Embedded Inference Serving Benchmarks

Model serving

Online inference

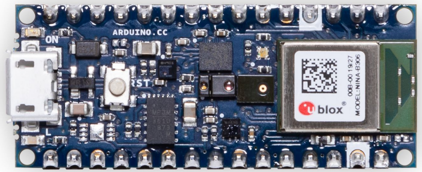
Batch inference

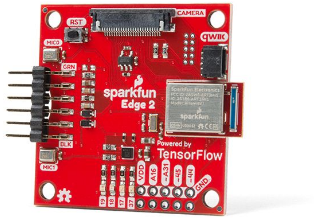
Streaming inference

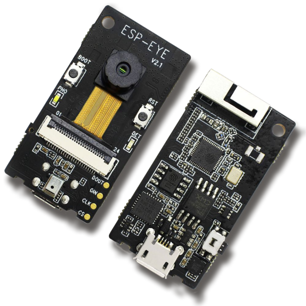
Embedded inference

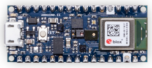
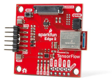
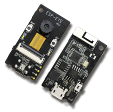
Explain predictions

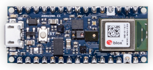
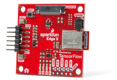
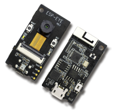

Himax

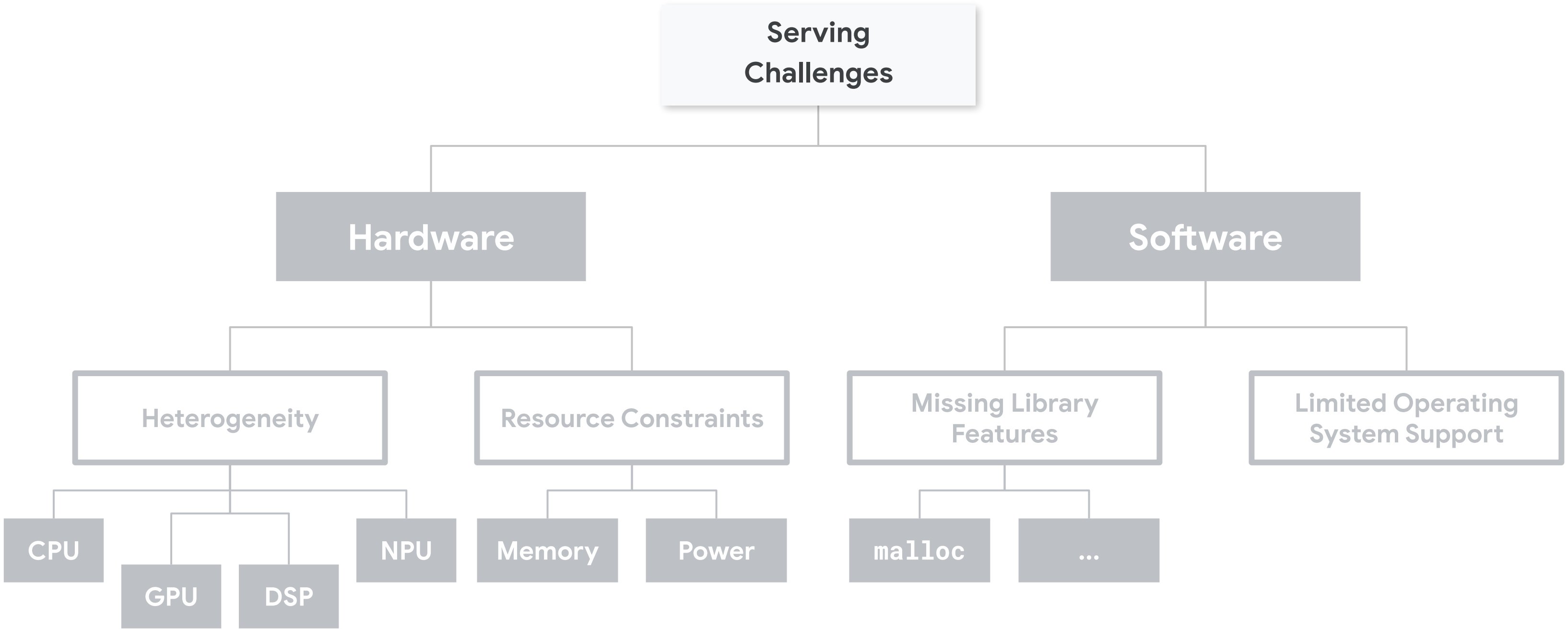

ARDUINO.CC
blox

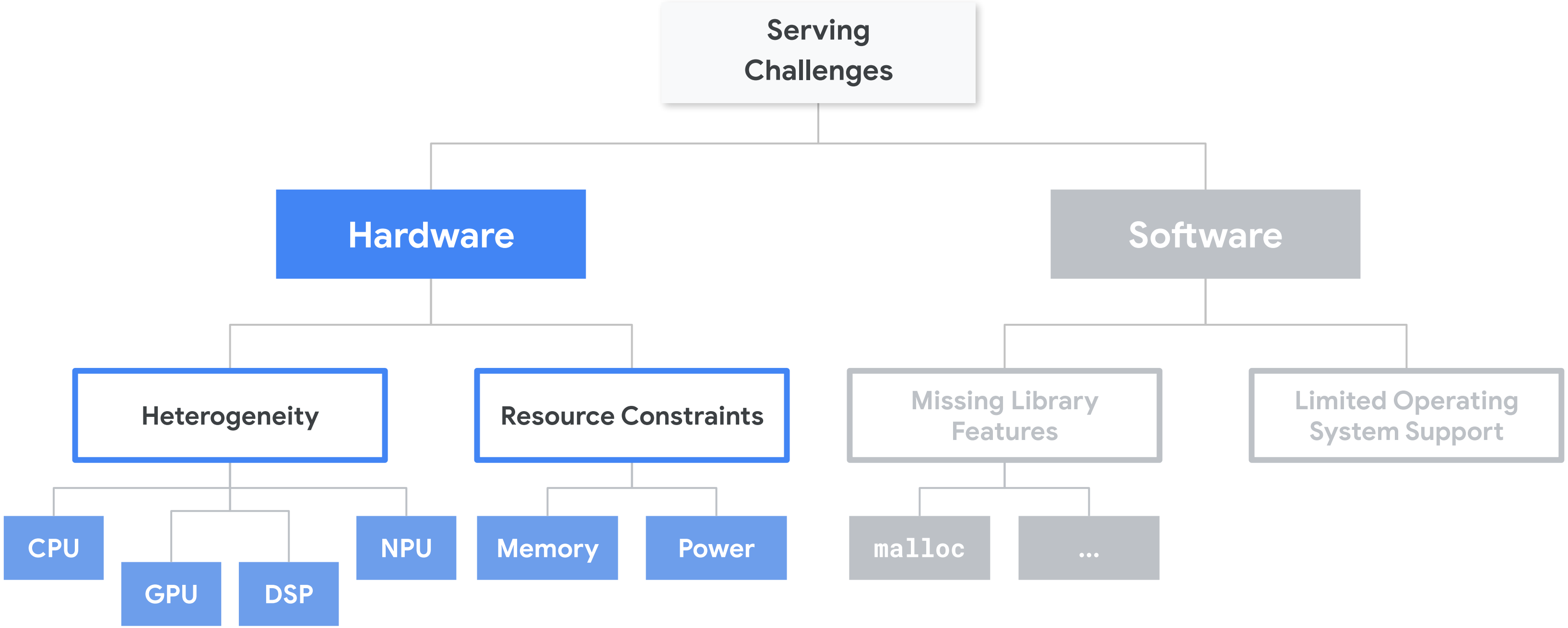

sparkfun
Edge 2
Powered by
TensorFlow

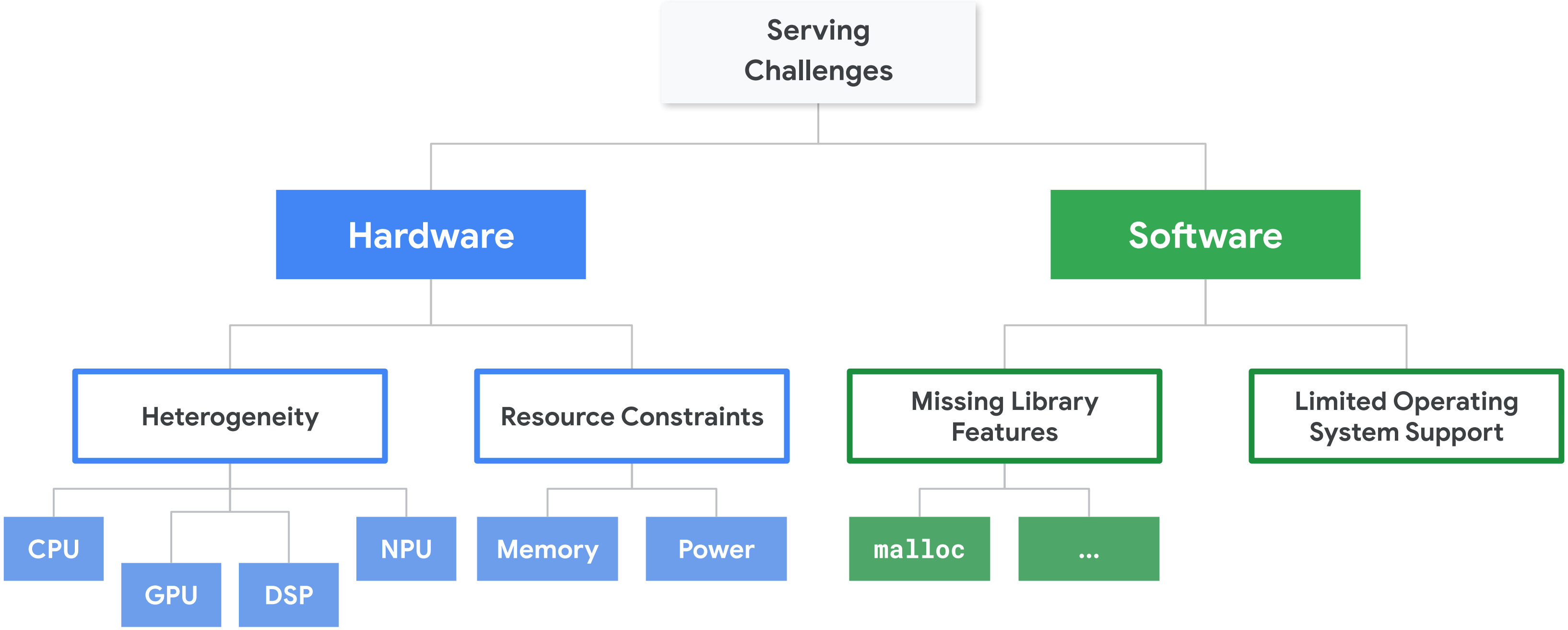

ESP-EYE

	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

	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

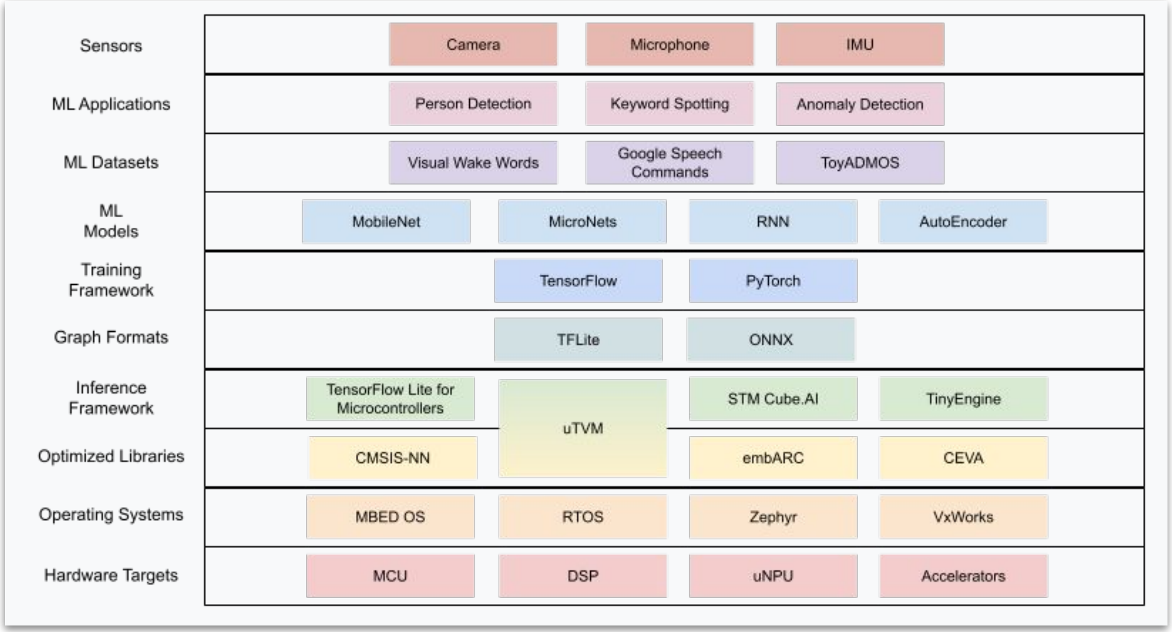

Serving Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


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


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

TinyML System Stack is Complicated

- Machine learning system stack is **complicated**
