Vietnamese Speech Command Recognition using Recurrent Neural Networks

Phan Duy Hung[1], Truong Minh Giang[2], Le Hoang Nam[3], Phan Minh Duong[4]
FPT University Hanoi, Vietnam

***Abstract*—Voice control is an important function in many mobile devices, in a smart home, especially in providing people with disabilities a convenient way to communicate with the device. Despite many studies on this problem in the world, there has not been a formal study for the Vietnamese language. In addition, many studies did not offer a solution that can be expanded easily in the future. During this study, a dataset of Vietnamese speech commands is labeled and organized to be shared with community of general language research and Vietnamese language study in particular. This paper provides a speech collection and processing software. This study also designs and evaluates Recurrent Neural Networks to apply it to the data collected. The average recognition accuracy on the set of 15 commands for controlling smart home devices is 98.19%.**

Keywords*—Vietnamese speech command; voice control; Recurrent Neural Networks

I. Introduction

Interaction and control of household devices is a fast trend, evident in the exponentially growing number of smart-homes. According to Statista, the number of active households worldwide is 67.4 million in 2019. And this number is expected to amount to 111.2 million by 2023 [1]. The goal of research in this field is to improve the interaction so that it is faster, more convenient and more flexible. Therefore, speech recognition and natural language processing with the support of Artificial Intelligence seems to be the inevitable route.

In "Binary Neural Networks for Classification of Voice Commands from Throat Microphone" [2], the authors uses binary classifiers and Neural Networks (NNs), together with a perceptual linear prediction method for feature extraction to increase the classification rate of voice commands captured using a throat microphone, comparing this method with a single NN. They create a dataset of 150 people (men and women). All the voice samples are captured in Brazilian Portuguese, with the digits "0" through "9" and the words "Ok" and "Cancel". The results show that a throat microphone is robust in noisy environment, achieving a 95.4% hit rate in a speech recognition system with multiple NNs using the one-against-all approach, while a simple NN could only reach 91.88%.

In "Design and Implementation of Voice Command Using MFCC and HMMs Method" [3], feature extraction methods used are the Mel frequency cepstral coefficient (MFCC). Early stages of MFCC split the input signal amplitude values into frames which are then processed by the mel-filter bank. The results of feature extraction are made into a codebook, which is then used as an input symbol on a Hidden Markov Model (HMM) to form a model for every word. During testing, the characteristics of the test signal that has been quantized are then matched with the model created in the previous step. The final system can recognize spoken words with an average accuracy of 93.89% in a noiseless environment, and 58.1% in noisy environments.

In recent years, Deep Learning have become one of the common approaches used in speech recognition (SR), with SR systems based on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) achieving great results in various SR benchmarks [4,5,6].

In "Speech Recognition Based on Convolutional Neural Networks" [7], Du Guiming et al. propose to use the CNN principles in frequency domain to normalize acoustic variations for speech recognition. Here the researchers use a 5-layer CNN. It can achieve isolated word recognition by training the CNN.

In "End-to-End Speech Command Recognition with Capsule Network" [8], Jaesung Bae, Dae-Shik Kim realize that CNNs are capable of capturing the local features effectively. They can be used for tasks which have relatively short-term dependencies, such as keyword spotting or phoneme-level sequence recognition. However, one limitation of CNNs is that, with maxpooling, they do not consider the pose relationship between low-level features. Motivated by this, the researchers use a capsule network to capture the spatial relationship and pose information of speech spectrogram features in both the frequency and time axes. Compared to CNN models, the capsule-network-based systems achieved much better results from both clean and noisy data.

The above studies have shown that deep learning is the most effective solution at the present time to improve accuracy. It also seems that the end result can minimize the dependence of the problem on a specific language when the learning data is big enough. The gap here is to evaluate and customize a network architecture that matches a particular language database. In addition, the solution should be easily expanded on in the future.

The main contribution of this paper is to develop a method for recognizing Vietnamese speech commands based on deep learning technologies. A Vietnamese command dataset that includes 15 commonly used commands for smart homes (Table I) has been labeled and is publicly available for the research community on GitHub [9]. In addition, the source code for the data collection software for both Android and iOS is also made available on Github. Users can easily contribute

data via software, and it can also be easily modified for other languages. New commands can also be added in the future.

