Artificial Intelligence

"Hey Alfred": Wake Word Detection

Replacing the turning-on button of the xArm robot (ALFRED) by a Wake Word Detection

Wim POIGNON

November - December 2021

Supervisior
Clément Duhart

Introduction

Intelligent voice assistant systems, such as smartphone assistants (e.g., Siri, Cortana, Google Now), Amazon Echo, and Google Home are becoming pervasive in our daily life. These human–machine communication systems are still emerging, mainly due to large researches in Deep Learning. Creating a personal voice assistant system improves the interaction with ALFRED, the xArm robot. This entire project consists of implementing all the voice assistant from the Automatic Speech Recognition (ASR) to Text-to-Speech (TTS) through Wake Word Detection. In this paper, we are focusing on the Wake Word Detection.

Replacing the Human Robot Interface (HRI) with speech commands is the next challenge for the innovative project ALFRED. The purpose of our trained voice assistant is to interact with a robotic arm, *e.g.* move to the right after saying *"move 20cm to the right"* or "*grab the red ball*". Voice assistant system uses multiple sub-domains from Deep Learning algorithms, like Transformers [1, 2] or Convolutional Neural Network (CNN)[3–5].

As the project is to code a voice assistant system, we design and train a Custom Model using a GPU because of parallel computing [6] and higher performance than CPU [7]. Recent speech enhancement research has shown that deep learning techniques are very effective in removing background noise [8].

In the last 5 years, Deep learning has become a powerful learning approach in speech recognition with greatly improved performance [9–11]. With the age, different models have appeared: Language [12], Pronunciation[13], or Acoustic Model [14] to name a few. The first model expresses the probability of any string in the language :

$$\mathbb{P}(W) = P\left(w_t | w_{t-k} \ldots w_{t-1}\right)$$

Then, Pronunciation Model maps the strings to the phonemes strings by finite state transducer composition [15]. And the last one, expresses the probability of acoustic observations generated by the phoneme models [16].

Audio Binary Classification

The Wake Word Detection project consists of classify audio files in a binary way to detect if a specific word is characterized. With the help of Deep Learning, we are building a Binary Classifier by taking an audio as input and expecting a Boolean. Our goal is to have a 90% accuracy, so we can deploy it on any innovative project.

Overview

This project is divided in three main parts. In the first place, we transform the audio-wave input into a specific image. In the second part, we are focusing on a Convolutional Neural Network (CNN) to extract key characteristics. Last part consists of programming a neural network to classify the featured map.

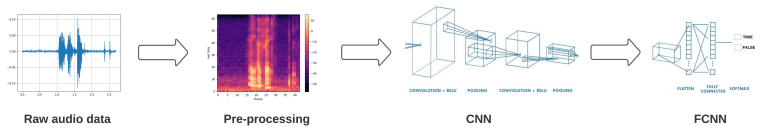

Figure 0.1: Entire process of the audio binary classification
CNN & FCNN image from Sumit Saha

Pre-processing

Before directly taking the audio sound as input, we need to prepare this audio data for the Deep Learning model. Typically a digital audio file is represented by the amplitude of sound at fixes intervals of time (called sample). Here, we are viewing the signal in the Time Domain (Fig. 0.2). However, with the raw audio data, it is difficult to extract key characteristics such as phonemes. It is used to transform this data in the Frequency Domain (Fig. 0.2). Spectrograms have become increasingly popular in recent times because they work well with Convolutional Neural Networks(CNN) [17, 18]. However, CNN models were built for natural images and 2-D spectrograms are different from natural images because natural images contain both space and time information. Spectrograms contain a temporal dimension, it makes them sequential data.

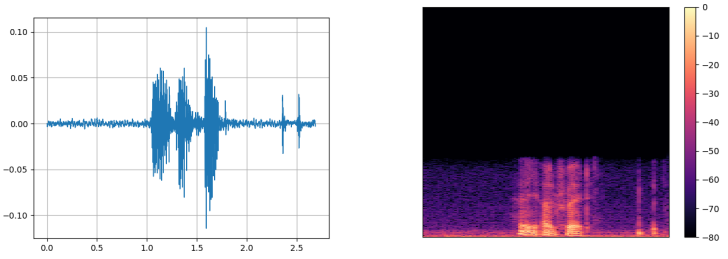

Figure 0.2: "*Hey Alfred*" - Time Domain (left) & Spectrogram (right)

Human are able to differentiate a sound with the frequencies, known as pitch. For instance, if we hear a acute sound, we know that we will have a dominance in the high frequencies. However, humans don't perceive frequencies linearly - we hear them on a logarithmic scale. It is why, the Mel scale was invented to relate real frequency to perceived frequency [19, 20]. After developing this idea, Mel Spectrogram has

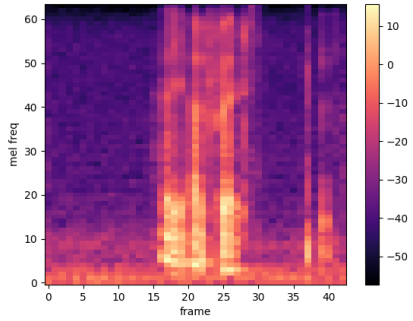

Figure 0.3: "*Hey Alfred*"- Mel spectrogram

appeared to represent a good way how we perceive sound on an image (Fig. 0.3)[21]. After pre-processsing audio data and converting into Mel spectrogram image, let's see how we can use this picture to actually classify audios.

