URBAN SOUND TAGGING DCASE 2019 CHALLENGE TASK 5

Technical Report

Linus Ng, Kenneth Ooi, Gan Woon Seng

Nanyang Technological University
School of Electrical and Electronic Engineering
Centre for Infocomm Technology, Singapore
linusng@ntu.edu.sg

ABSTRACT

Identifying urban noises and sounds is a challenging but important problem in the field of machine listening [1]. It enables and provides a realistic use case for detecting noises in an urbanised city - from noise complaints to detecting sounds or unusual noises that may indicate possible emergencies. The Urban Sound Tagging challenge as part of the DCASE 2019 challenge [2] [3] addresses the problem statement of urban noise control [1]. For this challenge, we are tasked to build a audio classifier to predict whether each of 23 sources of noise pollution is present or absent in a 10-second scene, as recorded by an acoustic sensor network. In this technical report, we will examine in some detail the performance of the audio classification models trained with different open external datasets.

Index Terms— Urban Sound Tagging, DCASE2019, Detection and Classification of Acoustic Scenes and Events 2019

1. INTRODUCTION

The impact of noise pollution on urban residents have proven effects on health [4] [5] [6]. However, the task to mitigate noise issues in an urbanised city is not easy. The first step towards this issue is to collect enough data points of noise detected for analysis - this also means that the classifiers have to be robust in order to consolidate the subsequent data points with data integrity [1]. A human can easily identify what noises are present in an acoustic scene if they listened intently, but it is still a difficult task for a computer to automatically recognize them. This is mainly because in the real world, there are too many different permutations and variations of how a sound can be produced. Nevertheless, the growing community that is aware of the consequences of noise pollution and the growing community of researchers in the field working on these issues hold promising outcomes for smart cities to address issues regarding noise pollution.

2. AUDIO EMBEDDINGS

The audio embeddings used to train all the models are extracted using the OpenL3 [7] open-source Python library. In consensus with the recent OpenL3 paper [7], our preliminary tests shows that the audio embeddings extracted from OpenL3 performs better than the VGGish [8] [9] audio embeddings when evaluating with the baseline system.

All embeddings extracted with OpenL3 are with the following configurations - mel256, emb_size 512 and content-type “env”.

3. DATASET

Our dataset used to train the models consists of the DCASE Challenge 2019: Urban Sound Tagging Task 5 development dataset, as well as audio files extracted from several sound classes from the following open external datasets:

- FSDKaggle2018 [10]
- FSDnoisy18k [11]
- UrbanSound8k [12]
- Urban-SED [13]
- ESC-50-master [14]

Note that the audio data extracted from the various sound classes of the open external datasets were stitched and split into 10-second audio files to fit the model training.

All of the 10-second audio files that were used for training the models were uploaded by us and can be downloaded from the following public Google Drive link [15] to achieve reproducible system outputs.

4. SUBMITTED MODELS

In this section, we will share the details of the four models we have submitted for the challenge. All four of the models were trained with different datasets and different model architecture configurations.

All of the annotated CSV files used to train the models are uploaded to our GitHub repository [16].

The steps to acquire our system outputs from the four models are shared at our GitHub repository [16] for system outputs reproducibility.

We will discuss in detail, the four models that were uploaded for submission, each, in the subsequent four subsections.

4.1. Model 1 - Re-Annotating the original annotations file

The first submitted model was trained with the original dataset, with our own annotations. Table 1 shows the summary of the model architecture trained with the original re-annotated dataset. The total number of training data used to train the model are the 2734 examples acquired from the development set and are evaluated with the 274 examples from the evaluation set.

•Annotator ID: 1001 and 1002 - manual annotations are done for all the audio files in the development set.

•Total training examples: 2734, 10-second audio files
•Total evaluation examples: 274, 10-second audio files

Table 1: Model 1 Architecture Summary

Layer type	Output Shape	Param
input(InputLayer)	(None, 512)	0
droppout_1 (Dropout)	(None, 512)	0
dense1 (Dense)	(None, 128)	65664
batch_normalization_1	(None, 128)	512
dropout_2 (Dropout)	(None, 128)	0
dense2 (Dense)	(None, 128)	16512
batch_normalization_2	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense3 (Dense)	(None, 128)	16512
batch_normalization_3	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense4 (Dense)	(None, 128)	16512
batch_normalization_4	(None, 128)	512
dropout_5 (Dropout)	(None, 128)	0
output (Dense)	(None, 23)	2967

Total params: 120,215
Trainable params: 119,191
Non-trainable params: 1,024

4.2. Model 2 - Appending manually annotated open external dataset audio files

The second model was trained with 2734 manually annotated audio files from subsection 4.1 of the development dataset and also with 71 individual manually annotated data of the data retrieved from the Urban-SED [13] and FSDKaggle2018 datasets [10].

