MULTI-TASK LEARNING AND POST PROCESSING OPTIMIZATION FOR SOUND EVENT DETECTION

Technical Report

Léo Cances, Thomas Pellegrini, Patrice Guyot

IRIT, Université de Toulouse, CNRS, Toulouse, France
{leo.cances,thomas.pellegrini,patrice.guyot}@irit.fr

ABSTRACT

In this paper, we report our experiments in Sound Event Detection in domestic environments in the framework of the DCASE 2019 Task 4 challenge. The novelty, this year, lies in the availability of three different subsets for development: a weakly annotated dataset, a strongly annotated synthetic subset, and an unlabeled subset. The weak annotations, unlike the strong ones, provide tags from audio events but do not provide temporal boundaries. The task objective is twofold: detecting audio events (multi label tagging at recording level), and localizing the events precisely within the recordings. First, we explore multi-task training to take advantage of the synthetic and unlabeled in domain subsets. Then, we applied various temporal segmentation methods using optimization algorithms to obtain the best performing segmentation parameters. On the multi-task itself, we explored two strategies based on convolutional recurrent neural networks (CRNN): 1) a single branch model with two outputs, 2) multi-branch models with two or three outputs. These approaches outperform the baseline of 23.7% in F-score by a large margin, with values of respectively 39.9% and 33.8% for the first and second strategy, on the official validation subset comprised of 1103 recordings[1].

***Index Terms*—** Sound event detection, weakly supervised learning, multi-task learning, convolutional neural networks, pseudo labelling

1. INTRODUCTION

In this report, we present the approaches developed to address the issue of sound event detection in domestic environments, in the framework of the task 4 of the DCASE 2019 challenge. This task evaluates systems for the large scale detection of sound events using weakly labeled data. For the 2019 edition, three datasets are available for development: a weakly annotated dataset, a strongly annotated synthetic subset, and an unlabeled subset.

Our processing pipeline relies on two phases. On one hand, we use convolutional recurrent neural networks (CRNN) that follow a multi-task learning paradigm [1].

On the other hand, the optimization stage works on the outputs of these models to find the best audio tagging and temporal localization algorithms. We will refer to this two optimizations as *at optimization* and *loc optimization*.

[1]Code is available at:
`https://github.com/topel/dcase19RCNNtask4`
`https://github.com/leocances/dcase2019`

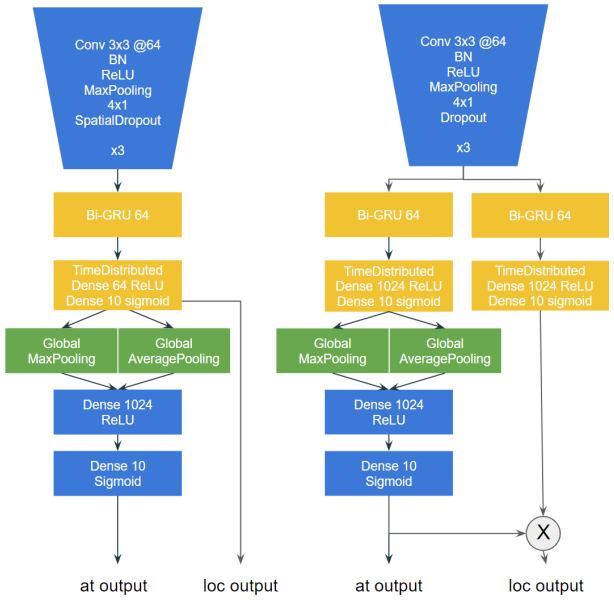

Figure 1: **Model 1** (left): Mono-branch CRNN dual task model. Localization is performed by a Time Distributed Dense layer. **Model 2** (right): Multi-branch CRNN. The localization output uses its own recurrent and time distributed layers. The goal is to obtain two separate branches specialized in either tagging or localizing. A final multiplication layer before the *loc output* is a mask enforcing the branch to focus only on the detected classes by the AT branch.

2. PROPOSED METHODS

The models are based on the baseline systems of last year DCASE 2018 challenge task 4 [2]. In extension, we use specific loss functions adapted to the different subsets available. Two strategies are explored, mono-branch model and multi-branch model. One output is dedicated to providing predictions at clip level (*at output*) and another one at frame level (*loc output*). This multi-task learning approach showed competitive performance when using mixed annotated data. Based on this framework, we proposed three models that are visible on Figures 1 and 2.

