRODUCTION

In the field of speech processing, keyword identification or detection involves detecting some words or phrases from a continuous stream of audio.

Keyword recognition has many practical applications such as indexing and searching, routing telephone calls, voice command, etc. A famous application of the keyword recognition system today is "Google Voice Search" **[1]** - This application continuously monitors the appearance of the keyword "Ok Google" to initialize the continuous voice recognition system. The keyword detection system is also applied in personal digital assistant systems such as Alexa or Siri to "wake up" when the names of these systems are called by voice.

In Vietnam, there have been a few authors who have been researching the field of Vietnamese speech processing in general, but the studies on the Vietnamese keyword speech recognition system is very rare. So, the keyword speech recognition approach has great potential for development in the field of speech processing in the world in general and in Vietnam in particular. This is the reason that we focus on researching some factors affecting the Vietnamese keyword spotting system in this paper.

In recent years, many keyword recognition techniques have been studied. Traditional methods for KWS are based on Hidden Markov Models with sequence search algorithms **[2]**. With the advances in deep learning, some KWS models based on deep neural networks (DNNs) are studied **[3]**. But a potential drawback of DNNs is that they ignore the structure and context of the input in time or frequency domains. Another approach is using Convolutional Neural Networks (CNN) to exploit local structures and patterns on the input signal **[4]**. CNNs have very good performance with high-dimensional data that are invariant to translation **[5]**. However, CNNs have also a drawback is that they cannot model the context over the entire frame without wide filters or great depth **[6]**. Recurrent Neural Networks (RNNs) are also studied for KWS **[7-8]**, to model dependency over time. RNNs are well-suited to deal with sequential data because long sequences can be processed step-by-step with limited memory of previous sequence elements **[5]**. Therefore, with some complementary advantages, it is possible to combine CNN and RNN for KWS,

as done in, by exploiting convolutional layers as feature extractors and by using the output for training an RNN **[6**, **9]**. Inheriting these previous research results, in this paper, we focus on developing a KWS system using the combination architecture of CNN and RNN and applying for Vietnamese far-field keyword spotting in a noise environment, namely at 1m and 2m distance. In section 2, we describe CNN-RNN architecture. In section 3, we present the experiments and the corresponding results, to show the effect of noise and 1m/2m distance to the performance of the Vietnamese keyword spotting system. And from there, some conclusions will be given in section 4.

2. CNN-RNN KEYWORD SPOTTING SYSTEM

2.1. CNN-RNN (CRNN) Architecture

In practical, CRNN model is used in an English keyword spotting system in **[6]** and their experiment results showed that CRNN is one of effective method in KWS system recently. This is a reason for us to choose CRNN is the model in our researching some factors affecting Vietnamese keyword spotting system. The end-to-end CRNN architecture of the KWS system is presented in figure 1.

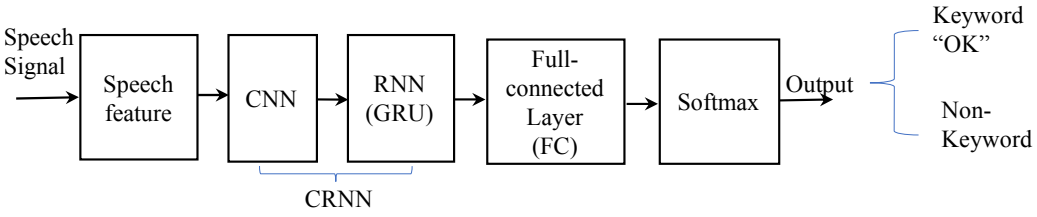


Figure 1. *A common Convolution recurrent neural networks (CRNN) architecture.*

The end to end process includes as follows: the raw time-domain inputs are converted to Mel frequency cepstrum coefficients, and then these 2-D MFCC features are given as inputs to the convolutional layer, in which 2-D on both time and frequency dimensions. The outputs of the convolutional neural network (CNN) are fed to recurrent neural networks (specifically, gated recurrent units (GRUs)). This process is implemented in the entire frame. Outputs of the recurrent layers are given to the fully connected (FC) layer. Lastly, softmax decoding is applied over two neurons, to obtain a corresponding scalar score. The detailed content of CNN and RNN will be presented in sections 2.2 and 2.3, respectively.