22 examples were taken from the Urban-SED dataset and 49 examples were taken from the "Bus" class of FSDKaggle2018 dataset. The audio data extracted from the datasets were manually annotated and appended into the annotation file for training.

Table 2 shows the summary of the model architecture trained with the aforementioned dataset.

To preserve consistency in our annotations file, we have used a set of numerical values for the sensor ID column to correspond with the audio files retrieved from the datasets.

•Sensor ID: 99 - audio files extracted from the Urban-SED dataset.

•Sensor ID: 98 - audio files extracted from the "Bus" sound class of the FSDKaggle2018 dataset.

•Annotator ID : 1001, 1002 and 1003 - manual annotations are done for all the audio files.

•Total training examples: 2805, 10-second audio files
•Total evaluation examples: 274, 10-second audio files

4.3. Model 3 and 4 - Appending automatically annotated open external dataset audio files

The third and fourth model were trained with all the data used in subsections 4.1 and 4.2, and also with audio data retrieved from the open external datasets mentioned in section 3.

Both model 3 and 4 were trained on the same data with the same annotation file. The only difference between both models is

Table 2: Model 2 and 3 Architecture Summary

Layer type	Output Shape	Param
input(InputLayer)	(None, 512)	0
droppout_1 (Dropout)	(None, 512)	0
dense1 (Dense)	(None, 128)	65664
batch_normalization_1	(None, 128)	512
dropout_2 (Dropout)	(None, 128)	0
dense2 (Dense)	(None, 128)	16512
batch_normalization_2	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense3 (Dense)	(None, 128)	16512
batch_normalization_3	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
output (Dense)	(None, 23)	2967

Total params: 103,191
Trainable params: 102,423
Non-trainable params: 768

the model architecture.

Model 3 has the same architecture as model 2 which is reflected on Table 2.

Table 3 shows the summary of the model 4 architecture.

Table 3: Model 4 Architecture Summary

Layer type	Output Shape	Param
input(InputLayer)	(None, 512)	0
droppout_1 (Dropout)	(None, 512)	0
dense1 (Dense)	(None, 256)	131328
batch_normalization_1	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense2 (Dense)	(None, 256)	65792
batch_normalization_2	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense3 (Dense)	(None, 256)	65792
batch_normalization_3	(None, 256)	1024
dropout_4 (Dropout)	(None, 128)	0
output (Dense)	(None, 23)	5911

Total params: 271,895
Trainable params: 270,395
Non-trainable params: 1,536

The workflow of extracting the audio files from the datasets for training is shown in Figure 1.

It is important to note that the auto-generated CSV file used for training for this model is solely based on the sound class of the open external dataset and determined by us for which column in the annotation file we would append the presence value for all extracted files of that particular sound class. Our file extracting script has neither any machine learning algorithm nor intelligence to determine the presence of the sound classes by itself, given the audio files.

Audio data extracted from the FSDnoisy18k dataset were merged and listened by us to remove the misrepresented data, present in the nature of the dataset.

•Sensor ID: 50, 51 - audio files extracted from "Bark" sound class of the FSDKaggle2018 dataset.

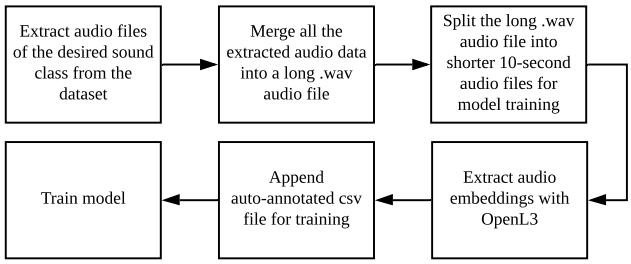

Figure 1: Workflow of extracting files from datasets with Python script for model training

•Sensor ID: 63 - audio files extracted from the "Bus" sound class of the FSDKaggle2018 dataset.

•Sensor ID: 80 - audio files extracted from the "siren" sound class of the UrbanSound8k dataset.

•Sensor ID: 81 - audio files extracted from the "car_horn" sound class of the UrbanSound8k dataset.

•Sensor ID: 95 - audio files extracted from the "chainsaw" sound class of the ESC-50-master dataset.

