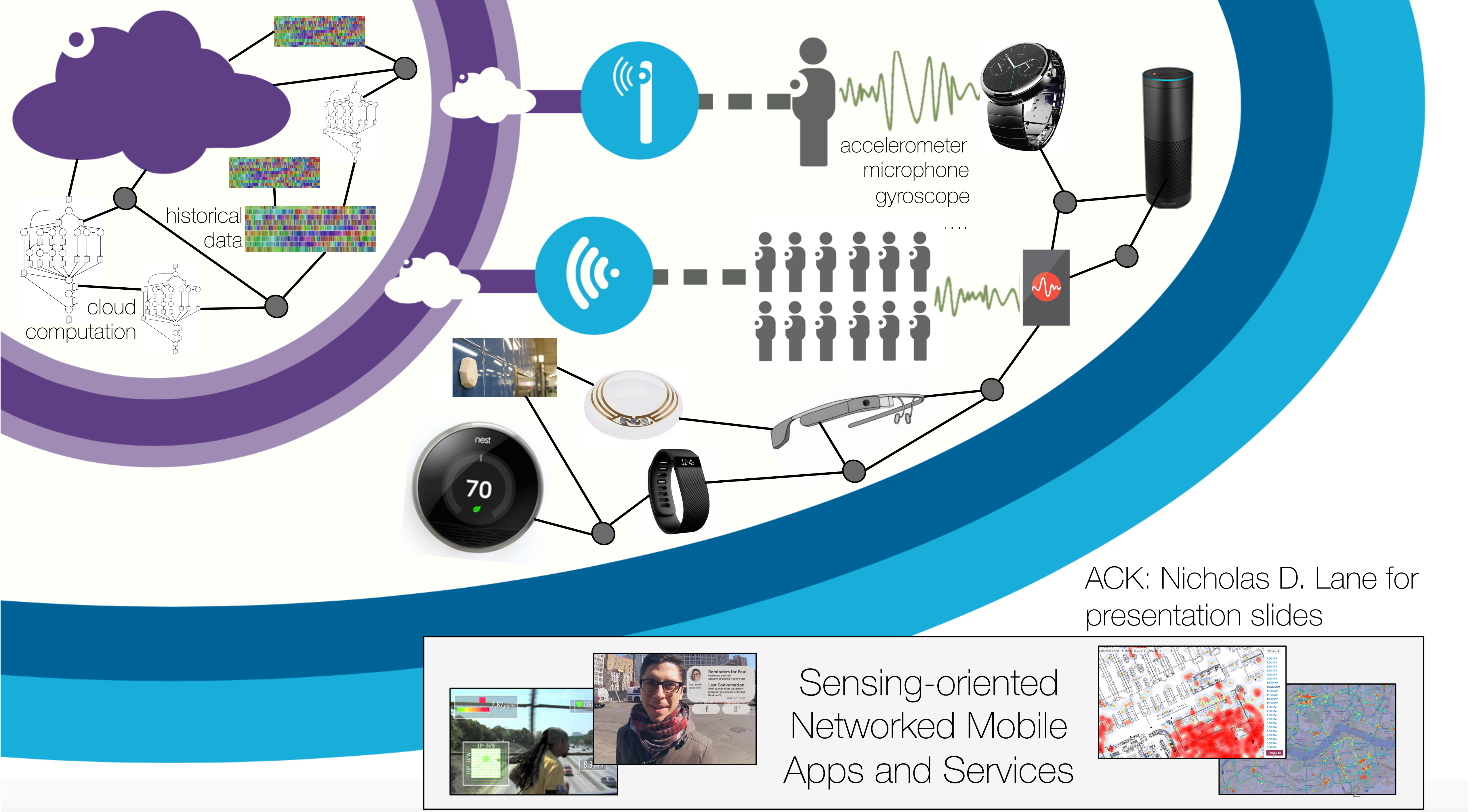
NOKIA Bell Labs

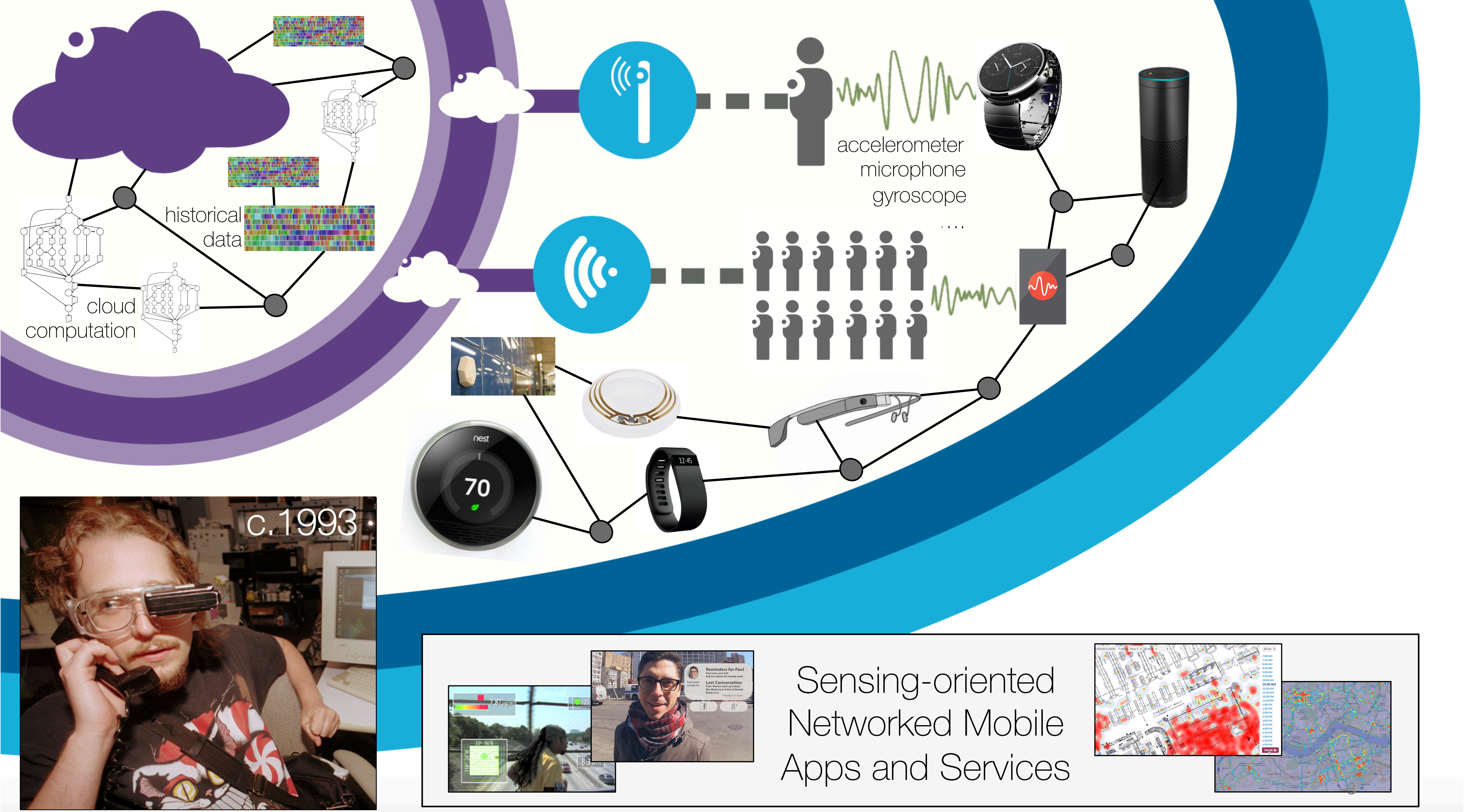
18th March, 2016
CoSDEO 2016, Sydney

Enabling Efficient Deep Learning Inference on Mobile Devices

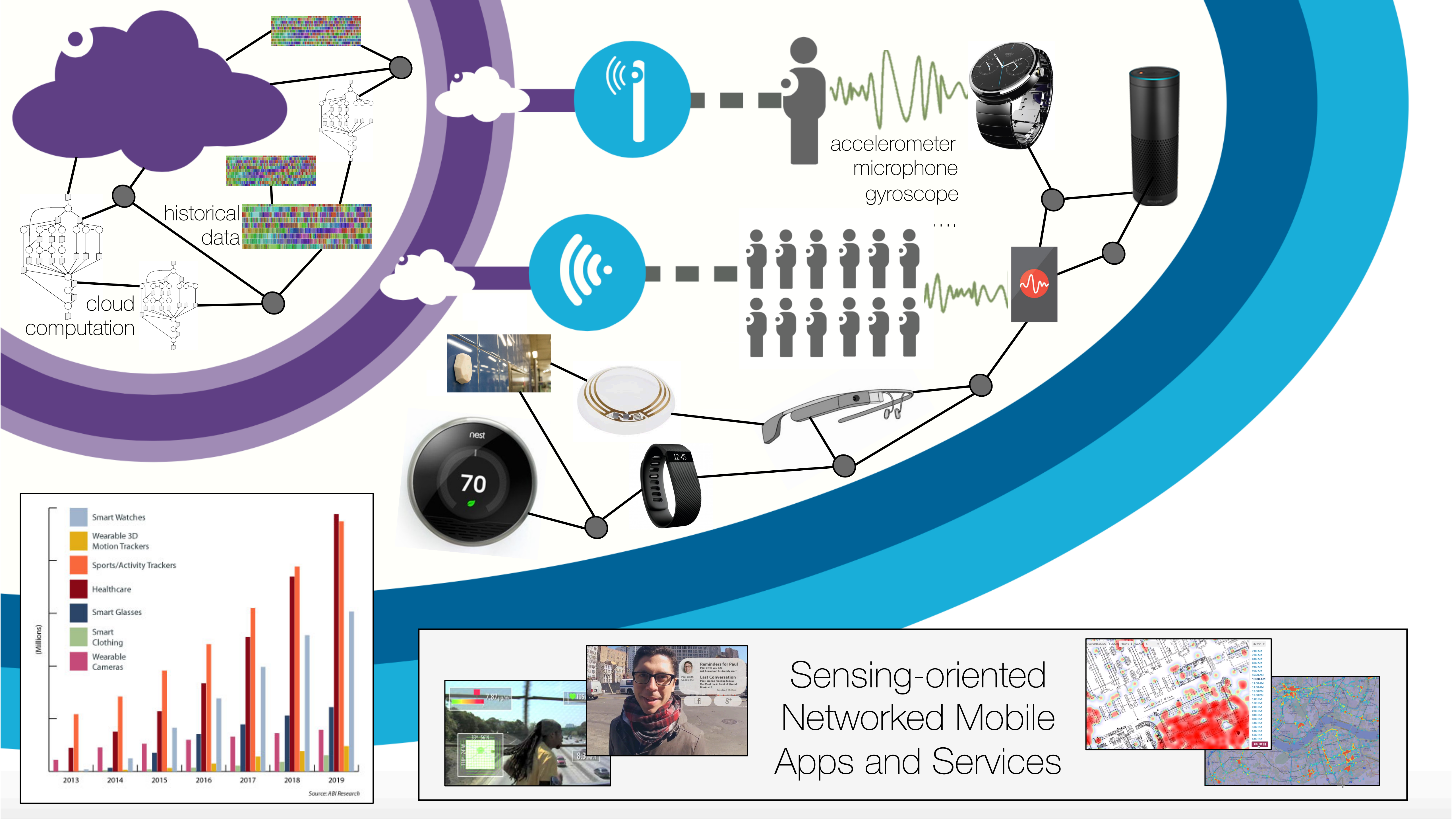
Sourav Bhattacharya

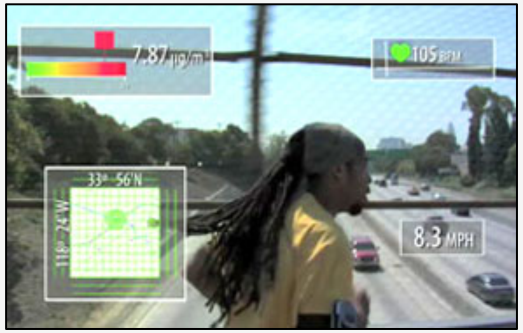


historical
data
cloud
computation
accelerometer
microphone
gyroscope
....
nest
70
12:45
ACK: Nicholas D. Lane for presentation slides
Sensing-oriented
Networked Mobile
Apps and Services

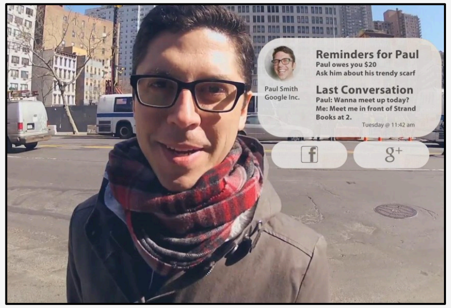


historical
data
cloud
computation
accelerometer
microphone
gyroscope
....
nest
70
12:45
c.1993
Sensing-oriented
Networked Mobile
Apps and Services

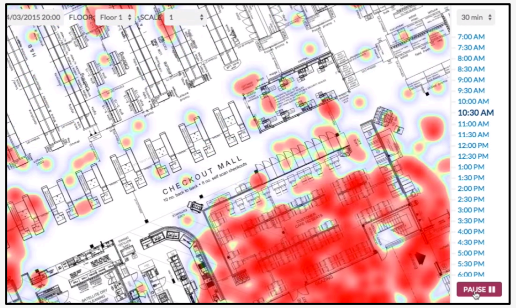

historical
data
cloud
computation
accelerometer
microphone
gyroscope
nest
70
12:45
Smart Watches
Wearable 3D
Motion Trackers
Sports/Activity Trackers
Healthcare
Smart Glasses
Smart
Clothing
Wearable
Cameras
(Millions)
2013
2014
2015
2016
2017
2018
2019
Source: ABI Research
Sensing-oriented
Networked Mobile
Apps and Services



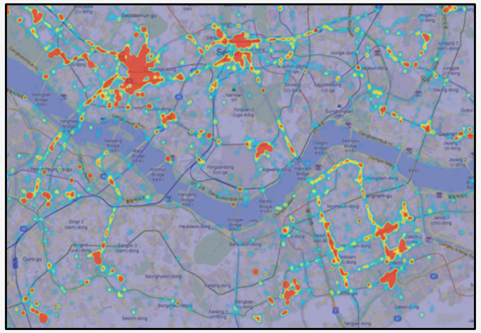
Mobile Health



Digital Assistants



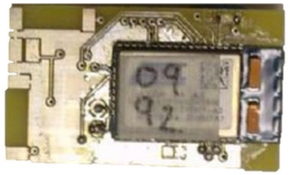
Quantified Enterprise

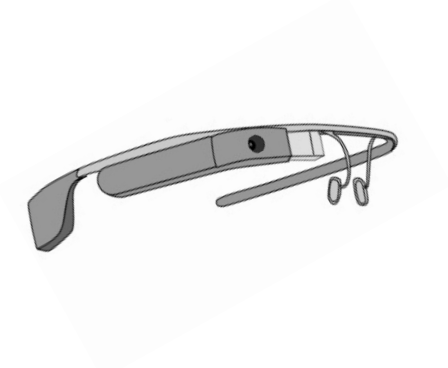


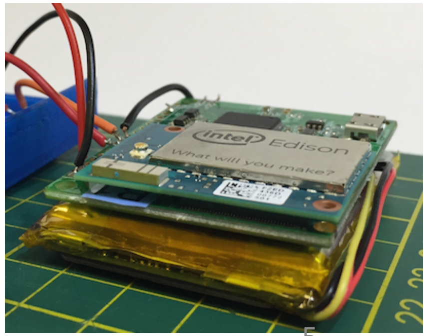
Urban Sensing

Consumer
Personal Sensing

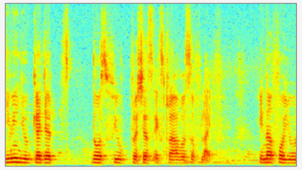
Sensor-driven Cities,
Enterprises & Organizations

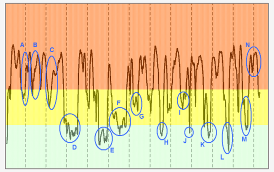






Audio Data





Inertial Data

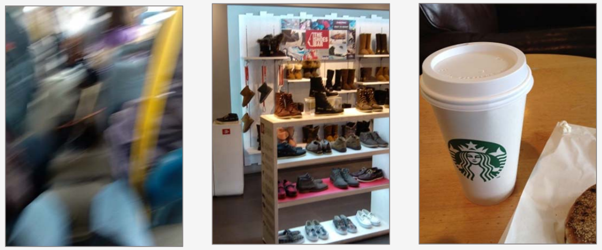
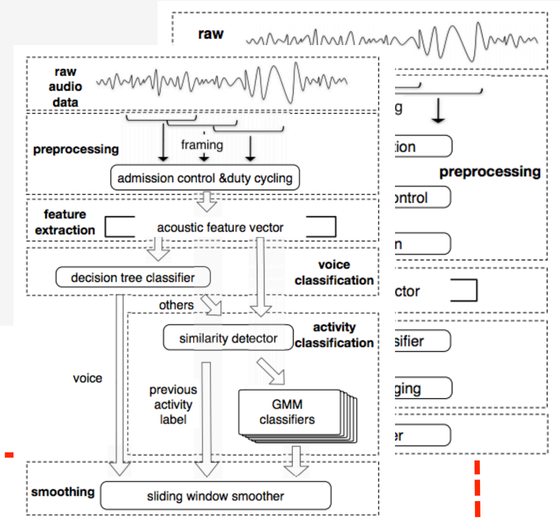


Image Data

Sensor Inference Pipelines



{stressed, not stressed}

{walking, running, sitting}

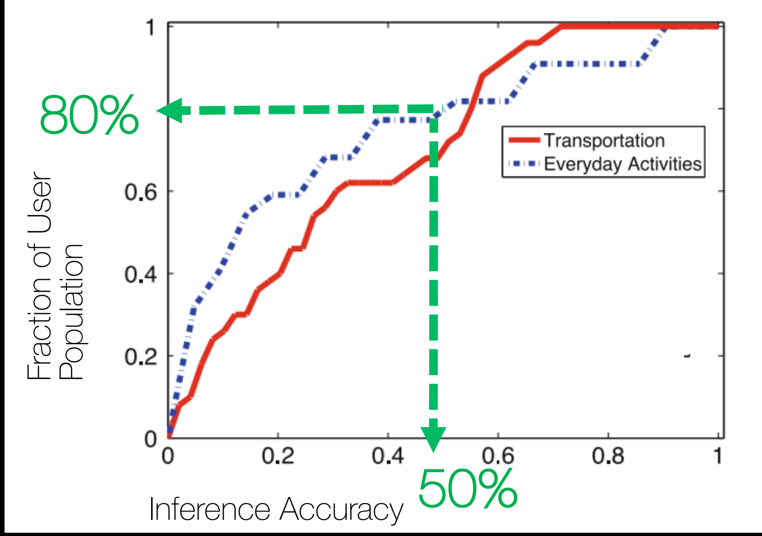
{music, conversation, male voice}

{shoes, subway, coffee cup}

Sensors	Computation	Resources

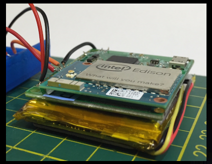
Sensor Inference is a core unifying process across all IoT / Wearable Systems

Sensor Inference Gap



Consumer
Personal
Sensing

Sensor Data-driven
Cities, Enterprises &
Organizations



High-value Behavior and Context Inferences Remain Unreliable in Real World Settings for Wearable/IoT Devices

Breakthroughs in Practical Modeling Problems Powered by Deep Learning

Speech
Recognition

Object
Recognition

Language
Translation

Natural Language
Processing

Face recognition

.....

Breakthroughs in Practical Modeling Problems Powered by Deep Learning

Speech Recognition

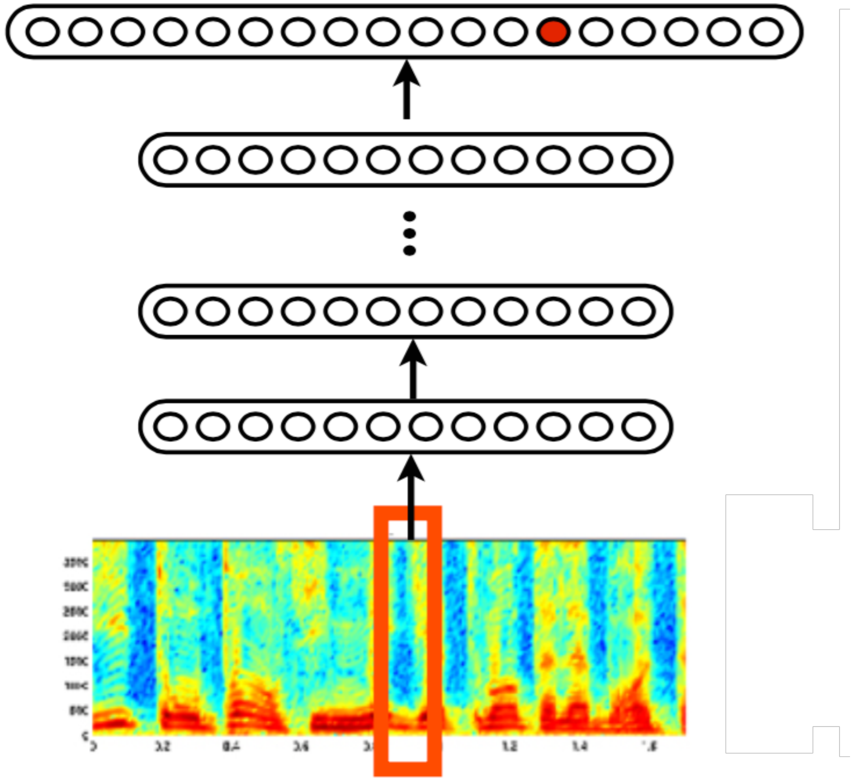
Object Recognition

Language Translation

Natural Language Processing

Face recognition

.....



Google Example

Launched in 2012 with the Jellybean Android release

Trained model <5 days on cluster of 800 machines

30% reduction in Word Error Rate for English

“Biggest single improvement in 20 years of speech research”

ACK: Jeff Dean (Google)

Breakthroughs in Practical Modeling Problems Powered by Deep Learning

Speech Recognition

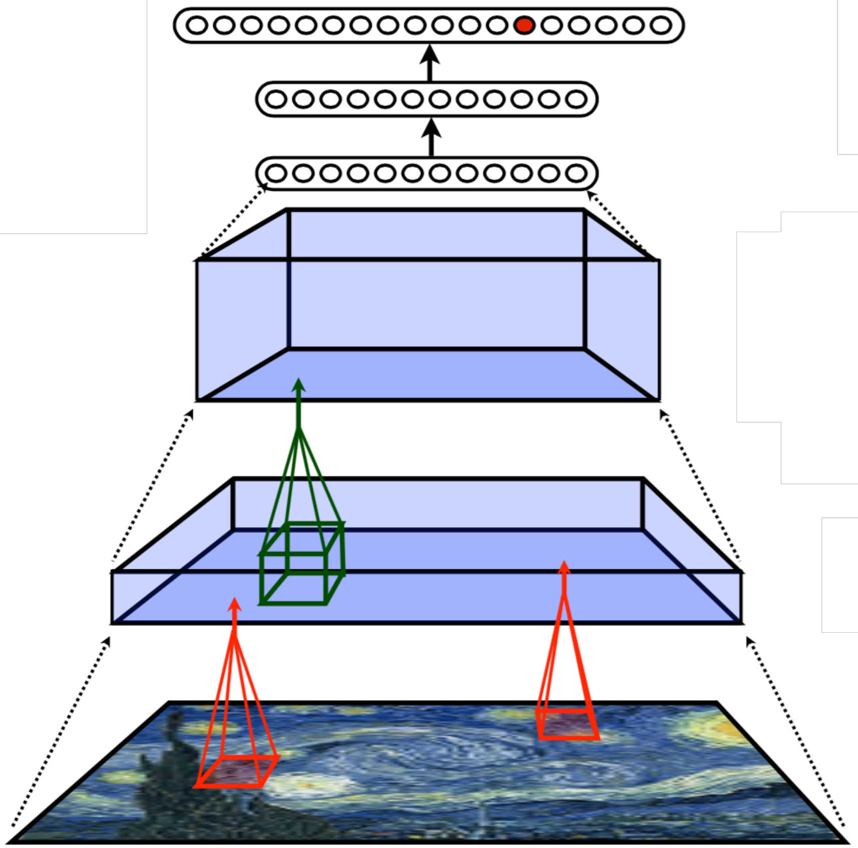
Object Recognition

Language Translation

Natural Language Processing

Face recognition

.....



