Why The Future of ML is Tiny and Bright

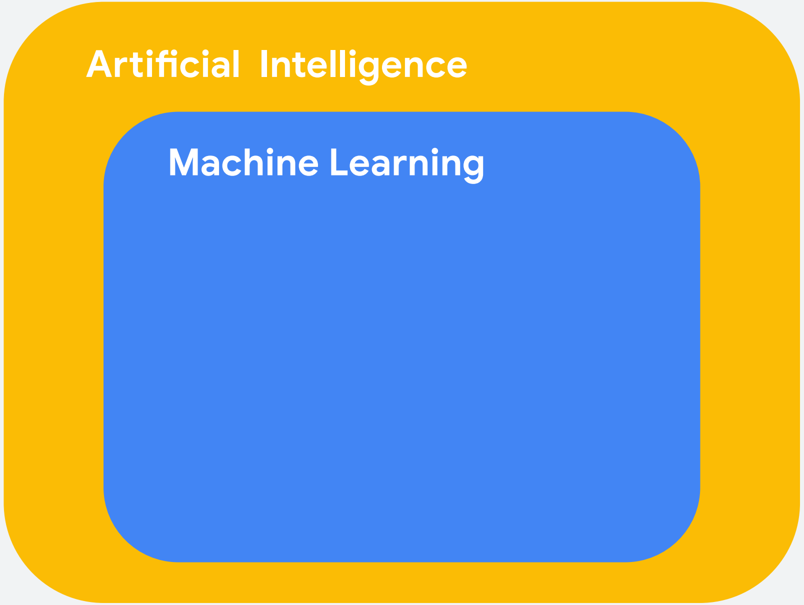
Challenges & Opportunities

Vijay Janapa Reddi, Ph. D. | Associate Professor |
John A. Paulson School of Engineering and Applied Sciences | Harvard University |
Web: http://scholar.harvard.edu/vijay-janapa-reddi

“Language”

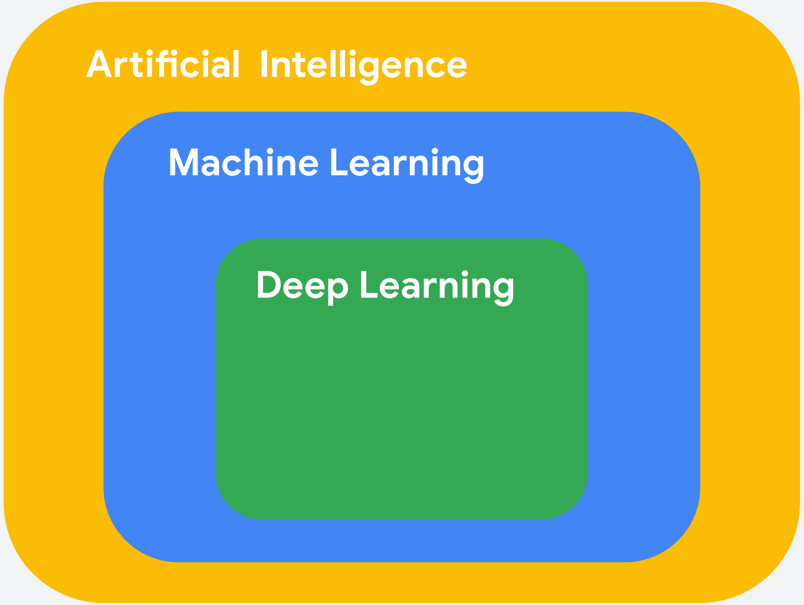
What is
Machine Learning?

1. **Machine Learning** is a subfield of **Artificial Intelligence** focused on developing algorithms that learn to **solve problems by analyzing data for patterns**



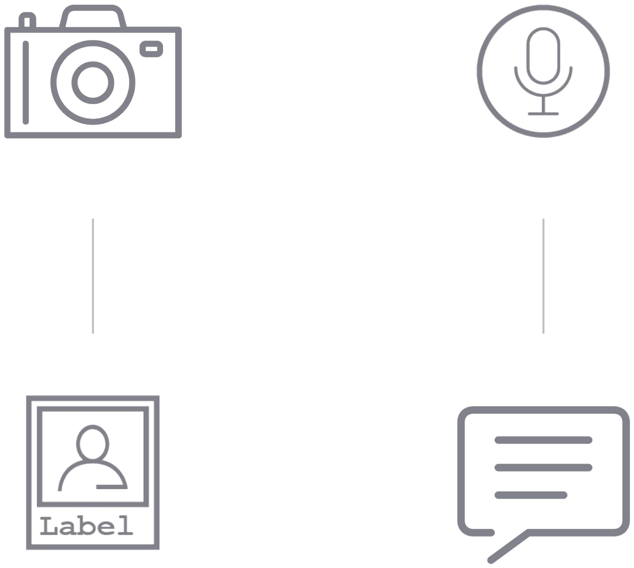
What is (Deep) Machine Learning?

1. Machine Learning is a subfield of Artificial Intelligence focused on developing algorithms that learn to solve problems by analyzing data for patterns
2. **Deep Learning** is a type of Machine Learning that leverages **Neural Networks** and **Big Data**



Applications of Machine Learning

Applications of Machine Learning



Applications of Machine Learning

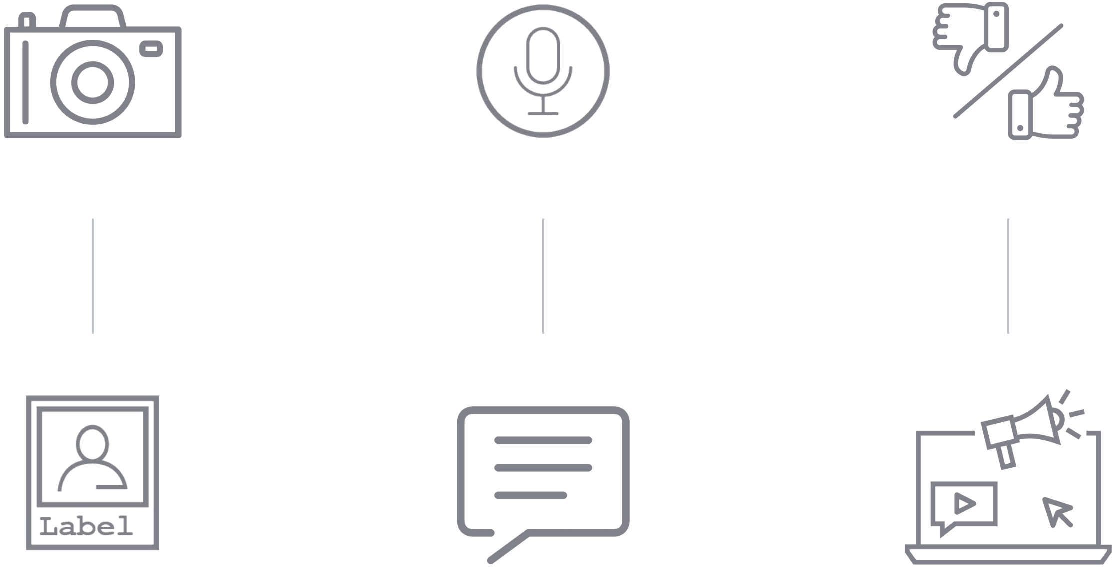
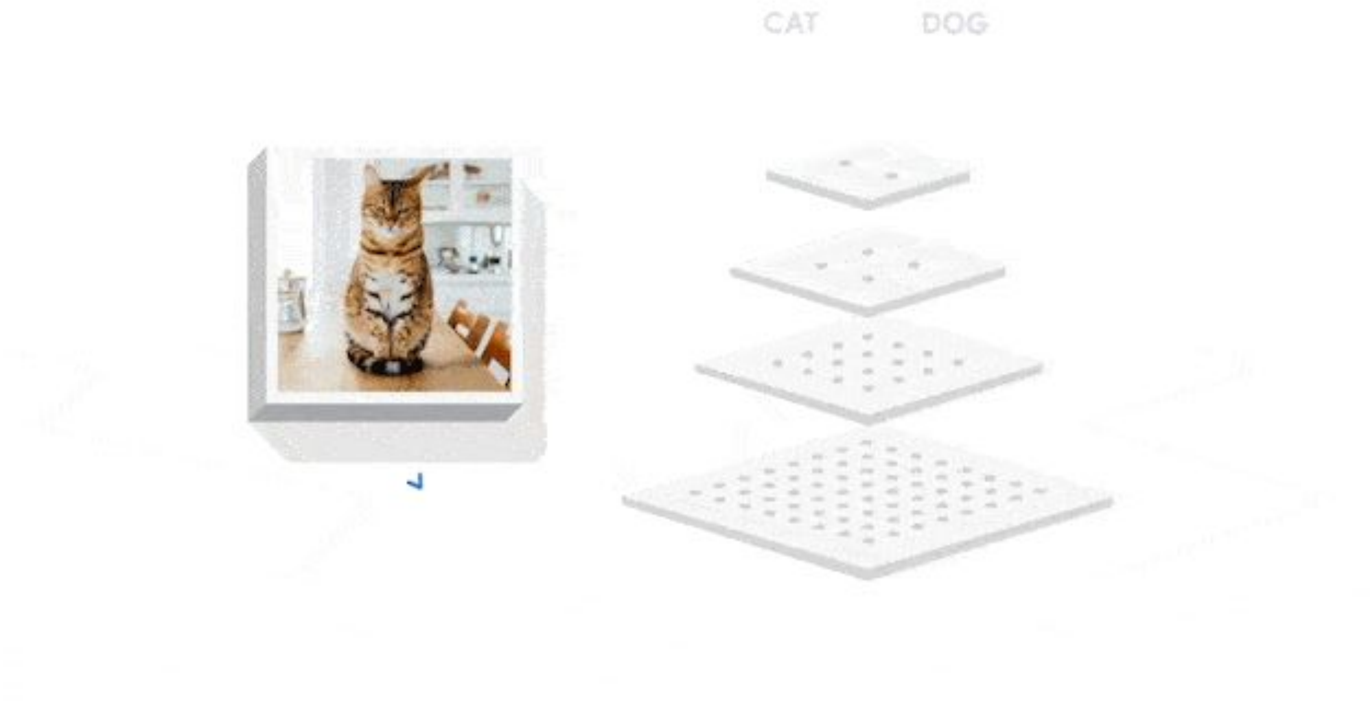
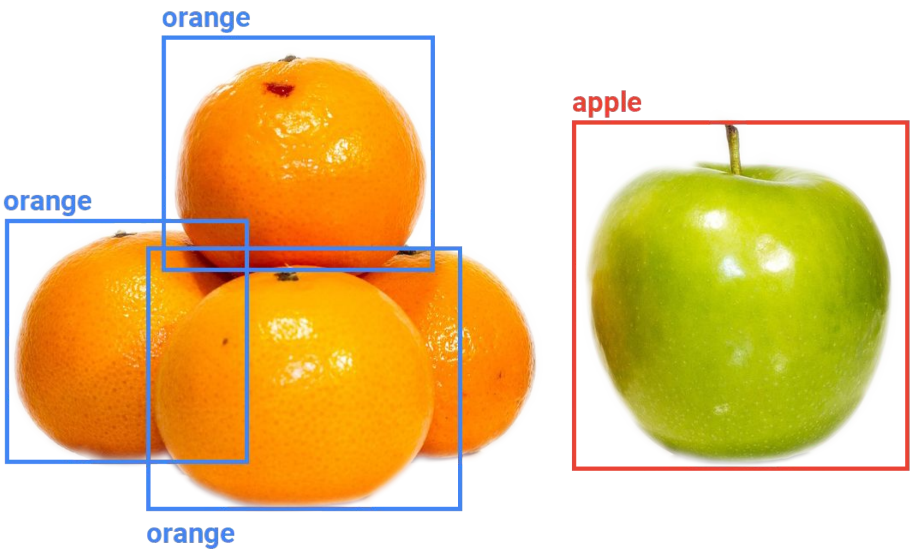