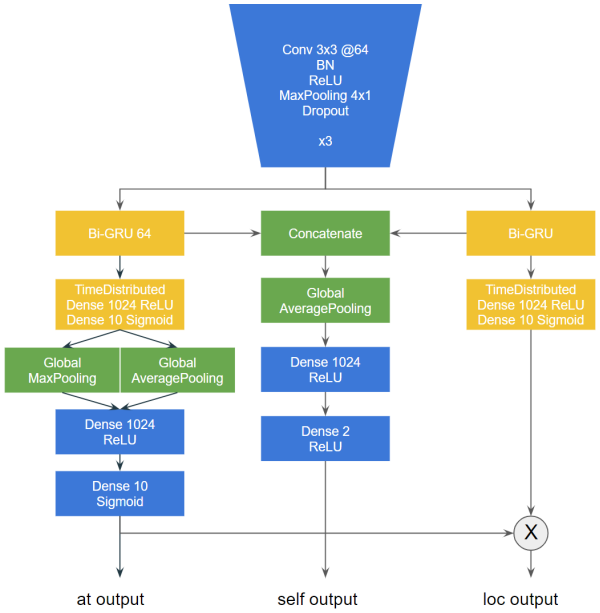

Figure 2: **Model 3**: CRNN triple task model. Compared to model 2, a third branch has been added. It allows the exploitation of unlabeled in domain data for a self supervised task.

Models 1 & 2

Models 1 and 2 were trained on both the weak and synthetic subsets for audio tagging (AT), and localization (LOC) in the case strong annotations are available. The synthetic subset was used for both AT and LOC tasks whereas the weak subset only for AT. A binary cross entropy loss function was used to train the models at both tagging (AT) and localization (LOC) outputs.

Model 3

For this model, a third branch was added for a self supervised task [3], exploiting the unlabeled in domain subset. This branch aimed to predict if the input was transformed using a left/right mirroring, an up/down mirroring, both at the same time or left as is. A binary cross entropy loss function was used to train the model for tagging, localization, and self supervision (SELF). When strong annotations or weak annotations are missing, the loss is set to zero for AT and/or LOC.

3. POST PROCESSING

This section presents the post processing steps used within our system, namely tagging, smoothing, segmenting, and optimizing. We extend our previous work on localization optimization [4] by adding optimization methods for tagging.

All models provide clip level and frame level predictions. These are the outputs of sigmoid activation layers. Therefore, thresholds must be applied to achieve tagging and segmentation. Given the diversity of the classes, those thresholds should adapt to each sound event category. On the localization task, we retain only the localization predictions of the predicted audio tags. Thus, the localization of sound events heavily depends on the quality of the tagging.

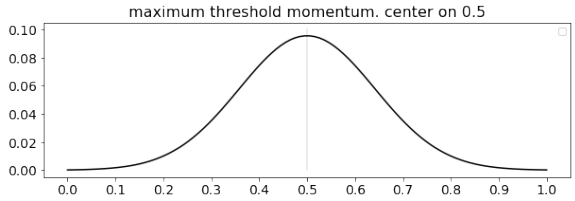

Figure 3: Maximum threshold momentum centered on a 0.5 value. At each iteration, the random δ is drawn from a normal distribution, centered on the value of the last best threshold of the targeted class during the optimizing process

Smoothing predictions proved to be essential to ensure best performance.

The tagging task is simpler, the AT output is binarized using either class independent (CI) thresholds or class dependent thresholds. We tested two approaches to estimate the best thresholds.

3.1. Threshold optimization for tagging

The purpose of this optimization step is to get the best tagging performance of the model. To do so, it must maximize the score for each class individually. We tested two way to achieve this.

- The simplest way consists of testing values between 0 and 1 with a specific precision, and this for each class c independently. Computation time is directly depending on the step size.
- The second approach, more complex, uses a genetic algorithm [5] that draws a value δ randomly drawn from a normal distribution $N(0.5, \sigma)$ (Figure 3). This value is then added to or subtracted from the current threshold th_c estimate. In this way, the modification will be higher around 0.5 and lower at the boundaries (0 and 1). σ is then used to set the delta decay value according to its distance from the center. The computation time depends on the number of iterations.