2.2. Convolutional Neural Network (CNN)

2.2.1. N-D discrete convolution of two matrix

For discrete, N-dimensional variables A and B, the following equation defines the convolution C of A and B:

$$\mathbf{C} = A * B \tag{1}$$

So, each component of matrix C is equal:

$$C(j_1, j_2, ..., j_N) = \sum_{k_1} \sum_{k_2} ... \sum_{k_N} A(k_1, k_2, ..., k_N).B(j_1 - k_1, j_2 - k_2, ..., j_N - k_N) \tag{2}$$

in which, each k_i runs overall values that lead to legal subscripts of A and B.

2.2.2. CNN architecture

As **[4]**, a typical CNN architecture is shown in figure 2.

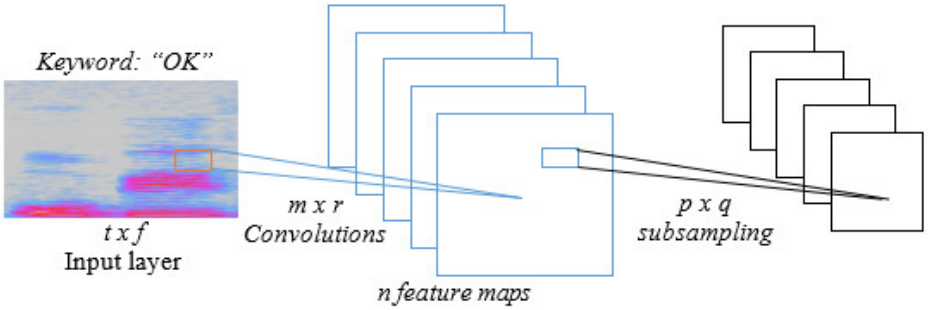


Figure 2. *A typical diagram of the convolutional neural network architecture* ***[4]****.*

In this architecture, the dimension of an input signal is $V \in R^{t x f}$, in which, t and f are the input feature dimension in time and frequency, respectively.

A weight matrix $W \in R^{(m x r) x n}$ is convolved with the full V, with a small local time-frequency patch of size *(m x r)*, where $m \leq t$ and $r \leq f$, and feature maps numbers n. The filter can stride by a non-zero amount of s in time and v in frequency. So, overall the convolutional operation produces n feature maps of size $\left(\frac{t - m + 1}{s} \times \frac{f - r + 1}{v}\right)$.

After performing convolution, these n feature maps are passed to a max-pooling layer, to remove variability in the time-frequency space that due to speaking style, channel distortions,... Assumedly, given a pooling size of $p \ x \ q$ and no-overlapping pooling, so pooling performs a sub-sampling operation to reduce the time-frequency space with the size of $\left(\frac{t - m + 1}{s.p} \times \frac{f - r + 1}{v.q}\right)$.

2.3. Recurrent Neural Networks (RNN): *Gated Recurrent Neural Networks*

In traditional, the feed-forward neural network consists of three main parts are the input layer, the hidden layer, and the output layer, in which: the first hidden layer is a full-connected layer with the input, second layer fully-connected with the first layer..., and then an output comes out of the last layer. The input and output of this neural network system are independent of each other. Thus, this model is not suitable for sequence problems, such as sentence completion because the next predictions (such as the next word) depends on its position in the sentence and word before it. And RNN was born with the main idea of using memory to store information the previous computations and then based on it can make the most accurate predictions for the current prediction step.

However, it has been firstly by Sepp (Joseph) Hochreiter (1991), and then also observed by Bengio et al. (1994) that is it difficult to train RNNs to capture long-term dependencies because the gradients tend to vanish or explode gradient.

