

# EFFICIENT KEYWORD SPOTTING USING DILATED CONVOLUTIONS AND GATING

Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy,  
Mathieu Poumeyrol, Thibaut Lavril

Snips, Paris, France

## ABSTRACT

We explore the application of end-to-end stateless temporal modeling to small-footprint keyword spotting as opposed to recurrent networks that model long-term temporal dependencies using internal states. We propose a model inspired by the recent success of dilated convolutions in sequence modeling applications, allowing to train deeper architectures in resource-constrained configurations. Gated activations and residual connections are also added, following a similar configuration to WaveNet. In addition, we apply a custom target labeling that back-propagates loss from specific frames of interest, therefore yielding higher accuracy and only requiring to detect the end of the keyword. Our experimental results show that our model outperforms a max-pooling loss trained recurrent neural network using LSTM cells, with a significant decrease in false rejection rate. The underlying dataset – “Hey Snips” utterances recorded by over 2.2K different speakers – has been made publicly available to establish an open reference for wake-word detection.

**Index Terms**— end-to-end keyword spotting, wake word detection, dilated convolution, open dataset

## 1. INTRODUCTION

Keyword spotting (KWS) aims at detecting a pre-defined keyword or set of keywords in a continuous stream of audio. In particular, wake-word detection is an increasingly important application of KWS, used to initiate an interaction with a voice interface. In practice, such systems run on low-resource devices and listen continuously for a specific wake word. An effective on-device KWS therefore requires real-time response and high accuracy for a good user experience, while limiting memory footprint and computational cost.

Traditional approaches in keyword spotting tasks involve Hidden Markov Models (HMMs) for modeling both keyword and background [1, 2, 3]. In recent years, Deep Neural Networks (DNNs) have proven to yield efficient small-footprint solutions, as shown first by the fully-connected networks introduced in [4]. More advanced architectures have been successfully applied to KWS problems, such as Convolutional Neural Networks (CNNs) exploiting local dependencies [5, 6]. They have demonstrated efficiency in terms of inference

speed and computational cost but fail at capturing large patterns with reasonably small models. Recent works have suggested RNN based keyword spotting using LSTM cells that can leverage longer temporal context using gating mechanism and internal states [7, 8, 9]. However, because RNNs may suffer from state saturation when facing continuous input streams [10], their internal state needs to be periodically reset.

In this work we focus on end-to-end stateless temporal modeling which can take advantage of a large context while limiting computation and avoiding saturation issues. By *end-to-end* model, we mean a straight-forward model with a binary target that does not require a precise phoneme alignment beforehand. We explore an architecture based on a stack of dilated convolution layers, effectively operating on a broader scale than with standard convolutions while limiting model size. We further improve our solution with gated activations and residual skip-connections, inspired by the WaveNet style architecture explored previously for text-to-speech applications [11] and voice activity detection [10], but never applied to KWS to our knowledge. In [12], the authors explore Deep Residual Networks (ResNets) for KWS. ResNets differ from WaveNet models in that they do not leverage skip-connections and gating, and apply convolution kernels in the frequency domain, drastically increasing the computational cost.

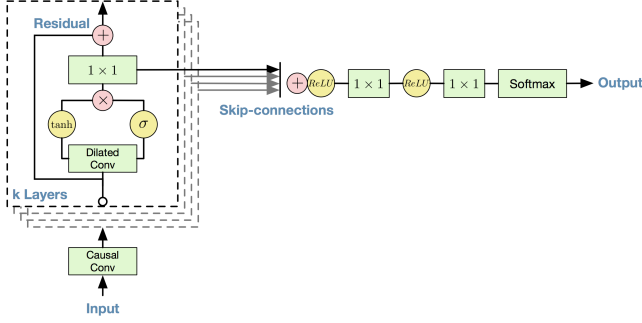
In addition, the long-term dependency our model can capture is exploited by implementing a custom “end-of-keyword” target labeling, increasing the accuracy of our model. A max-pooling loss trained LSTM initialized with a cross-entropy pre-trained network is chosen as a baseline, as it is one of the most effective models taking advantage of longer temporal contexts [8]. The rest of the paper is organized in two main parts. Section 2 describes the different components of our model as well as our labeling. Section 3 focuses on the experimental setup and performance results obtained on a publicly available “Hey Snips” dataset<sup>1</sup>.

## 2. MODEL IMPLEMENTATION

### 2.1. System description

The acoustic features are 20-dimensional log-Mel filterbank energies (LFBs), extracted from the input audio every 10ms

<sup>1</sup><https://research.snips.ai/datasets/keyword-spotting>



**Fig. 1.** WaveNet architecture [11].

over a window of 25ms. A binary target is used, see Section 2.4 for more details about labeling. During decoding, the system computes smoothed posteriors by averaging the output of a sliding context window containing  $w_{smooth}$  frames, a parameter chosen after experimental tuning. *End-to-end* models such as the one presented here do not require any post-processing step besides smoothing, as opposed to multi-class models such as [4, 5]. Indeed, the system triggers when the smoothed keyword posterior exceeds a pre-defined threshold.

