DCASE2024 TASK1: LOW-COMPLEXITY CLASSIFICATION OF ACOUSTIC SCENES BASED ON REDUCED SOUND DURATION AND VOTING

Technical Report

Fabrice Auzanneau

Université Paris-Saclay
CEA, List
F-91120, Palaiseau, France
fabrice.auzanneau@cea.fr

ABSTRACT

This report describes our approach to the DCASE (Detection and Classification of Acoustic Scenes and Events) Challenge for Task 1 ”Low-Complexity Acoustic Scene Classification” [1]. The task 1 of the DCASE challenge aims to classify acoustic scenes using devices with low computational power and memory. This task involves a combination of precision and complexity, which encourages participants to build efficient systems for acoustic scene classification (ASC). This year, an additional challenging real-world situation has been added: the limited availability of labeled data. The systems must take into account five scenarios that progressively limit the amount of training data. The largest set corresponds to the entire training data set, while the smallest contains only 5% of the audio clips in the training data set. Our approach is a combination of the use of a deep neural network and statistical processing. A network based on the YoloV8 topology is pruned to ensure that it meets the memory constraints of the challenge. The network is trained on half-length data (500ms), then quantized to reduce its size. During inference, each one-second sound is divided into two 500ms parts. Each half is used to make an inference, and both results are combined to improve the classification result. In the event of disagreement, a voting strategy is applied to decide on the correct category. Classification performance is thus improved by 2% for the smallest subset and by over 5% for the largest one.

Index Terms— DCASE, ASC, Task 1, Yolo, vote, quantization

1. INTRODUCTION

The DCASE (Detection and Classification of Acoustic Scenes and Events) challenge [2] serves as a prominent benchmark for evaluating acoustic scene classification (ASC) methods [3], focusing on classifying ambiances like airport, bus, subway stations, road traffic, and public squares. The goal of Task 1 in the DCASE2024 challenge is to recognize different acoustic scene classes using a low-complexity classification model. The development dataset includes audio recordings from ten distinct acoustic scenes across ten European cities [4]. Additionally, partially synthesized data was created from the original recordings. The task necessitates accurately labeling these scenes while keeping the model's complexity low to ensure adaptability across various devices. Therefore, a balance must be respected between reducing model parameters and maintaining accuracy.

The model is restricted to a maximum of 128k parameters (including zero values) and 30 million multiply-accumulate operations (MACs) per one-second sample. This year, an additional challenging real-world situation must be taken into account: the limited availability of labeled data. To this end, systems must consider five scenarios that progressively limit the amount of training data. The largest subset corresponds to the full train split, whereas the smallest subset only contains 5% of audio snippets of the full train dataset. This new challenge offers an opportunity to explore innovative approaches for developing high-performing ASC models with minimal computational requirements in real-life recording conditions.

In previous years, the results of the challenge have been very good. Various types of network, from the simplest convolutional network to more complex ones involving new exotic convolution layers [5] and knowledge distillation have been proposed. This year, new networks or learning strategies will be proposed, which will certainly achieve better results. The additional constraint of using incomplete datasets will, however, reduce the achievable performance, as the minimal dataset contains 20 times less data than the complete dataset.

Conventional approaches to this kind of problem involve the use of data augmentation strategies. The approach adopted here goes one step further. It is based on the observation that sounds from the acoustic scenes under consideration are, from the human ear's point of view, quite similar from one class to another, and often resemble noise. This means that any part of a given 1-second sound is very similar to any other part of the same sound. The duration of the scenes to be classified is very short (one second), which does not allow a human to recognize them with sufficient precision. In fact, this duration can be considered arbitrary, in the sense that below a certain length, it becomes impossible for a human to recognize them.

But an electronic system uses samples acquired at a fairly high sampling rate, ensuring that even for a short period of time, these samples are sufficiently numerous to be used by a neural network. Consequently, based on these two observations, we defined a strategy combining neural network recognition and the use of sound portions to increase the training dataset and votes to improve classification performance.

