The Case for Onloading Continuous High-Datarate Perception to the Phone

Seungyeop Han
University of Washington

Matthai Philipose
Microsoft Research

Abstract

Much has been said recently on off-loading computations from the phone. In particular, workloads such as speech and visual recognition that involve models based on "big data" are thought to be prime candidates for cloud processing. We posit that the next few years will see the arrival of mobile usages that require continuous processing of audio and video data from wearable devices. We argue that these usages are unlikely to flourish unless substantial computation is moved back on to the phone. We outline possible solutions to the problems inherent in such a move. We advocate a close partnership between perception and systems researchers to realize these usages.

1 Introduction

By the end of this decade, computer systems will provide the equivalent of turn-by-turn navigation for peoples' daily life. Given high-level goals and preferences, such as staying connected with family or eating local, such systems will continuously sense their users' state and environment and nudge them toward these goals. Several current trends point in this direction: continuously sensing wearables that help toward physical fitness goals are gaining popularity (top of Figure 1), wearing video accessories is acquiring cachet (bottom), viewing the phone as mobile assistant a la Siri is commonplace and emerging services such as Google Now have begun proactively notifying users of opportunities. We expect these trends to be distilled into a system that analyzes an audio/video stream from a wearable at modest latency and delivers relevant but sporadic feedback.

We believe that to be broadly useful and deeply engaging, understanding visual and auditory context is essential. A natural question is whether vision and speech algorithms are mature enough be usable in the near future to support research and early applications. We identify four specific speech/vision capabilities (continuous large-vocabulary conversational speech recognition, conversational partner identification, location/pose estimation and handled-object recognition) that we believe are central and argue that the state of the art is promising.

Figure 1: Continuous sensing accessories: Low-datarate (top) accessories based on inertial sensors support self-measurement applications. High-datarate (bottom) ones based on video are restricted to capture/display today.

Computer vision and speech are resource-intensive workloads, requiring considerable processing to yield timely results, memory for "big-data"-derived models and energy for processing large volumes of data. Recent work [7, 22] has suggested the possibility of finessing such constraints by offloading computations to the cloud wirelessly. For the "interactive perception" applications they target, where latency is critical and the system is used where WiFi and local servers are available, offloading has proved promising.

Offloading to the cloud is not as desirable for continuous high-datarate (CHDR) perception applications. First, availability is key in these applications, and in the not uncommon case [1, 29] that both WiFi and cellular connections are unavailable or spotty, offloading is infeasible. Second, processing continuously sensed data offers cloud providers far lower benefit per byte than, e.g., textual search queries, at much higher cost. Amortizing the sunk cost of a state-of-the-art and user-powered and

-purchased client is attractive. Third, video and audio data from wearables is private enough that processing it locally may be a major draw for users. Finally, modest latencies are often acceptable in these applications. On the other hand, executing recognition algorithms on the phone poses three fundamental questions. Does the phone have enough compute resources to return results at acceptable latency? Can power consumption be kept to acceptable levels? Can memory usage be limited, especially if the phone competes with "cloud-sized" models?

We argue that expected increases in mobile processing capabilities will allow a 2015 phone to process frames at acceptable latency. Further, we show early data to support the intuition that complementing the camera with lower-powered sensors often allows most frames to be ignored by the vision system. Combined with the over 10x increase in power efficiency expected in these processors, we believe that adhering to a conservative 10Wh battery budget is feasible. Finally, an examination of the scaling of recognition models leads us to believe that it is possible to cache them profitably on the mobile device, thus "onloading" the cloud to the phone.

Collaboration between systems and perception researchers have yielded remarkable recent progress in "cloud-scale" perception [3, 12, 19]. We advocate a similar partnership to bring eyes and ears to the phone.

2 Mobile CHDR-Perception Applications

Figure 2 sorts continuous perception sensors, perception primitives and applications by datarate. Continuous perception at low datarate (LDR) includes wrist-worn and phone-embedded RFID readers, light sensors, GPS- and WiFi-based location sensors and inertial sensors. Applications of these sensors have spawned entire industries, including those of location-based services, personal health monitoring via kinematics, natural UIs for mobile devices and object-self-identification and validation. However, the transition to CHDR sensors, illustrated by the dashed line after the 10^4 B/s mark in the figure, introduces two fundamental capabilities that promise even higher impact. The audio and video sensors that occupy this space not only encode semantically far richer information relevant to the user, they are also able to monitor a substantial area *around* their installation point unlike typical lower-datarate sensors that tend to monitor a single point.

