Practicable Soundmapping: JavaScript enabled Edge Compute

Tae Hong Park
New York University
New York, United States
thp1@nyu.edu

Minjoon Yoo
New York University
New York, United States
minjoon.yoo@nyu.edu

ABSTRACT

In this paper, we report on developments of the Citygram, a comprehensive sensor network platform for capturing, streaming, analyzing, mapping, visualizing, and providing easy access to spatiotemporal soundscape data. Launched in 2011, Citygram's recent strategic decision has resulted in system redesign to enable migration from a codebase built on multiple computer languages to a cross-platform single *JavaScript* codebase. Citygram now runs on V8 engines including standard web-browsers and in the *node.js* environment. The Citygram sensor network significantly alleviates problems concerning soundmapping complexities including operating system limitations, hardware dependency, software update and dissemination issues, data access mechanism, data visualization, and cost. This strategy has made practicable a key design philosophy for soundmapping: rapid sensor network growth for capturing spatiotemporally granular soundscapes. We summarize research and development for the following Citygram modules (1) cost-effective sensor module allowing high-value data transmission through edge compute paradigms, (2) machine learning module focusing on environmental sound classification, and (3) visualization and data access prototypes.

1. INTRODUCTION

In 2011, Citygram (CG) [1]–[10] was launched to address digital cartography through the lens of soundscape research. Several key technological advances over the past decade have made for a compelling opportunity to create such a soundmapping system: (a) the evolution of powerful, ubiquitous computing devices for citizen-science engagement and activation, (b) the feasibility of applying such devices as high-quality/cost-effective sensor hardware, (c) ubiquitous communication networks, and (d) the advancement of sound analysis techniques. Additionally, with the rapid maturation of *JavaScript*, *node.js*, and the notion of web-browsers as universal operating systems, we have recently redesigned our system and migrated our multi-language codebase to a *JavaScript* platform that facilitates capture, analysis, visualization, archival, and public access to soundscape data. Over the last few years, we have narrowed our research scope where the current focus is on urban "noisecapes" and the development of a cyber-physical systems (CPS) for capturing and sensing noise pollution.

In this paper, we report on the Citygram project which includes four main modules: (1) sensor network and edge compute, (2) analysis and machine learning, (3) visualization, and (4) citizen science participation as summarized in Section 2. The paper begins with an overview of the research context and brief literature review.

1.1 Research Context: Noise Pollution

Environmental footprints left by human and machine activity contribute in defining modern urban environments: air and ground transportation, construction, and leisure related activities are a very small number of such examples, all of which emit energy. Noise pollution is one of such omnipresent environmental spatiotemporal energy sources that urban communities have learned to deal with. Learning to deal with it, however, comes with serious associated health risks. "It means you've adapted to the noise … you're using energy to cope with the situation. That's wear and tear on your body" [11] says Bronzaft, a leading expert on mental and physical health effects of noise. Studies show that such "wear and tear" does not just cause hearing impairment but they can also lead to physiological disorders including adverse childhood learning, hypertension, sleep deprivation, gastrointestinal, cardiovascular, cardiopulmonary morbidity [12]–[22] as well as reduced task performance, work productivity, and social behavior [23]–[27]. This situation has only worsened with expanding megacities worldwide that are now, for the first time in human history, inhabited by more than half of the world's population. By 2050, 3/5 of the global population is expected to live in such megacities, generating noise pollution from under-, on-, and above-ground human activity.

