Inferring Precipitation from City Sounds

Daniel Kerrigan
New York University
Brooklyn, New York 11201, USA
djk525@nyu.edu

Eugene O'Friel
New York University
Brooklyn, New York 11201, USA
eugene.ofriel@nyu.edu

Abstract

In this paper, we construct binary classification models that predict precipitation using sounds from New York City. We tried four different representations of the audio data as model inputs: summary statistics of sound pressure level, predicted probabilities for coarse and fine grained urban sound labels, and audio embeddings. We achieved the best performance with the embeddings, with accuracy ranging from 79-87%. We also developed a tool to visualize the relationship between precipitation and the city's soundscape.

1. Introduction

The United Nations recently released a report on the urbanizing world. This report concluded that 55% of the world population already live in an urban environment and estimated that this number will increase to 68% within the next thirty years [8]. This trend of increasing urbanization highlights the importance of developing our understanding of urban informatics.

In any informatics field, access to proper data is paramount. Due to the nature of cities, this usually means relying on data collected by government agencies or private entities. However, such data could end up being too limited in scope. The alternative then is to build a sensor network to independently collect the necessary data.

That is precisely the case with the Sounds of New York City (SONYC) project [2, 7]. With its mandate to mitigate noise pollution, the SONYC project has set up a network of audio sensors around New York City in order to detect interesting auditory patterns. Whereas the research under the SONYC project umbrella focuses on noise pollution and labelling specific sound events, our research interest is tangential: How well can we synthesize the SONYC dataset with other datasets? In particular, we wanted to determine if we could use the SONYC dataset to train a binary classifier to predict whether or not an audio clip contains precipitation.

The rest of the paper is structured as follows. In section 2, we briefly discuss related work. In section 3, we detail the methods of our experimental process. In section 4, we go over our results. In section 5, we discuss our use of visualization to analyze the urban sound label data. In section 6, we summarize our results and state potential future work.

2. Related Work

Our goal to predict the presence of precipitation using audio-related data appears to be novel. Perhaps the closest piece of research is presented by Mydlarz *et al.* [6], which discusses using the SONYC dataset to predict potential sources of 311 noise complaints. Both this work and our work attempt to synthesize the SONYC dataset and an external dataset.

Our work leverages previous efforts of the SONYC project. In particular, as input to our models, we use predicted probabilities of the Urban Sound Tagging labels [3, 4] and audio embeddings [5]. These are discussed in more detail in sections 3.3.2 and 3.3.3.

3. Methods

3.1. Data Summary and Pre-Processing

We used two datasets in this experiment: the SONYC dataset and a precipitation dataset form the Iowa Environmental Mesonet [1]. The SONYC dataset is broken down by sensor, of which we only used 24 out of approximately 55 available due to poor data coverage in the rest. Each row of the SONYC dataset represents a 10 second interval of audio. Ideally, each sensor records for 10 seconds at random times over the course of a day, with an expectation of recording 30 seconds of each minute [6]. This would result in 4320 data points per sensor per day. A row comprises of: the timestamp, the sound pressure levels (SPLs) over the 10 seconds and its accompanying summary statistics, and the audio itself. For the precipitation dataset, there are only two relevant features: the timestamp and the cumulative precip-

[1] https://mesonet.agron.iastate.edu/request/download.phtml

itation amount over a given hour.

There are a couple notable points regarding these datasets. First, the finest reliable resolution available for the precipitation data was the cumulative precipitation amount in an hour interval. Considering the discrepancy between the temporal resolutions between this and the SONYC dataset, we made the assumption that if the precipitation dataset shows that it rained in a given hour, then it was raining for the whole hour. Furthermore, since we are performing a classification task, we converted the numerical value to binary outcomes: 1 for a nonzero amount of precipitation, 0 otherwise. In addition, we only used data from 2017.

The most important pre-processing step for the data is to join the two disparate datasets. This join is done as follows:

1. After sampling out roughly an even number of instances with and without precipitation from the precipitation dataset, match each such timestamp from the precipitation dataset to the closest timestamp in the SONYC dataset, for each sensor. The precipitation dataset's temporal resolution is hourly while the SONYC dataset's temporal resolution is in seconds. In general, this means that there will be a difference between the matched timestamps ranging from a few seconds to an upward of days. For example, if a sensor did not start taking data until March and the most recent instance of precipitation happened to be in early February, there would be a temporal difference of a few weeks.