Google Example

7-layer Convolutional Neural Network (CNN) won 2012 ImageNet Challenge **16.4%** top-5 result

24-layer CNN won 2014 ImageNet Challenge **6.67%** top-5 result

ACK: Jeff Dean (Google)

Breakthroughs in Practical Modeling Problems Powered by Deep Learning

Speech Recognition

Object Recognition

Language Translation

Natural Language Processing

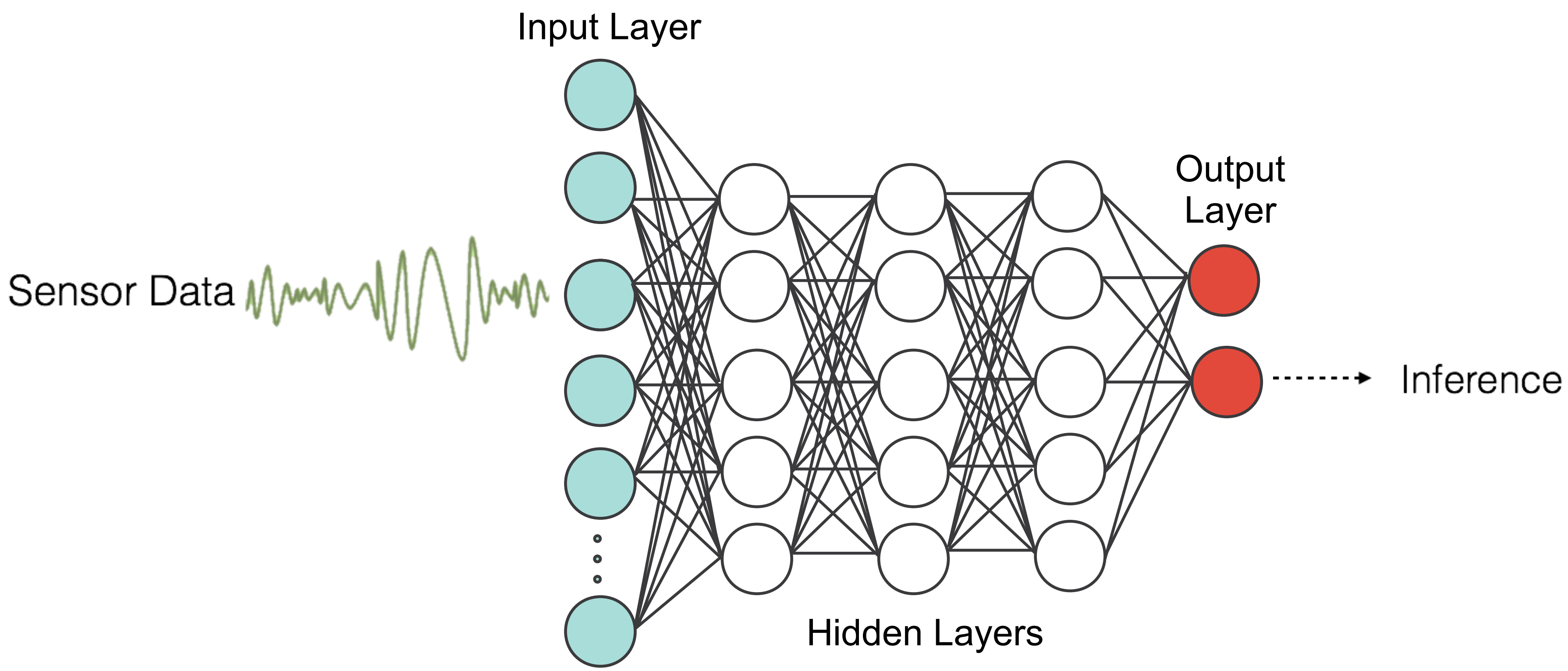
Face recognition

User Behavior

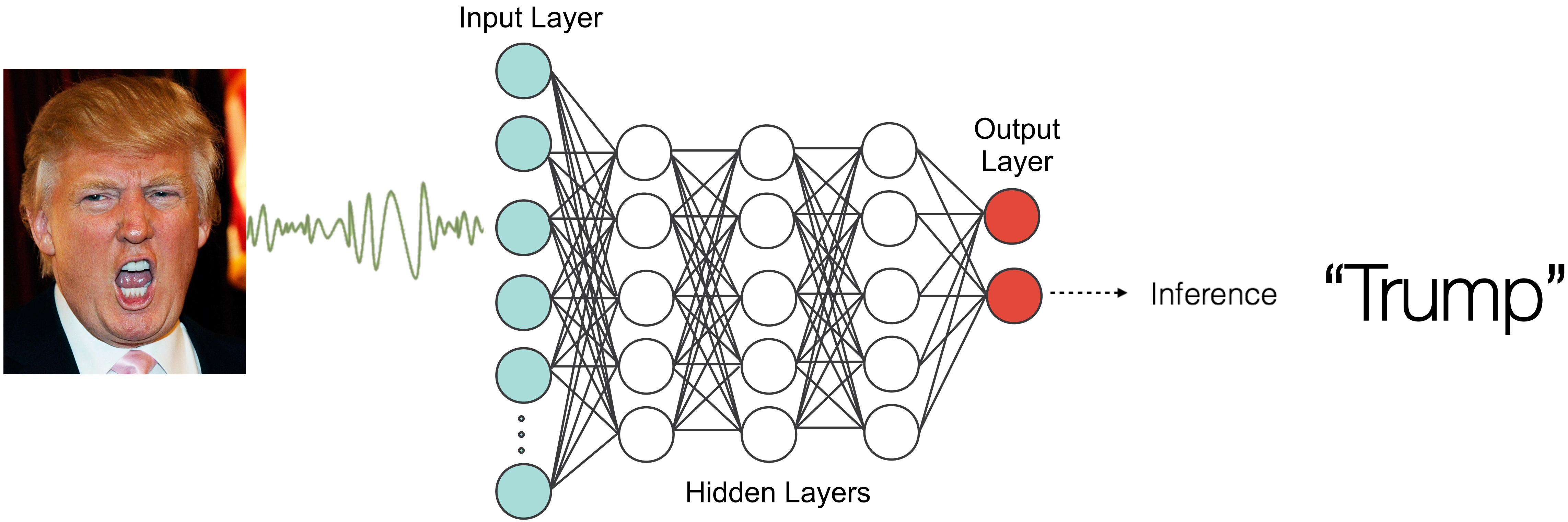
Context Modeling

?

Brief Deep Neural Network (DNN) Background

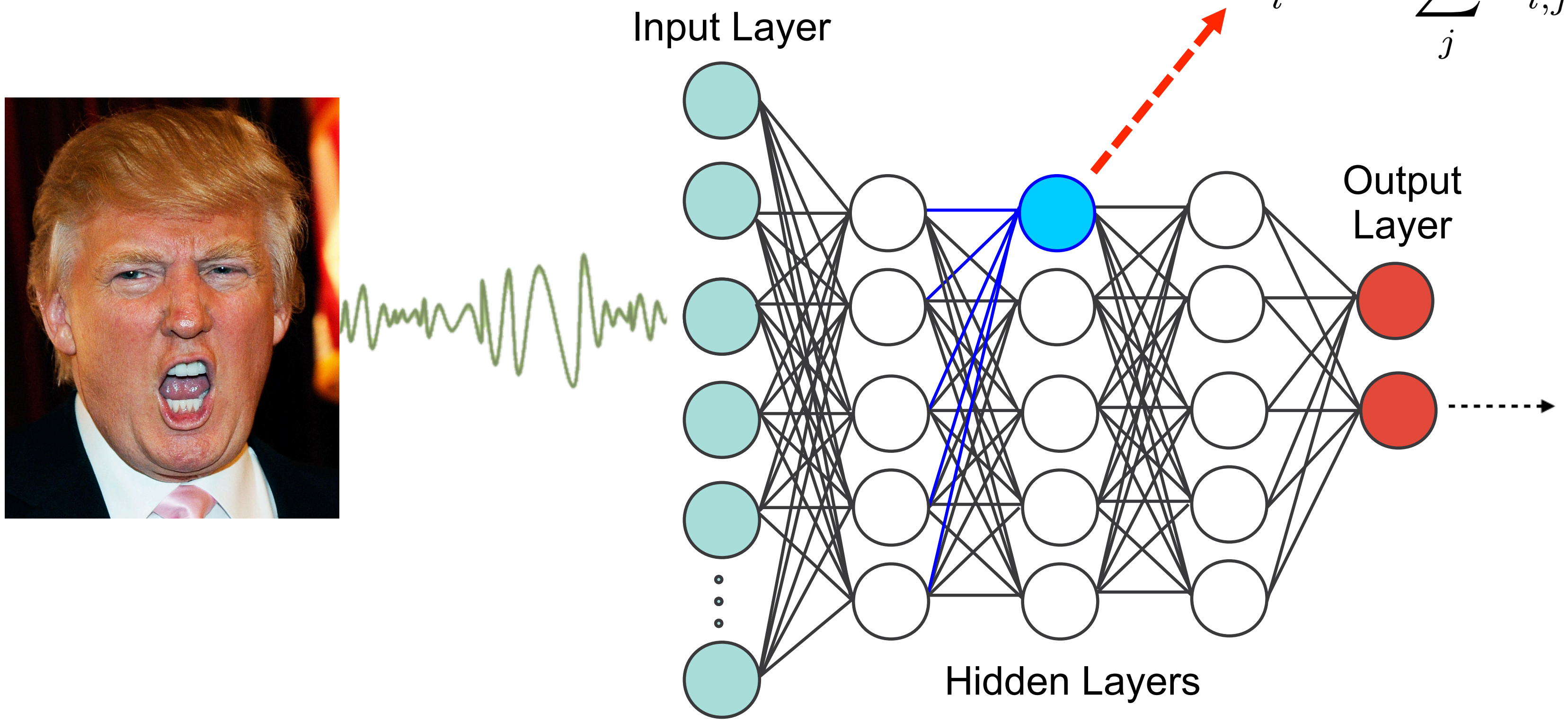


Brief Deep Neural Network (DNN) Background



Brief Deep Neural Network (DNN) Background

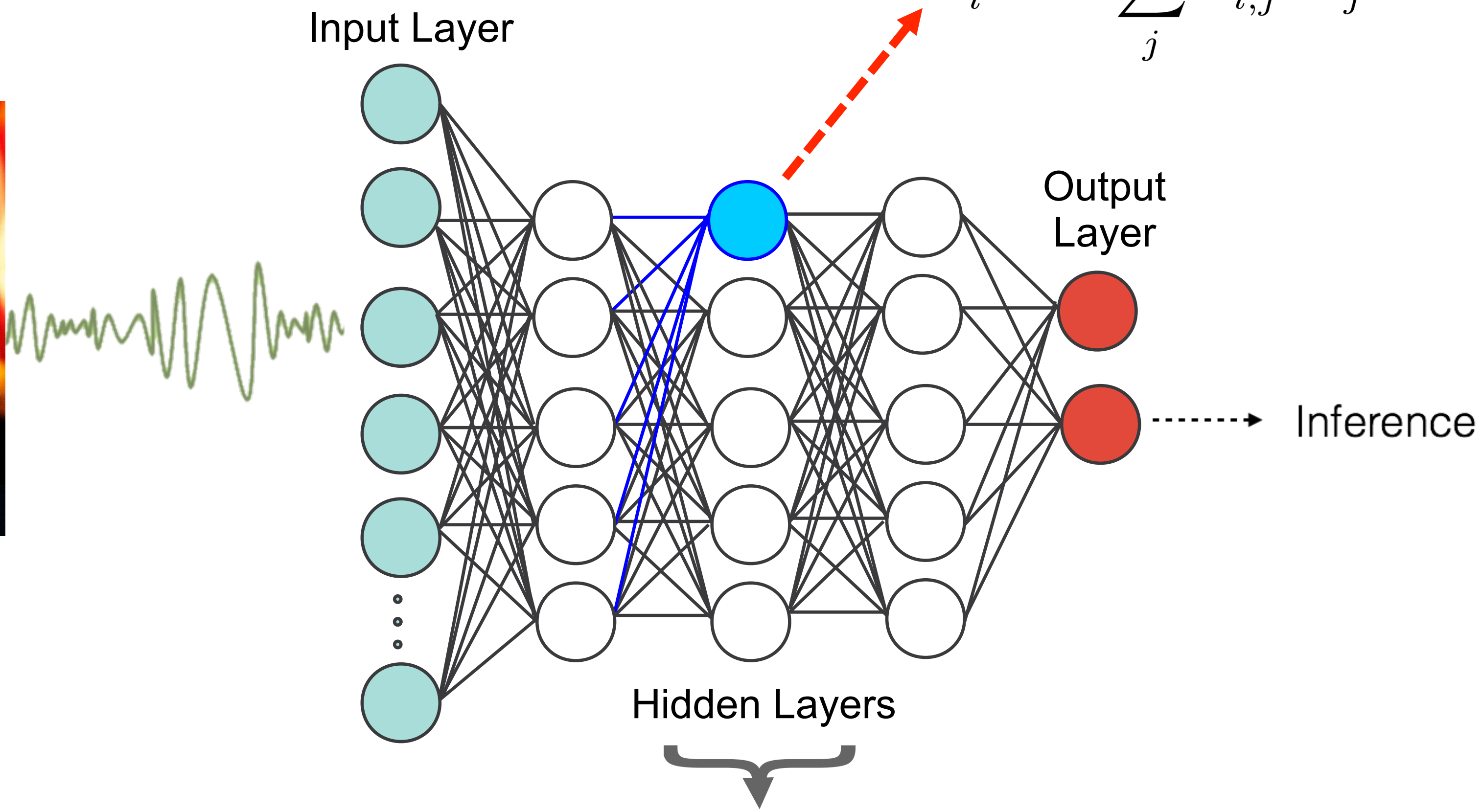
$$x_i^{L+1} = \sum_j w_{i,j}^{L+1} x_j^L \qquad \mathcal{A}_i^{L+1} = \frac{1}{1+\exp(-x_i^{L+1})}$$



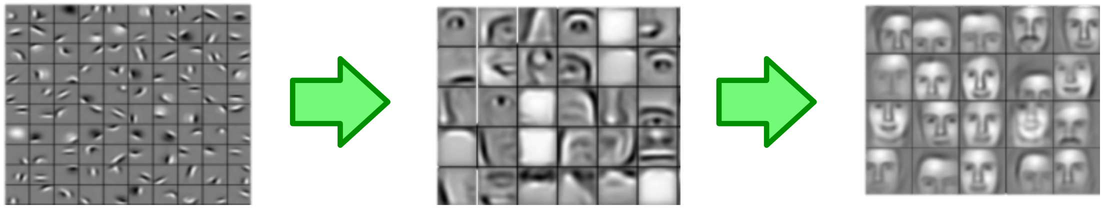
Inference "Trump"

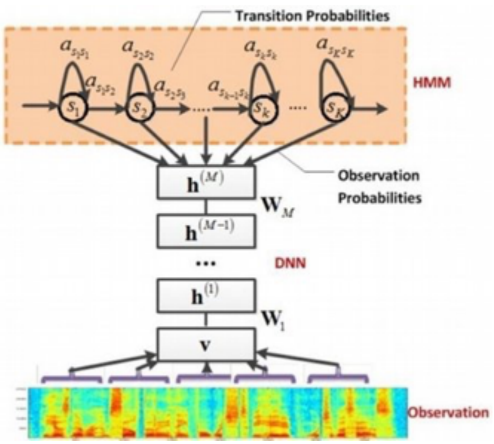
Brief Deep Neural Network (DNN) Background

$$x_i^{L+1} = \sum_j w_{i,j}^{L+1} x_j^L \qquad \mathcal{A}_i^{L+1} = \frac{1}{1+\exp(-x_i^{L+1})}$$

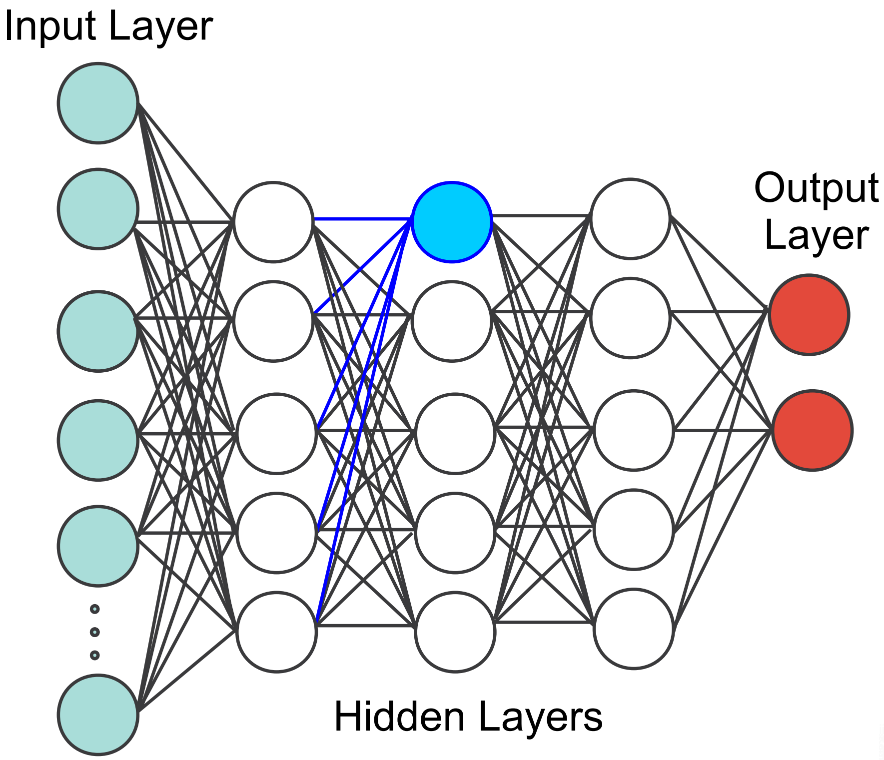


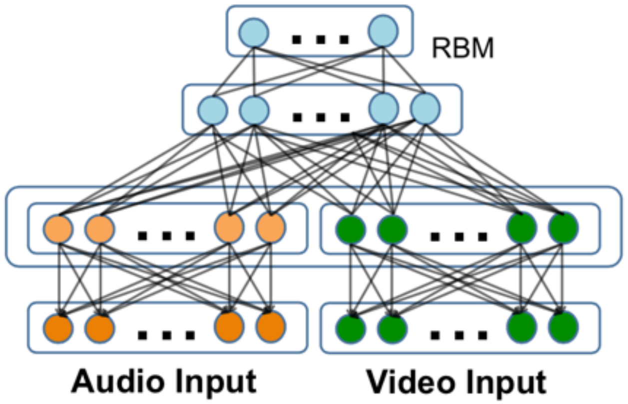
“Trump”



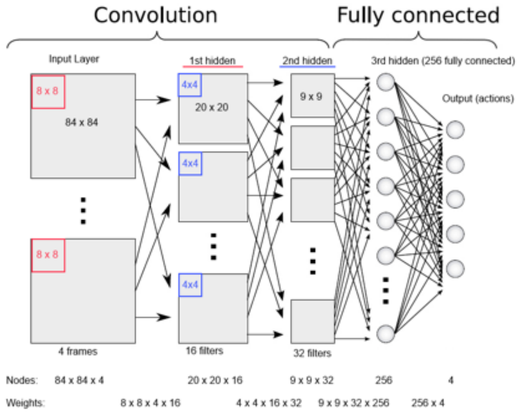


Hybrid DNN-HMM models

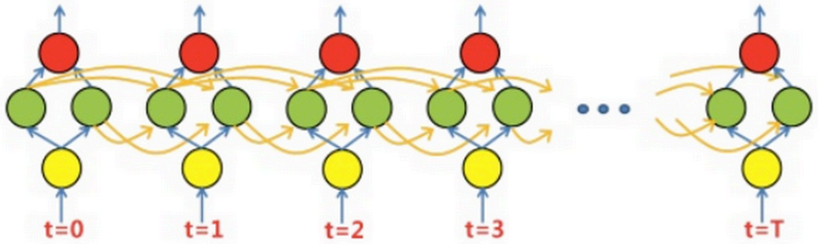




Multi-modal Stacked Networks

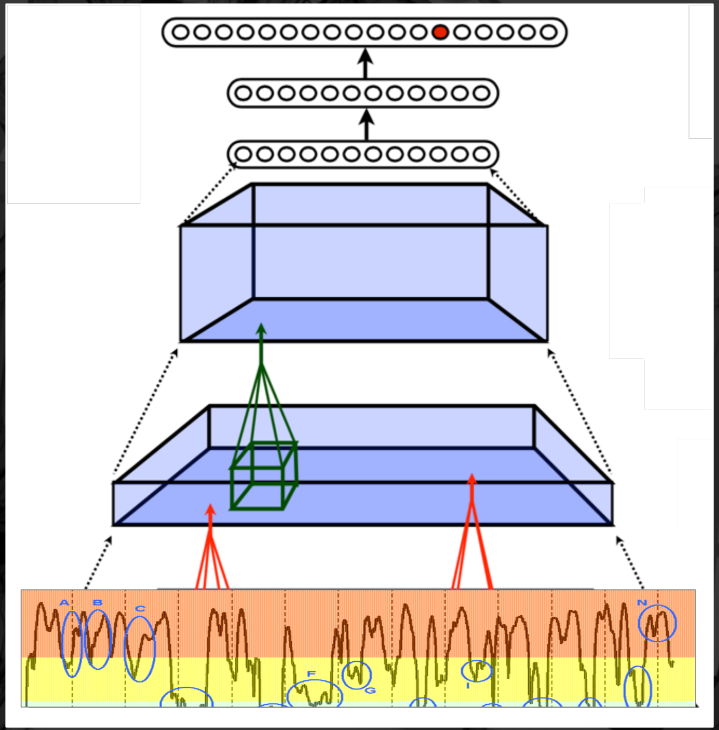


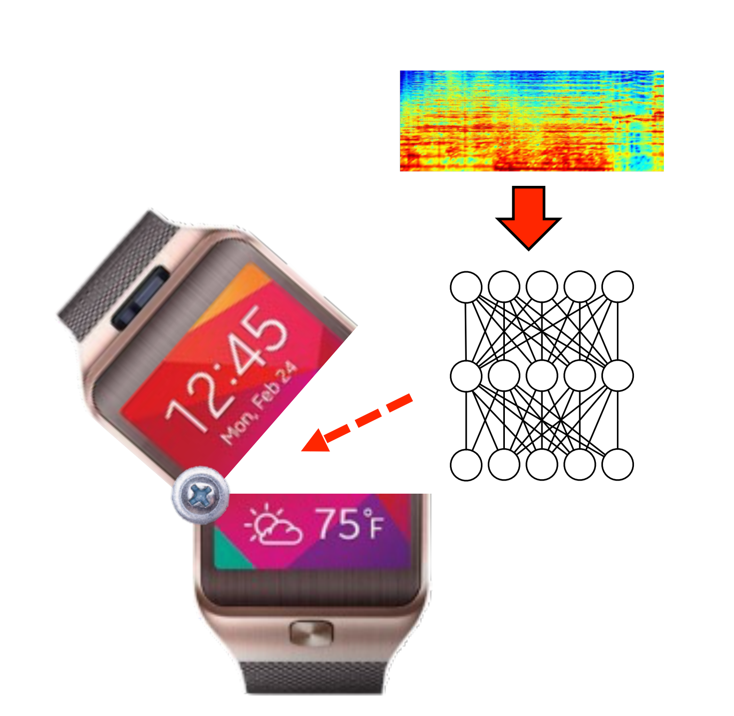
Convolutional Neural Networks



Recurrent Neural Networks

Deep Learning for Mobile Sensing: *Making Baby Steps with DeepEar and DeepX*

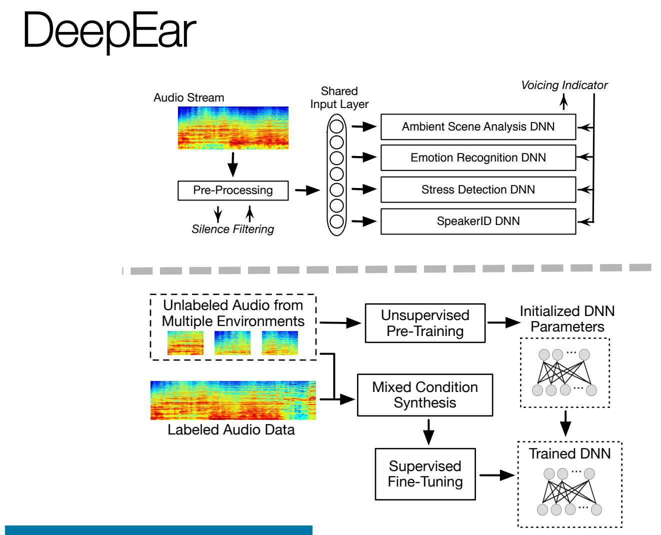




How should context and user behavior be modeled under Deep Learning?