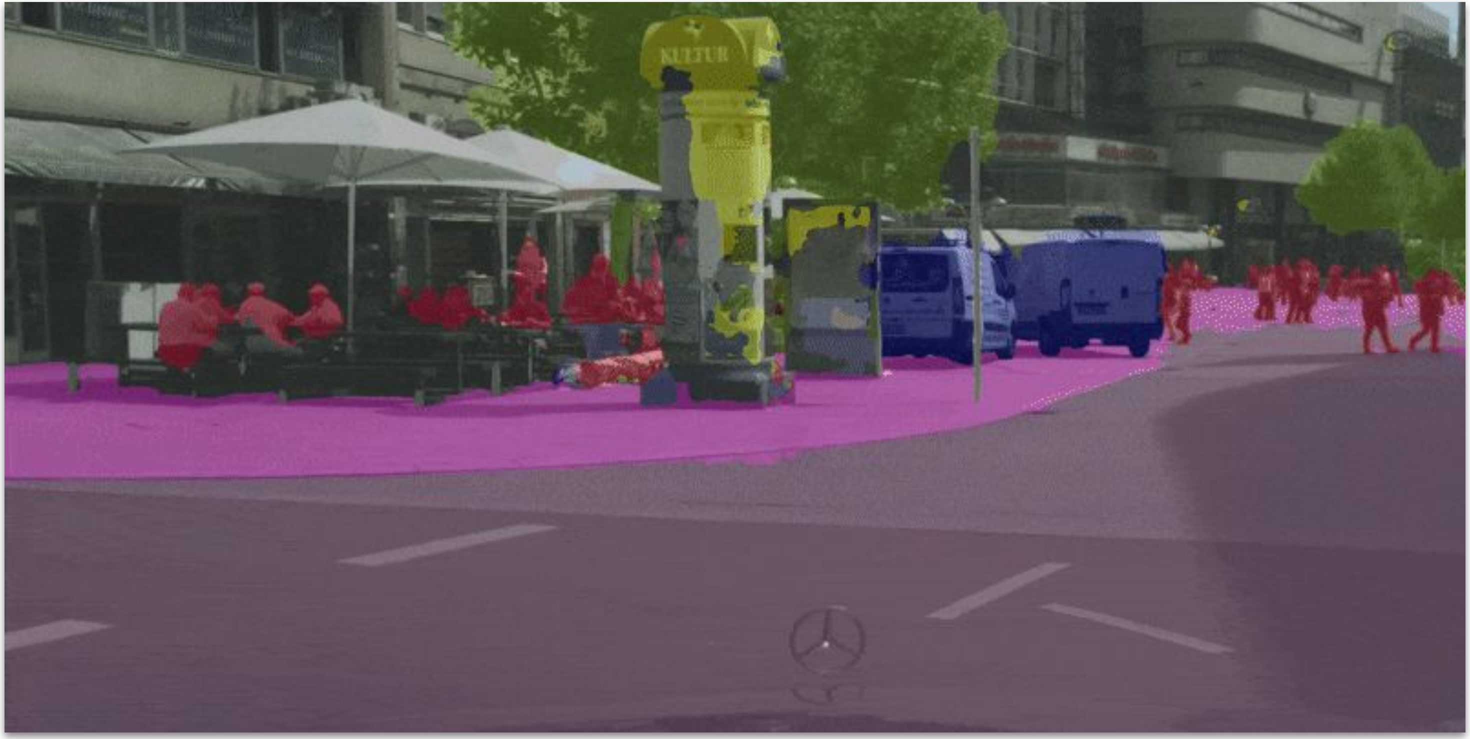
Image Classification



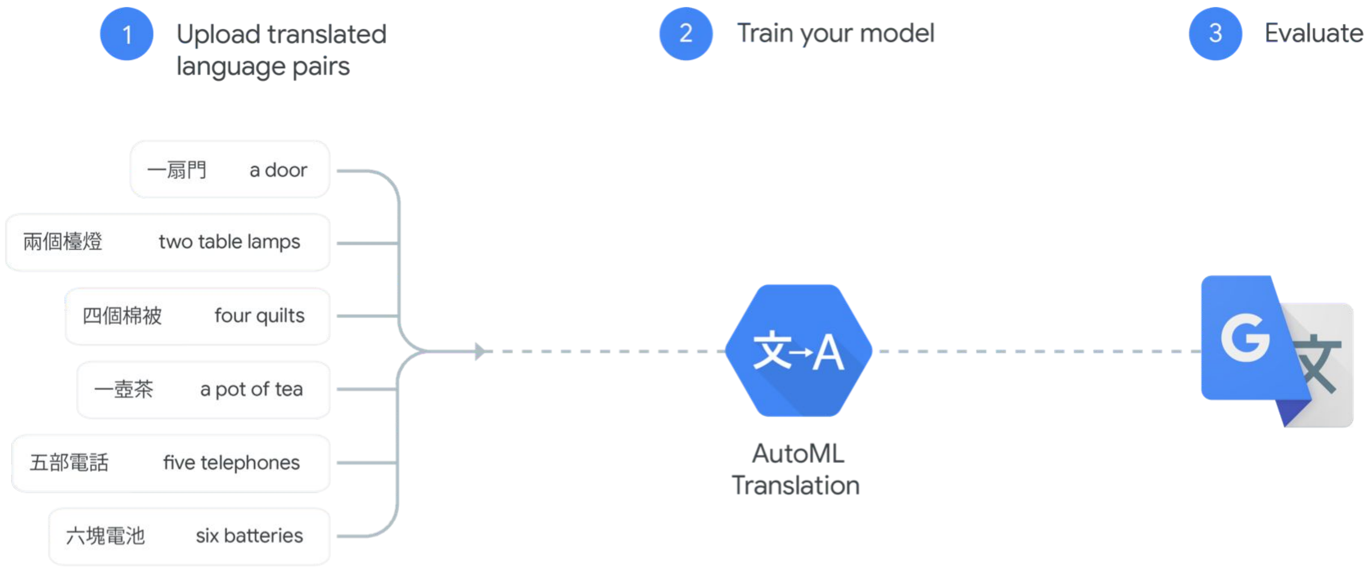
Object Detection



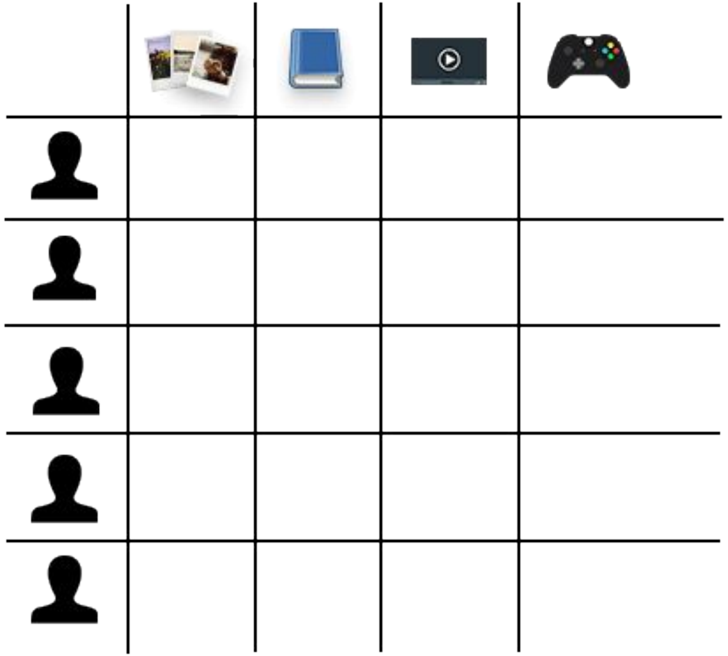
Segmentation



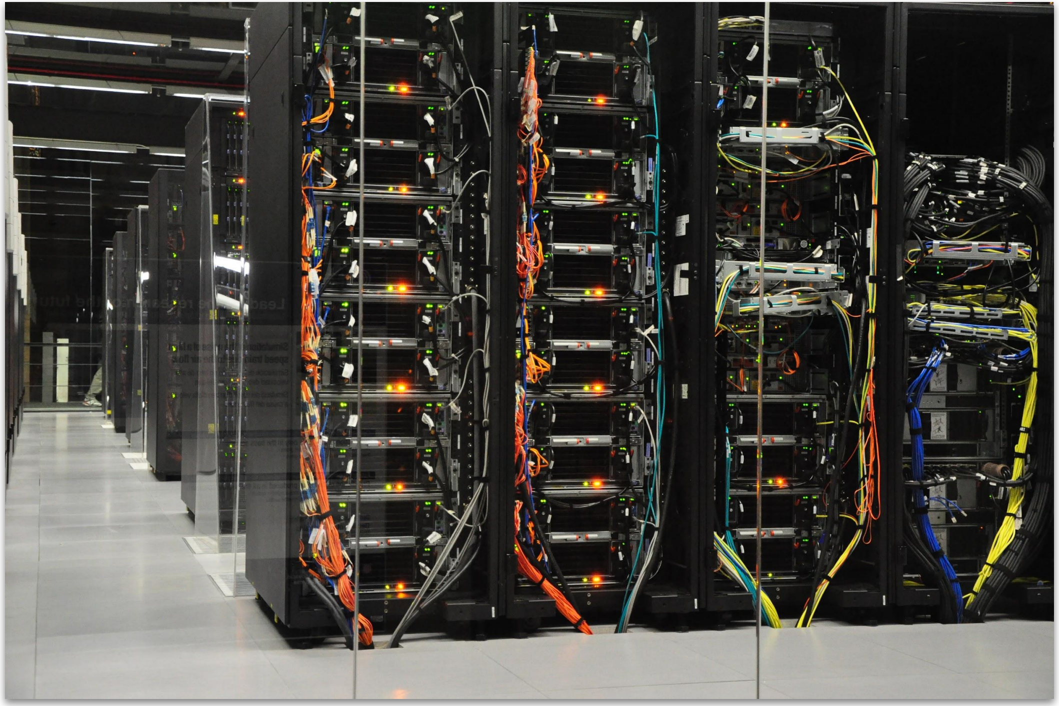
Machine Translation



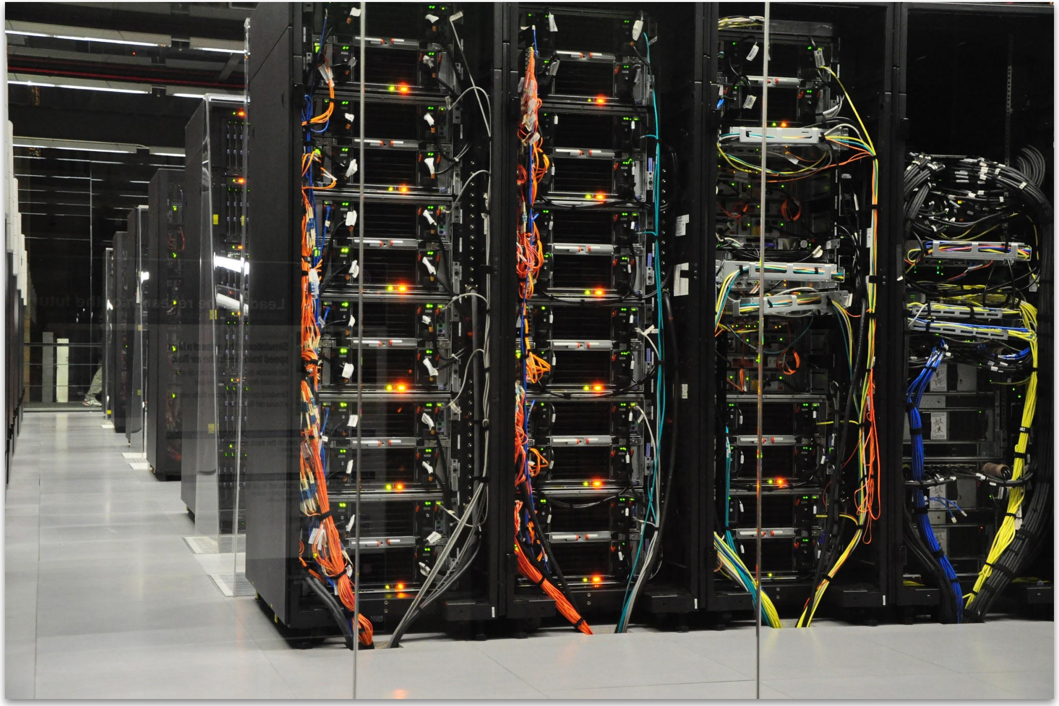
Recommendations



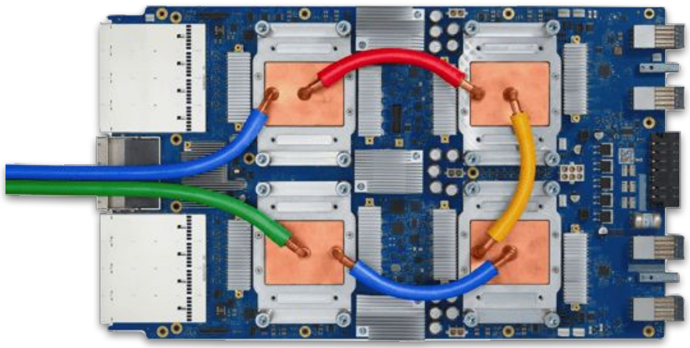
Datacenter

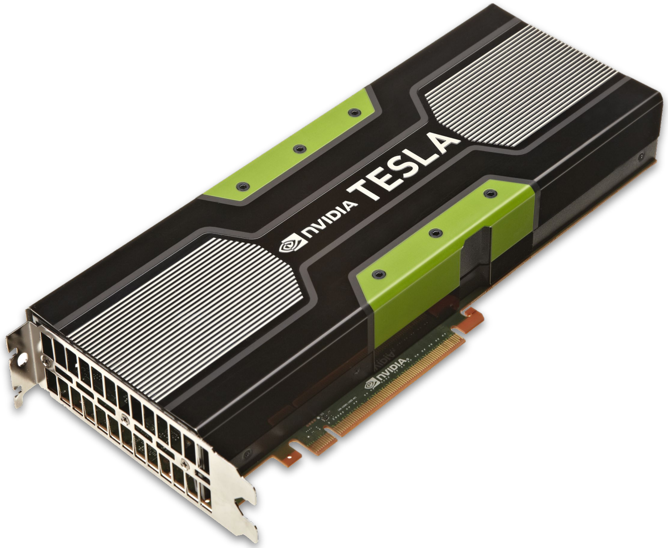


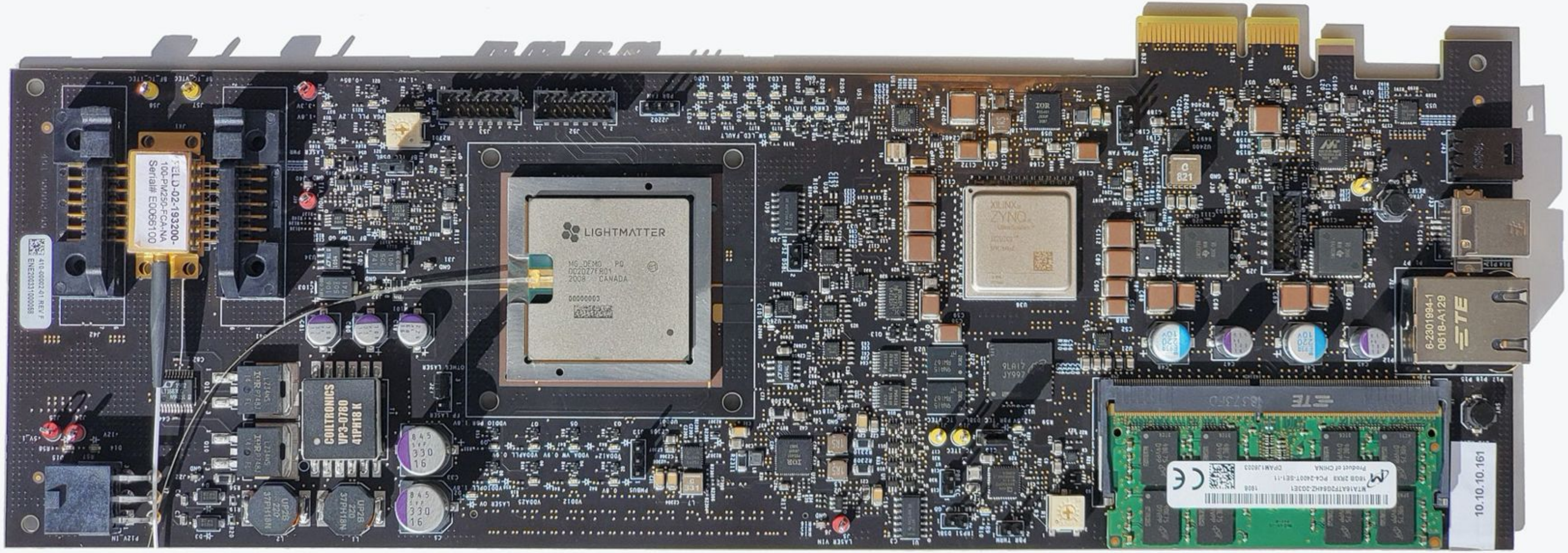
Datacenter



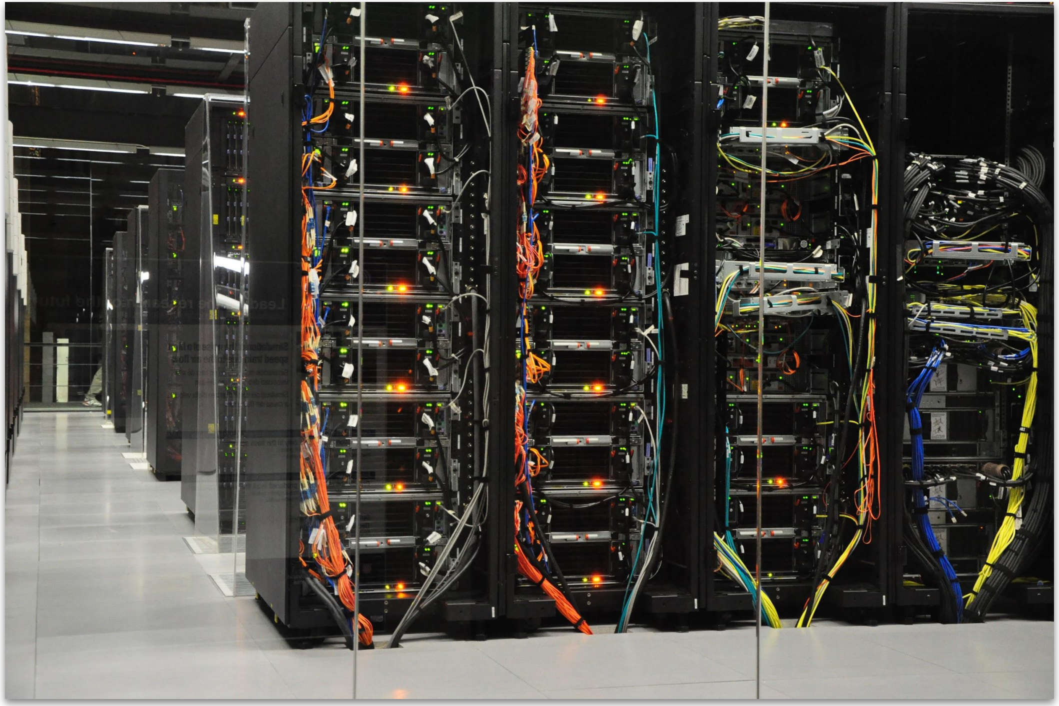
TPUs/GPUs

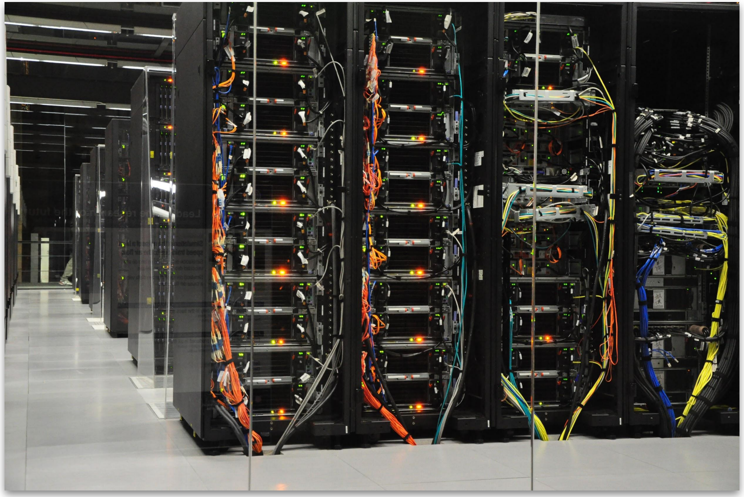


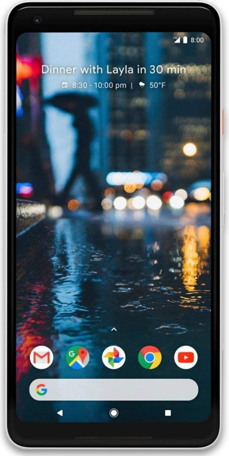


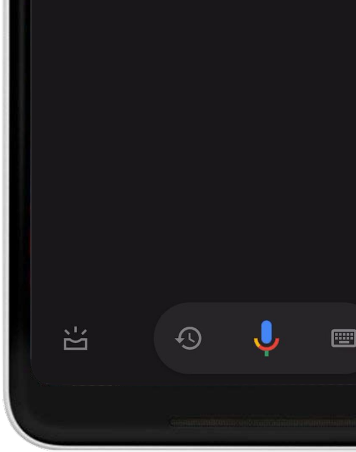

LIGHTMATTER
XILINX
ZYNQ
COILTRONICS
VP3-0780
41PH18 K
6-2301994-1
0818-A129
TE
10.10.10.161

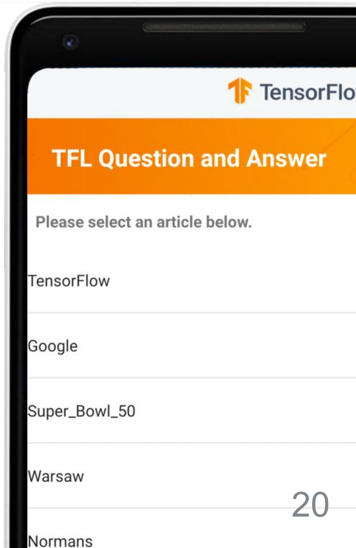
But… Bigger Is Not Always Better.

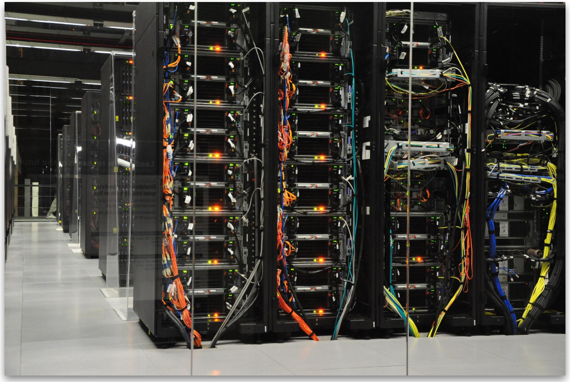




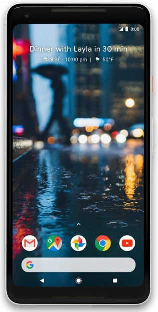

8:00
Dinner with Layla in 30 min
8:30 - 10:00 pm | 50°F



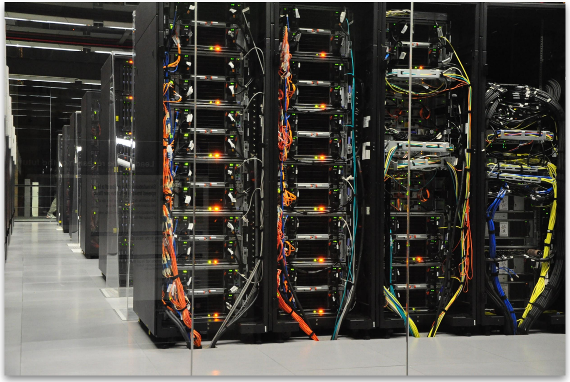

TensorFlo
TFL Question and Answer
Please select an article below.
TensorFlow
Google
Super_Bowl_50
Warsaw
Normans

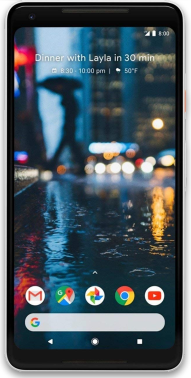


High power
High bandwidth
High latency



Low power
Low bandwidth
Low latency





Google Assistant

Endpoint Devices

Google Assistant

Endpoint Devices

Google Assistant

No Good Data Left Behind

5 Quintillion

bytes of data produced every day by IoT

<1%

of unstructured data is analyzed or used at all

Source: Harvard Business Review, What's Your Data Strategy?, April 18, 2017
Cisco, Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?, Feb 5, 2018

Tiny Machine Learning

What is Tiny Machine Learning (**TinyML**)?