- Many **different** models, datasets, models, frameworks, formats, compilers, libraries, operating systems, targets
- The **cross-product** makes it challenging to decipher system performance



Apples-to-apples comparison

What task?
What model?
What dataset?
What batch size?
What quantization?
What software libraries?
...

bench·mark

/ˈben(t)SHmärk/

See definitions in:

All | Technology | Surveying

noun

1. a standard or point of reference against which things may be compared or assessed.
   "a benchmark case"
   Similar: standard | point of reference | basis | gauge | criterion | specification
2. a surveyor's mark cut in a wall, pillar, or building and used as a reference point in measuring altitudes.

verb

evaluate or check (something) by comparison with a standard.
"we are **benchmarking** our performance **against** external criteria"

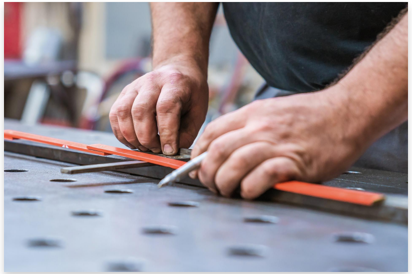
Definitions from Oxford Languages

Feedback

Benchmarking

Use to

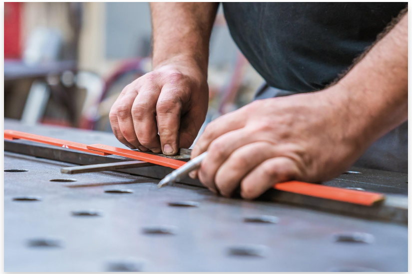
- **Compare** solutions



Benchmarking

Use to

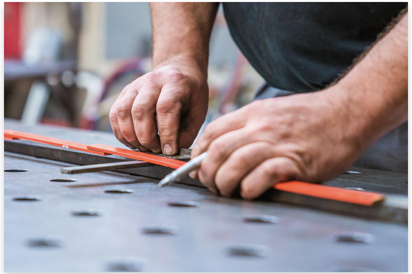
- **Compare** solutions
- **Inform** selection