This disadvantage of RNN is due to this architecture has no mechanism to filter unnecessary information. And GRU model was proposed by Cho et al. (2014) to overcome the disadvantages of RNN.

Introduced by Cho et al. in 2014 **[11]**, Gated Recurrent Unit (GRU) was proposed to solve the vanishing gradient problem which comes with a standard recurrent neural network.

GRU is a variation on Long Short-Term Memory (LSTM) recurrent neural networks. Both LSTM and GRU networks have additional parameters that control when and how their memory is updated.

And both GRU and LSTM networks can capture both long and short term dependencies in sequences, but GRU networks involve fewer parameters and so are faster to train.

GRU is a novel model type of RNN that proposed a new type of hidden unit. Figure 3 shows the graphical description of the proposed hidden unit.

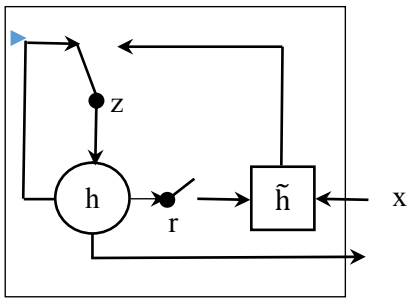


Figure 3. *Illustration of a gated recurrent unit:* z *and* r *are the reset and update gates;* h *and* h̃ *are the actual and candidate activations.*

The actual activation h_t^j of the *j*-th element of a hidden unit vector at time *t* is computed by:

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{3}$$

where z_t^j is an update gate that decides how much information from the previous hidden state will carry over to the current hidden unit. This helps the RNN to remember long-term information. The update gate z_t^j is computed as follows:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \tag{4}$$

where $(.)^j$ denote the *j*-th element of a vector.

The candidate activation $\tilde{h}_t^j$ in Eq[3], is computed by:

$$\tilde{h}_t^j = \phi(W x_t + U(r_t \odot h_{t-1}))^j \tag{5}$$

where r_t^j is a reset gate and $\odot$ is an element-wise multiplication. When the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only.

Summarily, GRUs using the internal memory capacity is valuable to store and filter the information using their update and reset gates.

3. EXPERIMENTS AND RESULTS

3.1. Dataset

We develop our KWS system for the keyword "OK". The reason we choose wake-up-word "OK" is that this is a popular word the people use in the world, and "OK" is the first word of wake-up word of the famous KWS system – Google Assistant. A special thing here is the word "OK" is read in Vietnamese phonetic transcription /*o ke*/, not in English phonetic transcription. So, this is perfectly suited to the Vietnamese keyword spotting system.

The entire data set consists of ~ 30.2 hours of the speech signal, including both non-

keyword and keyword. All are mono recordings with a sample rate of 16kHz and a bit resolution of 16 bits in a fairly clean environment at two distance values: 1m and 2m far from speakers. We asked native speakers of Vietnamese to read prompted sentences (which contained non-keyword or keyword) at a time.

Each person reads in a completely different scenario, including 5 sentences containing the keyword "OK" and 19 meaningful sentences without the keyword that are quoted from newspapers or paragraphs (containing approximately 30 words per this sentence). This ensures that no one reads the same script, so the context of the built dataset using in this paper is very diverse. The total number of words in the entire recording scenario is 2033 words.

Each sentence of each recording person is recorded simultaneously from 2 mono microphones: 1 microphone is 1m away from the speaker, and the remaining one is 2m away from the speaker.

The corpus consists of speech data spoken by 80 speakers, from the Northern and Southern of Vietnam, including 40 females and 40 males. Each keyword sentence is recorded 5 times at one distance value per person. Each non-keyword sentence is recorded 1 time at one distance value per person. There is 2 distance value in our recording: 1m and 2m. There are a total of 800 sentences containing the keyword and 3040 sentences containing the non-keyword.

The dataset is split into cross-validation of training, development and testing sets with a 6-2-2 ratio. The results show in section 3.4 to 3.6, is the average values of each experiment. This dataset used to design the baseline KWS model.