What is Tiny Machine Learning (**TinyML**)?

TinyML → Fastest-growing field of **ML**

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

Low power consumption

What is Tiny Machine Learning (**TinyML**)?

TinyML

- Fastest-growing field of **ML**
- Algorithms, hardware, software
- **On-device** sensor analytics
- **Low power** consumption
- **Always-on ML**

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

Low power consumption

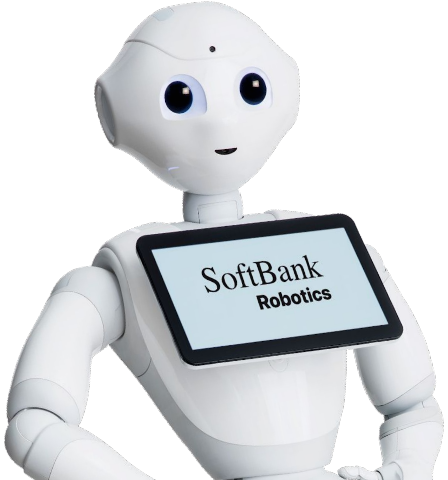
Always-on ML

Battery-operated

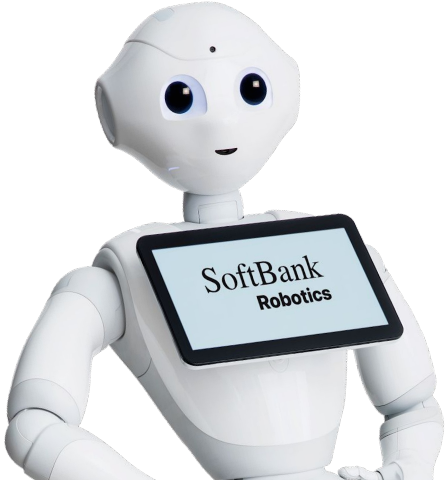

amazon

More Forward Looking Applications

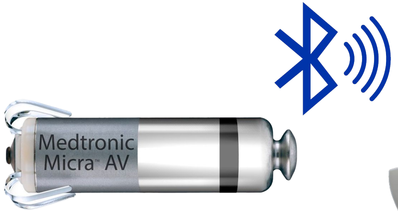
More Forward Looking Applications

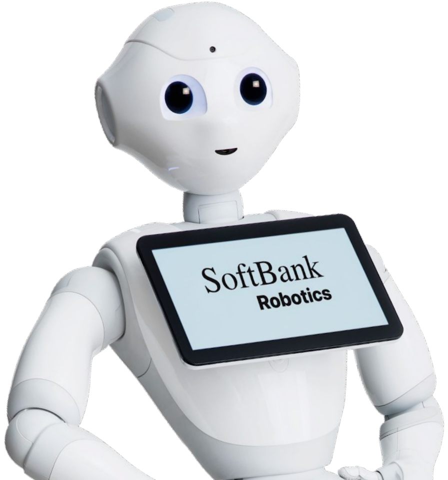


More Forward Looking Applications

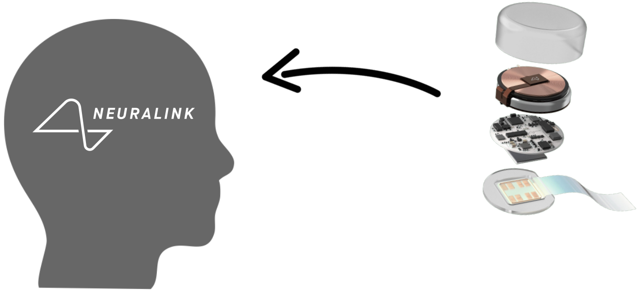


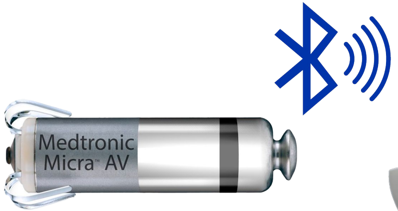
More Forward Looking Applications

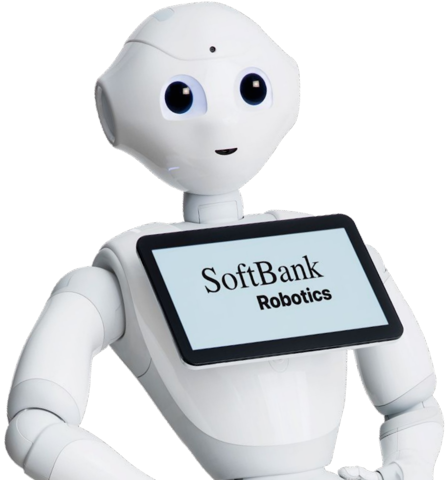


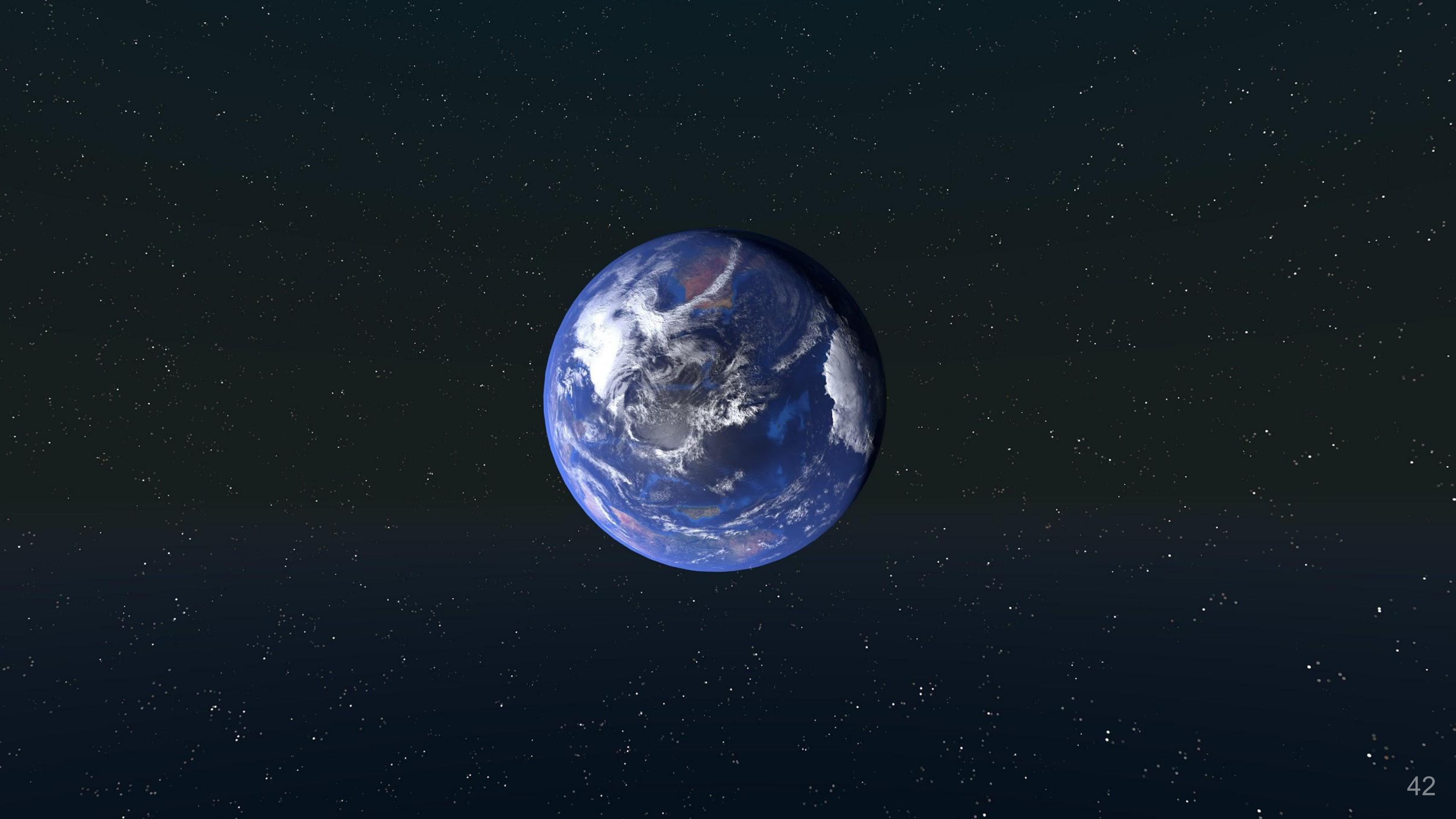


More Forward Looking Applications









Above
Female sperm whale (Image courtesy of Amanda Cotton)

Talking with whales

Project aims to translate sperm whale calls

By Leah Burrows | Press contact
April 22, 2021

This week, a team of scientists in partnership with the Government of Dominica and the National Geographic Society, officially launched an ambitious, interdisciplinary research initiative to listen to, contextualize, and translate the communication of sperm whales.

Project CETI (Cetacean Translation Initiative) will bring together leading cryptographers,

ElephantEdge

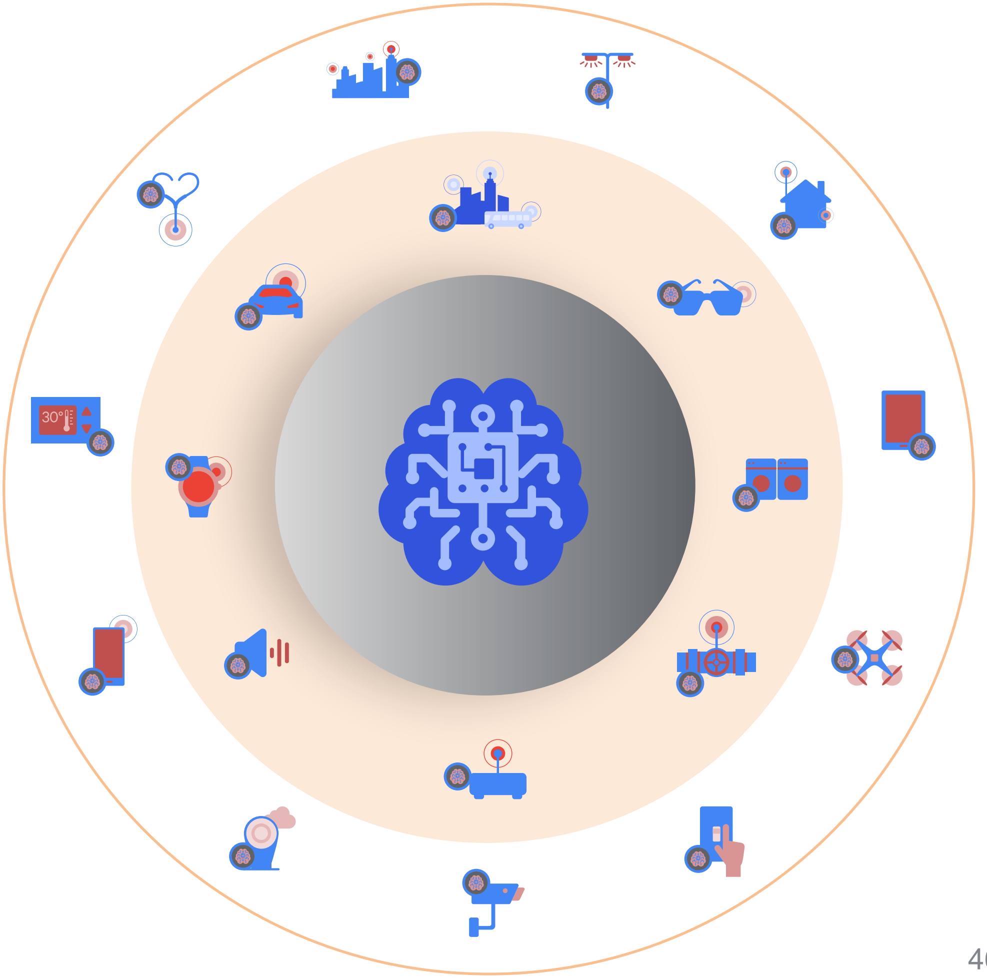
Building The World's Most Advanced **Wildlife Tracker**.

Source: https://wildlabs.net/resources/competition/challenge-elephantedge

ElephantEdge

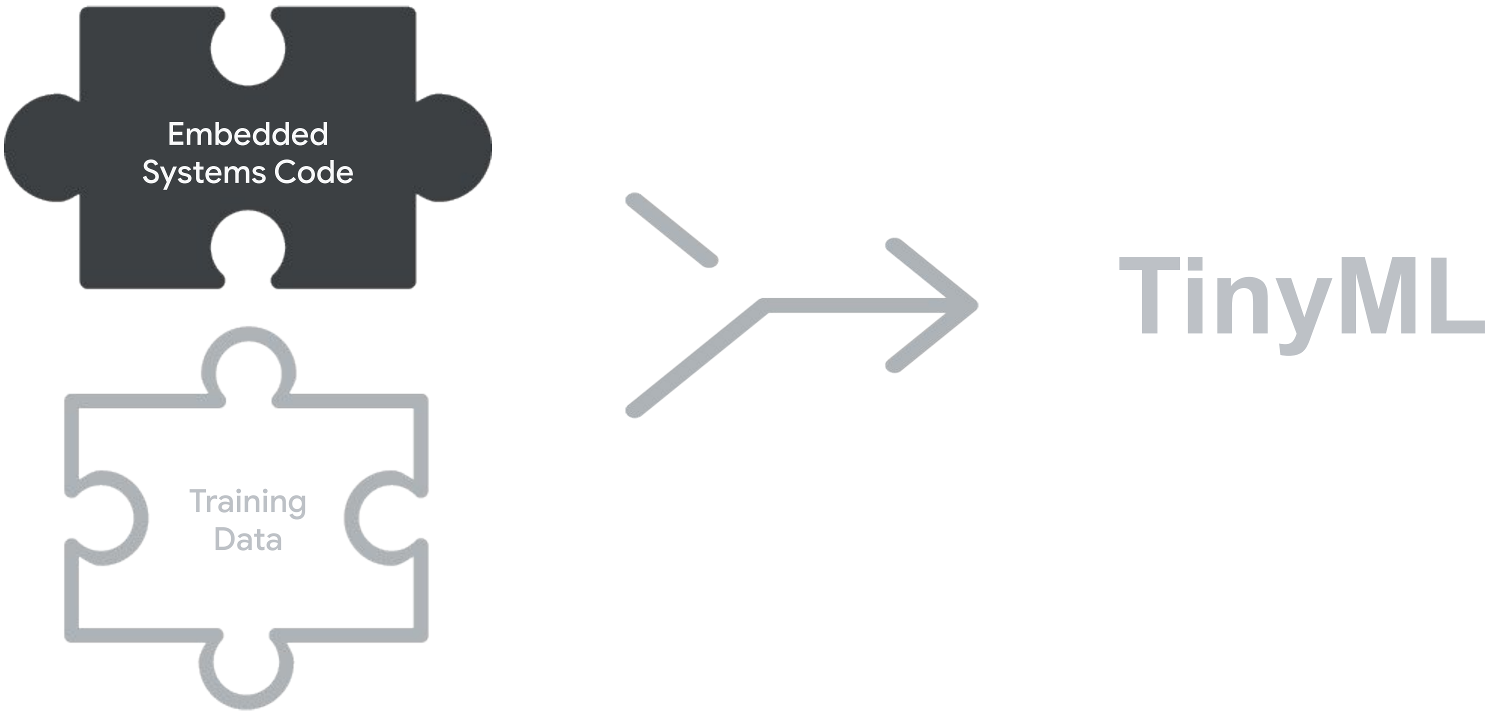
Risk Monitoring	"Know when an elephant is moving into a high-risk area and send real-time notifications to park rangers."
Conflict Monitoring	"Sense and alert when an elephant is heading into an area where farmers live."
Activity Monitoring	"Classify the general behavior of the elephant, such as when it is drinking, eating, sleeping, etc."
Communication Monitoring	"Listen for vocal communications between elephants via the onboard microphone."

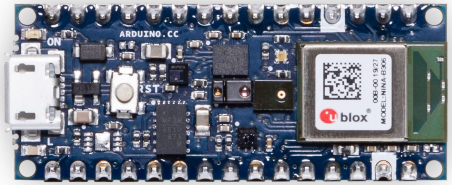
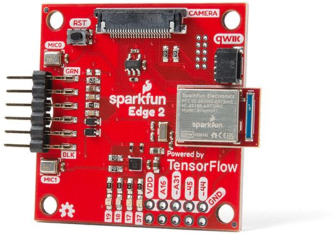
Massive tinyML opportunities in all verticals where machine intelligence meets physical world of billions of sensors



Technology for TinyML

What Makes **TinyML**?

