LARGE-SCALE WEAKLY SUPERVISED AUDIO TAGGING FOR SMART CARS

Carlos Roig Marí

MSc MET Student at UPC

ABSTRACT

In this paper, a modification of the winners of the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge from 2017 Task 4 Subtask A is presented. The system is expanded by adding a layer of complexity by using multiple resolution data inputs and extracting the features after a series of Convolutional Neural Networks, after that multiple feature maps are combined and fed into a Recursive Neural Network in order to obtain a classification. With the modifications proposed, a result of F value of 51.9% is obtained improving the baseline system.

Index Terms— Audio Tagging, CRNN, DCASE2017 challenge, gated linear unit

1. INTRODUCTION

DCASE2017 [1] is an annual challenge organized by the Audio research group of Tampere University of Technology, by Carnegie Mellon University and by the Inria. This work is focused on the Task 4 of the challenge: Large-Scale Weakly Supervised Sound Event Detection for Smart Cars that consists on the subtasks: (A) Audio Tagging and (B) Sound Event Detection.

Audio tagging is deciding to which class or tag a sound corresponds, without giving a temporal measure (timestamp), on the other hand, sound event detection is giving a temporal measure of where the sound is located in the complete audio file.

2. DATASET

The dataset provided by the organizers is based on the Audio Set [2], which is a large-scale dataset of manually annotated audio events from YouTube videos.
The task 4 dataset is a subset of the Audio Set focused on sound that would be useful for a smart car to recognize. The set is composed of 17 different sound events divided in two categories: "Warning" and "Vehicle". The warning sounds are sounds related to acoustic events that give some information regarding a not very common state of the traffic, like an ambulance siren, a police car siren or a scream from a person. The Vehicle sounds are the different set of noises and sounds that every vehicle can perform and are split in the different possible vehicles (car, bicycle, bus, truck, train…). A comparison of the data distribution can be found in (Table 1).

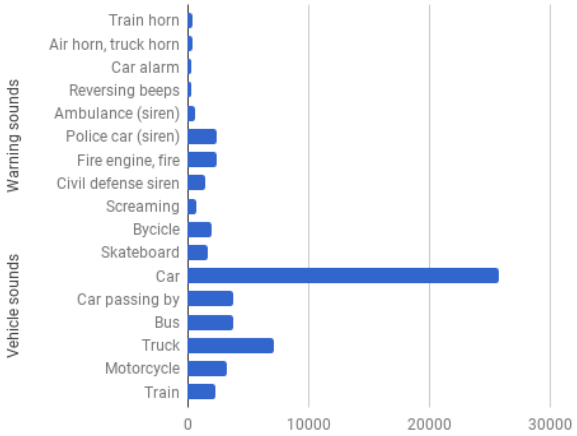

Table 1. Distribution of the different audio events.

The data distribution of the audio sounds are highly unbalanced. This distribution resembles a real world situation, where there will be a lot of cars at all moments, but a siren is a more sporadic event, for example.
Even though this distribution is realistic, when a Machine Learning model is being trained, having such a big divergence of data, will probably bias the model to predict the car class more than the other. In order to deal with this potential problem, some action will be taken in training stages.

3. IMPLEMENTATION

In the following section, the system proposed by the winners of the DCASE 2017 Task 4 - Subtask A is reviewed.

3.1. Winner DCASE 2017 Task 4 - Subtask A

The authors of the winner system of the DCASE 2017 Task 4 – Subtask A [3], propose a series of methods to deal with the different problems of the dataset. The base system is based on a Convolutional Recurrent Neural Network (CRNN) [4,5]. The network takes as input a T-F

representation of the signal (log Mel spectrogram) and passes these features through a series of Convolutional Neural Networks that extract the features in the frequency domain of the audio clip, then are fed into a Bi-RNN that captures the temporal evolution, and finally, RNN output is processed by a feed forward network to predict the class of the input signal. In order to deal with the different challenges of the task, the authors propose a series of methods that are: data balancing, modification of the activation functions of the neural network and fusion of results.

3.1.1. Data balancing

As it has been already explained in section 2, the data is highly unbalanced, so the authors of the winner system decided to implement a training with a fixed batch size with balance. The batch samples are randomly selected from the whole dataset but entrusting that there is at least one sample of each class per batch.