Benchmarking

Use to

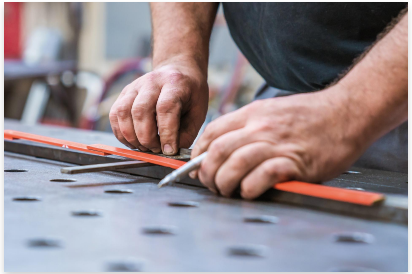
- **Compare** solutions
- **Inform** selection
- **Measure** and track progress



Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar**, advance** the field



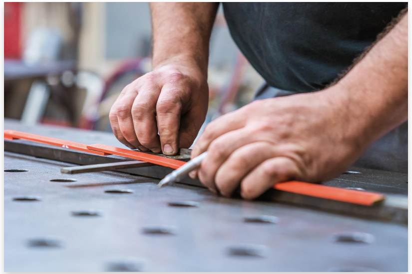
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous



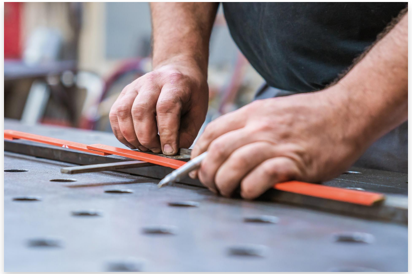
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



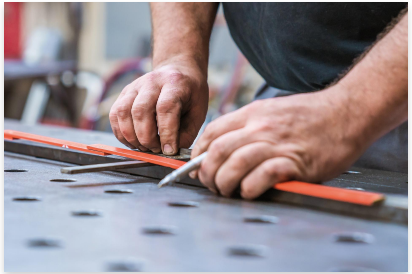
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar**, advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



Provides

- **Standardization** of use cases and workloads

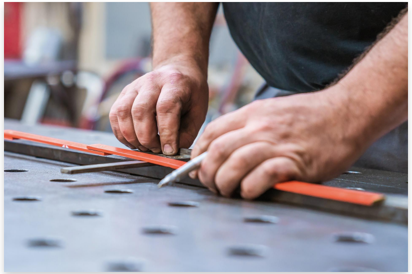
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



Provides

- **Standardization** of use cases and workloads
- **Comparability** across heterogeneous HW/SW systems

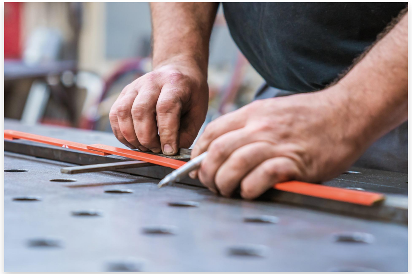
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



Provides

- **Standardization** of use cases and workloads
- **Comparability** across heterogeneous HW/SW systems
- **Complex characterization** of system compromises