Much has been written on the potential of vision- and speech-based applications. When these modalities are combined with the privileged location of a wearable, the opportunities are even greater. We sketch possible applications, mainly to emphasize the shift from the low-datarate regime. First, even simple analysis of everything one says is deeply revealing of mental state: affect, preferences, intent and social dynamics are revealed directly. Second, knowing what you are doing, and when, opens the door to a holy grail of medical intervention, that of effecting behavior change. Phones that know when you overstress, overeat, drink, smoke, procrastinate or irritate may intervene quickly to counter the behavior if you so desire. Third, phones can aid in complex socio-physical tasks: they can step you through changing the wipers on your car or traverse the bureaucracy at an unfamiliar airport. Fourth, phones can augment your cognitive capacity: they can recognize your conversation partner whose name you've forgotten or remind you to pick up the milk your wife requested. The powers inherent in these scenarios are a cause for worry if they are abused, but there is little doubt that they constitute a profound shift from, e.g., the "walking", "running" and "climbing" detection of today's popular inertial sensors.

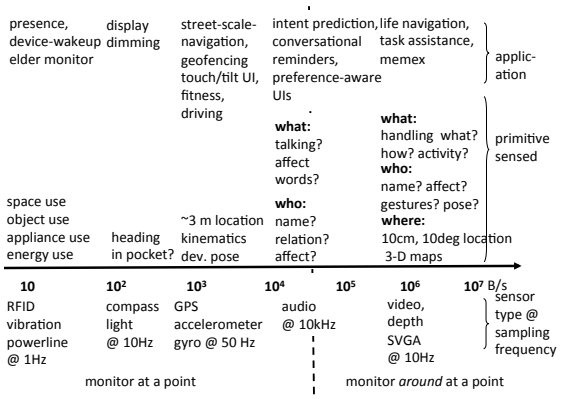

Figure 2: Perception capabilities vs. datarate

Even this brief list illustrates some important properties of CHDR-perception based applications. First, they often do not have very strong latency constraints: a delay of even a few seconds between observation and inference is acceptable in conversational reminders, intent-based action, lifelog querying or diet or exercise suggestions. Second, interesting events may only happen sporadically. For instance, most footage through the day will be irrelevant to the weight management application. Third, they are well served by multiple sensing modalities, including low-datarate ones: location, speech and object detection may all be relevant to a life log. Fourth, although low-datarate sensors cannot match the richness of recognition offered by CHDR sensors, they may usefully *gate* them: a humble light sensor can tell that the camera should not bother to look because it is too dark. Finally, and perhaps most importantly, CHDR perception clearly enables deeply valuable new capabilities, worthy of devoting a large fraction of a phone's resources.

	Speech	Face	Object	Location + Pose + Map
Approach	Mel-Frequency Cepstra / Deep Neural Network + n-grams / HMM [13]	Local Binary Patterns + Linear Discriminant Analysis / 1-vs-all SVM [26]	Deep Neural Network [18]	Compact Signatures / Vocabulary Tree / non-linear least squares [16]
Experimental setting	Large vocabulary conversational data without (Youtube tracks) [4] and with (telephone discussions) [23] strong language model.	>6k images of >600 celebrities from the Internet. Natural pose, lighting, occlusion, makeover effects [14].	Recognize 1000 different types of objects within 150,000 images from the Internet [8]	Real-time estimation over 800m in a 45x45m office floor; 10km rough terrain
Accuracy (%)	52, 84	58 (50 person) - 83 (5)	63 (top 1), 83-85 (top 5)	< 22cm RMS indoors; 10m rough terrain
Model size (GB)	0.5 (3-gram) - 600 (5-gram)[4]	1MB/person (with 100-dim LDA) [5]	3-6	1.8kB/keyframe, 18MB for office floor
Compute overhead	16kHz @ 160% of Intel Xeon E5640 core (CPU) [25]	30fps @ 8-core Intel SandyBridge CPU** [24]	30fps @ NVIDIA GTX 580 GPU*	30fps @ 2-core 2.4 GHz '09 Intel CPU