To build the noise KWS model, this dataset is augmented by applying Additive White Gaussian noise, with a power determined by a signal-to-noise (SNR) sampled from [-5,10] dB interval. In this task, each clean speech file is added to a random noise file at each SNR ratio.

3.2. Feature extraction, label generation, and training

The feature extraction module is common to both systems: the noise-KWS system and the clean-KWS system.

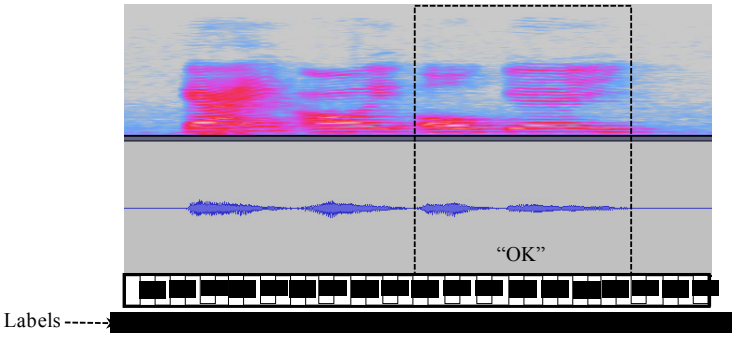


Figure 4. *An example of label generation in a speech signal input including "OK".*

In our paper, we generate acoustic features based on 13 Mel-Frequency Cepstral Coefficients (MFCC) and their 26 derivative ones, including 13 deltas and 13 delta-delta, computed every 10ms over a window of 25ms. For both two models, we use 16 frames for the input window of the CNN network, including 15 frames in the past and 1 frame in the

current time.

For label generation, we generate input sequences composed of pairs $<X_t, c>$, where X is a 1D tensor corresponding to MFCCs, and c is the class label (one of {0,1}). We assign labels of 1 to all sequence entries, part of a true keyword utterance "OK", and other entries are assigned a label of 0. More details for this labeling is illustrated in figure 4.

We use 32 convolutional filters CNN, and 2 recurrent layers – GRU, the output of the convolutional layer are fed to Gated recurrent units. We use the ADAM optimization algorithm for training.

3.3. Metrics

Three metrics are used to evaluate the performance of Vietnamese far-field keyword spotting systems because the non-keyword amount is more than the keyword ones: Precision, Recall, and F1-score.

3.4. Baseline KWS model

Baseline KWS model is built by a training model on the clean database as described in section 3.1.

Using the clean model, the precision, recall, and F1-score values are 99.2%, 100%, 0.996 respectively. Those results are high. However, the clean environment is an ideal case of the real environment. To use the KWS system on real applications, we need to consider the effect of noise on KWS performance. This will be presented in section 3.5.

Table 1. *The results of KWS system using the clean model on the clean testing set.*

	Precision (%)	Recall (%)	F1-score
Clean testing set	99.2	100	0.996

3.5. Noise KWS model setup

Some notations: Model_*k*dB is the trained model on the corpus with SNR of *k*dB (*k* is one of (-5, 0, 5, 10)).

Scenario 1: Using the clean model, the results on noise testing set with 4 SNR ratio (10dB, 5dB, 0dB, -5dB) are very low. The clean model is ineffective in the noise environments, and in the lower SNR environments especially.

Table 2. *The results of KWS using the clean model on some cases of noise testing sets.*

Model_Clean	Precision (%)	Recall (%)	F1-score
10dB noise testing set	58.48	22.19	0.29
5dB noise testing set	17.19	2.51	0.04
0dB noise testing set	0	0	0
-5dB noise testing set	0	0	0

The results from table 1, 2 show that the clean model, although it works well in a clean environment, shows a very ineffective performance in noise environments, especially in the noise environment with lower SNR ratio. This comment is obtained from the results of the clean model in 0dB or -5dB SNR in table 2.

Scenario 2: Using the different trained noise models that are called Model_*k*dB (in which k = 10; 5; 0; -5): we test on some cases of noise testing set, respectively. The results are shown in tables 3, 4, 5 and 6.