2. Drop rows that have an absolute difference between the timestamp from the precipitation dataset and its own timestamp that is more than 30 seconds. This ensures that the SONYC data is as close to the recorded precipitation as possible.

3. Sample an equal number of rows with and without precipitation. For our experiment we sampled 19000 each, for a total of 38000 rows.

The first and last steps are particularly important to offset the existing precipitation imbalance. There is no precipitation for the majority of the year in New York City. As a consequence, we noticed in our earlier iterations of experiments that without an evenly distributed dataset, the models would overfit and always predict no precipitation.

Of the 38000 total instances, we used 70% for training and validation and 30% for testing.

3.2. Models

We trained and tested four types of classification models: multi-layer perceptron neural network, K-nearest neighbors, random forest, and support vector machine. We trained and tested each model using four different representations of the audio clips as input: summary statistics of the sound pressure level, predicted probabilities of coarse-grained urban sound labels, predicted probabilities of fine-grained urban sound labels, and audio embeddings.

We used scikit-learn to train and test the models [2]. For the models trained on the sound pressure level and the coarse and fine urban sound label probabilities, we performed 5-fold cross validation grid search to determine the best model parameters. The grid search was performed separately for each input data type. Our jobs performing the grid search for the audio embeddings timed out on the HPC cluster, so we used the parameters from earlier validation experiments with the coarse label probabilities. We have listed the parameters for each model and input data in table 5 in the Appendix.

3.3. Input Data

We tried using four different representations of the audio data to train and test our models.

3.3.1 Sound Pressure Level

Each audio clip has an 80-dimensional vector that measures the SPL throughout the clip. SPL is measured in A-weighted decibels. The SONYC team had calculated several summary statistics for this vector, including mean, standard deviation, entropy, and the L2-norm of the difference between consecutive measurements. We used these four summary statistics as input to our simplest models.

3.3.2 Urban Sound Tagging Predictions

The SONYC team has been using crowd sourcing to label the audio data for particular urban sounds [3, 4]. The labels have two granularities, coarse and fine. For example, the coarse labels include one label for engine sounds, where as the fine labels divide that into small, medium, and large engines. There are 8 coarse labels and 23 fine labels. See table 6 in the Appendix for the full list of coarse and fine labels.

The SONYC team has used this data to train models to do multi-label classification, which they used to predict the probabilities of the coarse and fine labels for each clip in the 2017 audio data. Therefore, for each instance in our training and testing data, we have an 8-dimensional vector for the predicted probabilities of the coarse labels and a 23-dimensional vector for the predicted probabilities of the fine labels.

3.3.3 Audio Embeddings

An audio embedding is a feature representation of an audio clip. These feature representations were generated by

[2] https://scikit-learn.org

an open-sourced implementation of L^3-Net[3], a deep neural network [1, 5].

We did not generate the audio embeddings ourselves. We used audio embeddings of SONYC dataset from 2017 that were already generated by a member of the SONYC project. The audio embeddings we used represents a 10 second audio clip by a 20×512 matrix, which we flattened to use as an input.

4. Evaluation

Since we have a binary classification task, we evaluated our models using standard classification metrics: the F_1 score, accuracy, and the confusion matrix.

4.1. Sound Pressure Level

The models trained using the four SPL summary statistics performed poorly, each having an accuracy of about 59%. The full results are show in table 1.

	MLP NN	KNN	RF	SVM
F_1 score:	0.55	0.54	0.54	0.56
Accuracy:	59.32%	59.92%	59.91%	59.78%

	MLP NN		KNN	
	True:	False:	True:	False:
Positive:	2821	1791	2734	1635
Negative:	3941	2847	4097	2934

	RF		SVM	
	True:	False:	True:	False:
Positive:	2719	1621	2867	1784
Negative:	4111	2949	3948	2801

Table 1. Results of the models trained on SPL statistics.