TABLE I. LIST OF SPEECH COMMANDS

No	In Vietnamese	equivalent English meaning
1	Đô rê mon	Doraemon (Trigger word)
2	Bật đèn	Turn on light
3	Tắt đèn	Turn off light
4	Bật điều hòa	Turn on air conditioner
5	Tắt điều hòa	Turn off air conditioner
6	Bật quạt	Turn on fan
7	Tắt quạt	Turn off fan
8	Bật tivi	Turn on TV
9	Tắt tivi	Turn off TV
10	Mở cửa	Open door
11	Đóng cửa	Close door
12	Khóa cửa	Lock door
13	Mở cổng	Open gate
14	Đóng cổng	Close gate
15	Khóa cổng	Lock gate

The remainder of the paper is organized as follows. Section 2 describes the data collection process. The data processing step is presented in Section 3. Section 4 provides the selection and evaluation of deep machine learning architectures based on RNN. Then, Section 5 analyzes the application results of the selected architecture for the Vietnamese command dataset. Finally, conclusions are made in Section 6.

II. DATA COLLECTION AND PROCESSING

A. Data Collection Software

In order to ensure the robustness and high accuracy of the identification process, the collected data needs to meet a number of requirements such as diversity in age, gender and region. In addition, the software needs to be easy to use. The software interface should be friendly and easy to understand.

The "*SpeechCollection*" application is written for both Android and iOS to be accessible to all users. Volunteers who wish to participate in data collection can download the software via Google's Play Store or Apple's App Store. Each speech command is recorded at the sampling frequency of 8Khz. The results are shared through Google cloud. Data will then be reviewed by the research team for quality and information, and the results will be stored in the final data directory. The main functions of the application are described in Fig. 1. An activity diagram is shown in Fig. 2. Fig. 3 shows the interface of the application in the sequence that a participant contributes data.

B. Data Organization

Each participant in the data collection will contribute a directory tree. The outermost folder is Name_Email. Next are the age_gender_province_timestamp. The bottom is the subdirectories corresponding to the words in the list of 15 desired words.

All data will be compressed before being sent to Google Cloud to reduce upload time. The average upload time is about 5s with a 3G or Wi-Fi connection under normal conditions.

The data collected after 1 month of the research contains voices of 293 people with fairly balanced ratios between men and women, age groups (younger than 18, between 18 and 30, between 30 and 40, older than 40) and regions (Northern, Central and Southern).

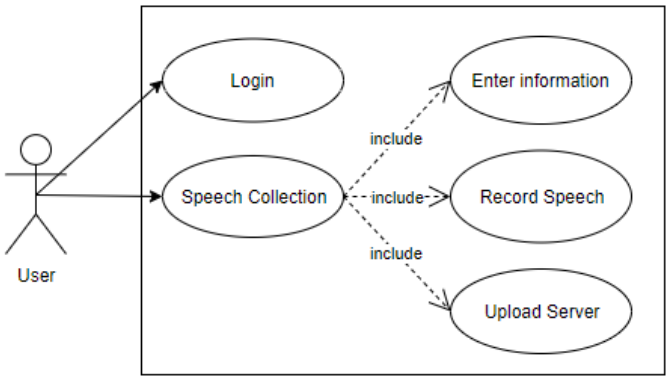


Fig. 1. Use Case Diagram.

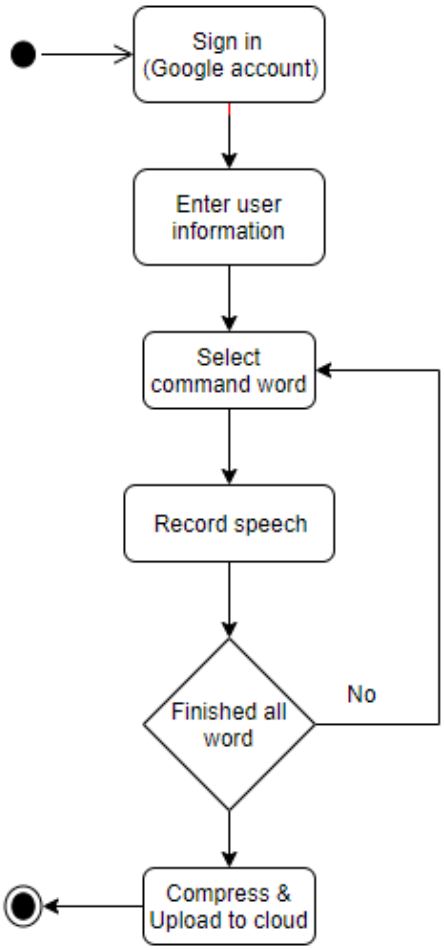


Fig. 2. Activity Diagram.