1.2 Related Work

In recent years, a variety of application workflows have been developed that leverage advances in telecommunications and computer science research to attempt noise measurements in cities. Some applications leverage mobile telecommunication devices in the context of citizen-science and represent models evoke the notion of geographic information system (GIS) evolving within a participatory framework to encourage civic engagement with community issues on the environment [28]. For example, in 2008, Sony developed *NoiseTube* [29] to measure personal exposure to noise using smartphones with the long-term goal of developing static crowd-sourced noise maps of urban areas. *NoiseTube* attempted to address the "lack of public involvement in the management of the commons" by empowering anyone with a smartphone to measure their exposure to noise pollution. As an instrumentation system, the application was simple: uploading dB SPL levels that would then be accessible via non-real-time online maps. Similar projects have followed, incrementally improving both on instrumentation and visualization. The mobile application *WideNoise* (2009) achieved significant global usage in metering noise exposure enabled by citizen-scientists [30]. *WideNoise* improved on *NoiseTube* in several respects: development of wide user base with a fully developed social user experience encouraging active participation. In addition, *WideNoise* allowed noise sample tagging features. Both *NoiseTube* and *WideNoise* have used dB SPL for noise measurement along with a simple calibration routine to balance the different dynamic responses of various smartphone models. A similar project was undertaken by *Motivity* in San Francisco (2010) using a small number of stationary decibel meters at key intersections [31]. Developed as an acoustic ecology project to demonstrate the efficacy high-traffic area noise metering, the project used an instrumentation system consisting of a small computer and microphone, placed at intersections within a 25-block San Francisco area. Similar to the aforementioned projects, *TenderNoise* also has employed the SPL metric, in line with standard noise measuring practice. Other commercially available noise monitoring examples include the *NoiseMap* [32] and *Tmote* Invent [33] (discontinued). Both have used a wireless sample-and-upload SPL level mechanism for noise monitoring. *Tmote* also put emphasis on efficacy of remote sensor power consumption, which resulted in significant power savings but at the cost of system latency. Another recent example is Sensor City [34] which aims to deploy hundreds of sensing units equipped with acoustic monitoring equipment around a small city in the Netherlands. This solution utilizes its own dedicated fiber-optic network and high-end calibrated audio recording devices. The project is taking a soundscape analysis approach and is looking to investigate human perception and evaluation of acoustic environments within urban settings. One of the most recent projects is *SONYC*, which is built on Citygram technologies and designs [2, 4, 5, 35, 36] and has gone through a number of name changes since its launch in mid/late 2013 with Park as project lead under the name *Noise*, *Sound*, *Citygram-Sound*, and now SONYC. Some observations that can be made from the above examples are: (1) projects approach a dynamic problem through a static solution: an audio-snapshotting, upload, and play strategy; (2) issues exist with reliability vs. granularity: systems utilize either crowd-sourcing methods for limited operating systems and platforms and/or cost-prohibitive custom systems, (3) projects are often not comprehensive: data capture, machine learning, visualization, access, and archiving are not simultaneously addressed; and (4) systems use platform dependent software and hardware.

2. THE CITYGRAM SYSTEM

Citygram is a comprehensive soundmapping system that includes the following modules: (1) sensor network: edge compute sensors and server technologies, (2) soundscape analysis: signal processing, feature extraction, and machine learning, (3) visualization: public interface for spatiotemporal data exploration and access, and (4) citizen science: technologies and designs to enable community participation and effective sensor network growth. With our system migration to the *JavaScript* platform, all of the modules are currently being developed to run on the web-browser and node.js platforms. This strategy has enabled Citygram software compatibility with most standard operating systems including *OS X*, *Linux*, *Windows*, *Android*, and partially on *iOS* (audio recording is currently disabled on Apple mobile operating systems).

Web Audio Conference WAC-2017, August 21–23, 2017, London, UK.

2.1 Plug-and-Sense Sensor Network

Our sensor network architecture has been developed from a top-down approach, with theory/design at the top and system building at the bottom guided by *plug-and-sense* design principles to make citizen-scientist and community participation practicable. Our current outputs [2, 4, 10] include hardware and pure software sensor prototypes. Hardware solutions are designed to run continuously as dedicated, standalone devices while software-only solutions are designed to run entirely on standard web-browsers to maximize community participation.

Each remote sensing station that captures, processes and analyzes data at the "edge" has been developed to address key issues associated with traditional sensor network designs: (1) *ease of deployment*, (2) *sparseness*: coverage of large areas with a small number of physically unwieldy and costly sensor instrumentation, (3) *reliability*: shortcomings resulting from overreliance on consumer-grade handheld devices, (4) *cost*, and (5) *dynamicity*: real-time, *event triggered* sensor network for smart bandwidth usage and efficient power consumption. Software is updated remotely, facilitating maintenance, software evolution, and eliminating the need for physical, onsite updates. All of the audio recording, analysis, data transmission, server, and data exploration modules are written in *JavaScript* and utilize libraries such as *Web Audio*, *WebSocket*, *d3.js*, *AngularJS*, as well as basic HTML containers. We initially used *Web Workers* threads for signal processing and analytics but this proved to be detrimental to memory management on single-board computers. We therefore implemented our own quasi-threading system resulting in 10 – 12 times less memory usage. Audio I/O for web-browsers and node.js have been implemented separately – for web-browsers we use W3C Web Audio and for node.js, we use *mic,js* for audio recording based on `sox` and `arecord` command line utilities for Mac/Windows and Linux respectively.