4.2. Urban Sound Tagging Predictions

When training and testing using the predicted probabilities of the coarse urban sound labels, our models achieved an accuracy interval of 65-70%. With the fine urban sound labels, we see an increase to 68-75%. The full results for the coarse labels are in table 2 and the fine labels in table 3.

4.3. Audio Embeddings

The binary classifiers trained with the audio embeddings give the best results, with an accuracy interval of 79-87%. For full results, please see table 4. Also note that there is a small but noticeable improvement among the models, in the increasing order: multi-layer perceptron, K-nearest neighbors, random forest, then support vector machines.

	MLP NN	KNN	RF	SVM
F_1 score:	0.67	0.63	0.69	0.66
Accuracy:	70.49%	65.62%	70.85%	69.36%

	MLP NN		KNN	
	True:	False:	True:	False:
Positive:	3362	1058	3392	1643
Negative:	4674	2306	4089	2276

	RF		SVM	
	True:	False:	True:	False:
Positive:	3794	1449	3417	1242
Negative:	4283	1874	4490	2251

Table 2. Results of the models trained on coarse label probabilities.

	MLP NN	KNN	RF	SVM
F_1 score:	0.70	0.67	0.74	0.71
Accuracy:	73.99%	68.52%	75.09%	73.11%

	MLP NN		KNN	
	True:	False:	True:	False:
Positive:	3482	779	3665	1586
Negative:	4953	2186	4146	2003

	RF		SVM	
	True:	False:	True:	False:
Positive:	3968	1140	3813	1210
Negative:	4592	1700	4522	1855

Table 3. Results of the models trained on fine label probabilities.

	MLP NN	KNN	RF	SVM
F_1 score:	0.74	0.81	0.80	0.86
Accuracy:	79.41%	81.26%	82.23%	87.03%

	MLP NN		KNN	
	True:	False:	True:	False:
Positive:	3391	70	4442	908
Negative:	5662	2277	4824	1226

	RF		SVM	
	True:	False:	True:	False:
Positive:	4180	538	4702	513
Negative:	5194	1488	5219	966

Table 4. Results of the models trained on audio embeddings.

5. Visualization

To complement our machine learning work, we developed a tool to explore the relationship between precipitation and the predicted probabilities for the Urban Sound Tagging labels [4]. The tool uses small multiples of histograms to compare the frequency of the predicted probabilities of a given label when there is precipitation versus when there is

[3] https://github.com/marl/openl3

[4] https://city-sounds-rain.now.sh/

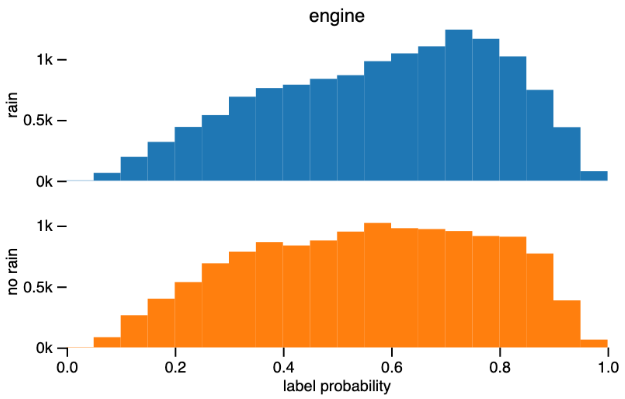


Figure 1. Comparison between the frequency of the predicted probabilities of the engine label when there is (blue) and is not (orange) precipitation.

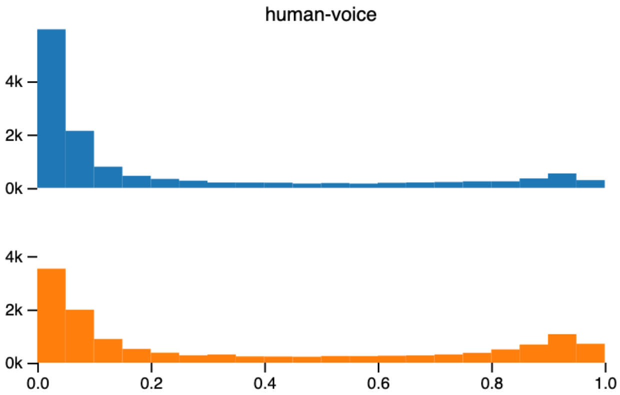


Figure 2. Comparison between the frequency of the predicted probabilities of the human voice label when there is (blue) and is not (orange) precipitation. Note the significantly higher frequency of low predicted probability when there is precipitation.

no precipitation.