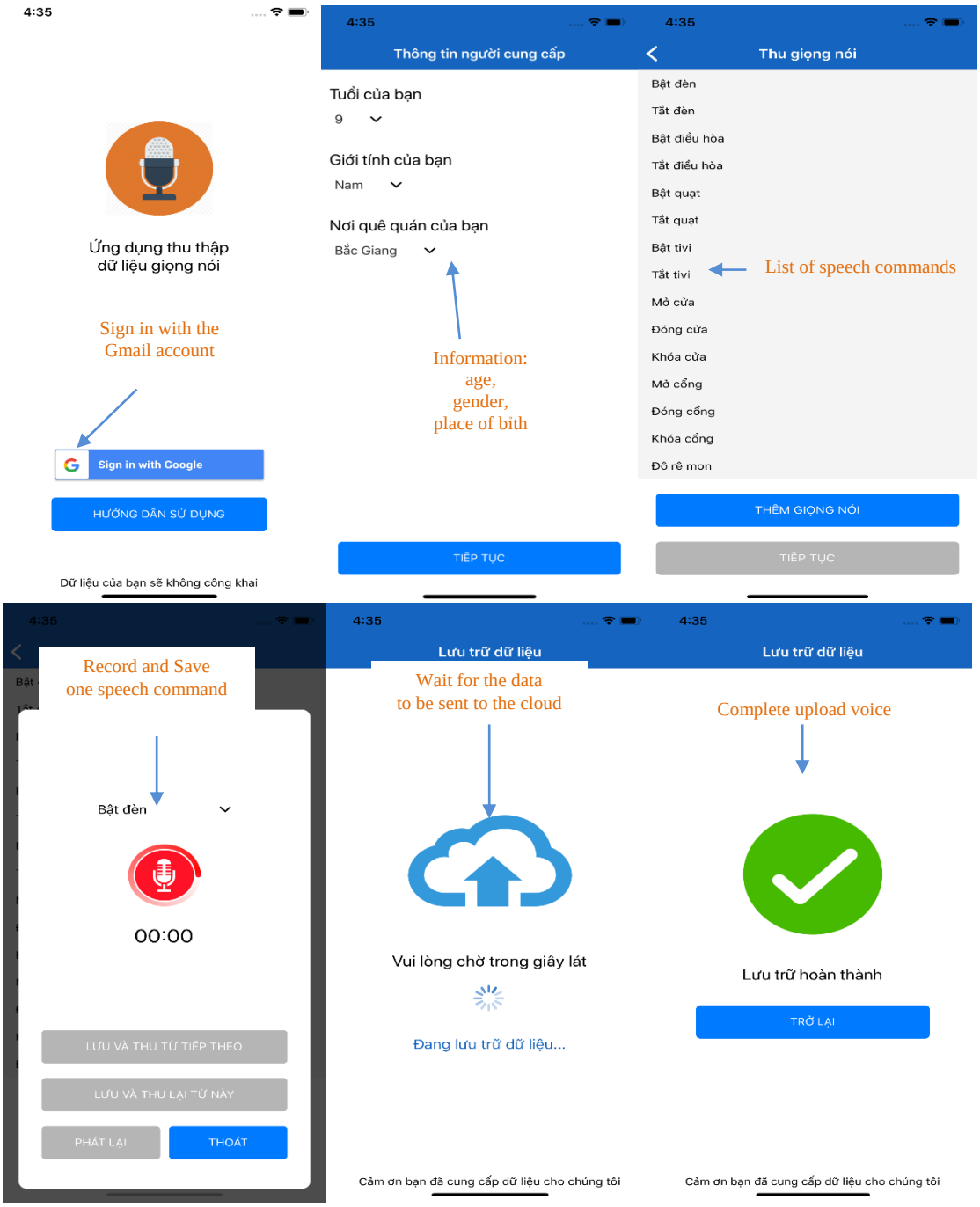


Fig. 3. Screen Flow of "SpeechCollection" Application.

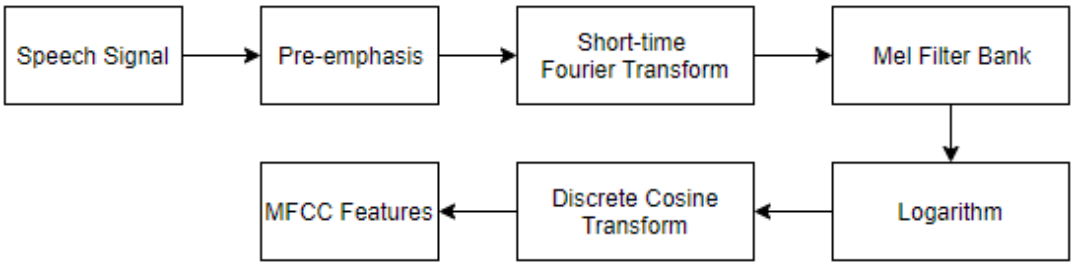


Fig. 4. MFCCs Extraction Algorithm.