•Sensor ID: 82 - audio files extracted from the "siren" sound class of the ESC-50-master dataset.

•Sensor ID: 83 - audio files extracted from the "car_horn" sound class of the ESC-50-master dataset.

•Sensor ID: 6xx - audio files extracted from the "Engine" sound class of the FSDnoisy18k dataset and "Bus" sound class of the FSDKaggle2018 dataset.

•Sensor ID: 7xx - audio files extracted from the "street_music" sound class of the UrbanSound8k dataset.

•Annotator ID : 1000 - auto-annotations.

•Annotator ID : 1001, 1002 and 1003 - manual annotations.

•Total training examples: 6308, 10-second audio files

•Total evaluation examples: 274, 10-second audio files

5. VALIDATION RESULTS

In this section, we will discuss our validation results in comparison to the baseline system results. Note that all of the audio data evaluated in this section were from the validation set of the development set. None of the evaluation data were used to train nor validate for this section.

In the subsequent subsections, we will evaluate our models with the validation data of the development set. The models evaluated in subsections 5.1 to 5.4 were trained with their corresponding model architectures as described previously in subsections 4.1 to 4.3. Do note that the validation sets used for evaluating the results shown in subsections 5.1 to 5.4 were re-annotated by us, as mentioned previously in section 4.

5.1. Model 1 validation results

Model 1 training examples were as described in subsection 4.1, without the validation data as training data and the evaluation set. Table 1 reflects the model architecture of model 1. Table 4 and 5 show model 1 fine and coarse level results on the validation data of the development set.

•Total training examples: 2291, 10-second audio files

•Total validation examples: 443, 10-second audio files

Table 4: Model 1 - Fine-level evaluation

Fine level evaluation:
Micro AUPRC: 0.7047615615618702
Micro F1-score: 0.5309278350515464
Macro AUPRC: 0.4739863126531032
Coarse Tag AUPRC:
- 1: 0.7603033731060234
- 2: 0.36972981160775104
- 3: 0.3914868780011485
- 4: 0.37925108885540787
- 5: 0.6313846462995871
- 6: 0.39355065142143975
- 7: 0.8396896222453848
- 8: 0.026494429688083396
Coarse level evaluation:
Micro AUPRC: 0.8124516033687691
Micro F1-score: 0.5785714285714286
Macro AUPRC: 0.6144343516836789
Coarse Tag AUPRC:
- 1: 0.8905281046572762
- 2: 0.558817094724297
- 3: 0.3914868780011485
- 4: 0.7602208473108467
- 5: 0.729272005869799
- 6: 0.6170873877574039
- 7: 0.9415680654605765
- 8: 0.026494429688083396

Table 5: Model 1 - Coarse-level evaluation

Coarse level evaluation:
Micro AUPRC: 0.8305582811682136
Micro F1-score: 0.6966115051221434
Macro AUPRC: 0.6020365834510626
Coarse Tag AUPRC:
- 1: 0.9233794546753092
- 2: 0.38295817573755814
- 3: 0.4323199056372349
- 4: 0.7340061895291758
- 5: 0.7627549377095648
- 6: 0.6028305091241098
- 7: 0.9412784769458594
- 8: 0.03676501824968795

5.2. Model 2 validation results

Model 2 training examples were as described in subsection 4.2, without the validation data as training data and the evaluation set. Table 2 reflects the model architecture of model 2. Table 6 and 7 show model 2 fine and coarse level results on the validation data of the development set.

- Total training examples: 2362, 10-second audio files
- Total validation examples: 443, 10-second audio files

Table 6: Model 2 - Fine-level evaluation

Fine level evaluation:
Micro AUPRC: 0.70523230548944 **Micro F1-score:** 0.5421994884910486 **Macro AUPRC:** 0.484365014958172
Coarse Tag AUPRC:
- 1: 0.7616344015763287 - 2: 0.4048027224298613 - 3: 0.41714647789075465 - 4: 0.394432023107248 - 5: 0.6382064424335204 - 6: 0.3977618617064199 - 7: 0.8372787197280702 - 8: 0.023657470793173197
Coarse level evaluation:
Micro AUPRC: 0.8133226354213801 **Micro F1-score:** 0.5916740478299379 **Macro AUPRC:** 0.6288269890759782
Coarse Tag AUPRC:
- 1: 0.8926409656727983 - 2: 0.5829973703783178 - 3: 0.41714647789075465 - 4: 0.7992761878346581 - 5: 0.7342144733834602 - 6: 0.6387636297486623 - 7: 0.941919336906001 - 8: 0.023657470793173197