Citygram (CG) currently includes a number of prototype sensor software and hardware solutions designed for easy deployed by non-experts under the *plug-and-sense* (PnS) design philosophy. The most fundamental Citygram PnS solution is the standard web-browser. Two CG PnS versions exist (1) fully functional user portal: allows full user exploration, soundscape sensing, soundscape analysis, user account management, and data down-streaming to client computer for personal use (e.g. sonification through Max or Supercollider as further described in Section 2.3), (2) *Simpler*: a lightweight version of the full CG exploration portal that only includes essential modules for data streaming and analysis with a stripped-down user interface as shown in Figure 1.

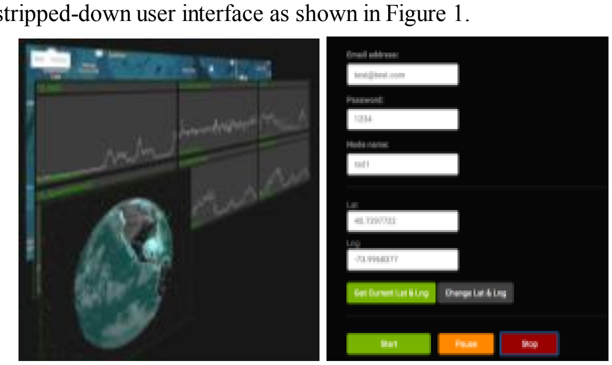

Figure 1. Full CG portal (L), lightweight sensor "node" (R).

The web-based soundscape sensor solution runs entirely on standard web-browsers, effectively allowing anyone with web-browser, microphone, and a personal computing device (e.g. mobile device, laptop, and desktop computer) to participate as an environmental edge compute node.

Additionally, as a standalone sensing edge compute node, most single board hardware devices (as well as "old" computers) can be used for soundscape sensing due to the flexibility of *JavaScript*. Our current fundamental hardware solution is based on inexpensive, configurable single-board Raspberry Pis (RPi) that are applicable to various environmental sensing situations running in the node.js or headless and standard chrome browser environments (e.g. *Raspbian*): single board computers like RPis are ideal for continuous streaming, autonomous operation without manual intervention, while performing analytics at the edge. CG PnS solutions perform computation and analysis necessary to reduce high-volume, low-value data and transmit high-value, low-bandwidth data from the edge to the cloud. This capability is a key enabler of data collection, and development of insights across a multitude of sensor devices at scale.

We are also currently developing a hardware PnS solution we call *window sensor*, which is designed to exploit the most common structural components in homes – windows. The window sensors eliminate the need for outdoor power and data cables as they are secured to the outside of window surfaces and powered wirelessly from an indoor power module as shown in Figure 2. Data is wirelessly transmitted to the RPi, which computes and sends analyzed results to the cloud via WiFi.

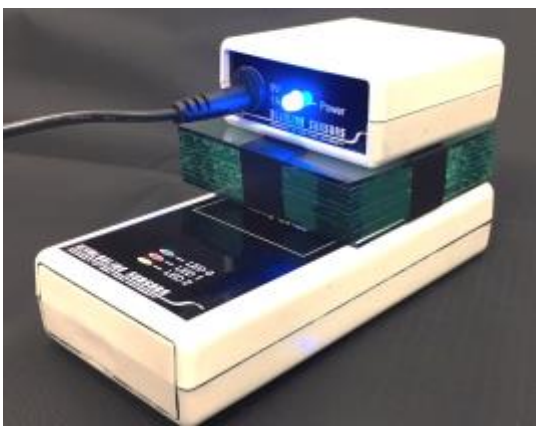

Figure 2. Citygram window-sensor solution showing outside-mic module (bottom), 25mm window glass (middle), and powered inside module (top),

2.2 Automatic Soundscape Analysis

Our noisescape-based machine learning (ML) efforts have focused on outdoor soundscape classification. The field of automatic outdoor soundscape recognition, unlike image and speech-based ML, is still in its nascent stages due to the following key challenges: (1) dearth in ground truth datasets, (2) emphasis on speech and image recognition, and (3) sonic complexity and diversity: soundscapes are unusually diverse with high inter-class correlation and intra-class variance, rendering the classification task fundamentally difficult. Despite these difficulties, the growing interest in soundscape recognition – something we have termed *Soundscape Information Retrieval* (SIR) [7] – can be witnessed in the creation of the DCASE computational audio scene analysis challenges [37].

