On creating small datasets for training embedded acoustic scene classification systems via time-frequency segmentation

Douglas Baptista de Souza, Janderson Ferreira, Fernanda Ferreira, and Michel Meneses

***Abstract*— Acoustic Scene Classification (ASC) systems have great potential to transform existing embedded technologies. However, research on ASC has put little emphasis on solving the existing challenges in embedding ASC systems. In this paper, we focus on one of the problems associated with smaller ASC models: the generation of smaller yet highly informative training datasets. To achieve this goal, we propose to employ the so-called multitaper-reassignment technique to generate high-resolution spectrograms from audio signals. These sharp time-frequency (TF) representations are used as inputs to a splitting method based on TF-related entropy metrics. We show via simulations that the datasets created through the proposed segmentation can successfully be used to train small convolutional neural networks (CNNs), which could be employed in embedded ASC applications.**

***Keywords*— Acoustic Scene Classification, Audio Segmentation, Multitaper-reassigned Spectrogram, Time-frequency Entropies**

I. Introduction

In the last few years, we have seen a growing interest in context-aware technologies [1]. The goal is to use the acquired knowledge on the acoustic scene to build or improve devices such as hearing aids, smartphones, and Internet of Things (IoT) systems. Acoustic scene classification (ASC) refers to the task of classifying recorded audio clips into pre-defined categories, which allows technologies to recognize the environment based on sound captured from the surroundings [1], [2]. The research on ASC models has historically relied on very large machine learning architectures and datasets [1], [3]. For example, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge and workshop for ASC and audio event detection (AED) [4], only included the task for low-complexity ASC systems in 2020 [5]. The fact the research on small ASC models has not gained much attention up to until recently, contrasts with the growing interest in embedded models for speech and audio recognition [6], [7], [8].

One of the problems preventing small ASC systems from taking off is the almost lack of benchmarking studies for low-complexity settings. In these applications, the strict requirements for energy consumption, memory usage, latency and data storage play a crucial role in the feasibility of embedding emerging technologies [9], [10]. In this regard, the research on speech recognition systems is considerably ahead of the one in ASC, as deep-learning models [8], [9], learning frameworks [11], and datasets [12] that make the embedding of these models possible have been already proposed years ago. By taking as reference the path followed by the speech recognition research, and considering current trends in embedded machine learning models, one could identify some topics that should be addressed in order to make small ASC systems viable. Examples are the research on i) frameworks such as [6] and [13] for on-device training of deep neural networks (DNNs), ii) Convolutional neural networks (CNNs) small enough to be embedded [8], [9], iii) segmentation techniques to reduce the size of existing datasets to allow more efficient model training and evaluation [14], be it for small models [15], or few-shot learning frameworks [16]. Although topic i) is important in applications in which online training is not always possible (e.g., smartphones and domestic assistants which need to work even in the absence of internet connection), one could argue that topics ii) and iii) necessarily impact the feasibility of i). Thus, in this paper, we focus on the second and third topics.

More specifically, we propose a segmentation method for extracting short segments from long ASC audio clips. In embedded applications, the input audio duration is often limited to a few seconds by the size of the memory buffer. On the other hand, public datasets available for training ASC systems like DCASE are commonly formed by much longer audio records [3]. In ASC frameworks that happen to perform a segmentation step, the split is often a simple random cropping of the long audio record, giving thousands of randomly-cut segments with very little information aggregated.

The segmentation method presented here consists of applying the so-called multitaper-reassignment technique [17], [18] to create sharp, high-resolution spectrograms from long audio clips. These spectrograms are then used as source for the application of a cutting criterion based on entropy-like metrics for time-frequency (TF) representations, such as the Rényi entropy [19] and the Jones-Park concentration metric [20]. The idea of the proposed technique is that a much smaller database to train ASC systems can be generated by carefully selecting the most reliable segments from audio clips. We show via simulations that the short audio segments extracted by using the proposed method allow for training models with much less parameters in comparison to common architectures in the ASC literature. The observed classification performances in the experimental study are close to the ones usually obtained by using very large ASC datasets and models.