C. Data Processing

1) Data filtering and trimming: The raw data contributed by user might contains a lot of silences and noises. So, the first step to filte and trim the silences at the beginning and the end [10]. Because the frequency range of speech signals is from 300Hz to 3400 Hz, a simple linear bandpass filter is used to eliminate out-of-band noise. Filtered data then is trimmed to remove silences. The data is then divided into continuous frames with a length of 0.05 seconds each. The Short Time Energy (STE) on each frame is calculated and compared to the average STE value. Frames with STE greater than the average value are retained whiel the rest is treated as the silence and removed.

2) Data augmentation: Data after trimming has different lengths, and the maximum length is less than 1.5 seconds. However, the data needs to be standardized to the same duration to make it easier to extract features used for machine learning. In addition, the data should be similar despite being collected in different environments. So, in this step the data needs to be augmented, and lengthen to a standard duration.

For background noises, audio recordings were conducted in 10 different environments (library, school, kitchen, room, road, etc.). Each recording has the same length of 2 seconds. These data are used then as background sounds, overlaid over the trimmed audio in the previous step at random. After data augmentation, we obtain the final dataset of approximately 3200 data samples for each speech command.

3) Feature extraction: Each 2-second sample of data is a time-series signal, from which features are extracted to provide a deep learning network input. The feature extraction algorithm is described as shown in Fig. 4 [10].

The first step is to apply a pre-emphasis filter on the signal to amplify the high frequencies.

The second step, a STFT transform is used because spectral analysis show that different timbres in speech signals corresponds to different energy distribution over the frequencies. The speech signal is segmented into frames of 25ms with an overlap of 15ms for each of the frame. The winstep is 10ms (25ms -15ms) and NFFT = 512, therefore each 2-second audio will be split into 200 frames, each FFT frame has NFFT / 2 + 1 = 257 frequency bins. The spectrogram has shape = (257, 200).

The third step utilizes the mel scale, a psychoacoustic scale of pitches of sounds. It is a scale that more closely represent what the human ears capture. The transfer formula is rather simple:

$$m = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \quad (1)$$

Each spectrum frame is multiplied with the corresponding filter, then the results are added to get a filter bank response. So, with M filters, this results in M filter bank energy vectors on a frame. In this study, the value of M selected is 13.

Finally, the logarithmic spectrum of the mel scale is converted into the time scale by using the DCT. A cepstrum is the result of taking the inverse transform of the logarithm of the estimated spectrum of a signal. Apply DCT on the 13 Log Filterbank Energies x(n) to have 13 Mel-scale cepstral coefficients. For each frame of spectrogram, there are 13 Mel-scale cepstral coefficients, so MFCC features of a 2-second sample of speech signal is a two-dimensional array with a shape of (13, 200).

III. Neural Network Architecture

A. Proposed Architecture

For Deep Networks, the focus is mostly on the two major architectures: CNNs for image modeling and Long Short-Term Memory (LSTM) Networks (Recurrent Networks) for sequence modeling.

The goal of a CNN is to learn higher-order features in the data via convolutions [11]. They are well suited to object recognition of faces, individuals, street signs, platypuses, and many other aspects of visual data. However, in [7,8], the authors have stated that one limitation of CNNs is that, with maxpooling, they do not consider the pose relationship between low-level features.

Recurrent Neural Networks are in the family of feed-forward neural networks [11]. They are different from other feed-forward networks in their ability to send information over time-steps. Recurrent Neural Networks take each vector from a sequence of input vectors and model them one at a time. This allows the network to retain state while modeling each input vector across the window of input vectors. Modeling the time dimension is a hallmark of Recurrent Neural Networks. Recurrent Neural Networks can have loops in the connections. This allows them to model temporal behavior and gain accuracy in domains such as time-series, language, audio, and text.

Long Short-Term Memory (LSTM) is a type of RNN architecture that addresses the vanishing/exploding gradient problem and allows learning of long-term dependencies [11]. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. Therefore, we use this LSTM architecture for training models in Automatic speech command recognition.

When using unidirectional RNNs as generative models, it is straightforward to draw samples from the model in sequential order. However, inference is not trivial in smoothing tasks, where we want to evaluate probabilities for missing values in the middle of a time series. In bidirectional RNNs, data processed in both directions processed with two separate hidden layers, which are then fed forward into the same output layer. Therefore, this can better exploit context in both directions. Hence bidirectional LSTMs usually perform better than unidirectional ones in speech recognition.