Over the past three years, we have systematically focused on both traditional engineered feature-based machine learning and deep learning models using soundscape benchmarking datasets [37]-[43]. Our current outputs yield mid 90% performance for *segmented* soundscape events. This is in line with current state-of-the-art soundscape machine/deep learning models. Our research also shows that machine learning systems with engineered features perform comparably to deep learning systems. For example, when using a Random Forest classifier (20 MFCC, 5-fold validation) and a convolutional neural network (CNN) model (kernel size of 20x20, 5x10 2d convolution layers, stride of 2, 9 fully connected layers, and 1 softmax layer), random forest actually yielded slightly better performance with 98% correct classification for 10 classes. However, when either model was tested on continuous soundscapes – requiring both acoustic event detection and acoustic event classification – the results were significantly inferior (0.2543 F1-score). This tendency was also observed across systems submitted for the DCASE competition [37, 38], all performing poorly for natural, complete soundscapes.

To address issues of ground truth dataset availability, which is critical to all machine learning research including SIR, we have developed two types of crowd-sourcing software: Sound Event Annotation (*SEASound*) and *Heardcha* (Figure 3). *SEASound* has been designed for 30-second full soundscape annotation (event and sound class), while *Heardcha* has been designed as a rapid labeling "app" to mirror *captcha* for identifying single 2-4 sound examples. As there is much confusion amongst human annotators what *is* and what *is not* an acoustic event, we are using our software tools to crowd-source "consensus" which in turn properly trains deep learning classifiers.

Figure 3. HeardCha UX: Users listen to sounds and identify noises. "Captcha for the blind." Enables crowd-sourced verification of soundscape data and events.

We are also currently harnessing deep learning techniques for soundscape classification and building on the work of several convolutional and recurrent neural network libraries written in pure *JavaScript* [44]-[46].

2.3 Exploration Portal

The Citygram visualization and exploration module has taken a multifaceted approach in order to address the following: (1) spatiotemporal mapping paradigms, (2) interfaces for experts and non-experts, (3) crowd-sourcing tools for machine learning, (4) community reporting/surveying tool, and (5) data access/download tools for offline, personal applications. To this end, we have built various multimodal-mapping prototypes supplemental to popular maps like *Google Maps*. The aim is to facilitate interactive data analysis and to "empower data analysts to formulate and assess hypotheses in a rapid, iterative manner – thereby supporting exploration at the rate of the human thought" [47]. The current prototypes advance real-time soundscape maps by overlaying them on traditional map interfaces. We have built our work on published works [2, 4, 7, 10, 48]. Ultimately, our exploration portal will serve as a “one-stop-shop” for soundscape data visualization, exploration, access, and community reporting and knowledge contribution. Figure 4 shows heatmap animation of lower Manhattan visualized in “birth” playback mode. Three playback modes are currently implemented in the Citygram exploration portal: (1) *birth*: for simultaneously visualizing all nodes in the CG archives at each of their respective “birthdays,” (2) *past*: for visualization outputs of sensors from a specific timestamp in the past, and (3) *active*: for visualization of current active sensor outputs.

Figure 4. Citygram heatmap for visualizing *birth*, *past*, and *active* soundscape data.

Additionally, the CG exploration portal includes a data down-streaming feature enabled by using a node.js-based bridging utility to stream data from the CG server, to the browser, client side node.js (via *localhost*), and finally to the user’s software application (e.g. *Max* or *Supercollider*) transmitted through the open sound control (OSC) protocol. Any OSC compatible software can be used in receiving spatiotemporal CG data.

3. CONCLUSION

In this paper, we have summarized recent developments of the Citygram project – a *JavaScript*-based platform for capturing, streaming, analyzing, visualizing, and providing easy access to spatiotemporal soundscape data. Our research represents an integrative research agenda that brings together acoustics, engineering, computer science, data science, hardware/telecommunications research, GIS, human-computer interaction, citizen science, and data visualization built on a *JavaScript*/web-browser framework. Our project is strongly geared towards community engagement and has strategically built a research program that embraces citizen-science engagement that simultaneously enables collaborative opportunities between researchers, policy-makers, educators, students, and citizen scientists, and also facilitates sensor network growth to obtain spatiotemporally dense data from cities around the world. Citygram is *scalable*, *portable*, and *transferrable* as it is designed to (a) be a community-driven effort made practicable through *plug-and-sense* sensors and CPS technologies, (b) have built-in participation incentives including actionable and re-actionable pathways for improving residents’ living conditions, (c) address urban noise pollution phenomenon shared by cities around the world, and (d) be adoptable for urban residents. We plan to launch Citygram’s S o u n d m a p p i n g O u r W o r l d! effort in the fall of 2017 making available our technologies and exploration portals for mapping sound-spaces around the world.