The rest of the paper is organized as follows. Section II discusses background elements involving the multitaper-

Douglas Baptista de Souza, Janderson Ferreira, Fernanda Ferreira, and Michel Meneses, SiDi, Rua Aguaçú, 171, prédio Jacarandá Loteamento Alphaville Campinas 13098321 - Campinas, SP - Brasil, e-mail: {d.batista, janderson.f, fernanda.f, m.meneses}@sidi.org.br.

reassignment technique, as well as existing TF-based entropy metrics. The proposed approach for segmenting long ASC audios is described in Section III. The experimental study and conclusions are shown in Sections IV and V, respectively.

II. Background Elements

A. The multitaper reassigned spectrogram

One of the key steps of any audio analysis framework is choosing a proper space to represent the signals. Often, TF transformations are employed to map the audio signals from time to TF domain. One of the most well-known TF transformation is the spectrogram

$$S_x^{(h)}(t,f) = \left|F_x^{(h)}(t,f)\right|^2 = \left|\int x(\tau)h(\tau-t)\mathrm{e}^{-j2\pi fs}d\tau\right|^2 \quad (1)$$

which is the squared magnitude of $F_x^{(h)}(t,f)$, the short-time Fourier transform (STFT) of the signal $x(t)$ computed by using window function $h(t)$ [21]. In speech recognition applications, the frequency axis of the spectrogram is often transformed to the log-mel scale. However, in applications of signal analysis and automatic segmentation, choosing adequate TF transformations (e.g., see [25]), or spectrogram enhancement methods (e.g., see [17]) are more common than using one particular frequency scaling method such as log mel.

Two powerful techniques for enhancing spectrograms are the multitapering and the reassignment methods [22], [23]. To understand their advantages, one has to approach the problem of computing spectrograms from a statistical standpoint. Consider the spectrogram (1) as an estimator for the Wigner-Ville Spectrum (WVS) [21], a stochastic quantity representing a candidate time-varying spectrum for a given (potentially nonstationary) input signal $x(t)$

$$W_x(t,f) = \int_{-\infty}^{\infty} \mathbb{E}\left[x\left(t+\frac{\tau}{2}\right)x^*\left(t-\frac{\tau}{2}\right)\right]\mathrm{e}^{-j2\pi\tau f}d\tau \quad (2)$$

where $\mathbb{E}[\cdot]$ is expectation operator. The quality of the spectrogram estimator could be assessed, for example, by its bias and variance. The multitapering approach allows to reduce the variance in the estimation of (2) while controlling the bias [18]. The idea is to use not one, but $k = 1, ..., K$ orthonormal windows $h_k(t)$ to compute (1), and then average the resulting collection of spectrograms to smooth out the fluctuations inherent to the estimation procedure, i.e.,

$$\widehat{S}_K^{(h_k)}(t,f) = \frac{1}{K}\sum_{k=1}^{K} S_x^{(h_k)}(t,f). \quad (3)$$

Popular choices for the window functions $\{h_k(t), k = 1, ..., K\}$ in (3) are the orthonormal Hermite polynomials and the Discrete Prolate Spheroidal Sequences (DPSS) [17].

The reassignment technique addresses the limitation of the spectrogram to localize the theoretical values of (2) at a given TF coordinate. Without entering into too much details, it can be shown that the spectrogram cannot estimate the values of $W_x(t,f)$ in a pointwise manner, meaning that the value of (1) at a given (t,f) point is actually composed by the contribution of WVS estimates over a neighbouring TF window. Hence, the spectrogram values stand for blurred estimates of (2). The reassignment method can improve such blurriness by replacing the values of the spectrogram w.r.t. a given TF window by the value of its center of mass, whose coordinates are given by

$$\begin{aligned}\hat{t} &= t + \mathrm{Re}[F_x^{(g)}(t,f)/F_x^{(h)}(t,f)] \\ \hat{f} &= f - \mathrm{Im}[F_x^{(m)}(t,f)/F_x^{(h)}(t,f)]\end{aligned} \quad (4)$$

where $F_x^{(g)}(t,f)$ and $F_x^{(m)}(t,f)$ are STFTs computed by using windows $g(t) = th(t)$ and $m(t) = \partial h(t)/dt$ instead of the usual $h(t)$ as in (1) [23]. With (4) at hand, the reassigned spectrogram can be estimated by means of

$$\widehat{\mathrm{RS}}_x^{(h)}(t,f) = \iint S_x^{(h)}(\tau,s)\delta(t-\hat{t}_{\tau,s})\delta(f-\hat{f}_{\tau,s})d\tau ds. \quad (5)$$