2. DATA AUGMENTATION

For each subset split of the dataset, sounds with a duration of one second are divided into smaller parts, each part becoming a new sound used for training. Therefore, all sounds used for training are coming solely from the same subset as the original full duration sound. No other external data was used for training the network, and the data from the test split was not used for training.

Two smaller durations were tested: 500 ms (division of the original duration by two) and 333 ms (division by three). The former proved more effective in our tests. Then, several strategies were tested for the creation of new sounds (figure 1):

1. Divide the original sound into two parts (from 0 to 500ms, and from 500ms to 1s),
2. Divide the original sound into five superimposed parts,
3. Random mixing of two files: a 500 ms sample is made up of two 250 ms samples from two different files, preserving the temporal placement,
4. Interpolation of two original sounds, then division of the new sound into two parts.

The first strategy increases the size of the training dataset by 2, the second by 5, the third and last strategies enable to choose the desired dataset size.

Comparison of training performances between these strategies have shown that the second is the most effective one. Therefore, the second data augmentation strategy was chosen to train the network.

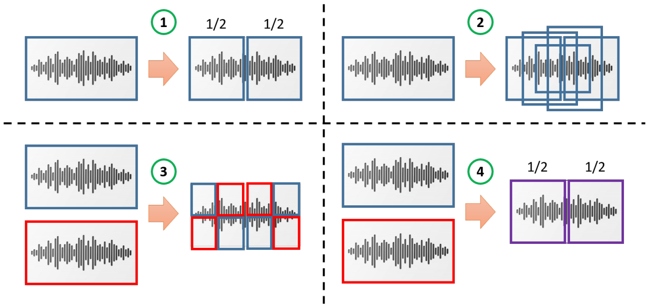

Figure 1: Data augmentation strategies studied.

3. MODEL SELECTION AND PRUNING

Given the constraints of the challenge, it is crucial to choose a neural network model that is efficient in terms of classification. Conventionally, each sound in the training dataset is converted into a mel-spectrogram, which is stored as an image. Initially in black and white, this image is transformed into a color image, ranging from green to blue to red as a function of amplitude, in order to increase contrast.

So, the network must perform well in terms of image classification. We chose a state-of-the-art network: YoloV8 [6]. The network has been pruned (structured pruning) and its internal parameters have been optimized so that the size of the network is compatible with the challenge rules. Figure 2 shows the network topology on a circle [7], traversed clockwise from 'input' to 'output' labels. On this diagram, dark blue dots represent Conv2D layers, green dots are for activation layers, yellow dots split the input tensors, and purple dots add or concatenate incoming tensors. The three last dots, near the output, represent the GlobalAveragePool, Flatten and fully connected layers of the classifier.

To preserve network performance, it was decided to train it in FP32. However, to respect the size constraint imposed by the challenge, the network would have needed to be reduced beyond its learning capacity. Post-training quantization in INT8 makes it possible to reduce the network's size by a factor of 4, which relaxes the constraint on the number of learnable parameters. The dimensions of the network were therefore chosen in order to obtain a number of parameters slightly lower than 128k: in INT8, this makes it possible to respect the size constraint. In our model, SiLU layers (Sigmoid Linear Unit) have been replaced by ReLU layers (Rectified Linear Units), which are very similar, because PyTorch does not currently support the quantization of Sigmoid related layers using x86 backend. The impact on performance is minimal.

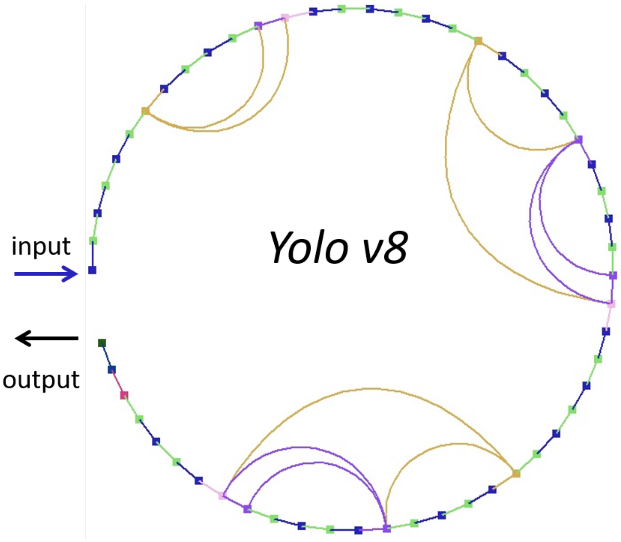

Figure 2: Yolo V8 network structure, with ReLU activation layers.