## 2.2. Neural network architecture

WaveNet was initially proposed in [11], as a generative model for speech synthesis and other audio generation tasks. It consists in stacked causal convolution layers wrapped in a residual block with gated activation units as depicted in Figure 1.

### 2.2.1. Dilated causal convolutions

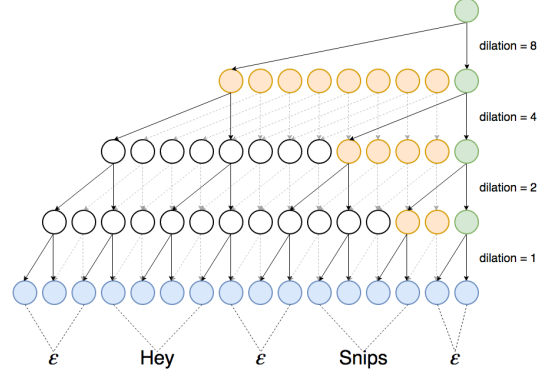
Standard convolutional networks cannot capture long temporal patterns with reasonably small models due to the increase in computational cost yielded by larger receptive fields. Dilated convolutions skip some input values so that the convolution kernel is applied over a larger area than its own. The network therefore operates on a larger scale, without the downside of increasing the number of parameters. The receptive field  $r$  of a network made of stacked convolutions indeed reads:

$$r = \sum_i d_i (s_i - 1),$$

where  $d_i$  refers to the dilation rate ( $d_i = 1$  for normal convolutions) and  $s_i$  the filter size of the  $i^{th}$  layer. Additionally, *causal* convolutions kernels ensure a causal ordering of input frames: the prediction emitted at time  $t$  only depends on previous time stamps. It allows to reduce the latency at inference time.

### 2.2.2. Gated activations and residual connections

As mentioned in [11], gated activations units – a combination of tanh and sigmoid activations controlling the propagation



**Fig. 2.** Dilated convolution layers with an exponential dilation rate of 1, 2, 4, 8 and filter size of 2. Blue nodes are input frame vectors, orange nodes are cached intermediate vectors used for streaming inference, green nodes are output vectors which are actually computed.  $\epsilon$  refers to background.

of information to the next layer – prove to efficiently model audio signals. Residual learning strategies such as skip connections are also introduced to speed up convergence and address the issue of vanishing gradients posed by the training of models of higher depth. Each layer yields two outputs: one is directly fed to the next layer as usual, but the second one skips it. All skip-connections outputs are then summed into the final output of the network. A large temporal dependency, can therefore be achieved by stacking multiple dilated convolution layers. By inserting residual connections between each layer, we are able to train a network of 24 layers on relatively small amount of data, which corresponds to a receptive field of 182 frames or 1.83s. The importance of gating and residual connections is analyzed in Section 3.3.2.

## 2.3. Streaming inference

In addition to reducing the model size, dilated convolutions allow the network to run in a streaming fashion during inference, drastically reducing the computational cost. When receiving a new input frame, the corresponding posteriors are recovered using previous computations, kept in memory for efficiency purposes as described in Figure 2. This cached implementation allows to reduce the amount of Floating Point Operations per Second (FLOPS) to a level suiting production requirements.

## 2.4. End-of-keyword labeling

Our approach consists in associating a target 1 to frames within a given time interval  $\Delta t$  before and after the end of the keyword. The optimal value for  $\Delta t$  is tuned on the dev set. Additionally, a *masking* scheme is applied, discarding background frames outside of the labeling window in positive samples. A traditional labeling approach, however, associates a target 1 to

		Train	Dev	Test
Hey Snips	utterances	5876	2504	2588
	speakers	1179	516	520
	max / speaker	10	10	10
Negative	utterances	45344	20321	20821
	speakers	3330	1474	1469
	max / speaker	30	30	30

**Table 1.** Dataset statistics.

all frames aligned with the keyword. In this configuration, the model has a tendency to trigger as soon as the keyword starts, whether or not the sample contains only a fraction of the keyword. One advantage of our approach is that the network will trigger near the end of keyword, once it has seen enough context. Moreover, our labeling does not need any phoneme alignment, but only to detect the end of the keyword, which is easily obtained with a VAD system. Furthermore, thanks to masking, the precise frontiers of the labeling window are not learned, making the network more robust to labeling imprecisions. The relative importance of end-of-keyword labeling and masking are analyzed in Section 3.3.2.