In this study, we implement and evaluate the performance of two models, unidirectional RNNs and bidirectional RNN for automatic speech command recognition.

B. Implementation of the Neural Network

The Keras interface was used to implement all neural networks on top of a Tensorflow backend. The Python library python_speech_feature is used to calculate MFCC for feature extraction.

1) Input of neural network: MFCC features are fed into the model. In each frame of audio, 13 features are obtained after the extraction algorithm. Each frame is added to the array in time order. Finally, the entire array of data after the process is added in the batch to train the model.

2) Models: To make sure the model works, it is first trained with a available dataset, the Google Speech Dataset V1 [12]. The Google Speech dataset V1 has 35 commands with audio files of 1 second in length. So, for each audio file, MFCC features is a two-dimensional array with a shape of (13, 100).

In the first model, unidirectional RNNs, speech command recognition depends on the data series over time, so on the top level of model, two LSTM layers are used to extract special features with long-term dependent of audio data. Then, the weighted average of the LSTM output is fed into 3 fully connected layers in the end for classification (Fig. 5).

In the second model, bidirectional RNNs, two Bidirectional LSTM (BiLSTM) are used. BiLSTM contains two single LSTM networks that are used simultaneously and independently to model the input chain in two directions: from left to right (forward LSTM) and from right to left (backward LSTM). Finally, the weighted average of the LSTM output is fed into 3 fully connected layers for classification (Fig. 6).

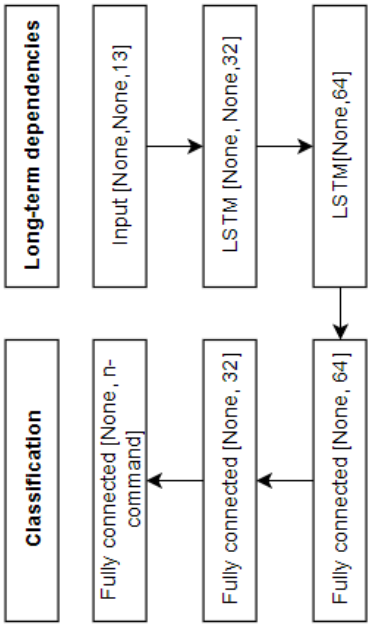


Fig. 5. LSTM Model for Speech Command Recognition.

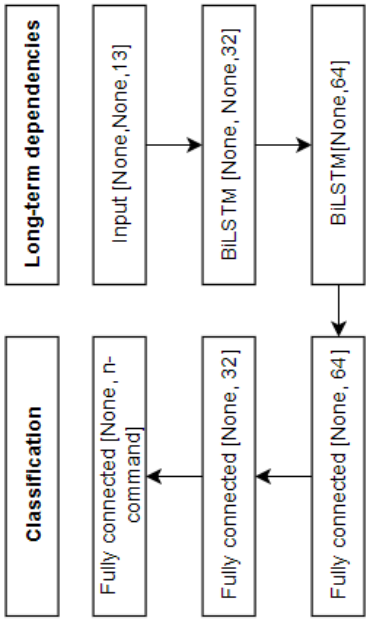


Fig. 6. BiLSTM Model for Speech Command Recognition.

In both model, activation of LSTM is a tanh function, and recurrent activation is hard_sigmoid function. All of these parameters are updated during the training process on the data sets labeled via the back-propagation algorithm with an Adam optimizer with learning rate of 0.001. The batch size used was 64. The LSTM model has 38,115 parameters and the BiLSTM model have 89,315 parameters.

C. Experiments and Model Analyzing

Each model was trained for a maximum of 10 epochs. The recognition results of both proposed models are compared with the results of Douglas Coimbra de Andrade et al. [13] as shown in Table II and Fig. 7. That comparison proves that both proposed models have very good results.

TABLE II. ACCURACY RESULTS ON THE GOOGLE SPEECH COMMAND DATASET V1

Model	Accuracy (%)	Trainable Parameters	Epochs
Douglas Coimbra de Andrade	94.3	202K	40
Douglas Coimbra de Andrade(V2)	93.9	202K	40
Unidirectional LSTM (ours)	92.1	38k	10
Bidirectional LSTM (ours)	94.6	89k	10

