URBAN SOUND CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS FOR DCASE 2020 CHALLENGE

Technical Report

Itxasne Díez Gaspón[*]

Noismart
C/ Ogoño nº1 5º Piso
Edificio Elkartegi
48930 Getxo, Las Arenas, Bizkaia
itxasne@noismart.com

Peio González, Ibon Saratxaga[†]

HiTZ Center – Aholab
University of the Basque Country UPV/EHU
1 Torres Quevedo Sq.
48013 Bilbao, Spain
ibon.saratxaga@ehu.eus

ABSTRACT

This technical report describes our system proposed for Task 5 - Urban Sound Tagging. The system has a core architecture based on Convolutional Neural Networks. This neural network uses log melspectrogram features as input and this input is processed by two CNN layers. The output of the convolutional stack is processed by several fully connected layers plus an output layer to produce the classification decision.

Spatiotemporal context data is also available and we propose a multi-input architecture, with two input branches that are merged for the final processing. The spatiotemporal context information is processed by an additional neural network of 2 fully connected layers. Its output is merged with the output of the CNN stack and the resulting data is fed to the fully connected output block. In this report, we describe the proposed models in detail and compare them to the baseline approach using the provided development datasets. Finally, we present the results obtained with the validation split from the dataset.

Index Terms— Urban Sound Tagging, CNN, DNN, multi-input

1. INTRODUCTION

In recent years, there has been an increase in the development of Smart Cities, where automated monitoring systems are intended to manage aspects such as traffic and pollution more efficiently.

One of the major challenges that researchers have to face is to detect segments of different sound events in large recordings obtained from continuously operating sensors deployed in the field.

For the last year, we have been working on the development of urban sound detection and classification system taking into account the research line of Piczak[1], using Convolutional Neural Networks for audio classification, and the research line of Bello et al[2] on the deployment of a sensor network in an urban environment, that integrates machine listening technics to process the audio automatically.

The objective of task 5 is aligned with this research line: it aims to detect and classify urban sound recordings using not only audio recordings but also contextual information of the recording environment: place, time, sensor, etc., the so-called spatiotemporal context (STC) data. We present two systems for this task. The first one is a model for classifying urban sounds using only audio information as input. The second one is a multi-input model that uses both audio and spatiotemporal context (STC) data as input.

The present report is divided in the following sections: in section 2, we describe the task in detail. In section 3, we present the proposed model, the experiments that have been done are described in section 4 and the results in section 5. The report ends with some conclusions.

2. TASK DESCRIPTION

Task-5, Urban Sound Tagging, aims to predict urban sound tags not only using audio signals but also spatiotemporal context data.

The task provides all the audio recordings files and metadata that gathers all the information about the recording time, localization and tag annotation. In the following paragraphs, we describe de audio and STC dataset, provided for the task.

2.1 Audio Dataset

The provided audios have been recorded using the sensor network deployed in New York[3]. All the audios were recorded with identical microphones, gain settings and a duration of 10 seconds with a sample rate of 48 KHz. The recordings are grouped into a train set with 13,538 recordings from 35 sensors, validate set with 4,308 recordings from 9 sensors and test set with 669 recordings of 48 sensors. For the evaluation, dataset DCASE also provides a new metadata with no tags. Train and validate set is used in the development stage while test set is used for obtaining final model

[*] This work has been supported by the Dept. of Economic Development and Infrastructure of the Basque Government (BIKAINTEK)

[†] This work has been partially supported by the Dept. of Education of the Basque Government code IT1355-19

results. The annotations related to de audio dataset are explained in paragraph 2.3.

2.2 Spatiotemporal Context Data

Along with the audio recordings, information about the place and time where they were carried out is also provided. This STC data includes spatial information: borough, block and lot (BBL) identifiers, latitude, longitude and temporal information quantized to hour level including week, and day information. Additionally, a unique identifier for the sensor of the recording is provided. The spatiotemporal data used for the baseline system provided by the organisation of the Challenge are latitude and longitude values, hour of the day, day of the week, and week. We will use also BBL information and sensor ID in one of our systems.

2.3 Labels

The sound event categories are divided in a two level taxonomy consisted of 23 fine-grained tags and 8-grained tags. Several events can occur simultaneously, and thus more than one tag can be assigned to the same recording. DCASE also provides a metadata file in which each row represents a multi-label annotation. The presence or absence of a tag is represented with a 0 or 1 respectively. In addition, each row includes an identification to distinguish the annotators: citizen science volunteer, SONYC team member and ground truth agreed upon the SONYC team.

3. PROPOSED ARCHITECTURE

The main aim of our work is to improve the detection and classification of urban sounds using audio data, for that reason, the common core of the architecture that we propose is based on a Convolutional Neural Network (CNN) for audio processing. Moreover, as task 5 proposed to include spatiotemporal context data for the prediction of tags, we also propose a second system using a multi-input model.

