**VARIOUS NOTES FOR PAPER AND PRESENTATION**

[**Proposed End-to-End Architecture** 1](#_Toc45891597)

[**Preprocessing** 3](#_Toc45891598)

[**Dataset** 3](#_Toc45891599)

**Abstract**

We explore the application of ~~deep residual learning~~ and ~~dilated convolutions~~ to the keyword spotting task.

The proposed model establishes a new state-of-the-art accuracy of ~~94.1%~~ on Google Speech Commands dataset V2 (for the 20-commands recognition task), while still keeping a small footprint of only ~~202K~~ trainable parameters.

**Keywords**: human voice, command recognition, attention mechanism, deep learning, keyword spotting

**Introduction**

The goal of keyword spotting is to detect a relatively small set of predefined keywords in a stream of user utterances, usually in the context of an intelligent agent on a mobile phone or a consumer “smart home” device. Such a capability complements full automatic speech recognition, which is typically performed in the cloud. Because cloud-based interpretation of speech input requires transferring audio recordings from the user’s device, there are significant privacy implications. Therefore, on-device keyword spotting has two main uses: First, recognition of common commands such as “on” and “off” as well as other frequent words such as “yes” and “no” can be accomplished directly on the user’s device, thereby sidestepping any potential privacy concerns. Second, keyword spotting can be used to detect “command triggers” such as “hey Siri”, which provide explicit cues for interactions directed at the device. It is additionally desirable that such models have a small footprint (for example, measured in the number of model parameters) so they can be deployed on low power and performance-limited devices.

In recent years, neural networks have been shown to provide effective solutions to the small-footprint keyword spotting problem. Research typically focuses on a tradeoff between achieving high detection accuracy and having a small footprint. Compact models are usually variants derived from a full model that sacrifice accuracy for a smaller model footprint, often via some form of sparsification.

In this work, we focus on ~~convolutional neural networks (CNNs)~~, a class of models that has been successfully applied to small-footprint keyword spotting in recent years. In particular, we explore the use of ~~residual learning techniques~~ and ~~dilated convolutions~~.

<https://www.researchgate.net/publication/332553888_End-to-End_Environmental_Sound_Classification_using_a_1D_Convolutional_Neural_Network>

**Proposed End-to-End Architecture**

The aim of the proposed end-to-end architecture is to handle audio signals of variable lengths, learning directly from the audio signal, a discriminative representation that achieves a good classiﬁcation performance on diﬀerent speech commands.

2.1. Variable Audio Length

One of the challenges of using 1D CNNs in audio processing is that the length of the input sample must be ﬁxed but the sound captured from the environment may have various duration. Therefore, it is necessary to adapt a CNN to be used with audio signals of diﬀerent lengths. Moreover, a CNN must be used for continuous prediction of input audio signals of environmental sounds. One way to circumvent this constraint imposed by the CNN input layer is to split the audio signal into several frames of ﬁxed length using a sliding window of 1 second width.

Several other conﬁgurations can also be derived from subtle modiﬁcations of the base model (shown in Figure 2) to adapt it to shorter or longer audio inputs, as shown in Table 1. This implies modifying the number of convolutional layers as well as the number and the dimension of ﬁlters and the stride. However, for long contiguous audio recordings, instead of increasing the input dimension of the network, which also implies increasing the number of parameters, and consequently its complexity, it is preferable to split the audio waveform into shorter frames.

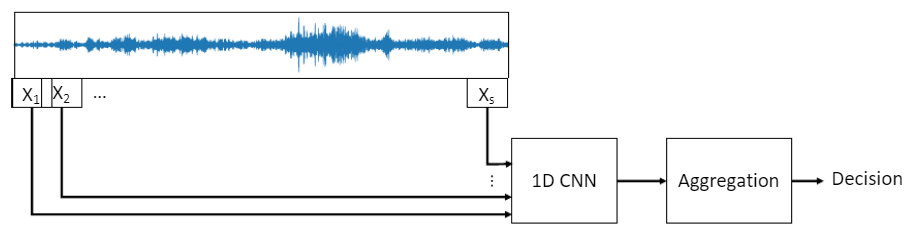
In this way, we keep the network compact and it can process audio waveforms of any length.

In the case of multiclass classiﬁcation, the number of neurons of the output layer is the number of classes. Using softmax as the activation function for the output layer, each output neuron indicates the membership degree of the input samples for each class. During the training process, the parameters of the network are adjusted according to the back-propagated classiﬁcation error and the parameters of the network are optimized to minimize an appropriate loss function (Goodfellow et al., 2016).

**Aggregation of Audio Frames**

In the case where the input audio waveform X is split into S frames denoted asX1, X2, . . . , XS, during the classiﬁcation we need to aggregate the CNN predictions to come up to a decision on X, as illustrated in Figure 4. For such an aim, diﬀerent fusion rules can be used to reach a ﬁnal decision, such as the majority vote or the sum rule, which are denoted in Equations 5 and 6 respectively.

When there are K classes, we generate K values and them for an audio input, we choose the class with the maximum yi value.



It is to computational expensive to use the model for each frame. So we can use a VAD (Voice Activity Detection) or Silence filter to let the model process just the frames that contains voice.

This component must be very compact, low computational expensive and very fast, like just ignoring anything under 40db, it can be a piece of software or even a little piece hardware like some home device are using.

**Preprocessing**

The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope. It models the characteristics of the human voice.

**Dataset**

The core words are "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", and "Nine".

To help distinguish unrecognized words, there are also ten auxiliary words, which most speakers only said once.

These include "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree", and "Wow".