### 3. EXPERIMENTS

#### 3.1. Open dataset

The proposed approach is evaluated on a crowdsourced close-talk dataset. The chosen keyword is “Hey Snips” pronounced with no pause between the two words. The dataset contains a large variety of English accents and recording environments. Around 11K wake word utterances and 86.5K (~96 hours) negative examples have been recorded, see Table 1 for more details. Note that negative samples have been recorded in the same conditions than wake-word utterances, therefore arising from the same domain (speaker, hardware, environment, etc.). It thus prevents the model from discerning the two classes based on their domain-dependent acoustic features.

Positive data has been cleaned by automatically removing samples of extreme duration, or samples with repeated occurrences of the wake word. Positive dev and test sets have been manually cleaned to discard any mispronunciations of the wake word (e.g. “Hi Snips” or “Hey Snaips”), leaving the training set untouched. Noisy conditions are simulated by augmenting samples with music and noise background audio from Musan [13]. The positive dev and test datasets are augmented at 5dB of Signal-to-noise Ratio (SNR).

The full dataset and its metadata are available for research purposes<sup>2</sup>. Although some keyword spotting datasets are freely available, such as the Speech Commands dataset [14] for voice commands classification, there is no equivalent in the specific

wake-word detection field. By establishing an open reference for wake-word detection, we hope to contribute to promote transparency and reproducibility in a highly concurrent field where datasets are often kept private.

#### 3.2. Experimental setup

The network consists in an initial causal convolution layer (filter size of 3) and 24 layers of gated dilated convolutions (filter size of 3). The 24 dilation rates are a repeating sequence of  $\{1, 2, 4, 8, 1, 2, 4, 8, \dots\}$ . Residual connections are created between each layer and skip connections are accumulated at each layer and are eventually fed to a DNN followed by a softmax for classification as depicted in Figure 1. We used projection layers of size 16 for residual connections and of size 32 for skip connections. The optimal duration of the end-of-keyword labeling interval as defined in Section 2.4 is  $\Delta t = 160ms$  (15 frames before and 15 frames after the end of the keyword). The posteriors are smoothed over a sliding context window of  $w_{smooth} = 30$  frames, also tuned on the dev set.

The main baseline model is a LSTM trained with a max-pooling based loss initialized with a cross-entropy pre-trained network, as it is another example of end-to-end temporal model [8]. The idea of the max-pooling loss is to teach the network to fire at its highest confidence time by back-propagating loss from the most informative keyword frame that has the maximum posterior for the corresponding keyword. More specifically, the network is a single layer of unidirectional LSTM with 128 memory blocks and a projection layer of dimension 64, following a similar configuration to [8] but matching the same number of parameters than the proposed architecture (see Section 3.3.1). 10 frames in the past and 10 frames in the future are stacked to the input frame. Standard frame labeling is applied, but with the frame masking strategy described in Section 2.4. The authors of [8] mentioned back-propagating loss only from the last few frames, but said that the LSTM network performed poorly in this setting. The same smoothing strategy is applied on an window  $w_{smooth} = 8$  frames, after tuning on dev data. For comparison, we also add as a CNN variant the base architecture `trad-fpool3` from [5], a multi-class model with 4 output labels (“hey”, “sni”, “ps”, and background). Among those proposed in [5], this is the architecture with the lowest amount of FLOPS while having a similar number of parameters as the two other models studied here (see Section 3.3.1).

The Adam optimization method is used for the three models with a learning rate of  $10^{-3}$  for the proposed architecture,  $10^{-4}$  for the CNN, and  $5 \cdot 10^{-5}$  for the LSTM baseline. Additionally, gradient norm clipping to 10 is applied. A scaled uniform distribution for initialization [15] (or “Xavier” initialization) yielded the best performance for the three models. We also note that the LSTM network is much more sensitive to the chosen initialization scheme.

<sup>2</sup><https://research.snips.ai/datasets/keyword-spotting>

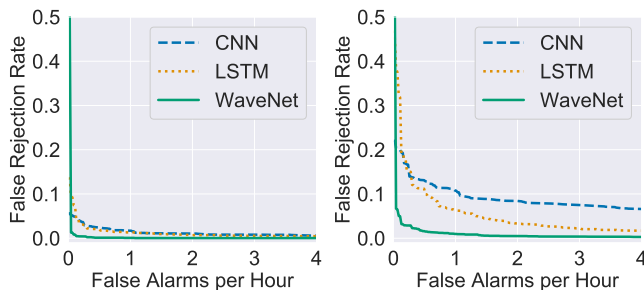
Model	Params	FLOPS	FRR clean	FRR noisy
WaveNet	222K	22M	0.12	1.60
LSTM	257K	26M	2.09	11.21
CNN	244K	172M	2.51	13.18

**Table 2.** Number of parameters, multiplications per second, and false rejection rate in percent on clean (FRR clean) and 5dB SNR noisy (FRR noisy) positive samples, at 0.5 false alarms per hour.

