AUDIAS System for the ALBAYZIN 2024 WuW Detection Challenge

Enrique Ernesto A. Doñate[1], *Doroteo T. Toledano*[1]

[1]AUDIAS, Universidad Autónoma de Madrid, Spain

enrique.dealvear@estudiante.uam.es, doroteo.torre@uam.es

Abstract

In response to the ALBAYZIN 2024 Wake-Up Word Detection Challenge, we propose a model inspired by the challenge's baseline model. The objective of this model is to accurately detect the phrase "Okey Aura" in audio segments and determine the precise time it is spoken.

The model consists of three key components. First, it extracts Mel Frequency Cepstral Coefficients (MFCC) from the audio input. The second and central component is a Residual Neural Network (ResNet), which efficiently processes the extracted features, taking advantage of residual connections to improve learning and feature extraction. Finally, the classification head processes the logits from the ResNet to determine whether the wake-up word sequence is present in the audio.

While the model is based on the baseline provided in the challenge, notable improvements were made, particularly in the data preprocessing phase and structural adjustments of the network, which significantly enhance its performance.

The model's performance was evaluated using the challenge metrics: Detection Cost Function (DCF) and Time Error Metric (TEM). Compared to the baseline, our model achieved a maximum of 50% reduction in TEM and a $25\% - 40\%$ reduction in DCF in the validation tests and similar results in the final evaluation, demonstrating significant improvements in both accuracy and temporal precision.

Index Terms: keyword spotting, speech recognition, human-computer interaction

1. Introduction

ALBAYZIN challenges have been organized biannually for over 20 years with the support of the Spanish Thematic Network on Speech Technologies, within the context of the IberSPEECH biannual conference. One of the challenges proposed this year was the Wake Up Word detection challenge (WuW), organized by Telefónica Innovación Digital, to assess the state of the art keyword spotting systems, in addressing various industrial needs such as accuracy, inference delay, computational load, and energy efficiency.

The WuW detection challenge revolves around the problem of Keyword Spotting (KWS) in audio segments. This problem consists in the detection of a sequence of words. For this, the KWS system is continuously listening until it detects the activation words, at which point it activates a secondary system to perform the task required after the activation sequence.

Traditionally the keyword spotting systems utilized Hidden Markov Models (HMMs) and Gaussian Markov Models (GMMs) for this task, and while they were effective, they usually struggle with real time audio processing, because of the computational cost. So they became obsolete when deep learning models took over this area.

Neural Networks became the new state of the art after HMMs and GMMs, using Convolutional Neural Networks (CNNs) [1, 2], Recurrent Neural Networks (RNNs) and Attention based Neural Networks such as Transformers [3]. These ones based on attention mechanisms trying to capture long time dependencies from larger contexts.

In our system we try to adopt the *LambdaResNet* architecture used in [4] and based in the Lambda layer architecture [5] for the WuW detection challenge. This architecture is the one used for the baseline model of the challenge. *LambdaResNet* architecture combines the purpose of capturing the long-range interactions without the attention maps of the Transformer and the state of the art *ResNet15* [6] for keyword spotting based on convolutions.

The model is trained on a dataset provided by the challenge organizers that consists of audio segments recorded in Madrid. This audios contain phrases, usually the activation phrase "Okey Aura" and others with variants of parts or the whole target phrase. This is the data used for the training of this model. The data is augmented by techniques like adding noise, volume shifting and reverberation. This process will be explained further in section 3.

2. Proposed Model

As mentioned, the model is based on the *LambdaResNet18* architecture used in the baseline provided for the challenge, but changing some aspects of the structure as shown in Figure 1. The model structure consists of 23 layers grouped in 3 fundamental blocks.

2.1. Feature extractor block

The first block of the network consists of the feature extraction block which is the first layer. The input audio segments are 1.5 seconds long, sampled at $16,000Hz$. The feature extraction block first computes the MelSpectrogram using a window size of 32 ms and a hop of 16 ms to extract 128 Mel bands. From these Mel bands, it then calculates the Mel Frequency Cepstral Coefficients with 13 MFCCs. Once the feature extraction is complete, the processed data is passed to the next block.

