arm AI

AI Virtual Tech Talks Series

Machine learning for embedded systems at the edge

Arm and NXP

Kobus Marneweck, Product Manager, Arm
Anthony Huereca, Systems Engineer, NXP

June 16, 2020

AI VIRTUAL TECH TALKS SERIES

Date	Title	Host
Today	Machine learning for embedded systems at the edge	Arm and NXP
June, 30	tinyML development with Tensorflow Lite for Microcontrollers and CMSIS-NN	Arm
July, 14	Demystify artificial intelligence on Arm MCUs	Cartesiam.ai
July, 28	Speech recognition on Arm Cortex-M	Fluent.ai
August, 11	Getting started with Arm Cortex-M software development and Arm Development Studio	Arm
August, 25	Efficient ML across Arm from Cortex-M to Web Assembly	Edge Impulse

Visit: developer.arm.com/solutions/machine-learning-on-arm/ai-virtual-tech-talks

SPEAKERS

Kobus Marneweck, Senior Product Manager
Arm

Anthony Huereca, Embedded Systems Engineer
NXP Semiconductor

AGENDA

- ML on the edge
- eIQ deployment
  - Arm support for TFLμ
  - TensorFlow
  - Glow
  - Getting started
- The future
- Wrap-up

Machine Learning on the Edge

SECURE CONNECTIONS
FOR A SMARTER WORLD

EXAMPLE EMBEDDED AI APPLICATIONS

Image Classification

- Identify what camera is looking at
  - Coffee pods
  - Empty vs full trucks
  - Factory defects on manufacturing line
  - Produce on supermarket scale
- Personalization based on facial recognition
  - Appliances
  - Home
  - Toys
  - Auto
- Security Video Analysis

Audio Analysis

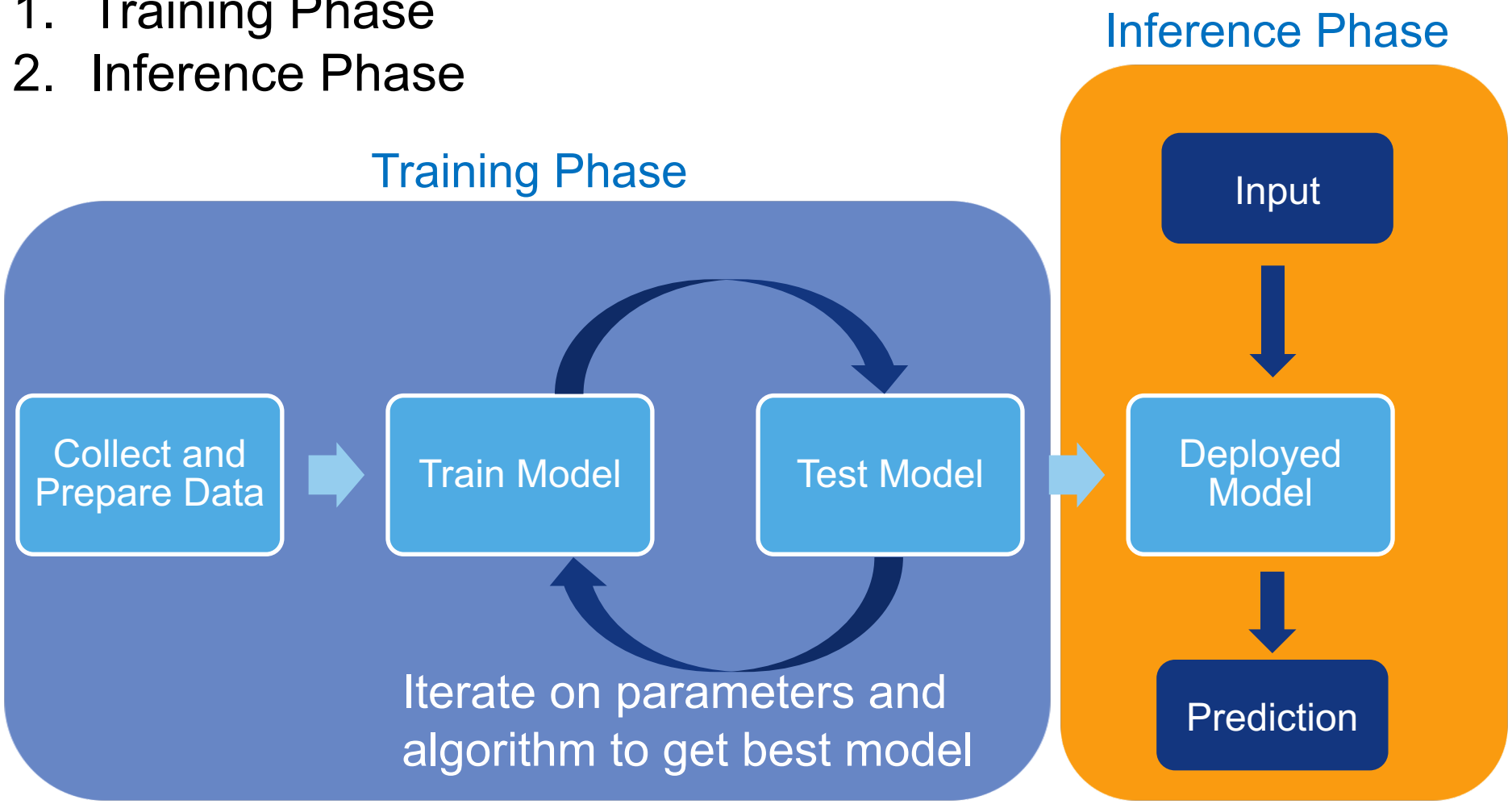
- Keyword actions
  - “Alexa”/“Hey Google”
- Voice commands
- Alarm Analytics
  - Breaking glass
  - Crying baby