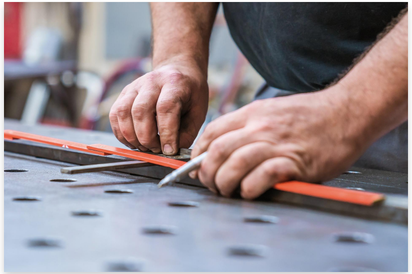
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



Provides

- **Standardization** of use cases and workloads
- **Comparability** across heterogeneous HW/SW systems
- **Complex characterization** of system compromises
- **Verifiable and Reproducible** results

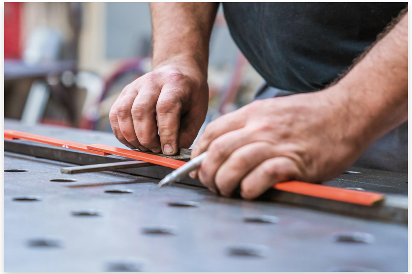
Benchmarking

Use to

- **Compare** solutions
- **Inform** selection
- **Measure** and track progress
- **Raise** the bar, **advance** the field

Requires

- **Methodology** that is both fair and rigorous
- **Community** support and consensus



Provides

- **Standardization** of use cases and workloads
- **Comparability** across heterogeneous HW/SW systems
- **Complex characterization** of system compromises
- **Verifiable and Reproducible** results

MLPerf

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

Encourage innovation to improve the state-of-the-art of ML

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

Encourage innovation to improve the state-of-the-art of ML

Accelerate progress in ML via **fair and useful measurement**

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

Encourage innovation to improve the state-of-the-art of ML

Accelerate progress in ML via **fair and useful measurement**

Serve both the **commercial and research communities**

Goals

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

Encourage innovation to improve the state-of-the-art of ML

Accelerate progress in ML via **fair and useful measurement**

Serve both the **commercial and research communities**

Keep **benchmarking affordable** so that all can participate

Wide Array of ML Tasks

Task Category	Use Case
Audio	Audio Wake Words Context Recognition Control Words Keyword Detection
Image	Visual Wake Words Object Detection Gesture Recognition Object Counting Text Recognition
Physiological / Behavioral Metrics	Segmentation Anomaly Detection Forecasting Activity Detection
Industry Telemetry	Sensing Predictive Maintenance Motor Control

Wide Array of ML Tasks

Task Category	Use Case	Model Type
Audio	Audio Wake Words Context Recognition Control Words Keyword Detection	DNN CNN RNN LSTM
Image	Visual Wake Words Object Detection Gesture Recognition Object Counting Text Recognition	DNN CNN SVM Decision Tree KNN Linear
Physiological / Behavioral Metrics	Segmentation Anomaly Detection Forecasting Activity Detection	DNN Decision Tree SVM Linear
Industry Telemetry	Sensing Predictive Maintenance Motor Control	DNN Decision Tree SVM Linear Naive Bayes

Wide Array of ML Tasks

Task Category	Use Case	Model Type	Datasets
Audio	Audio Wake Words Context Recognition Control Words Keyword Detection	DNN CNN RNN LSTM	Speech Commands Audioset ExtraSensory Freesound DCASE
Image	Visual Wake Words Object Detection Gesture Recognition Object Counting Text Recognition	DNN CNN SVM Decision Tree KNN Linear	Visual Wake Words CIFAR10 MNIST ImageNet DVS128 Gesture
Physiological / Behavioral Metrics	Segmentation Anomaly Detection Forecasting Activity Detection	DNN Decision Tree SVM Linear	Physionet HAR DSA Opportunity
Industry Telemetry	Sensing Predictive Maintenance Motor Control	DNN Decision Tree SVM Linear Naive Bayes	UCI Air Quality UCI Gas UCI EMG NASA's PCoE

A Principled Approach to Subsetting

Big Questions	**Inference**
1. Benchmark definition	What is the definition of a benchmark task?

A Principled Approach to Subsetting

Big Questions	**Inference**
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?

A Principled Approach to Subsetting

Big Questions	**Inference**
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?
3. Metric definition	What is the measure of “performance” in ML systems?

A Principled Approach to Subsetting

Big Questions	Inference
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?
3. Metric definition	What is the measure of "performance" in ML systems?
4. Implementation equivalence	How do submitters run on different hardware/software systems?

A Principled Approach to Subsetting

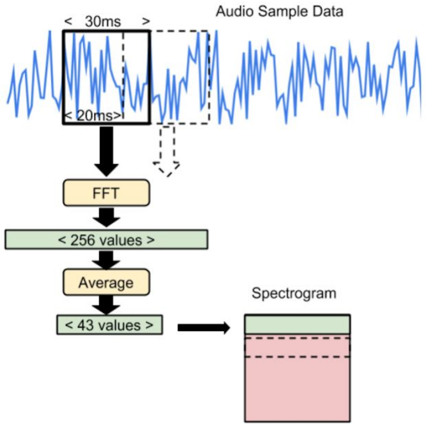
Big Questions	**Inference**
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?
3. Metric definition	What is the measure of “performance” in ML systems?
4. Implementation equivalence	How do submitters run on different hardware/software systems?
5. Issues with optimizations	Quantization, calibration, and/or retraining?

A Principled Approach to Subsetting

Big Questions	**Inference**
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?
3. Metric definition	What is the measure of “performance” in ML systems?
4. Implementation equivalence	How do submitters run on different hardware/software systems?
5. Issues with optimizations	Quantization, calibration, and/or retraining?
6. Results	Do we normalize and/or summarize results?

MLPerf "Tiny" Tasks

Keyword Spotting



Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." *arXiv preprint arXiv:1804.03209* (2018).