Using either a classic exhaustive approach or the genetic algorithm is directly driven by the application, and the thresholds precision desired. On the 10 classes of task 4, for a precision of 10^2 on each threshold, the exhaustive search is more than enough and instantaneous. However, for a 10^5 precision, it is 10 times longer than the genetic algorithm for the same results. The latter is then more suitable for large datasets with many classes such as Audioset [6] and its 632 classes.

3.2. Frame level probability curve smoothing

Frame level predictions are smoothed. It removes noise in the probabilities, limiting the number of small segments or small gaps created during the segmentation process. In our work, we use a smoothed moving average. The smoothing of the temporal prediction output by the model can be class dependent as the smoothing window size may change with the class.

3.3. Segmenting

The parametric methods require optimization. They can either be class independent or class dependent. We tested two of them

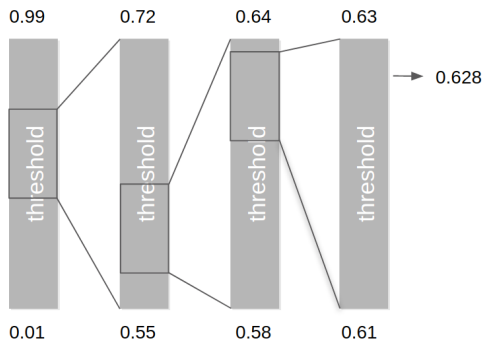

Figure 4: Example of the dichotomic search of the best threshold of a specific class. The same method is applied to the smoothing window size

called i) class (in)dependent absolute (CIA CDA) and ii) class (in)dependent hysteresis (CIH CDH).

(i) Absolute thresholding refers to directly applying a unique and arbitrary threshold to the temporal predictions without using their statistics. This naïve approach still yields exploitable results that can get close to the best ones in some cases. It is also the approach with the shortest optimization time due to the single parameter to optimize.

(ii) Hysteresis thresholding consists of two thresholds. One of them is used to determine the onset of an event, and the second one its offset. This algorithm is used when probabilities are unstable and changing at a high pace. It should, therefore, decrease the number of events detected by the algorithm and reduce the insertion and deletion rates, giving a better error rate than the Absolute threshold approach.

Optimization for localization

The segmentation methods presented exploit arbitrary parameters to locate with precision sound events. The search for the best parameter combination is a meticulous work that is often difficult to automatize. Indeed, depending on the number of parameters to tune, the search space growth is exponential, and the execution time often exceeds reasonable times. Consequently, we implemented a smarter exploration method called dichotomic search, (Figure 4)

For every parameter to tune, the user provides global boundaries and, in between these boundaries, the algorithm tries every combination with a coarse resolution and picks the one that yields the best score. From this combination, new smaller boundaries are computed. The complete process is then repeated in between the new limits, with every step increasing the precision of each parameter and reducing the search space. It stops when the number of steps given by the user is reached.

The dichotomous search algorithm, when compared to an exhaustive search of all the possible combinations, considerably reduces the time needed to reach a near optimal solution with excellent accuracy. However, the execution time is still dependent on the number of parameters to tune and the amount of iterations for every step. The total number of combinations increases exponentially.

4. DATA

The methods are evaluated on a subset of the Google Audioset and FreeSound [7] (FSD), provided within the task 4. All audio clips are 10 second long and contain one or multiple sound events among 10 different classes, some overlapping with each other.

The training set consists of 1578 weakly annotated clips, an unlabeled in domain subset of 14412 clips and a synthetic subset of 2045 clips, strongly annotated, created using Scaper and FSD. The test set contains 1168 clips strongly annotated from human annotators. For the training of the two first models, the weak and synthetic subsets are used. With the third one, the unlabeled in domain subset is also used for a self supervised task.

Each recording is converted to a mono signal with a sampling rate of 22050 Hz. Log Mel filter banks are used as features. Each recording is split into 431 frames by 64 mel bands. Data augmentation is performed on both the raw signals and the mel spectrograms. Table 4 describes the different transformations used.

Transformation	**Model 1**	**Model 2**	**Model 3**
Time stretch [0.9, 1.1]	✓	✓	✓
Pitch shift [3, 3]	✓	✓	✓
Level [0.8, 1.2]	✓	✓	✓
Noise (normal 10 db)		✓	✓
Mirror: left / right			✓
Mirror: up / down			✓

Table 1: The different data augmentation transformations used for each system. The first four are applied on the signal and the last two on the mel spectrograms.