Anomaly Detection

- Identify factory issues before they become catastrophic
- Smartwatch health monitoring
- Motor performance monitoring
- Sensor Analysis

MACHINE LEARNING PROCESS

1. Training Phase
2. Inference Phase



INFERENCE ON THE EDGE

- Inference is using a model to make a prediction on new data
- Data can come from embedded camera, microphone, or sensors

Two possibilities:

Inference on the Cloud

- Requires network bandwidth
- Latency issues
- Cloud computing costs

Inference on the Edge

- **Increased privacy and security**
- Faster response time and throughput
- Lower Power
- Don’t need internet connectivity

NXP Enablement for Machine Learning

SECURE CONNECTIONS
FOR A SMARTER WORLD

NXP BROAD-BASED MACHINE LEARNING SOLUTIONS AND SUPPORT

eIQ™ ML Enablement

- eIQ (edge intelligence) for edge AI/ML inference enablement
- Based on open source technologies (TensorFlow Lite, Arm NN, Glow, ONNX, OpenCV)
- Support for i.MX 8 family, i.MX RT1050/1060/600
- Fully integrated into NXP development environments (MCUXpresso, Yocto/Linux)
- BYOM – Bring Your Own Model

DIY

Third Party SW and HW

- Coral Dev Board
- i.MX 8M Development Kit for Amazon® Alexa Voice Service w/ DSP Concepts
- Au-Zone Network Development Tools
- Arcturus video applications
- SensiML tools for sensor analysis