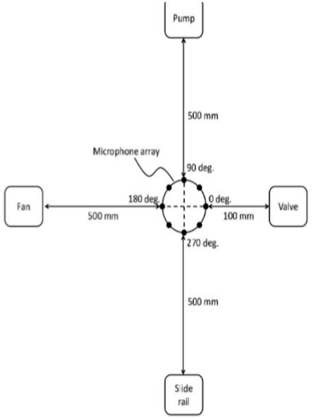
Visual Wake Words

(a) 'Person'

(b) 'Not-person'

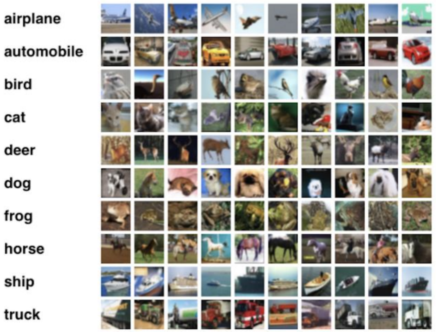
Chowdhery, Aakanksha, et al. "Visual wake words dataset." *arXiv preprint arXiv:1906.05721* (2019).

Anomaly Detection

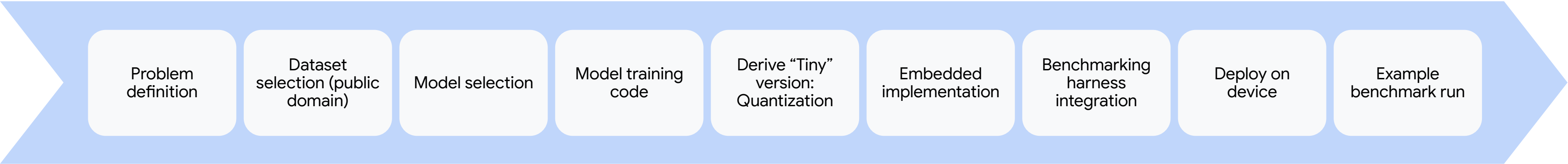


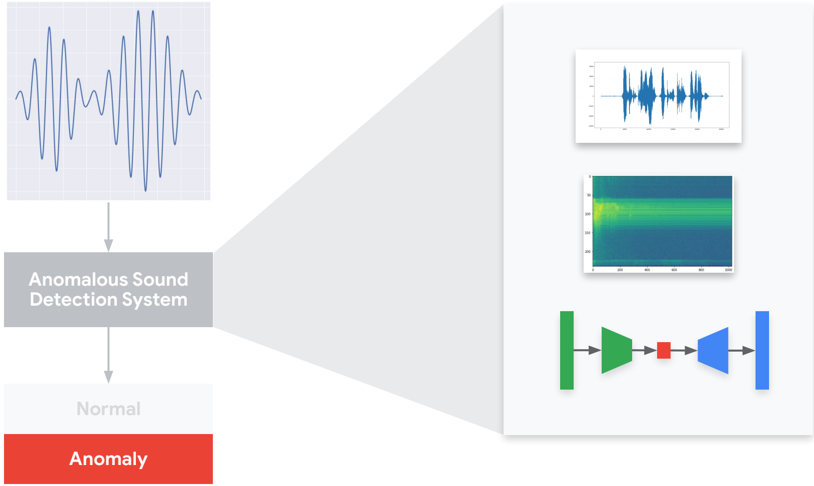
Purohit, Harsh, et al. "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection." *arXiv preprint arXiv:1909.09347* (2019).

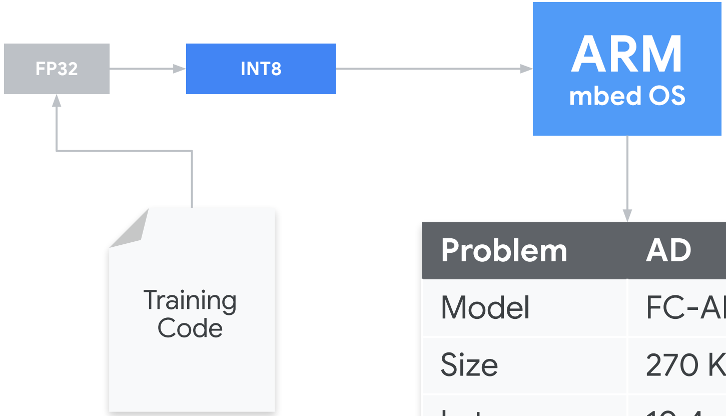
Tiny Image Classification



Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.



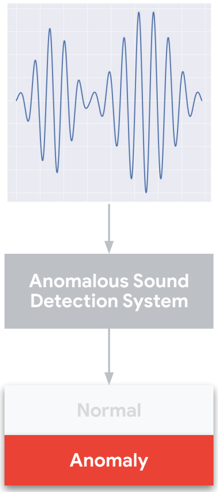


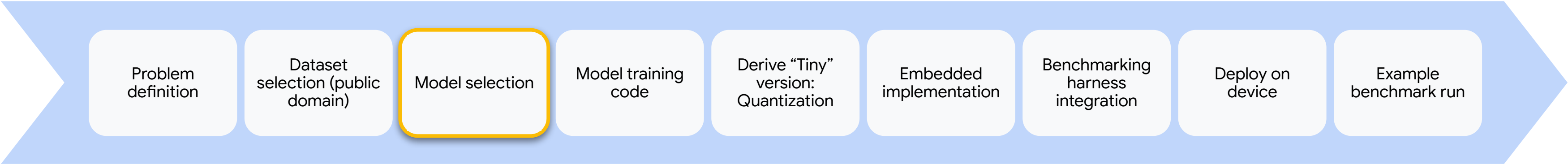


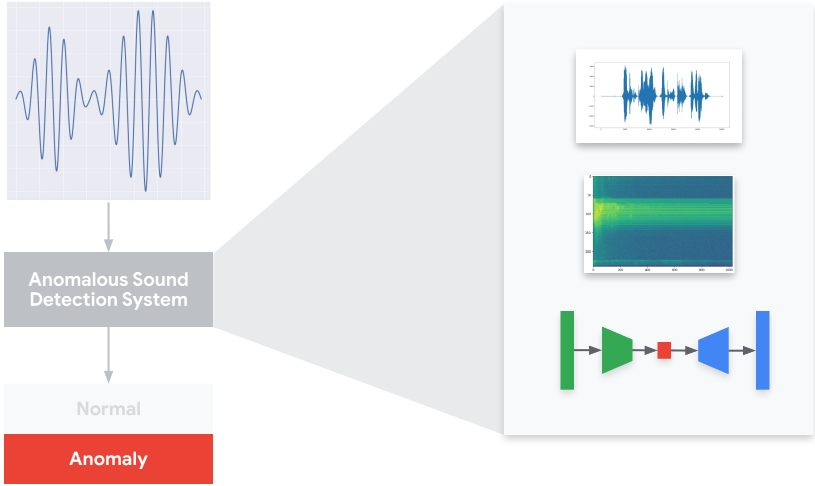
Problem	AD
Model	FC-AE
Size	270 Kpar
Latency	10.4 ms/inf.
Accuracy	.86 AUC
Energy	516 µJ/inf.