The results of the KWS system using Model_*k*dB in the scenario 2 show that if we train model in a specific environment, the best result is obtained from the testing set in the same environment by the highest F1-score: for example, using model_*k*dB, the best performance is obtained from the *k*dB noise testing set.

And when we training model at a certain SNR ratio, we receive better results in environments with higher SNR, and poorer results in environments with lower SNR.

Table 3. *The results of KWS using Model_10dB on some cases of noise testing sets.*

Using Model_10dB	Precision(%)	Recall (%)	F1-score
10dB noise testing set	98.95	99.03	0.99
5dB noise testing set	99.33	97.82	0.984
0dB noise testing set	99.71	91.25	0.944
-5dB noise testing set	92.5	54.97	0.656

Table 4. *The results of KWS using Model_5dB on some cases of noise testing sets.*

Using Model_5dB	Precision (%)	Recall (%)	F1-score
10dB noise testing set	98.31	98.57	0.983
5dB noise testing set	98.94	98.07	0.984
0dB noise testing set	98.36	87.07	0.911
-5dB noise testing set	89.51	41.61	0.538

Table 5. *The results of KWS using Model_0dB on some cases of noise testing sets.*

Using Model_0dB	Precision (%)	Recall (%)	F1-score
10dB noise testing set	95.48	99.17	0.971
5dB noise testing set	97.32	98.59	0.979
0dB noise testing set	98.96	97.22	0.98
-5dB noise testing set	99.46	80.71	0.869

Table 6. *The results of KWS using Model_-5dB on some cases of noise testing sets.*

Using Model_-5dB	Precision (%)	Recall (%)	F1-score
10dB noise testing set	78.34	98.68	0.861
5dB noise testing set	84.26	99.16	0.902
0dB noise testing set	91.45	99.37	0.947
-5dB noise testing set	96.24	98.59	0.971

3.6. Far-field experiments

In the building dataset in the far-field problem, an example in smart home KWS application, because the number of recording microphones are limited, so it is important to find the appropriate distance position from the recording microphone to the speaker: if the distance among these microphones is close to each other, then it will result in redundant data, but if the distance among these microphones is too far away, it may lead to lack context for training data.

To consider the effect of distance to the quality of our Vietnamese KWS system, at each test in section 3.6 we kept the same recording environment conditions for each test, the only difference here among the training models is that each model is derived from only recording data at one fixed distance position: either 1m or 2m to the speaker.

In our experiment, because we only have two recording microphones, so we put 1 microphone at 1m away from the speaker, and the remaining microphone at 2m away from the speaker. Is the distance between two recording microphones about 1m needed? Or should it be further than 1m? These experiments in section 3.6 will help the suggestion for the answer to this question.

In this section, we performed two scenarios as followings:

Scenario 1: with balance training corpus between at 1m and 2m distance, we use the Model_*k*dB obtained from section 3.5 and test in the same noise environment: *k*dB noise testing set, to observe the effect of microphone distances to the speaker.

Results on 1m and 2m are shown in table 7. We see that in the same condition of training and testing environment, the difference among the performance of our Vietnamese KWS system is not significant in the far-field distance at 1m and 2m if we build evenly both 1m distance and 2m distance case in the training corpus.

Table 7. *Comparison results in 1m and 2m of KWS system using Model_kdB on kdB noise testing sets.*

	Precision (%)		Recall (%)		F1-score	
	1m	2m	1m	2m	1m	2m
Model_clean on Clean testing set	99.47	98.96	100	100	0.994	0.989
Model_10dB on 10dB noise testing set	98.96	98.96	100	98.75	0.994	0.987
Model_5dB on 5dB noise testing set	98.95	98.44	100	98.75	0.994	0.985
Model_0dB on 0dB noise testing set	99.48	98.44	98.13	98.75	0.987	0.985
Model_-5dB on -5dB noise testing set	98.07	96.18	99.37	98.75	0.982	0.975

Scenario 2: with unbalance training corpus between at 1m and 2m distance: the model is obtained from the only training data that is recorded at 1m distance, and then test this model on the data that is recorded at 1m and then 2m distance; then inversely, the model is

obtained from the only training data that is recorded at 2m distance, and then test this model on the data that is recorded at 1m and then 2m distance. These experiments are performed at 2 representative cases: one with a Clean environment and the other one is presented for much more noise – that is in the environment with SNR ratio = -5dB. The results are presented in table 8, 9.

