International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Implementation of Voice Recognition Via CNN and LSTM

Gyeongseop Shin, Sang-Hong Lee

Abstract: ***The voice recognition system uses CNN a lot. This is because CNN has the optimized ability to recognize and classify targets. CNN, however, has a problem that the bigger the object to be recognized, the more expensive the computational costs are. In this paper, we are going to solve these problems through MFCC feature extraction and model roll combining CNN and LSTM to present the possibility of performing voice recognition even through low-cost devices.***

Keywords: Voice recognition, CNN, LSTM, MFCC.

I. INTRODUCTION

Recently, there has been an increase in equipment that recognizes commands through voice [1][2][3]. This is because it is convenient for users to give commands through voice rather than by pressing a button that is always in the same position, and also for manufacturers to make only one microphone than by creating a large number of buttons corresponding to the command. Voice recognition technology has been developed in the past through a number of studies, but it is now almost dead and the main method used in speech recognition is voice recognition through deep learning [3][4][5]. Voice recognition through deep learning started with the recognition of speech-level speech at first, and has now reached the recognition of speech-level speech at the current sentence-level.

II. RELATED RESEARCH

A. CNN

CNN is the English abbreviation for the Convolutional Neural Network, which refers to an artificial neural network algorithm created by imitating human optic nerve structures [6].

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Gyeongseop Shin, Department of Computer Science & Engineering, Anyang University, Anyang-si, Republic of Korea. Email: rudtjq0842@naver.com
Sang-Hong Lee*, Department of Computer Science & Engineering, Anyang University, Anyang-si, Republic of Korea. Email: shleedosa@gmail.com

CNN is specialized in recognizing images and has shown great strength in winning the 2012 ILSVRC (ImageNet Large Scale Visual Recognition Challenge), the University of Toronto in Canada. CNN's theoretical foundation was already completed in the 1990s, but it was not able to see light until the late 2000s due to the lack of high-performance computing devices to apply it. Currently, it is used in various fields due to development of GPU or TPU, which is a high-performance computing device, and there are many cases where it has been applied in voice recognition. They can change voice data into image form through pretreatment and then insert it into a model made on CNN to recognize voice.

B. RNN

RNN is an English abbreviation for the recursive Neural Network, a neural network algorithm with recursive connections in which the output of the neuron is feedback by input [7]. This can also be seen as having memory that currently stores information about the calculated results. RNNN is primarily used to recognize sequential information because of the nature of the state values in the previous state entering the input of the following calculations. Along with CNN, it is making great progress not only in voice recognition but also in machine translation.

C. OpenNMT

OpenNMT is an open-source project launched in December 2016 and is used in several research and industrial applications. OpenNMT was named the TOP 30 Machine Learning Project, which was surveyed by Mybridge for Professional in 2018. OpenNMT mainly achieves many achievements in machine translation, but not just machine translation but also voice recognition. If the original voice is provided, it applies STFT (Short-Time Fourier Transform) and then enters it into CNN to extract the voice characteristics and then converts the voice into text using LSTM decoder based on this feature [3].

D. DeepSpeech

DeepSpeech is an open-source project published by Mozilla and based on Baidu's DeepSpeech paper, it recognizes speech sounds by recognizing characteristics from voice data and matching them to the alphabet of the language. That is why DeepSpeech is one of the models that supports multi-language voice recognition, not just English voice.

E. MFCC (Mel - Frequency Cepstral Coefficient)

In this paper, we use the method of extracting characteristics from speech through MFCC.

MFCC is a technique to extract the characteristics of sound in a way first introduced by Davis and Mermelstein in the 1980s. Instead of targeting the entire entered sound, the spectrum for this section is analyzed and characterized over a given interval [8][9]. If you enter the original voice data directly into the model without extracting the characteristics in the same way as MFCC, the computational costs will be very high, which is shown by the comparison below.

III. DATA AND PREPROCESSING

The goal of this paper is to find a low-cost voice recognition method while maintaining the accuracy that is followed. The data used in this paper are three voice command data created by the TensorFlow and AIY teams, which contain 65,000 one-second voices recorded differently by different characters for 30 short words. This is a result of the participation of end users through the AIY website. The table below is a summary of the data set.