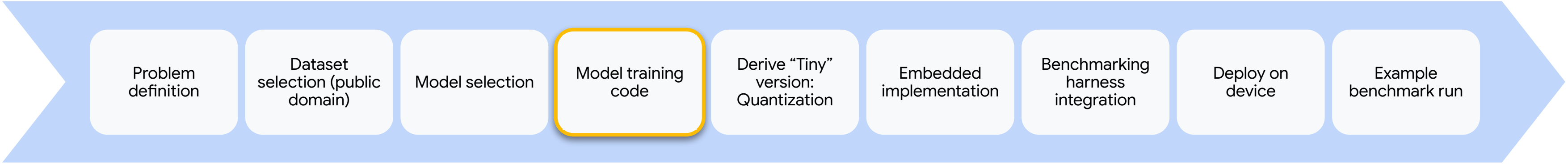
- Problem definition
- Dataset selection (public domain)
- Model selection
- Model training code
- Derive “Tiny” version: Quantization
- Embedded implementation
- Benchmarking harness integration
- Deploy on device
- Example benchmark run

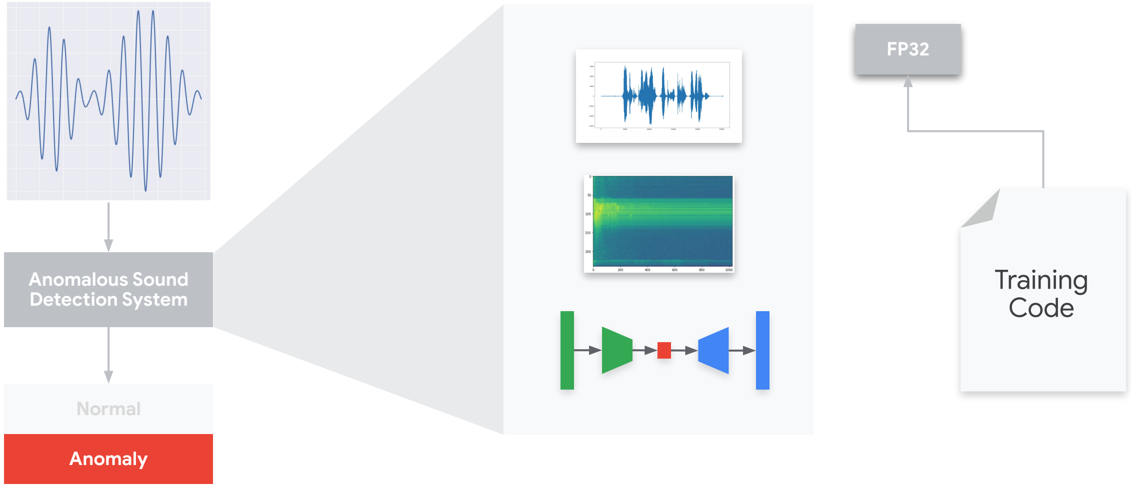
Problem definition | **Dataset selection (public domain)** | Model selection | Model training code | Derive “Tiny” version: Quantization | Embedded implementation | Benchmarking harness integration | Deploy on device | Example benchmark run



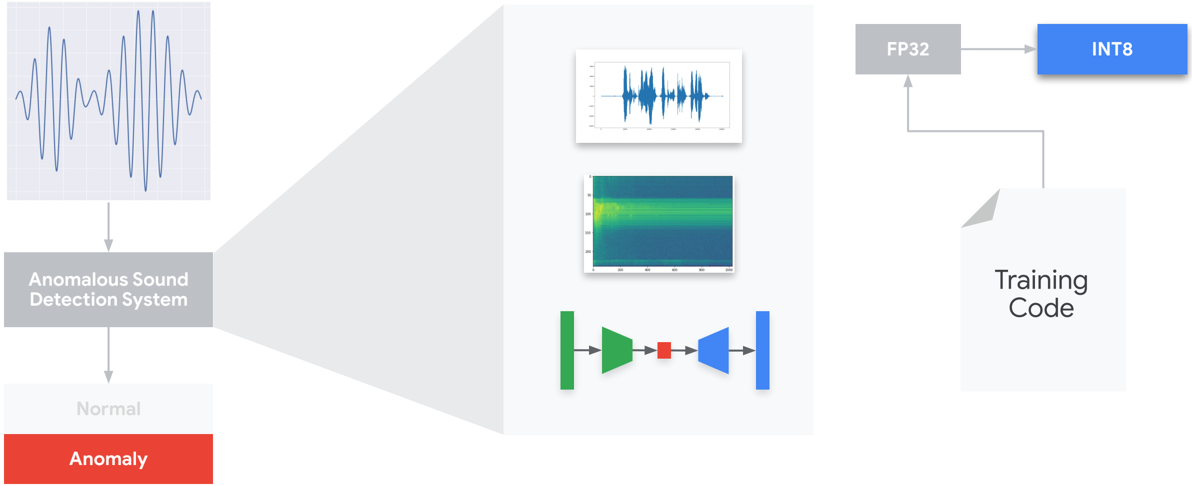

Problem definition
Dataset selection (public domain)
Model selection
Model training code
Derive "Tiny" version: Quantization
Embedded implementation
Benchmarking harness integration
Deploy on device
Example benchmark run