*GPU used for training. Fraction of GPU, frame rate for testing not specified **Fraction of server unspecified

Table 1: The state of the art in state estimation

3 CHDR Perception Algorithms

As per the previous section, we consider four basic capabilities, continuous speech recognition, conversation partner identification, location and pose estimation and object recognition as the core capabilities of an initial CHDR system. These are old, hard problems in artificial intelligence. One barrier to working on systems issues in CHDR perception is the concern that these basic AI problems still only have brittle, highly specific, non-performance-optimized solutions that are generally not suitable for practical use. We argue that given recent advances, these capabilities have matured to the point of providing a practical basis for system design.

Table 1 summarizes the state of the art in solving the core perception problems. Four trends are worth noting:

Realistic test sets Most communities have now adopted large, realistic "challenge datasets" as common benchmarks. For instance, object recognition targets a thousand different objects from over a million images extracted from the Internet. Success on these datasets transfers to real-world applications.

Good performance Recognition is not perfect. However, phone conversations can be transcribed at 85% accuracy, and Internet scale object recognition works at 63-83% accuracy for top-1/top-5 recognition. Localization is accurate to 10cms over office buildings. These are usable rates for many apps.

Stability in algorithms Most leading perception systems today choose from a small set of statistical classifiers. Each field still has its favorite small set of features (e.g., "SIFT" in vision, "MFCC" in speech) that serve as intermediate representations before classification. However, the number of options has fallen sufficiently that efficient embedded or silicon implementation of algorithms is under way. Perhaps more interestingly, a new class of classifier (so called "Deep Networks") is rapidly superseding all other classifiers and feature extractors across many vision and speech problems. Using these algorithms as a basis for understanding systems tradeoffs is therefore increasingly practical.

Efficient implementations Increasingly, careful attention is paid to performance. For the most part, this is because perception modules are used to serve Internet queries, e.g., for face recognition and speech transcription. Although these server-based algorithms are not directly transferable to on-phone implementations, the extensive performance characterizations are available. In some cases, such as vision-based localization, the leading algorithms are already optimized for embedded ("mobile") performance (on robots!).

4 Implications for System Design

We argue below that several factors nudge CHDR perception systems toward performing the bulk of their processing on the phone. We then use numbers from Table 1 and current hardware trends to argue that, given careful system design,"cloud-quality" perception on the phone is potentially feasible. Finally, we identify the key resources that applications on the phone may share during CHDR perception, and sketch operating system functionality that could support this sharing conveniently, efficiently and safely.

4.1 Why Not to Offload

We believe that network availability, core network bandwidth, cellular transmission power, privacy, application

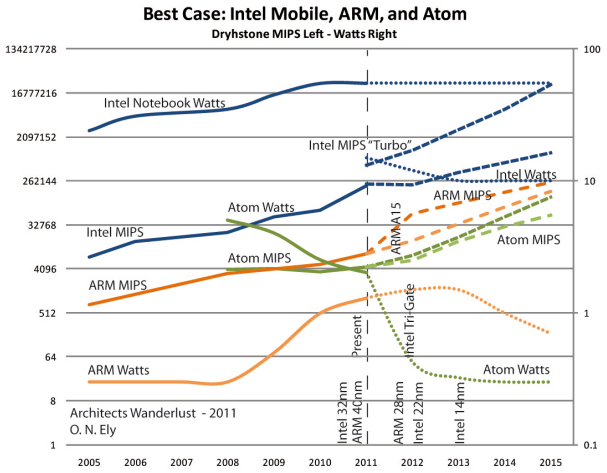

Figure 3: Projection of CPU capabilities and power consumption, copied from [9]. According to the green line (Atom), performance of mobile CPU will be comparable to that of current notebook CPU, and battery consumption will be reduced by an order of magnitude.

requirements and the economics of cloud use all favor most CHDR processing happening on the phone. CHDR usages are fundamentally *high availability*: your eyes and ears cannot have the same spotty availability as WiFi or even cellular networks [1, 29]. Even when the (cell) network is available, streaming the equivalent of HD video from *every phone continuously* would place a prohibitive load on cellular base stations and the core network, even at a 1% duty cycle via clever multimodal gating. Given that the cell radio roughly doubles phone power usage, local processing is attractive (especially given falling joules/cycle relative to slow-changing joules/byte transmitted). Given the sensitivity of the data involved, a guarantee that raw data does not leave the phone may be a powerful draw for many users.