Convolutional Neural Network

CNN based models have been used for a variety of tasks from Environment Sound Classification [22], Music Genre Classification [5] to Generative Audio [23]. The main purpose is to extract useful features from the input (it is also why, it is important to have a clean input image). To do it, in image processing, we know a wide range of different filters. Each of this convolutional transformation helps to extract unique aspects (horizontal, vertical edges, ...). In Deep Learning, there exist more or less complex models that use a large number of convolutional layers, such as ResNet15 or InceptionV3. In our case, we are using a simple 3x3 Convolutional Network. Each convolutional layer is followed by a ReLU activation function and a MaxPooling subsampling function as shown below (Fig. 0.4) [24].

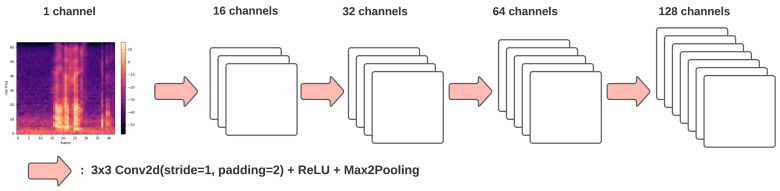

Figure 0.4: Convolutional Neural Network

Binary Classification by using Neural Network

Nowadays various Neural Network has been already developed with great benchmarks. The forefather of neural networks is the Fully Connected Neural Network, where, as the name suggests, all the layers are joined together. The performance of this model can change with the number of layers [25].

Results

Our train model has, at the moment, an accuracy of 56.2%. Actually, we have only around 40 samples of 2 seconds each one. However, to increase

our data, we can use the SpecAugment[26], a simple data augmentation method for speech recognition.

Product Specific Success Criteria

Example of PSSC

Items	Description	Mark (total 20)
Pre-processing	Collecting data	2
	Transform data to a dataset	2
	SpecAugment (add new data virtualy)	2
CNN	Do a simple 3x3 CNN	2
Binary Classification	Implement a FCNN	2
	Implement and compare LSTM	2
Implementation	Work in real time	3
	Add it to Alfred Arm Robot	5
Total		20

Evaluation and User Studies

The code is available on GitHub: https://github.com/wimausberlin/voice-assistant-system.git.

Results

After augmented physically the dataset (from 40 samples to 250 samples of 2 seconds each), the accuracy of our trained model has increased from 56.2% to 90.4%. In addition, with the SpecAugment method, our model reached an accuracy of 94%.

Dataset	SpecAugment	Accuracy
40 samples		56.2%
250 samples		90.4%
250 samples	x	94%

Table 0.1: Trained model accuracy

Limitation

The accuracy of the trained model is, in reality, biased principally due to the noise background. The dataset was only created in a specific environment (noise, personal voice, same pitch), The test dataset is not a realistic raw data. To avoid this phenom, the solution is to improve our dataset quality by recording the hot word in different environments (quite-noisy areas, different people, changing intonation).

In addition, the wake word detection can only recognize one voice because of the trained dataset made by one voice. To fix it, the easiest way is to augment the dataset physically by asking random people to pronounce multiple times the wake word.

Finally, the deployed application is tricky and not optimal. At first, the raw audio file must be saved on the computer and secondly can be treated by another script. This operation make some time (~ 1s.) due to the load of the model.

Future works

The deep learning model will have a principal streamer inference in future works. No need to record our voice before, a permanent streamer will analyze all two-second segments 1/8s. apart. To reduce space complexity, the constant recorded audio data will be stored in the buffer and be cleaned after 5 seconds (the time to make a prediction).

Moreover, this project is attended to be included in the innovative project ALFRED. However, ALFRED has a unique architecture with multiple applications. The purpose of this wake word detection is to implement it on ALFRED, by creating a class for the micro device and adding the real-time streamer with the activation word at the beginning of the entire system. This will allow us to start to use the arm robot in an easier way than typing some code lines in the terminal.