3.1.2 Learnable gated activation function

This topic is the main point introduced by the authors of the winner system, they propose a new type of activation called gated linear units (GLUs), which has been already used in other topics, like language modelling [6], and replace the common Rectified Linear Units (ReLU) that are commonly used in Deep Learning. GLUs are defined as:

$$\mathbf{Y} = (\mathbf{W} * \mathbf{X} + \mathbf{b}) \odot \sigma(\mathbf{V} * \mathbf{X} + \mathbf{c}),$$

where σ is the sigmoid non-linearity, ⊙ is the element-wise product, * is the convolutional product, **W** and **V** are convolutional filters, **b** and **c** are the biases. **X** is the input T-F representation and **Y** is the output.
So the output of each layer is a linear projection $(\mathbf{W} * \mathbf{X} + \mathbf{b})$ modulated by the gates $\sigma(\mathbf{V} * \mathbf{X} + \mathbf{c})$, this kind of units help with the vanishing gradient problem in deep networks by providing a linear path for the gradients while retaining non-linear capabilities through the sigmoid operation. GLUs can be seen as an attention scheme on the time-frequency (T-F) bin of each feature map.

3.1.3 Fusion of results

The network optimizer is the Stochastic Gradient Descent (SGD) algorithm that gradually improves the results of the neural network. The authors perform a fusion of results of the final epochs in order to obtain a more stable result.

4. MODIFICATIONS AND RESULTS

For this project, the winner system has been modified at different levels. The first part of the project has been a tuning of the input size of the T-F representation of the audio signal. The second modification has been to test multiple modified T-F representations and combine the outputs of the CNN before the RNN to try to obtain a richer feature map before classification. The image of the whole system can be seen in Figure 1.

4.1. Input result tuning

In the first part of this project, the system tested was using the same subset to validate the network parameters and to test the network, since this is a very bad practice in any machine learning task, the following results have to be seen with a skeptical eye.
The different values tested for the frequency axis are: 64, 128 and 256 Mel frequencies and for the temporal axis 30, 60, 120, 240 and 360 values for the window size, always with a 50% overlap between frames.

F	T	Precision	Recall	F_value
64	**240**	0.513	**0.554**	0.533
128	**240**	**0.535**	0.548	**0.542**
128	**120**	0.498	0.518	0.508
64	**360**	0.506	0.523	0.515

Table 2. Best results with the same validation and test set.

As seen in Table 2, the results for the original system (F=64, T=240) are only, partially improved by changing the frequency F to 128.
The following results are with a system where both training, validation and test are different sets, being training a random 90% split of the original training set, the remaining 10% is the validation set and the test set is the previous validation set. The challenge test set does not have available ground truth labels, so it could not be used.

F	T	Precision	Recall	F_value
64	**240**	**0.459**	**0.531**	**0.493**
128	**240**	0.352	0.365	0.358
128	**120**	0.381	0.421	0.4
128	**360**	0.431	0.446	0.438

Table 3. Best results with different validation and test set.

In this case, the original values are clearly better than the rest, even though that the decision process was not revealed by the authors, probably a similar process was performed leading to their decision of using F=64 and T=240.

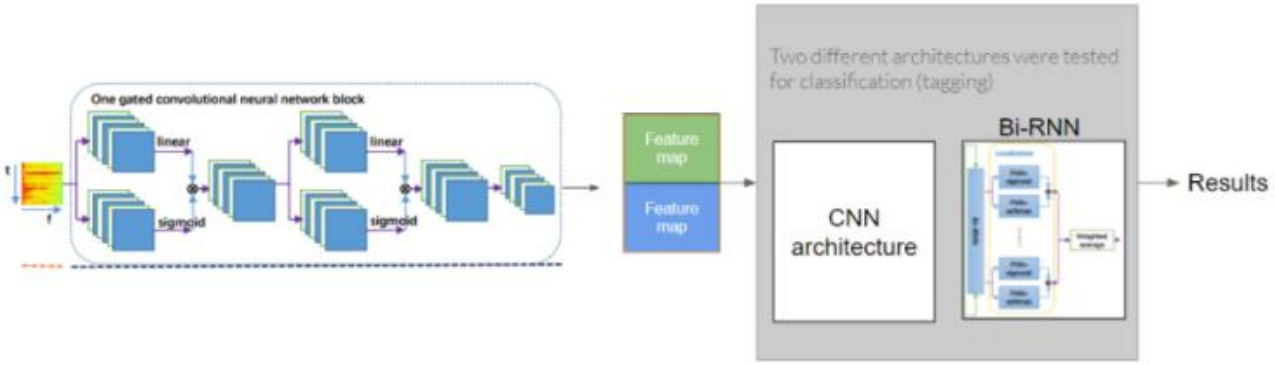

Figure 2. A representation of the full architecture implemented in this paper. The first part corresponds to the CNN of the winner system used to extract the features. Then the feature maps are combined and fed into a CNN architecture or a Bi-RNN to obtain a classification result.

