A MULTI-RESOLUTION APPROACH TO SOUND EVENT DETECTION IN DCASE 2020 TASK4

Diego de Benito-Gorron, Daniel Ramos, Doroteo T. Toledano

AUDIAS Research Group
Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente, 11, 28049 Madrid, SPAIN
{diego.benito, daniel.ramos, doroteo.torre}@uam.es

ABSTRACT

In this paper, we propose a multi-resolution analysis for feature extraction in Sound Event Detection. Because of the specific temporal and spectral characteristics of the different acoustic events, we hypothesize that different time-frequency resolutions can be more appropriate to locate each sound category. We carry out our experiments using the DESED dataset in the context of the DCASE 2020 Task 4 challenge, where the combination of up to five different time-frequency resolutions via model fusion is able to outperform the baseline results. In addition, we propose class-specific thresholds for the F_1-score metric, further improving the results over the Validation and Public Evaluation sets.

Index Terms— DCASE 2020 Task 4, CRNN, Mean Teacher, Multi-resolution, Model fusion, Threshold tuning, PSDS

1. INTRODUCTION

Sound Event Detection (SED) systems aim to determine the temporal locations of several categories of acoustic events in a given audio clip. In contrast with the usual single-resolution approach used to train these systems, we propose a multi-resolution analysis of the audio features (mel-spectrograms) in order to take advantage of the diverse temporal and spectral characteristics found in different sound events.

Our experiments are based on the DCASE 2020 Task 4 baseline, which consists of a convolutional recurrent neural network (CRNN) trained using the Mean Teacher algorithm [1]. Additionally, class-specific thresholds for the F_1-score metric [2] are proposed, replacing the default global value of 0.5.

2. DATASET

The dataset used for Sound Event Detection in DCASE 2020 Task 4 is DESED (Domestic Environment Sound Event Detection) [3, 4], which is composed of real and synthetic recordings. Real recordings include the Weakly-labeled training set (1578 clips), the Unlabeled training set (14412 clips), the Validation set (1168 clips) and the Public Evaluation set (692 clips). Synthetic recordings have been generated using the Scaper library [5] and the provided JAMS file, obtaining a Synthetic training set with 2536 strongly-labeled clips.

Work developed under project DSForSec (RTI2018-098091-B-I00), funded by the Ministry of Science, Innovation and Universities of Spain and FEDER

Event	N.	Mean	Std.
Alarm bell / ringing	587	1.10	1.43
Blender	370	2.36	2.04
Cat	731	1.11	0.81
Dishes	1123	0.61	0.49
Dog	824	0.92	0.93
Electric shaver / toothbrush	345	4.61	2.69
Frying	229	5.06	3.07
Running water	270	3.81	2.53
Speech	2760	1.13	0.82
Vacuum cleaner	343	5.87	3.28

Table 1: Number of examples and mean and standard deviation of their durations (in seconds) for each sound category in the Synthetic training set.

The Weakly-labeled, Unlabeled and Synthetic training sets are used to train the neural networks. 20% of the Synthetic training set is reserved for validation. The DESED Validation set is used to tune hyper-parameters and perform model selection. In addition, we provide results over the Public Evaluation set.

3. PROPOSED SOLUTIONS

3.1. Multi-resolution analysis

The DCASE 2020 Task 4 challenge consists in the detection and classification of 10 different sound events that differ in duration and spectral characteristics. Based on these differences we hypothesize that a multi-resolution feature extraction approach could provide improvements in the detection of at least some of these sound events. One of the goals of our participation in the DCASE 2020 Task 4 challenge is to empirically validate this hypothesis.

The baseline system provided by the organizers of this challenge, as well as most systems developed by participants in previous similar evaluations, rely on a mel-spectrogram which transforms the audio recording to process into a 2-D image that is taken as the input of a deep neural network. The mel-spectrogram computation can be adjusted based on a few parameters: the audio sampling frequency, the size of the FFT, the analysis window type and length and the number of Mel filters used. A particular set of these parameters define a single time-frequency resolution working point in the spectral analysis.

A time-frequency resolution working point can be better or worse suited to detect a specific type of sound event depending on

the characteristics of that sound, more specifically depending on its temporal and spectral characteristics. It is easy to show that the different sounds have different lengths. Table 1 shows the mean and standard deviation of the duration of the 10 different types of sounds in the Synthetic training set, and notable differences are evident. Our hypothesis is that differences similar to the ample difference in lengths may also exist in terms of temporal and spectral characteristics of the different types of events.