Table 7: Model 2 - Coarse-level evaluation

Coarse level evaluation:
Micro AUPRC: 0.8313466523196942 **Micro F1-score:** 0.695447409733124 **Macro AUPRC:** 0.6096578379978903
Coarse Tag AUPRC:
- 1: 0.920495231471101 - 2: 0.4328742943300048 - 3: 0.44364271805380967 - 4: 0.7232290404193488 - 5: 0.7655086202138199 - 6: 0.6173389470047207 - 7: 0.9417192822273209 - 8: 0.03245457026299661

5.3. Model 3 validation results

Model 3 training examples were as described in subsection 4.3, without the validation data as training data and the evaluation set. Table 2 reflects the model architecture of model 3. Table 8 and 9 show model 3 fine and coarse level results on the validation data of the development set.

- Total training examples: 5865, 10-second audio files
- Total validation examples: 443, 10-second audio files

Table 8: Model 3 - Fine-level evaluation

Fine level evaluation:
Micro AUPRC: 0.6945221832875335 **Micro F1-score:** 0.5141342756183747 **Macro AUPRC:** 0.4629859509853046
Coarse Tag AUPRC:
- 1: 0.7615672229607717 - 2: 0.31770891381459193 - 3: 0.39179005391119565 - 4: 0.35750943821148007 - 5: 0.6172214777158617 - 6: 0.39941599248878706 - 7: 0.8293271108729994 - 8: 0.02934739790674948
Coarse level evaluation:
Micro AUPRC: 0.8044603535827634 **Micro F1-score:** 0.5551470588235293 **Macro AUPRC:** 0.6047664181043608
Coarse Tag AUPRC:
- 1: 0.8892177365227486 - 2: 0.4943761200118349 - 3: 0.39179005391119565 - 4: 0.7496095804160442 - 5: 0.7048115516950683 - 6: 0.6389891383453257 - 7: 0.9399897660259199 - 8: 0.02934739790674948

Table 9: Model 3 - Coarse-level evaluation

Coarse level evaluation:
Micro AUPRC: 0.8223582862600867 **Micro F1-score:** 0.6866614048934491 **Macro AUPRC:** 0.6006143907062194
Coarse Tag AUPRC:
- 1: 0.9128796301870704 - 2: 0.40842386298041206 - 3: 0.4190758726279409 - 4: 0.7314555690540094 - 5: 0.7806091982457952 - 6: 0.5757221857024303 - 7: 0.9367107223596156 - 8: 0.040038084492480204

5.4. Model 4 validation results

Model 4 training examples were as described in subsection 4.3, without the validation data as training data and the evaluation set. Table 3 reflects the model architecture of model 4. Table 10 and 11 show model 4 fine and coarse level results on the validation data of the development set.

- Total training examples: 5865, 10-second audio files
- Total validation examples: 443, 10-second audio files

Table 10: Model 4 - Fine-level evaluation

Fine level evaluation:
Micro AUPRC: 0.7016436596169223 **Micro F1-score:** 0.5360824742268041 **Macro AUPRC:** 0.47471078030840796
Coarse Tag AUPRC:
- 1: 0.766574617408802 - 2: 0.33698853046878374 - 3: 0.4330119732859027 - 4: 0.3153054956939642 - 5: 0.6499850093496782 - 6: 0.4289514544907675 - 7: 0.8351856623632293 - 8: 0.03168349940613612
Coarse level evaluation:
Micro AUPRC: 0.8121814768000737 **Micro F1-score:** 0.5785714285714286 **Macro AUPRC:** 0.6105712232144662
Coarse Tag AUPRC:
- 1: 0.8917658433962752 - 2: 0.4913093431590879 - 3: 0.4330119732859027 - 4: 0.6994036610490526 - 5: 0.7368409487362989 - 6: 0.6571570896179902 - 7: 0.9433974270649857 - 8: 0.03168349940613612

Table 11: Model 4 - Coarse-level evaluation

Coarse level evaluation:
Micro AUPRC: 0.8332458192033845 **Micro F1-score:** 0.6934594168636723 **Macro AUPRC:** 0.6243610327220096
Coarse Tag AUPRC:
- 1: 0.9175266436049039 - 2: 0.5062138257614275 - 3: 0.44289505593495937 - 4: 0.749673692899699 - 5: 0.7832021256102135 - 6: 0.6095763682890174 - 7: 0.9388719843770116 - 8: 0.04692856529884437

5.5. Validation results - Summary

To provide a comparative evaluation of the validation results against the baseline system, we have evaluated the models with the default validation set. Tables 12 and 13 include the baseline system fine and coarse level results taken from the DCASE 2019 Task 5 [2] page. The terms under **Model** column: {*model_number*}*_orig_val* of tables 12 and 13 reflects our model's performance on the original validation set annotations (not re-annotated) to keep a consistent comparison against the baseline system. Lastly, evaluation with the re-annotated validation set at the last four rows of tables 12 and 13 for models 1 to 4, as shown in subsections 5.1 to 5.4.