Figure 1 shows the pair of histograms that correspond to the coarse “engine” label. The height of a blue bar corresponds to the number of training instances with precipitation and with a predicted probability for the engine label in the given range. The probabilities are split into 20 bins, each 5% wide. The orange bars have an equivalent interpretation for training instances with no precipitation.

We chose this design because we wanted to see if there were any labels where there were noticeable differences between the histogram for precipitation and the histogram for no precipitation. Figure 2 shows the pair of histograms for the coarse “human-voice” label. Note how there is a higher frequency of a low predicted probability of human voice when there is precipitation compared to when there is no precipitation. Conversely, there is a higher frequency of a high predicted probability of human voice when there is no precipitation compared to when there is. Intuitively, this makes sense since people are less likely to be outside when it is raining.

Figure 3 shows a screenshot of the complete tool, with one pair of histograms per coarse label. The user can also choose to show one pair per fine label. In addition, the tool lets users create one pair of histograms for each sensor for a particular label, as shown in figure 4. In this figure, the fine “person or small group talking” label is shown for each sensor. Many of the individual sensors exhibit the trend between precipitation and humans talking that was discussed with figure 2.

6. Conclusions

Overall, we consider our experiments a success, but there is room for improvement. We see a clear trend between the granularity of the input data and model performance. Given this, one avenue for improvement in this project would be working with the audio data itself, rather than indirect descriptions of the audio. We believe that this could lead to even better performing models.

Another area for improvement is that we could treat this problem as a regression problem rather than a classification problem. Intuitively, there is most likely a difference in audio between light and heavy precipitation.

Also, we implicitly made the simplifying assumption that each row is independent. We believe this is a reasonable assumption because the precipitation classification relies primarily on the content of the audio. However, we might be able to improve the predictions if we could introduce some time-dependent methods as well.

Lastly, it would be interesting to extend the visualization tool to help users interpret or explain the results of the models. For example, we could help the user explore what features are most important in determining that an instance has rain.

7. Project Resources

The code for our data processing and model training and testing can be found on GitHub at https://github.com/NYU-VisML-2020/CitySoundsWeather.

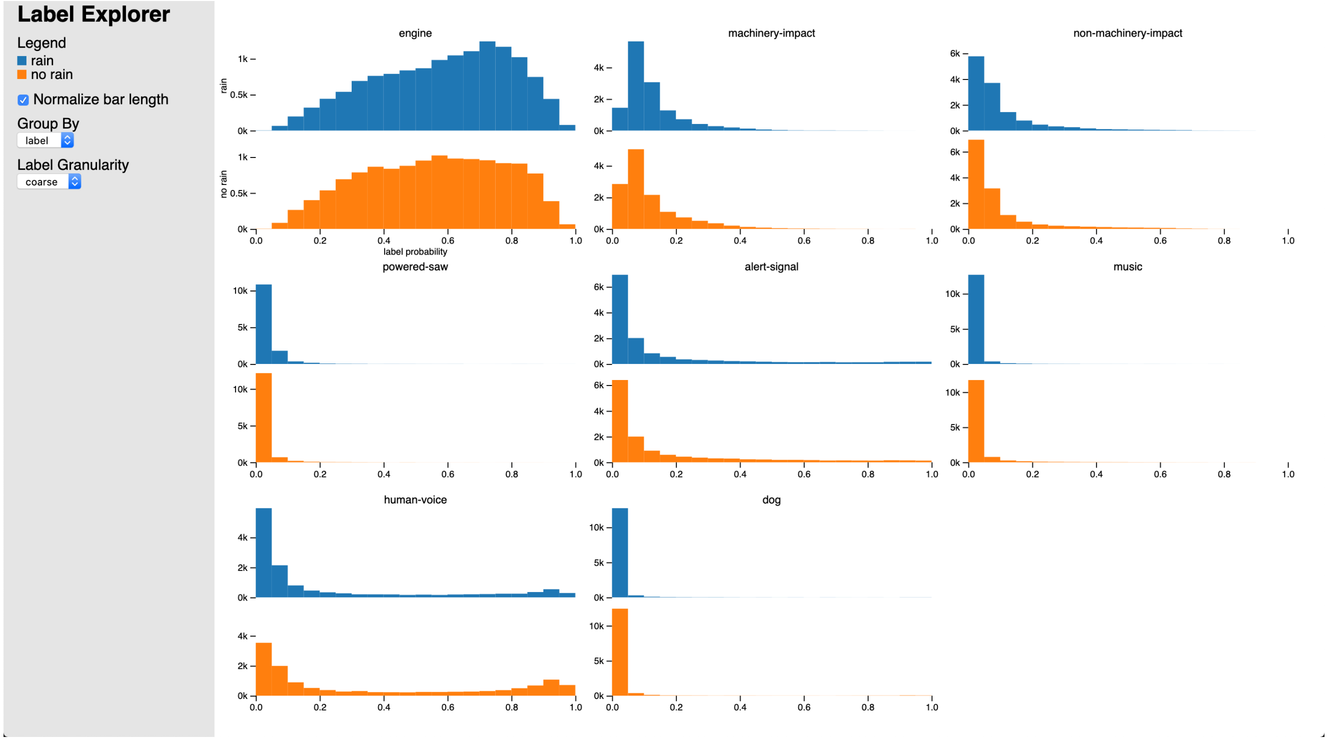
The code for the visualization tool is in a separate repository at https://gitlab.com/dkerriga/city-sounds-rain.