In general, multitapering and the reassignment have been employed as separate, individual techniques to improve the spectrogram computation. In [17], however, the authors fused both methods into the multitaper-reassigned spectrogram, which can be obtained by combining (3) and (5) as follows:

$$\widehat{S}_K^{(h_k)}(t,f) = \frac{1}{K}\sum_{k=1}^{K}\widehat{\mathrm{RS}}_x^{(h_k)}(t,f). \quad (6)$$

Despite of its advantages, the multitaper-reassigned spectrogram has not yet been fully explored in signal analysis and segmentation. This observation is particularly true in audio applications. An example of the ability of the multapering-reassignment technique to improve the spectrogram estimation is shown in Fig. 1. In this figure, a 10-second audio clip from the DCASE 2020 challenge database [5] is analyzed through the traditional [Fig. 1 (a)] and the multitaper-reassigned [Fig. 1 (b)] spectrograms, computed by means of (1) and (6), respectively. Notice the low-energy nonstationary background is completely suppressed in the traditional spectrogram. Also, the low-frequency, slowly-nonstationary fluctuations, as well as the harmonic structure between the time marks of 6 and 7.5 seconds can be better spotted in Fig. 1 (b).

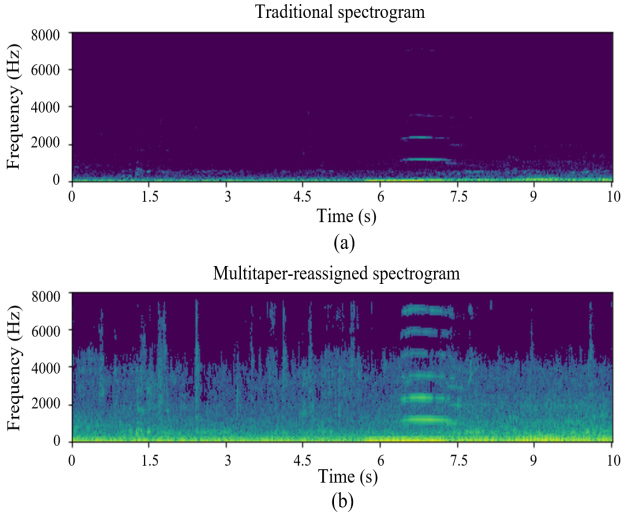

Fig. 1. Examples of spectrograms computed by means of (1) and (6). The original audio signal is a 10-second long recording made inside a bus.

B. Measuring the concentration of time-frequency frames

Thanks to the analogy between probability and sinal energy densities [21], estimates of the WVS such as multitaper-reassigned spectrograms can be interpreted (to a certain extent) as approximate probability distributions in two dimensions. Thus, according to this interpretation, more concentrated (peaky) spectrograms would correspond to smaller entropy values. The concentration in TF domain has been used as quality measure for TF representations by many authors [17], [24], [25]. The idea being that highly-concentrated representations are normally associated with the case in which the signal energy is less spread over a smaller TF region, meaning that estimation and classification tasks based on highly-concentrated TF representations can be more reliable [19].

Different criteria have been proposed to measure the concentration of TF representations [24]. Here, we make use of two well-known TF concentration metrics: the Jones-Park ratio [20] and the Rényi entropy [19]. For the multitaper-reassigned spectrogram of (6), the former metric is given as

$$R_{\text{JP}} = \iint \left[\hat{S}_K^{(h_k)}(t,f)\right]^2 dtdf \Big/ \left[\iint \hat{S}_K^{(h_k)}(t,f)dtdf\right]^2 \tag{7}$$

and evaluates the spread of (6) over a TF region. The more concentrated (6), the larger the value of (7). Conversely, entropy-like measures such as the Rényi entropy of order α

$$R_{\text{RN}} = \frac{1}{1-\alpha}\log\left\{\iint \left[\hat{S}_K^{(h_k)}(t,f)\right]^{\alpha} dtdf\right\} \tag{8}$$

take smaller values for more concentrated cases, which correspond to a scenario with less TF entropy. Note that (8) is parametrized by α. In the experimental study of Section IV, we explore Rényi entropy of two different orders.