Finally, as CHDR perception becomes more heavily used and spreads to third parties (unlike e.g., speech processing in Siri and Google Now today), the cost of cloud processing will become material. In particular, given that much of what a person does every day is not monetizable, and that significant cloud resources are required for processing, more efficient use of the substantial on-phone resources paid for by end-users will become attractive.

4.2 Why Onloading is Feasible

Of course, these attractions of on-phone processing are moot if, as intuition dictates, basic resource constraints make it impractical. In particular, CHDR workloads are known to require large models, sophisticated algorithms and high volumes of streaming input. Do phones have the processing power, memory and battery capacity to process handle them?

The message from row 4 of Table 1 is that a 2- to 8-core server-class CPU of today provides adequate compute power for state-of-the-art CDHR algorithms *at full frame rate*. Figure 3 depicts the upper end of projected power/performance of mobile CPUs by 2015, compared to notebook CPUs. Most notably, silicon process shrinks to 14nm should make the performance of 1W mobile CPUs (Atom and ARM) comparable to that of year-2011 (4-core, given the MIPS rating) *notebook* CPUs, where today there is a 10x gap. Assuming a generous 10x performance difference between server and notebook CPUs, and that projections are optimistic by 10x, we need to bridge a gap from 1000x today to 100x in 2015.

There are two possible lines of attack. First, DSPs, GPUs and ASICs can yield 10x to 100x improvements in speed with similar improvements in power/flop [28, 15]. As row 1 of Table 1 hints, CHDR algorithms are converging (e.g., around Deep Neural Networks) and stabilizing sufficiently that, as with media decoders, system designers should consider implementing these algorithms on more specialized processing fabrics. Second, as Table 2 shows, vision and sound need not be processed at full frame rate. Consider conversation-partner detection. Essentially, since conversations are rare, we can use inexpensive sensors (such as voice detectors, accelerometers and light sensors) to predict if the camera should bother reading a frame. Our experiments with 1 week of data from 3 researchers indicate that such gating can reduce frames analyzed by 98.5%. Much work exists on automatically gating low-datarate sensors in this way [2]. System designers should consider extending this work to high-datarate sensors and making gating a first-class system service, yielding a further 10-50x effective speedup.

Running within an acceptable power envelope will likely require a variety of techniques. Today's 40+W notebook will run for less than 15 minutes on a 10Wh phone battery. To extend this to 10h while using 20% of the phone battery for CHDR processing will will need at least 200x improvements in efficiency. In fact, Figure 3 indicates that 2015 phones will be 50x more efficient than today's laptop. Further the above gains in performance from DSP/ASIC implementations and sensor gating should apply to power efficiency as well. Finally, image sensors themselves can consumer significant amounts of power. Techniques to match their power consumption tightly to information extracted from them may be valuable [20].

Toward memory size, as Table 1 shows, models may require substantial space, sometimes proportional to the number of classes being distinguished. Intriguingly, in some cases such as speech recognition, small but good models could already fit within the 8GB DRAM that seems reasonable for 2015. Localization on the other hand requires roughly 20MB *per building floor* and face

Drop frames if	Fraction left to process
None	100%
No voice (no one interacting)	6%
High acceleration (image is blurred)	2.5%
Low light (image is too dark)	1.5%

Table 2: Gating HDR perception

recognition requires 1MB *per person*. It is clearly infeasible to load into phone DRAM a comprehensive model of every building, person or object in the world. The saving grace is that human routines have high locality. It is quite conceivable that, just as with offline maps, relevant models for a given person can be cached on the phone, based on simple context such as location. System designers must carefully consider the semantics of missing in the cache, updating models and efficient representations, but we believe that "offline models" are feasible. Note that the second level of the cache need not be the cloud: given recent trends in NVRAM [17], it is quite feasible for cache misses to be relatively cheap since most models can be stored on the phone.