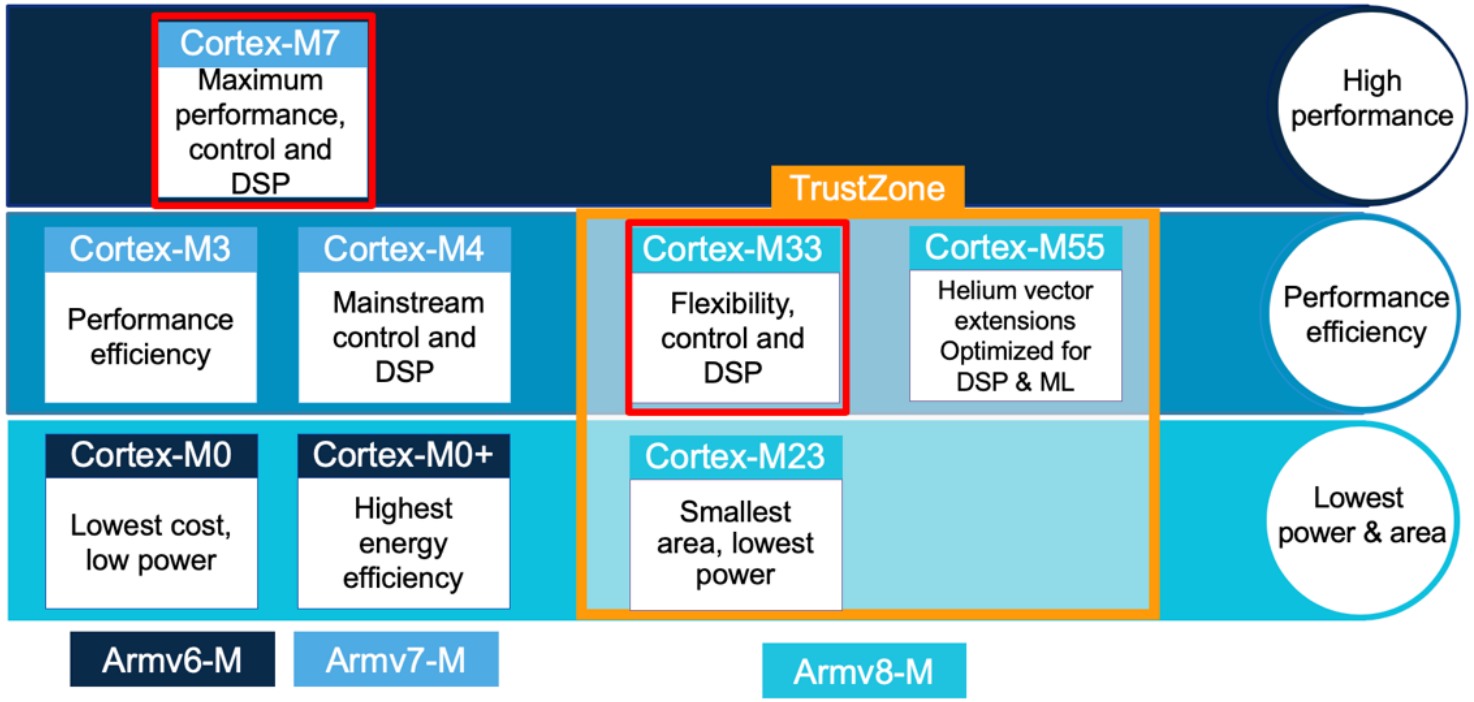
.... And more

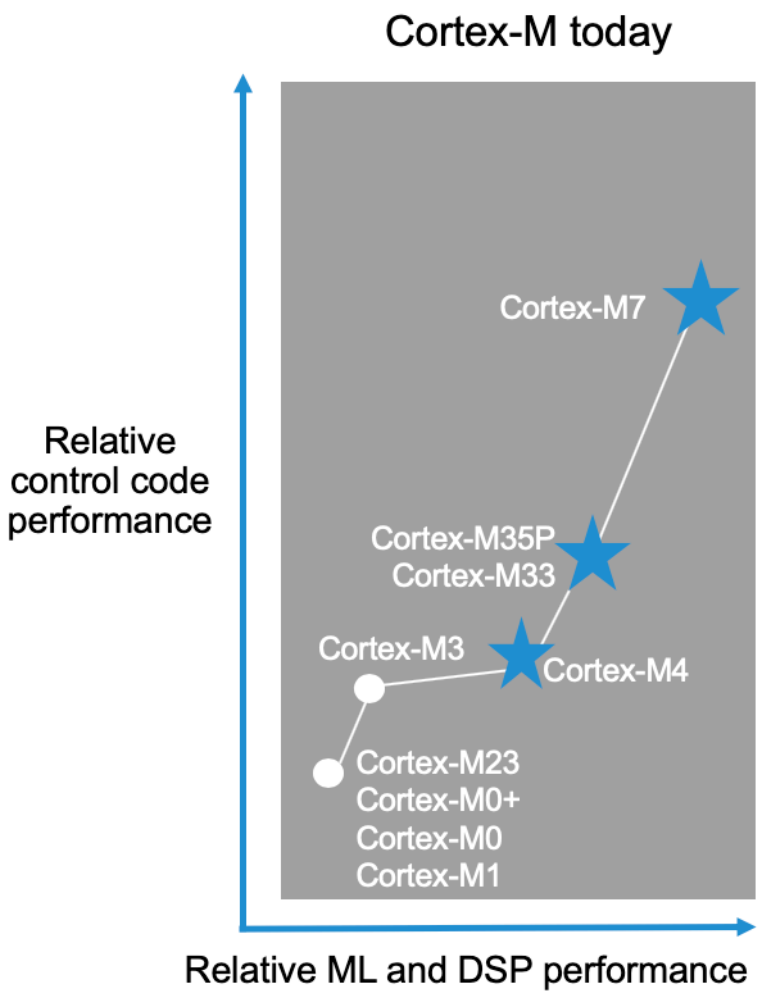
Turnkey Solutions

- Alexa Voice Services (AVS) solution
  - i.MX RT106A (kit – SLN-ALEXA-IOT)
- Local voice control solution
  - i.MX RT106L (kit – SLN-LOCAL-IOT)
- Face & emotion recognition solution
  - i.MX RT106F (kit – SLN-VIZN-IOT)

Fully Tested

ARM CORTEX-M PORTFOLIO





Well suited for ML & DSP applications

CORTEX-M7: HIGHEST PERFORMANCE CORTEX-M

High performance – dual-issue processor

- Achieves 2.14 DMIPS/MHz, 5.01 CoreMark/MHz
- Achieves 1.4GHz in 16FFC (typ config with caches and FPU)