The multi-input model is fed with two different input data, log-melspectrogram for audio files and spatiotemporal context data corresponding to each audio file. The model is implemented with two input branches. The first one is the core architecture of a CNN stack that is fed with mel features. The second branch consists of two fully connected hidden layers that processes the spatiotemporal context data. The outputs of the two models are then merged and the resulting output data is fed to a final classification block. Figure 1 shows a diagram with the architecture of our model. The blue blocks represent the common core and the blue blocks together with the green blocks represent the multi-input architecture.

3.1 Data Preprocessing

The audio recordings have a length of 10 seconds, and they are too long to be entered in the neural network. Hence, the audios are fragmented into 1 second frames with a overlapping of 0.5s between the consecutive fragments.

The original sample rate of 48,000Hz is kept for the feature calculation. For each frame, we obtain log-Mel spectrogram with 128 Mel bands and a window length of 42ms and 21ms overlap. The mel-spectrograms are calculated using librosa library and each frame is ZScore normalized independently. The resulting tensor is fed to the CNN branch.

3.1.1. STC Preprocessing

Spatiotemporal data is used as the input of the STC branch. For the multi-input model, we have used two different configurations of the STC data. The first configuration (STC1 set) includes the same data used in the baseline provided by the organisation: latitude, longitude and time information. In the second configuration (STC2 set) we have added extra data such as sensor identifier, borough and block identifiers. Some of the variables of the STC data are one hot encoded and the other ones are ZScore normalized. STC data are replicated for the different fragments that come out of each recording.

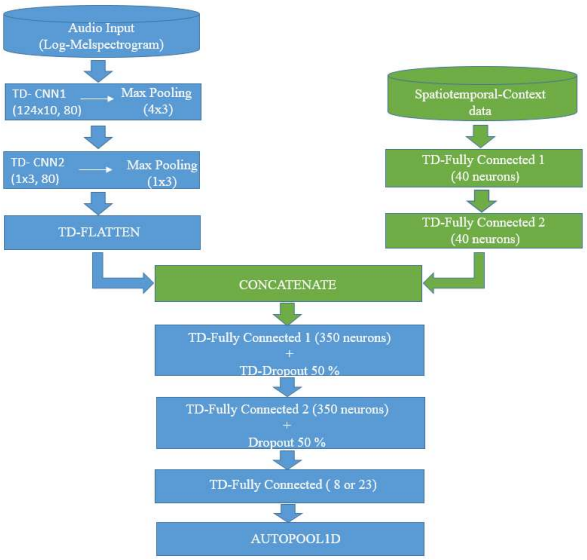

Figure 1. Proposed architecture

3.2 Model architecture

As explained before, we have evaluated two kind of architectures, described in the following sections.

3.1.2. Audio only architecture (A1)

The general core of the architecture is a CNN that uses as an input a 4-dimension tensor that is processed by two convolutional layers with 80 filters each. The kernel size of the first layer is (124 x10) and it is followed by a (4x3) max-pooling. We use a filter that covers almost all the frequency range of the spectrogram, but still it has some freedom to move in frequency in order to capture patterns corresponding to variable fundamental frequencies. The second convolutional layer has a kernel size of (1x3) and (1x3) max-pooling. The output of the second layer is flattened and fed to a block of two fully connected hidden layers with 350 neurons each

followed by an output layer that produces the classification scores. All convolutional and fully connected layers have relu activation, except the output layer that has a softmax activation. After each fully connected hidden layer we apply a 50% dropout.

In order to produce a single classification for the whole recording all the frames pertaining to the same recording are fed sequentially to the network using a Time Distributed wrapper in every layer. The output decisions for each of these fragments are pooled together using an adaptive pooling operator, the Autopool1D [4].

3.1.3. Audio+STC architecture (A2)

The Audio+STC architecture allows two different data inputs: audio data and spatiotemporal context data. The multi-input architecture consists of two different branches, a CNN branch for audio input and STC branch for spatiotemporal context data. The CNN branch consists of two convolutional layers with the same configuration of A1 for audio processing.

The STC branch consists of two fully connected hidden layers of 40 neurons each with relu activation. Before merging the output of the CNN branch with the output of the spatiotemporal context branch, we flatten the output of the CNN branch.

The merged data is used as the input of the final classification block that has the same configuration of A1: two fully connected hidden layers with 350 neurons each, relu activation, and an output layer with softmax activation for the classification. After each layer, we apply 50% dropout. In order to produce a single classification for the whole recording we also use Time Distributed layer and Autopooling1D.

3.3 Training

The models were trained using stochastic gradient descent as loss and Adam optimizer with a learning rate of 0.001. L2 regularization with a factor of 1e-5 it is also used. The development models were trained using early stopping with a patience value of 20, using validation loss as stop and 100 epochs. For the test train we used the validation data as training data, and early stopping of 10 epochs using the training loss and increased the epochs to 150 epochs. The annotation data used for the training and test stage is the same as used for the baseline. We use crowdsource annotation for training and then the ground truth for evaluation.

4. EXPERIMENTS

We carried out three experiments:

- Model 1: using the audio only architecture based on a CNN in which we use as an input the audio files and no spatiotemporal context data.
- Model 2: using the Audio+STC multi input architecture with two branches, so audio and spatiotemporal context data is processed by different branches. The STC data used as input is the previously described STC1 set, the same as the one used in the baseline.
- Model 3: using also the Audio+STC multi-input architecture. In this case, the STC2 set is used as input, which adds sensor identification, borough and block to the data in STC1.