III. Proposed segmentation scheme

Having determined which TF transformation to use and the criterion to evaluate the concentration of the TF frames, the proposed method to segment long ASC clips is the following:

1) For a given long ASC audio signal, compute its multitaper-reassigned spectrogram with (6) for all of its time extent (i.e., the spectrogram analysis should stretch over all duration T of the original audio).
2) Choose a metric to evaluate the concentration in TF domain of the windowed frames (e.g., Jones-Park ratio).
3) Choose a window of duration $\mathcal{T} = t_f - t_i$, with t_i and t_f being the start and end points of the audio segment.
4) Evaluate the concentration of the TF block using the metric chosen in step 2 and store its value.
5) Considering a time step of Δt, sweep the window of analysis over time to get another frame of the original multitaper-reassigned spectrogram.
6) Based on the stored metric values, select the window segment giving the largest concentration in TF domain and get its corresponding t_i and t_f.
7) Use t_i and t_f to cut a segment of duration $\mathcal{T}$ from the original audio.

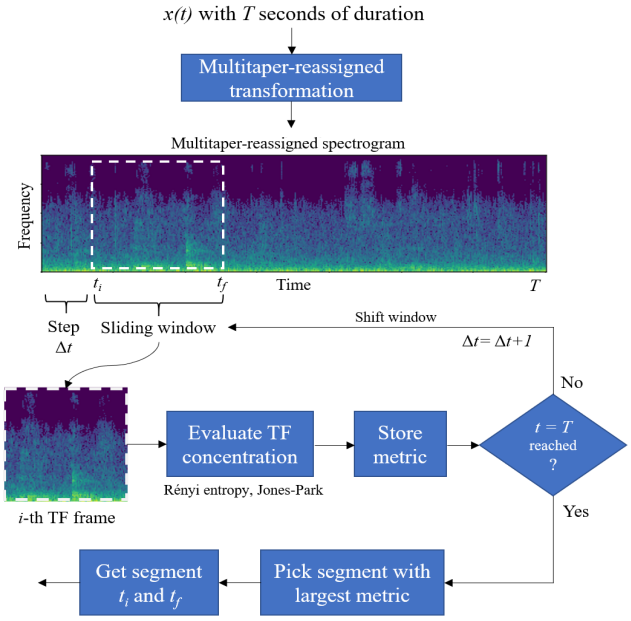

Fig. 2. Proposed scheme for segmenting long ASC audio clips by using multitaper-reassigned spectrogram and metrics to evaluate TF concentration.

The scheme above is also depicted in Fig. 2. The experiment on applying the proposed segmentation to an ASC system prone to be embedded is shown in the next section.

IV. Experimental study

A. On the chosen dataset

The ASC audio database from the DCASE 2020 challenge (Task 1A) has been considered in this experiment [5]. More precisely, this dataset is used for the task of classifying long audio scene samples (with ten seconds of duration), which correspond to audio clips recorded in different locations (cities) around the world. In this work, aiming at making the experiment better adapted to the average Brazilian urban areas[1], it has been determined that some classes originally seen in the dataset such as "tram", "airport", and "metro station", for example, would not be considered. These audio scenes are present only in a small portion of the Brazilian cities. After filtering out the unwanted labels, we have arrived at the following four scenes (labels y's) for the classification task: $y = \{$"Bus", "Park", "Street traffic", and "Shopping mall"$\}$. Using a reduced number of labels also facilitates the task of training smaller, low-complexity machine learning models.

The DCASE 2020 audio files are divided into two subgroups: the development (train) and the evaluation (test) datasets, but audio labels are provided only for the former. Thus, researchers participating in the DCASE 2020 challenge can train their models in the development dataset, but the evaluation samples can be used only for label prediction. The DCASE committee does not provide the ground truth labels for the evaluation dataset (not without proper, writing request), but suggests that researchers can arbitrarily divide the >14k samples from the development dataset into train, validation,

[1]This is a consideration defined *a priori* for this research work.