TABLE I: The Arrangement of Channels

Label	Number	Time	Capacity
bed	1713	28min 33sec	51.3MB
bird	1731	28min 51sec	51.8MB
cat	1733	28min 53sec	51.9MB
dog	1746	29min 06sec	52.4MB
down	2359	39min 19sec	71.1MB
eight	2352	39min 12sec	70.7MB
five	2357	39min 17sec	71.1MB
four	2372	39min 32sec	71.4MB
go	2372	39min 32sec	71.0MB
happy	1742	29min 02sec	52.3MB
house	1750	29min 10sec	52.6MB
left	2353	39min 13sec	71.0MB
marvin	1746	29min 06sec	52.5MB
nine	2364	39min 24sec	71.3MB
no	2375	39min 35sec	71.2MB
off	2357	39min 17sec	71.0MB
on	2367	39min 27sec	71.0MB
one	2370	39min 30sec	71.1MB
right	2367	39min 27sec	71.2MB
seven	2377	39min 37sec	71.6MB
sheila	1734	28min 54sec	52.3MB
six	2369	39min 29sec	71.6MB
stop	2380	39min 40sec	71.6MB
three	2356	39min 16sec	70.9MB
tree	1733	28min 53sec	51.9MB
two	2373	39min 33sec	71.3MB
up	2375	39min 35sec	71.0MB
wow	1745	29min 05sec	52.2MB
yes	2377	39min 37sec	71.5MB
zero	2376	39min 36sec	71.8MB
total	64721	17h 58min 41sec	1.90GB

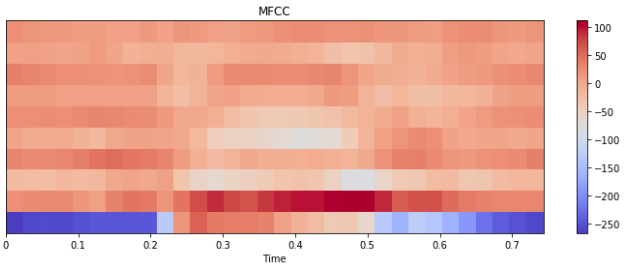

Fig. 1. The example of MFCC.

Voice is usually stored in PCM format, but voice recognition usually uses a method that stores 16,000 times per second. For example, if you call up a one-second-long voice 'dog', the data is stored, such as '894 1109 1136 895 1014 1011 1081 893 802 821 454 755 1188...' as the length of the voice increases, a space of 1 second*16000 will be required. Space is needed to store the values relative to the size of the input data, which leads to a waste of large space.

MFCC, on the other hand, takes up only less space (1,32) for the same voice data and retains only more important data, since the MFCC performs operations on each frame after the voice is cut into a smaller frame and then the operation is performed on each frame. Fig. 1 shows the result of character extraction through MFCC on the 'dog' voice data and output on the screen.

IV. IMPLEMENTATION OF THE VOICE RECOGNITION SYSTEM

In this paper, we use a combined model of CNN and LSTM, rather than a single configuration model using CNN alone, to reduce computational costs. In addition, a dropout layer was placed in the middle of the model to prevent over-connections and to finally perform the role of classifying the voice as the dense layer. Read the data from the voice file, extract the features via MFCC, enter them into the models combined with CNN and LSTM to recognize the voice and eventually classify the voice and return the results.

The implementation of the system takes place on the Ubuntu 16.04 operating system and the hardware configuration is Ryzen7 2700X, GTX 1060, 16GB RAM. The development of full-scale applications is conducted through the Jypeter Notebook in Anaconda 4.7.12 and the written language is Keras (backend: TensorFlow), Python's library.

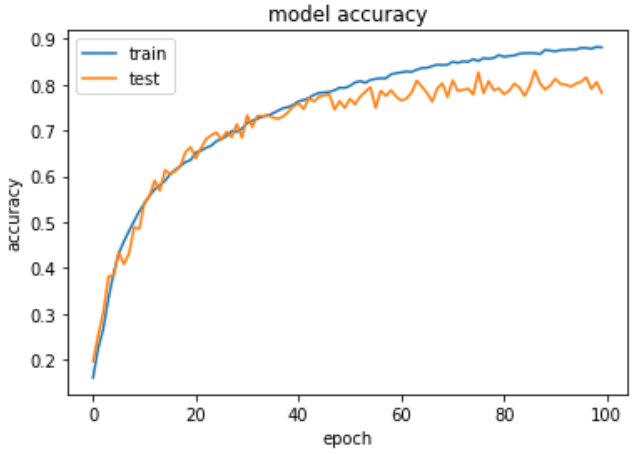

Fig. 2. The accuracy graph of CNN model. MFCC.

The performance comparison of this paper is conducted through a model made from CNN that uses the same data in the kagle. Fig. 2 shows the accuracy of the model with a verification accuracy of about 80% for the 12 voice data. Learning accuracy rises slightly after 40 epoch, but for verification accuracy there is little change after 40 epoch. Therefore, learning accuracy and verification accuracy will differ by about 10 percent once 100 epoch is finally reached.