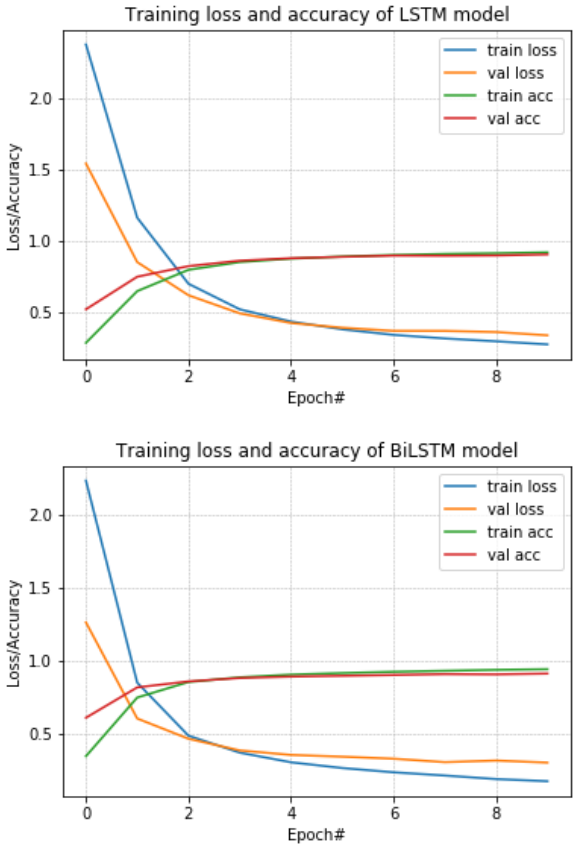


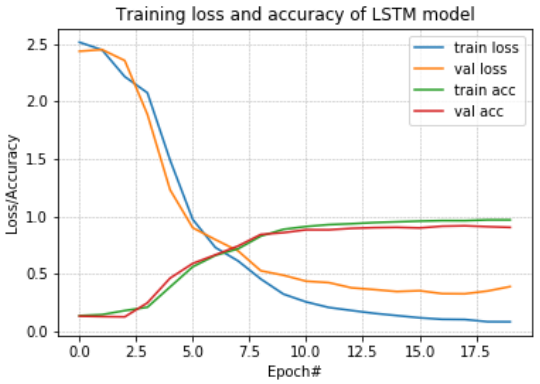
Fig. 7. Accuracy of LSTM (Above) and Accuracy of BiLSTM (Below).

IV. Apply to Recognition of Vietnamese Speech Commands

Using both of the above models for the collected Vietnamese command data set, the accuracy is shown in Fig. 8.

After 20 epochs, we can see that the BiLSTM model (98.19%) gives better results than the LSTM model (97.09%). However there is overfitting in both models. Next, dropout regularization is used for reducing overfitting and improving the generalization of deep neural networks technology. Dropout is a technique where randomly selected neurons are ignored during training, i.e. "dropped-out". This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. Dropout in Keras is implemented by randomly selecting nodes with a given probability (e.g. 20%) after each update cycle. This creates a small random amount of noise during learning and makes the architecture more flexible when processing speech data. The results of both models with dropout are shown in Fig. 9 which have significantly reduced overfitting.

Classification report of BiLSTM and Confusion matrix for 15 speech commands with Background are also shown in Table III and Fig. 10. We see that the two most confused commands are Turn off the fan and turn on the fan (in Vietnamese).



(a) Training Loss and Accuracy of LSTM, 97.09%.

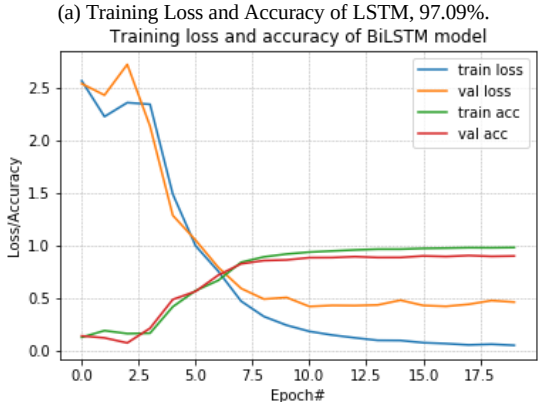
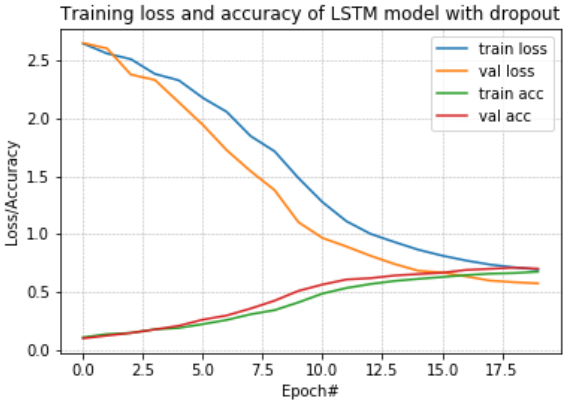


Fig. 8. (b) Training Loss and Accuracy of BiLSTM, 98.19%.