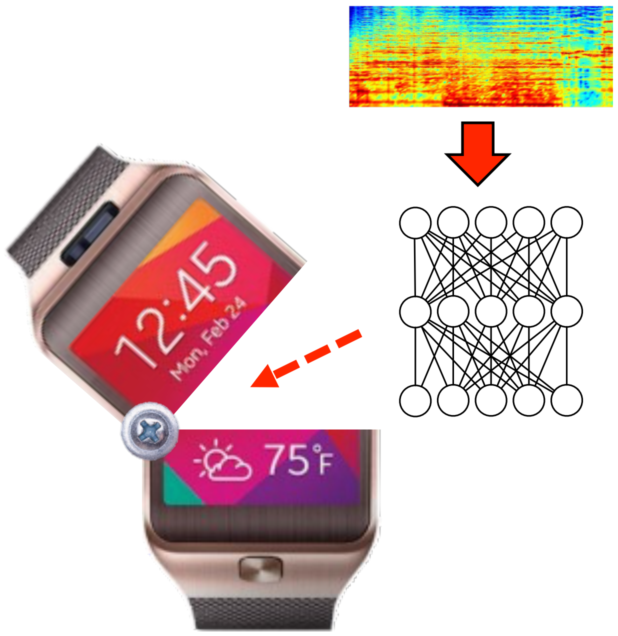
How can we scale down Deep Learning algorithms to run on wearables and phones?

Deep Learning for Mobile Sensing: *Making Baby Steps with DeepEar and DeepX*



NewScientist

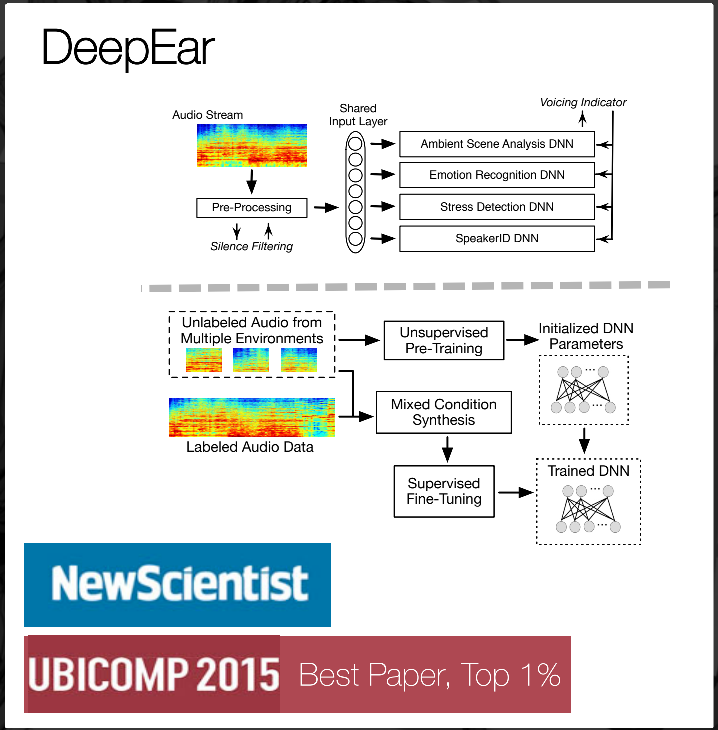
UBICOMP 2015 Best Paper, Top 1%

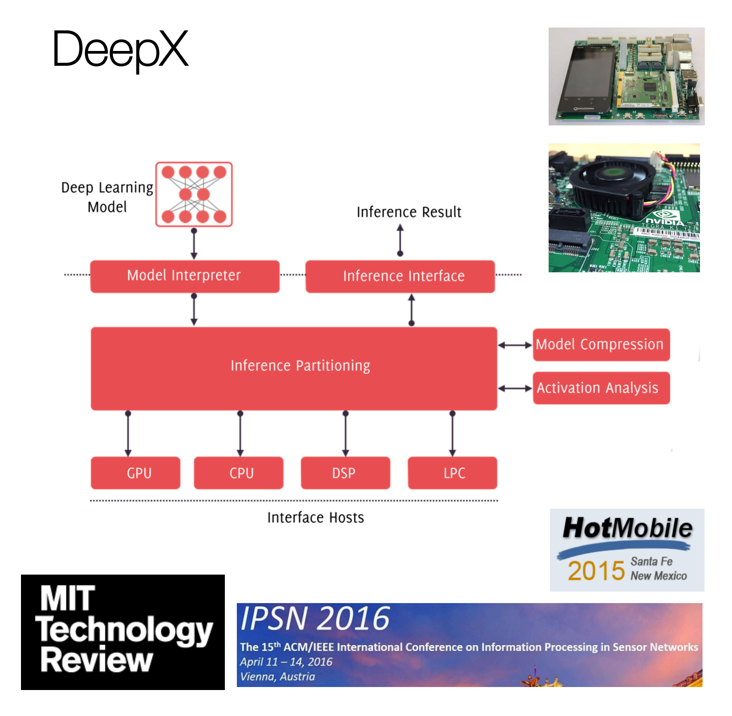


How should context and user behavior be modeled under Deep Learning?

How can we scale down Deep Learning algorithms to run on wearables and phones?

Deep Learning for Mobile Sensing: *Making Baby Steps with DeepEar and DeepX*



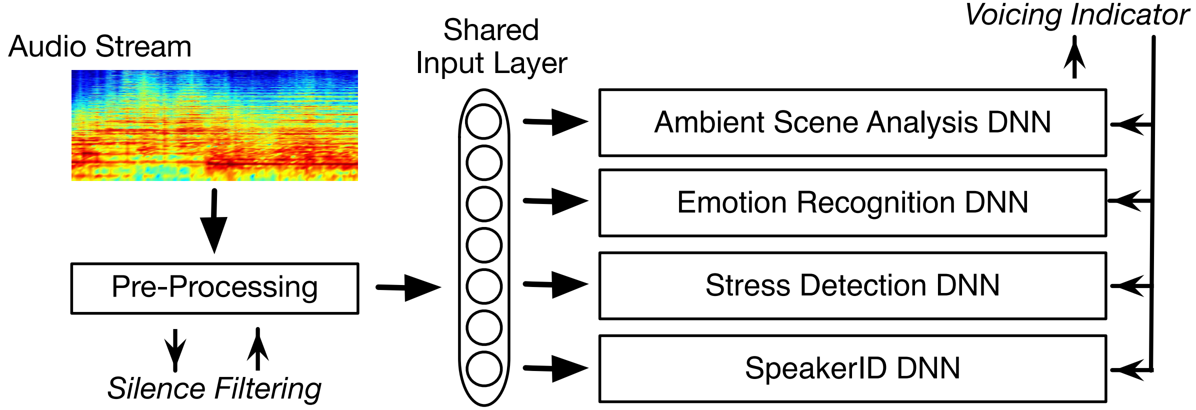


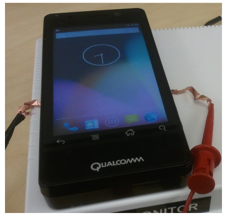
How should **context and user behavior** be modeled under Deep Learning?

How can we **scale down** Deep Learning algorithms to run on wearables and phones?

DeepEar Design: Operation and Model Architecture

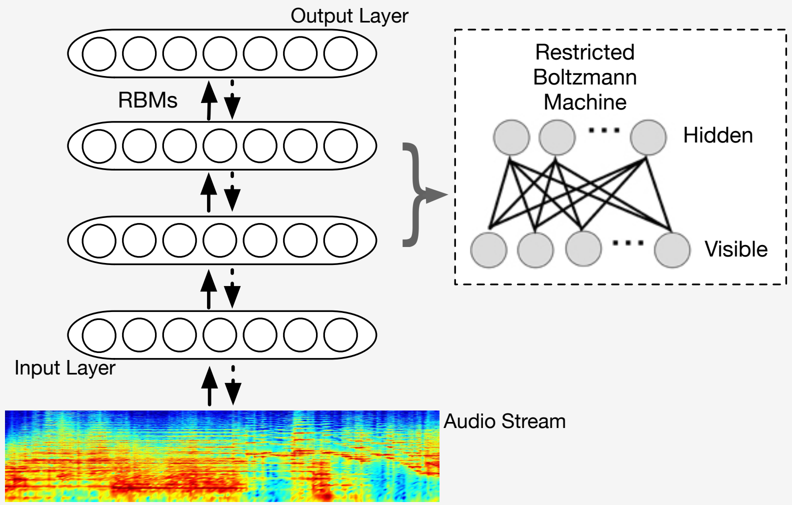
Architecture





Wearables & Smartphones

Internal DNN Design



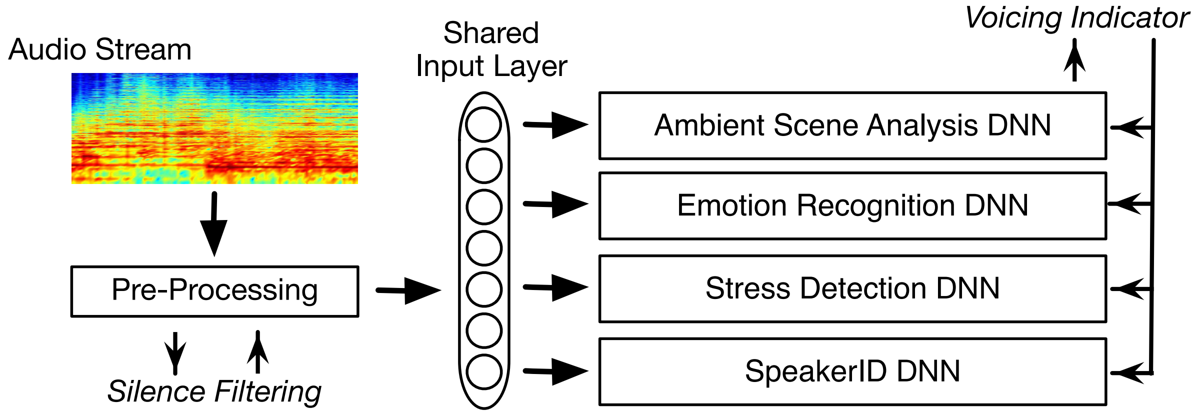
- Concurrent Model Execution
- Same Architecture Across all DNNs
- No Task Specific Features
- MFCC or Freq. Bank Representation
- Modest Model Complexity

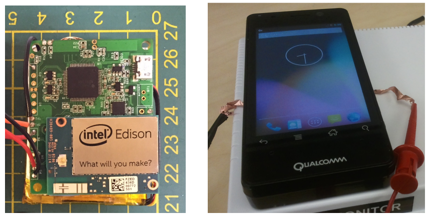
Total Layers	Hidden Layers	Units per Hidden Layer	Total Parameters
5	3	1024	2.3M

"DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments using Deep Learning", Nicholas D. Lane, Petko Georgiev, Lorena Qendro – *UbiComp 2015*

DeepEar Design: Operation and Model Architecture

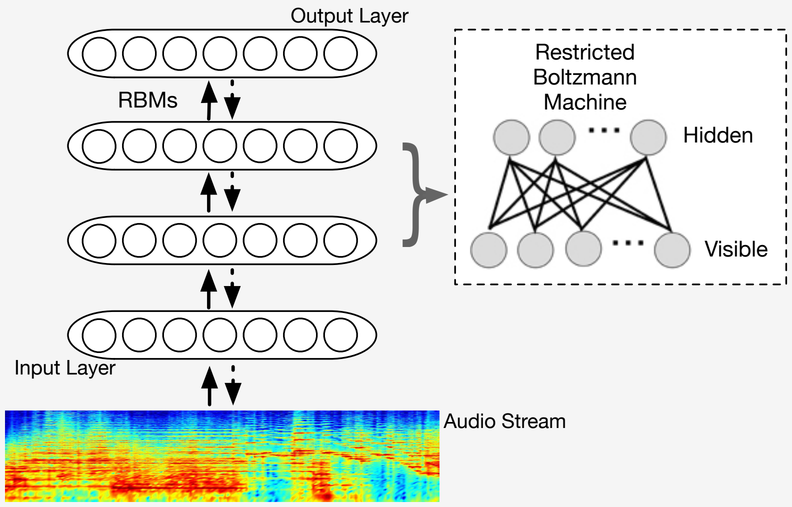
Architecture





Wearables & Smartphones

Internal DNN Design



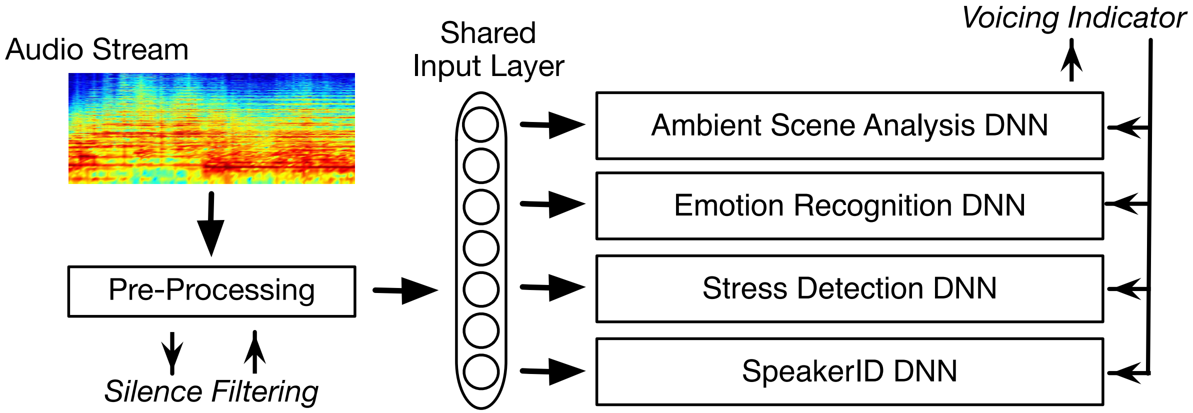
- Concurrent Model Execution
- Same Architecture Across all DNNs
- No Task Specific Features
- MFCC or Freq. Bank Representation
- Modest Model Complexity

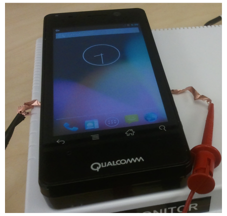
Total Layers	Hidden Layers	Units per Hidden Layer	Total Parameters
5	3	1024	2.3M

"DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments using Deep Learning", Nicholas D. Lane, Petko Georgiev, Lorena Qendro – *UbiComp 2015*

DeepEar Design: Operation and Model Architecture

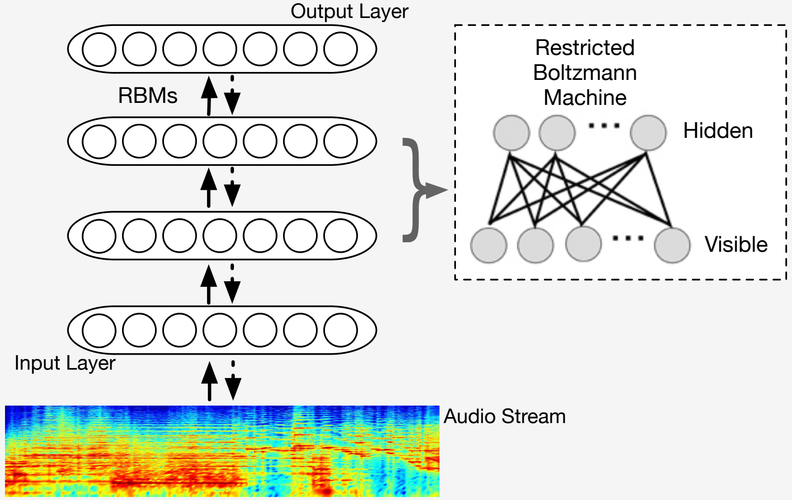
Architecture





Wearables & Smartphones

Internal DNN Design



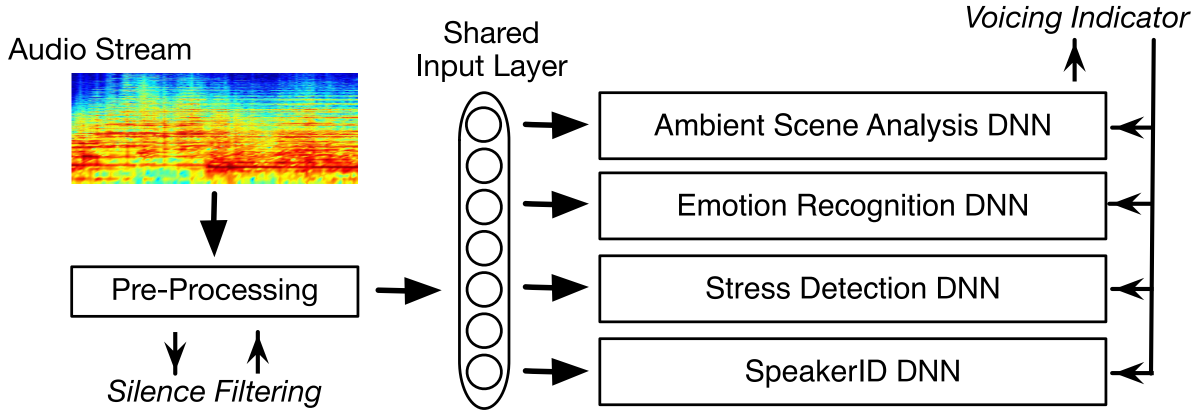
- Concurrent Model Execution
- Same Architecture Across all DNNs
- No Task Specific Features
- MFCC or Freq. Bank Representation
- Modest Model Complexity

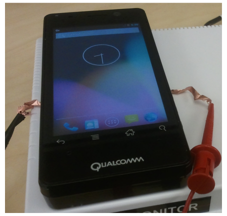
Total Layers	Hidden Layers	Units per Hidden Layer	Total Parameters
5	3	1024	2.3M

"DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments using Deep Learning", Nicholas D. Lane, Petko Georgiev, Lorena Qendro – *UbiComp 2015*

DeepEar Design: Proof-of-Concept Prototype

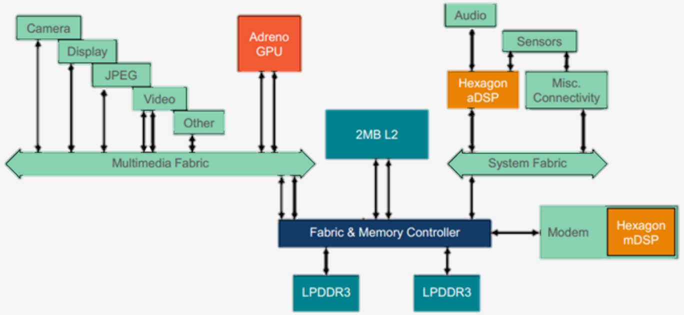
Architecture





Wearables & Smartphones

Smartphone Prototype



Qualcomm SnapDragon 800 SoC

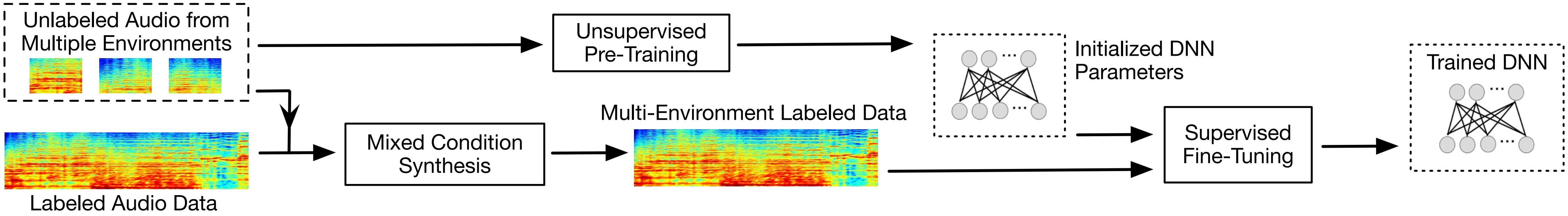
- DSP to Microphone only
- Memory Acute Bottleneck
- Reduced Architecture but with Negligible Accuracy Loss

Audio Sensing Task	DNN Size (Original)	DNN Size (Downscaled)	Period
Ambient Scene Analysis	3×1024	3×256	$1.28s$
Emotion Recognition	3×1024	3×512	$5.00s$
Speaker Identification	3×1024	3×512	$5.00s$

"DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments using Deep Learning", Nicholas D. Lane, Petko Georgiev, Lorena Qendro – *UbiComp 2015*

DeepEar Design: Model Training Pipeline

Overview



Distinctions from Typical Training Phases

(1) Use of Pre-Training

- Secondary use of unlabeled data
- Compensates for lack of labeled data

(2) Role of Labels and Unlabeled Data

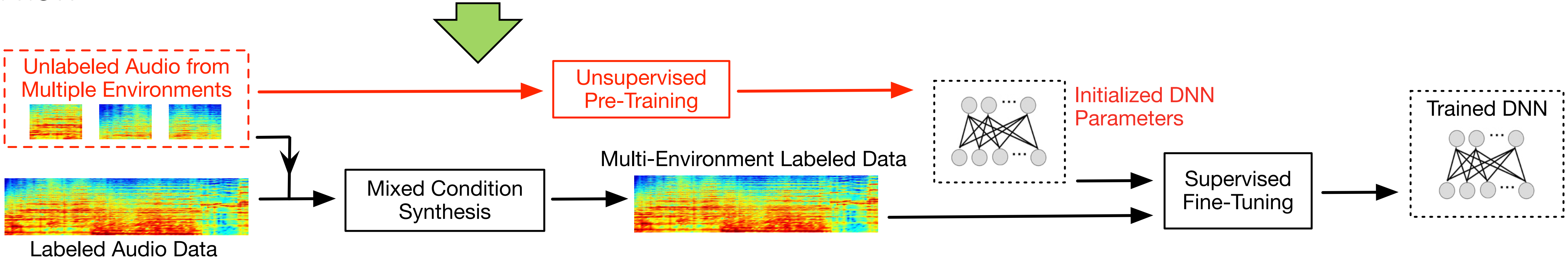
- Specifically Capture Environment Diversity
- Label Synthesis (includes intensity)

(3) No Task-Specific Stages

- No task selected features or stages
- All training for models virtually the same

DeepEar Design: Model Training Pipeline

Overview



Distinctions from Typical Training Phases

(1) Use of Pre-Training

- Secondary use of unlabeled data
- Compensates for lack of labeled data

(2) Role of Labels and Unlabeled Data

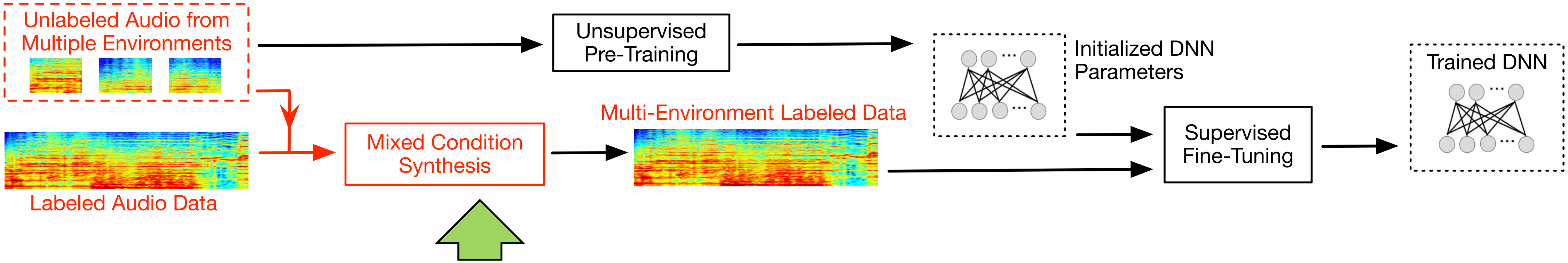
- Specifically Capture Environment Diversity
- Label Synthesis (includes intensity)