Retains all of the Cortex-M benefits

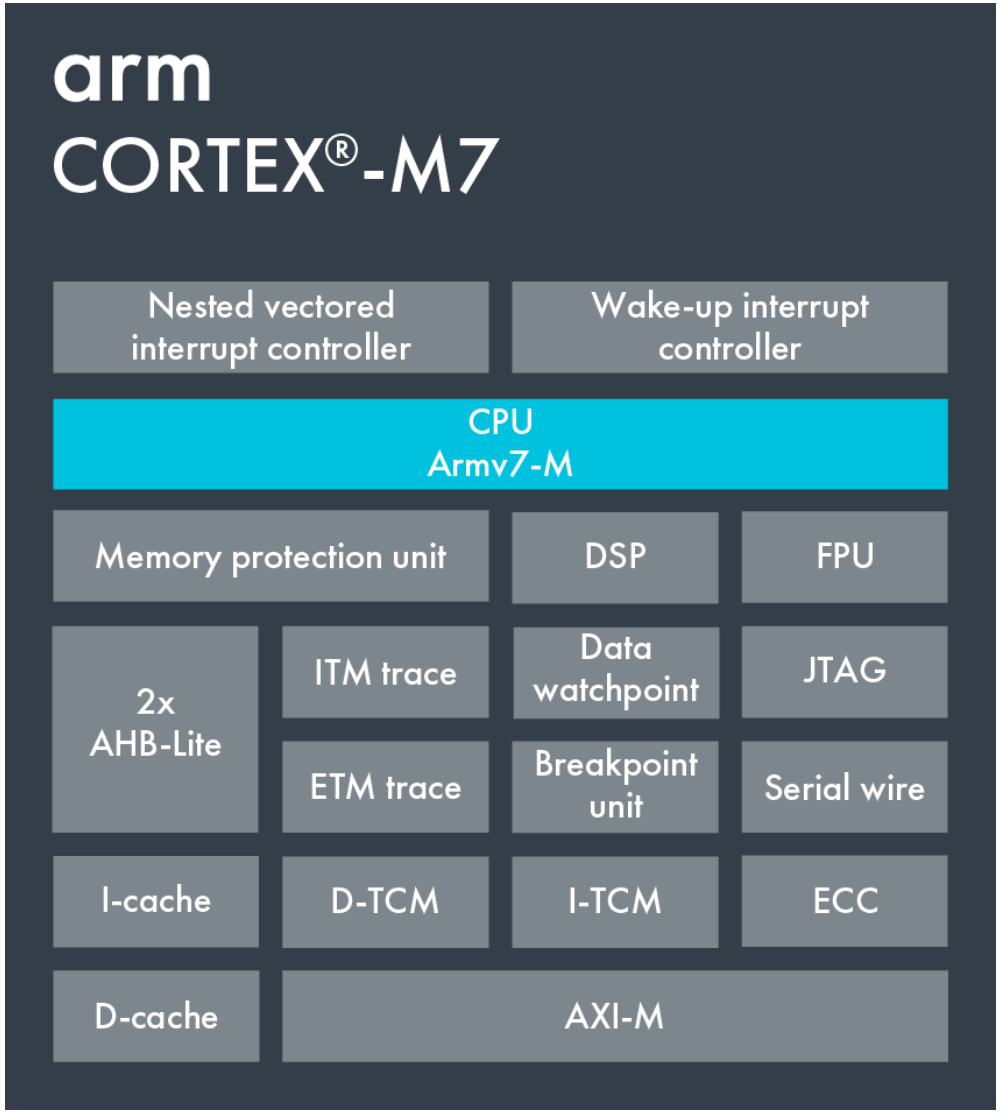
- Ease-of-use, low interrupt latency

Flexible memory interfaces

- Up to 16MB TCM for critical data and code
- Up to 64KB I-cache and D-cache
- AXI master interface

Performance

- Floating-point Unit (FPU) – Single precision (SP) and double precision (DP), sustained 2x 32bit or 2x 16bit MACs per cycle
- Digital signal processing (DSP) extension



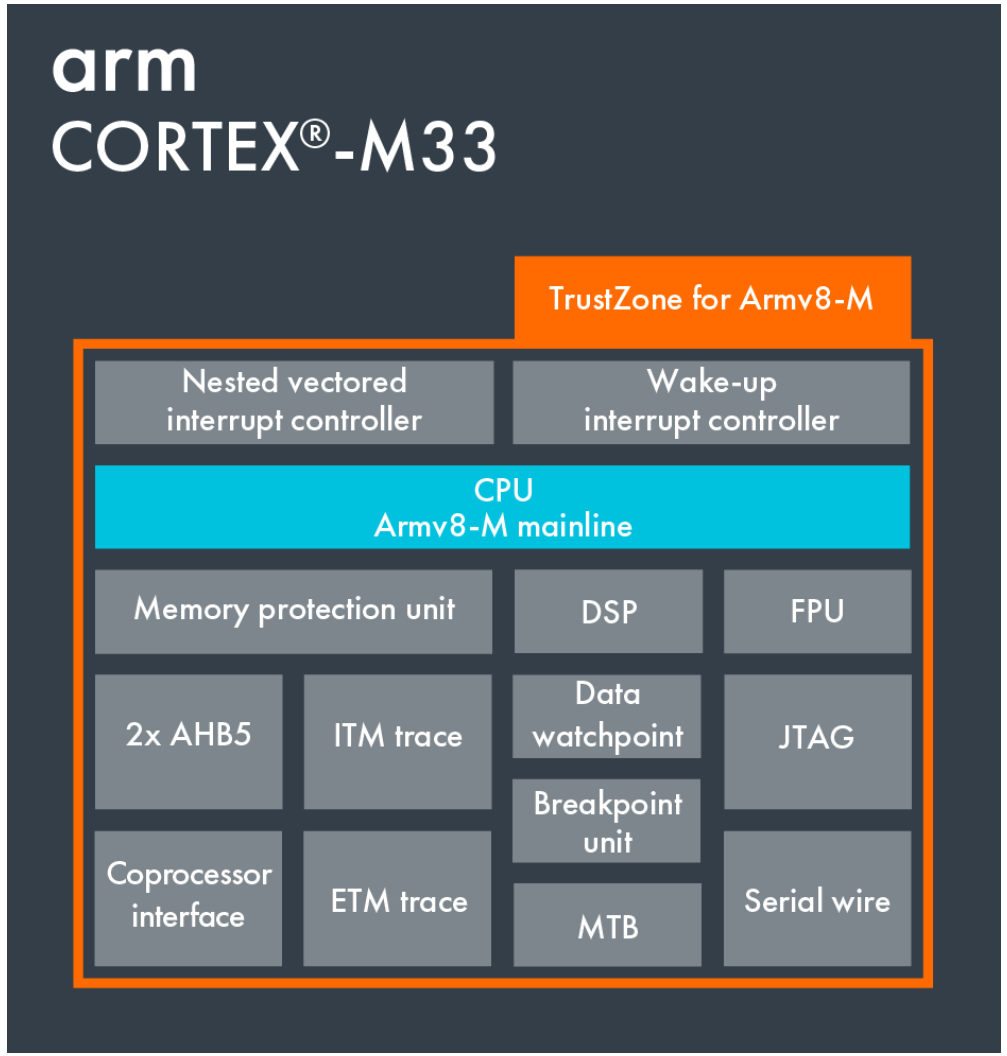
CORTEX-M33: NEXT-GENERATION CORTEX-M WITH TRUSTZONE SECURITY

Industry-standard 32bit processor

- 3-stage pipeline, Harvard architecture
- Extremely flexible design configurations