4. VOTING STRATEGY

The dimensions of the spectrogram images (number of horizontal and vertical pixels) used for training are chosen according to the number of MACs indicated in the challenge constraints. Indeed, the number of MACs of a network with a given image size is directly related to the product of the image dimensions. So, from a given number of MACs for a reference image size, it is very simple to choose the image size which allows the limit imposed by the challenge rules to be reached.

But, as mentioned above, we decided to implement a voting strategy to improve the recognition performance. The principle is to divide the sound used for the inference, lasting one second, into several portions each lasting 500ms, and to make the inference on each portion. A vote is then carried out in the event of disagreement between the different results.

However, each inference consumes MACs, which must be taken into account to respect the challenge constraint: one vote on two inferences implies that each inference does not exceed 15 MMACs, one vote on 3 inferences limits them to 10 MMACs, etc. And reducing the number of MACs for each inference implies reducing the size of the spectrogram images used, therefore reducing their semantic content. The tests showed that a vote on two inferences preserves the quality of the training images, while significantly improving the recognition performance.

Consequently, the chosen image dimensions are 80 pixels (in time - horizontal axis) and 320 pixels (in frequency - vertical axis).

Four different kinds of votes have been tested:

- SUM: sum over the predictions of each class,
- TOP1: count only Top1 results (top1 gets 1, others get 0). This is not really relevant for 2 inferences,
- ESPER: weighted sum over Top5 results (first gets 5, second gets 4, etc),
- MAJO: majority vote over the three other strategies.

For all tests, the SUM strategy has proven to be the most efficient.

5. QUANTIZATION

As mentioned above, the model was quantized in INT8, using PyTorch. Both Post Training Quantization (PTQ) and Quantization-Aware Training (QAT) were tested. Quantization has a side effect that can reduce classification performance. The class is calculated with PyTorch's Argmax function, which returns the index of the maximum value of the input tensor elements. But if this maximum value is present several times in the tensor, the index returned is the smallest one. For example, for the tensor [-19.4427, -11.5829, 0.8273, -7.8598, 6.6188, **11.9965**, -22.7521, 1.2410, **11.9965**, -11.1692] obtained for the sound file street traffic-milan-1087-40139-0-a, the correct indices are 5 and 8 but the value returned is 5, whereas the correct class is 8. This kind of error is very unlikely to occur with FP32-encoded numbers, but is not uncommon with INT8s.

For a tensor of size N quantized over b bits, the probability that the maximum value occurs several times is:

$$p = 1 - \frac{N}{2^{bN}} \sum_{k=0}^{2^b-1} k^{N-1}, \tag{1}$$

which is equal to 1.94% for $b = 8$ and $N = 10$. This can be approximated with a fairly good precision with:

$$p = \frac{N}{2^{b+1}}, \tag{2}$$

Voting on two inference results considerably reduces the consequences of this artifact and improves the performance of the quantized network, by enabling the correction of prediction errors.

6. RESULTS

According to NeSsi [8], the total number of parameters of the model is equal to 114988, and the total number of MACs is 13.51 millions. A vote on two inferences will then consume 27 MMACs.

The networks were trained on each split for 200 epochs, with the batch size varying with the size of the split. Each network is trained on a single split and never sees data from other splits. The performance of the FP32 networks was good when tested on the spectrogram images from the largest split, ranging from over 45% recognition for split 5 to over 90% on split 100 (table 1), showing very good generalisation capacity. However, against all expectations, performance collapsed when the networks were tested on the test split, dropping to around 35%. Data normalization and device simulation using impulse response from the MicIRP dataset were tested but did not show any improvement.

