MULTI-RESOLUTION MEAN TEACHER FOR DCASE 2020 TASK 4

Technical Report

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

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

ABSTRACT

In this technical report, we describe our participation in DCASE 2020 Task 4: Sound event detection and separation in domestic environments. A multi-resolution feature extraction approach is proposed, aiming to take advantage of the different lengths and spectral characteristics of each target category. 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 set.

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

1. INTRODUCTION

This paper describes our submission to DCASE 2020 Task 4. Our participation is based on the provided baseline system and follows the scenario of sound event detection without source separation pre-processing. This baseline is a convolutional recurrent neural network (CRNN) trained using the Mean Teacher algorithm [1]. We propose a multi-resolution analysis of the audio features (mel-spectrograms) used to train the neural network, in contrast with the single-resolution approach of the baseline. Additionally, class-specific thresholds for the F_1-score metric 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) [2, 3]. DESED 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 [4] and the provided JAMS file, obtaining a Synthetic training set with 2536 strongly-labeled clips.

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.

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

	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.

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. These sound events differ in duration and spectral characteristics. One of the hypotheses we wanted to explore with our participation in the DCASE 2020 Task 4 challenge is whether a multi-resolution feature extraction approach could provide improvements in this context.

Most systems developed for previous similar evaluations, and also the baseline system provided for this challenge, rely on a mel-spectrogram which essentially transforms the audio into a 2-D image that is later analyzed using a deep neural network. The computation of the mel-spectrogram depends on several parameters (such as the sampling frequency used for the audio, the number of points of the FFT, the type and length of the analysis window and the number of mel filters) which essentially define a single time-frequency resolution working point.

A particular point in time-frequency resolution can be more or less appropriate to detect a specific type of sound event depending on its characteristics, particularly of its length and spectral distribution. In this evaluation the different types of sound events to detect have different lengths and spectral characteristics. For instance, it is particularly easy to show that the different sounds have different lengths by analyzing the mean and standard deviation of the duration of the 10 different types of sounds in the Synthetic training set. This information is presented in Table 1.

Given that the different sound events to detect and classify

have so different lengths, it seems plausible that using several time-frequency working points in the feature extraction stage could improve sound detection and classification results. In previous experiments [5], we achieved modest improvements using multi-resolution analysis in a problem (automatic speech recognition) where the differences in the lengths and spectral characteristics of the sounds (all of them human voice phones) were much smaller.

To explore this possibility we have used several mel-spectrogram computations using different parameters to use several (up to 5) different time-frequency resolution working points. Our approach is based on the baseline provided by the organization. We have replicated the baseline several times, modifying each instance to handle a different time-resolution working point. Finally, we have fused the frame-level estimation of the class posteriors provided by each subsystem.

To define the different time-frequency resolution working points we have taken as a reference the point defined by the baseline and have defined other points by increasing and decreasing the time and frequency resolution. In this way, we have defined 5 time-frequency resolution working points. All of them 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 length, window length, window hop and number of mel filters) are modified to increase time or frequency resolution as described below for each time-frequency resolution working point.

1. ***BS*** (baseline). The baseline uses an analysis window of length $L = 128$ ms. which makes it difficult to accurately detect events smaller in time than the window length ($L = 128$ ms.). 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. Therefore it will be difficult to detect changes in frequency closer than that. This frequency resolution is later limited in a non-linear way by the use of the Mel filterbank with 128 filters. The specific parameters used by the baseline are the following.
   - FFT length: $N = 2048$ samples.
   - Window length: $L = 128$ ms. ($L = 2048$ samples).
   - Window hop: $R = 15.94$ ms. ($R = 255$ samples).
   - Number of Mel filters: 128.
2. ***T++*** (twice better time resolution). We halve the analysis window to a length of $L = 64$ ms., which makes it possible to detect shorter events as small as the new window length ($L = 64$ ms.). On the other hand, the frequency resolution decreases to $4/1023 \times 16000 \approx 62.5$ Hz. We also halve the number of Mel filters.
   - FFT length: $N = 1024$ samples.
   - Window length: $L = 64$ ms. ($L = 1024$ samples).
   - Window hop: $R = 8$ ms. ($R = 128$ samples).
   - Number of Mel filters: 64.
3. ***F++*** (twice better frequency resolution). We double the analysis window length to $L = 256$ ms. This makes it difficult to detect events smaller than the new window length ($L = 256$ ms.). On the other hand, we get a much better frequency resolution of $4/4095 \times 16000 \approx 15.5$ Hz. To keep this increased frequency resolution we double the number of Mel filters.
   - FFT length: $N = 4096$ samples.
   - Window length: $L = 256$ ms. ($L = 4096$ samples).
   - Window hop: $R = 32$ ms. ($R = 512$ samples).
   - Number of Mel filters: 256.
4. ***T+*** (intermediate point between ***BS*** and ***T++***). Analysis window of length $L = 96$ ms. and frequency resolution of $4/1536 \times 16000 \approx 41.7$ Hz. Also an intermediate number of Mel filters is used.
   - FFT length: $N = 2048$ samples.
   - Window length: $L = 96$ ms. ($L = 1536$ samples).
   - Window hop: $R = 12$ ms. ($R = 192$ samples).
   - Number of Mel filters: 96.
5. ***F+*** (intermediate point between ***BS*** and ***F++***). Analysis window of length $L = 192$ ms. and frequency resolution of $4/3072 \times 16000 \approx 21$ Hz. Also an intermediate number of Mel filters is used.
   - FFT length: $N = 4096$ samples.
   - Window length: $L = 192$ ms. ($L = 3072$ samples).
   - Window hop: $R = 24$ ms. ($R = 384$ samples).
   - Number of Mel filters: 192.