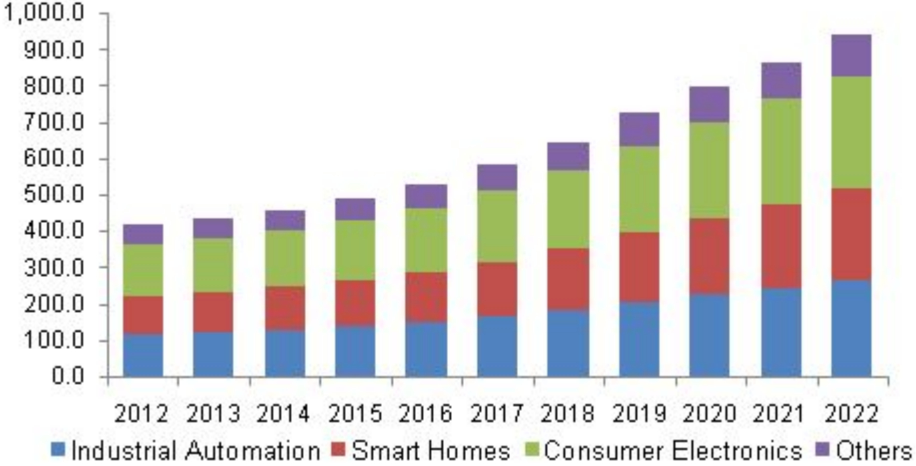



Himax

ARDUINO.CC
ON
RST
blox

CAMERA
RST
qwiic
sparkfun
Edge 2
Powered by
TensorFlow

ESP-EYE
V2.1

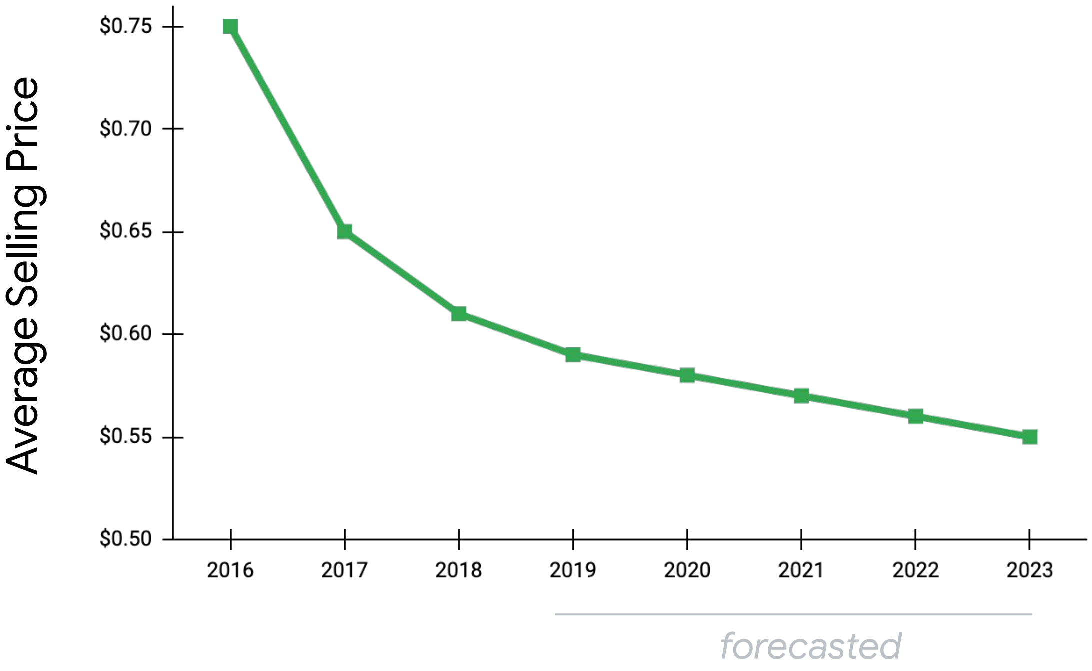
250 Billion

MCUs today

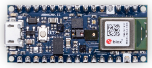
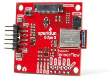
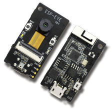


IoT Microcontroller Market Size, Share & Trends Analysis Report By Product (8-bit, 16-bit, 32-bit) By Application (Industrial Automation, Smart Home, Consumer Electronics) And Segment Forecasts To 2022

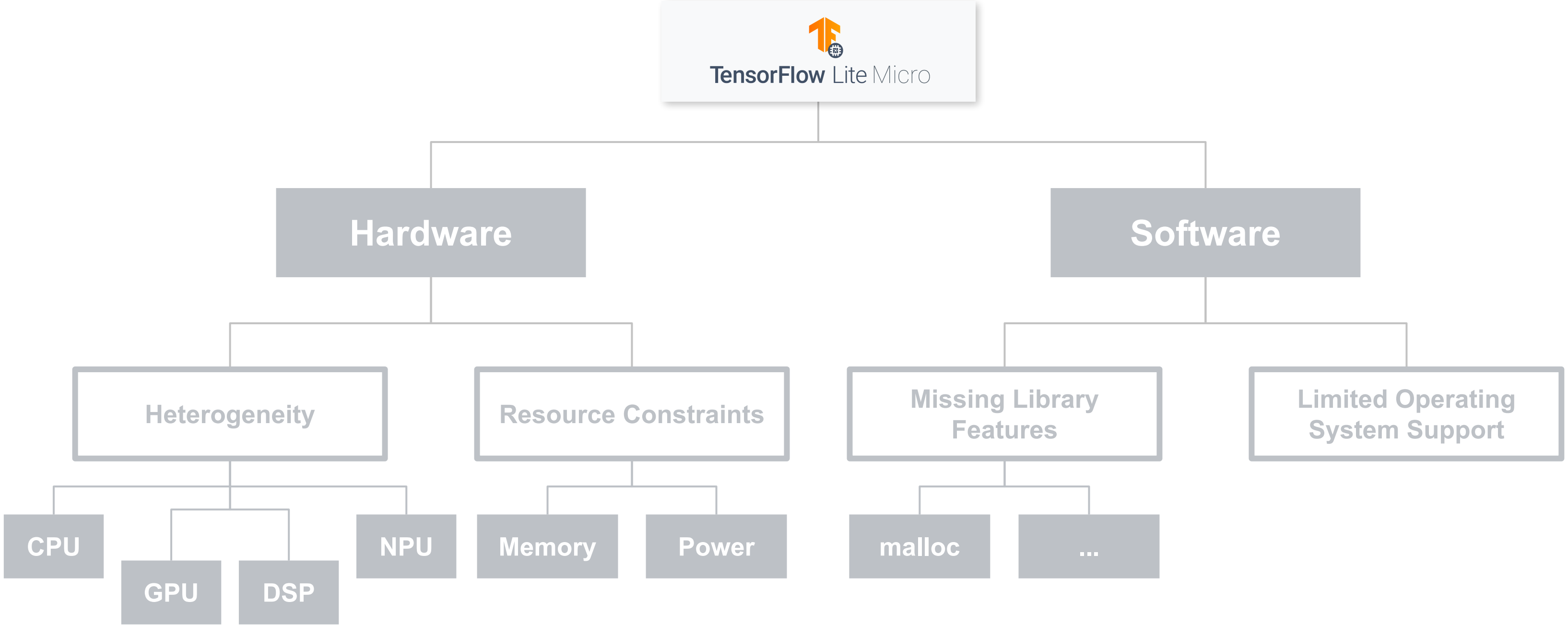
MCU Pricing Forecast

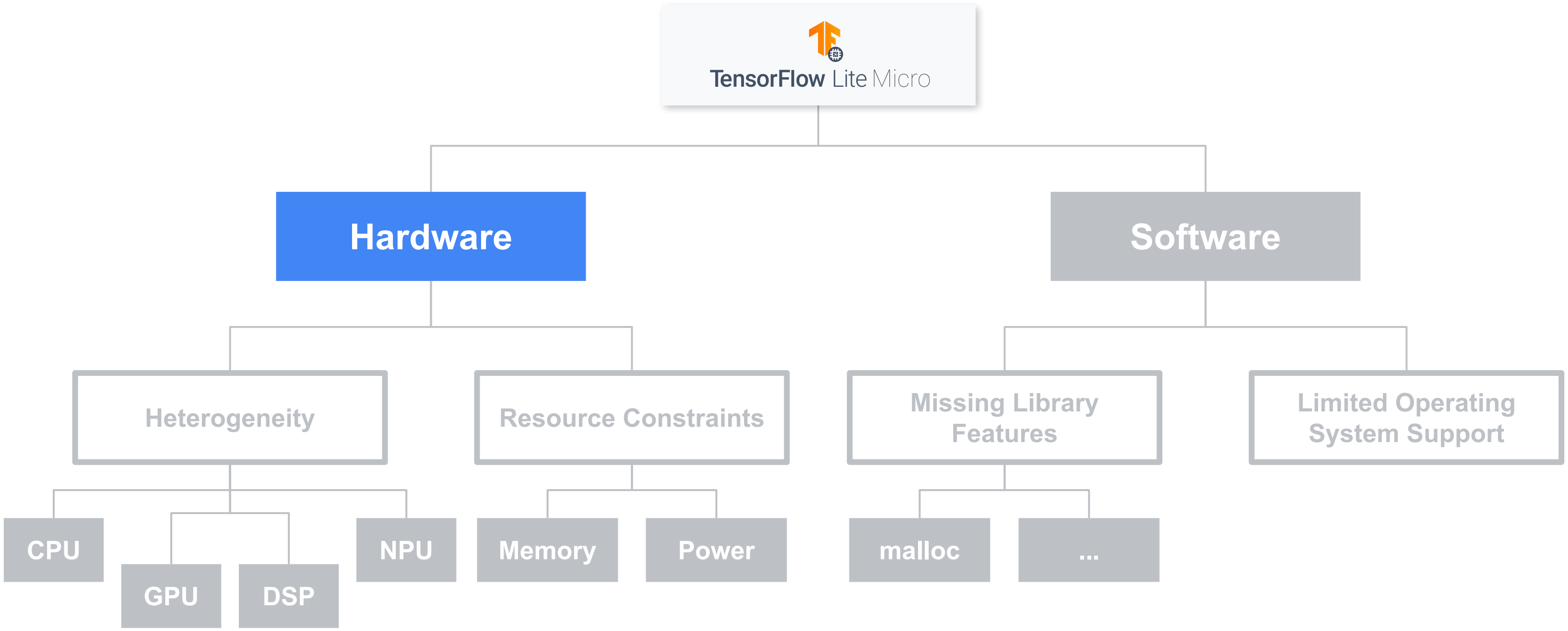


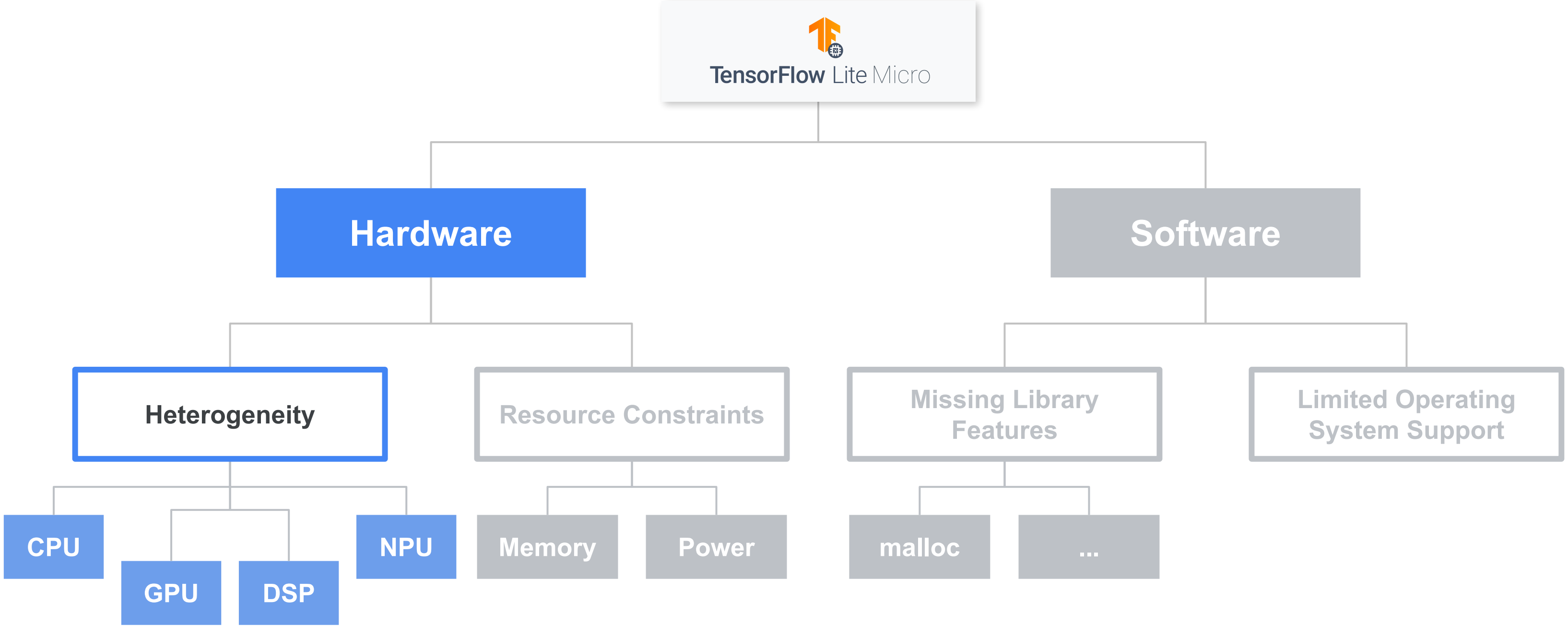
Source: IC Insights

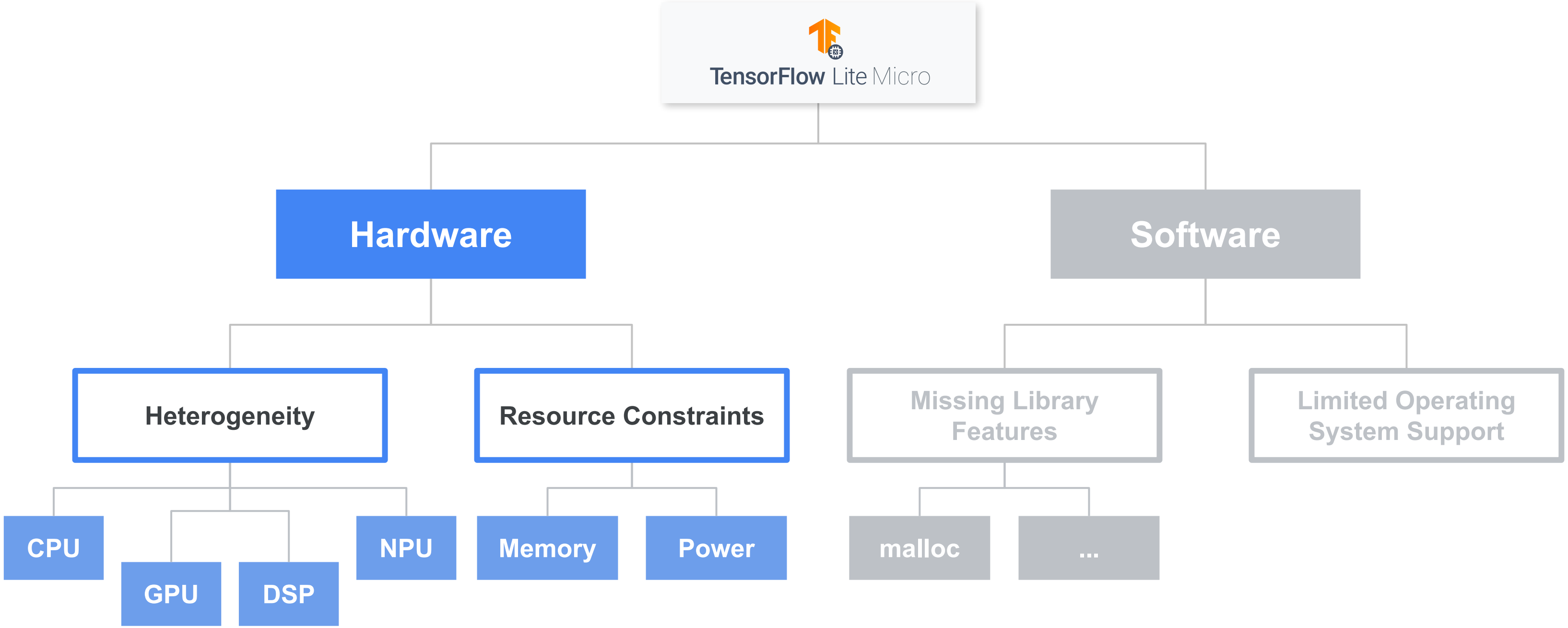
	Board	MCU / ASIC	Clock	Memory	Sensors	Radio
	Himax WE-I Plus EVB	HX6537-A 32-bit EM9D DSP	400 MHz	2MB flash 2MB RAM	Accelerometer, Mic, Camera	None
	Arduino Nano 33 BLE Sense	32-bit nRF52840	64 MHz	1MB flash 256kB RAM	Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color	BLE
	SparkFun Edge 2	32-bit ArtemisV1	48 MHz	1MB flash 384kB RAM	Accelerometer, Mic, Camera	BLE
	Espressif EYE	32-bit ESP32-D0WD	240 MHz	4MB flash 520kB RAM	Mic, Camera	WiFi, BLE