We are hosting a demo of the tool at https://city-sounds-rain.now.sh/.

References

[1] R. Arandjelović and A. Zisserman. Look, listen and learn, 2017.

[2] 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. *Commun. ACM*, 62(2):68–77, Jan. 2019.



Label Explorer

Legend

🟦 rain

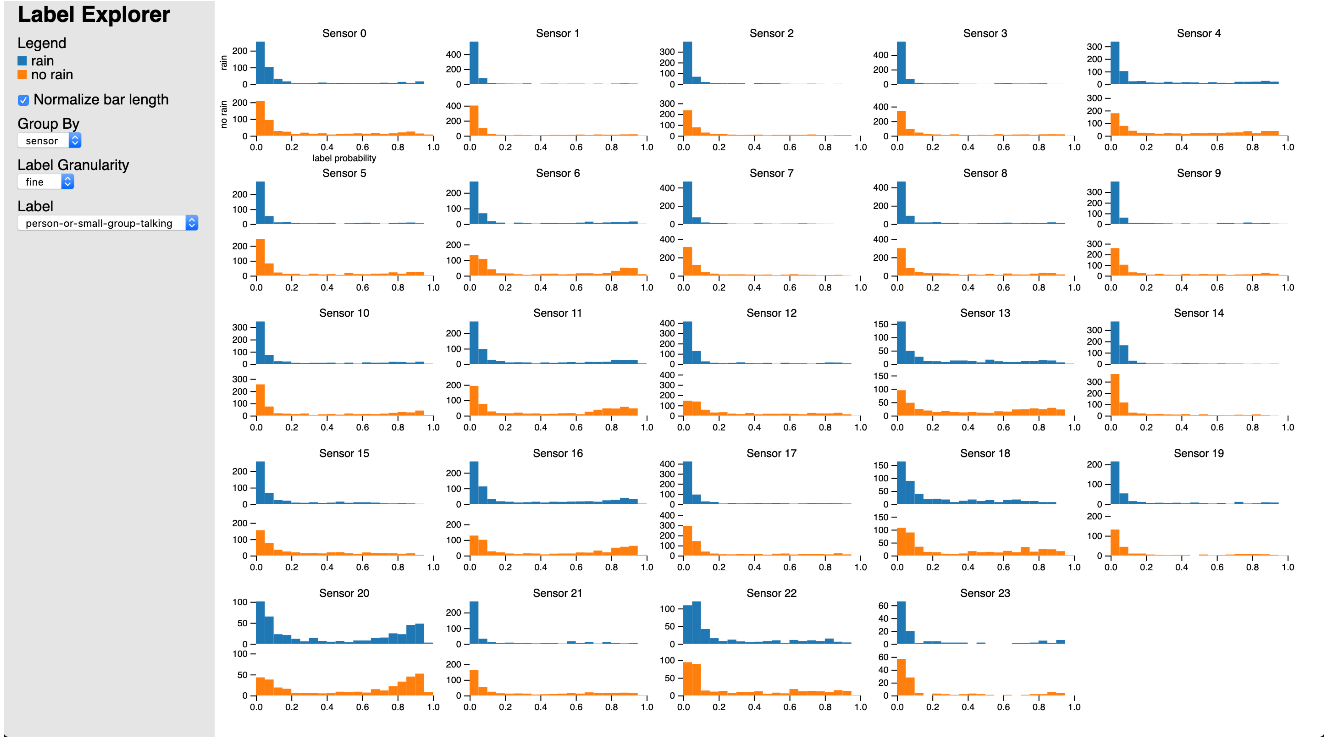
🟧 no rain

☑ Normalize bar length

Group By: label

Label Granularity: coarse

Figure 3. The interface of our visualization tool, with one pair of histograms per coarse label. In each pair, the blue histogram represents training instances with precipitation, and the orange without.



Label Explorer

Legend

🟦 rain

🟧 no rain

☑ Normalize bar length

Group By: sensor

Label Granularity: fine

Label: person-or-small-group-talking

Figure 4. The user can also select to show one pair of histograms for each sensor, for a particular label. In this figure, the “person or small group talking” label is shown. The sensor IDs have been anonymized.

[3] M. Cartwright, G. Dove, A. E. Méndez Méndez, J. P. Bello, and O. Nov. Crowdsourcing multi-label audio annotation tasks with citizen scientists. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[4] M. Cartwright, A. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello. Sonyc urban sound tagging (sonyc-ust): A multilabel dataset from an urban acoustic sensor network. pages 35–39, 10 2019.

[5] 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)*, pages 3852–3856, 2019.

[6] C. Mydlarz, C. Shamoon, and J. P. Bello. Noise monitoring and enforcement in new york city using a remote acoustic sensor network. *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, 255(2):5509–5520, 2017.