TABLE I
NUMBER OF AUDIO SAMPLES CONSIDERED FOR EACH LABEL

Partition	Bus	Shopping mall	Park	Street traffic	Total samples
Train	964	1089	1029	796	3878
Validation	20	99	36	36	191
Test	456	252	375	609	1692

and test subsets, as long as audio scenes recorded from same locations are kept within the same partition. Thus, following the proposed scheme, we have partitioned the DCAS E2020 development dataset as shown in Table I.

B. Experiment design choices

Different options can be considered to design an experiment to evaluate the applicability of the proposed segmentation scheme to embedded ASC applications. We list below the requirements guiding the experiment of this section.

- We adopt the criterion of [8] and [9] and limit the number of model parameters to less than 250k.
- Some machine learning models need to be trained on device in an offline manner (e.g., see [7]). To account for such a possibility and to avoid a prohibitive training time, we only train the chosen DNN model for 10 epochs.
- Although the DCASE 2020 audio dataset for the considered task is sampled at 48 kHz, we have resampled it to 16 kHz, which is a sample rate better suited to low-complexity audio recognizers (e.g., see [10]).
- To account for the maximum size of audio buffer available in common microprocessors, we considered audio clips of $\mathcal{T} = 2$ seconds of duration as input to the system. Such a duration is a common choice in embedded audio applications, since it allows buffering short utterances like in [26], or longer ones as in [27].

C. On the chosen model

Taking into account the points above, the segmentation task can thus be seen as the one of finding candidate 2-second clips within the 10-second audio samples, by employing the segmentation method described in Section III. The chosen model is the CNN developed by Tensorflow/Google Research teams (see [8]), and made freely available in [28] as a part of a tutorial to create a simple spoken keyword recognizer [12]. Despite of being a relatively small architecture in comparison to other CNNs used for speech recogntion, the model of [28] contains more than the maximum number of 250k parameters, as defined above. To ensure the model meets the size requirement, we reduced the size of its last dense layer from 128 to 32 neurons. Moreover, we removed the two dropouts layers presented in the original architecture, as using dropouts in training sessions restricted to few epochs and a small dataset can affect the learning performance. By carrying out these modifications, the size of the CNN could be reduced from about 1.6M to less than 220k parameters. An overview of the obtained CNN model is shown in Table II. For more details about the original architecture and its layers, see [8] and [28].

TABLE II
ARCHITECTURE AND NUMBER OF PARAMETERS IN EACH LAYER

Layer (type)	Output Shape	Param #
resizing_45 (Resizing)	(None, 32, 32, 1)	0
normalization_46 (Normalization)	(None, 32, 32, 1)	3
conv2d_92 (Conv2D)	(None, 30, 30, 64)	640
conv2d_93 (Conv2D)	(None, 28, 28, 32)	18264
max_pooling2d_46 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten_46 (Flatten)	(None, 6272)	0
dense_92 (Dense)	(None, 32)	200736
dense_93 (Dense)	(None, 4)	132
Total params: 219975		

D. Choosing features and segmentation setups

To segment the 2-second clips from the selected audios, the following methods have been considered (see Fig. 2):
a) proposed segmentation with Rényi entropy (8) and $\alpha = 5$,
b) proposed segmentation with Rényi entropy (8) and $\alpha = 3$,
c) proposed segmentation with Jones-Park metric (7),
d) search for the highest-energy segment in time domain,
e) crop of random segments along the audio.

The values of α chosen for the segmentation schemes a) and b) are commonly used in the literature for computing the Rényi entropy [18], [19], [17]. The method d) stands for simply sweeping over the extent of a given audio signal $x(t)$ in time and evaluating the most energetic 2-second segment, i.e., finding t_f and t_i maximizing $\int_{t_i}^{t_f} |x(t)|^2 dt$, given that $t_f - t_i = 2$ seconds. Finally, the method e) stands for simply cutting random 2-second segments from the longer audio clips, and is one of the most used cropping techniques in the ASC literature (e.g., see [4]). The input features for the CNN have been considered as the spectrograms (1) computed by employing $h(t)$ as Hanning window, making use of 512 bins for computing the FFT, and 256 points for the stride and the window length. Note that, although the multitaper-reassigned transformation was selected to generate the representations employed in the segmentation step, simple spectrograms have been chosen as model input features so the overall framework could meet the desired low-complexity and latency constraints.