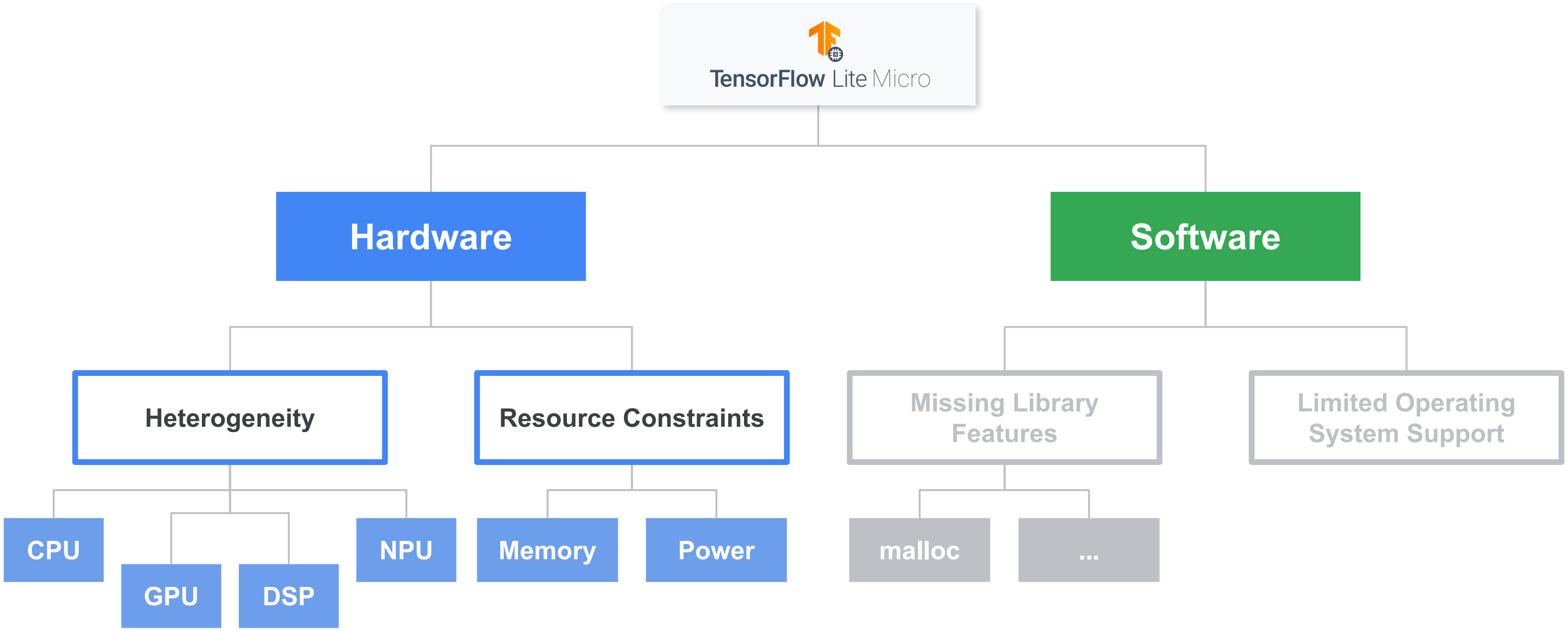
TensorFlow Lite Micro

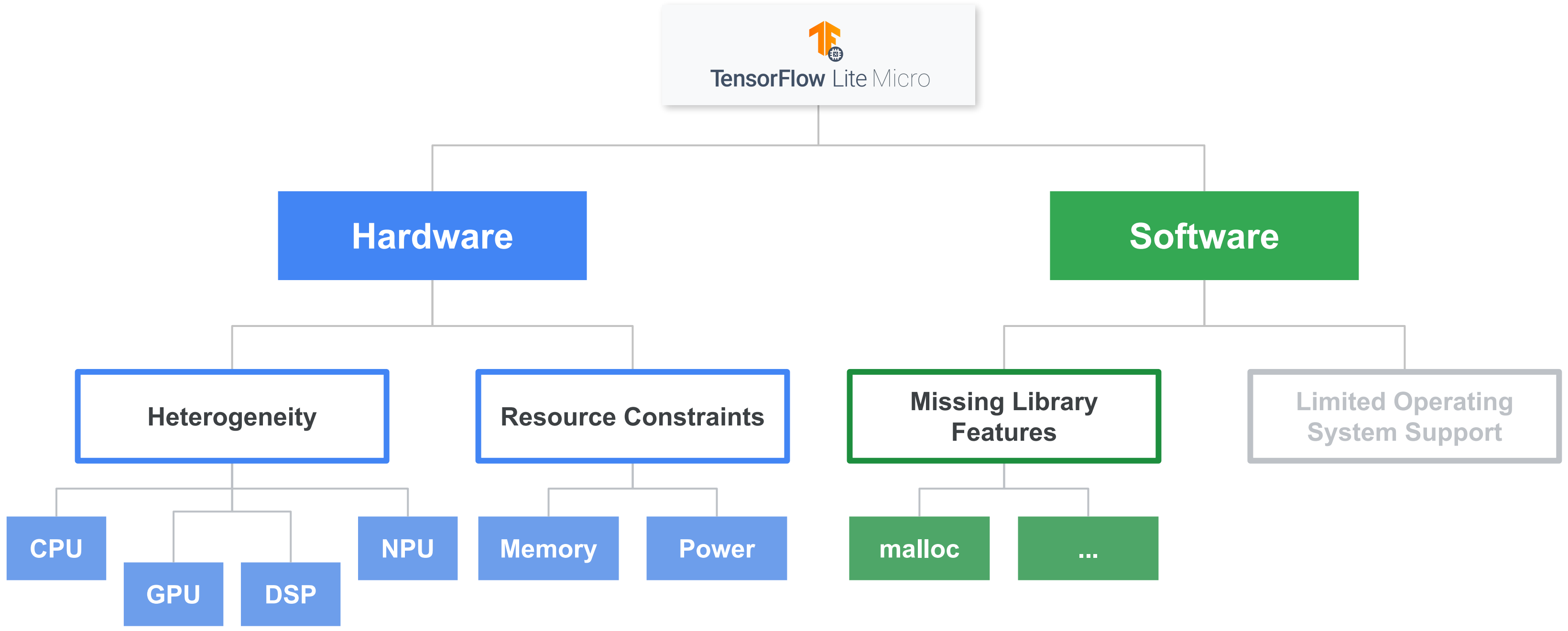

TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...

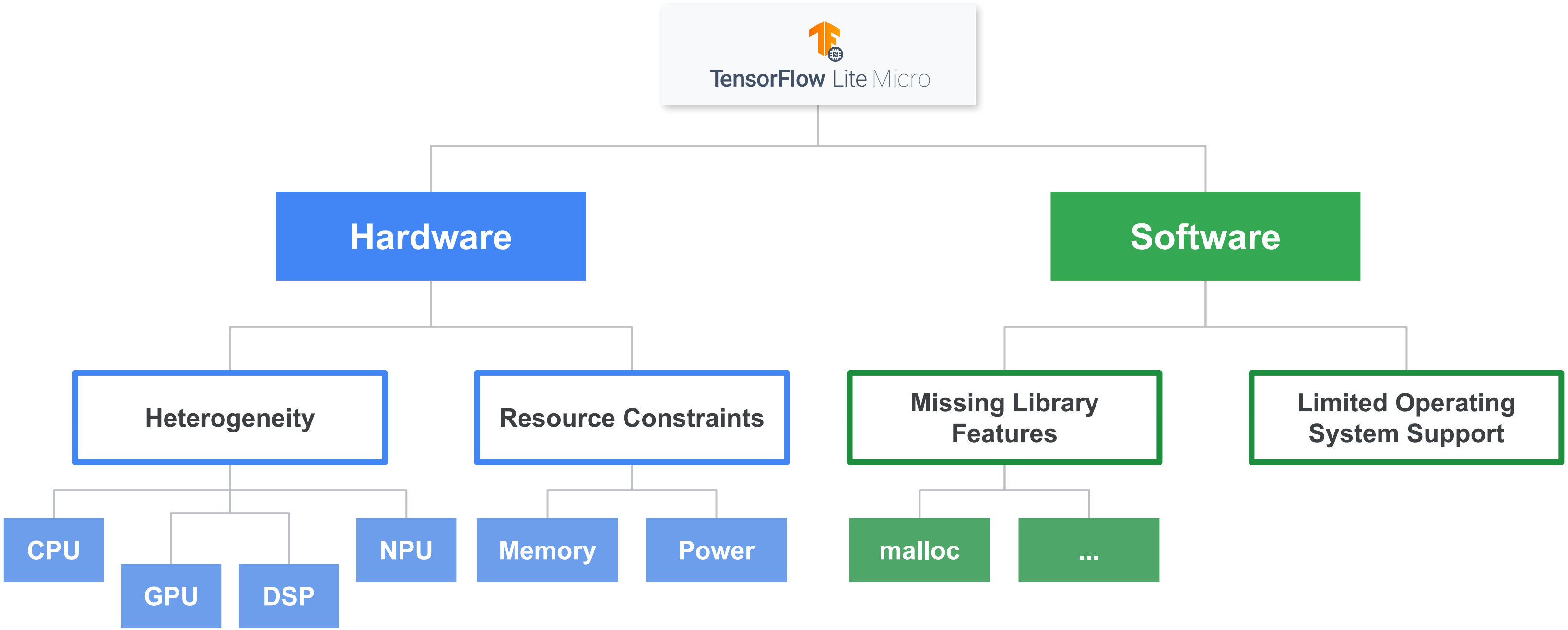

TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...

TensorFlow Lite Micro

...

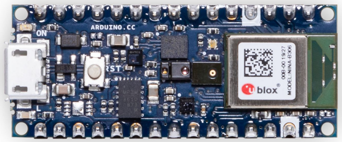
Arduino
BLE Sense 33

Himax
WE-I Plus EVB

SparkFun
Edge 2

Espressif
EYE

...



TensorFlow Lite Micro in a Nutshell

Compatible with the TensorFlow training environment.

Built to fit on **embedded systems**:

- Very **small binary footprint**
- **No** dynamic memory allocation
- **No** dependencies on complex parts of the standard C/C++ libraries
- **No** operating system dependencies, **can run on bare metal**
- Designed to be **portable** across a wide variety of systems

arXiv:2010.08678v3 [cs.LG] 13 Mar 2021

TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems

Robert David[1] Jared Duke[1] Advait Jain[1] Vijay Janapa Reddi[1 2]
Nat Jeffries[1] Jian Li[1] Nick Kreeger[1] Ian Nappier[1] Meghna Natraj[1]
Shlomi Regev[1] Rocky Rhodes[1] Tiezhen Wang[1] Pete Warden[1]

Abstract

TensorFlow Lite Micro (TFLM) is an open-source ML inference framework for running deep-learning models on embedded systems. TFLM tackles the efficiency requirements imposed by embedded-system resource constraints and the fragmentation challenges that make cross-platform interoperability nearly impossible. The framework adopts a unique interpreter-based approach that provides flexibility while overcoming these unique challenges. In this paper, we explain the design decisions behind TFLM and describe its implementation. We present an evaluation of TFLM to demonstrate its low resource requirements and minimal run-time performance overheads.

1 Introduction

Tiny machine learning (TinyML) is a burgeoning field at the intersection of embedded systems and machine learning. The world has over 250 billion microcontrollers (IC Insights, 2020), with strong growth projected over coming years. As such, a new range of embedded applications are emerging for neural networks. Because these models are extremely small (few hundred KBs), running on microcontrollers or DSP-based embedded subsystems, they can operate continuously with minimal impact on device battery life.

The most well-known and widely deployed example of this new TinyML technology is keyword spotting, also called hotword or wakeword detection (Chen et al., 2014; Gruenstein et al., 2017; Zhang et al., 2017). Amazon, Apple, Google, and others use tiny neural networks on billions of devices to run always-on inferences for keyword detection—and this is far from the only TinyML application. Low-latency analysis and modeling of sensor signals from microphones, low-power image sensors, accelerometers, gyros, PPG optical sensors, and other devices enable consumer and industrial applications, including predictive maintenance (Goebel et al., 2020; Susto et al., 2014), acoustic-anomaly detection (Koizumi et al., 2019), visual object detection (Chowdhery et al., 2019), and human-activity recognition (Chavarriaga et al., 2013; Zhang & Sawchuk, 2012).

Unlocking machine learning's potential in embedded devices requires overcoming two crucial challenges. First and foremost, embedded systems have no unified TinyML framework. When engineers have deployed neural networks to such systems, they have built one-off frameworks that require manual optimization for each hardware platform. Such custom frameworks have tended to be narrowly focused, lacking features to support multiple applications and lacking portability across a wide range of hardware. The developer experience has therefore been painful, requiring hand optimization of models to run on a specific device. And altering these models to run on another device necessitated manual porting and repeated optimization effort. An important second-order effect of this situation is that the slow pace and high cost of training and deploying models to embedded hardware prevents developers from easily justifying the investment required to build new features.

Another challenge limiting TinyML is that hardware vendors have related but separate needs. Without a generic TinyML framework, evaluating hardware performance in a neutral, vendor-agnostic manner has been difficult. Frameworks are tied to specific devices, and it is hard to determine the source of improvements because they can come from hardware, software, or the complete vertically integrated solution.

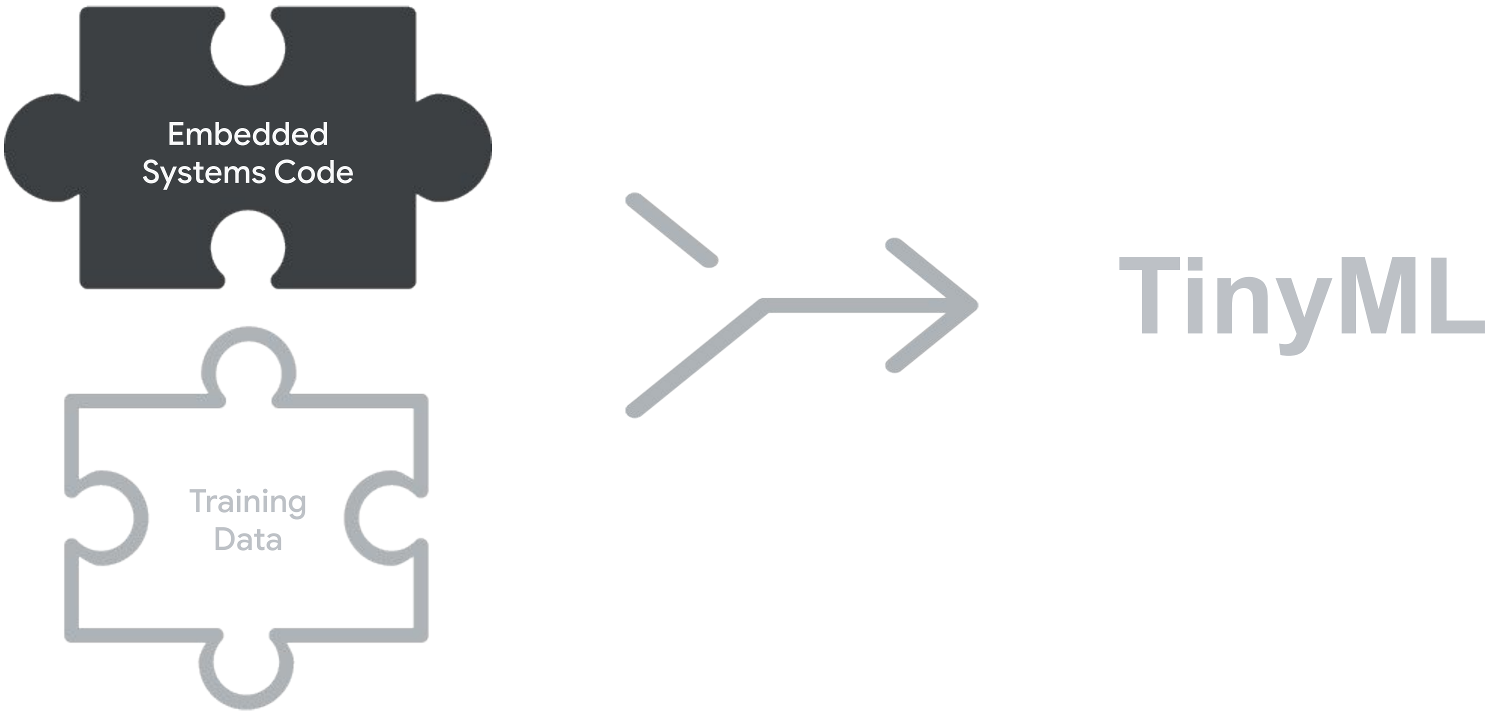
The lack of a proper framework has been a barrier to accelerating TinyML adoption and application in products. Beyond deploying a model to an embedded target, the framework must also have a means of training a model on a higher-compute platform. TinyML must exploit a broad ecosystem of tools for ML, as well for orchestrating and debugging models, which are beneficial for production devices.

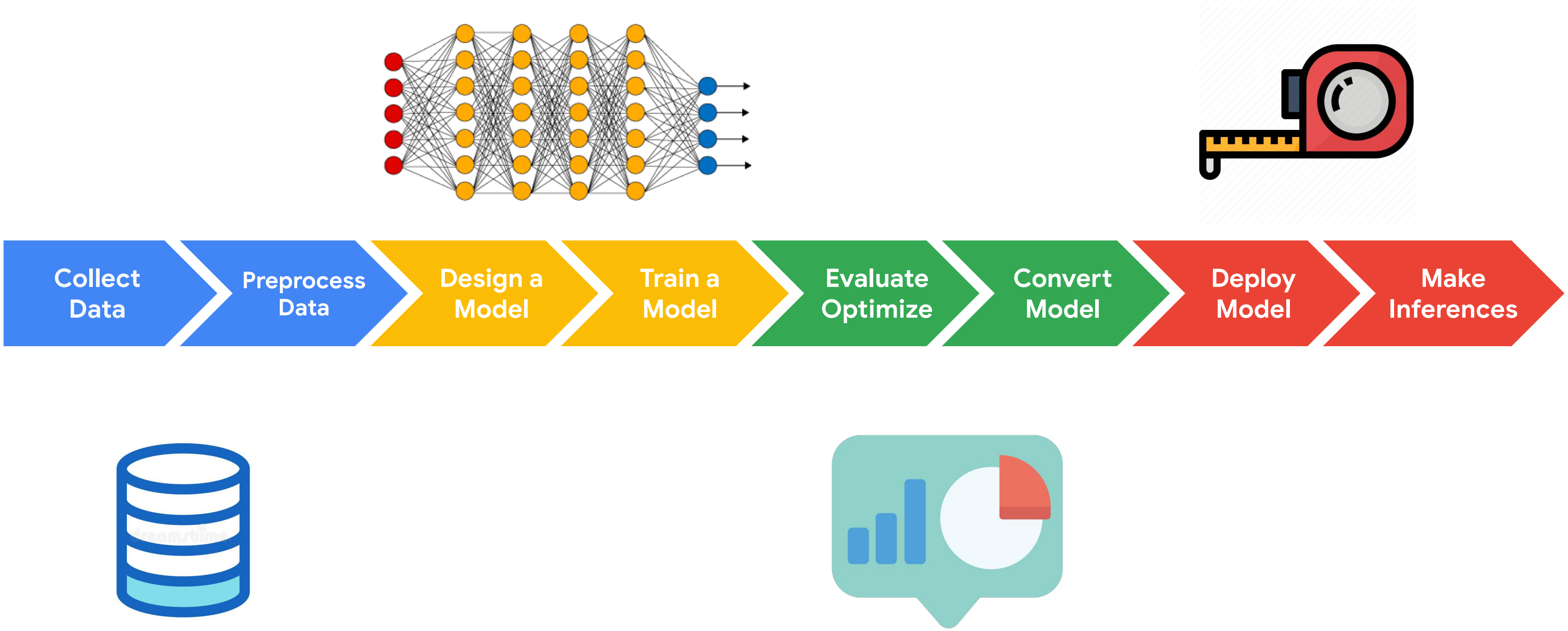
Prior efforts have attempted to bridge this gap. We can distill the major issues facing the frameworks into the following:

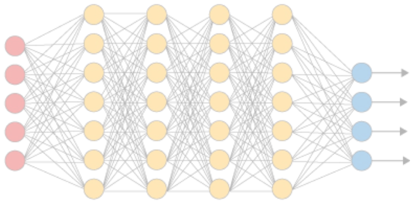
[1]Google [2]Harvard University. Correspondence to: Pete Warden <petewarden@google.com>, Vijay Janapa Reddi <vj@eecs.harvard.edu>.