Since the types of sound events in the challenge are so different, it seems likely that the use of several time-frequency resolutions at feature extraction could improve sound detection and classification results. Our group applied a similar multi-resolution approach [6] to a different problem (automatic speech recognition) where the differences in the types of sounds (human phones) were much smaller, achieving modest but consistent improvements.

To test our hypothesis we have computed up to five mel-spectrogram features using different spectral analysis parameters, so that we have up to five different time-frequency resolution working points. Our system is based on the baseline provided by the organizers of the challenge. Essentially, we have replicated the baseline system several times, but modifying each instance to work with a different time-frequency resolution working point. All these instances are finally fused at the frame-level, by combining the frame-level estimation of the class posteriors provided as output by each subsystem.

We have defined the five time-frequency resolution working points taking as reference the point defined by the baseline system. We have defined other four points by increasing and decreasing the time and frequency resolution. The five time-frequency resolution working points used share in common with the baseline the use of a sampling frequency of $f_s = 16000$ Hz. and the use of a Hamming window. The rest of the parameters (FFT size, window length, window hop and number of Mel filters) are modified to increase time or frequency resolution as described below and in Table 2 for each of the time-frequency resolution working points used.

1. ***BS*** (Baseline). The baseline uses an analysis window of length $L = 128$ ms and a window hop of $R = 15.94$ ms (255 samples). Both parameters are related to the temporal resolution of the analysis. On the other hand, the frequency resolution is limited by the width of the main lobe of the Hamming window, $8\pi/(L-1) = 8\pi/2047$ rad/sample, which corresponds to a frequency resolution of $4/2047 \times 16000 \approx 31$ Hz. However, this frequency resolution is later more limited in a non-linear way by the use of the Mel filterbank with 128 filters.
2. ***T++*** (Twice better time resolution). We halve the analysis window to a length of $L = 64$ ms and the window hop to $R = 8$ ms, which essentially doubles the time resolution. We also halve the number of Mel filters, which along with the previous changes roughly halves the frequency resolution.
3. ***F++*** (Twice better frequency resolution). We double the analysis window length to $L = 256$ ms and the window hop to $R = 32$ ms, which essentially halves the time resolution. We also double the number of Mel filters, which along with the previous changes roughly doubles the frequency resolution.
4. ***T+*** (Intermediate point between ***BS*** and ***T++***). Analysis window of length $L = 96$ ms, window hop $R = 12$ ms. An intermediate number of Mel filters is used ($n_{mel} = 96$). In this case and in the next one, we have taken the FFT length (N) as the smallest power of 2 greater that L, but we do not expect differences if N is set to L as in the previous cases.
5. ***F+*** (Intermediate point between ***BS*** and ***F++***). Analysis window of length $L = 192$ ms, window hop $R = 24$ ms. An intermediate number of Mel filters is used ($n_{mel} = 192$).

	N	**L**	**R**	$\mathbf{n_{mel}}$
$\mathbf{T_{++}}$	1024	1024	128	64
$\mathbf{T_{+}}$	2048	1536	192	96
BS	2048	2048	255	128
$\mathbf{F_{+}}$	4096	3072	384	192
$\mathbf{F_{++}}$	4096	4096	512	256

Table 2: FFT length (N), window length (L), window hop (R) and number of Mel filters of the five proposed time-frequency resolution working points. N, L and R are reported in samples, using a sample rate $f_s = 16000Hz$.

3.2. Model fusion

Fusion has been performed considering that, for each event, a two-class classification task is performed independently of the other events. Thus, for a given event i, classification between classes $\{\theta_{i,0}; \theta_{i,1}\}$ is performed, where $\theta_{i,0}$ means "event *i* not detected" and $\theta_{i,1}$ means "event *i* detected". Alternatively, we will call this two-class classification task a *detection* task.

For each detection task i, with classes $\{\theta_{i,0}, \theta_{i,1}\}$, a different *score* is generated by each of the CRNN detectors involved, as a time series with a given time resolution. Thus, a final score s_i must be computed for each event in this unit of time, in order to make decisions, by means of the fusion of all the individual scores from all the individual detectors, namely $(s_i^{(1)}, ..., s_i^{(K)})$. We perform this fusion as a late integration, before score binarization and median filtering. By convention, the lower a score, the stronger the support to $\theta_{i,0}$, and the higher a score, the stronger the support to $\theta_{i,1}$. If we have K different detectors, the final score is obtained as the average of the scores in this way:

$$s_i = \frac{1}{K}\sum_{j=1}^{K} s_i^{(j)} \tag{1}$$

The interpretation of the scores of each of the detectors is as follows. Each of the scores is taken from the output of one of the detectors, a CRNN trained with a cross-entropy criterion. Therefore, the output of the jth CRNN can be interpreted as two probabilities, namely $P^{(j)}(\theta_{i,1}|x)$ and $P^{(j)}(\theta_{i,0}|x) = 1 - P^{(j)}(\theta_{i,1}|x)$, where x is the audio observation at this particular moment in time. In the case that the detectors have different output frame rates, each output is interpolated along the time dimension to fit the highest frame rate. Then, we compute each of the scores of the detectors in the following way:

$$s_i^{(j)} = \text{logit}(P(\theta_{i,1}|x)) \equiv \log\frac{P^{(j)}(\theta_{i,1}|x)}{1 - P^{(j)}(\theta_{i,1}|x)} \tag{2}$$

The inverse of the logit operator is the well-known sigmoid function.

Moreover, $\text{logit}(P^{(j)}(\theta_{i,1}|x))$ is decomposed as follows:

$$\text{logit}(P(\theta_{i,1}|x)) = \text{logit}(P(\theta_{i,1})) + \log\frac{P^{(j)}(x|\theta_{i,1})}{P^{(j)}(x|\theta_{i,0})} \tag{3}$$

	$\mathbf{T}_{++}$	$\mathbf{T}_{+}$	**BS**	$\mathbf{F}_{+}$	$\mathbf{F}_{++}$
Alarm bell / ringing	42.1 ± 1.5	**43.8** ± 2.1	42.0 ± 1.4	42.2 ± 3.1	41.0 ± 2.0
Blender	**32.9** ± 3.2	32.3 ± 1.4	27.4 ± 1.6	30.0 ± 2.6	30.9 ± 3.9
Cat	38.4 ± 1.8	40.0 ± 1.8	**41.0** ± 2.1	39.3 ± 3.9	34.7 ± 2.3
Dishes	20.8 ± 1.5	21.9 ± 1.1	20.8 ± 2.1	**22.6** ± 1.7	21.0 ± 1.2
Dog	15.1 ± 0.7	**17.1** ± 2.6	16.5 ± 1.0	12.3 ± 1.1	12.8 ± 2.7
Electric shaver / toothbrush	32.8 ± 4.2	35.5 ± 4.7	37.2 ± 2.9	36.2 ± 5.4	**41.1** ± 2.9
Frying	23.5 ± 2.2	**23.9** ± 2.3	20.9 ± 4.8	**23.9** ± 2.2	22.2 ± 2.6
Running water	**31.7** ± 3.3	29.8 ± 2.2	30.4 ± 2.6	27.6 ± 1.8	27.2 ± 1.6
Speech	42.7 ± 3.1	**47.1** ± 2.9	45.2 ± 1.5	46.2 ± 2.6	46.3 ± 1.8
Vacuum cleaner	40.1 ± 1.7	39.9 ± 2.3	38.9 ± 3.3	**44.5** ± 4.1	40.1 ± 5.0
Total macro	32.0 ± 1.3	**33.1** ± 0.9	32.0 ± 1.1	32.5 ± 1.5	31.7 ± 1.0

Table 3: Event-based F_1-score (%) over the Validation set for each event category obtained with different time-frequency resolution working points. Mean ± standard deviation computed across 5 trainings with random initializations.

where $P(\theta_{i,1})$ is the prior probability of detection; and the likelihood ratio $\frac{P^{(j)}(x|\theta_{i,1})}{P^{(j)}(x|\theta_{i,0})}$ is the actual information about detection of an event as extracted by the jth detector CRNN. Therefore, an average fusion has the following interpretation in probabilistic terms:

$$s_i = P(\theta_{i,1}) + \frac{1}{K}\sum_{j=1}^{K} \log \frac{P^{(j)}(x|\theta_{i,1})}{P^{(j)}(x|\theta_{i,0})} \tag{4}$$

Thus, the average fusion is equivalent to averaging the information extracted by all the K detectors for each event, by keeping unaltered the prior probabilities.

3.3. F_1-score threshold tuning

If the posterior class probabilities $P(\theta_{i,1}|x)$ are properly computed (i.e., calibrated), the decisions to be made in order to optimize the expected cost in a Bayesian scenario are trivial to obtain, according to Bayes decision rule. However, given that in the evaluation the prior probabilities of the evaluation test set are not given, and are not possible to compute reliably, the task of making a decision is pointless, since the prior information is not known, hence a decision threshold cannot be set in any sound way. For the same reasons, setting a prior of 0.5 in this scenario is also pointless and unsound, since we do not know how to optimize the threshold to achieve a minimum expected cost, as the prior probabilities are not known.