### 3.3. Results

#### 3.3.1. System performance

The performance of the three models is first measured by observing the False Rejection Rate (FRR) on clean and noisy (5dB SNR) positives samples at the operating threshold of 0.5 False Alarms per Hour (FAH) computed on the collected negative data. Hyper parameters are tuned on the dev set and results are reported on the test set. Table 2 displays these quantities as well as the number of parameters and multiplications per second performed during inference. The proposed architecture yields a lower FRR than the LSTM (resp. CNN) baseline with a 94% (resp. 95%) and 86% (resp. 88%) decrease in clean and noisy conditions. The number of parameters is similar for the three architectures, but the amount of FLOPS is higher by an order of magnitude for the CNN baseline while resulting in a poorer FRR in a noisy environment. Figure 3 provides the Detection Error Tradeoff (DET) curves and shows that the WaveNet model also outperforms the baselines on a whole range of triggering thresholds.



**Fig. 3.** DET curves for the proposed architecture (green) compared to the LSTM (dotted yellow) and CNN (dashed blue) baselines in clean (left panel) and 5dB SNR noisy (right panel) environments.

#### 3.3.2. Ablation analysis

To assess the relative importance of some characteristics of the proposed architecture, we study the difference in FRR observed once each of them is removed separately, all things

	FRR clean	FRR noisy
Default labeling	+0.36	+1.33
No masking	+0.28	+0.46
No gating	+0.24	+2.57

**Table 3.** Variation in FRR (absolute) for the proposed architecture when removing different characteristics separately, all things being equal.

being equal. Table 3 shows that the end-of-keyword labeling is particularly helpful in improving the FRR at a fixed FAH, especially in noisy conditions. Masking background frames in positive samples also helps, but in a lower magnitude. Similarly to what is observed in [10], gating contributes to improving the FRR especially in noisy conditions. We finally observed that removing either residual or skip connections separately has little effect on the performance. However, we could not properly train the proposed model without any of these connections. It seems to confirm that implementing at least one bypassing strategy is key for constructing deeper network architectures.

## 4. CONCLUSION

This paper introduces an end-to-end stateless modeling for keyword spotting, based on dilated convolutions coupled with residual connections and gating encouraged by the success of the WaveNet architecture in audio generation tasks [11, 10]. Additionally, a custom frame labeling is applied, associating a target 1 to frames located within a small time interval around the end of the keyword. The proposed architecture is compared against a LSTM baseline, similar to the one proposed in [8]. Because of their binary targets, both the proposed model and the LSTM baseline do not require any phoneme alignment or post-processing besides posterior smoothing. We also added a multi-class CNN baseline [5] for comparison. We have shown that the presented WaveNet model significantly reduces the false rejection rate at a fixed false alarm rate of 0.5 per hour, in both clean and noisy environments, on a crowdsourced dataset made publicly available for research purposes. The proposed model seems to be very efficient in the specific domain defined by this dataset and future work will focus on domain adaptation in terms of recording hardware, accents, or far-field settings, to be deployed easily in new environments.

## 5. ACKNOWLEDGEMENTS

We thank Oleksandr Olgashko for his contribution in developing the training framework. We are grateful to the crowd of contributors who recorded the dataset. We are indebted to the users of the Snips Voice Platform for valuable feedback.

## 6. REFERENCES

- [1] Richard C Rose and Douglas B Paul, "A hidden markov model based keyword recognition system," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 129–132.
- [2] Jay G Wilpon, Lawrence R Rabiner, C-H Lee, and ER Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.
- [3] JG Wilpon, LG Miller, and P Modi, "Improvements and applications for key word recognition using hidden markov modeling techniques," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 309–312.
- [4] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on*. IEEE, 2014, pp. 4087–4091.
- [5] Tara N Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [6] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.
- [7] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *International Conference on Artificial Neural Networks*. Springer, 2007, pp. 220–229.
- [8] Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 474–480.
- [9] Pallavi Baljekar, Jill Fain Lehman, and Rita Singh, "On-line word-spotting in continuous speech with recurrent neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 536–541.
- [10] Shuo-Yiin Chang, Bo Li, Gabor Simko, Tara N Sainath, Anshuman Tripathi, Aäron van den Oord, and Oriol Vinyals, "Temporal modeling using dilated convolution and gating for voice-activity-detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5549–5553.
- [11] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio.," in *SSW*, 2016, p. 125.
- [12] Raphael Tang and Jimmy Lin, "Deep residual learning for small-footprint keyword spotting," *arXiv preprint arXiv:1710.10361*, 2017.
- [13] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.
- [14] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
- [15] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.