2.2. LambdaResNet block

The second block is the main component of the model, following the structure described in [4] for the residual Lambda block. This LambdaBlock consists of a convolutional layer followed by a ReLU activation and Batch Normalization, and then the Lambda layer as proposed in [5] with a Batch Normaliza-

tion. The number of channels for each residual layer follows the implementation of the original *ResNet* and *LambdaResNet18*, ranging from 24 to 60, with the sequence $\{24, 36, 48, 60\}$, and the residual layers are processed sequentially. In the proposed model, the first two of the four residual layers contain three residual Lambda blocks sequentially, while the other two contain two blocks. A stride of 2 is applied to each convolutional layer within the residual blocks.

2.3. Classification head

Finally, the classification head is a fully connected network composed of two linear layers. The first layer has 60 units, with a ReLU activation function, while the second layer has 2 outputs and uses a Softmax activation. This fully connected network produces the required output: a 2-dimensional vector, where the first component represents the probability that the audio does not contain the activation phrase, and the second component represents the complementary probability.

3. Experiments

3.1. Model training

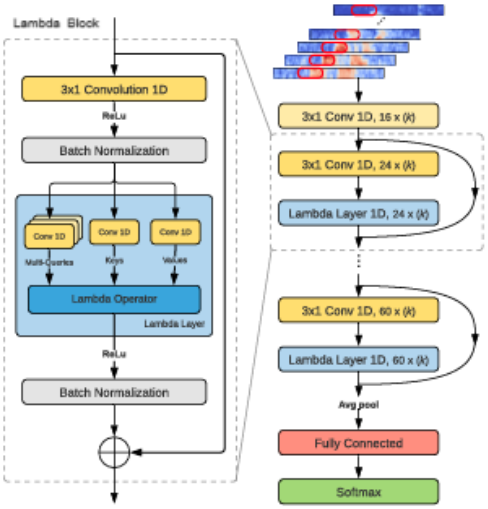
3.1.1. Training data

As stated in the Evaluation plan [7] the database consists of 1247 utterances from 80 speakers, stored in Waveform Audio File Format (WAV) by using a Pulse-Code Modulation (PCM) encoded with two bytes per sample at a rate of $16KHz$. The metadata is provided within a Tab Separated Values (TSV) format in files with the .tsv extension.

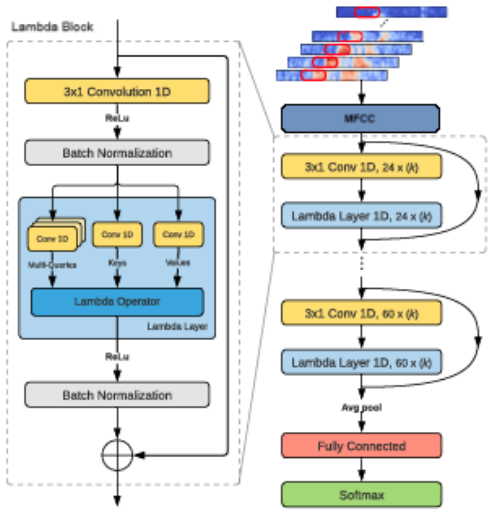
The database includes positives examples, that is, containing the keyphrase and negative examples which have phonetically similar phrases to the target for example "Okey Laura". Also there are some audios in the dataset consisting of the activation phrase and a command or some context, for example "Okey Aura, sube el volumen". The data is split in the .tsv metadata files *train.tsv*, *test.tsv*, *dev.tsv* and *other.tsv*, consisting on the training, development and test split, with the *other.tsv* containing those utterances with the keyphrase and more content. Finally there is a *test-extended.tsv* containing the *test.tsv* audios with noise recorded and provided in the *noise.tsv* to extend the test data. We can see the metadata of the training set in Table 1