Wide choice of options for differentiated products

- TrustZone security foundation with up to two memory protection units (MPUs)
- Digital signal processing (DSP) extension with SIMD, single-cycle MAC, saturating arithmetic
- Floating-point Unit (FPU)
- Coprocessor interface
- Arm Custom Instructions
- Powerful debug and non-intrusive real-time trace (ETM, MTB)

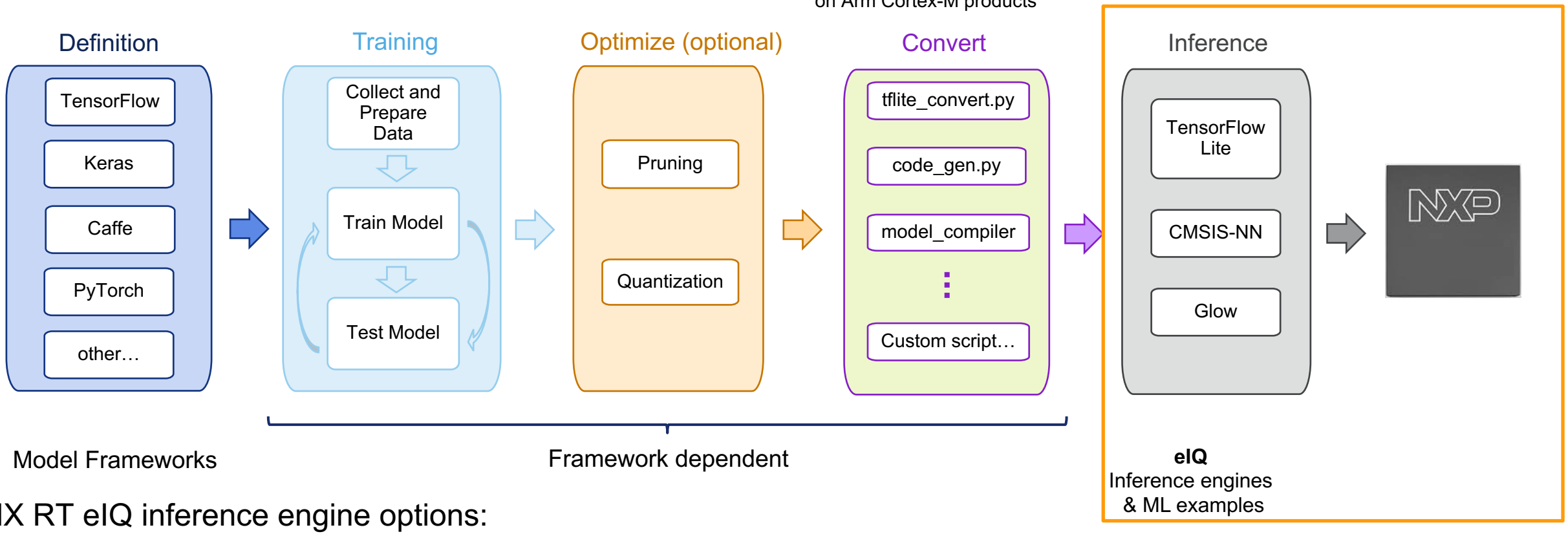


eIQ

SECURE CONNECTIONS
FOR A SMARTER WORLD

MACHINE LEARNING PROCESS

Note: There is no unified method for converting neural networks from different frameworks to run on Arm Cortex-M products



i.MX RT eIQ inference engine options:

- CMSIS-NN – Can be used for several different model frameworks
- TensorFlow Lite – Used for TensorFlow model frameworks
- Glow – Machine Learning compiler for several different model frameworks (Coming in July)

Collection of Libraries and Development Tools for Building Machine Learning Apps Targeting NXP MCUs and App Processors

Deploying open-source inference engines

Integration and optimization of neural net (NN) inference engines (Arm NN, Arm CMSIS-NN, OpenCV, TFLite, ONNX, etc.)

End-to-end examples demonstrating customer use-cases (e.g. camera → inference engine)

Support for emerging neural net compilers (e.g. Glow)

Suite of classical ML algorithms such as support vector machine (SVM) and random forest

BYOM – Bring Your Own Model

Integrated into Yocto Linux BSP and MCUXpresso SDK

No separate SDK or release to download

- iMX: New layer meta-imx-machinelearning in Yocto
- MCU: Integrated in MCUXpresso SDK middleware

Supporting materials for ease of use

Documentation: eIQ White Paper, Release Notes, eIQ User's Guide, Demo User's Guide

Guidelines for importing pretrained models based on popular NN frameworks (e.g. TensorFlow, Caffe)

Training collateral for CAS, DFAEs and customers (e.g. lectures, hands-on, video)