4.2. Multiple feature input sizes

With the multiple original systems trained, a very common technique has been tested for the final classification system. First the network is loaded with the weights of the best validation accuracy. Then all the input samples are fed into the network and after the full CNN the resulting feature map is stored. This process is performed for all the different input feature sizes obtaining multiple feature maps for each audio clip.

With the extracted features, two networks are trained to classify the audio samples. The first architecture is a straight forward Convolutional Neural Network without much complication, a simpler version of a VGG-like network [7]. The second architecture tested is based on the Bi-RNN of the winners of the challenge with deeper LSTMs [8].

The results of both architectures are compared with the baseline system that is the original system given by the authors, trained with 90% of the training set, validated on the remaining 10% and the results are given on the test set. Same methodology than the one performed throughout this paper.

Architecture	Precision	Recall	F-score
Baseline	0.459	0.531	0.493
2-Feat. Map Bi-RNN	**0.494**	**0.546**	**0.519**
3-Feat. Map CNN	0.432	0.375	0.401
3-Feat. Map Bi-RNN	0.494	0.546	0.502

Table 4. Results using different architectures and feature maps.

From the results, it is clearly seen that adding a little bit more of complexity with multiple T-F input sizes to the system helps compared to the baseline system. But adding too much complexity results in a worse performance.

Another interesting result is that classical models that perform quite well in image classification, for example, are not working at all for a task like audio tagging with T-F input data. Classic architectures like VGG use the lower layers of the network to understand basic parameter like edges or colors, and as the network goes deeper more complex shapes start to appear. For a task like this it is not clear that a sound maps properly to a specific shape so, structures that are based on finding shapes and colors in the input may not perform well or at least worse than recurrent models that are intuitively better for audio tasks.

5. CONCLUSIONS

The system proposed is a very interesting attention based system, with some new techniques implemented in the field of audio tagging.

The results of the final multi resolution feature input are not a big improvement with regard to the baseline, but it is clear that by adding a bit of complexity to the proposed system we can improve the results.

Another interesting conclusion is that classical methods for image, work quite poorly for a different task like audio tagging but where the full pipeline is very similar, having an image (RGB or T-F, depending on the task) using a deep neural network to extract the interesting features of the input and finally some type of network to classify and obtain a result or label.

A final remark is that the system implemented by the authors is though for both audio tagging and sound event detection, while the improvement proposed in this paper is only tested for audio tagging and may perform very bad in a sound event detection problem.

The code for this project is available at: https://github.com/Roiginbag/DSAP-Audio-Tagging-DCASE2017

6. ACKNOWLEDGMENTS

Even though this was an academic project, I wanted to thank the authors of the DCASE2017 Task 4 Subtask A. For releasing their code publicly, even that it was not mandatory for the challenge and especially to Josep Pujal from UPC for allowing the use of the university computing cluster.

7. REFERENCES

[1] DCASE2017 Challenge Task 4 page: http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-large-scale-sound-event-detection#description

[2] J. F. Gemmeke, D. P. W. Ellis et al, "Audio Set: An Ontology and Human-labeled Dataset for Audio Events"

[3] Y. Xu, Q. Kong, et al. "Large-scale Weakly Supervised Audio Classification Using Gated Convolutional Neural Networks", arXiv preprint arXiv:1710.00343, 2017

[4] Sharath Adavanne, Pasi Pertila, and Tuomas Virtanen, ¨ "Sound event detection using spatial features and convolutional recurrent neural network," arXiv preprint arXiv:1706.02291, 2017

[5] Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, Tuomas Virtanen, et al., "Convolutional recurrent neural networks for polyphonic sound event detection," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 6, pp. 1291–1303, 2017.

[6] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," arXiv preprint arXiv:1612.08083, 2016

[7] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXic preprint arXiv:1409.1556, 2014

[8] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", Neural Computation 9(8):1735-1780, 1997