In table 8, the model that is trained with only the recording data at 1m distance from the speaker at the clean environment is called Model_Clean_1m on Clean testing set and the one is trained with only the recording data at 2m distance from the speaker at the clean environment is called Model_Clean_2m on Clean testing set.

In table 9, the model that is trained with only the recording data at 1m distance from the speaker at the noise environment with SNR ratio = -5dB is called Model_-5dB_1m on -5dB testing set and the one is trained with only the recording data at 2m distance from the speaker at the noise environment with SNR ratio = -5dB is called Model_-5dB_2m on -5dB testing set.

So, we have all four models in this scenario. Each model is tested with the recording data at 1m and the recording data at 2m, respectively. And these testing data are the same recording environment conditions with the training data. The results in 4 cases in tables 8 and 9 show that if using the same model, the difference in the quality of our Vietnamese keyword spotting system at 1m and 2m distance is not significant. This result initially gives us an idea about how to choose the distance between the microphones to the speaker - may be the distance between the microphone placed next to each other should be greater than 1m - in the building database collection problem for far-field KWS systems that have limited recording microphone equipment. This also can help reduce the amount of redundant data that is considered the same context, thereby helping the training model will be faster, but the quality is not affected a lot. Of course, to confirm this problem, we will continue to do more experiments with many other recording distances in future work.

Table 8. *Comparison testing results in 1m and 2m distance at the clean environment of Vietnamese KWS system using Model_Clean_1m/or 2m (the obtained model from only the recording data at 1m/or 2m distance in the clean environment).*

	Precision (%)		Recall (%)		F1-score	
	1m	2m	1m	2m	1m	2m
Model_Clean_1m on Clean testing set	98.06	97.92	100	100	0.989	0.988
Model_Clean_2m on Clean testing set	97.77	99.48	100	100	0.987	0.997

Table 9. *Comparison testing results in 1m and 2m distance at SNR ratio = -5dB of Vietnamese KWS system using Model_-5dB_1m/or 2m (the obtained model from only the recording data at 1m/or 2m distance in the noise environment at SNR ratio = -5dB).*

	Precision (%)		Recall (%)		F1-score	
	1m	2m	1m	2m	1m	2m
Model_-5dB_1m on -5dB testing set	96.92	96.25	99.37	98.13	0.980	0.970
Model_-5dB_2m on -5dB testing set	98.43	98.96	98.75	98.37	0.984	0.990

4. CONCLUSIONS

In this paper, we presented an approach based on the combination of CNN and RNN for the Vietnamese far-field keyword spotting in the noise environment. The obtained results show that the noise trained models outperform the clean trained model (the baseline system) in any environment (clean or noise from SNR -5dB to 10dB). In the building speech database in the far-field KWS system with the limited number of microphones, to avoid data redundancy in similar contexts, and lack of data in non-similar contexts, the distance between the microphone placed next to each other may be greater than 1m. In future work, more experiments need to be proposed with pre-processing to robust with different noise environments. And of course, more experiments in far-field at some distance positions among microphones will be performed so that we can confirm the suitable distance between the microphone placed next to each other in the Vietnamese far-field keyword spotting system applications, for example in the smart home application.

Acknowledgment: *This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2018-PC-064.*

REFERENCES

[1]. J. Schalkwyk et al., *"Google Search by Voice: A Case Study,"* Google, Inc, 1600 Amphitheater Pkwy Mountain View, CA 94043, USA.

[2]. R. C. Rose and D. B. Paul, *"A hidden Markov model-based keyword recognition system,"* in International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, USA, 1990, pp. 129–132, DOI: 10.1109/ICASSP.1990.115555.