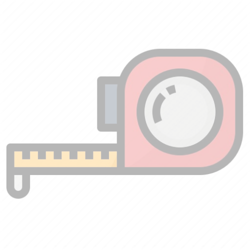
Proceedings of the 4^{th} MLSys Conference, San Jose, CA, USA, 2021.

What Makes **TinyML**?

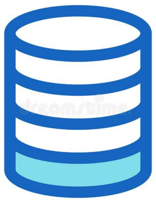


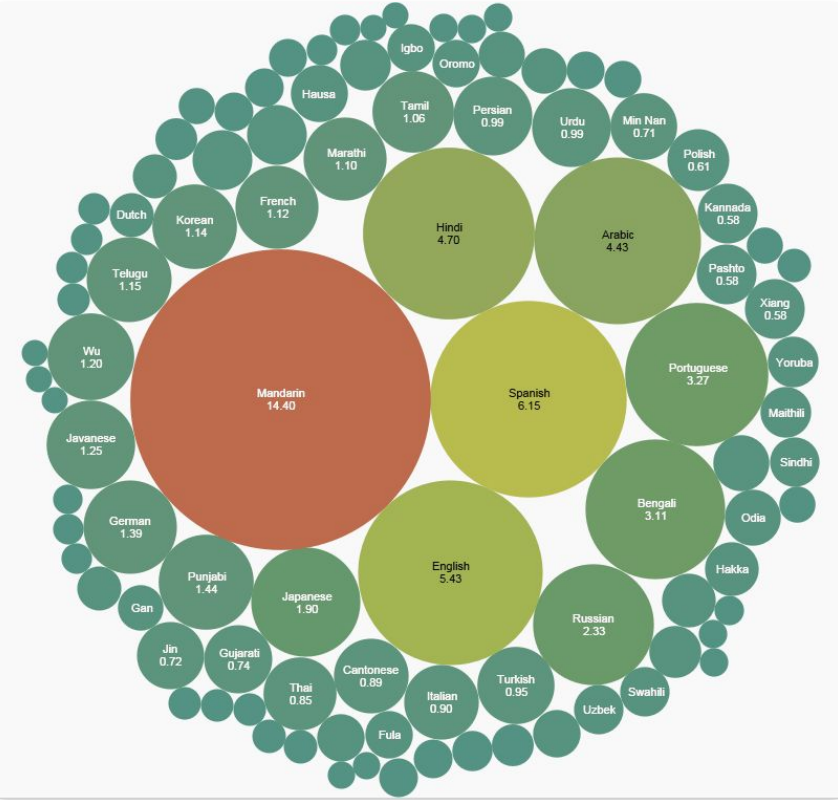

Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences



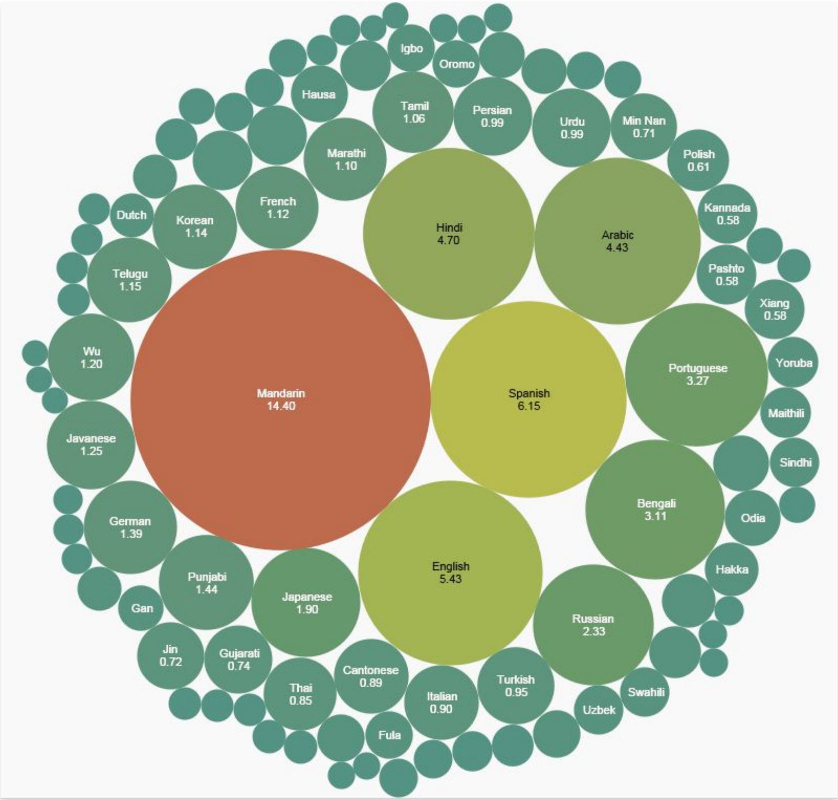


Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences

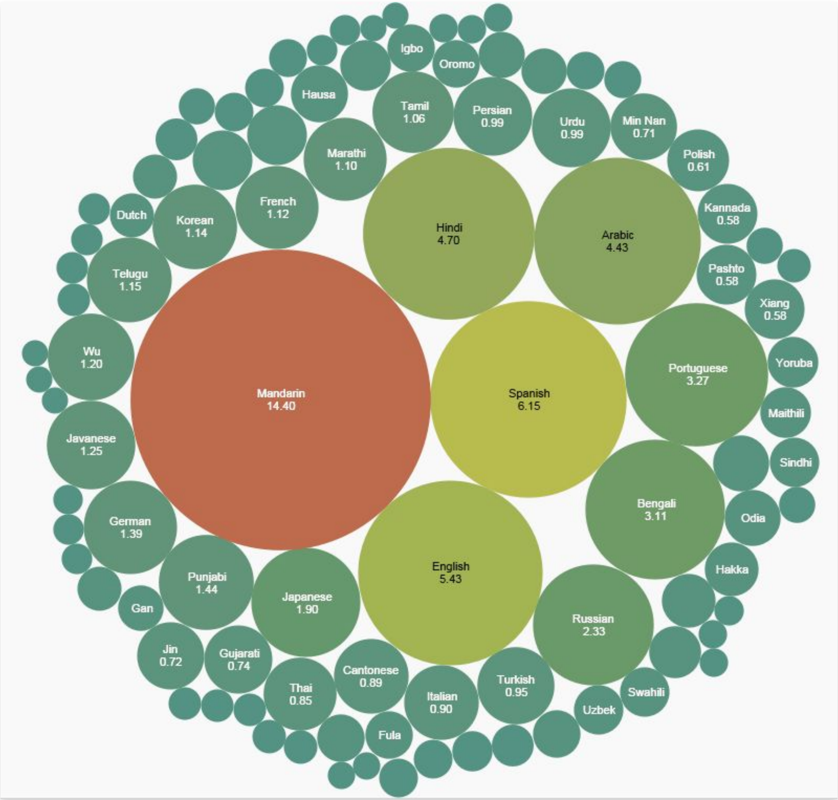


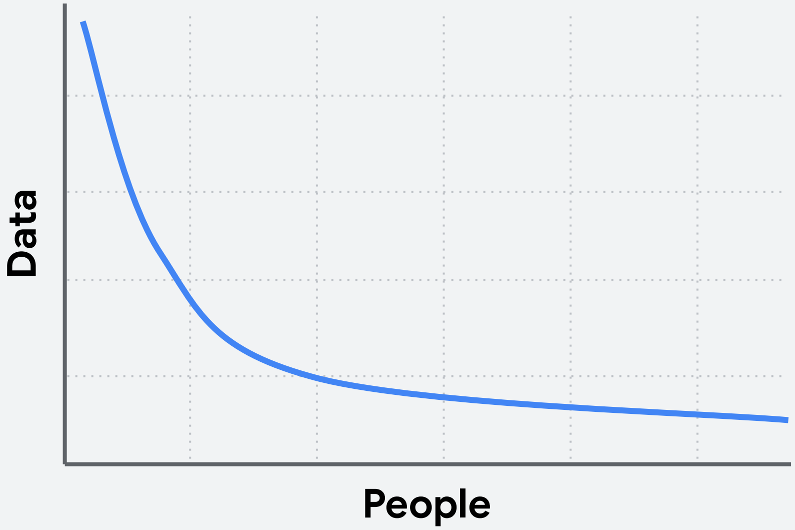


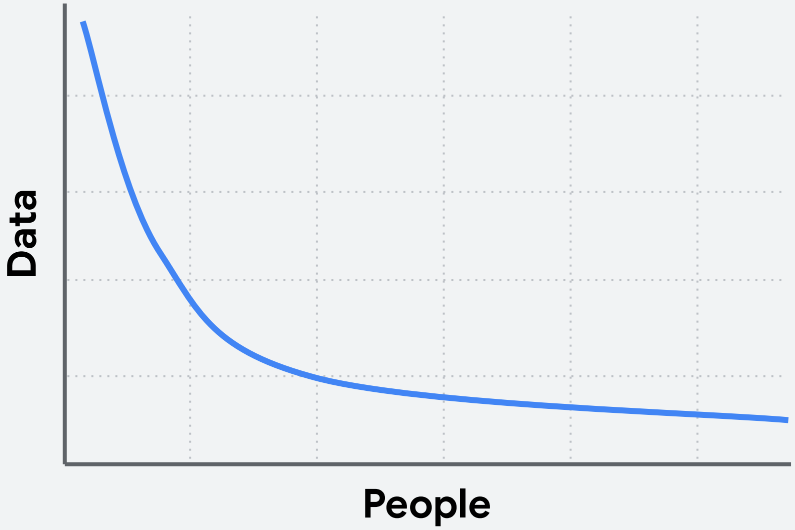
- Speech commands for the **whole planet**?



- Speech commands for the **whole planet**?
- For **more than** just voice assistants


Mandarin
14.40
Spanish
6.15
English
5.43
Hindi
4.70
Arabic
4.43
Portuguese
3.27
Bengali
3.11
Russian
2.33
Japanese
1.90
Punjabi
1.44
German
1.39
Javanese
1.25
Wu
1.20
Telugu
1.15
Korean
1.14
French
1.12
Marathi
1.10
Tamil
1.06
Persian
0.99
Urdu
0.99
Min Nan
0.71
Polish
0.61
Kannada
0.58
Pashto
0.58
Xiang
0.58
Turkish
0.95
Italian
0.90
Cantonese
0.89
Thai
0.85
Gujarati
0.74
Jin
0.72
Dutch
Hausa
Igbo
Oromo
Yoruba
Maithili
Sindhi
Odia
Hakka
Swahili
Uzbek
Fula
Gan


Data
People


Data
People

Data Engineering

Requirements

- Problem definition
- Permissions & rights
- Machine & human usable format

Data Engineering

Requirements → **Gathering**

- Problem definition
- Permissions & rights
- Machine & human usable format

- People
- Collection
- Labeling
- Data sources

Data Engineering

Requirements

- Problem definition
- Permissions & rights
- Machine & human usable format

Gathering

- People
- Collection
- Labeling
- Data sources

Refinement

- Processing
- Validation
- Augmentation

Data Engineering

Requirements → **Gathering** → **Refinement** → **Sustainment**

Requirements
- Problem definition
- Permissions & rights
- Machine & human usable format

Gathering
- People
- Collection
- Labeling
- Data sources

Refinement
- Processing
- Validation
- Augmentation

Sustainment
- Storage
- Security
- Errors
- Versioning

Data Engineering

Requirements

- Problem definition
- Permissions & rights
- Machine & human usable format

Gathering

- People
- Collection
- Labeling
- Data sources

Refinement

- Processing
- Validation
- Augmentation

Sustainment

- Storage
- Security
- Errors
- Versioning

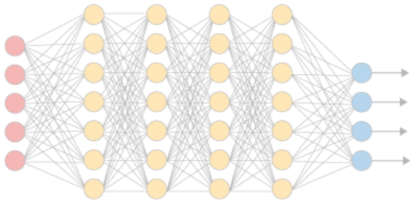
Datasets require ***significant effort***

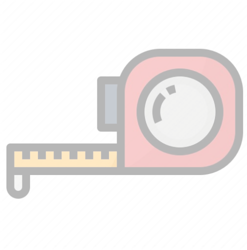
These **massive** machine learning datasets are ***constructed by hand***

- **Common Voice**—**5000+** hours of spoken audio
- Common Objects in Context (**COCO**)—**2.5M+** labeled images
- **ImageNet**—**4M+** labeled images
- **Waymo—1,950** 20-second driving segments
- **KITTI 360**—**73KM+** of annotated driving data

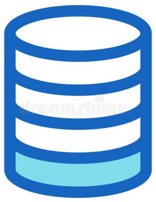
Data Engineering is costly and tedious.

Democratize Data Engineering





- Collect Data
- Preprocess Data
- Design a Model
- Train a Model
- Evaluate Optimize
- Convert Model
- Deploy Model
- Make Inferences



Automatic Keyword Dataset Generation

Specify Wanted Keywords

1. Up
2. Down
3. Yes
4. No
5. …
6. …

…

265. ...

Automatic Keyword Dataset Generation

Specify Wanted Keywords

1. Up
2. Down
3. Yes
4. No
5. …
6. …

…

265. ...

Search CV Sentences

en/validated.tsv

He gazed **up** the steep bank.	clip_29132.mp3
"**Yes**," said Harry sullenly.	clip_34212.mp3
Pencils **down**, time is over.	clip_54972.mp3
Get that ladder **up** here.	clip_28213.mp3
There is **no** help for it.	clip_38311.mp3
...	...

All sentences containing keywords

Automatic Keyword Dataset Generation

Specify Wanted Keywords

1. Up
2. Down
3. Yes
4. No
5. …
6. …

…

265. ...

Search CV Sentences

en/validated.tsv

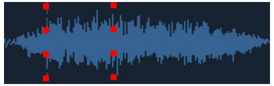
He gazed **up** the steep bank.	clip_29132.mp3
"**Yes**," said Harry sullenly.	clip_34212.mp3
Pencils **down**, time is over.	clip_54972.mp3
Get that ladder **up** here.	clip_28213.mp3
There is **no** help for it.	clip_38311.mp3
...	...