4. REFERENCES

[1] T. Park, J. Turner, C. Jacoby, and A. Marse, “Locative Sonification: Playing the World Through Citygram,” 2013.

[2] T. H. Park, M. Musick, J. Turner, C. Mydlarz, J. H. Lee, J. You, and L. DuBois, “Citygram: One Year Later ...,” in International Computer Music Conference Proceedings, 2014.

[3] C. Shamoon and T. Park, “New York City’s New Noise Code and NYU’s Citygram-Sound Project,” … -NOISE NOISE-CON Congr. …, 2014.

[4] T. H. Park, J. Turner, M. Musick, J. H. Lee, C. Jacoby, C. Mydlarz, and J. Salamon, “Sensing Urban Soundscapes,” in Workshop on Mining Urban Data, 2014.

[5] C. Mydlarz, C. Shamoon, M. Baglione, and M. Pimpinella, “The design and calibration of low cost urban acoustic sensing devices,” 2015.

[6] C. Shamoon and T. H. Park, “Noise Control in megacities as evidenced in New York City (NYC),” in Noise Con, 2014.

[7] T. H. Park, J. Lee, J. You, M.-J. Yoo, and J. Turner, “Towards Soundscape Information Retrieval (SIR),” in Proceedings of the International Computer Music Conference 2014, 2014.

[8] M. Musick, J. Turner, and T. Park, “Interactive auditory display of urban spatio-acoustics,” 20th Int. Conf. …, 2014.

[9] C. Shamoon and T. Park, “Discussion of New York City’s new noise code,” in INTER-NOISE …, 2014.

[10] T. H. Park, J. Turner, C. Jacoby, A. Marse, M. Music, A. Kapur, and J. He, “Locative Sonification: Playing the World Through Citygram,” in International Computer Music Conference Proceedings (ICMC), 2013, pp. 11–17.

[11] “Arline Bronzaft Seeks a Less Noisy New York - NYTimes.com.” [Online]. Available: http://www nytimes.com/2013/10/07/nyregion/arline-bronzaft-seeks-a-less-noisy-new-york html?_r=2&adxnnl=1&ref=health&adxnnlx=1401455249-ggbv5pzX04mLZgCmp6WXpg. [Accessed: 30-May-2014].

[12] T. Lang, C. Fouriaud, and M.-C. Jacquinet-Salord, “Length of occupational noise exposure and blood pressure,” Int. Arch. Occup. Environ. Health, vol. 63, no. 6, pp. 369–372, 1992.

[13] Y. M. Zhao, S. Z. Zhang, S. Selvin, and R. C. Spear, “A dose response relation for noise induced hypertension.,” Br. J. Ind. Med., vol. 48, no. 3, pp. 179–184, 1991.

[14] W. D. Ward, “Noise and Human Efficiency,” Ear Hear., vol. 8, no. 4, pp. 254–255, 1987.

[15] F. J. H. Van Dijk, A. M. Souman, and F. F. De Vries, “Non-auditory effects of noise in industry,” Int. Arch. Occup. Environ. Health, vol. 59, no. 2, pp. 133–145, 1987.

[16] K. D. Kryter and others, “The effects of noise on man.,” Eff. noise man., 1970.

[17] W. Passchier-Vermeer and W. F. Passchier, "Noise exposure and public health.," Environ. Health Perspect., vol. 108, no. Suppl 1, p. 123, 2000.

[18] G. W. Evans, P. Lercher, M. Meis, H. Ising, and W. W. Kofler, "Community noise exposure and stress in children," J. Acoust. Soc. Am., vol. 109, no. 3, pp. 1023–1027, 2001.

[19] A. L. Bronzaft, "Noise pollution: A hazard to physical and mental well-being," Handb. Environ. Psychol., pp. 499–510, 2002.