(3) No Task-Specific Stages

- No task selected features or stages
- All training for models virtually the same

DeepEar Design: Model Training Pipeline

Overview



Distinctions from Typical Training Phases

(1) Use of Pre-Training

- Secondary use of unlabeled data
- Compensates for lack of labeled data

(2) Role of Labels and Unlabeled Data

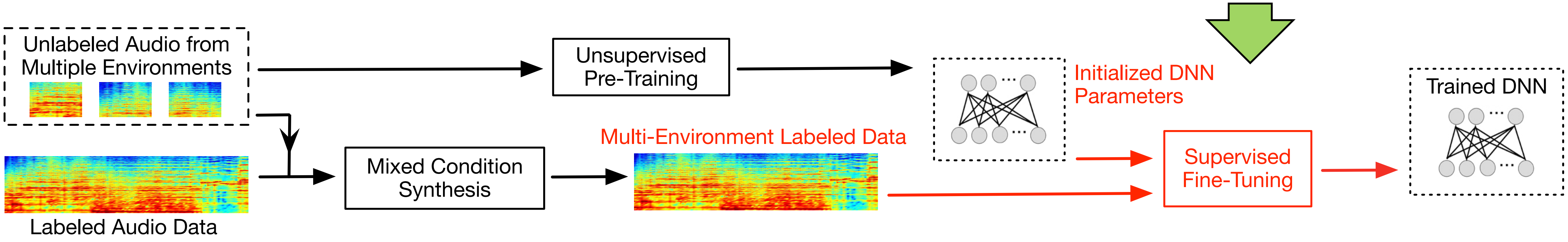
- Specifically Capture Environment Diversity
- Label Synthesis (includes intensity)

(3) No Task-Specific Stages

- No task selected features or stages
- All training for models virtually the same

DeepEar Design: Model Training Pipeline

Overview



Distinctions from Typical Training Phases

(1) Use of Pre-Training

- Secondary use of unlabeled data
- Compensates for lack of labeled data

(2) Role of Labels and Unlabeled Data

- Specifically Capture Environment Diversity
- Label Synthesis (includes intensity)

(3) No Task-Specific Stages

- No task selected features or stages
- All training for models virtually the same

Experiment Methodology

Baseline Systems

- EmotionSense (UbiComp 2010)
- StressSense (UbiComp 2012)
- SpeakerSense (Pervasive 2011)
- SoundSense (MobiSys 2009)

Model Setup

- Speaker Identification :: {23 different speakers}
- Stress Detection :: {stressed, not stressed}
- Emotion :: {happiness, sadness, fear, anger, neutral}
- Ambient Scene :: {music, traffic, voicing, other}

Audio Datasets

- Labeled data for each model setup
- Background noise 168 place visits

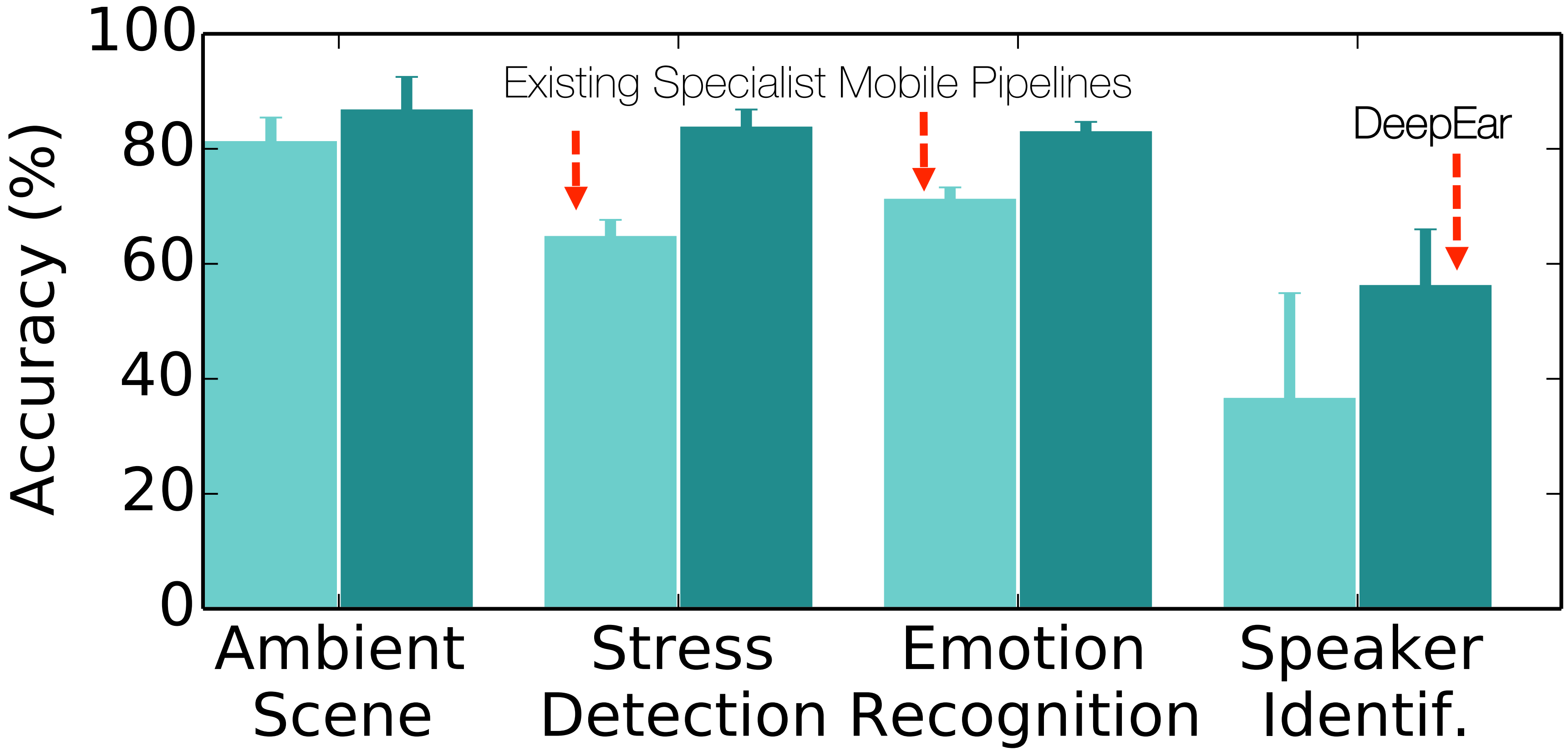
  (50 unique places)

Place Visit Dataset (WWW '14)

DeepEar outperforms specialist mobile audio sensing pipelines across multiple scenarios

168

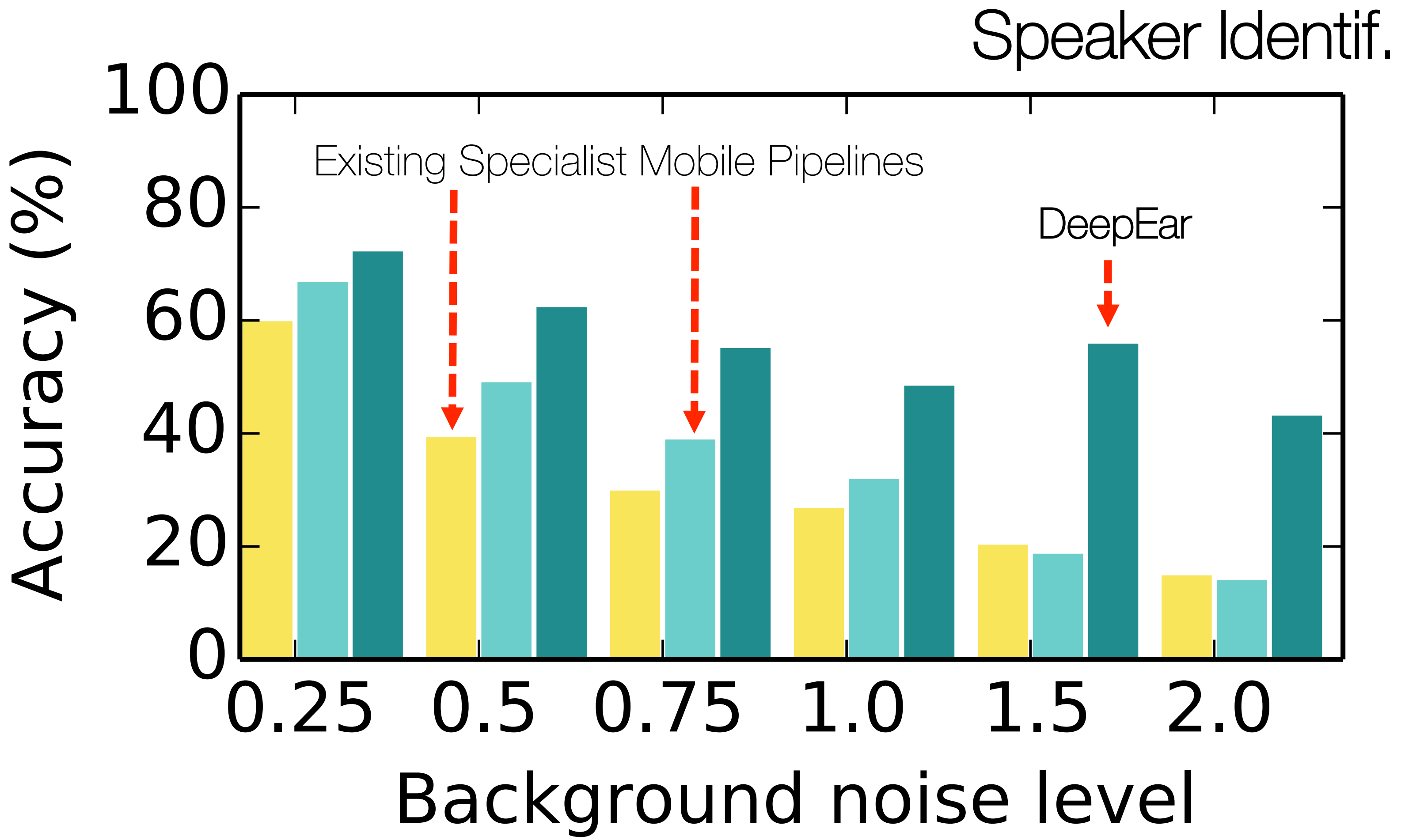
Place Visits



DeepEar shows increased robustness to a wide spectrum of background noise levels

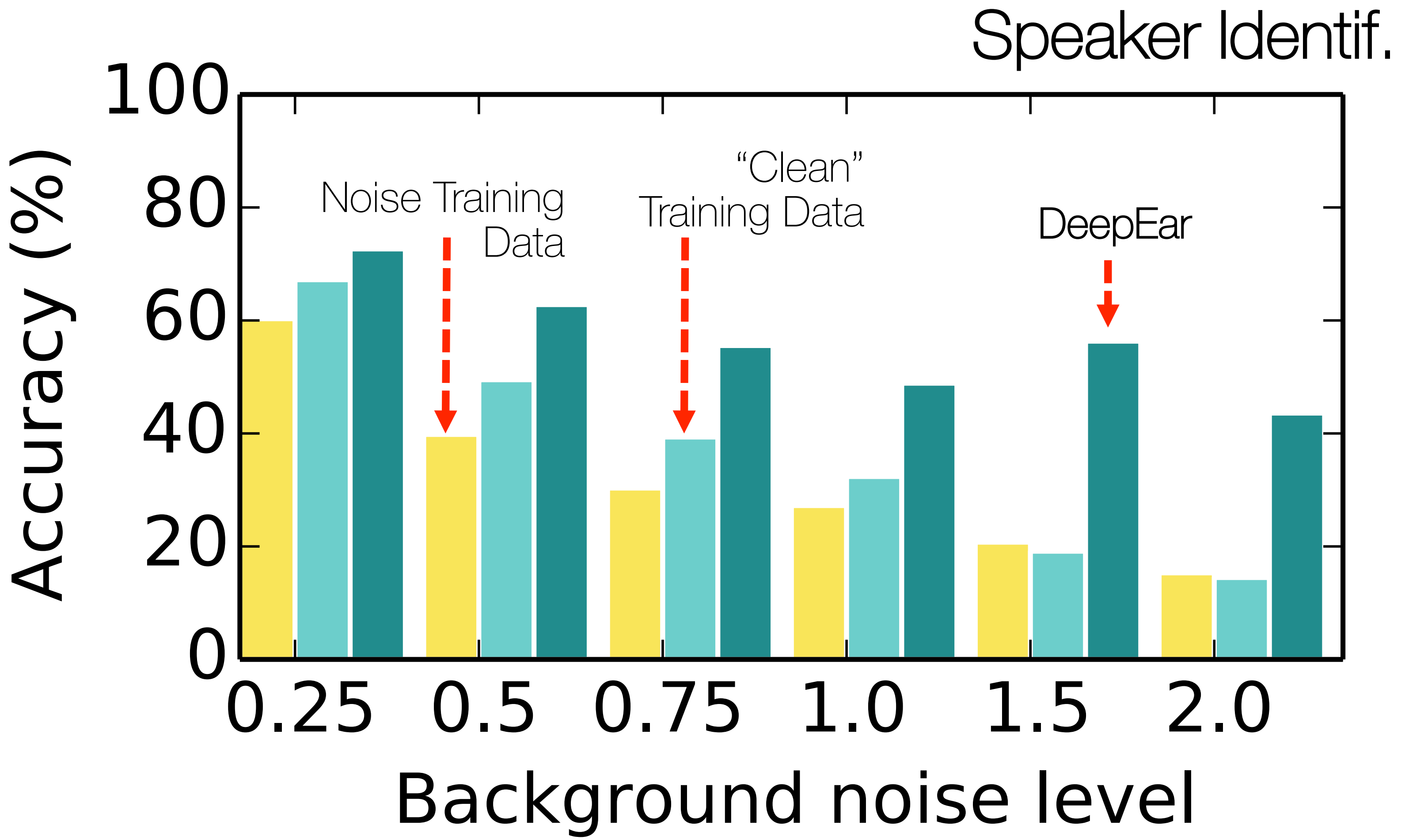
168

Place Visits



DeepEar shows increased robustness to a wide spectrum of background noise levels

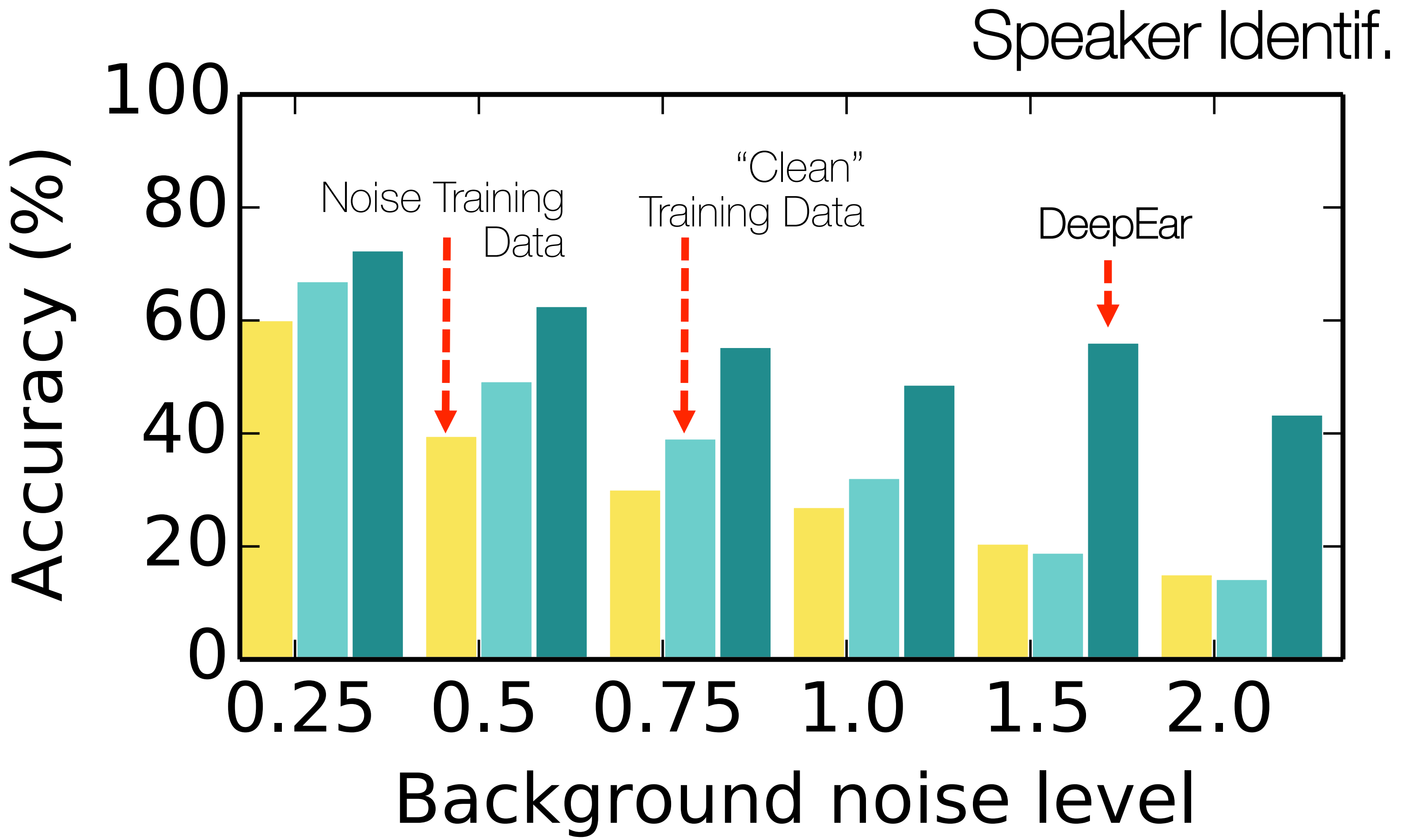
168
Place Visits



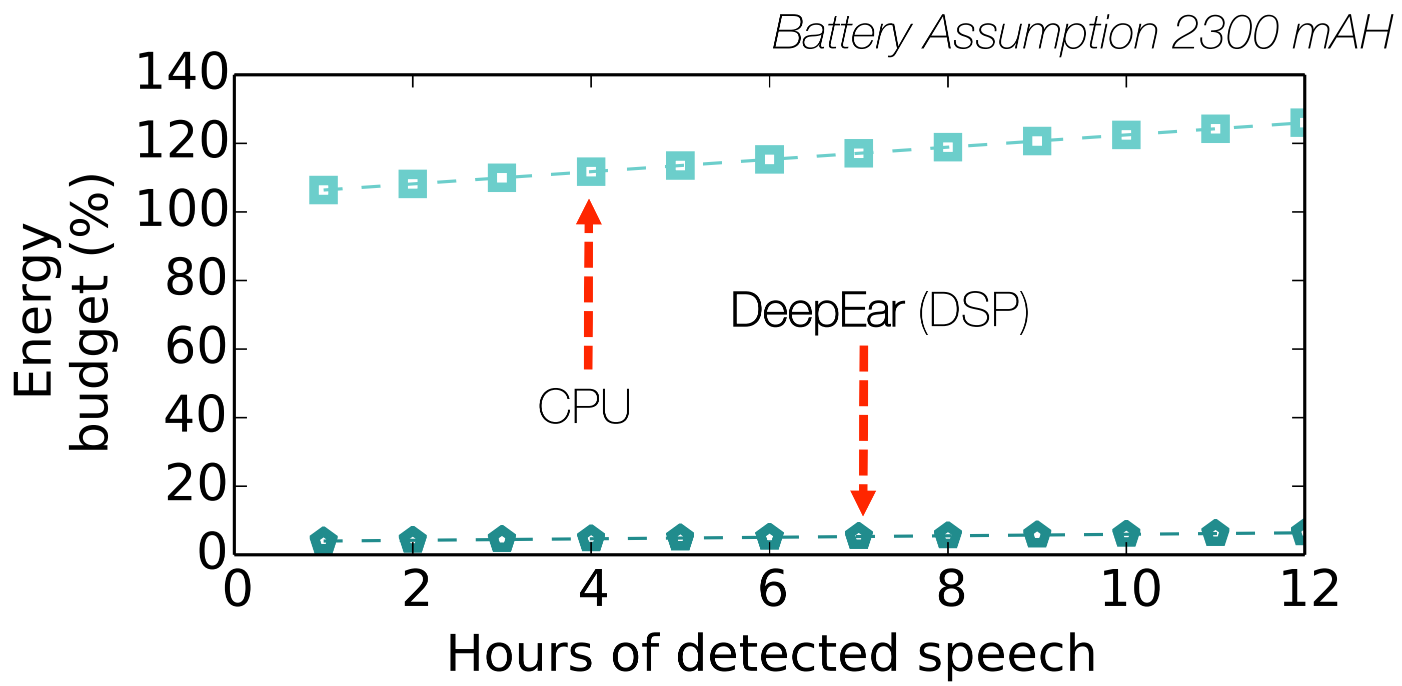
DeepEar shows increased robustness to a wide spectrum of background noise levels

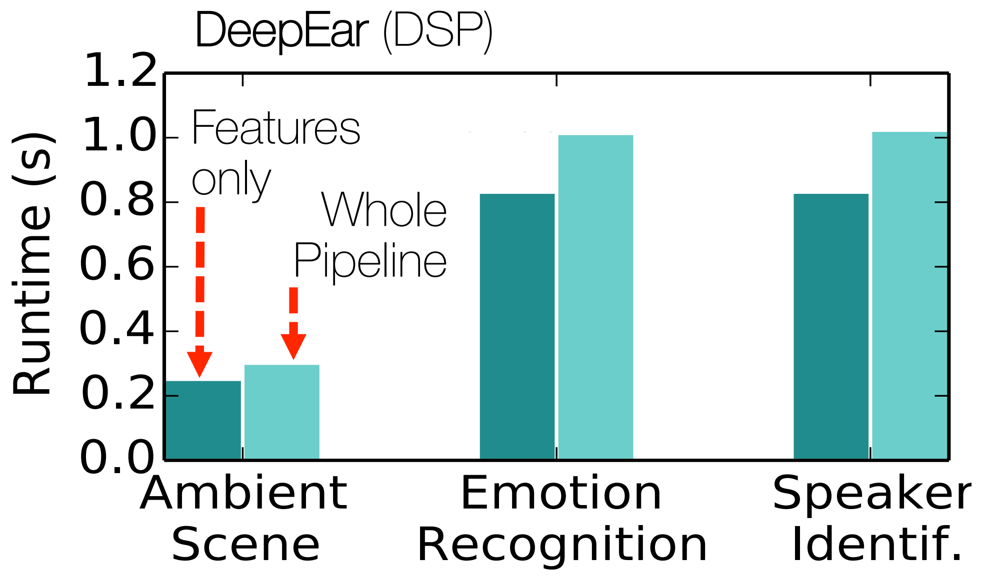
Results also hold for:

- ✅ Stress Detection
- ✅ Emotion Recognition
- ✅ Ambient Scene

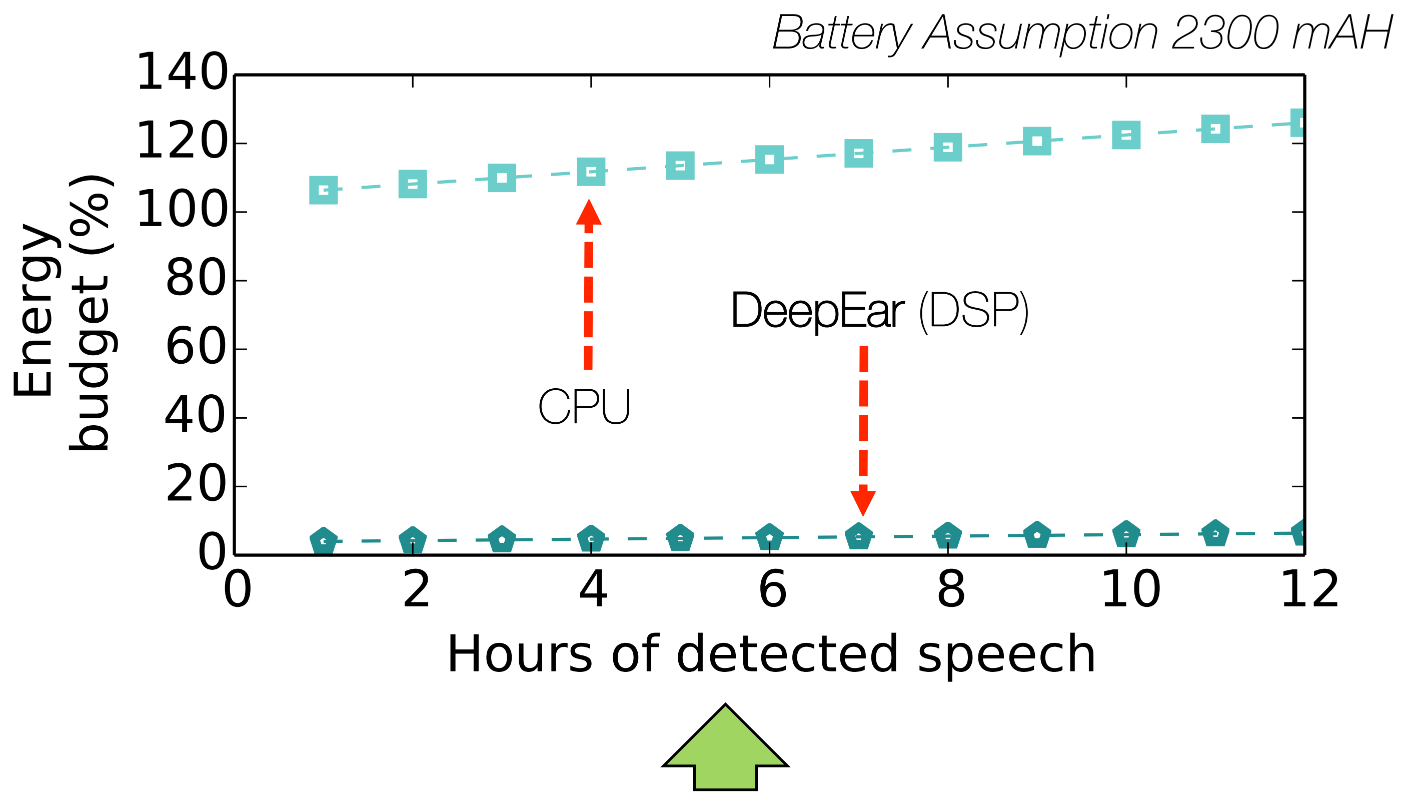


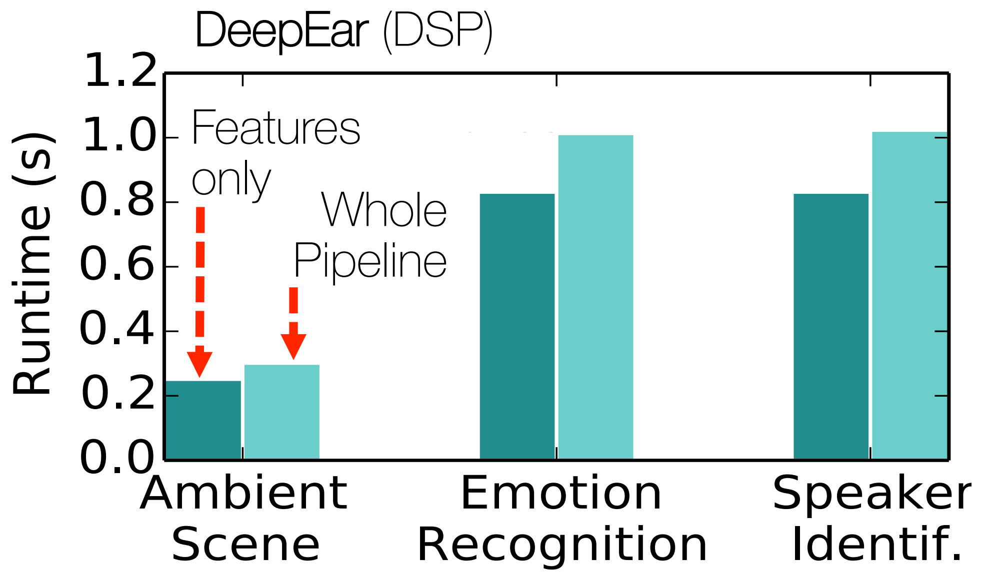
DeepEar Performance: Low-energy overhead and three simultaneous inferences in near real-time



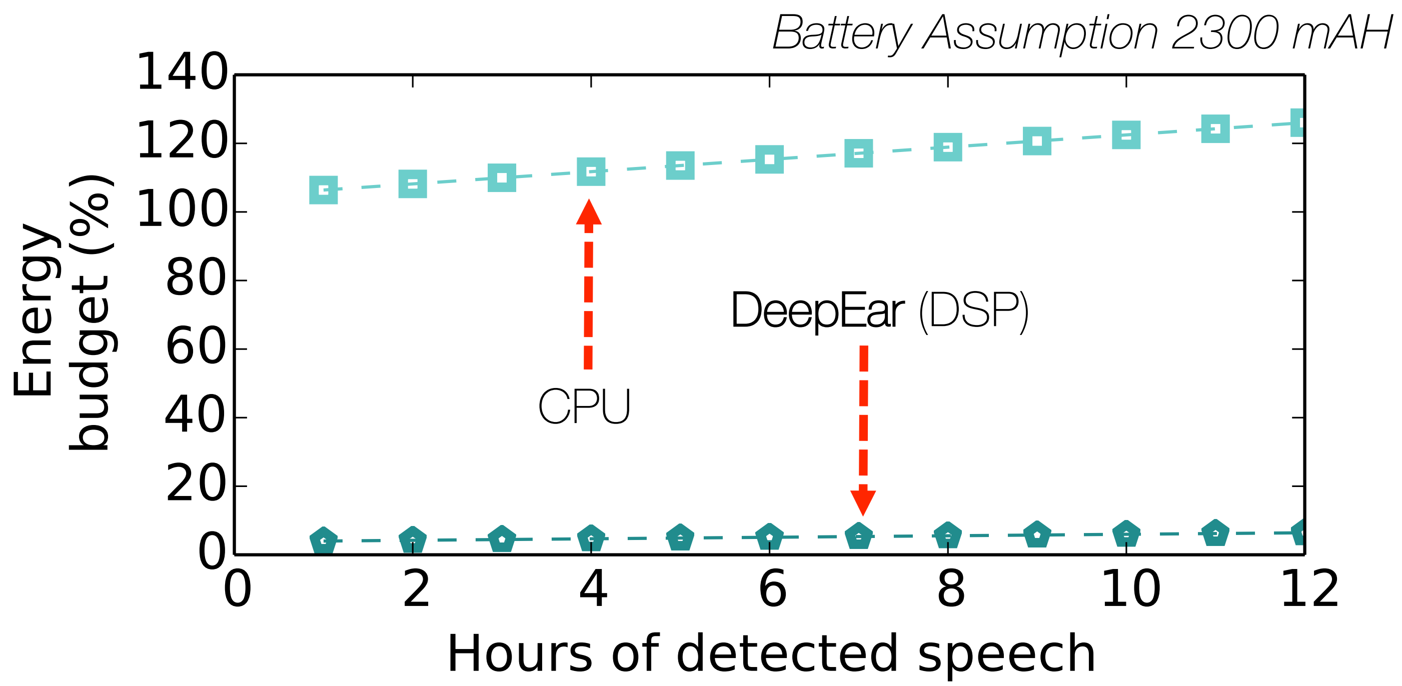


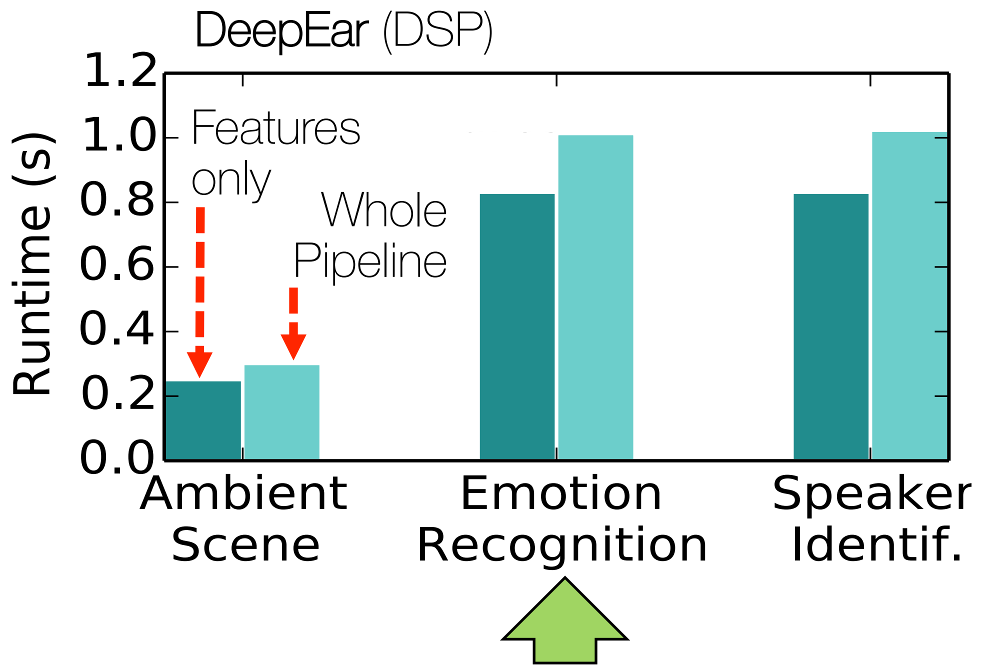
DeepEar Performance: Low-energy overhead and three simultaneous inferences in near real-time



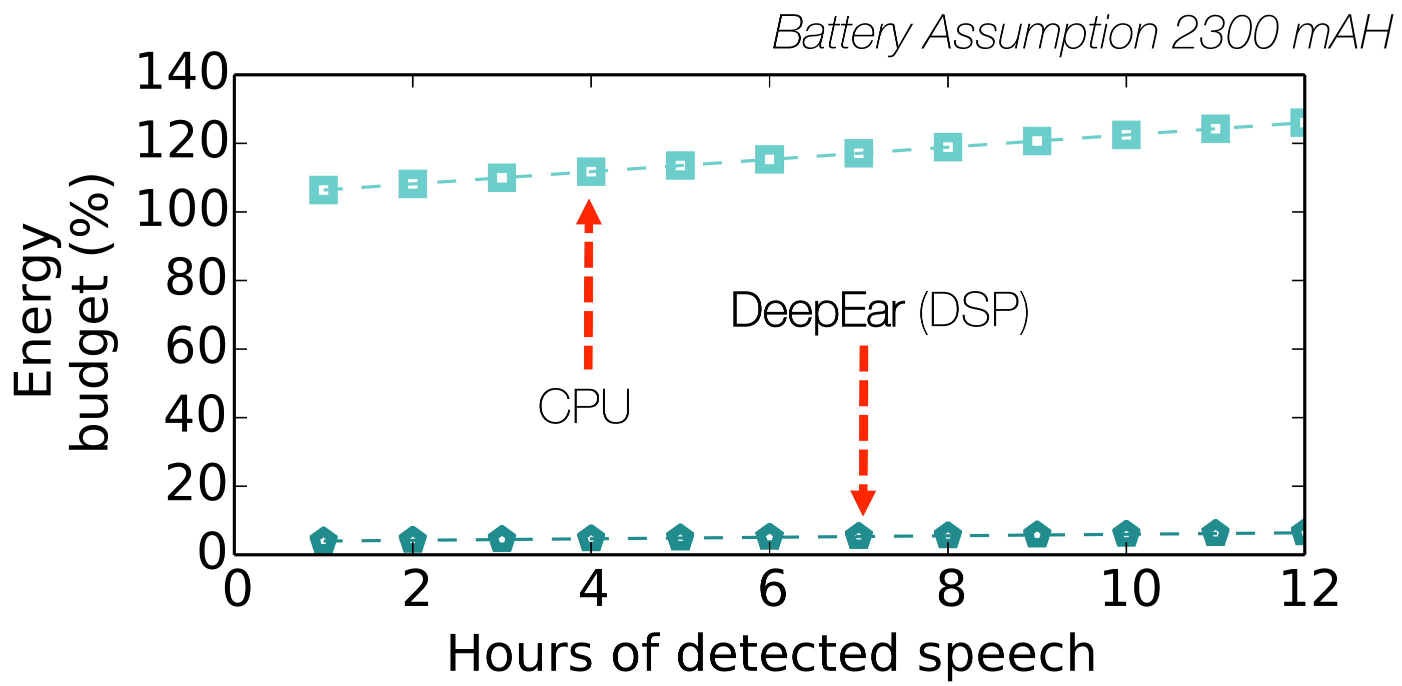


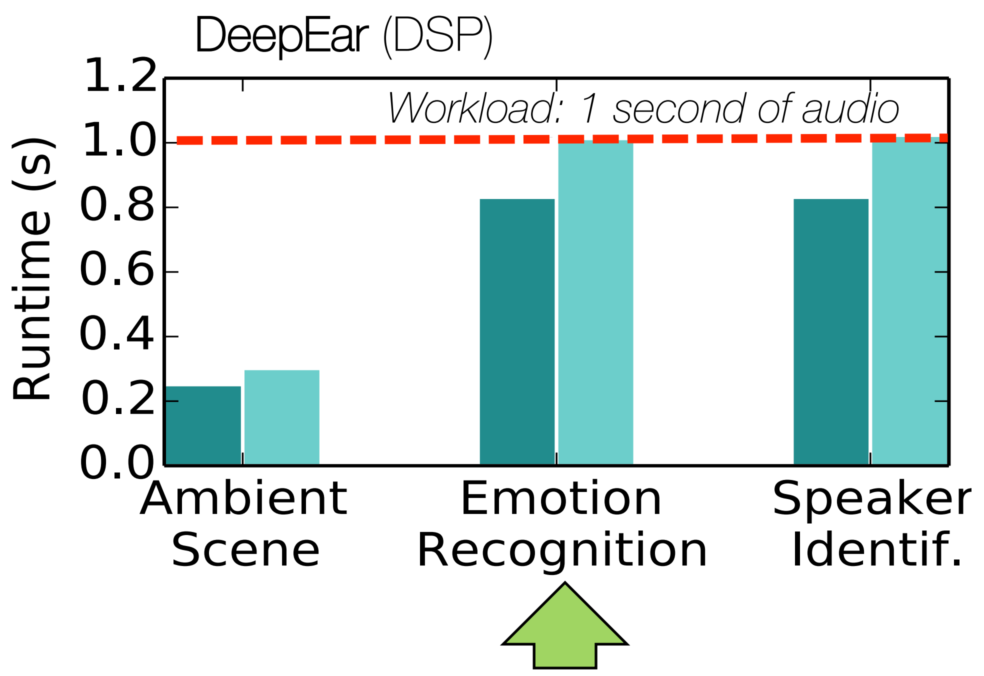
DeepEar Performance: Low-energy overhead and three simultaneous inferences in near real-time



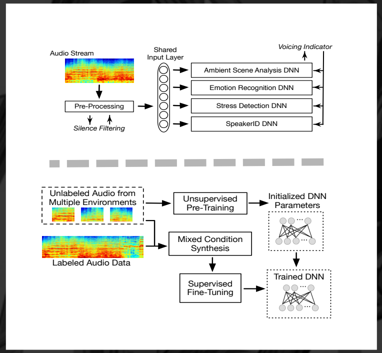


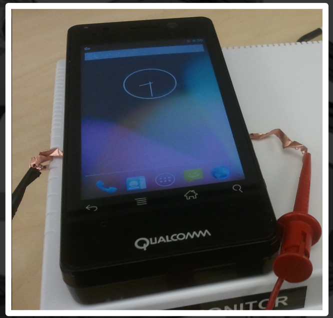
DeepEar Performance: Low-energy overhead and three simultaneous inferences in near real-time





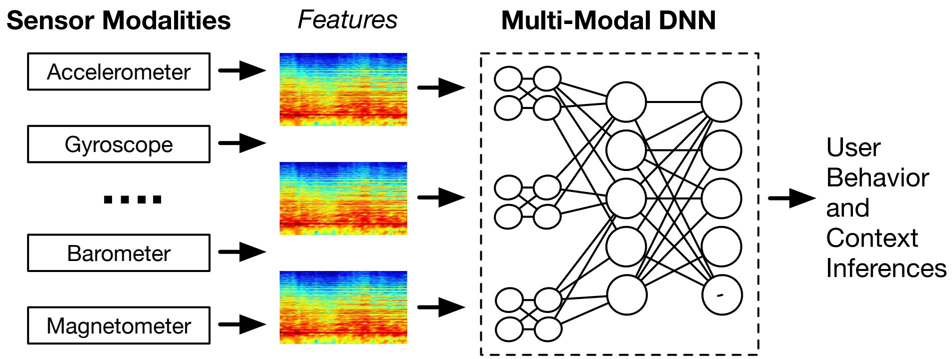
DeepEar: Progress Towards Mobile Deep Learning

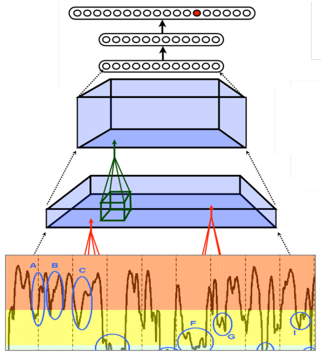


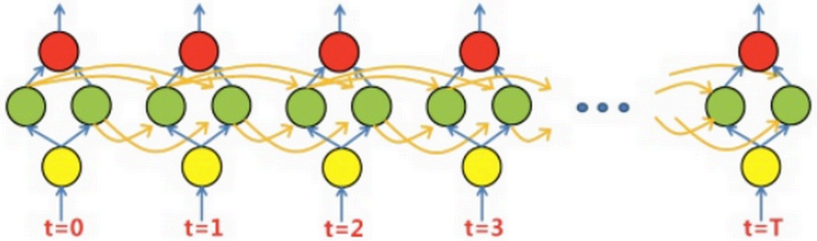


How should context and user behavior be modeled under Deep Learning?

Ongoing study of deep models for multimodal and especially inertial data

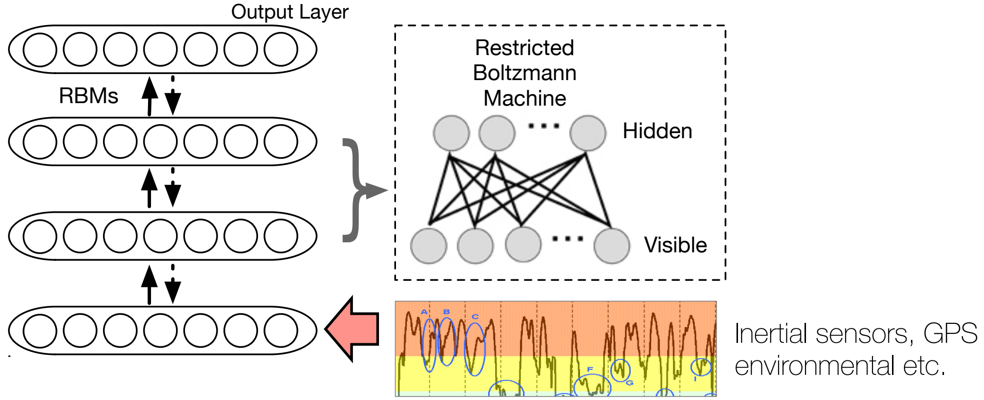


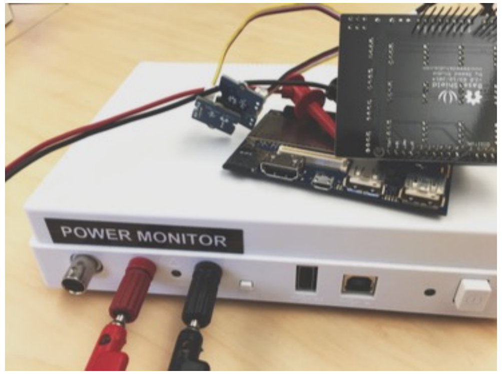


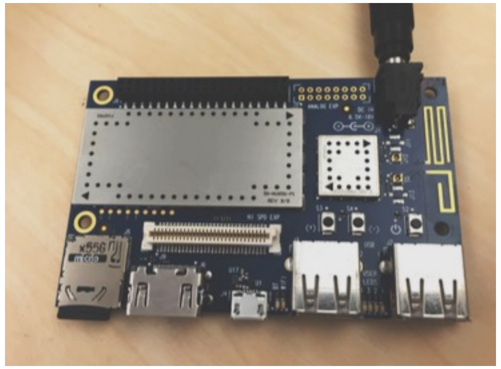


Latest Modeling Result: Smartwatch prototype with Context & Activity Inferences from a Deep Model

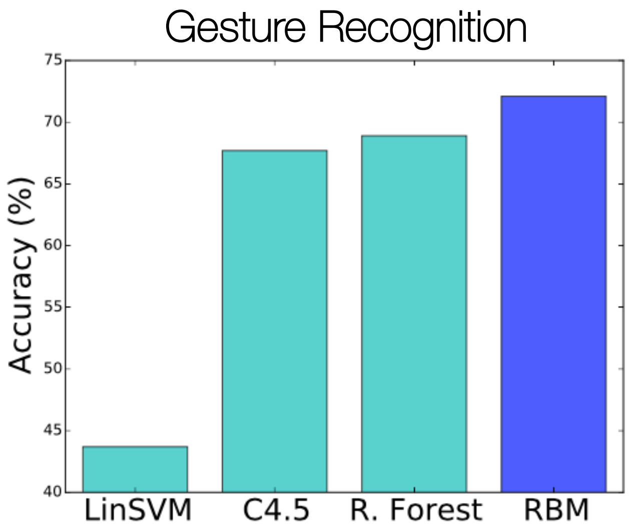
"From Smart to Deep: Robust Activity Recognition on Smartwatches using Deep Learning", Sourav Bhattacharya, Nicholas D. Lane – *WristSense 2016*

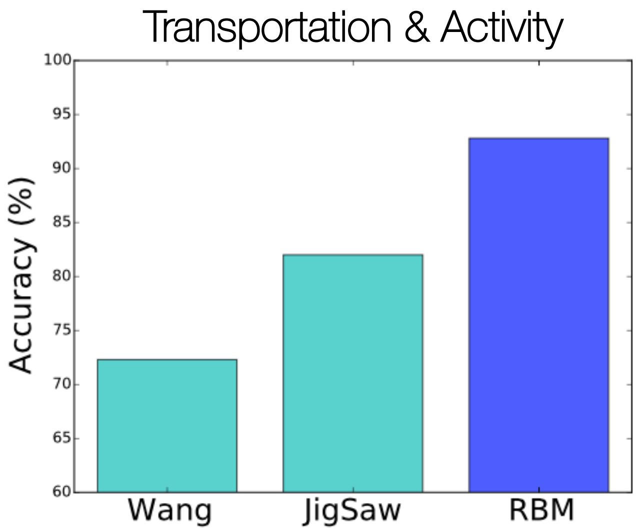


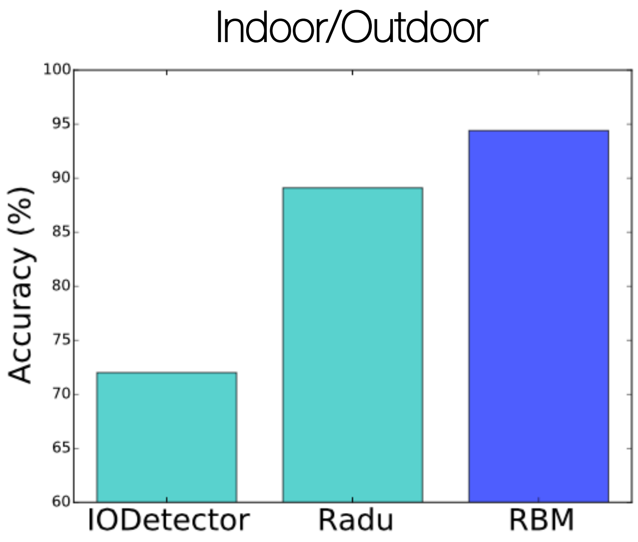




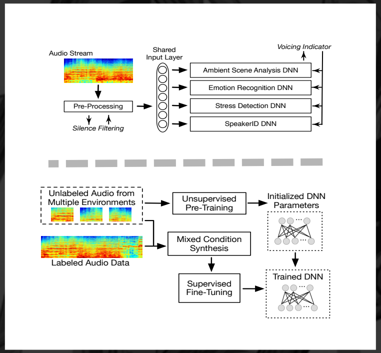
Memory	Battery Life	Execution Time *(whole pipeline)*	Execution Time *(RBM model-only)*
1066KB	32 hrs	5.00 msec	0.94 msec.