All sentences containing keywords

Extract Keyword Utterances

Input 5-10 second mp3

*"... gazed **up** the steep..."*

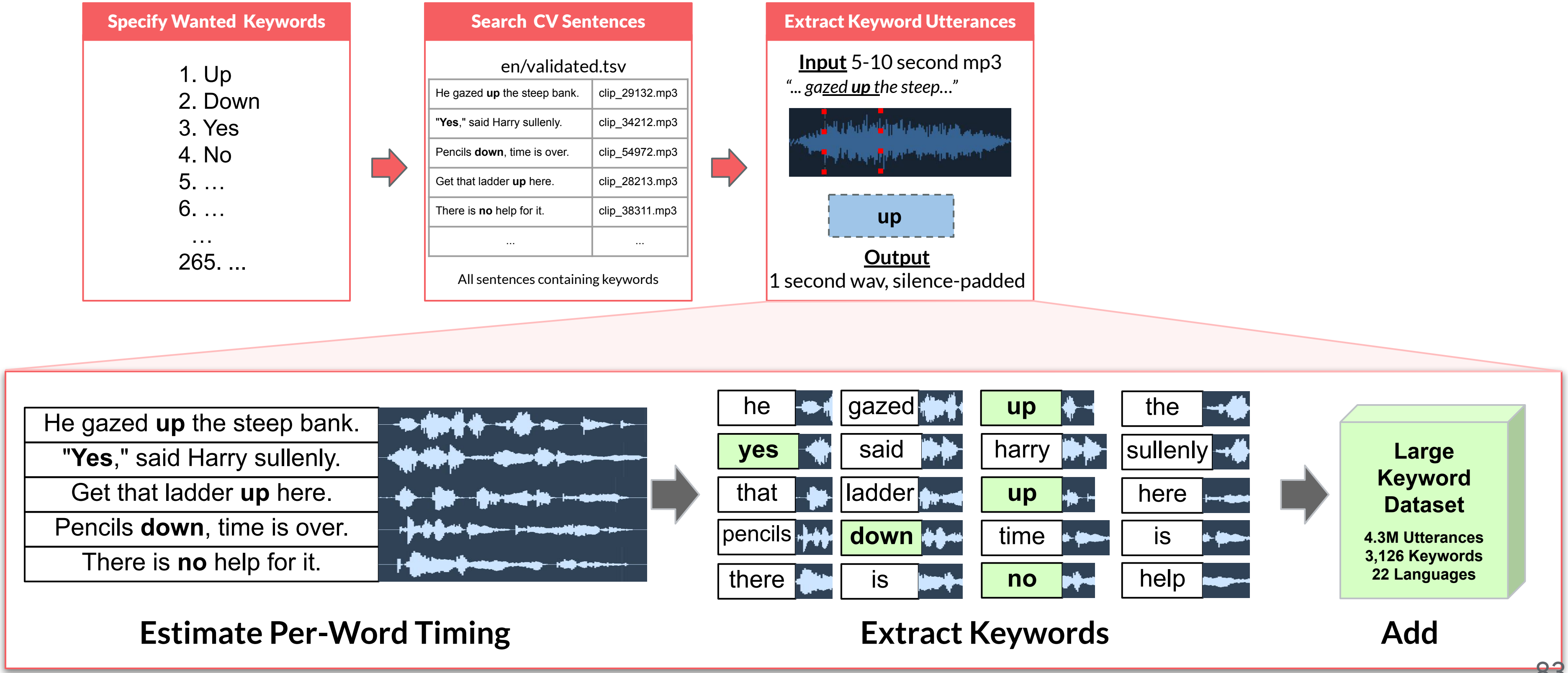


up

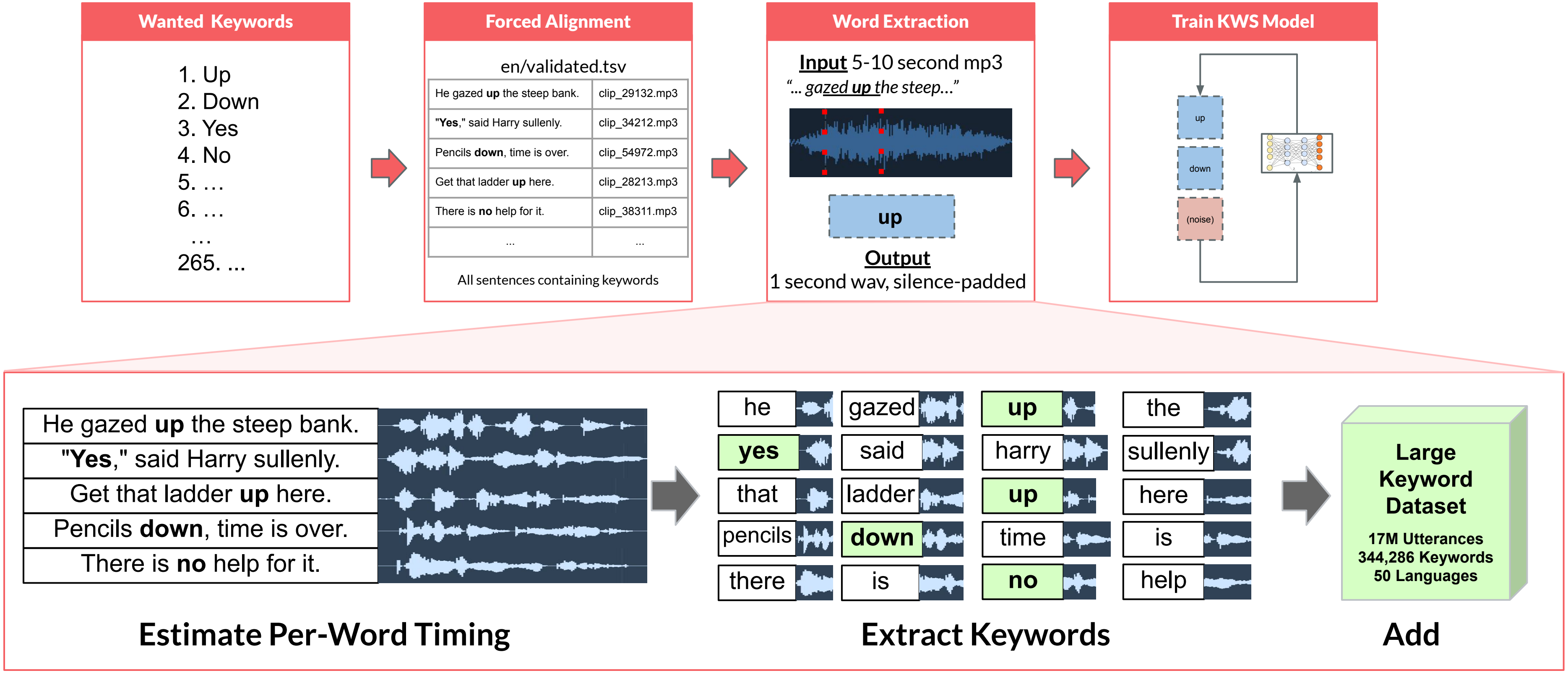
Output

1 second wav, silence-padded

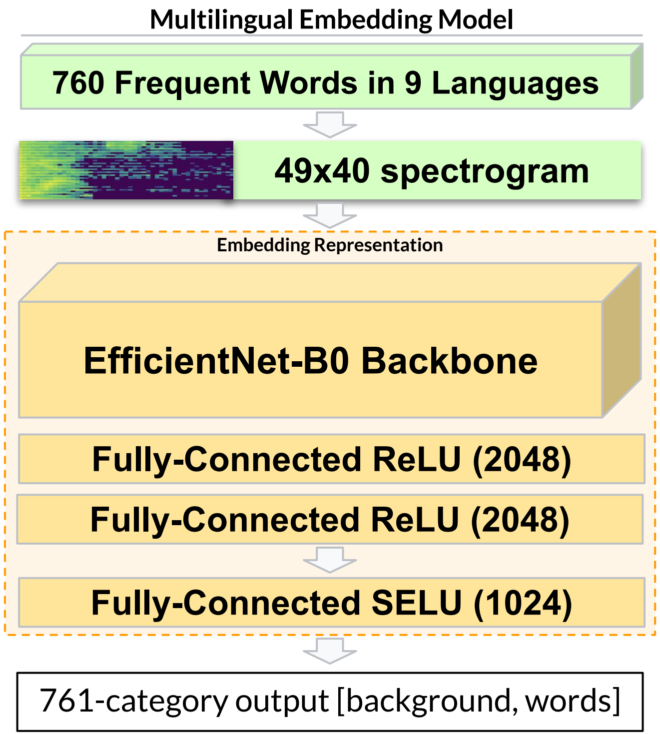
Automatic Keyword Dataset Generation



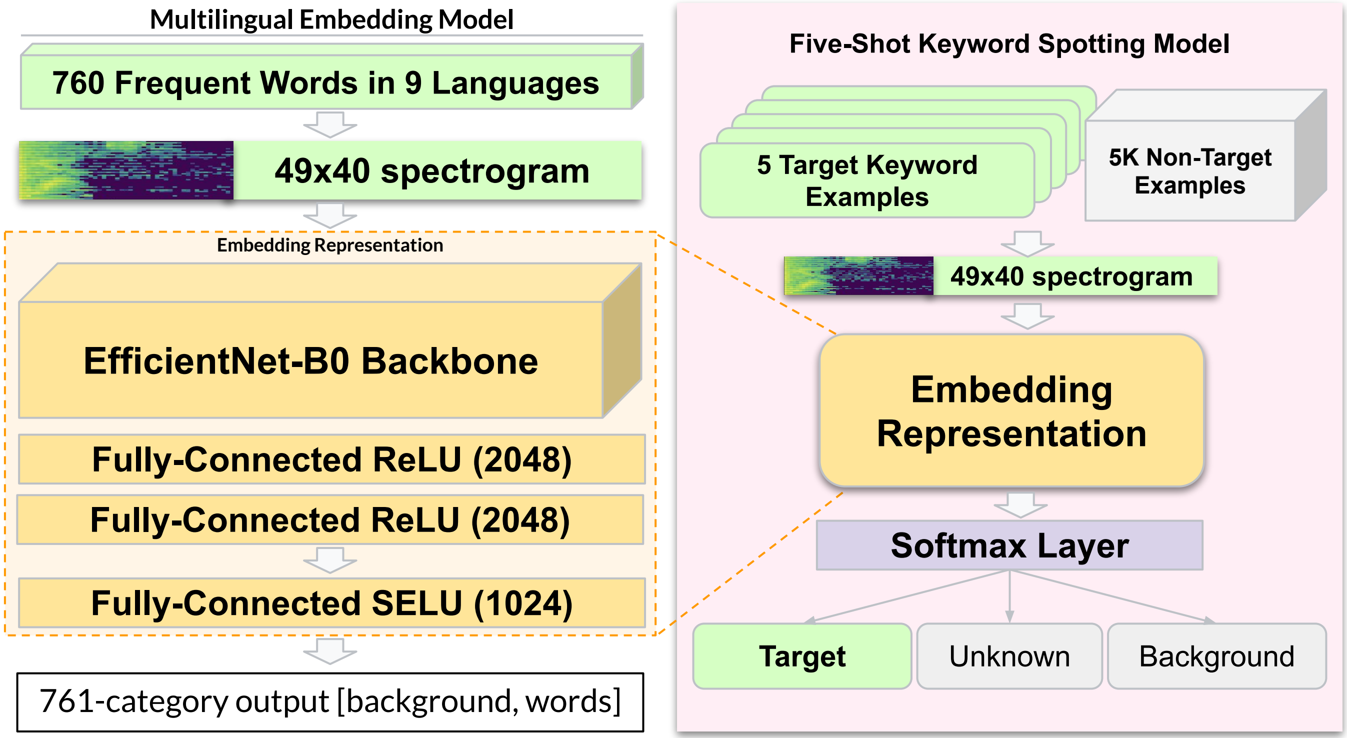
Automatic Keyword Dataset Generation



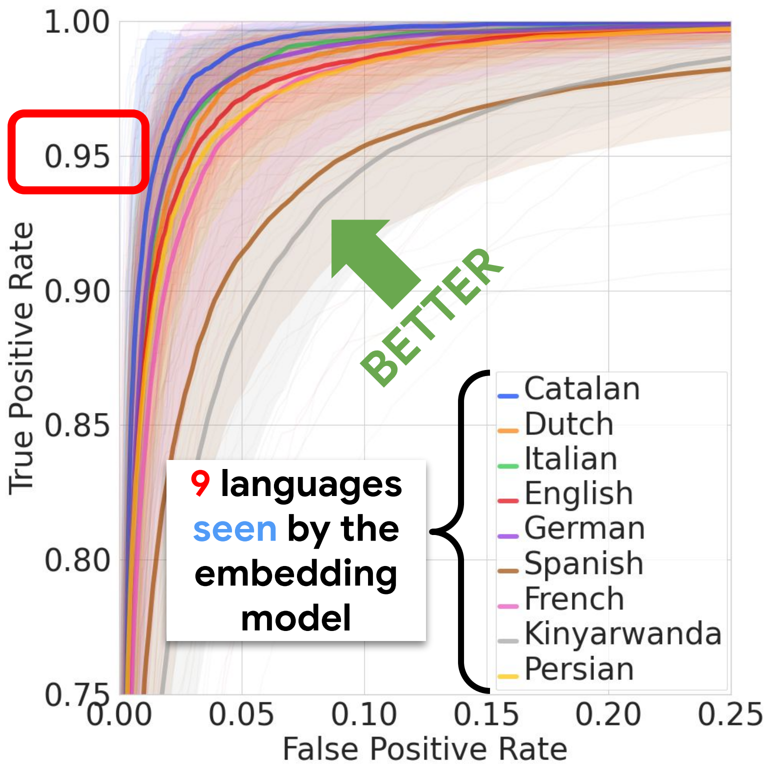
Nine-Language Embedding Model



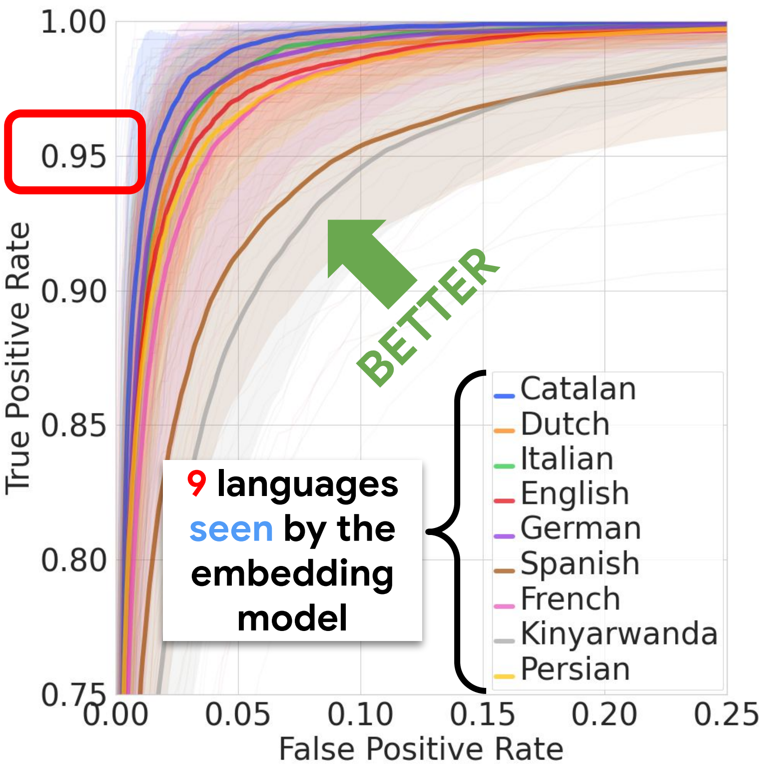
Nine-Language Embedding Model

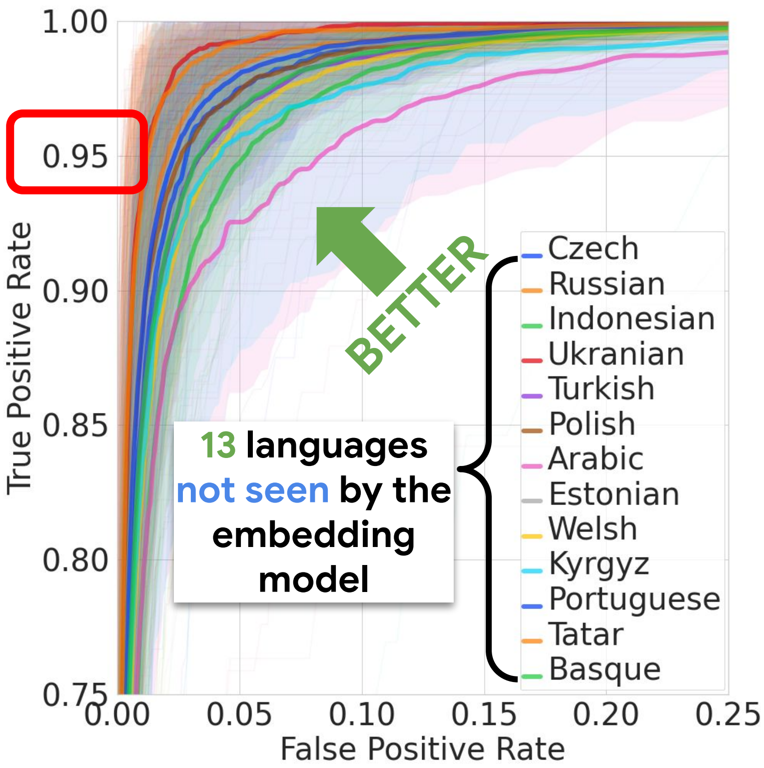


Generalizing to **Any** Language



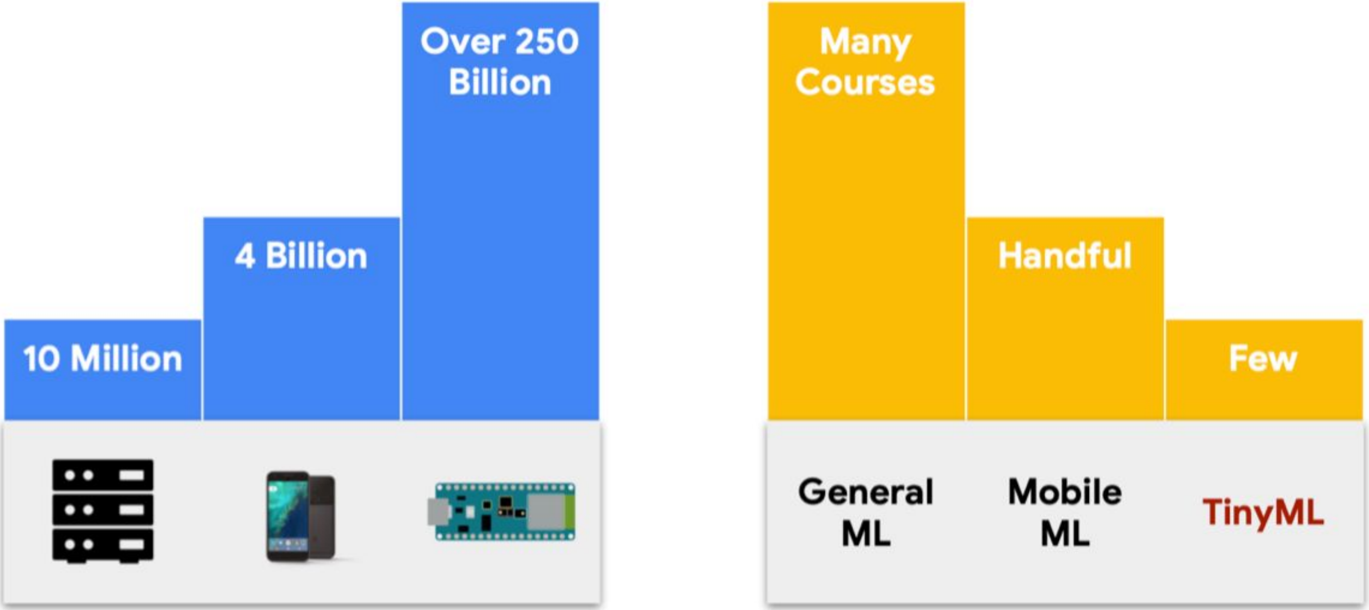
Generalizing to **Any** Language





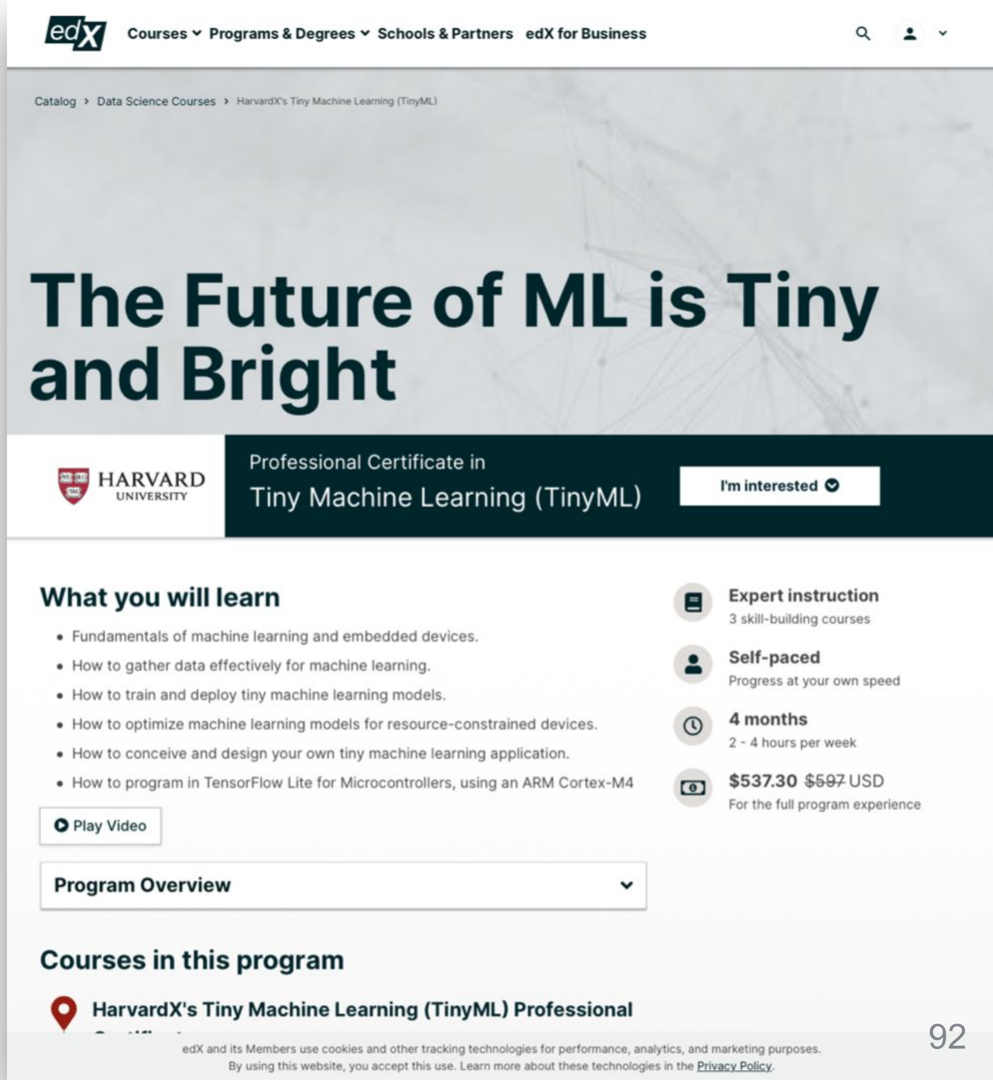
Challenge:
1000 Words in 1000 Languages

Widening Access to Applied ML

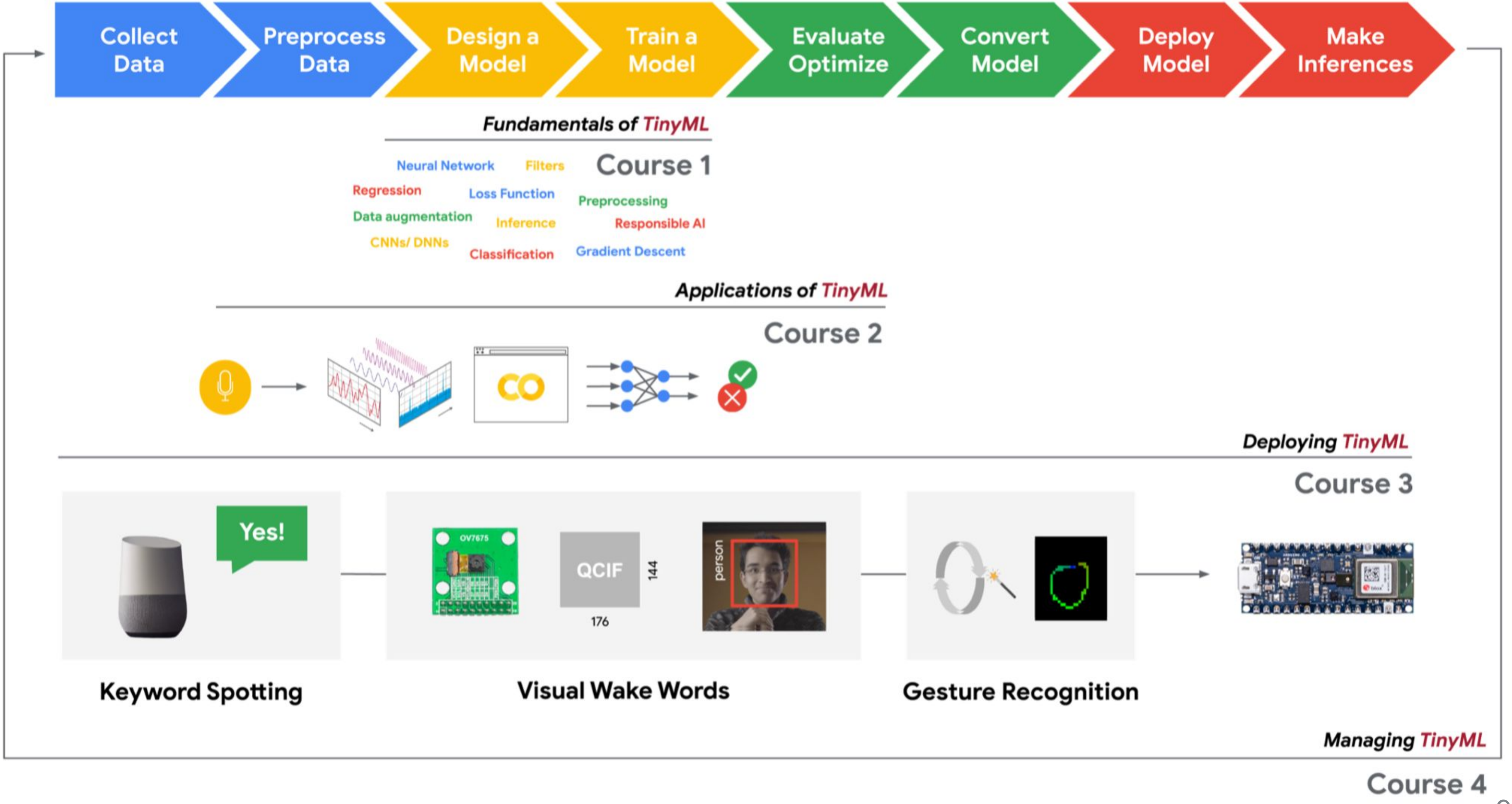

10 Million
4 Billion
Over 250 Billion
Many Courses
Handful
Few
General ML
Mobile ML
TinyML

HARVARD UNIVERSITY

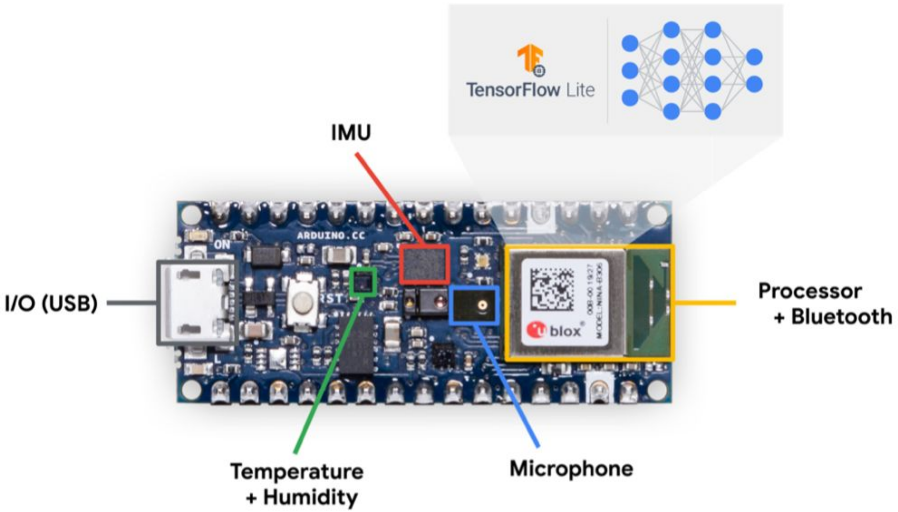
Google

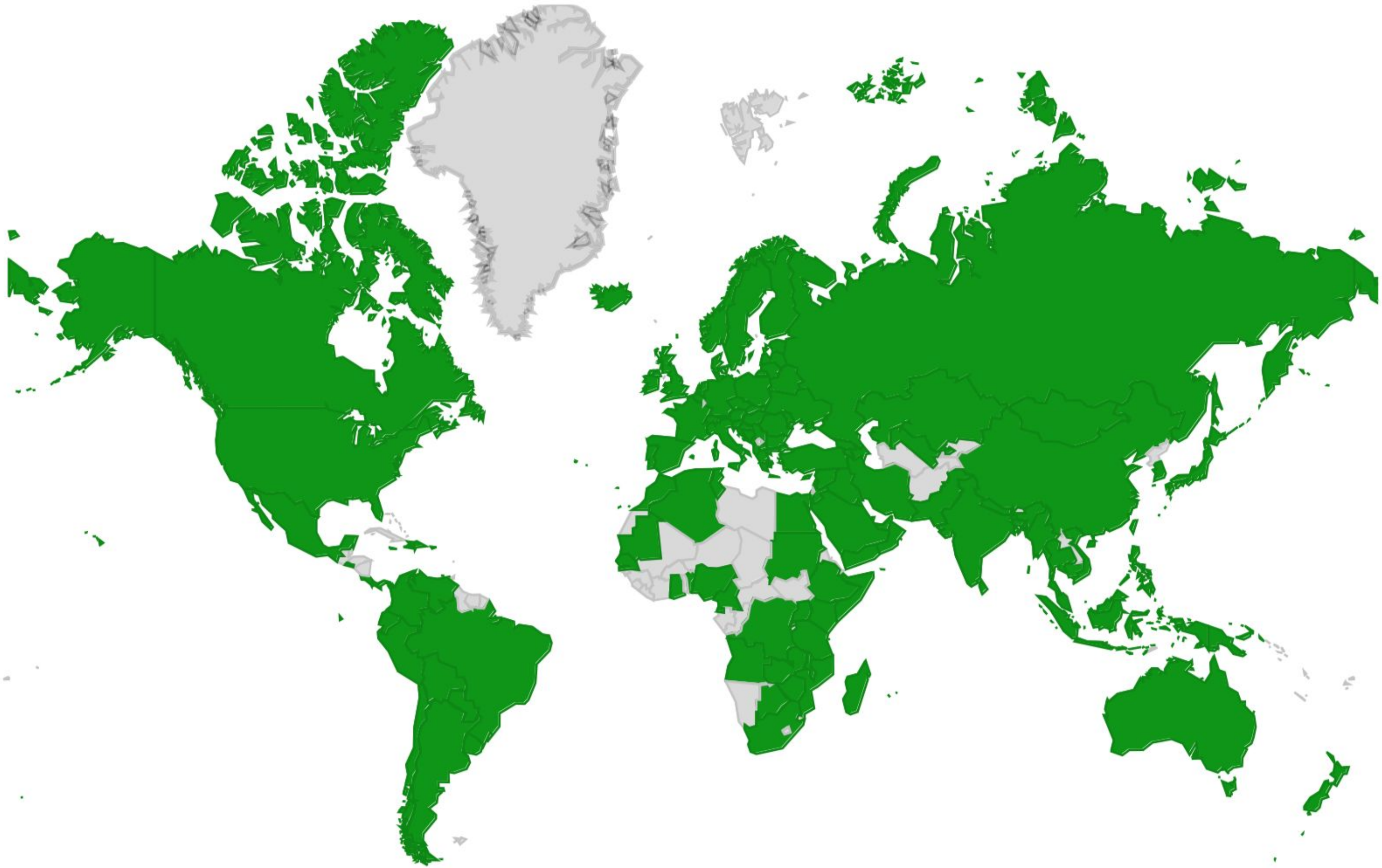


Need for Full-Stack ML Developers


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
Fundamentals of TinyML
Course 1
Neural Network
Filters
Regression
Loss Function
Preprocessing
Data augmentation
Inference
Responsible AI
CNNs/ DNNs
Classification
Gradient Descent
Applications of TinyML
Course 2
Deploying TinyML
Course 3
Yes!
OV7675
QCIF
144
176
person
Keyword Spotting
Visual Wake Words
Gesture Recognition
Managing TinyML
Course 4


TINY MACHINE
LEARNING KIT
ARDUINO
edX
TensorFlow


TensorFlow Lite
IMU
I/O (USB)
Processor
+ Bluetooth
Temperature
+ Humidity
Microphone



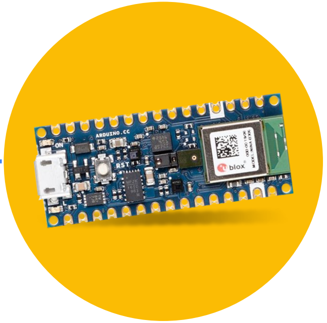
Responsible AI: Human-Centered Design

START → DESIGN → DEVELOPMENT → DEPLOYMENT → END

Course 1
Fundamentals of TinyML

Course 2
Applications of TinyML

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

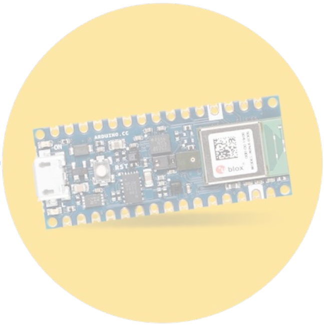
START → **DESIGN** → DEVELOPMENT → DEPLOYMENT → END

Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

START | DESIGN | DEVELOPMENT | DEPLOYMENT | END

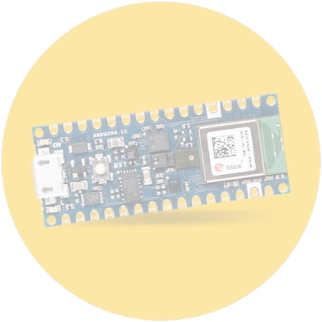
Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

- **What data** will be collected to train the model?
- Is the dataset **biased**?
- How can we **ensure** the model is **fair**?

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

START → DESIGN → DEVELOPMENT → DEPLOYMENT → END

Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

- **What data** will be collected to train the model?
- Is the dataset **biased**?
- How can we **ensure** the model is **fair**?

Course 3
Deploying TinyML

- How will **model drift** be monitored?
- How should **security breaches** be addressed?
- How should the user's **privacy** be protected?

Widening Access to Applied ML

- Broaden the reach of applied AI/ML resources globally
- From the Big Tech & Ivory Tower to the Greater Commons
- Focus on end-to-end ML application development

Widening Access to Applied Machine Learning

Vijay Janapa Reddi, Brian Plancher, Susan Kennedy, Laurence Moroney, Pete Warden, Anant Agarwal, Colby Banbury, Massimo Banzi, Benjamin Brown, Sharad Chitlangia, Radhika Ghosal, Rupert Jaeger, Srivatsan Krishnan, Daniel Leiker, Mark Mazumder, Dominic Pajak, Dhilan Ramaprasad, J. Evan Smith, Matthew Stewart, Dustin Tingley

Harvard University
Google

Abstract

Despite the expanding role of machine learning (ML), most ML resources and experts are in just a few countries and organizations. Broadening access to both computational and educational resources is critical to diffusing ML innovation. We suggest that TinyML, which applies ML to resource-constrained embedded devices, is an attractive means to this end. The required computing hardware is low cost and globally accessible, and it naturally encourages self-contained, end-to-end application development. Future ML engineers must have experience with the entire development process from data collection to deployment, and they must understand the ethical implications of their designs before deploying them. To this end, a collaboration between academia (Harvard University) and industry (Google