Moreover, it is well known that the F_1-score and the minimum of the Bayes decision rule have different operating points. Therefore, optimizing the threshold for each of the event detection tasks to achieve minimum expected cost is pointless, since the criterion to be optimized is the F_1-score.

In order to overcome these problems, we have tuned different thresholds to the different events for each fused score s_i in order to optimize the F_1-score of each event. We have done this empirically, by experimenting in the Validation set. However, even tuning thresholds for the Validation set does not guarantee good decisions, since the prior probabilities of the evaluation test set can vary, and there is no way to predict in which way.

4. EXPERIMENTS AND RESULTS

Our experiments are based upon the baseline system[1] released by the DCASE Team. While we keep the structure of the CRNN and the training parameters, the feature extraction process is adapted to the working points described in 3.1.

The reported F_1-scores are event-based, considering a 200 ms collar on onsets and a 200 ms or 20% of the events length on offsets. Additionally, we provide Polyphonic Sound Detection Score (PSDS) [7] results, which are evaluated in DCASE 2020 Task 4 as a contrastive measure. The baseline system achieves 34.8% event-based F_1-score and 0.610 PSDS over the DESED Validation set.

[1] https://github.com/turpaultn/dcase20_task4

	Threshold
Alarm bell / ringing	0.31
Blender	0.49
Cat	0.65
Dishes	0.31
Dog	0.69
Electric shaver / toothbrush	0.61
Frying	0.29
Running water	0.45
Speech	0.83
Vacuum cleaner	0.65

Table 4: Binarization thresholds used in the *5res-thr* system.

4.1. Single-resolution results

Table 3 shows the F_1 results obtained with each of the feature resolution points over the DESED Validation set. Five systems have been trained for each resolution point, using different random initializations of the network. The mean and the standard deviation of the obtained F_1-scores are reported. We have tried to get insights into the reasons for the differences found in Table 3, but have not arrived to solid conclusions yet. In any case, the results described in Table 3 support our hypothesis that different types of sound events are detected better with different time-frequency resolution analyses.

4.2. Multi-resolution results

Aiming to aggregate the information of multiple resolutions in the sound event detection system, we have combined networks trained with different time-frequency resolution working points, following the procedure described in 3.2. A combination of five networks trained using the *BS* resolution with different initializations is studied as well.

	Base	5×BS	3res	5res	5res-thr
A. bell/ringing	*39.0*	45.0	46.1	47.2	**48.2**
Blender	*31.6*	38.3	46.4	49.5	**50.0**
Cat	*45.0*	42.0	42.2	45.2	**47.3**
Dishes	*25.0*	23.2	22.1	23.9	**25.2**
Dog	*21.7*	19.6	17.7	18.6	**22.3**
E. shaver/toothb.	*36.0*	41.6	41.8	46.8	**49.0**
Frying	*24.4*	26.7	30.0	29.7	**34.3**
Running water	*31.7*	36.9	38.2	39.6	**41.6**
Speech	*49.0*	47.6	48.0	49.9	**55.6**
Vacuum cleaner	*44.4*	47.7	54.8	58.7	**61.0**
Total macro	*34.8*	36.9	38.7	40.9	**43.4**

Table 5: Event-based F_1-score (%) results of combined models over the Validation set. The *Base* column references the Baseline System results as reported by the organizers.

	Base	5×BS	3res	5res	5res-thr
A. bell/ringing	*42.3*	45.8	44.9	46.5	**47.8**
Blender	*32.6*	43.7	**46.2**	44.2	43.7
Cat	***70.4***	69.4	60.8	66.2	68.4
Dishes	***28.4***	28.1	24.4	25.4	23.5
Dog	***30.9***	22.0	18.2	18.7	23.9
E. shaver/toothb.	*27.0*	44.1	51.2	53.2	**56.0**
Frying	*29.0*	28.3	**42.1**	40.6	36.0
Running water	*28.4*	**33.7**	28.1	31.8	31.9
Speech	*50.8*	51.9	50.3	51.3	**59.3**
Vacuum cleaner	*41.5*	44.6	52.2	52.8	**57.6**
Total macro	*38.1*	41.2	41.8	43.0	**44.8**

Table 6: Event-based F_1-score (%) results of combined models over the Public Evaluation set (eval 2019). The *Base* column references the Baseline System (2020) weights provided by the organizers.