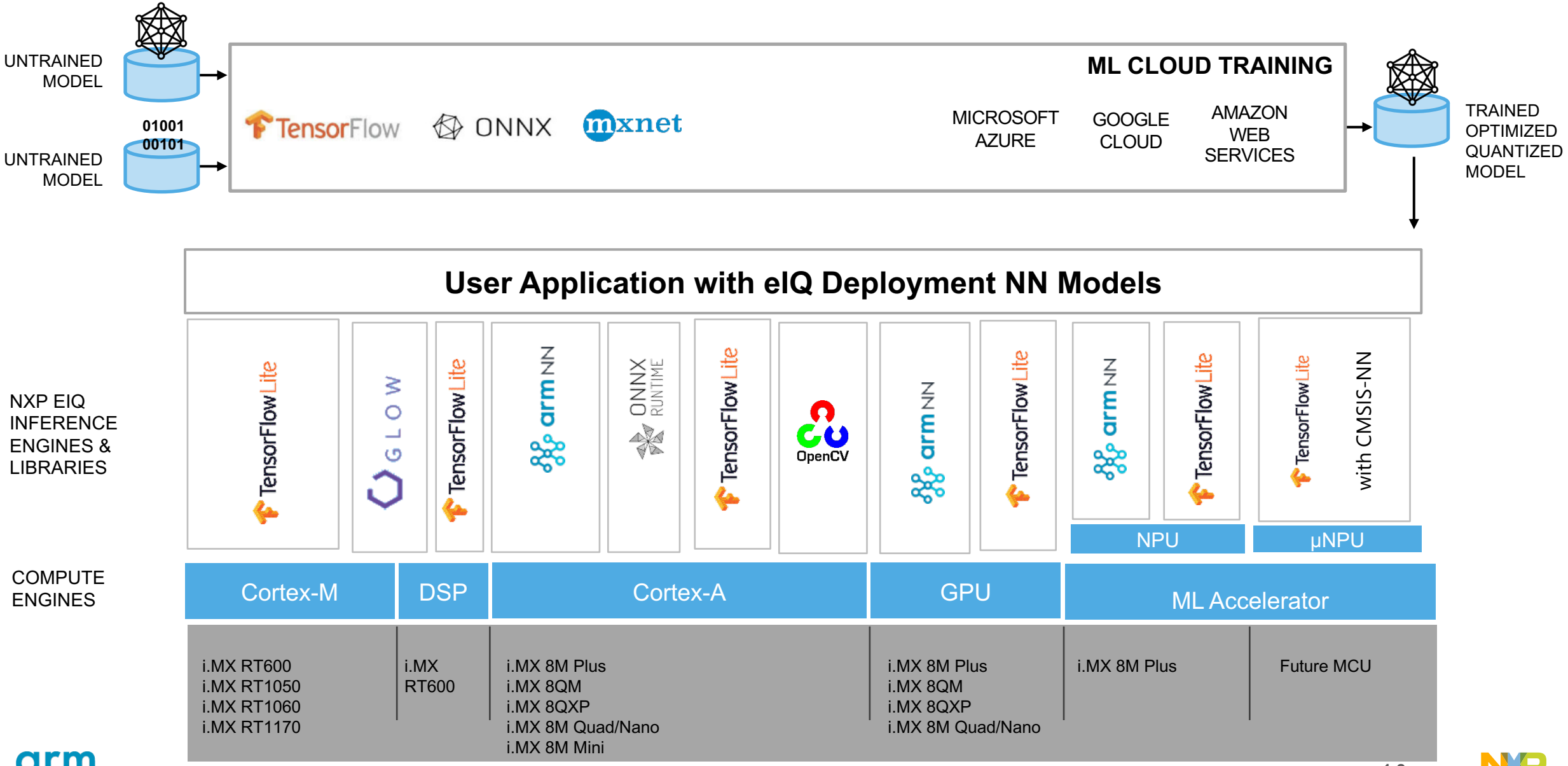
eIQ DEMO

- Retrained a Mobilenet model written in TensorFlow to identify 5 different flower types
- Use eIQ to run model on i.MX RT1060 EVK
  - Lab at https://community.nxp.com/docs/DOC-343827
  - Lab steps can be used for any types of images you're interested in

