Building Binary Distortion Classifier on Large Unlabeled and Imbalanced Audio Dataset via Active Learning

Lizi Chen[1], Ana Elisa Mendez Mendez[2], and Yu Wang[2]

[1] *Courant Institute of Mathematical Sciences, New York University*
[2] *Music and Audio Research Laboratory, New York University*

Abstract

Building a classifier with a large unlabeled and imbalanced dataset can be challenging and require huge amount of human resources for annotation. In this project, we propose an active learning (AL) approach to the problem. As part of the Sounds of New York City (SONYC) project, we have collected 17-years worth of unlabeled urban sound recordings and have detected a specific kind of interference noise/distortion that needs to be classified across the entire dataset. We started with manually labeling small initial training and test sets. Then by using AL with a pool of 100,000 unlabeled examples, we obtained a precision of 95% and recall of 91% when training with only 68 labeled samples. Compared to the random sampling, recall improved by 11%. The results indicate that AL can be a good tool for a much more efficient and intelligent training and labeling process.

1 Introduction

Sounds of New York City (SONYC) [1] is an NSF funded project (Award Number: 1544753) [2] looking to provide technological solutions to four different problems:

1. The systematic, constant monitoring of noise pollution at city scale.
2. The accurate description of acoustic environments in terms of its composing sources.
3. Broadening citizen participation in noise reporting and mitigation.
4. Enabling city agencies to take effective, information-driven action for noise mitigation.

The focus of SONYC is to fight against noise pollution since it has become one of the biggest issues affecting quality of life in New York City. More specifically, 9 out of 10 adults in the city are exposed to noise beyond the limits of what is considered to be harmful by the Environmental Protection Agency (EPA). The SONYC team created a cyber-physical system as shown in Figure 1. This system includes a O2O(Online-to-Offline) distributed network of acoustic sensors and citizens for large-scale noise reporting. SONYC currently has 45 acoustic sensors deployed in the city constantly recording, and has collected about 17-years worth of audio data. Ultimately, these sensors will use machine listening methods to constantly provide a description of the acoustic environment. The collected information is analyzed and visualized by a cyber-infrastructure to identify important patterns of noise pollution that can be used by city agencies to deploy the necessary resources to act in the physical world.

To maintain the network properly running, we need to make sure that all of its components

are working correctly. One of the problems SONYC is currently facing is soft microphone failure detection on acoustic sensors. Hard microphone failures such as disconnections are easy to detect. However, soft failures such as audio distortion, internal/external noise, and degrading are hard to detect without actually listening to the recorded audio. One particular type of interference noise/distortion has been found in the recordings of multiple sensors. Being able to correctly identify these unusual, distorted signals is crucial for us to produce correct sound source labels and to find possible causes of this distortion. Therefore, this project aims to build an audio distortion binary classifier that can sort out distorted audio across all SONYC acoustic sensors. The classifier will also enable downstream analysis such as finding distortion patterns in historical recordings and detecting distortion in real time on the sensors.

Supervised learning is a standard method and works well for training a binary classifier; however, it requires large amount of labeled training and testing data to produce a robust model. SONYC has 17-years worth of audio data, but almost all of the recordings are unlabeled. Listening to every second of the recordings and labeling the audio would take a considerable amount of time and annotators. Moreover, in the SONYC dataset, we have much more normal (not distorted) audio than distorted audio (details in section 3.1), which makes labeling even harder. Therefore, although the model we are trying to build is relatively simple, the unlabeled and imbalanced nature of SONYC dataset requires a unique training approach.

For this project we propose using active learning, a semi-supervised machine learning method that queries for the label of the most informative instances to increase classification performance. With active learning we can train the classifier in a more efficient way with the least possible human resources.