Split	FP32 model	FP32 model with normalization
5	47.0%	47.6%
10	54.5%	55.4%
25	66.6%	67.1%
50	78.1%	80.8%
100	93.2%	92.8%

Table 1: FP32 model performance on spectrogram image recognition, validation on images from split 100

Due to this performance drop, validation was performed on the complete training dataset, instead of the test dataset, using the SUM vote. Table 2 compares the performances of the full precision (FP32) and 8-bits quantized networks, using Post Training Quantization (PTQ) and Quantization-Aware Training (QAT), while performing inference on the sound files of the complete training dataset. The vote strategy enables to improve the classification performance from 2% for the smallest split to up to 6% for the largest one.

	FP32		PTQ		QAT	
Split	No vote	Vote	No vote	Vote	No vote	Vote
5	41.9%	43.7%	41.8%	43.5%	43.5%	44.8%
10	48.1%	50.0%	45.9%	48.1%	48.3%	50.3%
25	54.1%	57.9%	50.6%	53.7%	56.3%	59.3%
50	53.9%	58.9%	60.6%	64.8%	63.8%	68.0%
100	64.0%	70.4%	66.3%	72.4%	68.7%	74.6%

Table 2: Comparison of sound classification results on the full training dataset, using the SUM vote strategy.

Table 3 shows the final results of the quantized QAT model on the test dataset. Even if performance has dropped, voting can still improve results by up to 3%.

	QAT	
Split	No vote	Vote
5	30.8%	31.9%
10	33.6%	35.1%
25	36.2%	38.4%
50	37.5%	39.9%
100	38.0%	41.3%

Table 3: Sound classification results of QAT model on the test dataset, using the SUM vote strategy.

Table 4 compares the sizes of the saved unquantized and quantized models.

7. CONCLUSION

In this report, we have presented a new approach to Task 1 of the DCASE 2024 challenge, which focuses on the efficiency of the ASC

FP32	PTQ	QAT
498600	168440	169220

Table 4: Size of models in bytes

system. A YoloV8-based network was selected and pruned, and data augmentation and voting strategies were implemented using sound portions. Each sound is divided into two parts, allowing two inferences, and the sum of the two tensors is sent to a Softmax layer to calculate the probabilities of each class. The networks trained on the different splits were quantized in QAT and tested on the evaluation dataset. This strategy makes it possible to correct certain classification errors, at the cost of reducing the number of MACs allowed for each inference.

It is likely that the combination of the techniques presented here (sound partitioning, voting) with knowledge distillation of a large teacher network into a smaller one would make it possible to avoid the generalization problem observed and ultimately obtain much better classification results.

8. ACKNOWLEDGMENT

This research was funded in whole or in part by the French National Research Agency (ANR) under the Confluence project (ANR-23-EDIA-0003-03). With a view to its open access publication, the copyright holder applies a CC-BY open access license to any article accepted for publication resulting from this submission.

9. REFERENCES

[1] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-Efficient Low-Complexity Acoustic Scene Classification in the DCASE 2024 Challenge," 2024. [Online]. Available: https://arxiv.org/abs/1706.10006

[2] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 Challenge," 2022, _eprint: 2206.03835.

[3] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic Scene Classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[4] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[5] Y. Cai, P. Zhang, and S. Li, "TF-SepNet: An Efficient 1D Kernel Design in Cnns for Low-Complexity Acoustic Scene Classification," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 821–825.

[6] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] Chersi, F., "AIDGE: A Framework for Deep Neural Network Development, Training and Deployment on the Edge," in *EEAI23 - European Conference on EDGE AI Technologies and Applications*, Athens, Greece, Oct. 2023.

[8] A. Ancilotto, "NeSsi," 2003. [Online]. Available: https://github.com/AlbertoAncilotto/NeSsi