4.3 Sharing and OS Support

As we discussed in the previous section, moving even individual CHDR perception applications to the phone will require careful engineering to allow effective use of heterogeneous hardware, gating sensors, cached models and power-proportional image sensors. In a typical phone of the future, however, we expect dozens of apps to simultaneously using perception capabilities. Given tight resource constraints, we expect these apps to share not only the related hardware but also intermediate results in computations such as feature extractors, classifiers and cached models. Enabling such sharing while maintaining programmability, efficiency and safety will require OS support.

In line with previous work [6, 21, 22], we believe that CHDR perception applications are well abstracted as dataflow graphs. Instantiating, sharing and scheduling these graphs on hardware are core tasks for the operating system. When these dataflow graphs process personal video and audio data however, we believe that several additional issues assume importance.

Information gating: For power and performance reasons we do not expect pipelines to be able process more than roughly 1% of sensor data. Fortunately, new data is often not interesting (e.g., there is no face in the frame), incrementally useful (e.g., it is the same face), or too noisy (e.g., too much motion blur). We posit a *gating framework* that works across the dataflow graph and is therefore cross-application that predicts whether graph nodes are worth executing (and avoids execution if appropriate) based on inexpensive node outputs. Applications could presumably extend and query the baseline OS-based predictor and controller.

Privacy: Personal video and audio pose obvious privacy concerns. Strong mechanisms to mitigate these concerns that lead to simple guarantees for users are essential. For instance, one guarantee could be that no information that leaves the phone can recreate raw images or audio in a manner that faces or words are discernible [10, 27]. Information-flow based techniques could ensure that all dataflow graph sinks must pass through relevant obfuscating computations. On the other hand, certain kinds of information may only be exported in ways that maintain differential privacy. Exposing these options to programmers as extensions of the basic declarative dataflow framework should be feasible and valuable.

Model management: Several applications may use locally cached fragments of large global models (e.g. for spoken language, multi-view stereo based localization, etc). Managing cached models is challenging. Providing consistent semantics for the cache with good performance across multiple client applications, sending new information back to the cloud-based cache efficiently and privately (so models may be improved) and exploiting emerging technologies such as NVRAMs for local L2 caching all require careful consideration.

Statistical dispatch: Subscription to events is a standard way in which producers and consumers interact in dataflow-style settings. The consumer typically provides the producer a pattern to dispatch on. Conventionally (e.g., in a geofencing systems or accelerometer-based activity recognition systems), the pattern provides either class labels of interest or simple predicates on these (e.g., "location = Starbucks", "activity = running"). With rich datatypes, however, we believe that consumers will increasingly want to provide statistical matchers (e.g., an application defining a spoken natural language frontend may provide examples of spoken commands with which its handler should be invoked, with the expectation that "similar" phrasings will also invoke it [11]). Composing these statistical dispatchers efficiently at installation and providing fast, secure dispatch are valuable OS services.

5 Summary

We point out the impending arrival of a new class of applications based on continuous high-datarate perception using wearable devices. These applications will dramatically push the boundaries of utility of mobile devices. Excitingly, the underlying perception algorithms have reached enough maturity that application and systems work can begin. Good performance will require significant reconsideration of mobile/cloud tradeoffs.

References

[1] BALASUBRAMANIAN, A., MAHAJAN, R., AND VENKATARAMANI, A. Augmenting mobile 3g using wifi. In *Mobisys* (2010).

[2] BENBASAT, A., AND PARADISO, J. Groggy wakeup – automated generation of power-efficient detection hierarchies for wearable sensors. In *BSN* (2007), pp. 59–64.

[3] BRANTS, T., POPAT, A. C., XU, P., OCH, F. J., AND DEAN, J. Large language models in machine translation. In *EMNLP-CoNLL* (2007), pp. 858–867.

[4] CHELBA, C., BIKEL, D., SHUGRINA, M., NGUYEN, P., AND KUMAR, S. Large scale language modeling in automatic speech recognition. Tech. rep., Google, 2012.