[7] C. Mydlarz, M. Sharma, Y. Lockerman, B. Steers, C. Silva, and J. P. Bello. The life of a new york city noise sensor network. *Sensors*, 19(6), 2019.

[8] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2018 Revision*. United Nations, New York, 2019.

Appendix

Model	**Parameter**	**SPL**	**Coarse Labels**	**Fine Labels**	**Embeddings**
Random Forest					
	`max_depth`	8	16	64	64
	`min_samples_split`	4	4	8	32
	`n_estimators`	200	400	300	200
SVM					
	`C`	0.5	500	500	10
	`gamma`	auto	scale	scale	scale
	`kernel`	rbf	rbf	rbf	rbf
KNN					
	`algorithm`	auto	auto	auto	auto
	`n_neighbors`	500	50	50	50
	`p`	1	1	1	1
	`weights`	distance	uniform	distance	distance
MLP					
	`activation`	tanh	relu	relu	relu
	`alpha`	0.001	0.0001	0.0001	0.001
	`hidden_layer_sizes`	(16, 16, 16)	(32, 64)	(32, 64)	(32, 64)
	`learning_rate_init`	0.001	0.001	0.01	0.001
	`solver`	adam	adam	adam	adam

Table 5. Parameters used for each model and input data type. For parameters not listed, the scikit-learn defaults were used.

engine	**alert-signal**
small-sounding-engine	car-horn
medium-sounding-engine	car-alarm
large-sounding-engine	siren
	reverse-beeper
machinery-impact	**human-voice**
rock-drill	person-or-small-group-talking
jackhammer	person-or-small-group-shouting
hoe-ram	large-crowd
pile-driver	amplified-speech
non-machinery-impact	**dog**
non-machinery-impact	dog-barking-whining
powered-saw	**music**
chainsaw	stationary-music
small-medium-rotating-saw	mobile-music
large-rotating-saw	ice-cream-truck

Table 6. Coarse (bold) and fine Urban Sound Tagging labels.