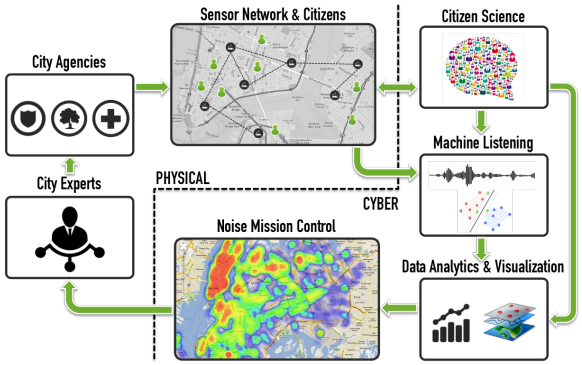


Figure 1: The SONYC Cyber-Physical System
Source: https://wp.nyu.edu/sonyc/

2 Related Work

As discussed in [3], active learning is a part of machine learning where the algorithm chooses the data from which it will learn from from an unlabeled data pool based on how uncertain it is about the available examples (queries). This technique can be used for tasks like speech recognition, information extraction, classification, and filtering. There are three different scenarios in which the learner can request queries: pool-based sampling, stream-based selective sampling and membership query synthesis. In this project we fo-

cus on the pool-based sampling strategy, which allows for a large amount of unlabeled data to be collected at once and assumes that we have a small set of labeled data. The stream-based selective sampling samples one unlabeled instance at a time from the actual distribution. After the instance is selected, the learner has to decide whether to request a label for it or discard it. Finally, in the membership query synthesis setting, the learner typically also generates samples from which it can also query.

Previous works by [4] and [5] talk about the benefits of active learning in sound classification for reducing annotation resources in the cases where datasets are too large to be labeled by humans. Zhao et al. [4] proposed a novel active learning method to save annotation efforts when preparing training data. They trained a classifer with the UrbanSound8k dataset [6], a dataset with 10 classes of urban sound, and used Mel-frequency cepstral coefficients (MFCCs) as input features. The resulted accuracy was 97%. Han et al. [5] used active learning and self-training with the same purpose: reducing human annotation efforts. Self-training is a semi-supervised learning (SSL) technique that uses a pre-trained model on a smaller set of labeled data to automatically annotate unlabeled data. The combination of both approaches greatly reduces human efforts in data annotation. The dataset used by the authors is unbalanced, but to obtain better results, they balanced it prior to training. The unweighted average recall (UAR) for the AL method was 69.3%, with 43% less of labeled data than when performing training with randomly sampled data. With the proposed SSAL approach, they obtained 10% more statistical significance than with AL only, while using 15.4% less data.

3 Methods

Figure 2 shows the framework of our proposed method. First, we train a binary classifier with initial training data. Then we build an unlabeled data pool from where the active learner searches to get the query which is most informative for learning. In each active learning loop, a human annotator listens to the queried audio example, label it and add it back to the training set. The model will then be retrained with updated training data and pool.

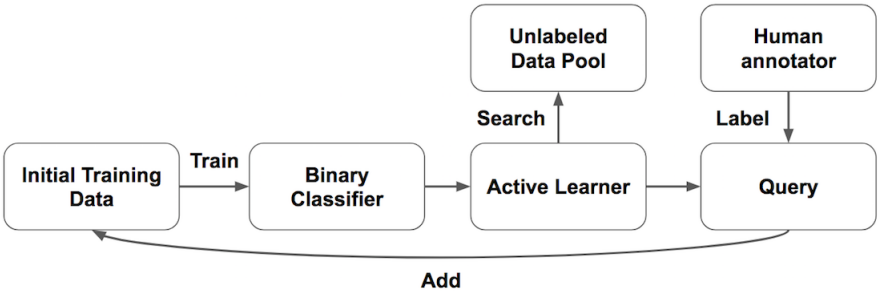


Figure 2: Framework of proposed method

3.1 Dataset

From previous work in machine listening and crowdsourcing annotation tasks, we have precomputed VGGish [7] embeddings for one tenth of the SONYC data. The original VGGish audio classification model; published by Google, was trained over the YouTube-8M dataset and can be used as a feature extractor to generate high-level 128-D embedding [8]. We

use these VGGish embeddings as the input features for our distortion classifier. Each embedding was computed from one-second audio segment and saved with information of recording sensor ID and timestamp.

Previous work also produced 150 clusters by running K-means algorithms; which for example, by listening to the 10 audio samples closest to each cluster centroid, produced clusters of engine sound, dog barking, and others. Within the 150 clusters, 17 clusters were found containing samples with the distortion that we are targeting. We estimate that $\approx 6.87\%$ of the recorded audio is distorted. This is a rough estimation since not all audio in those 17 clusters are guaranteed to be distorted, and not all audio in the rest of the clusters are guaranteed to be normal. We still can get the idea of the imbalanced nature of the dataset in terms of distorted/normal audio ratio, and we use such estimate to inform the following sampling process for initial training data.