Anomalous Sound Detection System
Normal
Anomaly

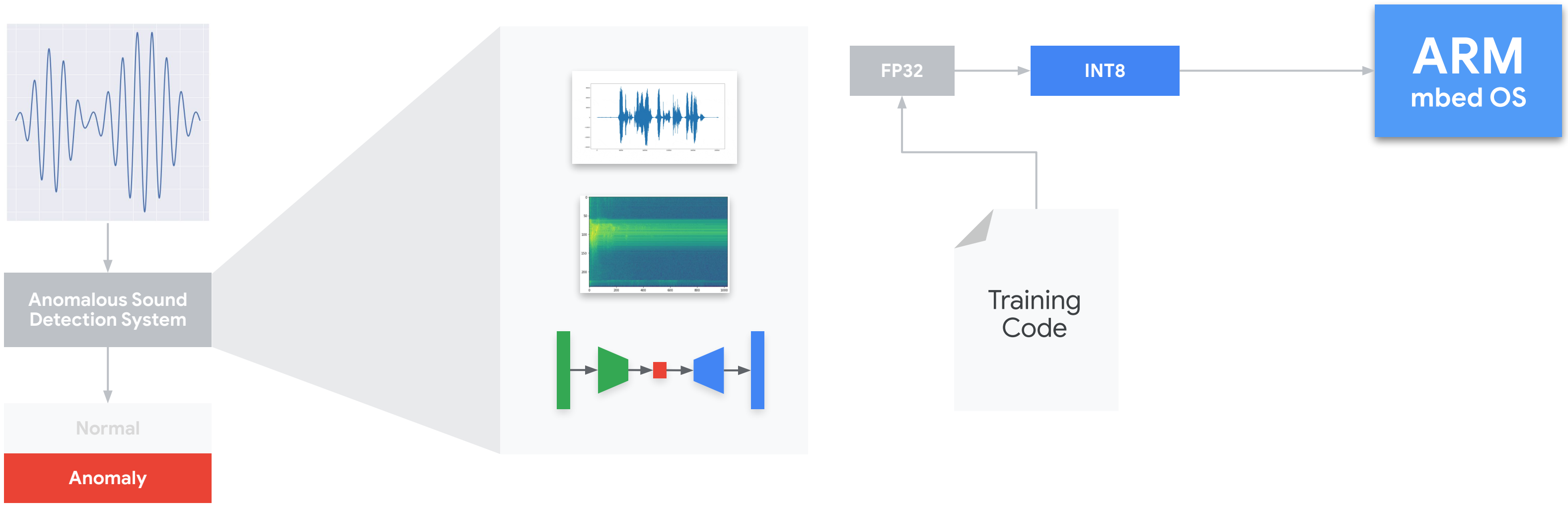

Problem definition
Dataset selection (public domain)
Model selection
Model training code
Derive “Tiny” version: Quantization
Embedded implementation
Benchmarking harness integration
Deploy on device
Example benchmark run

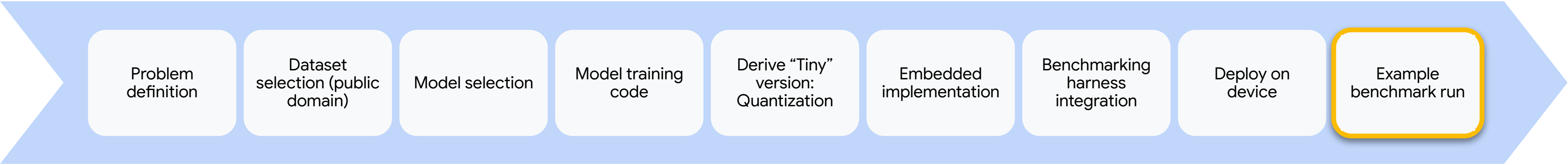

Anomalous Sound Detection System
Normal
Anomaly
FP32
Training Code

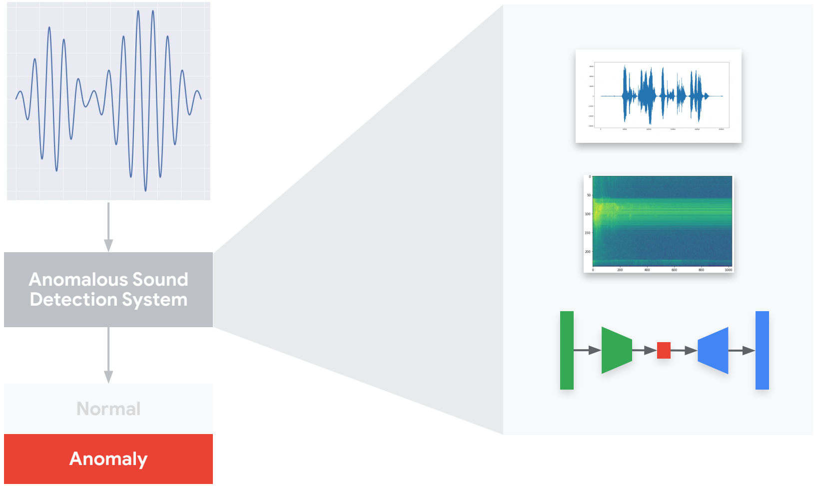

Problem definition
Dataset selection (public domain)
Model selection
Model training code
Derive “Tiny” version: Quantization
Embedded implementation
Benchmarking harness integration
Deploy on device
Example benchmark run

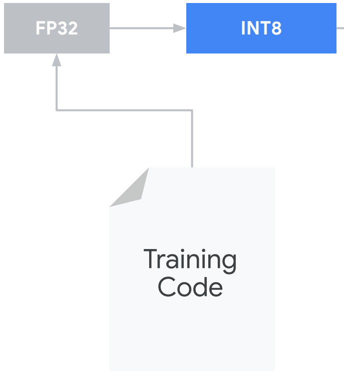

Anomalous Sound Detection System
Normal
Anomaly
FP32
INT8
Training Code

- Problem definition
- Dataset selection (public domain)
- Model selection
- Model training code
- Derive "Tiny" version: Quantization
- Embedded implementation
- Benchmarking harness integration
- Deploy on device
- Example benchmark run









Problem	AD
Model	FC-AE
Size	270 Kpar
Latency	10.4 ms/inf.
Accuracy	.86 AUC
Energy	516 µJ/inf.