[20] P. Woolner and E. Hall, "Noise in schools: a holistic approach to the issue," Int. J. Environ. Res. Public Health, vol. 7, no. 8, pp. 3255–3269, 2010.

[21] P. Knipschild, "V. Medical effects of aircraft noise: community cardiovascular survey," Int. Arch. Occup. Environ. Health, vol. 40, no. 3, pp. 185–190, 1977.

[22] B. R. Slater, "Effects of noise on pupil performance.," J. Educ. Psychol., vol. 59, no. 4, p. 239, 1968.

[23] L. Goines and L. Hagler, "Noise pollution: A modern plague," South. Med. J., vol. 100, no. 3, pp. 287–294, 2007.

[24] P. Leather, D. Beale, and L. Sullivan, "Noise, psychosocial stress and their interaction in the workplace," J. Environ. Psychol., 2003.

[25] J. Errett and E. Bowden, "Effects of noise on productivity: does performance decrease over time?," Archit. Eng. …, 2006.

[26] L. Cantley and D. Galusha, "Association between ambient noise exposure, hearing acuity, and risk of acute occupational injury," Scand. J. …, 2014.

[27] H. Amjad-Sardrudi, "Effect of noise exposure on occupational injuries: a cross-sectional study," J. Res. …, 2012.

[28] S. Cinderby, C. Snell, and J. Forrester, "Participatory GIS and its application in governance: the example of air quality and the implications for noise pollution," Local Environ., vol. 13, no. 4, pp. 309–320, 2008.

[29] N. Maisonneuve, M. Stevens, and L. Steels, "Measure and map noise pollution with your mobile phone," DIY HCI-A Showc. Methods, Communities Values Reuse Cust., 2009.

[30] "widenoise @ cs.everyaware.eu."

[31] "6b7a188967cffacab89c11b450076b9316b850f6 @ tendernoise.movity.com."

[32] I. Schweizer, R. Bärtl, A. Schulz, F. Probst, and M. Mühlhäuser, "NoiseMap - Real-time participatory noise maps," Second International Workshop on Sensing Applications on Mobile Phones 2011, p. 1-5.

[33] C. So-In, N. Weeramongkonlert, C. Phaudphut, B. Waikham, C. Khunboa, and C. Jaikaeo, "Android OS mobile monitoring systems using an efficient transmission technique over Tmote sky WSNs," in Proceedings of the 2012 8th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2012, 2012.

[34] D. Steele, D. Krijnders, and C. Guatavino, "The Sensor City Initiative: cognitive sensors for soundscape transformations," in Proceedings of GIS Ostrava 2013: Geoinformatics for City Transformations, 2013.

[35] C. Mydlarz, S. Nacach, T. Park, and A. Roginska, "The design of urban sound monitoring devices," Audio Eng. Soc. …, 2014.

[36] C. Mydlarz, S. Nacach, and A. Roginska, "The Implementation of MEMS Microphones for Urban Sound Sensing," … Soc. Conv. 137, 2014.

[37] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2013, pp. 1–4.

[38] "Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop," 2016.

[39] K. J. Piczak, "ESC: Dataset for environmental sound classification," in Proceedings of the 23rd ACM international conference on Multimedia, 2015, pp. 1015–1018.

[40] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-markovian ensemble voting," in RO-MAN, 2012 IEEE, 2012, pp. 509–514.

[41] M. Janvier, X. Alameda-Pineda, L. Girinz, and R. Horaud, "Sound-event recognition with a companion humanoid," in Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on, 2012, pp. 104–111.

[42] J. Salamon, C. Jacoby, and J. Bello, "A Dataset and Taxonomy for Urban Sound Research," … ACM Int. Conf. …, 2014.

[43] E. Hemery and J.-J. Aucouturier, "One hundred ways to process time, frequency, rate and scale in the central auditory system: a pattern-recognition meta-analysis," Front. Comput. Neurosci., vol. 9, 2015.

[44] ConvNetJS: https://github.com/karpathy/convnetjs

[45] Brain: https://github.com/harthur/brain

[46] Synaptic: https://github.com/cazala/synaptic

[47] J. Heer and S. Kandel, "Interactive analysis of big data," XRDS Crossroads, ACM Mag. Students, vol. 19, no. 1, pp. 50–54, 2012.

[48] T. Park, Z. Li, and W. Wu, "Easy Does It: The Electro-Acoustic Music Analysis Toolbox.," ISMIR, 2009.