3.1.1 Labeling Initial Training, Validation, and Test Data

Given the observation of clustering results mentioned above, we selected 15 sensors with the highest probability to contain distorted audio to work with. For each sensor, we listened to its recordings and manually labeled 20 positives (distorted audio) and 20 negatives (normal audio) one-second examples. Later on, we split the 15 sensors into 1:7:7 for training, validation and test sets, making sure that no sensor appears in more than one set. The reasons for keeping the initial training set small is so that we can see a clearer trend of how efficient active learning can be, as well as having more test data. We decided to start with the most extreme case for training, which is to only use two examples as initial training data (one positive and one negative). As a result, we get an initial training set of size 2, validation set of size 280, and a test set of size 280. All three sets are balanced.

3.1.2 Unlabeled Data Pool

We built an unlabeled data pool via randomly sampling 100,000 audio examples from 31 sensors that are not in the validation or test sets.

3.2 Binary Classifier

For the classifier, we use the random forest classifier implemented in Scikit-Learn with default parameters and 100 estimators.

3.3 Active Learning Framework

modAL [9] is an active learning framework built on top of scikit-learn, which is customizable and easy to work with. In this project, we use modAL as the base structure for the active learning loop. Then we modify and add additional functions in the loop to enable us to get the queries, retrieve the audio, listen to the audio, and manually input new labels during training.

modAL provides three basic uncertainty sampling strategies to query examples that the current model θ is the most uncertain about within the unlabeled data pool. Uncertainty is measured using label probability: $P_\theta(\hat{y}|x)$.

1. Least Confident (LC):
$$x^*_{LC} = \arg\max_{x \in R^d}(1 - P_\theta(\hat{y}|x))$$

2. Smallest Margin (SM):
$$x^*_{SM} = \arg\min_{x \in R^d}\big(P_\theta(\hat{y_i}|x) - P_\theta(\hat{y_j}|x)\big),\ i \neq j$$
$\hat{y_i}$ and $\hat{y_j}$ are the two most probable labels for x under the current model.

3. Label Entropy - to choose the data points whose label entropy is maximum:
$$x^*_{LE} = \arg\max_{x \in R^d} \sum P_\theta(\hat{y}|x) \log P_\theta(\hat{y}|x)$$

In this project, we use the Least Confident strategy. One of the possible future works would be experimenting with different query strategies and see how it affects model performance.

4 Experiments

4.1 Active Learning

First, we train a random forest classifier with initial training data of two examples. Following this, an Active Learner within modAL is created by specifying: initial classifier, unlabeled data pool, query strategy (Least Confident), number of instances retrieved in each iteration (1), and 100 iterations.

In each iteration:

1. The Active Learner returns one query result from the pool.
2. Human annotator retrieves and listens to the corresponding one-second audio, then assigns a label to it.
3. Remove the query from unlabeled data pool, and add it into training set with its label.
4. Retrain and save the classifier.

Figure 3 shows how the predicted probability (of being positive) on validation data changed with training iterations. The predicted probabilities for true positives and true negatives evolve from highly overlap to bimodal, improving the class separability. For each model, we experiment with different decision thresholds on validation data and record the best F_1 score. F_1 scores takes into account both precision and recall, which are important when evaluating on unbalanced dataset. The result in Figure 4 shows that with active learning, the F_1 score increases significantly within 20 iterations, which means when the learner has retrained 20 new training examples. The final chosen model is the one with the highest F_1 score 0.98 on the validation data, which occurs on iteration 66 with decision threshold of 0.33.