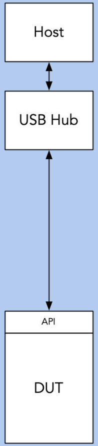
Metrics

Latency

Small fast dataset

Loop of inferences

No data-dependent execution



a.

```
Runtime requirements have been met.
Performance results for window 10:
  # Inferences :          1000
  Runtime      :        10.524 sec.
  Throughput   :        95.020 inf./sec.
Runtime requirements have been met.
-----------------------------------------
Median throughput is 95.019 inf./sec.
-----------------------------------------
```

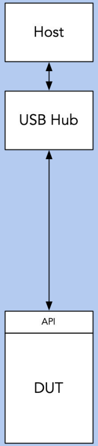
Metrics

Latency

Small fast dataset

Loop of inferences

No data-dependent execution



a.

```
Runtime requirements have been met.
Performance results for window 10:
  # Inferences :         1000
  Runtime      :       10.524 sec.
  Throughput   :       95.020 inf./sec.
Runtime requirements have been met.
-----------------------------------------
Median throughput is 95.019 inf./sec.
-----------------------------------------
```

Accuracy

Evaluate on larger dataset

Top-1 accuracy & AUC

CLOSED: meet threshold
v.
OPEN: part of the metrics

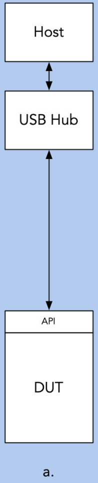
Metrics

Latency

Small fast dataset

Loop of inferences

No data-dependent execution




```
Runtime requirements have been met.
Performance results for window 10:
  # Inferences :          1000
  Runtime      :        10.524 sec.
  Throughput   :        95.020 inf./sec.
Runtime requirements have been met.
------------------------------------------
Median throughput is 95.019 inf./sec.
------------------------------------------
```

Accuracy

Evaluate on larger dataset

Top-1 accuracy & AUC

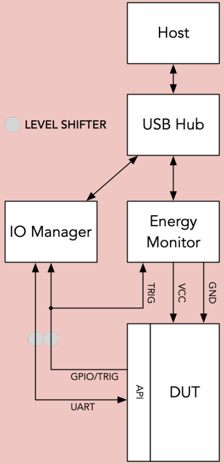
CLOSED: meet threshold
v.
OPEN: part of the metrics

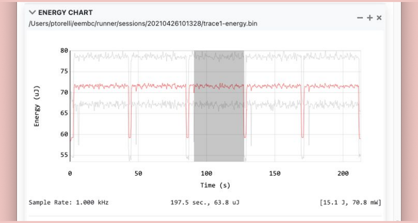
Energy

No "cherry-picking"

Power Monitor setup

Median result





Emerging TinyML Use Cases

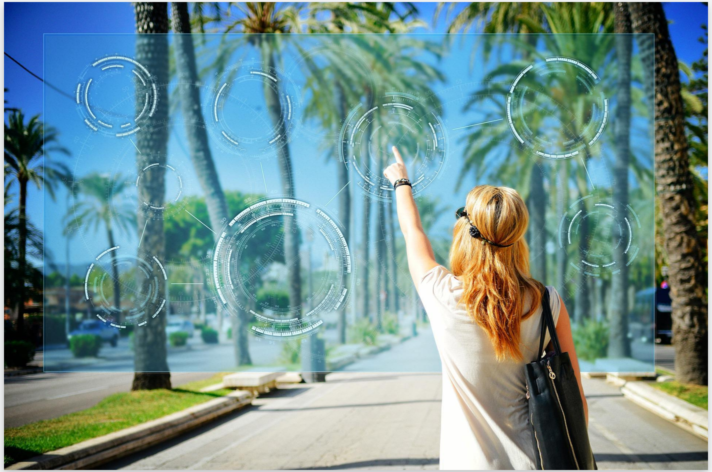
Example: Smart shoes

- Kicking
- Penalty kicking
- Passing
- Dribbling
- ...

Emerging TinyML Use Cases

Example: Augmented Reality

- Eye tracking
- Hand tracking
- Computer vision
- Superresolution
- ...



Toward Emerging Multi-DNN Models

Pipelined DNNs

Keyword Spotting

Speech Processing

- **Back-to-back execution**
- **Execution dependency**

Toward Emerging Multi-DNN Models

Pipelined DNNs

Keyword Spotting | Speech Processing

- Back-to-back execution
- Execution dependency

Concurrent DNNs

Eye Tracking | Obstacle Detection | Video Processing

- Concurrent execution
- Execution deadline

Toward Emerging Multi-DNN Models

Pipelined DNNs

Keyword Spotting

Speech Processing

- Back-to-back execution
- Execution dependency

Concurrent DNNs

Eye Tracking

Obstacle Detection

Video Processing

- Concurrent execution
- Execution deadline

Concurrent & Pipelined DNNs

Obstacle Detection

Eye Tracking → Foveated Rendering

- Challenges from both pipelined and concurrent

MLPerf

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

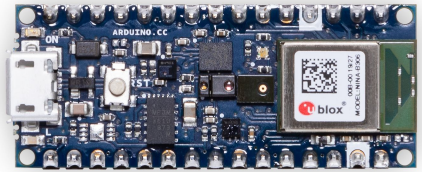
Encourage innovation to improve the state-of-the-art of ML

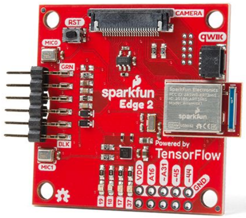
Accelerate progress in ML via **fair and useful measurement**

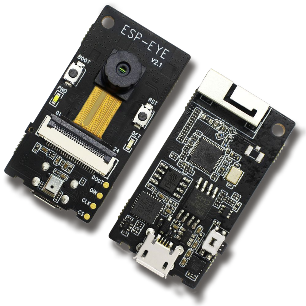
Serve both the **commercial and research communities**

Keep **benchmarking affordable** so that all can participate


Himax


ARDUINO.CC
blox


sparkfun
Edge 2
Powered by
TensorFlow


ESP-EYE