[3]. G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, *"Model Compression Applied to Small-Footprint Keyword Spotting,"* presented at the Interspeech 2016, 2016, pp. 1878–1882, DOI: 10.21437/Interspeech.2016-1393.

[4]. T. N. Sainath and C. Parada, *"Convolutional Neural Networks for Small-footprint Keyword Spotting,"* in Proceedings of Interspeech 2015, pp. 1478–1482.

[5]. F. Colangelo, F. Battisti, A. Neri, and M. Carli, *"Convolutional recurrent neural networks for audio event classification,"* detection and Classification of Acoustic Scenes and Events 2018.

[6]. S. Ö. Arık et al., *"Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting,"* in Interspeech 2017, 2017, pp. 1606–1610, DOI: 10.21437/Interspeech.2017-1737.

[7]. K. Hwang, M. Lee, and W. Sung, *"Online Keyword Spotting with a Character-Level Recurrent Neural Network,"* arXiv:1512.08903, 2015.

[8]. S. Fernandez, A. Graves, and J. Schmidhuber1, *"An Application of Recurrent Neural Networks to Discriminative Keyword Spotting,"* in Artificial Neural Networks, Springer, pp. 220–229, 2007.

[9]. C. Lengerich and A. Hannun, *"An end-to-end architecture for keyword spotting and voice activity detection,"* arXiv:1611.09405.

[10]. K. Choi, G. Fazekas, M. Sandler, and K. Cho, *"Convolutional recurrent neural networks for music classification,"* in 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 2392–2396, DOI: 10.1109/ICASSP.2017.7952585.

[11]. K. Cho et al., *"Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,"* arXiv:1406.1078, 2014.

TÓM TẮT

ĐÁNH GIÁ MỘT SỐ YẾU TỐ ẢNH HƯỞNG ĐẾN ĐỘ CHÍNH XÁC CỦA HỆ THỐNG NHẬN DẠNG TỪ KHOÁ TIẾNG VIỆT

Ngày nay, hệ thống nhận dạng từ khóa (KWS) đóng vai trò quan trọng trong các ứng dụng sử dụng tiếng nói như trong các hệ thống khai thác dữ liệu, định tuyến cuộc gọi, tổng đài chăm sóc khách hàng, điện thoại thông minh hay trong hệ thống nhà thông minh điều khiển bằng giọng nói… Với mục tiêu nghiên cứu một số yếu tố ảnh hưởng đến chất lượng của hệ thống nhận dạng từ khóa tiếng Việt, chúng tôi đã xây dựng các mô hình hệ thống sử dụng sự kết hợp của mạng nơ ron tích chập (CNN) và mạng nơ ron hồi quy (RNN, cụ thể là GRU) trong môi trường không có nhiễu và môi trường có nhiễu tại khoảng cách đặt micro đến người thu âm là 1m và 2m. Trong thử nghiệm với môi trường nhiễu, kết quả cho thấy, các mô hình được huấn luyện trong môi trường nhiễu hoạt động tốt hơn mô hình được huấn luyện trong môi trường sạch. Trong thử nghiệm về khoảng cách đặt micro đến người thu âm cho ta thấy, tại vị trí đặt micro là 1m và 2m không làm ảnh hưởng nhiều đến chất lượng của các hệ thống nhận dạng từ khóa tiếng Việt. Kết quả này là một cơ sở tham khảo cho việc xác định các vị trí đặt micro phù hợp trong bài toán xây dựng cơ sở dữ liệu tiếng nói tránh sự dư thừa về dữ liệu thu âm.

Từ khóa: Nhận dạng từ khóa; Nhận dạng tiếng nói; Khoảng cách xa; Mạng nơ ron tích chập; Mạng nơ ron hồi quy.

Received 06th April 2020
Revised 15th May 2020
Published 12th June 2020

Author affiliations:
Hanoi University of Science and Technology (HUST).
*Corresponding author: xuan.tranthianh@hust.edu.vn.