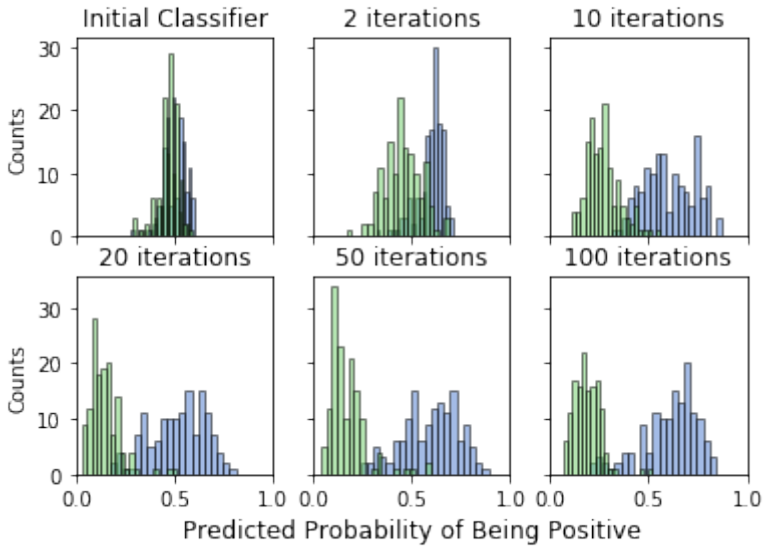


Figure 3: Predicted probabilities (of being positive) on validation data with different training iterations. Blue: true positives. Green: true negatives.

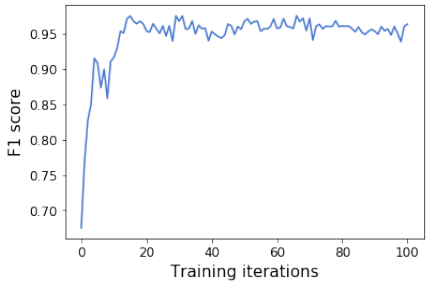


Figure 4: F_1 scores at different training iterations

4.2 Baseline Model: Random Sampling

As a baseline model for comparison, we also trained a classifier with the same amount of training data as in the active learning experiment, but with data randomly sampled instead of queried by active learner. Within randomly sampled examples, only 8% of them are positive due to the imbalanced nature of SONYC dataset. Figure 5 shows the resulting model predicted probabilities on validation data. There is a strong peak for true negatives. But for true positives, the predicted probabilities have a very broad range, indicating the model learned little about it. This could be attributed to the rareness of positive training examples.

Besides random sampling, we have also looked into some unsupervised learning methods. Although further investigation and experiments are required for more compatible comparison, the preliminary results in the Appendix show generally worse performance comparing to active learning.

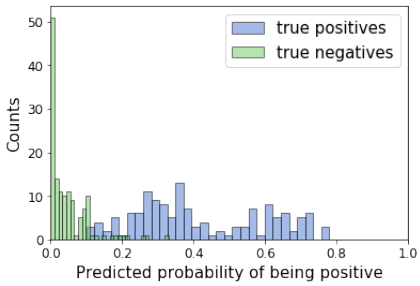


Figure 5: Predicted probabilities on validation data for random sampling model

5 Results and Discussion

5.1 Characteristics of Queries

One interesting observation we had during training is that the queries made by the active learner are much more balanced than the original dataset. Within 100 queries, we got 46% of positive examples, which is much higher than 8% for random sampling and 6.87% for our initial estimation. The other observation is that some of the queries are really challenging to label even for a human annotator. These characteristics indicate that active learner indeed was doing uncertainty informed search and making queries close to the decision boundary. These characteristics of the queries can be very useful for other tasks, such as labeling examples for extremely minority class, or building a challenging test set. Active learning can help with more efficient and intelligent labeling process.

5.2 Model Performances

Table 1 shows the resulted precision and recall on test data. Training with active learning achieves precision of 0.95 and recall of 0.91 with only 68 labeled data in total (2 initial training data, 66 new data queried by active learner). Which is very efficient comparing to the original size of the SONYC dataset. Also, active learning outperforms the baseline models: supervise learning with randomly sampled training data. The random sampling model has much lower recall, since it saw very little positive examples during training due to the imbalanced nature of the dataset.

	Test Precision	Test Recall
Active Learning	0.95	0.91
Random Sampling	0.94	0.80

Table 1: Precisions and recalls on test data.

5.3 Error Analysis

In order to understand how we can improve the model performance further, we listened to all misclassified examples. Most of them have predicted probabilities very close to the decision threshold. We also found that many false negatives contain multiple sound sources behind interference noise, such as siren, car horn, or people talking. This observation led us to think that VGGish features may provide much more information than we actually need, since the VGGish model was originally trained to classify multiple audio events. While

our goal is doing simple binary distortion classification, too much detail information about the audio segment can have negative effect. Therefore, one possible future work is to try using lower level features such as Short-time Fourier Transform (STFT) or Mel-frequency Cepstral Coefficients (MFCCs) as model input.

