Spoken Keyword Spotting

Prakhar Gupta, *IISc Bangalore*

***Abstract*—With the rapid development of mobile devices, speech-related technologies are becoming increasingly popular. For example, Google offers the ability to search by voice on Android phones, while personal assistants such as Google Now, Apple's Siri, Microsoft's Cortana and Amazon's Alexa, all utilize speech recognition to interact with these systems. Now always-on speech recognition is generally not preferred due to its energy inefficiency and network congestion, also processing such a large amount of audio stream will require more time and adds to the latency and can have privacy issues. One way to solve this problem is by having the device continuously listen to specific keywords for initiating voice input. On successfully detecting such words, full-scale speech recognition is triggered on the cloud (or on the device). This KWS system is always-on, hence it is preferred to have a low memory footprint and computation complexity, along with having high accuracy.**

***Index Terms*—Keyword Spotting, CNNs, Deep Neural Networks, SVMs**

I. Introduction

Keyword Spotting (KWS) aims at detecting predefined keywords in an audio stream, and it is a potential technique to provide the desired hands-free interface. There is an extensive literature in KWS, although most of the proposed methods are not suitable for low-latency applications in computationally constrained environments. There are several approaches using Deep Neural Networks, Recurrent Neural Networks, LSTMs, CNNs with lots of variations. In this report, we have used Google Speech Commands dataset v1 [1] which contains 65,000 WAVE audio files of thirty words recorded by people with different demographic. Each recording is about one-second long and contains a single word recorded at 16KHz sampling rate. The dataset is categorized into 6798 validation recordings, 9916 test recordings and rest as training recordings. In this project, have used a CNN-DNN approach where the last layer is of 31 neurons for 31 classes, 1 additional class for background noise. For test, have taken *sheila* as the keyword which needs to be detected.

II. Literature Review

Traditional approaches for KWS are based on Hidden Markov Models with Keyword/Filler Hidden Markov Model (HMM) [2] which still remains highly competitive. In this generative approach, an HMM model is trained for each keyword, and a filler model HMM is trained from the non-keyword segments of the speech signal (fillers). At runtime, these systems require Viterbi decoding. Due to its high complexity, discriminative KWS approaches [3] were explored which are based on deep neural networks that are more appropriate for mobile devices. [4] trained a DNN architecure to directly predict the keyword(s) or subword units of the keyword(s) which follows a posterior handling method to produce a final confidence score. DNNs are not explicitly designed to model translational variance within speech signals, which can exist due to different speaking styles. Thus CNNs turn to be a befitting candidate for KWS task, because they capture translational invariance with far fewer parameters than DNNs by averaging the outputs of hidden units in different local time and frequency regions. [5] used LSTMs based feature extractor, they represent each keyword using a fixed-length feature vector obtained by running the keyword audio through a word-based LSTM acoustic model. CNNs are also capable of embedding speech by transforming them from input representations to meaningful representations. [6] presented a CNN architecture with less parameters by using strides in frequency. They also presented an improve performance by pooling in time and frequency. More recent architectures like [7] broadcasted residual learning method to achieve high accuracy with small model size and computational load.

III. KWS architecture

This KWS system is implemented in Python using TensorFlow framework and scikit-learn libraries. The input to the system is the log Mel Filterbank energies of the speech signal calculated with a window of length 25ms and stepsize 10ms. We train a 31 classes classifier based on CNN architecture using the training files in the dataset v1. The deep network transforms the input representations into meaningful representations, with which the final layer perform a classification. For CNNs, we used 6 Convolution layers, with the last convolution layer having a depth of 512. Further, 4 dense layers are used, with the last layer corresponding to 31 classes. In our model, there are around 930k trainable parameters.

IV. Experimental Results

Below, we have compared the results with several other architectures.

TABLE I
Baseline Model accuracy

Models	V1 dataset
DNN	86.7
CNN+strd	92.7
DSCNN	95.4
GRU	94.7
LSTM	94.8
CRNN	95.0
Att-RNN	95.6
BC-ResNET -8	**98.0**
This Work	**94.8**

Ran the model for 25 epochs, the loss and accuracies are shown below.