and Arduino) produced a four-part massive online open course (MOOC) that provides application-driven instruction on the development of end-to-end solutions using TinyML. The course is openly available on the edX platform and has no prerequisites, beyond basic programming and was specifically designed for learners from diverse backgrounds. At the time of this writing, 35,000 learners have enrolled on edX. The first two courses progress from an overview of fundamental ML topics to greater detail on TinyML algorithms and applications. The third and fourth courses delve into ML-model deployment and ML-life-cycle management using microcontroller development boards. The courses introduce pupils to real-world applications, ML algorithms, data-set engineering, and the ethical considerations of these technologies through hands-on programming and deployment of TinyML applications in both the cloud and their own microcontrollers. To facilitate continued learning, community building, and collaboration beyond the course, we launched a standalone website, a Discourse forum, and an optional course-project competition. We also released the course materials publicly. Our hope is that these resources inspire and guide the next generation of ML practitioners and educators as well as further broaden access to cutting-edge ML technologies.

Figure 1: We designed a new applied-ML course motivated by real-world applications, covering not only the software (ML algorithms) and hardware (embedded systems) but also the product life cycle and responsible AI. To make it accessible and scalable, as well as to provide hands-on components, we focused on the emerging TinyML domain and released the course as a MOOC on edX.

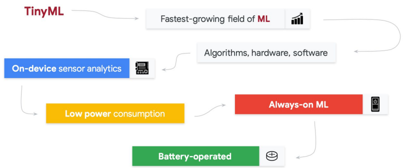
1 Introduction

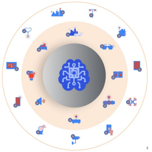
The past two decades have seen dramatic progress in machine learning (ML) from a purely academic discipline to a widespread commercial technology that serves a range of sectors. ML allows developers to improve business processes and human productivity through data-driven automation. Given applied ML's ubiquity and success, its commercial use should only increase. Existing ML applications cover a wide spectrum that includes digital assistants [1, 2], autonomous vehicles [3, 4], robotics [5], health care [6], transportation [7, 8], and security [9], education [10, 11], etc. New use cases are rapidly emerging, every few days there is a new ML use case.

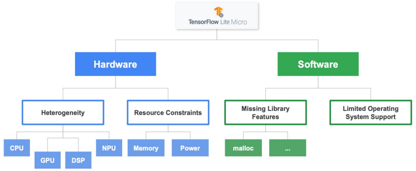
The mass proliferation of this technology and associated jobs have great potential to improve society and uncover new opportunities for technological innovation, societal prosperity, and individual growth. But it all rests on the assumption that everyone, globally, has unfettered access to ML technologies, which isn't the case.

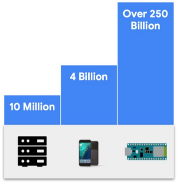
Widening access to applied ML faces three challenges. First is a shortage of ML educators at all levels [12, 13]. Second is insufficient resources to run ML models, especially as data sets continue to balloon. Training and running ML models often requires costly, high-performance hardware. Third is a growing gap between industry and academia, as even the best academic institutions and research labs struggle to keep pace with change. Addressing these critical issues requires innovative education and workforce training to prepare the next generation of applied ML engineers.

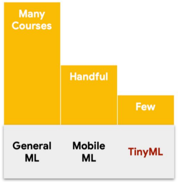
This paper presents a pedagogical approach, developed as an academic/industry collaboration led by Harvard University and Google, to address these challenges and thereby widen access to applied ML. We employ both cloud computing and low-cost hardware. Specifically, we use Google's free, open-source TensorFlow

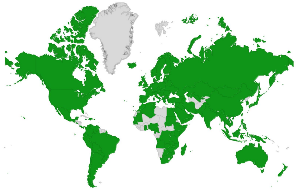












Conclusion

- **Why AI is going tiny**
  - BLERP
- **How it can change the world**
  - Unlocking real-time AI
  - AI for Social Good
- **What shrinks it**
  - Challenges in terms of “Code” and “Data”

The Future of AI is Tiny and Bright

Challenges & Opportunities

Vijay Janapa Reddi
Harvard University