6 Conclusion

In this project, we try to build a generalized binary distortion classifier with a large unlabeled and imbalanced SONYC audio dataset. With the proposed active learning training framework, the resulted classifier achieves 0.95 precision and 0.91 recall when only trained on 68 labeled data. The result shows that by doing intelligent querying with active learning, we can train the model in a very efficient way. Also, compared to the randomly sampled learner, we achieved 11% more recall from the advantage of more balanced queried data from active learner. High recall is important in our case, since we would want to make sure the distorted audio files are filtered out before the following sound source classification is performed. The result of this project shows the potential of active learning to save huge amount of human resources when dealing with large unlabeled and imbalanced datasets. The method proposed in this project is not only useful for SONYC, but can also generally apply to other domains with similar challenges and tasks.

Appendix A Unsupervised Models

As part of the thinking process, we tried to look beyond Active Learning and research more methodologies that may help in tasks that are lacking labels. In this section, we show our investigations over Unsupervised visualization method (t-SNE [10]), Weakly Supervised Learning method (Label Propagation [11, 12]), and Unsupervised Learning method (K-Means).

Premise: *This part of work did not use the imbalanced dataset, except one case in K-Means. Rather, the work is mostly done with data sets from 15 clusters that has previously been observed with more distortion, meaning that conclusion and methods exploited in this section are based on presumably balanced data. Further investigation will be done with total random data selection.*

A.1 t-SNE

As a well-known dimension reduction algorithm, the t-SNE algorithm preserves the local distances of the high-dimensional data in mapping to low-dimensional data. Thus, we propose to use t-SNE to show the 128-D VGGish features in a 2D space. As an example, with a balanced pool[1] of 10000 unlabeled instances and 600 labeled instances (300 positive, 300 negative), we plot the t-SNE graph in figure 6. The green dots refer to positive points, the red spots refer to the negatives, and the blue dots are unlabeled. From left to right, the plots each have perplexity of 5, 20, and 50. We can see that the positive data is much easier to cluster when we set a higher perplexity value, whilst the negative instances are scattered consistently over the blue dots. However, there is no clear boundary that can be drawn from these plots to indicate the edge of the positive instance cluster. Such observation tells us that t-SNE does have the ability to extract the distortion features; however, in order to calculate actual measurement of the dimension reduction, we need help from another classification algorithm to split class edges.

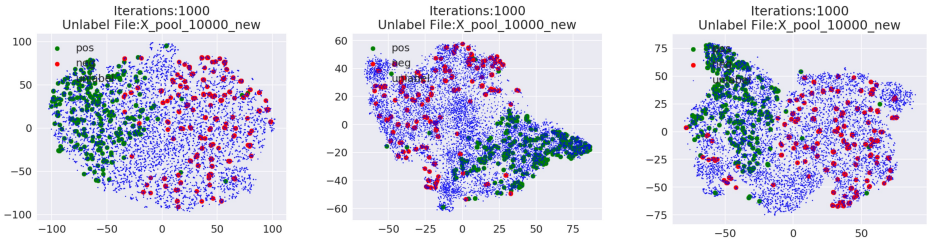


Figure 6: t-SNE plot over balanced pool
10000 unlabeled (presumably balanced), 300 positive, 300 negative.
Perplexity: [5, 20, 50]

A.2 Label Propagation

Inspired by t-SNE plots, we use a Graph-based approach to find the clustering of each class. Our assumption is that if x_i and x_j are close in the same high density region, y_i is the same as y_j. Therefore, given the low-dimensional (2D) result data from t-SNE, we

[1]Due to time constrain, results of total random data is not published in this report.

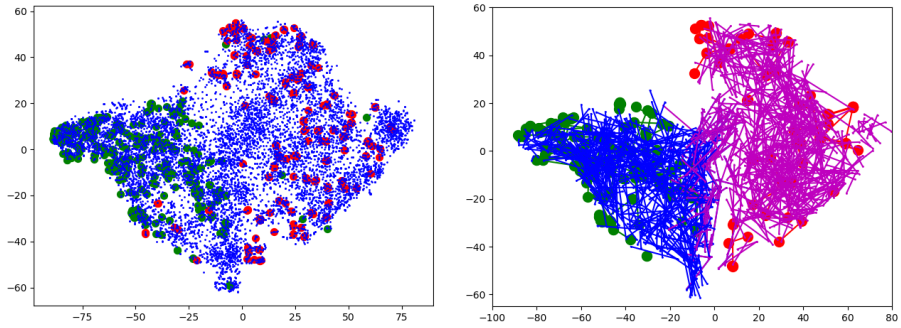
construct an N-by-N affinity matrix $\mathbf{P}$ and an N-by-2 label matrix $\mathbf{F} = [\mathbf{F}_{labeled}, \mathbf{F}_{unlabeled}]$, where