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 $P^{(j)}(\theta_{i,0}|x) = 1 - P^{(j)}(\theta_{i,0}|x)$, where x is the audio observation at this particular moment in time. 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:

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

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:

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

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

It is worth saying that the prior probability of detection of each event, $P(\theta_{i,1})$, is either computed from the training/validation set, or given by the evaluation rules somehow. In this evaluation, however, these prior probabilities are not specified for the testing evaluation set. Moreover, the empirical prior probabilities of detection vary from the training, validation and other datasets given in the evaluation, and there is not an indication of whether the prior probabilities will be the same in the evaluation test set as in the training/validation sets or not. That makes impossible to compute the prior probabilities of detection. This will have consequences in the decision-making stage, as described below.

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, and 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. Results are shown below. 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.

Because of the reasons above, it is worth saying that we believe that this situation of not knowing the prior probabilities makes the task to lose relevance, as long as the empirical priors of the testing set are not specified. Making decisions to optimize expected costs or precision-recall-based measures as F_1-scores, as a task, is not properly defined unless the empirical priors in the evaluation test set are known, or predictable in some way. Thus, it might happen that extremely well-performing detectors fail to obtain a good F_1-score just because they are not properly designed for the 0.5 threshold. As the optimal thresholds strongly depends on the empirical prior, measuring primary performance by F_1-score without prior specification leads to potentially misleading overall evaluation results, in our opinion.

4. EXPERIMENTS AND RESULTS

Our experiments are based upon the baseline system[1] released by the DCASE Team. The general structure of the CRNN and the training parameters are kept, while the resolution parameters for feature extraction are changed as described in 3.1. The pooling layers of the convolutional stage of the network had to be adapted to the number of mel-filters used in each resolution point so that the input dimension to the recurrent stage is consistent.

The reported F_1-scores are event-based and computed with a 200 ms collar on onsets and a 200 ms or 20% of the events length collar on offsets. Additionally, the Polyphonic Sound Detection Score (PSDS) [6] results of the submitted systems are presented. The baseline system achieves 34.8% event-based F_1-score and 0.610 PSDS over the DESED Validation set.

4.1. Single-resolution results

Table 2 shows the event-based F_1-score results for the DESED Validation set obtained with each of the feature resolution points described in 3.1. For each resolution point, five systems have been trained with different random initializations of the network. The mean and the standard deviation of the obtained F_1-scores are reported.

4.2. Multi-resolution results

In order to include multi-resolution information in the sound event detection system, networks trained with different feature resolutions have been combined following the procedure described in 3.2.

Table 4 shows event-based F_1 results for several model combinations. Combining models trained with different feature resolutions provides a larger improvement. In the case of the *3res* system, 38.7% macro F_1 is obtained by combining resolutions T_{++}, BS and F_{++}. The combination of the five proposed resolution points (*5res*) reaches a macro-F_1 of 40.9% over the Validation set.

The F_1 results can be further improved by adjusting the binarization thresholds to their optimal values as described in 3.3, reaching 43.4% macro-F_1 (*5res-thr*), which is our best result over the Validation set. The thresholds used by this system are listed in Table 3.

The PSDS performances of the described model combinations are presented in Table 5. Figure 1 shows the three PSDS curves of the *5res* model over the Validation set. It should be noted that varying the F_1-score operation point does not affect the PSDS computation, therefore the PSDS results of the *5res-thr* model are identical to those of the *5res* model.

5. CONCLUSIONS

In this paper we described our participation in DCASE 2020 Task 4, which follows the scenario of SED without source separation.

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

	$\mathbf{T}_{++}$	$\mathbf{T}_{+}$	**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 2: 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.

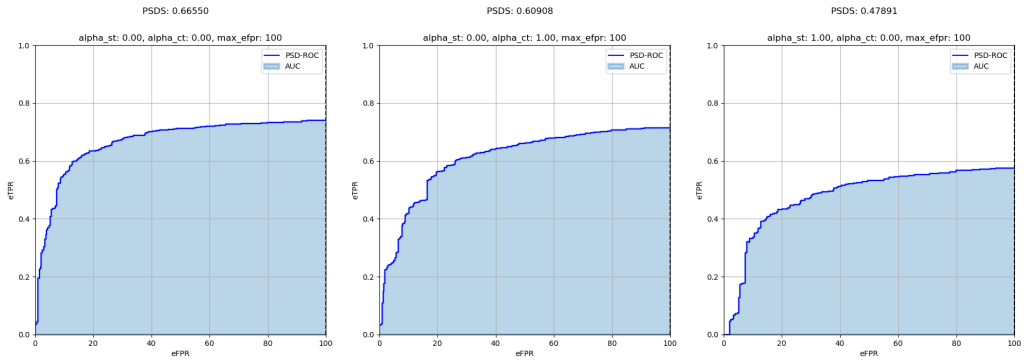

Figure 1: PSDS (left), PSDS cross-trigger (center) and PSDS macro (right) curves for the 5-resolution system computed over the Validation set.

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

Our system builds on the baseline provided by the organization, implementing three main improvements: 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. Furthermore, 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%).

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

Table 4: 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.

	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 5: 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.

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] 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

[3] 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

[4] 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.

[5] 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.

[6] 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.