E. Training and testing performances

The CNN model has been trained and tested five times considering the dataset described in Section IV-A and the segmentation methods a) to e). The results are given as average accuracy scores in Table III. Note that these scores actually stand for the per-class true-positive rates (TPRs) (see [29]).

In Table III, it can be seen that the best classification results are for the methods based on the Rényi entropy ($\alpha = 5$ and 3), while the worst performance is given by the random segmentation, which is the cropping scheme usually employed in the literature (e.g., see [4]). It should be remarked that the obtained per-class classification performances are in certain cases better or close to those obtained by using much bigger DNN models[2], usually with few million parameters and large training databases generated by using a very larger number

[2]Note that the focus of this work is to improve segmentation and dataset creation for ASC, not to beat state-of-the-art classification performances.

of clips randomly cut from audio. Here, we used only one (carefully segmented) clip per long audio record.

To see the obtained results from another perspective consider, for example, the reported performances of [30] and [31], which are models with 4.31M and 3.2M parameters, respectively. In [31], the recognition rates for Bus, Shopping mall, Park, and Street traffic have been 86.9%, 65.7%, 89.9%, and 89.6%, respectively. Although models such as [30] and [31] have been trained and tested using the full (i.e., ten classes) ASC database (thus making the classification task more difficult), the performance obtained here under the restrictive implementation requirements defined in Section IV-B are very satisfactory for the considered experiment.

TABLE III

AVERAGE CLASSIFICATION ACCURACIES (OR TPRS) FOR THE CONSIDERED SEGMENTATION METHODS AND CLASSES

Segmentation method	Bus	Shopping mall	Park	Street traffic	Avg. all classes
a) Renyi $\alpha = 5$	87%	95%	77%	82%	86%
b) Renyi $\alpha = 3$	95%	92%	67%	83%	84%
c) Jones-Park	85%	98%	69%	81%	82%
d) Energy cut	74%	94%	67%	88%	80%
e) Random cut	87%	97%	65%	75%	79%

V. CONCLUSIONS

In this paper, a segmentation method to build small datasets for training low-complexity acoustic scene classification (ASC) systems was proposed. The rationale behind the method was to employ the multitapering-reassignment technique to generate sharp spectrograms, from where a criterion based on spectrogram concentration was adopted for selecting candidate audio segments. Two metrics for assessing spectrogram concentration were considered: Rényi entropy and Jones-Park ratio. To evaluate the proposed method, we designed an experiment in which a convolutional neural network (CNN) was adapted to an embedded ASC task scenario. The obtained classification performance of audio scene samples when using the proposed segmentation schemes for generating training samples outperformed the competing methods considered.

REFERENCES

[1] J. Abeßer, "A review of deep learning based methods for acoustic scene classification," Appl. Sci., vol. 10, no. 6, pp. 1-16, Mar. 2020.

[2] A. M. Basbug and M. Sert, "Analysis of Deep Neural Network Models for Acoustic Scene Classification," in Proc. Signal Process. Commun. Appl. Conf., Sivas, Turkey, Apr. 2019, pp. 1-4.

[3] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification and domestic audio tagging," in Proc. DCASE 2016, Budapest, Hungary, Sep. 2016, pp. 1-5.

[4] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," IEEE Trans. Multimed., vol. 17, no. 10, pp. 1733-1746, Oct. 2015.

[5] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in Proc. DCASE 2020, Tokyo, Japan, Nov. 2020, pp. 1-5.

[6] S. Bouguezzi, H. Faiedh and C. Souani, "Slim MobileNet: An Enhanced Deep Convolutional Neural Network," in Proc. IEEE SSD Int. Multi-Conference on Systems, Signals and Devices, Monastir, Tunisia, Mar. 2021, pp. 12-16.

[7] A. Parnami and M. Lee, "Few-shot keyword spotting with prototypical networks," arXiv preprint arXiv:2007.14463, 2020.

[8] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in Proc. INTERSPEECH, Dresden, Germany, Sep. 2015, pp. 1478-1482.

[9] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," arXiv preprint arXiv:1703.05390, 2017.