$$p_{ij} = \frac{D_{ij}}{D_{i1} + D_{i2} + \ldots + D_{ik}} = \frac{D_{ij}}{\sum_{k=1}^{n} D_{ik}}, \text{ where } D_{ij} \text{ is the cosine distance.}$$

$$\text{Labeled instance: } f_i = [\mathbf{I}_{(y_i=1)}, \mathbf{I}_{(y_i=1)}]$$

$$\text{Unlabeled instance: } f = [-1, -1]$$

In 500 iterations of propagation $\mathbf{F}^{t+1} = \mathbf{P} \cdot \mathbf{F}^t$ and true label clamping, we can cluster positive unlabeled data over the labeled instances, as shown in Figure 7. On the 100k balanced pool, a 1k iterations training of Label Propagation algorithm results in 0.88 precision and 0.83 recall.



(a) t-SNE result plot (b) Label Propagation

Figure 7: Run Label Propagation over low-dimensional data in t-SNE space.

A.3 K-Means

K-Means produces insight in how critical it is to construct the unlabeled pool. By running K-Means over different unlabeled data pools, figure 8 demonstrates how different unlabeled pools result differently. The blue, green and orange lines yielded from balanced pools, and the red line is from absolute random selected data. Results are given in the following section.

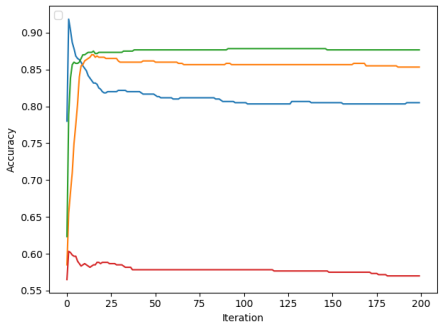


Figure 8: K-Means over different unlabeled data pools.
Blue: 1k balanced, Orange: 10k balanced, Green: 100k balanced, Red: 100k Random.

A.4 Result

	Test Precision	Test Recall
Label-Propagation on t-SNE result	0.54	0.41
Unsupervised Learning (K-means)	0.61	0.55

Table 2: Precisions and recalls with Cross Validation on 100k Random Selected Data.

A.5 Unsupervised Learning Summary

1. t-SNE result is mainly for visualization. We can also use other classification or clustering algorithms on such low-dimensional data to actually construct a programmatic-usable classifier.
2. Results from both Label Propagation and naive K-Means do show gradual progression yet not as good as Active Learning.
3. As shown in Figure 8, Unsupervised Learning algorithm K-Means heavily relies on the structure of data, a dataset with random distribution versus consistent prior distribution can produce drastically different results.
4. Further fine-tuning is needed unless 1) current accuracy rate is tolerable, 2) human annotation can improve no more or is beyond budget.

References

[1] "SONYC, Sound of New York City," 2018. https://wp.nyu.edu/sonyc/.

[2] D. Corman, "NSF award search: Award number: 1544753 - CPS: Frontier: SONYC: A Cyber-Physical System for Monitoring, Analysis and Mitigation of Urban Noise Pollution," Aug 2016.

[3] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[4] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 751–755, March 2017.

[5] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments.," *PLoS ONE*, vol. 11, no. 9, pp. 1 – 23, 2016.

[6] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), pp. 1041–1044, ACM, 2014.

[7] "VGGish." Available at https://github.com/tensorflow/models/tree/master/research/audioset#vggish.

[8] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *CoRR*, vol. abs/1609.08675, 2016.

[9] D. Tivadar, "modAL, A modular active learning framework for python3," 2018. Available at https://cosmic-cortex.github.io/modAL/.

[10] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[11] X. Zhu, J. Lafferty, and R. Rosenfeld, *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2005.

[12] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*. The MIT Press, 1st ed., 2010.