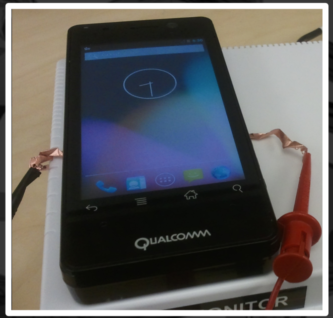






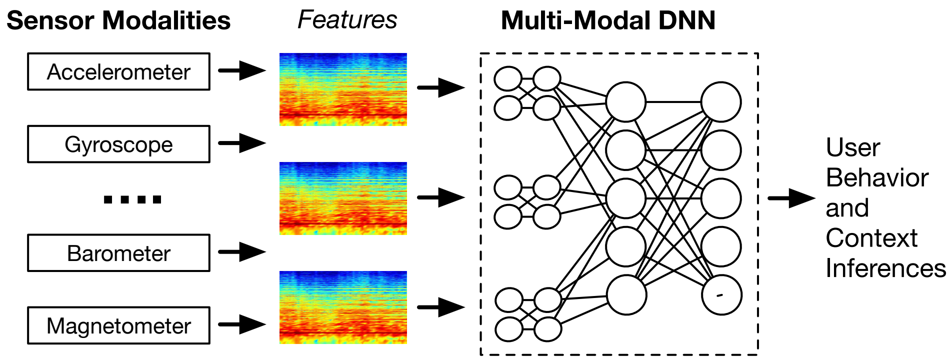
DeepEar: Progress Towards Mobile Deep Learning

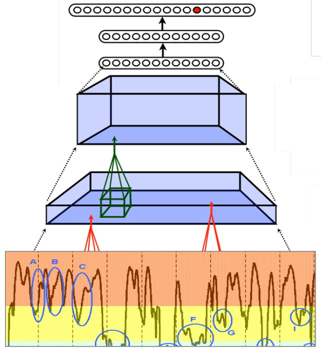


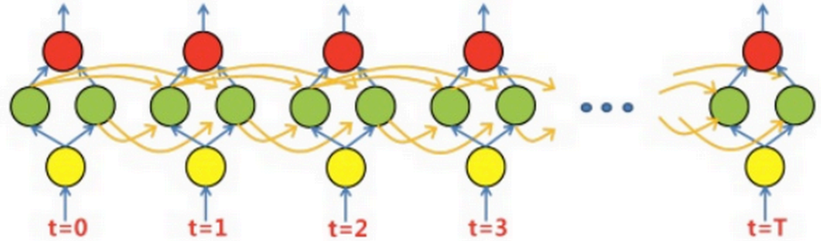


How should context and user behavior be modeled under Deep Learning?

Ongoing study of deep models for multimodal and especially inertial data







How can we scale down Deep Learning algorithms to run on wearables and phones?

Representative Mobile Hardware Bottlenecks

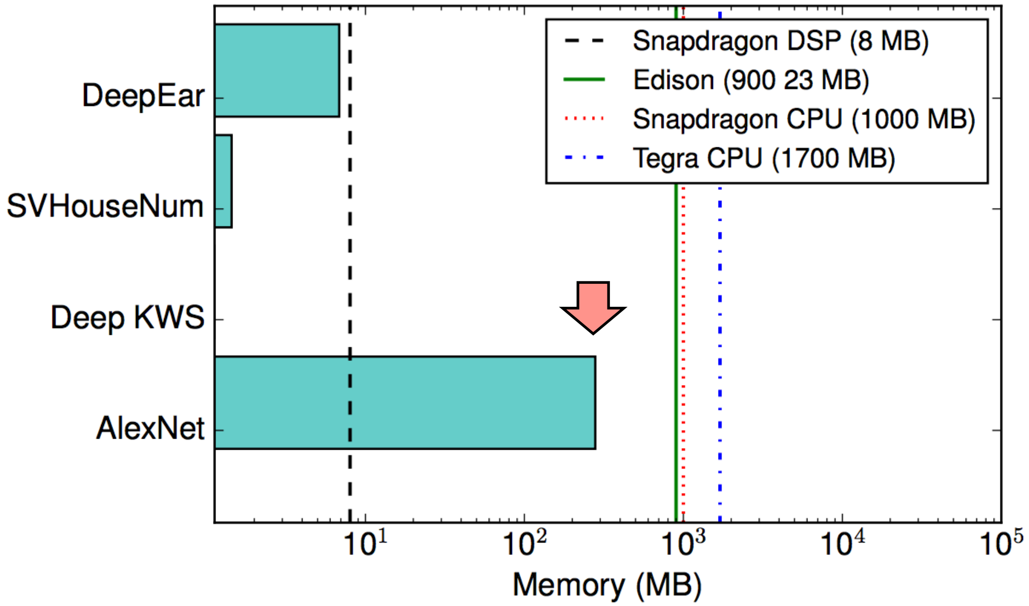
Target Models

	Type	Size	Architecture
AlexNet	CNN	60.9M	$c{:}5^{\imath}$; $p{:}3^{\ddagger}$; $h{:}2^{\star}$; $n{:}\{\text{all } 4096\}^{\dagger}$
SVHN	CNN	313K	$c{:}2^{\imath}$; $p{:}2^{\ddagger}$; $h{:}2^{\star}$; $n{:}\{1600{,}128\}^{\dagger}$
Deep KWS	DNN	241K	$h{:}3^{\star}$; $n{:}\{\text{all } 128\}^{\dagger}$
DeepEar	DNN	2.3M	$h{:}3^{\star}$; $n{:}\{\text{all } 512 \text{ or } 256\}^{\dagger}$

$^{\imath}$*convolution layers*; ‡*pooling layers*; **hidden layers*; †*hidden nodes*

Execution Time *(msec.)*

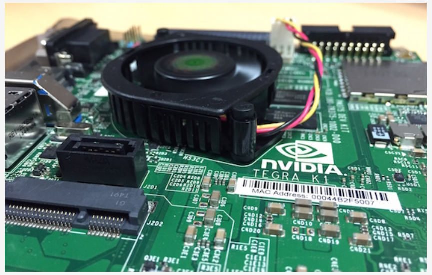
	Tegra		Snapdragon		Edison
	CPU	GPU	CPU	DSP	CPU
Deep KWS	0.8	1.1	7.1	7.0	63.1
DeepEar	6.7	3.2	71.2	379.2	109.0
AlexNet	600.2	49.1	159,383.1	-	283,038.6
SVHN	15.1	2.8	1,616.5	-	3,562.3



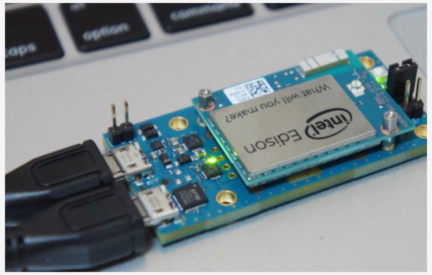
Target Platforms

Snapdragon 800

Nvidia Tegra K1

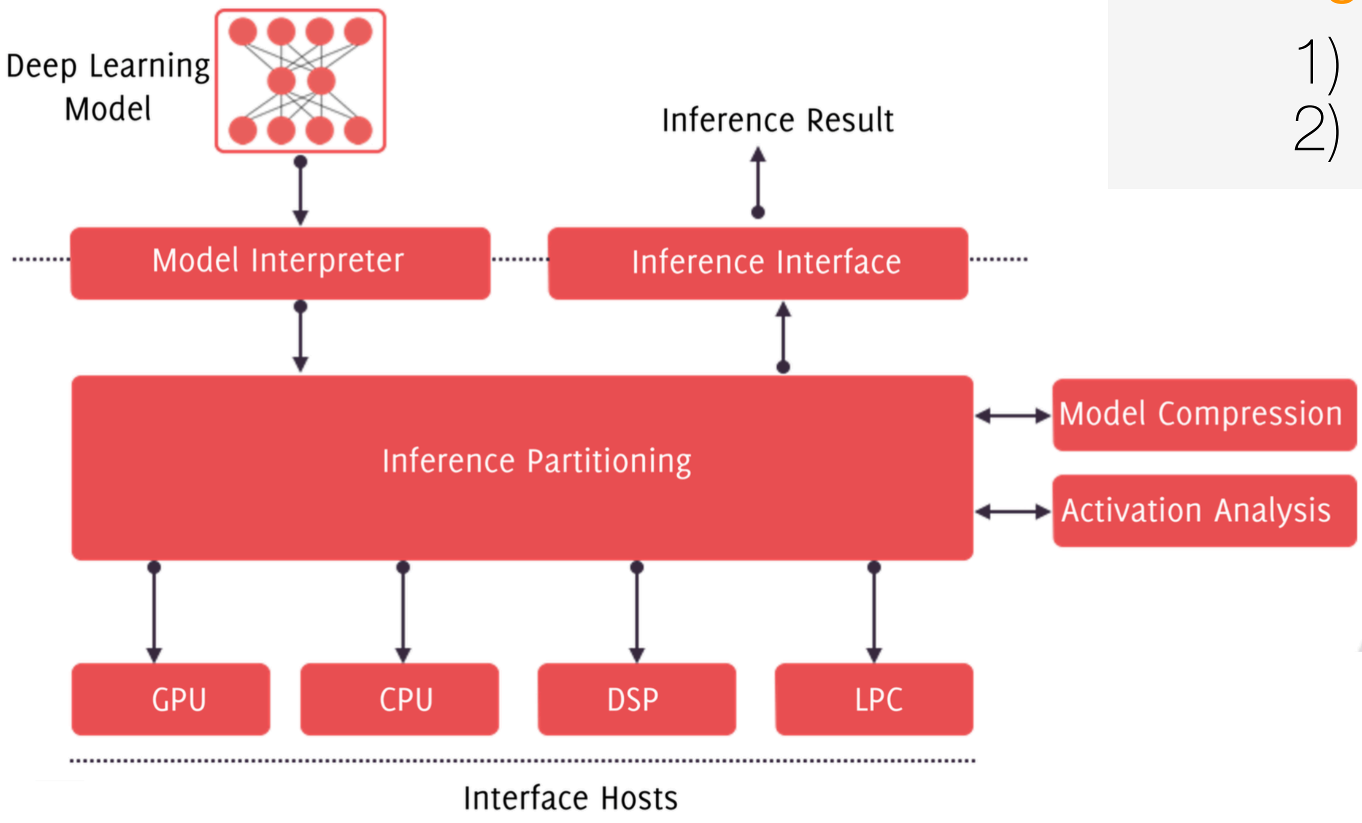


Intel Edison

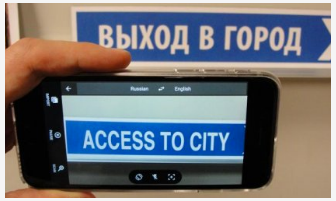


"An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-of-Things Devices",
Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Fahim Kawsar – *IoT-App 2015*

DeepX: Enabling Efficient Deep Learning Inference for Wearables, Smartphones and IoT Devices



Current Wearable/IoT Deep Learning Solutions



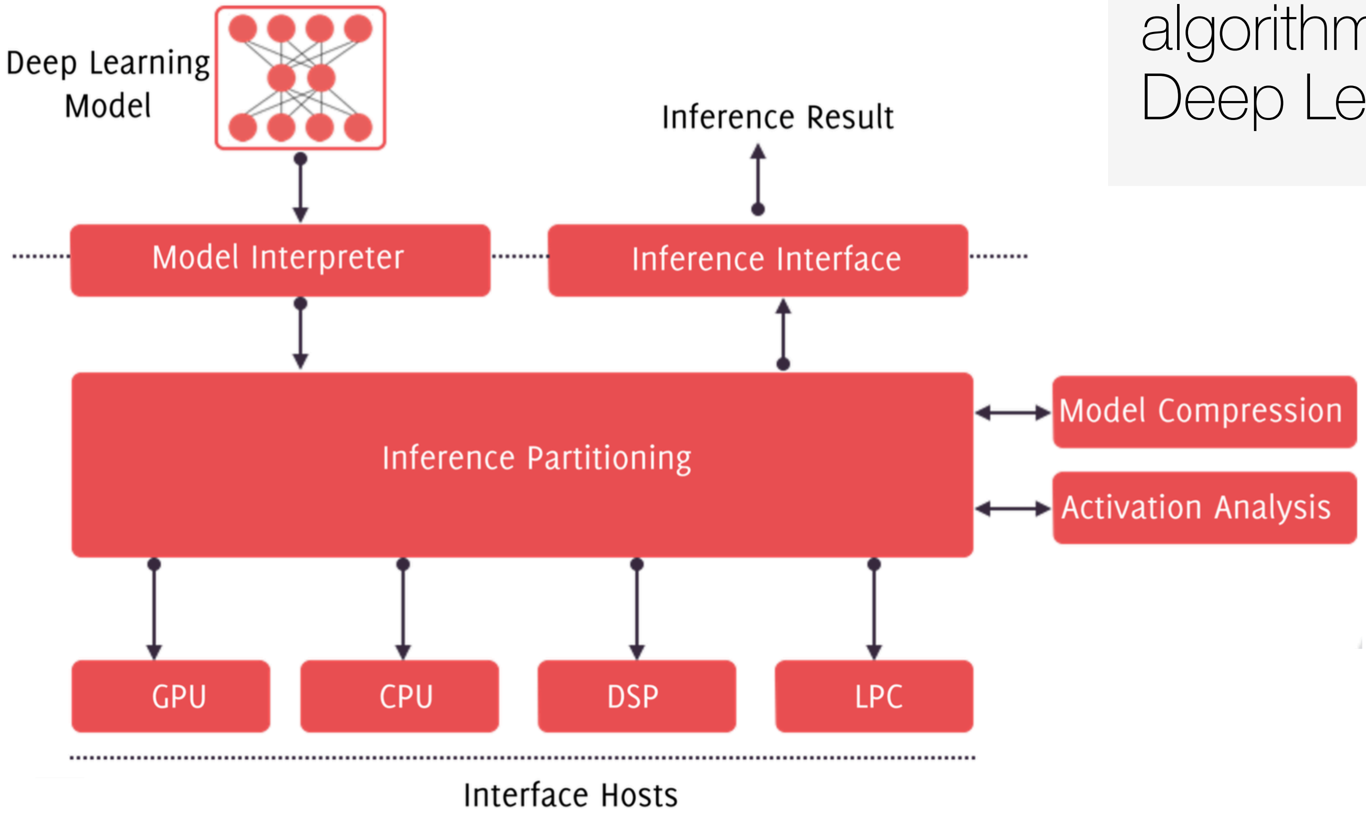
1) Cloud-bound
2) Task-specific Hand-optimized Models

DeepX Advantages

- Enhanced Privacy with increased on-device execution
- Seamless optimal network assistance
- Allow use of the state-of-the-art modeling algorithms

"DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices",
Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lorena Qendro, Fahim Kawsar – *IPSN 2016*

DeepX: Enabling Efficient Deep Learning Inference for Wearables, Smartphones and IoT Devices

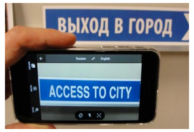


GOAL

Develop general purpose resource control algorithms for optimizing the inference stage of all Deep Learning models

DeepX Techniques

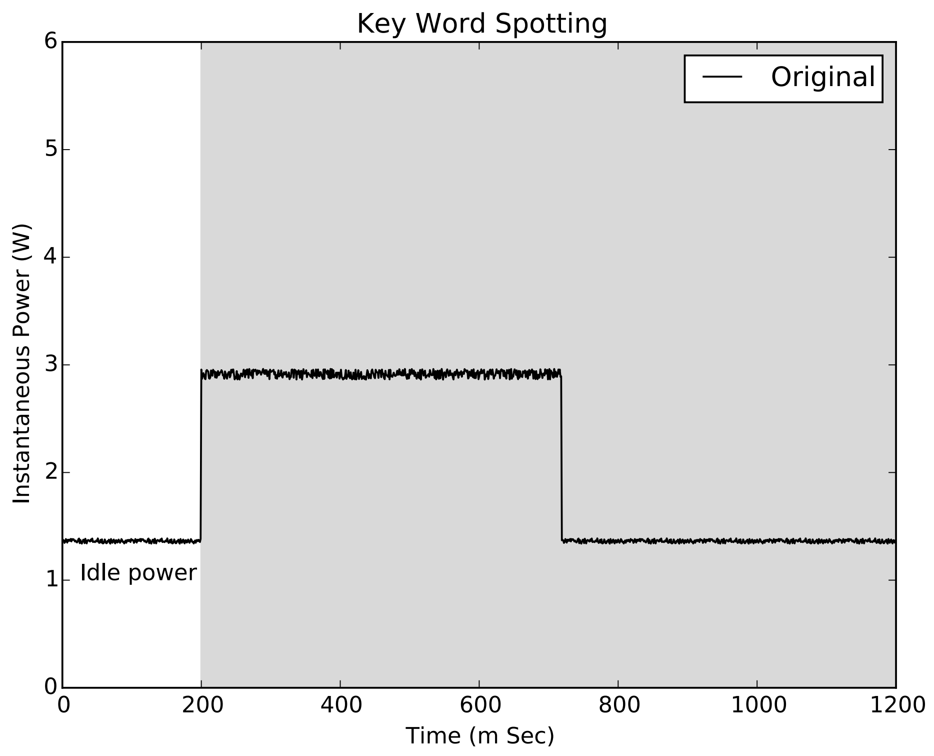
- Online Model Compression
- Activation / Static Analysis
- Redundancy Identification
- Model Partitioning
- …



Small Cells
Apps
Mobile OSs

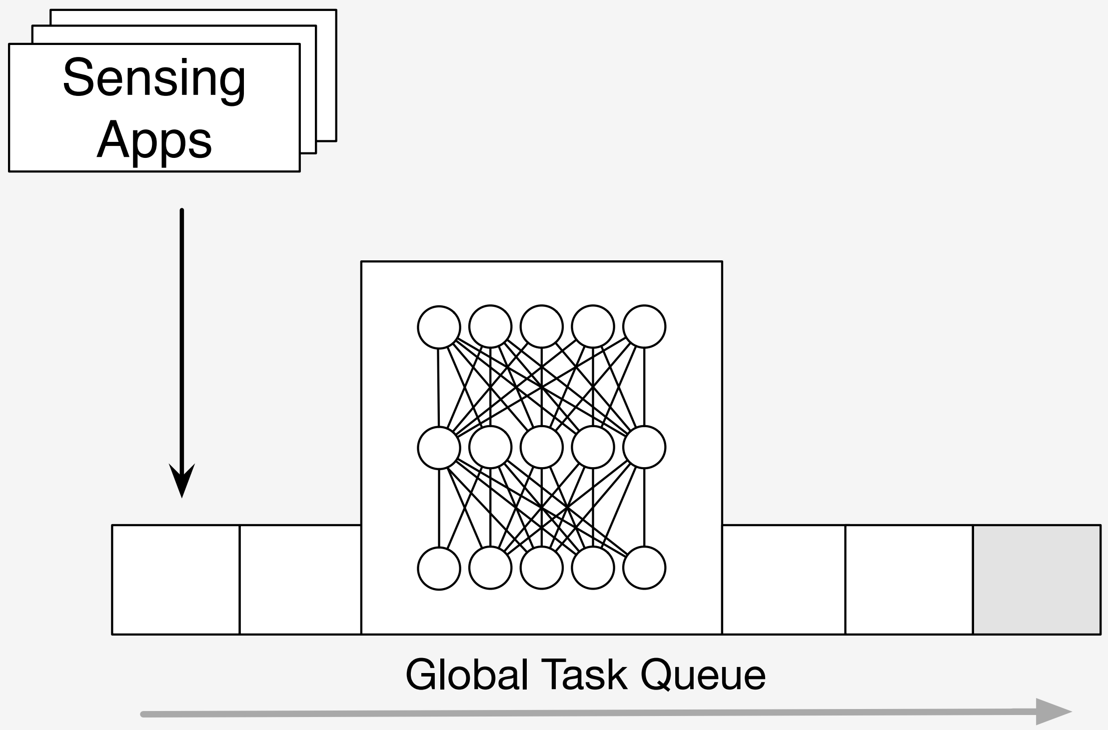
"DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices",
Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lorena Qendro, Fahim Kawsar – *IPSN 2016*

Model Compression



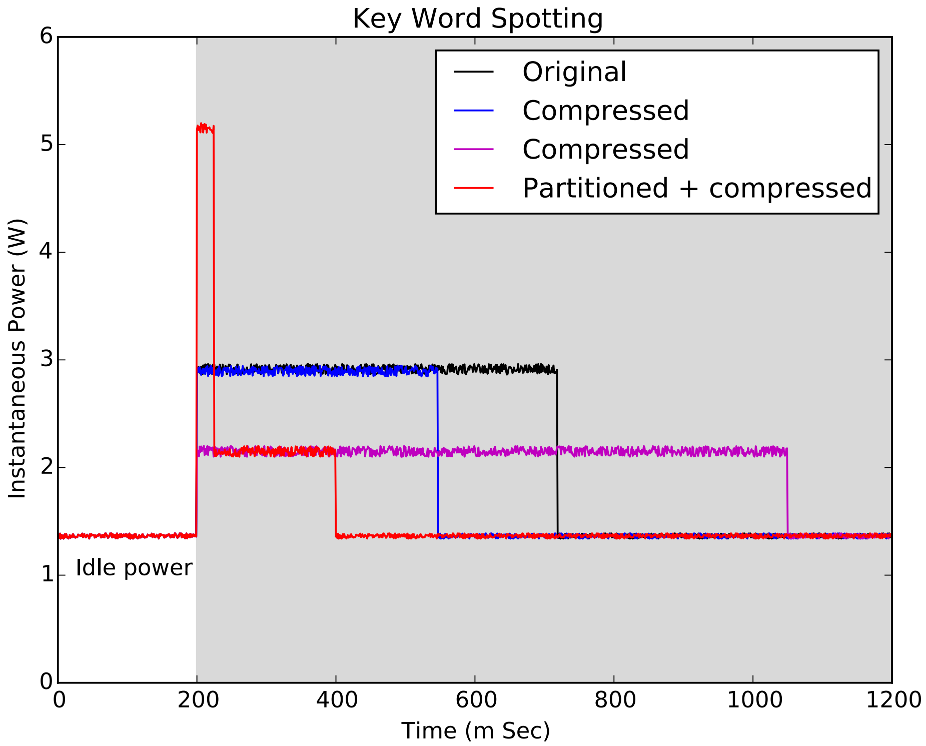
Goal: Graceful resource control through accuracy trade-off