(a) Training Loss and Accuracy of LSTM with Dropout.

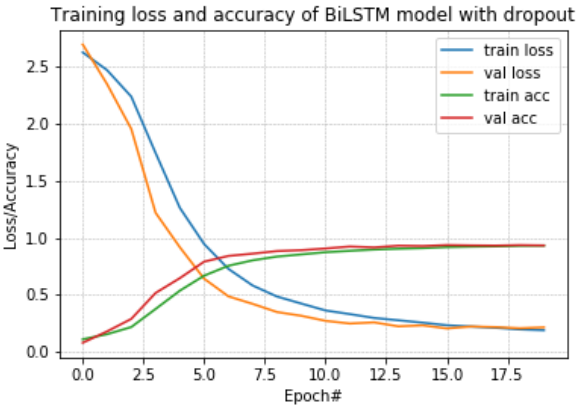


Fig. 9. (b) Training Loss and Accuracy of BiLSTM with Dropout.

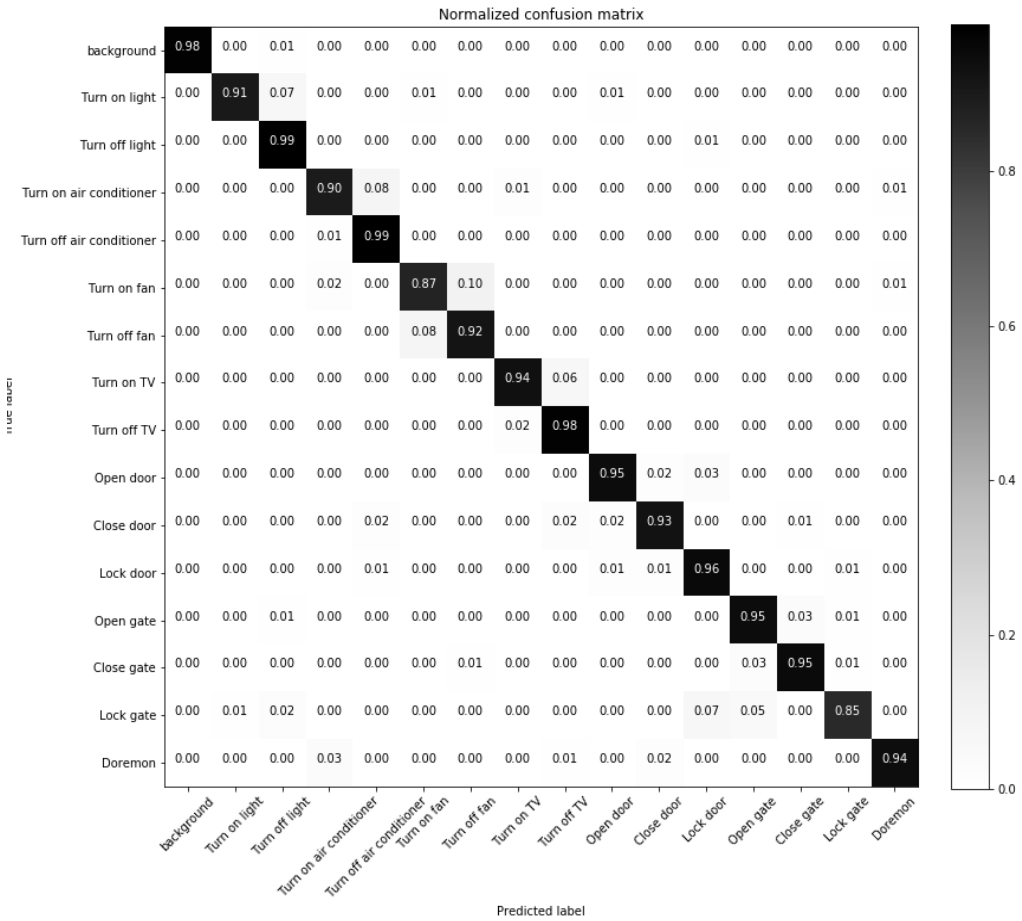


Fig. 10. Confusion Matrix of 15 Commands and Background.