Table 1: *Metadata in the Okey Aura WuW Dataset*

Metadata	Values
SpeakerID	Anonymized string of 16 chars
Age	20s, 30s, 40s...
Gender	Male, Female, Non-binary
Accent	Castilian, Andalusian...
Microphone_Distance	Close, Two steps away
Room_size	Small, Medium
Prosody	Unknown, Neutral, Annoyed, Friendly
Transcription	Spoken user utterance
Label	WuW or NonWuW
Similarity	Exact, Okey, Aura...
Audio_Length	Length of the audio in seconds
Start_Time	Start of the spoken content
End_Time	End of the spoken content



(a) LambdaResNet18 image obtained from [4]



(b) LRNMFCC

Figure 1: *Architecture comparison of both LambdaResNet18 and the proposed model.* ***Figure 1(a)*** *LambdaResNet18 starts with MelSpectrograms as the preprocessing step, followed by* 3×1 *convolutional filters and Lambda layers within the LambdaResNet structure, with alternating stride values of 2 and 1. The activation functions are ReLU, and Batch Normalization is included in each LambdaResNet layer. Each LambdaResNet layer consists of 2 blocks sequentially using this structure.* ***Figure 1(b)*** *LRNmfcc mainly differs of the previous model in the computation of the MFCCs from the MelSpectrogram, removing the initial* 3×1 *convolutional layer, and adding one more block to the first two LambdaResNet layers. As a result, the first two layers contain 3 blocks each instead of 2.*

3.1.2. Data preprocessing and training

For this challenge the data and metadata provided as mentioned in section 3.1.1 contains a lot of information of each audio, so we are going to preprocess the training data to get a better performance of the model. First to train the model we had to choose some parameters for our models we explain in Table 2.

For the preprocess and training we follow these steps:

1. **Data Labeling:** As previously mentioned, the audio segments contain the keyphrase, defined by the "Start_Time" and "End Time" in the samples labeled as "WuW". The rest of the audio is considered "Non-WuW". For training, we use segments with a "Time_Window" of 1.5 seconds, and the label assigned to each subsplit of the audio may vary. We chose to assign a label of "1" (corresponding to "WuW") to segments where the time window covers at least 85% of the keyphrase utterance. The remaining segments are labeled as "NonWuW" and assigned a label of "0". For audios originally labeled as "Non-WuW", each of their time windows is consistently labeled as "0".
2. **Data augmentation:** After labeling, we augmented the data by randomly selecting instances of the labeled segments. For data augmentation, we added noise using the *noise.tsv* file provided with the Okey Aura Wake-Up Word Dataset, applying different levels of SNR. We also performed volume shifting by multiplying the audio signal by a random number in the interval $(0.5, 2)$, and added reverberation. The reverberation was applied using the torch function `torchaudio.functional.fftconvolve()` as described in [8]. Since this function requires a Room Impulse Response (RIR) audio sample, we used samples from the VOiCES Dataset [9], specifically the audios in the "room-response" section of the dataset, the impulse for each one of the rooms recorded. The VOiCES Dataset is a Creative Commons-licensed speech dataset, designed for acoustically challenging and reverberant environments, providing robust labels and ground truth data for tasks such as transcription, denoising, and speaker identification.
3. **Batch class balance:** Given the large imbalance between the "NonWuW" and "WuW" audio classes, we opted to balance the batches to ensure that each batch contains instances from both classes, maintaining approximately a $30 - 70\%$ split of "WuW" and "NonWuW" respectively. Additionally, we included 15% more batches consisting only of "NonWuW" samples to help the model reduce the false acceptance rate.
4. **Training:**The model was trained using a Cross-Entropy loss function, with parameters updated every batch of 32 samples over 50 epochs. The training aimed to minimize the loss on the *dev.tsv* dataset split. The optimization process employed the *Adam* optimizer with a learning rate of 0.001.

4. Tests results

We trained four different models LRNMFCC256, LRNMFCC256-2, LRNMFCC128, LRNMFCC128-2 and compared them to the baseline, the specifications for each one and the results in DCF and TEM for the *test.tsv* and *test-extended.tsv* are included in Table 4.