Model Partitioning



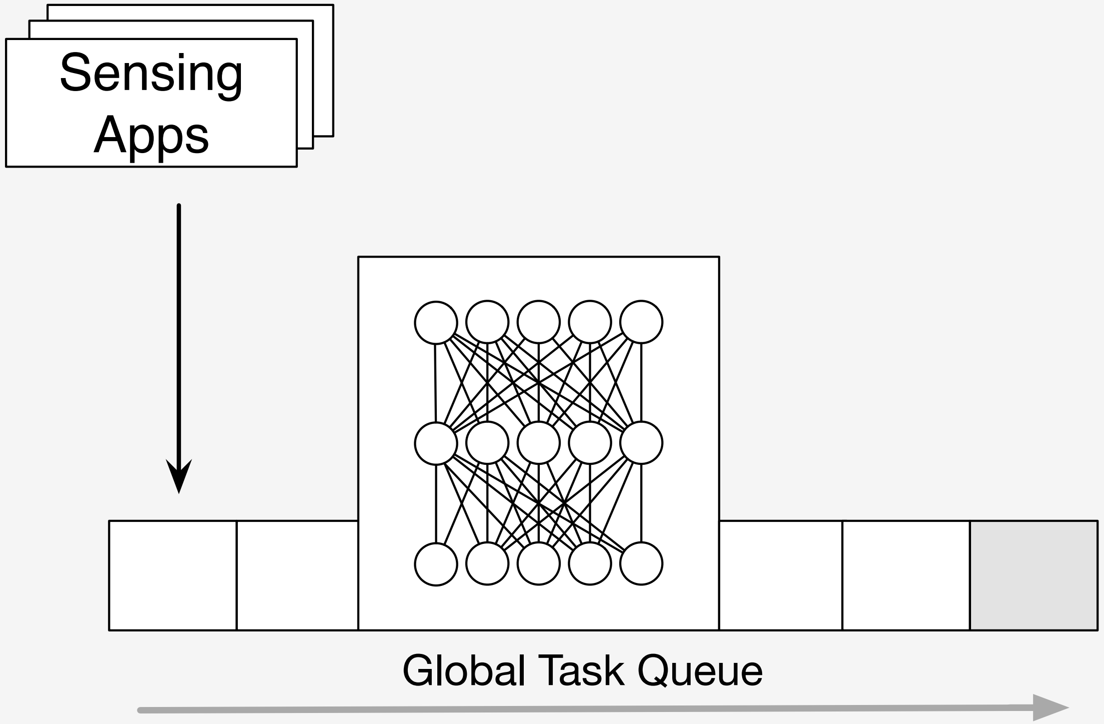
Goal: Reduce bottlenecks and increase resource utilization

Model Compression



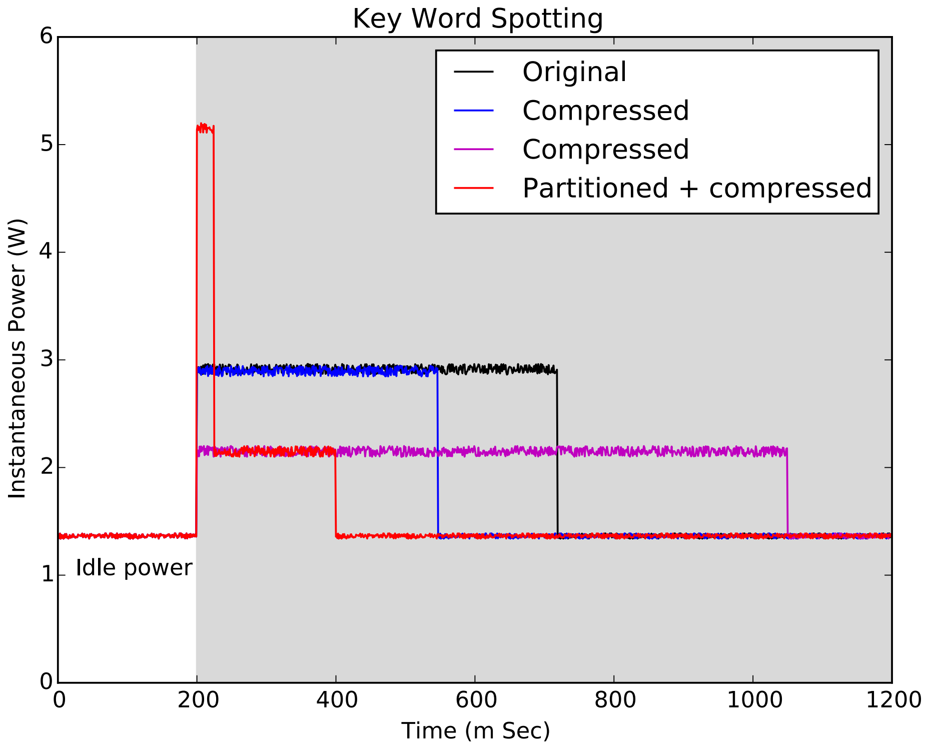
Goal: Graceful resource control through accuracy trade-off

Model Partitioning



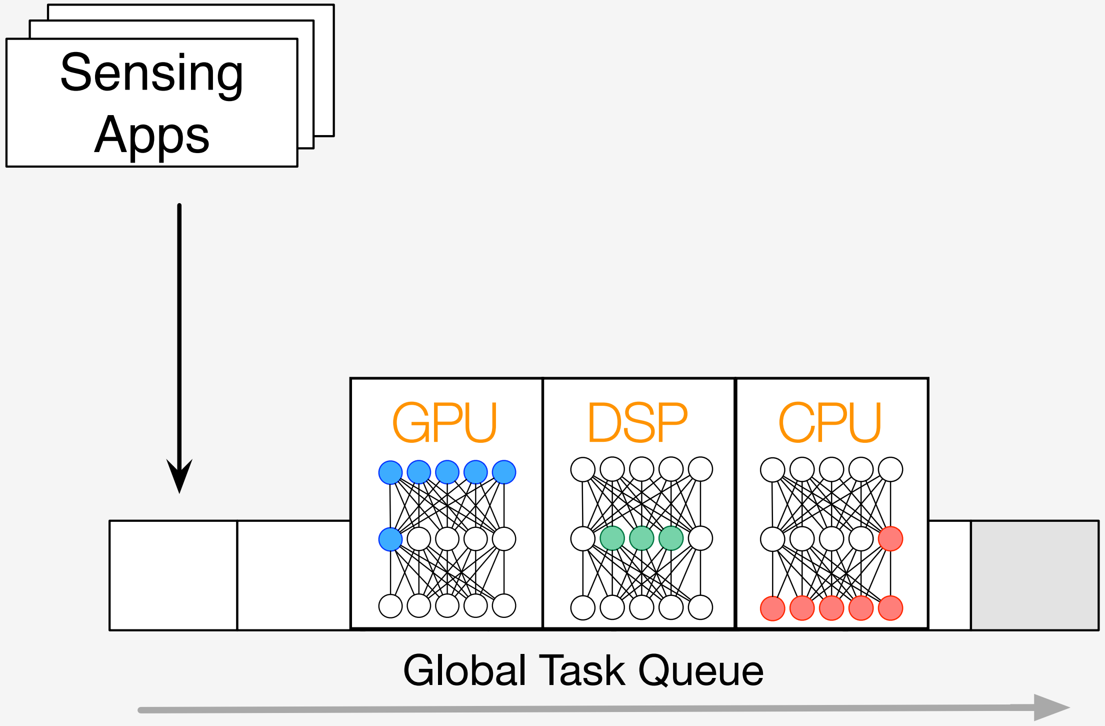
Goal: Reduce bottlenecks and increase resource utilization

Model Compression



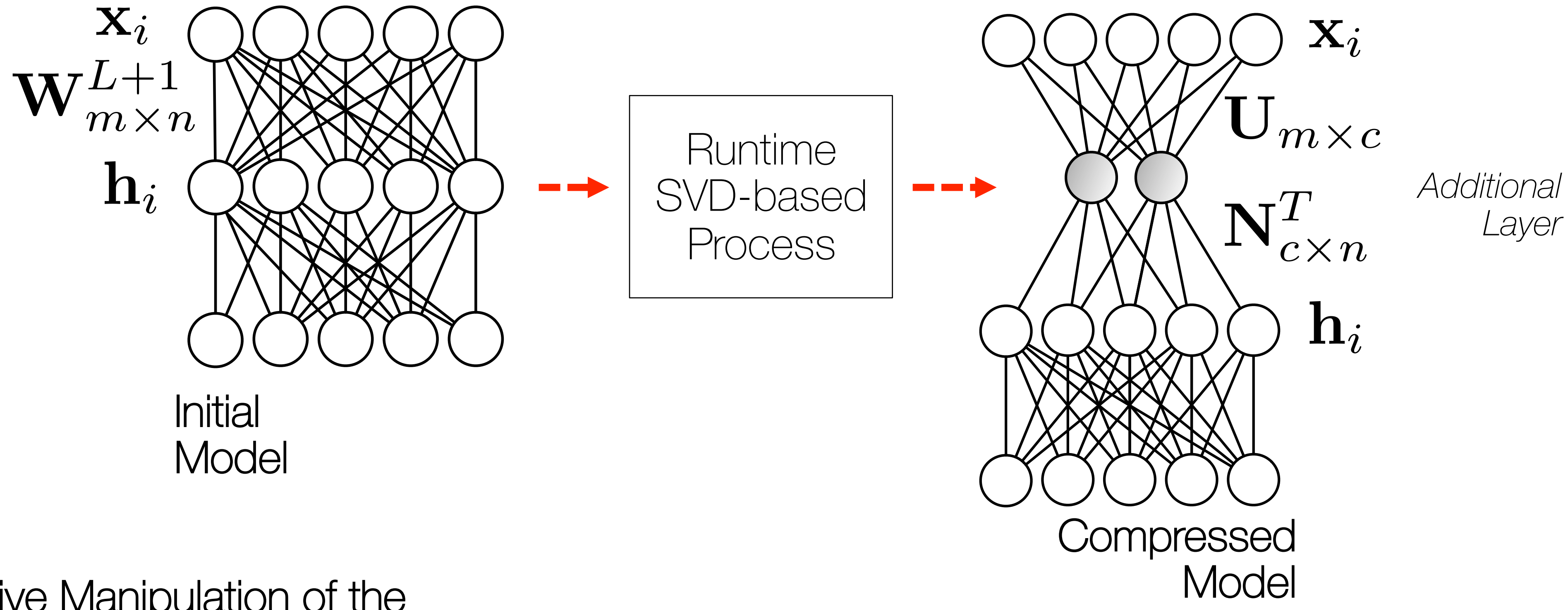
Goal: Graceful resource control through accuracy trade-off

Model Partitioning



Goal: Reduce bottlenecks and increase resource utilization

Example Model Compression Technique



Representative Manipulation of the Weight Matrix

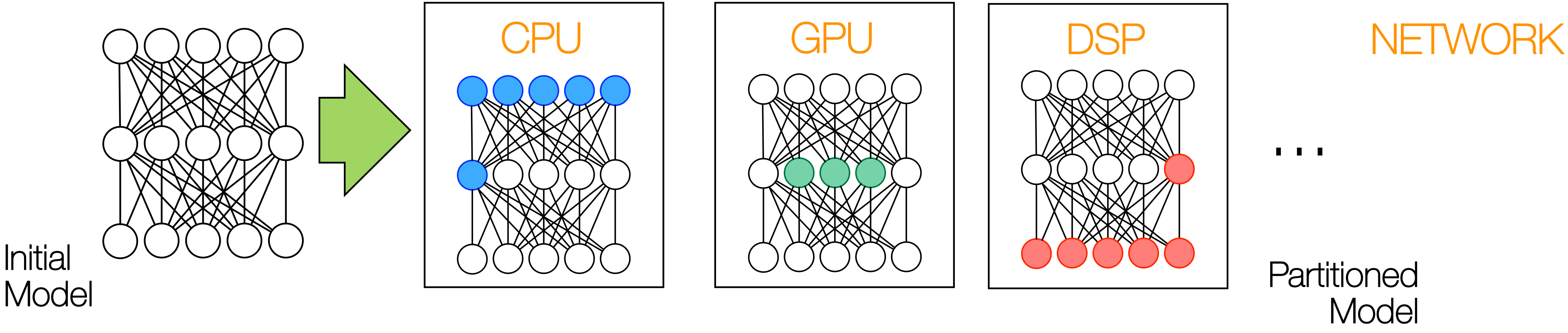
$$W^{L+1}_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^{T}_{n\times n}$$

$$\hat{W}^{L+1}_{m\times n} = U_{m\times c}\Sigma_{c\times c}V^{T}_{c\times n}$$

$$\hat{W}^{L+1}_{m\times n} = U_{m\times c}N^{T}_{c\times n}$$

- Applicable at runtime (SVD approach)
- Without retraining model or have local test data
- Inspired by existing SVD-methods
  - *[Xue et al. '13, He et al. '14]*
- Redundancy Estimation $\mathscr{E}(W^{L+1}_{m\times n}, \hat{W}^{L+1}_{m\times n}) = \sqrt{\frac{\Sigma_{i=1}^{m}(w_i - \hat{w}_i)^2}{m}},$

Example Model Partitioning Process



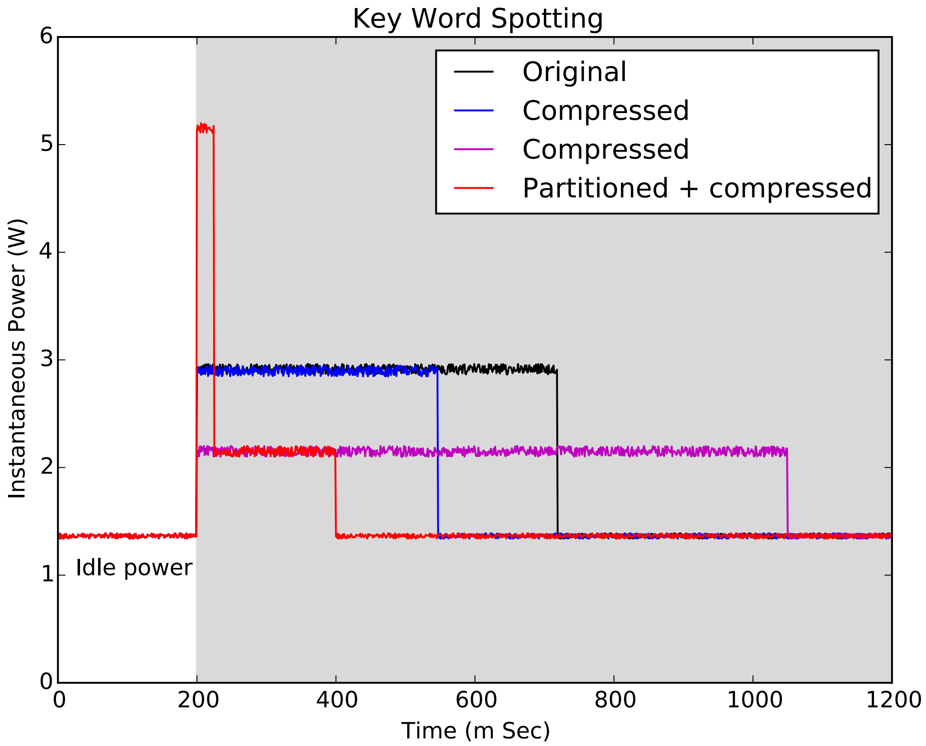
Partition Stages: Load Estimation & Monitoring → Partition Assignment → Results Collation

Simplified Optimization

$$
\begin{aligned}
\text{min.} \quad & \alpha \sum_{i=1}^{P} E_i B_i + \beta \max_{i \in \mathcal{P}} \{T_i B_i\} \\
\text{s.t.} \quad & \sum_{i=1}^{P} B_i = N \\
& B_i \leq L_i, \forall i \in \mathcal{P}, \\
& B_i \geq 0, B_i \in \mathcal{Z}, \forall i \in \mathcal{P},
\end{aligned}
$$

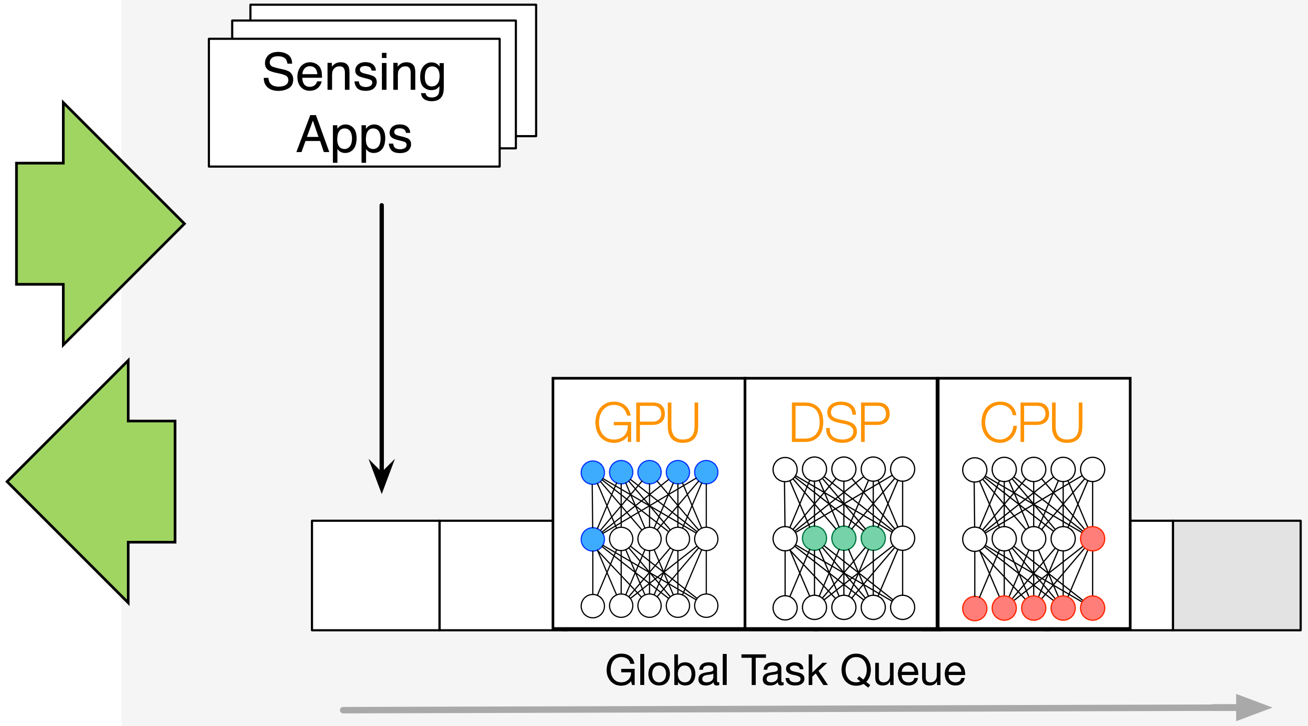
- Reduced Dependencies using Static Analysis
- Projections/Reductions for Novel Partitioning
- Processor assignment based on efficiency & load
- Tuning of compression strength

Model Compression



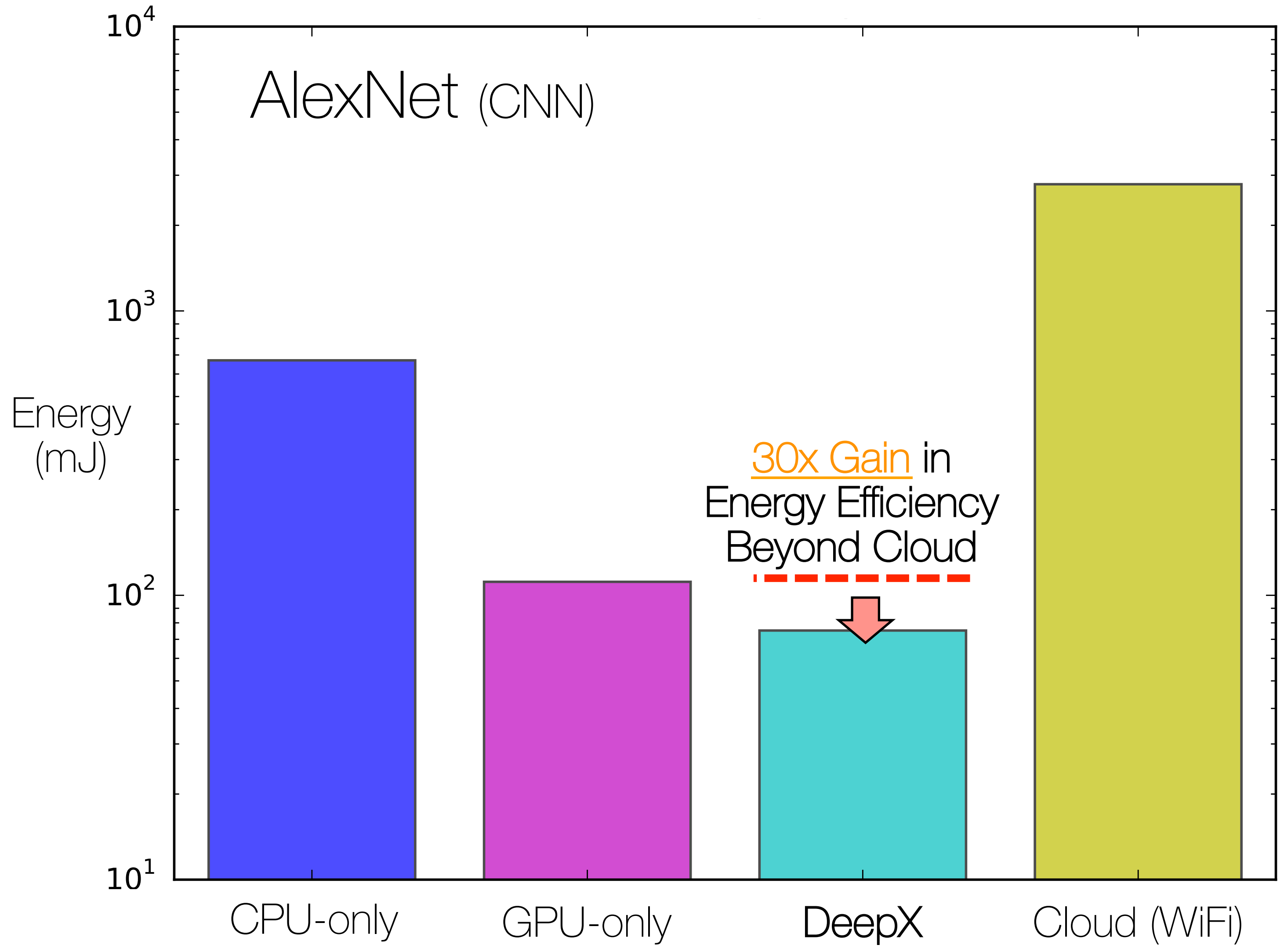
Goal: Graceful resource control through accuracy trade-off

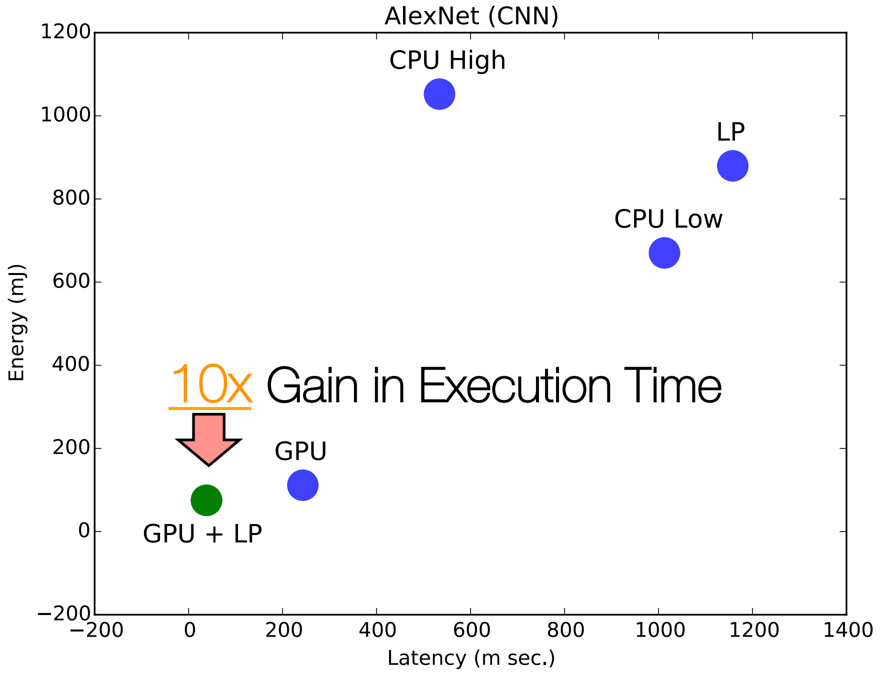
Model Partitioning



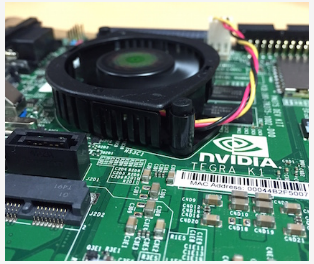
Goal: Reduce bottlenecks and increase resource utilization