TABLE III. CLASSIFICATION REPORT OF BILSTM

Command	precision	recall	f1-score	support
Turn on light	0.98	0.91	0.94	349
Turn off light	0.90	0.99	0.94	339
Turn on air conditioner	0.94	0.90	0.92	328
Turn off air conditioner	0.90	0.99	0.94	328
Turn on fan	0.90	0.87	0.88	328
Turn off fan	0.89	0.92	0.90	330
Turn on TV	0.97	0.94	0.95	330
Turn off TV	0.91	0.98	0.94	320
Open door	0.95	0.95	0.95	320
Close door	0.95	0.93	0.94	321
Lock door	0.89	0.96	0.92	321
Open gate	0.92	0.95	0.94	321
Close gate	0.96	0.95	0.96	318
Lock gate	0.98	0.85	0.91	320
Doraemon	0.98	0.94	0.96	329
Background	1.00	0.98	0.99	346

V. CONCLUSION AND PERSPECTIVES

This work contributes a Vietnamese command dataset including 15 commonly used commands for smart homes. After analysis and evaluation, we suggest using the BiLSTM model for recognizing Vietnamese speech commands. In addition, research also gives software for collecting data on Android and iOS. The identification result of the BiLSTM model on this dataset is very good, with accuracy averaging 98,19%.

The solution can be easily expanded on, for example adding commands, adding data. So, future results can be improved to better meet actual problems. The solution can also be transferred to other languages.

This work can serve as a good reference for many fields in Deep learning, for example, Pattern Recognition [14], CNN [15], Optimization Methods and Regularization in Deep learning [16], etc.

REFERENCES

[1] Statista. 2019. [Online] Available at: https://www.statista.com/outlook/389/100/smart-appliances/worldwide#market-users (Accessed on 5 May 2019).

[2] F. C. Ribeiro, R. T. S. Carvalho, P. C. Cortez, V. H. C. De Albuquerque and P. P. R. Filho, "Binary Neural Networks for Classification of Voice Commands From Throat Microphone," in IEEE Access, vol. 6, pp. 70130-70144, 2018. DOI: 10.1109/ACCESS.2018.2881199.

[3] M. Sidiq, W. T. Agung Budi and S. Sa'adah, "Design and implementation of voice command using MFCC and HMMs method," 2015 3rd International Conference on Information and Communication Technology (ICoICT), Nusa Dua, 2015, pp. 375-380. DOI: 10.1109/ICoICT.2015.7231454.

[4] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," in IEEE Access, vol. 7, pp. 19143-19165, 2019. DOI: 10.1109/ACCESS.2019.2896880.

[5] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments". ACM Trans. Intell. Syst. Technol. 9, 5, Article 49 (April 2018), 28 pages. DOI: https://doi.org/10.1145/3178115.

[6] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications". ACM Comput. Surv. 51, 5, Article 92 (September 2018), 36 pages. DOI: https://doi.org/10.1145/3234150.

[7] D. Guiming, W. Xia, W. Guangyan, Z. Yan and L. Dan, "Speech recognition based on convolutional neural networks," 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, 2016, pp. 708-711. DOI: 10.1109/SIPROCESS.2016.7888355.

[8] Bae, J., Kim, D, "End-to-End Speech Command Recognition with Capsule Network". Proc. Interspeech 2018, 776-780, DOI: 10.21437/Interspeech.2018-1888.

[9] VSC Group. 2019. [Online] Available at: https://github.com/VSC-FU2019/VSC_FU.

[10] J. O. Smith III, Spectral audio signal processing, https://www.dsprelated .com/freebooks/sasp/ (Accessed on 10 Avril 2019).

[11] P. Josh and G. Adam, Deep Learning, A Practitioner's Approach, Chapter 4, 2017, O'Reilly Media, Inc.

[12] TensorFlow. 2019. [Online] Available at: https://www.tensorflow.org/tutorials/sequences/audio_recognition, (Accessed on 5 May 2019).

[13] D. C. de Andradea, S. Leob, M. L. Da Silva Vianac, C. Bernkopf, A neural attention model for speech command recognition, 2018, https://arxiv.org/pdf/1808.08929.pdf.

[14] P. D. Hung, D. Q. Linh, "Implementing an Android Application for Automatic Vietnamese Business Card Recognition", Pattern Recognit. Image Anal. (2019) 29: 156. DOI: https://doi.org/10.1134/S1054661819010188.

[15] N. T. Nam, P. D. Hung, "Pest detection on Traps using Deep Convolutional Neural Networks". In Proceedings of the 2018 International Conference on Control and Computer Vision (ICCCV '18). ACM, New York, NY, USA, 33-38. DOI: https://doi.org/10.1145/3232651.3232661.

[16] N. T. Nam, P. D. Hung, "Padding Methods in Convolutional Sequence Model: An Application in Japanese Handwriting Recognition". In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing (ICMLSC 2019). ACM, New York, NY, USA, 138-142. DOI: https://doi.org/10.1145/3310986.3310998.