[5] CHEN, D., CAO, X., WANG, L., WEN, F., AND SUN, J. Bayesian face revisited: A joint formulation. In *ECCV (3)* (2012), pp. 566–579.

[6] CHU, D., KANSAL, A., LIU, J., AND ZHAO, F. Mobile apps: Its time to move up to condos. In *HotOS* (2011).

[7] CUERVO, E., BALASUBRAMANIAN, A., KI CHO, D., WOLMAN, A., SAROIU, S., CHANDRA, R., AND BAHL, P. Maui: Making smartphones last longer with code offload. In *MobiSys* (2010).

[8] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In *CVPR* (2009).

[9] ELY, O. N. Intel process advantage? 2012 to 2015 processor projections. `http://bit.ly/XnOhIT`, 2011.

[10] ENEV, M., JUNG, J., BO, L., REN, X., AND KOHNO, T. Sensorsift: balancing sensor data privacy and utility in automated face understanding. In *ACSAC* (2012).

[11] HAN, S., PHILIPOSE, M., AND JU, Y.-C. Nlify: Third-party programming support for spoken natural language interfaces. Tech. rep., Microsoft Research, 2012.

[12] HARB, B., CHELBA, C., DEAN, J., AND GHEMAWAT, S. Back-off language model compression. In *INTERSPEECH* (2009), pp. 352–355.

[13] HINTON, G., DENG, L., YU, D., DAHL, G. E., RAHMAN MOHAMED, A., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P., SAINATH, T. N., AND KINGSBURY, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE 29*, 6 (nov. 2012), 82 –97.

[14] HUANG, G. B., MATTAR, M., BERG, T., AND LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[15] KLUG, B., AND SHIMPI, A. L. Qualcomm snapdragon s4 (krait) performance preview - 1.5 ghz msm8960 mdp and adreno 225 benchmarks. `http://bit.ly/11TZwga`, 2012.

[16] KONOLIGE, K., BOWMAN, J., CHEN, J., MIHELICH, P., CALONDER, M., LEPETIT, V., AND FUA, P. View-based maps. In *Proceedings of Robotics: Science and Systems* (Seattle, USA, June 2009).

[17] KOUKOUMIDIS, E., LYMBEROPOULOS, D., STRAUSS, K., LIU, J., AND BURGER, D. Pocket cloudlets. In *ASPLOS* (2011).

[18] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS* (2012).

[19] LE, Q. V., RANZATO, M., MONGA, R., DEVIN, M., CHEN, K., CORRADO, G., DEAN, J., AND NG, A. Building high-level features using large scale unsupervised learning. In *ICML* (2012).

[20] LIKAMWA, R., PRIYANTHA, B., PHILIPOSE, M., ZHONG, L., AND BAHL, P. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Mobisys* (2013).

[21] LU, H., YANG, J., LIU, Z., LANE, N. D., CHOUDHURY, T., AND CAMPBELL, A. T. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys* (2010).

[22] RA, M.-R., SHETH, A., MUMMERT, L., PILLAI, P., WETHERALL, D., AND GOVINDAN, R. Odessa: Enabling interactive perception applications on mobile devices. In *Mobisys* (2011).

[23] SEIDE, F., LI, G., AND YU, D. Conversational speech transcription using context-dependent deep neural networks. In *INTERSPEECH* (2011), pp. 437–440.

[24] TAIGMAN, Y., AND WOLF, L. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition. *CoRR abs/1108.1122* (2011).

[25] VANHOUCKE, V., SENIOR, A., AND MAO, M. Z. Improving the speed of neural networks on CPUs. In *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning* (2011).

[26] WOLF, L., HASSNER, T., AND TAIGMAN, Y. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 10 (2011), 1978–1990.

[27] WYATT, D., CHOUDHURY, T., AND BILMES, J. Conversation detection and speaker segmentation in privacy-sensitive situated speech data. In *INTERSPEECH* (2007).

[28] ZHANG, N., AND BRODERSEN, B. The cost of flexibility in systems on a chip design for signal processing applications. *University of California, Berkeley, Tech. Rep* (2002).

[29] ZHOU, P., ZHENG, Y., LI, Z., LI, M., AND SHEN, G. Iodetector: A generic service for indoor outdoor detection. In *MobiSys* (2012).