Conclusion

Keyword detection is a difficult task, given the great diversity to solve the problem and all the parameters to consider. We present a deep learning model for keyword recognition and its process. We trained the model on a small dataset to detect a specific pitch. We evaluate the proposed model on a test set and find the method is able to detect a specific voice with noise around it and does not require any extra model parameters.

Bibliography

[1] Jacob Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019 (cited on page 1).

[2] Yiming Wang et al. Wake Word Detection with Streaming Transformers. 2021 (cited on page 1).

[3] Lonce Wyse. 'Audio Spectrogram Representations for Processing with Convolutional Neural Networks'. In: (June 2017) (cited on page 1).

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 'ImageNet Classification with Deep Convolutional Neural Networks'. In: Commun. ACM 60.6 (May 2017), pp. 84–90. DOI: 10.1145/3065386 (cited on page 1).

[5] Mingwen Dong. Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification. 2018 (cited on pages 1, 3).

[6] Song Jun Park. 'An Analysis of GPU Parallel Computing'. In: 2009, pp. 365–369. DOI: 10.1109/HPCMP-UGC.2009.59 (cited on page 1).

[7] Ebubekir Buber and Banu Diri. 'Performance Analysis and CPU vs GPU Comparison for Deep Learning'. In: Oct. 2018, pp. 1–6. DOI: 10.1109/CEIT.2018.8751930 (cited on page 1).

[8] Soha Nossier et al. 'An Experimental Analysis of Deep Learning Architectures for Supervised Speech Enhancement'. In: Electronics 10 (Dec. 2020), p. 17. DOI: 10.3390/electronics10010017 (cited on page 1).

[9] Rishita Anubhai Dario Amodei. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. 2015 (cited on page 1).

[10] George Saon et al. The IBM 2016 English Conversational Telephone Speech Recognition System. 2016 (cited on page 1).

[11] Xiong Xiao, Shinji Watanabe, and Erdogan. 'Deep beamforming networks for multi-channel speech recognition'. In: 2016, pp. 5745–5749. DOI: 10.1109/ICASSP.2016.7472778 (cited on page 1).

[12] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 'LSTM Neural Networks for Language Modeling'. In: INTERSPEECH. 2012 (cited on page 1).

[13] Vipul Arora, Aditi Lahiri, and Henning Reetz. 'Phonological feature-based speech recognition system for pronunciation training in non-native language learning'. In: 143 (Jan. 2018), pp. 98–108. DOI: 10.1121/1.5017834 (cited on page 1).

[14] Geoffrey Hinton et al. 'Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups'. In: IEEE Signal Processing Magazine 29.6 (2012), pp. 82–97. DOI: 10.1109/MSP.2012.2205597 (cited on page 1).

[15] Timothy J. Hazen et al. 'Pronunciation modeling using a finite-state transducer representation'. In: Speech Communication 46.2 (2005). Pronunciation Modeling and Lexicon Adaptation, pp. 189–203. DOI: https://doi.org/10.1016/j.specom.2005.03.004 (cited on page 1).

[16] Dong Yu and Jinyu Li. 'Recent progresses in deep learning based acoustic models'. In: 4.3 (2017), pp. 396–409. DOI: 10.1109/JAS.2017.7510508 (cited on page 1).

[17] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. 2016 (cited on page 2).

[18] Sander Dieleman, Philemon Brakel, and Benjamin Schrauwen. 'Audio-based Music Classification with a Pretrained Convolutional Network'. In: ISMIR. 2011 (cited on page 2).

[19] S. S. Stevens and J. Volkmann. 'The Relation of Pitch to Frequency: A Revised Scale'. In: 53.3 (1940), pp. 329–353 (cited on page 2).

[20] S. Umesh, L. Cohen, and D. Nelson. 'Fitting the Mel scale'. In: vol. 1. 1999, 217–220 vol.1. DOI: 10.1109/ICASSP.1999.758101 (cited on page 2).

[21] David Dalmazzo and Rafael Ramirez. 'Mel-spectrogram Analysis to Identify Patterns in Musical Gestures: a Deep Learning Approach'. In: Nov. 2020 (cited on page 3).

[22] Andrey Guzhov et al. ESResNet: Environmental Sound Classification Based on Visual Domain Models. 2020 (cited on page 3).

[23] Aaron van den Oord et al. WaveNet: A Generative Model for Raw Audio. 2016 (cited on page 3).

[24] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. 'Rethinking CNN Models for Audio Classification'. In: (July 2020) (cited on page 3).

[25] S.H. Shabbeer Basha et al. 'Impact of fully connected layers on performance of convolutional neural networks for image classification'. In: 378 (2020), pp. 112–119. DOI: https://doi.org/10.1016/j.neucom.2019.10.008 (cited on page 3).

[26] Daniel S. Park et al. 'SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition'. In: (Sept. 2019). DOI: 10.21437/interspeech.2019-2680 (cited on page 4).