Table 5 shows event-based F_1 results for several model combinations. A larger improvement is observed when combining models trained with different feature resolutions, suggesting that the information extracted with different time-frequency resolutions is complementary. The *3res* system, which combines resolutions T_{++}, BS and F_{++}, obtains 38.7% macro-F_1 over the Validation set. The combination of the five proposed resolution points (*5res*) reaches a macro-F_1 of 40.9%. On the other hand, the *5×BS* system, which uses a single resolution point, provides a smaller improvement upon the baseline (36.9% macro-F_1).

F_1-scores can be further improved by adjusting the binarization thresholds to their optimal values as described in 3.3. This way, *5res-thr* reaches 43.4% macro-F_1, which is our best result over the Validation set. Table 4 lists the thresholds used by *5res-thr*.

Improvements in macro-F_1 are observed as well over the DESED Public Evaluation set when applying multi-resolution and threshold tuning, as shown in Table 6. The *5res-thr* system achieves 44.8% macro-F_1, our best result over the Public Evaluation set. The results show that the thresholds adjusted over the Validation set provide higher F_1 over the Public Evaluation set in 7 out of 10 event categories.

In addition to F_1-scores, the challenge proposes the PSDS metric as an alternative measure, although it is not used to rank the submitted systems. The PSDS performances of the described model combinations over the Validation set are presented in Table 7. The *5res* system obtains our best PSDS result, 0.666. It should be noted

	α_{ct}	α_{st}	Base	5×BS	3res	5res
PSDS	**0**	**0**	*0.610*	0.635	0.657	**0.666**
PSDS cr-tr.	**1**	**0**	*0.524*	0.564	0.595	**0.609**
PSDS macro	**0**	**1**	*0.433*	0.451	0.467	**0.479**

Table 7: PSDS, PSDS cross-trigger and PSDS macro results of combined models over the Validation set. α_{ct} is the weight related to the cost of cross-trigger. α_{st} is the weight related to the cost of instability across classes. The *Base* column references the Baseline System results as reported by the organizers.

	α_{ct}	α_{st}	Base	5×BS	3res	5res
PSDS	**0**	**0**	***0.718***	0.650	0.671	0.685
PSDS cr-tr.	**1**	**0**	*0.625*	0.581	0.612	**0.627**
PSDS macro	**0**	**1**	***0.586***	0.504	0.512	0.534

Table 8: PSDS, PSDS cross-trigger and PSDS macro results of combined models over the Public Evaluation set. α_{ct} is the weight related to the cost of cross-trigger. α_{st} is the weight related to the cost of instability across classes. The *Base* column references the Baseline System (2020) weights provided by the organizers.

that varying the F_1-score thresholds does not affect the PSDS computation, as PSDS already considers different thresholds. Therefore, the PSDS results of the *5res-thr* model are identical to those of the *5res* model.

Table 8 show the PSDS performances over the Public Evaluation set. Higher PSDS are obtained by those systems with more different resolutions, although our results in this dataset are slightly inferior to those obtained by the pre-trained baseline system.

The *5res* and *5res-thr* systems have been submitted to DCASE 2020 Task 4, both outperforming the baseline in the Evaluation set. While the baseline achieves 34.9% macro-F_1, *5res* reaches 37.9% and *5res-thr* 38.2%. PSDS improves as well, obtaining 0.575, above the baseline result of 0.496.

5. CONCLUSIONS

In this paper we proposed a multi-resolution Sound Event Detection approach in the context of DCASE 2020 Task 4. Our system builds on the baseline provided by the organization, implementing three main contributions: multi-resolution analysis, model fusion and threshold tuning.

The baseline system achieved 34.8% event-based F_1-score and 0.610 PSDS over the DESED Validation set. The improvement obtained using model fusion was larger when combining models trained with different time-frequency resolutions, reaching 40.9% event-based F_1 and 0.666 PSDS when combining five resolution points. Additionally, we explored the possibility of choosing a different binarization threshold for each event category, obtaining an additional improvement in F_1 of 2.5 points (43.4%). Moreover, improvements in macro-F_1 held over the Public Evaluation set and the 2020 Evaluation set.

6. REFERENCES

[1] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, May 2016. [Online]. Available: http://dx.doi.org/10.3390/app6060162

[3] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, New York City, United States, October 2019. [Online]. Available: https://hal.inria.fr/hal-02160855

[4] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *ICASSP 2020 - 45th International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, 2020. [Online]. Available: https://hal.inria.fr/hal-02355573

[5] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[6] D. T. Toledano, M. P. Fernández-Gallego, and A. Lozano-Diez, "Multi-resolution speech analysis for automatic speech recognition using deep neural networks: Experiments on timit," *PloS one*, vol. 13, no. 10, 2018.

[7] Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 61–65.