After training the models we used them to predict the audios that contain the wake-up word and the times in the audio where it locates.The model has as show in Table 2 some parameters which determine its behaviour. "Time Window" determines the length of the audio segment that the model has to process. The "Threshold" determines the probability that the model has to give to the "WuW" segments to determine if it is "WuW" or "NonWuW", by doing Probability of WuW $>$ Threshold. "N_positives" is the number of time windows where the model has to classify as "WuW" to determine that the audio is a "WuW" instance. "Hop" is the time of the step that determines the start of the time window that the model process, so the audio divides in N segments of length $1.5s$ with a starting time of Hop $\times\, n$ for $n \in \{0, ..., N-1\}$ and the N is determined by the length of the audio using:

$$N = \frac{\text{Audio length} - 1.5}{\text{Hop}}$$

Then for evaluation metrics as the evaluation plan [7] mentions are the Detection Cost Function (DCF) and Time Error Metric (TEM), using the formulas for DCF in the following equation with parameters shown in Table 3.

$$DCF(\theta) = C_{\text{Miss}} \times P_{\text{Miss}}(\theta) \times P_{\text{wuw}} + C_{\text{FA}} \times P_{\text{FA}}(\theta) \times (1 - P_{\text{wuw}})$$

$$P_{Miss} = \frac{\text{Number of misses}}{\text{Number of wake-up word samples}}$$

$$P_{FA} = \frac{\text{Number of misses}}{\text{Number of non-wake-up word samples}}$$

Parameter	Value
P_{wuw}	0.5
C_{Miss}	1
C_{FA}	1.5

Table 3: *DCF parameters*

And TEM in equation 1, with $T_{i_{ref}}$ being the time of reference marked in the *test.tsv* as "Start_Time" or "End_Time", and $T_{i_{pred}}$ the predicted ones.

$$TEM = median([TE(1), TE(2), ..., TE(N)]) \quad (1)$$

$$TE(n) = TE_s(n) + TE_e(n)$$

$$TE_s(n) = |T_{s_{ref}}(n) + T_{s_{pred}}(n)|$$

$$TE_e(n) = |T_{e_{ref}}(n) + T_{e_{pred}}(n)|$$

The model tests were conducted using an NVIDIA GeForce RTX 4060 Laptop GPU, in a Windows 11 OS, a 12^{th} Gen Intel(R) Core(TM) i5-12450H CPU and $16GB$ of RAM. The results, as shown in Table 4, indicate that all the models outperformed the baseline, which is a *LambdaResNet18* model, in nearly every aspect. However, in some cases, the TEM metric results are slightly lower, though the difference is marginal. The execution time is also influenced by the hop length, which explains why the models "LRNMFCC128" and "LRNMFCC128-2" showed slower performance but achieved significantly better results in both the TEM and DCF metrics compared to the other models.

Parmeter	Meaning	Values
Sampling_rate	Sampling rate for each audio	$16000Hz$
Time_Window	Time window covered by every data sample	$1.5s$
Hop	Hop between windows	$0.256s$, $0.128s$
Threshold	Minimum probability for acceptance	0.65
N_positives	Minimum number of windows to detect a WuW event	2

Table 2: *Model and audio parameters*

Name	Hop	Threshold	DCF	DCF extended	TEM	Time	Time ext
baseline	$0.256s$	0.50	0.2767	0.5450	0.4890	$12.414s$	$42.321s$
LRNMFCC256	$0.256s$	0.65	0.1989	0.1294	0.5330	$12.194s$	$41.164s$
LRNMFCC256-2	$0.256s$	0.65	0.1927	0.1794	0.5360	$11.395s$	$42.209s$
LRNMFCC128	$0.128s$	0.65	0.1517	0.2278	0.1900	$21.468s$	$79.590s$
LRNMFCC128-2	$0.128s$	0.65	0.1694	0.2156	0.2820	$25.817s$	$82.190s$