[10] A. Sehgal and N. Kehtarnavaz, "A Convolutional Neural Network Smartphone App for Real-Time Voice Activity Detection," IEEE Access, vol. 6, pp. 9017-9026, Feb. 2018.

[11] A. Rayaluru, S. R. Bandela and K. K. T, "Speech Emotion Recognition Using Feature Selection with Adaptive Structure Learning," in Proc. IEEE Int. Symp. on Smart Electronic Systems, Rourkela, India, Dec. 2019, pp. 233-236.

[12] P. Warden, "Speech commands: a dataset for limited-vocabulary speech recognition," arXiv: 1804.03209 [cs.CL], vol. 1, pp. 1–11, Apr. 2018.

[13] M. Mitra, "Neural processor in artificial intelligence advancement," J. Auton. Intell., vol. 1, no. 1, May 2018, pp. 2-14.

[14] M. K. Nammous, K. Saeed, and P. Kobojek, "Using a small amount of text-independent speech data for a BiLSTM large-scale speaker identification approach," J. King Saud Univ. Sci., Apr. 2020, pp. 1-7.

[15] T. Hirvonen, "Speech/Music Classification of Short Audio Segments," in Proc. IEEE Int. Symp. Multimed., Chengdu, China, Jul. 2014, pp. 135-138.

[16] B. Shi et al., "Few-Shot Acoustic Event Detection Via Meta Learning," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Barcelona, Spain, May 2020, pp. 76-80.

[17] J. Xiao and P. Flandrin, "Multitaper Time-Frequency Reassignment for Nonstationary Spectrum Estimation and Chirp Enhancement," IEEE Trans. Signal Process., vol. 55, no. 6, pp. 2851-2860, June 2007.

[18] J. Xiao, "Contributions to nonstationary spectrum estimation and stationarity tests in the time-frequency plane," Ecole Normale Supérieure de Lyon, 2008. [Online]. Available: `https://tel.archivesouvertes.fr/tel-02167626/document`.

[19] R. G. Baraniuk, P. Flandrin, A. J. E. M. Janssen, and O. J. J. Michel, "Measuring time-frequency information content using the Renyi entropies," IEEE Trans. Inf. Theory, vol. 47, no. 4, pp. 1391-1409, May 2001.

[20] D. Jones and T. Parks, "A high resolution data-adaptive time-frequency representation," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Dalas, Texas, Apr. 1987, pp. 681-684.

[21] P. Flandrin,*Time-Frequency/Time-Scale Anaylsis*. San Diego, CA, USA: Academic, 1999.

[22] R. D. Martin and D. J. Thomson, "Robust-resistant spectral estimation," Proc. IEEE, vol. 70, no. 1, pp. 1097-1114, Sep. 1982.

[23] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," IEEE Trans. Signal Process., vol. 43, no. 5, pp. 1068-1089, May 1995.

[24] I. Shafi, J. Ahmad, S. I. Shah, and F. M. Kashif, "Techniques to Obtain Good Resolution and Concentrated Time-Frequency Distributions: A Review," EURASIP J. Adv. Signal Process., vol. 2009, no. 673539, pp. 1-43, Jun. 2009.

[25] M. Dakovic, L. Stankovic, and T. Thayaparan, *Time-Frequency Signal Analysis with Applications*, 1st ed. Boston, MA, USA: Artech House, 2013.

[26] S. Choi et al., "Temporal convolution for real-time keyword spotting on mobile devices," in Proc. INTERSPEECH, Graz, Austria, Sep. 2019, pp. 3372–3376.

[27] G. Chen, C. Parada and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Florence, Italy, May 2014, pp. 4087-4091.

[28] Tensorflow, "Simple audio recognition: Recognizing keywords," [Online]. Available: `https://www.tensorflow.org/tutorials/audio/simple_audio`, Accessed on: May 29, 2021.

[29] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manag., vol 45, no. 4, pp. 427-437, Jul. 2009.

[30] W. Gao and M. McDonnell, "Acoustic scene classification using deep residual networks with focal loss and mild domain adaptation," in Proc. DCASE 2020, Tokyo, Japan, Nov. 2020, pp. 1-2.

[31] J. Liu, "Acoustic scene classification with residual networks and attention," in Proc. DCASE 2020, Tokyo, Japan, Nov. 2020, pp. 1-3.