Efficient Mobile Execution of Large-scale Deep Learning Models



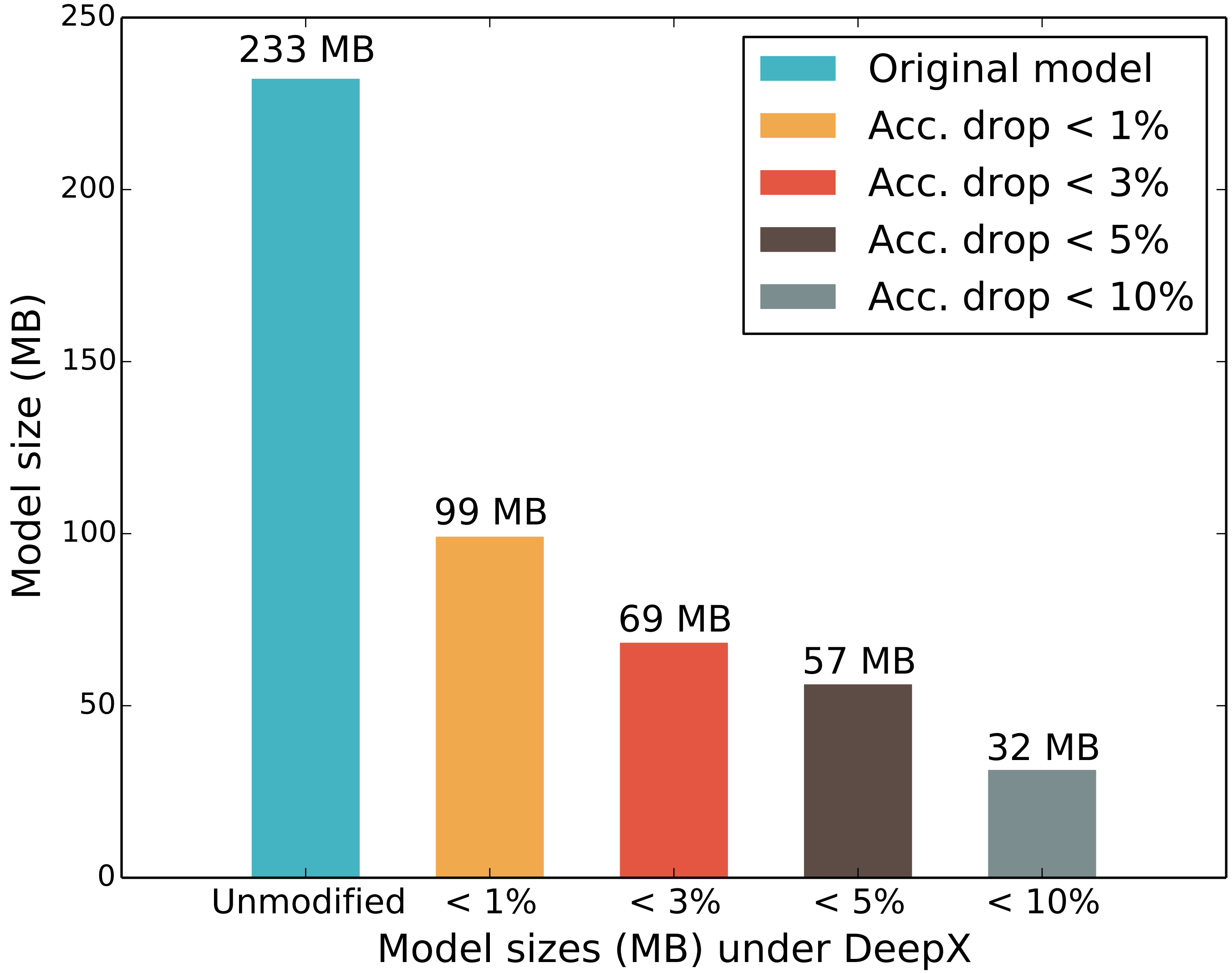


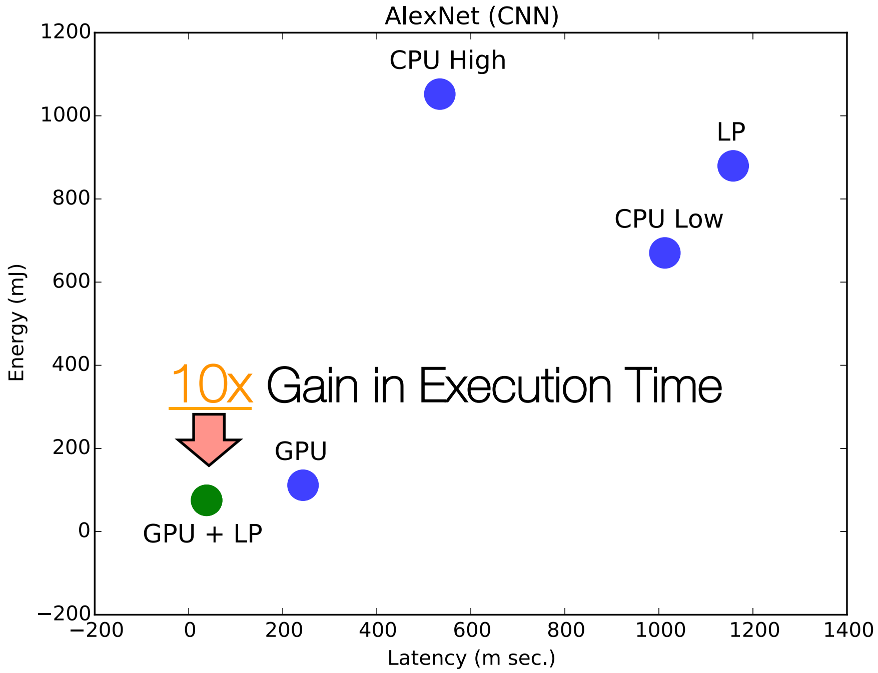
Prototype Platform



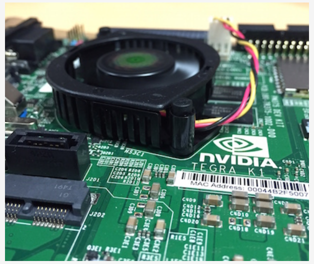
Nvidia Tegra K1 SoC

Efficient Mobile Execution of Large-scale Deep Learning Models



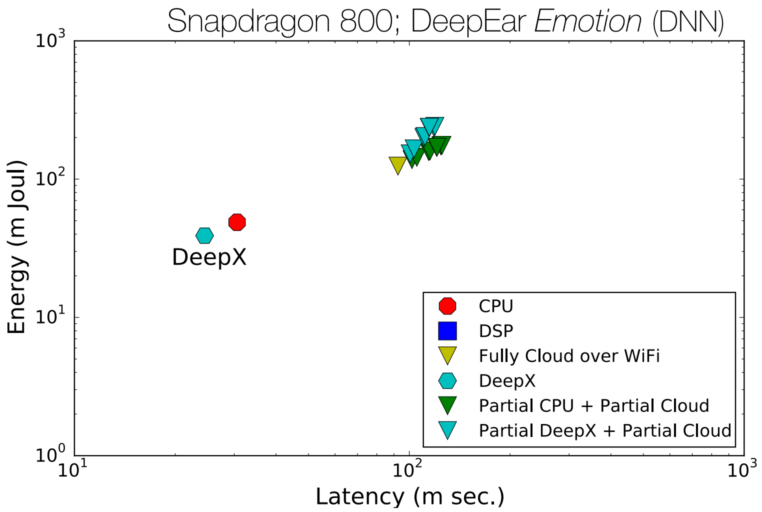


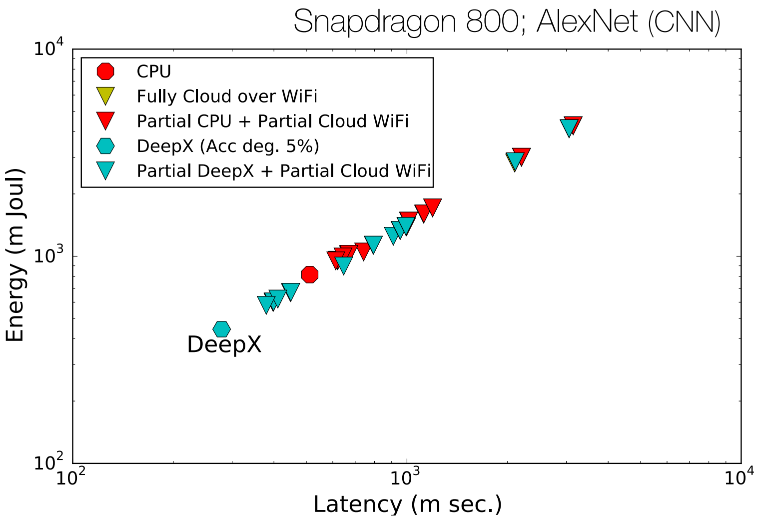
Prototype Platform

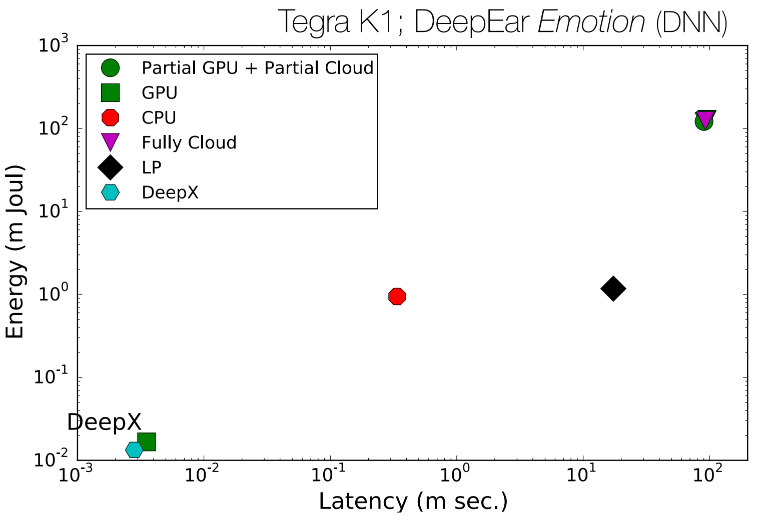


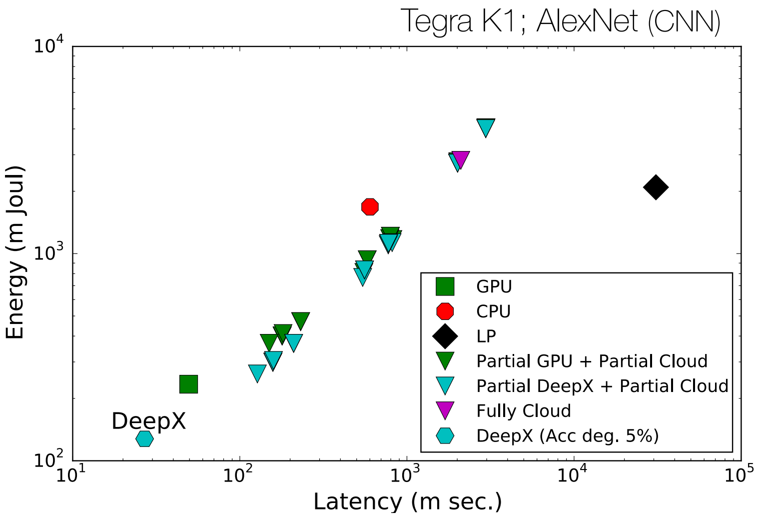
Nvidia Tegra K1 SoC

Efficient Mobile Execution of Large-scale Deep Learning Models





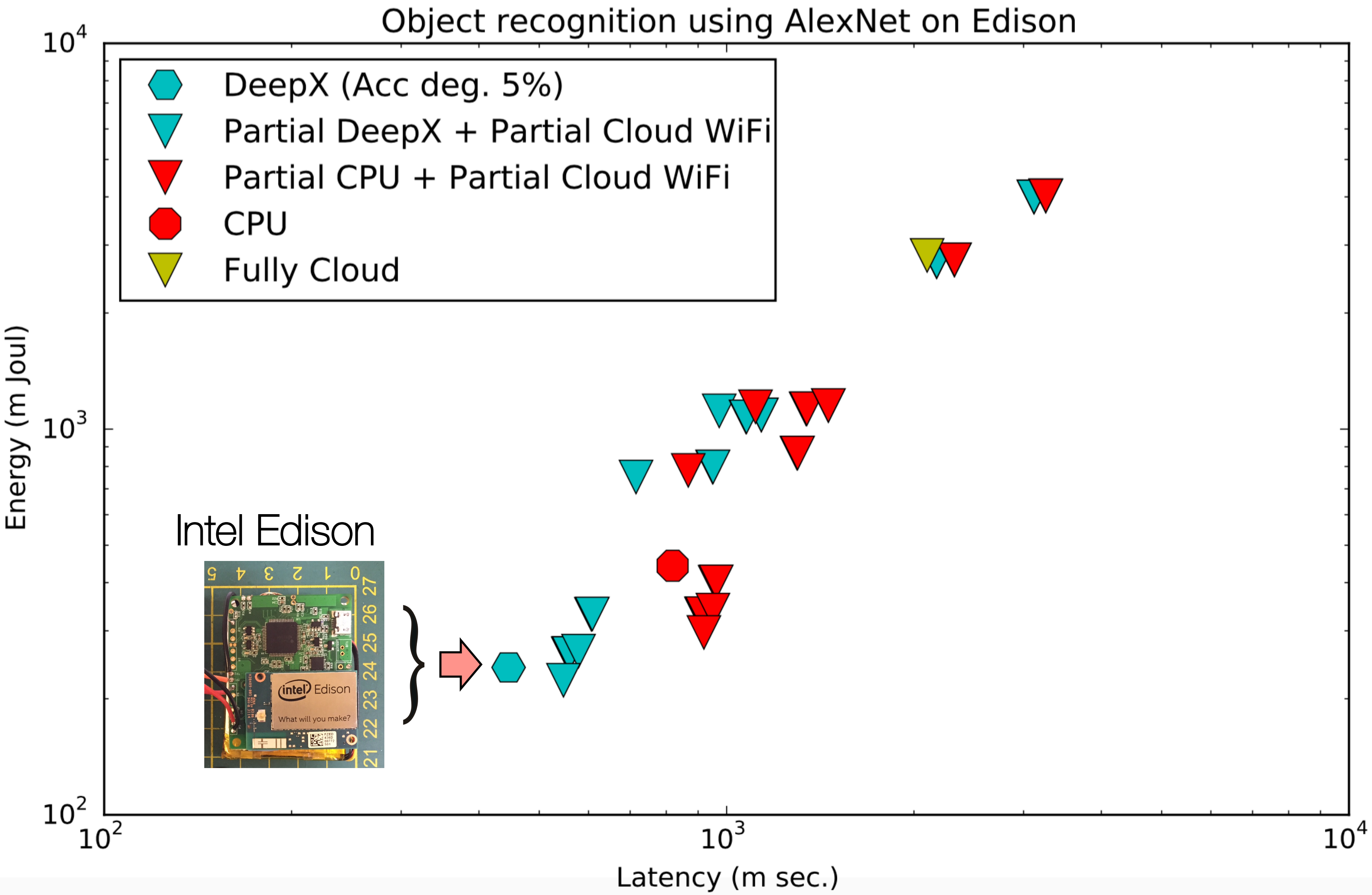




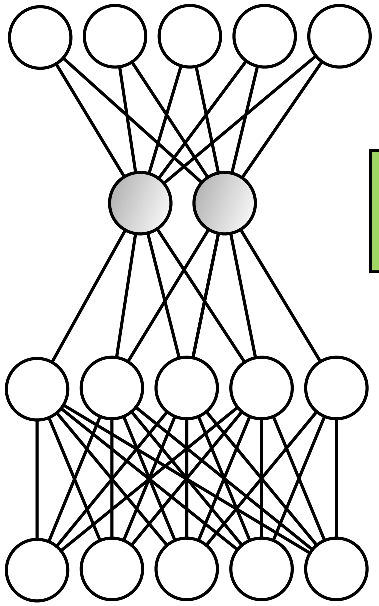
Results also hold for:

- Broad Sensor Modalities
- IoT and Wearable Platforms
- Various Deep Architectures
- Network Conditions

Latest DeepX Result: Complete framework running on Ultra-wearable Hardware

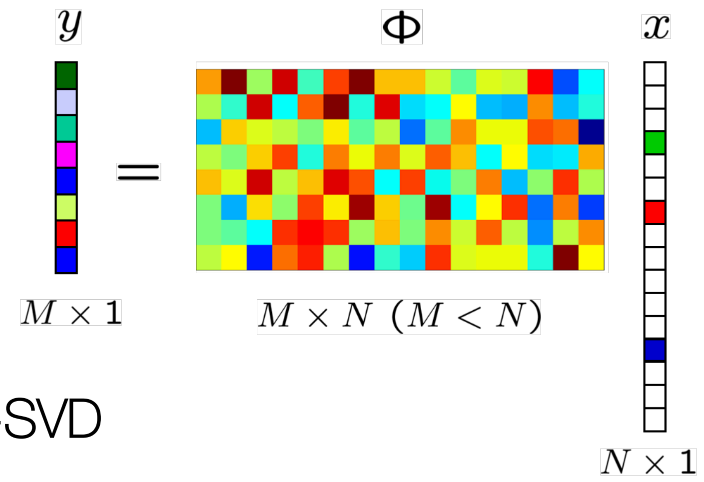


Enabling Breakthrough: Sparsification of Layers for Extreme Compression Required by Ultra-Wearables



Compressed Model

Approach: Use Compressive Sensing Theory to Reduce Layer Representation



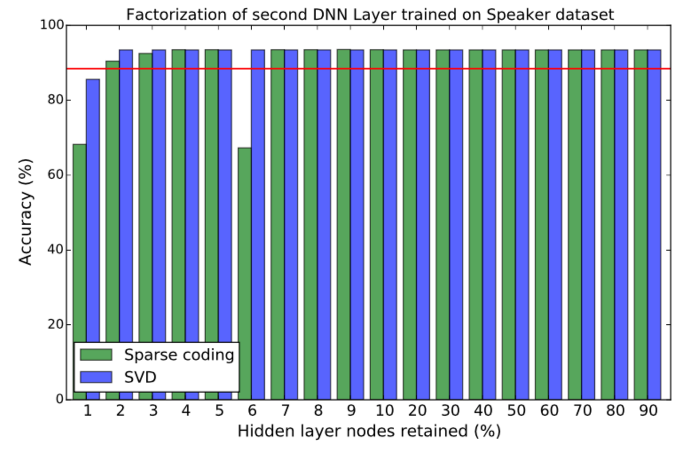
Usage of K-SVD Algorithm

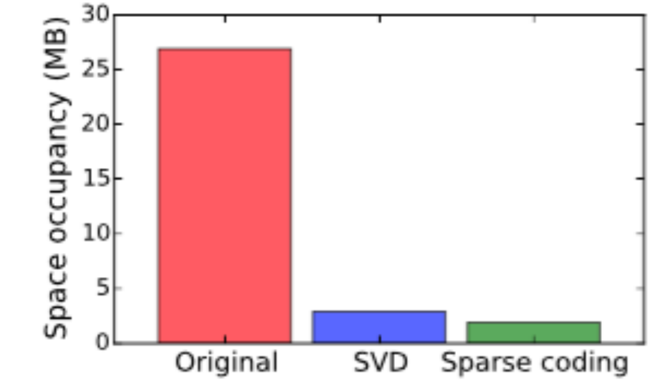
$$\min_{\boldsymbol{\mathcal{B}},\boldsymbol{A}} ||\boldsymbol{W}^L - \boldsymbol{\mathcal{B}} \cdot \boldsymbol{A}||_2^2 \quad s.t.\ \forall i\ ||\boldsymbol{a}^i||_0 \leq K$$

$$\boldsymbol{W}^L \approx \boldsymbol{\mathcal{B}} \cdot \boldsymbol{A}$$

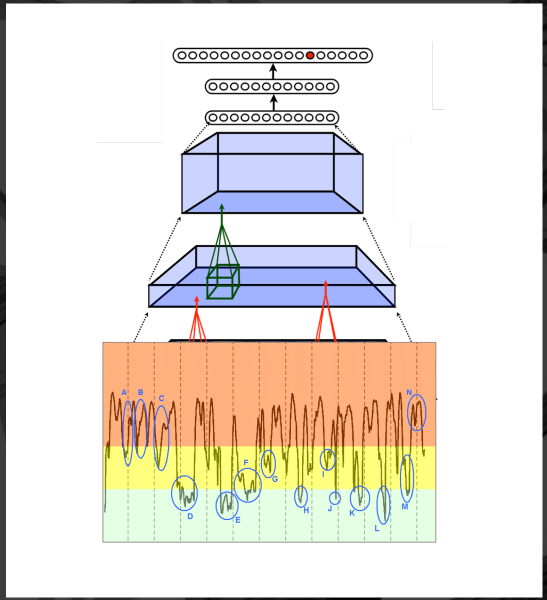
Motivation: SVD-based compression can not lower resource needs to match ultra-wearables without destroying accuracy

Representative Trade-offs

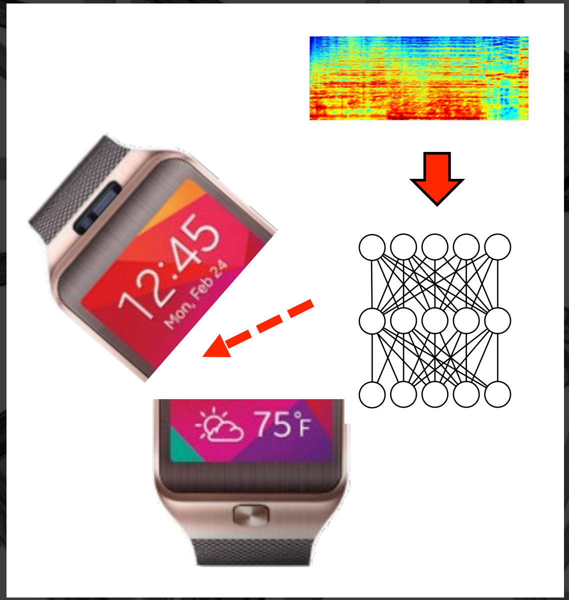




Narrowing the Sensor Inference Gap using Deep Learning on Wearable and IoT Devices

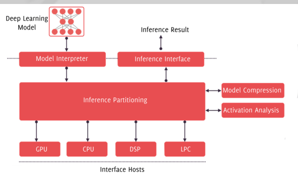


New Deep Modeling Methods for Wearable/ Mobile Sensors

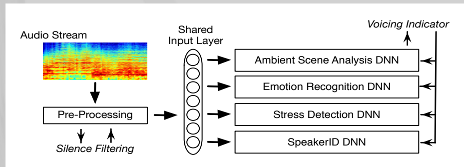


Missing Support for Scarce Resource Deep Model Execution

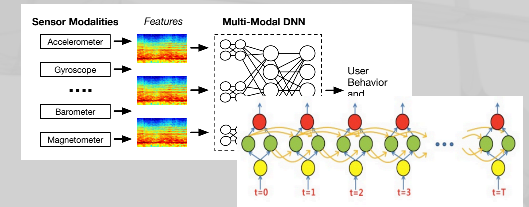
DeepX



DeepEar



. . .



MIT Technology Review

NewScientist

UBICOMP 2015 Best Paper, Top 1%

Thanks!

Questions?

Nicholas D. Lane
Sourav Bhattacharya
Claudio Forlivesi
Fahim Kawsar

Further Reading

"Can Deep Learning Revolutionize Mobile Sensing?", Nicholas D. Lane, Petko Georgiev – *HotMobile 2015*

"DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments using Deep Learning", Nicholas D. Lane, Petko Georgiev, Lorena Qendro – *UbiComp 2015*

"An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-of-Things Devices", Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Fahim Kawsar – *IoT-App 2015*

"DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices", Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lorena Qendro, Fahim Kawsar – *IPSN 2016*

"From Smart to Deep: Robust Activity Recognition on Smartwatches using Deep Learning", Sourav Bhattacharya, Nicholas D. Lane – *WristSense 2016*

Sourav Bhattacharya
sourav.bhattacharya@bell-labs.com

NOKIA Bell Labs