Table 4: *Metrics and information for each model compared in terms of DCF, TEM and time needed to run the standard test and the extended test*

Name	DCF	TEM	minDCF
baseline	0.4026	0.5534	0.3799
LRNMFCC256	0.2734	0.5764	0.2580
LRNMFCC256-2	0.3060	0.5601	0.2915
LRNMFCC128	0.2374	**0.3086**	0.2373
LRNMFCC128-2	**0.2323**	0.3148	**0.2215**

Table 5: *Metrics and results for the Evaluation*

5. Evaluation results

The models proposed in Section 4 were submitted to the ALBAYZIN 2024 Wake-Up Word Detection Challenge, where they were tested using data collected via an online form. This data has conditions similar to those of the training and development sets. Specifically, it consists of long audio files of varying lengths, containing either positive or negative Wake-up Word (WuW) samples, combined with domestic background noises at various Signal-to-Noise Ratio (SNR) levels.

The metrics used to evaluate the results include DCF, TEM, and minDCF (Equation 2), where θ is the threshold explained in Section 4. The minDCF metric is used to find the threshold that minimizes DCF and to check the loss in performance due to imperfect threshold setting.

$$minDCF = \min_{\theta}\{DCF(\theta)\} \tag{2}$$

As shown in Table 5 the baseline *LambdaResNet18* model is outperformed by all the proposed systems in terms of DCF and minDCF. Additionally, both LRNMFCC128 and LRNMFCC128-2 outperform the baseline in TEM and achieve the best results for DCF and minDCF.

6. Conclusion

In this work, we presented several models for wake-up word detection for the ALBAYZIN 2024 Wake-Up Word Detection Challenge, building upon the baseline *LambdaResNet18* model. Through systematic experimentation, we demonstrated that all proposed models were able to outperform the baseline in key metrics, particularly in DCF and sometimes in TEM. The introduction of MFCC features combined with residual Lambda blocks proved to be a highly effective strategy, resulting in significant performance gains despite some trade-offs in execution time due to varying hop lengths. Specifically, our models such as "LRNMFCC128" and "LRNMFCC128-2" showed superior detection capabilities, yielding a maximum of 50% reduction in TEM and a maximum of 40% reduction in DCF.

7. Acknowledgements

This work has been partially funded by project PID2021-125943OB-I00 (Spanish Ministry of Science and Innovation and ERDF). Authors also thank Red Temática en Tecnologías del Habla (RTTH) for their support in ALBAYZIN evaluations and IberSPEECH 2024 organization committee for hosting the ALBAYZIN evaluations.

8. References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[2] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *2015 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2015, pp. 1478–1482.

[3] I. López-Espejo, Z.-H. Tan, J. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 9, pp. 138 617–138 646, 2021.

[4] B. Tura, S. Escuder, F. Diego, C. Segura, and J. Luque, "Efficient keyword spotting by capturing long-range interactions with temporal lambda networks," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6797–6801.

[5] I. Bello, "Lambdanetworks: Modeling long-range interactions without attention," *CoRR*, vol. abs/2102.08602, 2021. [Online]. Available: https://arxiv.org/abs/2102.08602

[6] R. Tang and J. Lin, "Deep residual learning for small footprint keyword spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5484–5488.

[7] F. López and J. Luque, "Albayzin evaluation 2024: Wake-up word detection challenge," Telefónica Innovación Digital and Universidad Autónoma de Madrid, September 2024.

[8] PyTorch, "Audio data augmentation tutorial," https://pytorch.org/audio/stable/tutorials/audio_data_augmentation_tutorial.html, 2024.

[9] C. Richey, M. A. Barrios, Z. Armstrong, C. Bartels, H. Franco, M. Graciarena, A. Lawson, M. K. Nandwana, A. Stauffer, J. van Hout, P. Gamble, J. Hetherly, C. Stephenson, and K. Ni, "Voices obscured in complex environmental settings (voices) corpus," 2018. [Online]. Available: https://iqtlabs.github.io/voices/