Table 12: Fine-Level Validation Results Summary

Fine-Level evaluation			
Model	**Micro AUPRC**	**Micro F1-score**	**Macro AUPRC**
Baseline	67.17%	50.15%	42.75%
1_orig_val	71.42%	58.40%	46.39%
2_orig_val	71.59%	59.13%	47.43%
3_orig_val	71.20%	57.17%	45.20%
4_orig_val	71.75%	57.31%	46.72%
1	70.48%	53.09%	47.40%
2	70.52%	54.22%	48.44%
3	69.45%	51.41%	46.30%
4	70.16%	53.61%	47.47%
Coarse-Level evaluation			
Model	**Micro AUPRC**	**Micro F1-score**	**Macro AUPRC**
Baseline	74.25%	50.66%	52.97%
1_orig_val	73.89%	59.12%	57.57%
2_orig_val	74.07%	59.83%	58.53%
3_orig_val	73.94%	57.36%	56.78%
4_orig_val	74.47%	57.31%	57.49%
1	81.25%	57.86%	61.44%
2	81.33%	59.17%	62.88%
3	80.45%	55.51%	60.48%
4	81.22%	57.86%	61.06%

Table 13: Coarse-Level Validation Results Summary

Coarse-Level evaluation			
Model	**Micro AUPRC**	**Micro F1-score**	**Macro AUPRC**
Baseline	76.16%	67.41%	54.23%
1_orig_val	76.55%	68.42%	59.36%
2_orig_val	76.81%	67.59%	59.14%
3_orig_val	76.09%	66.83%	59.11%
4_orig_val	75.96%	67.10%	58.69%
1	83.06%	69.67%	60.20%
2	83.13%	69.54%	60.97%
3	82.24%	68.67%	60.06%
4	83.32%	69.35%	62.44%

6. EVALUATION RESULTS

As part of the challenge rules, the annotated CSV file for the evaluation set is kept private in order to perform a comparative evaluation of all competing systems in the Urban Sound Tagging challenge [2]. Therefore, the section for evaluation results would be added and modified when the results for our submitted system outputs are disclosed by the organisers.

7. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, Feb 2019.

[2] http://dcase.community/challenge2019/task-urban-sound-tagging/.

[3] http://dcase.community/challenge2019/.

[4] M. S. Hammer, T. K. Swinburn, and R. L. Neitzel, "Environmental noise pollution in the united states: Developing an effective public health response," *Environmental Health Perspectives*, vol. 122, no. 2, pp. 115–119, Feb. 2014. [Online]. Available: https://doi.org/10.1289/ehp.1307272

[5] A. Bronzaft and G. Van Ryzin, "Neighborhood noise and its consequences: Implications for tracking effectiveness of nyc revised noise code," 06 2019.

[6] M. Basner, W. Babisch, A. Davis, M. Brink, C. Clark, S. Janssen, and S. Stansfeld, "Auditory and non-auditory effects of noise on health," *The Lancet*, vol. 383, no. 9925, pp. 1325–1332, Apr. 2014. [Online]. Available: https://doi.org/10.1016/s0140-6736(13)61613-x

[7] J. Cramer, H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3852–3856.

[8] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://arxiv.org/abs/1609.09430

[9] J. F. Gemmeke, D. P. W. 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 *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[10] E. Fonseca, X. Favory, J. Pons, F. Font, M. Plakal, D. P. W. Ellis, and X. Serra, "Fsdkaggle2018," 2019. [Online]. Available: https://zenodo.org/record/2552860

[11] E. Fonseca, M. Collado, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Fsdnoisy18k," 2019. [Online]. Available: https://zenodo.org/record/2529934

[12] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2017.

[14] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," 2015. [Online]. Available: https://doi.org/10.7910/DVN/YDEPUT

[15] https://drive.google.com/file/d/1_FnfVw8AxwcKXHh1-rvvdloF9tfioY3U/view?usp=sharing.

[16] http://github.com/linusng/sonyc-ust-challenge-2019/.