This is because the model does not have enough storage space to store more important information and no other information, which causes overconformity. It is a model made up of CNN and has about 8 million parameters consisting of CNN and Dense layers to distinguish 12 voices.

On the other hand, the CNN and LSTM models used in this paper achieved 80% verification accuracy for 30 voice data, and the parameters used in the model were also much lighterthan those made up of CNN only. Fig. 3 is a graph of the accuracy of the model proposed in this paper.

For the model proposed in this paper, the learning speed is almost the same as that of CNN only, but there is little difference between learning accuracy and verification accuracy when finally achieved at 100 epoch. It also shows that 8 million parameters have been reduced to about 200,000, making them much lighter. Comparisons between the two show that the models that combined CNN and LSTM are more recognizable for voice and have lower computational costs than the models that use CNN only CNN.

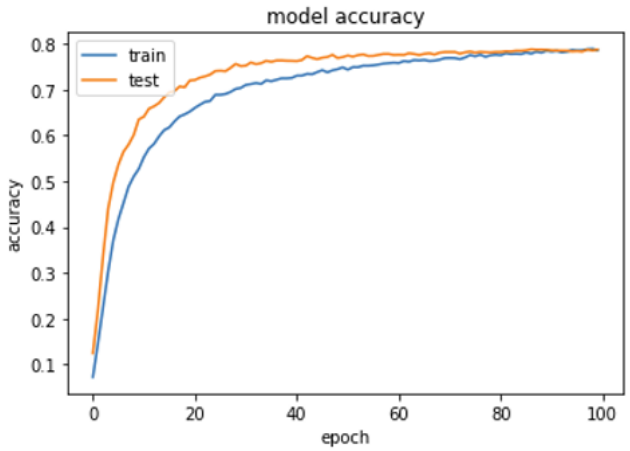

Fig. 3. The accuracy graph of CNN + RNN model.

V. CONCLUDING REMARKS

Through MFCC feature extraction and the model linking CNN and LSTM, CNN found that only CNN could perform a better level of voice recognition at a much lower cost than the model used for voice recognition, and that it could be applied to devices with less capacity and computing power. The simple connecting model of CNN and LSTM used in this paper has a problem that the accuracy is about 80 percent, which is not yet sufficient to be used in practice. In addition to models with more complex structures and MFCC, it is necessary to further improve accuracy through feature extraction or postprocessing.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1055423).

REFERENCES

1. Sarah M. Simmons, Jeff K. Caird, Piers Steel, "A meta-analysis of in-vehicle and nomadic voice-recognition system interaction and driving performance", Accident Analysis & Prevention, Vol. 106, 31-43, 2017.
2. Turker Tuncer, Sengul Dogan, "Novel dynamic center based binary and ternary pattern network using M4 pooling for real world voice recognition", Applied Acoustics, Vol. 156, 176-185, 2019.
3. Fatih Ertam, "An effective gender recognition approach using voice data via deeper LSTM networks", Applied Acoustics, Vol. 156, 351-358, 2019.
4. Haytham M. Fayek, Margaret Lech, Lawrence Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition", Neural Networks, Vol. 92, 60-68, 2017.
5. Angel Mario Castro Martinez, Sri Harish Mallidi, Bernd T. Meyer, "On the relevance of auditory-based Gabor features for deep learning in robust speech recognition", Computer Speech & Language, Vol. 45, 21-38, 2017.
6. Roxana ZahediNasab, Hadis Mohseni, "Neuroevolutionary based convolutional neural network with adaptive activation functions", Neurocomputing, Vol. 381, 306-313, 2020.
7. Haiqing Ren, Weiqiang Wang, Chenglin Liu, "Recognizing online handwritten Chinese characters using RNNs with new computing architectures", Pattern Recognition, Vol. 93, 179-192, 2019.
8. Savitha S. Upadhya, A. N. Cheeran, J. H. Nirmal, "Thomson Multitaper MFCC and PLP voice features for early detection of Parkinson disease", Biomedical Signal Processing and Control, Vol. 46, 293-301, 2018.
9. Zakariya Qawaqneh, Arafat Abu Mallouh, Buket D. Barkana, "Deep neural network framework and transformed MFCCs for speaker's age and gender classification", Knowledge-Based Systems, Vol. 115, 5-14, 2017.

AUTHORS PROFILE

Gyeongseop Shin is now a senior at Anyang University. His research focuses on deep learning systems and HCI systems.

Sang-Hong Lee received the B.S., M.S., and Ph.D. degrees in computer science from Gachon University, Korea in 1999, 2001, and 2012, respectively. He is currently an assistant professor in the department of computer engineering at Anyang University, Korea. His research focuses on deep learning systems, neuro-fuzzy systems, and biomedical prediction systems.