We have trained each model using two labels. First, we trained it using coarse labels for predicting just coarse labels. In a second experiment, we trained the model using fine labels and obtained the coarse labels from the fine label prediction.

5. EVALUATION

During the development stage, we evaluated our models on the provided validation dataset. The challenge uses three different metrics for classification: micro-Area Under the Precision-Recall Curve (AUPRC), F1-Score and macro-AUPRC.

As the ranking of task 5, will take into account the macro-averaged AUPRC scores, we used it for choosing the best model for final test. For calculating the metrics, we use the scripts provided by DCASE task5.

The results of the baseline and the 3 proposed models are shown in table 1 and table2. Table 1 gathers the result obtained for coarse labels predictions using coarse labels for the training, and table 2 gathers the results obtained for fine labels predictions.

Table 1. AUPRC for the baseline system and our models. All three models values for coarse grained-labels

Tag names	*Baseline*	*Model 1*	*Model 2*	*Model 3*
engine	*0,6429*	0,6618	0,59	0,5186
machinery -impact	*0,5098*	0,3361	0,306	0,447
non-machinery-impact	*0,4474*	0,511	0,4858	0,4754
powered-saw	*0,5194*	0,3819	0,1961	0,2761
alert-signal	*0,8283*	0,8125	0,7214	0,7709
music	*0,3151*	0,1634	0,1378	0,2335
human-voice	*0,9073*	0,8666	0,8519	0,873
dog	*0,0568*	0,1392	0,1469	0,1483
Micro AUPRC	*0,7329*	0.7087	0,6504	0,6667
Micro F1-score(@0.5)	*0,6149*	0,5817	0,5346	0,5346
Macro AUPRC	*0,5278*	0,4844	0,4295	0,4679

Regarding the coarse-grained classifiers, as it can be seen in table 1, our best-performing model is model one that only uses as an input audio data. It does not outperform the baseline in the overall classification, but it has better results in three of the categories.

Concerning the model 3 that uses extra spatiotemporal context data, such as sensor identifier, borough and block, it can be said that improves the results of model 2 using just time and geographic coordinates but do not outperform Model 1, thus the addition of STC data in does not seem to improve the performance of the audio only system.

Table 2. AUPRC for the baseline system and our models. All three models values for fine grained-labels

Tag names	*Base-line*	Model 1	Model 2	Model 3
engine	*0,8500*	0,8437	0,7796	0,781
machinery -impact	*0,6021*	0,538	0,3406	0,3572
non-machinery-impact	*0,4192*	0,5157	0,5457	0,5306
powered-saw	*0,7200*	0,5433	0,6118	0,5609
alert-signal	*0,8518*	0,878	0,8437	0,8537
music	*0,6145*	0,2545	0,1935	0,3447
human-voice	*0,9593*	0,9447	0,9328	0,9424
dog	*0,0463*	0,2189	0,0497	0,2059
Micro AUPRC	*0,8352*	0,809	0,7739	0,795
Micro F1-score(@0.5)	*0,7389*	0,735	0,6738	0,7025
Macro AUPRC	*0,6323*	0,591	0,5372	0,572

The same remarks apply to fine grained classifiers. As it can be seen in table 2, our best model is obtained using only audio data. It does not outperform the baseline in the general classification, but it gets better results in some particular categories. Again the addition of STC does not improve the audio only results and in this case the benefits of using the extended STC data set (model 3) instead of the basic one (model 2) is less clear.

6. CONCLUSIONS

We proposed two different models as part of the task 5. An audio only architecture based on CNN for audio processing and a multi-input architecture for processing audio and STC data. The inputs used as input of the models were log-melspectrogram and spatio-temporal context data.

For fine labels classification, the best result that we obtain is using the model with no context data. Among the models, trained with both audio and context data, it can be said that model 3, that has extra spatiotemporal data is better than model 2.

For coarse labels classification, the results are similar. We obtain better result for the model with no spatiotemporal data. And we obtain better results for model 3 in the case of taking into account the spatiotemporal context data.

As general conclusion, it can be said that none of the models outperforms the baseline results that uses a pre-trained model. Further research is required using additional databases and improved architectures to obtain better performance and to take advantage of the use of contextual data.

7. REFERENCES

[1] K. J. Piczak, “Environmental Sound Classification with Convolutional Neural Networks,” *2015 IEEE Int. Work. Mach. Learn. Signal Process.*, 2015.

[2] J. P. Bello *et al.*, “SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise,” *Commun. ACM*, 2019.

[3] M. Cartwright *et al.*, “SONYC Urban Sound Tagging (SONYC-UST): A multilabel dataset froma an urban acoustic sensor network,” no. October, pp. 35–39, 2019.

[4] B. McFee, J. Salamon, and J. P. Bello, “Adaptive Pooling Operators for Weakly Labeled Sound Event Detection,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 11, pp. 2180–2193, Nov. 2018.