5. RESULTS

Table 2 shows the gains generated by the different segmentation methods. The comparison is made using a fixed threshold of 0.5 for all classes and no smoothing. Whereby, optimizing the segmentation process gives a significant performance increase. Table 3 shows the detailed event and segment based class wise (Sb & Eb) macro F-scores of the three systems presented above on the development set 3. The best tagging and localization parameters were obtained using the optimization steps described in the article. Best performance was achieved using model 1 and CD hysteresis algorithm.

	Model 1	**Model 2**	**Model 3**
No optimization	15.9%	21.5%	24.9%
CI Absolute threshold	29.1%	26.2%	27.8%
CD absolute threshold	36.9%	33.3%	31.8%
CD hysteresis	**39.9%**	33.8%	32.2%

Table 2: Impact of the three post processing methods on the event based macro F score on the validation task 4 subset. **Class independent (CI)** refers to parameters identical for each class. Two **class dependent (CD)** refers to parameters different for each classes.

We submitted two systems to the challenge using CD hysteresis . The submision names are:

- **Model 1**: PELLEGRINI_IRIT_task4_1
- **Model 2**: CANCES_IRIT_task4_2

	Baseline		Model 1		Model 2		Model 3	
Class	Eb	Sb	Eb	Sb	Eb	Sb	Eb	Sb
Alarm bell ringing			42.3%	76.5%	31.8%	71.3%	25.5%	74.2%
Blender	-	-	40.4%	57.9%	27.5%	45.3%	30.1%	52.9%
Cat	-	-	45.9%	57.8%	28.2%	52.9%	19.4%	54.4%
Dishes	-	-	26.6%	47.9%	21.6%	49.0%	24.7%	46.7%
Dog	-	-	17.8%	31.8%	19.3%	49.5%	13.4%	56.6%
Electric shaver toothbrush	-	-	48.5%	60.6%	52.3%	64.5%	49.1%	65.4%
Frying	-	-	35.5%	61.4%	47.9%	63.1%	44.3%	60.4%
Running water	-	-	30.7%	61.6%	17.3%	54.1%	18.5%	59.3%
Speech	-	-	39.9%	79.9%	40.0%	78.2%	44.9%	78.8%
Vacuum cleaner	-	-	68.7%	69.7%	51.8%	66.9%	52.1%	66.2%
Macro-F1	23.7%	55.2%	**39.9%**	60.5%	33.8%	59.5%	32.2%	**61.5%**

Table 3: F-scores for our three multi-task models on the validation Task 4 subset. **Eb** stands for Event-based and **Sb** for Segment-based macro-F-scores.

6. CONCLUSION

In this work, we proposed three models for sound event detection. Our methods were adapted to use different kinds of subsets, including strong and weak annotations. We used a multi-task learning approach to benefit from these mixed annotated data.
Moreover, we mainly focused our efforts on exploring optimization methods to estimate the best performing thresholds to make decisions on the detection and localization of sound events. We introduced different optimization algorithms that drastically boost model's performance. Our best model performs localization and audio tagging in a same single branch. It achieved a 39.9% event based score on the DCASE 2019 task 4 validation dataset, on which the baseline system yielded 23.7%. Without threshold optimization for tagging nor for localization, the same model achieved a 15.9% F-score.

7. ACKNOWLEDGMENT

This work was partially supported by the Agence Nationale de la Recherche LUDAU (Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning) project (ANR-18-CE23-0005-01).
Experiments presented in this paper were carried out using the OSIRIM platform that is administered by IRIT and supported by CNRS, the Region Midi-Pyrénées, the French Government, ERDF (see http://osirim.irit.fr/site/en).

8. REFERENCES

[1] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[2] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. Parag Shah, "Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, Woking, United Kingdom, Nov. 2018, submitted to DCASE2018 Workshop. [Online]. Available: https://hal.inria.fr/hal-01850270

[3] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *CoRR*, vol. abs/1803.07728, 2018. [Online]. Available: http://arxiv.org/abs/1803.07728

[4] L. Cances, P. Guyot, and T. Pellegrini, "Evaluation of post-processing algorithms for polyphonic sound event detection," no. arXiv:1906.06909.

[5] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[6] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans: IEEE, 2017, pp. 776–780.

[7] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proc. of the 18th ISMIR Conference*, Suzhou, 2017, pp. 486–493.