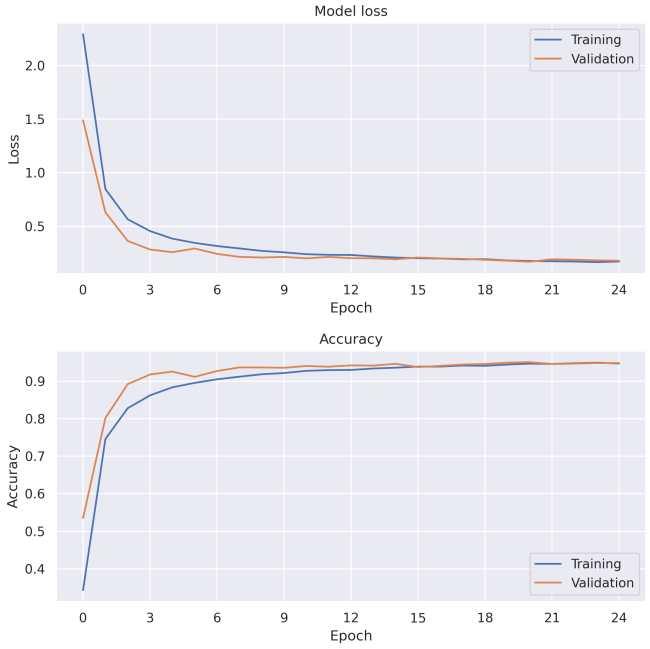

Fig. 1. Loss and Accuracy curve over epochs.

TABLE II
TRAINING AND VALIDATION LOSS AND ACCURACY

Epochs	Training Loss	Validation Loss	Training Accuracy	Validataion Accuracy
1	2.29	1.49	0.34	0.54
2	0.85	0.63	0.75	0.80
3	0.56	0.36	0.83	0.89
4	0.45	0.28	0.86	0.92
5	0.38	0.26	0.88	0.93
6	0.34	0.29	0.90	0.91
7	0.32	0.24	0.91	0.93
8	0.29	0.21	0.91	0.94
9	0.27	0.21	0.92	0.94
10	0.26	0.21	0.92	0.94
11	0.24	0.20	0.93	0.94
12	0.23	0.21	0.93	0.94
13	0.23	0.20	0.93	0.94
14	0.22	0.20	0.93	0.94
15	0.21	0.19	0.94	0.95
16	0.20	0.21	0.94	0.94
17	0.20	0.20	0.94	0.94
18	0.19	0.20	0.94	0.94
19	0.19	0.19	0.94	0.95
20	0.18	0.18	0.94	0.95
21	0.18	0.17	0.95	0.95
22	0.17	0.19	0.95	0.95
23	0.17	0.19	0.95	0.95
24	0.16	0.18	0.95	0.95
25	0.17	0.18	0.95	0.95

V. CONCLUSION

A small footprint re-configurable CNN-DNN based KWS system is implemented in this work. Have made a comparison with v1 dataset provided by Google Speech commands with various state of the art results.

REFERENCES

[1] Pete Warden. “Speech commands: A dataset for limited-vocabulary speech recognition”. In: *arXiv preprint arXiv:1804.03209* (2018).

[2] Richard C Rose and Douglas B Paul. “A hidden Markov model based keyword recognition system”. In: *International Conference on Acoustics, Speech, and Signal Processing*. IEEE. 1990, pp. 129–132.

[3] Shima Tabibian. “A survey on structured discriminative spoken keyword spotting”. In: *Artificial Intelligence Review* 53.4 (2020), pp. 2483–2520.

[4] Guoguo Chen, Carolina Parada, and Georg Heigold. “Small-footprint keyword spotting using deep neural networks”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 4087–4091. DOI: 10.1109/ICASSP.2014.6854370.

[5] Guoguo Chen, Carolina Parada, and Tara N Sainath. “Query-by-example keyword spotting using long short-term memory networks”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 5236–5240.

[6] Tara Sainath and Carolina Parada. “Convolutional neural networks for small-footprint keyword spotting”. In: (2015).

[7] Oleg Rybakov et al. “Streaming keyword spotting on mobile devices”. In: *arXiv preprint arXiv:2005.06720* (2020).