eIQ Deployment Overview

SECURE CONNECTIONS
FOR A SMARTER WORLD

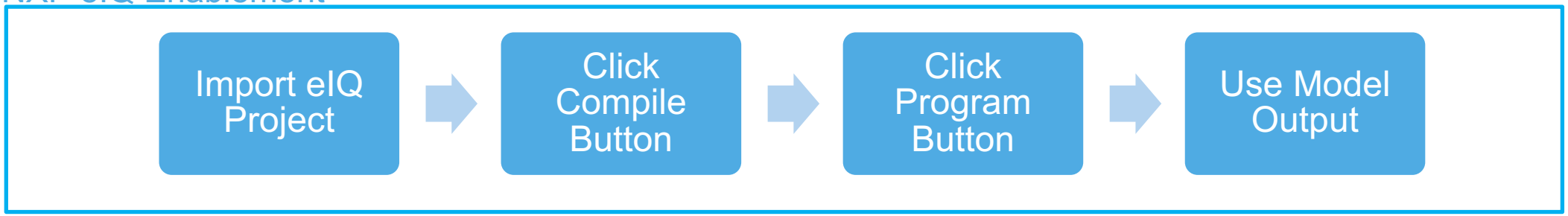
ADDITIONAL FLAVORS of NXP eIQ™ MACHINE LEARNING DEVELOPMENT ENVIRONMENT



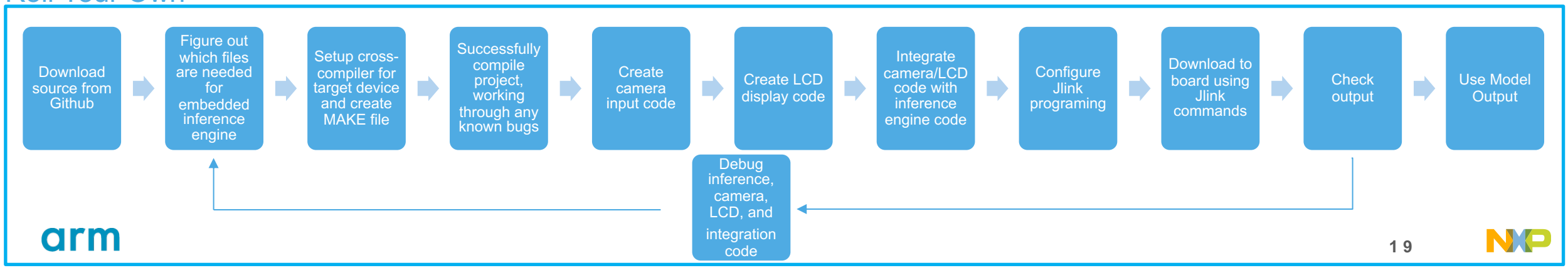
eIQ ADVANTAGES

- eIQ implements performance enhancements with CMSIS-NN for Cortex M cores and DSP
  - Up to 2.4x improvement in inference time in TensorFlow Lite over original code
- eIQ inference engines work out-of-the-box and are already tested and optimized.
  - Get up and running in minutes instead of weeks

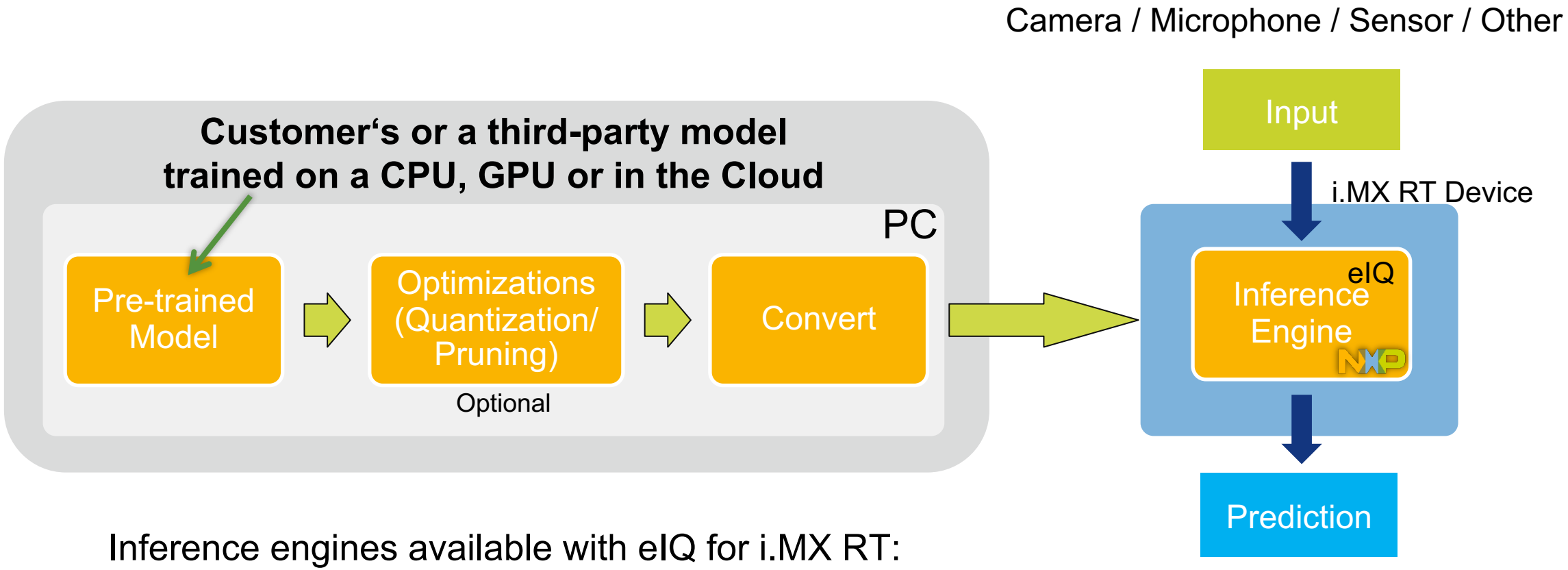
NXP eIQ Enablement



Roll Your Own



eIQ FOR I.MX RT



Inference engines available with eIQ for i.MX RT:

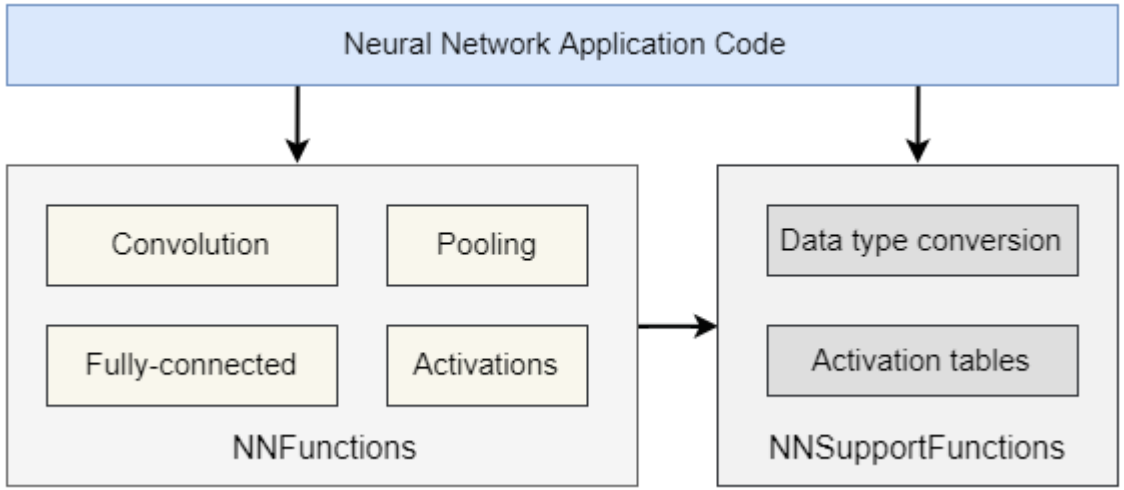
- CMSIS-NN – Can be used for several different model frameworks
- TensorFlow Lite – Used for TensorFlow model frameworks
- Glow – Machine Learning compiler for several different model frameworks (Coming in July)

Arm support for TFLμ

SECURE CONNECTIONS
FOR A SMARTER WORLD

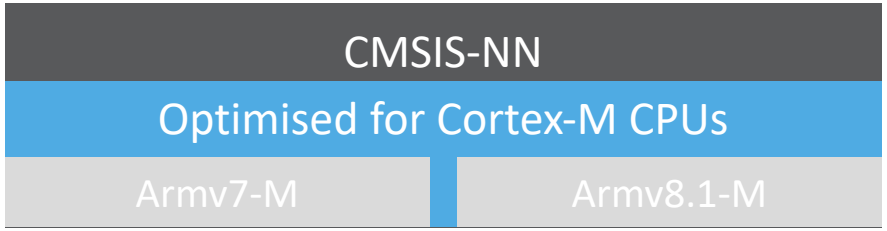
CMSIS-NN INFERENCE

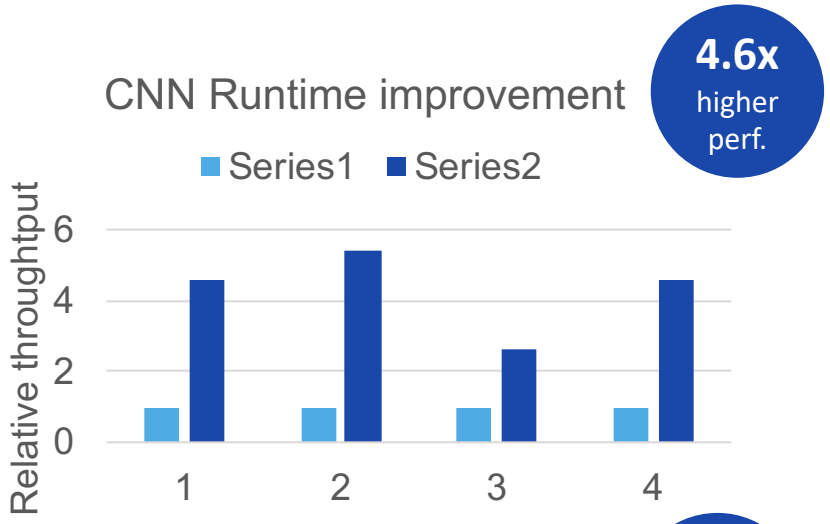
- Developed by Arm
- API to implement common model layers such as convolution, fully-connected, pooling, activation, etc, efficiently at a low level
- Conversion scripts (provided by Arm) to convert models into CMSIS-NN API calls.
- **CMSIS-NN optimized the implementation of inference engines like TFLite micro (https://www.tensorflow.org/lite/microcontrollers)**

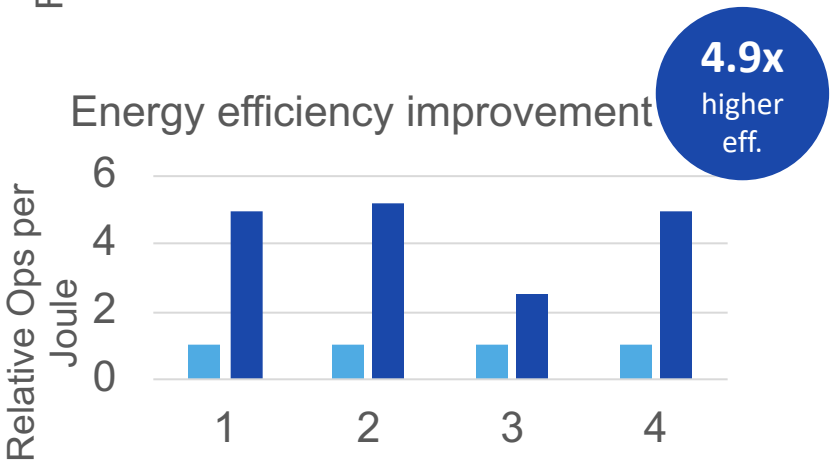


CMSIS-NN OPTIMIZED FOR PERFORMANCE

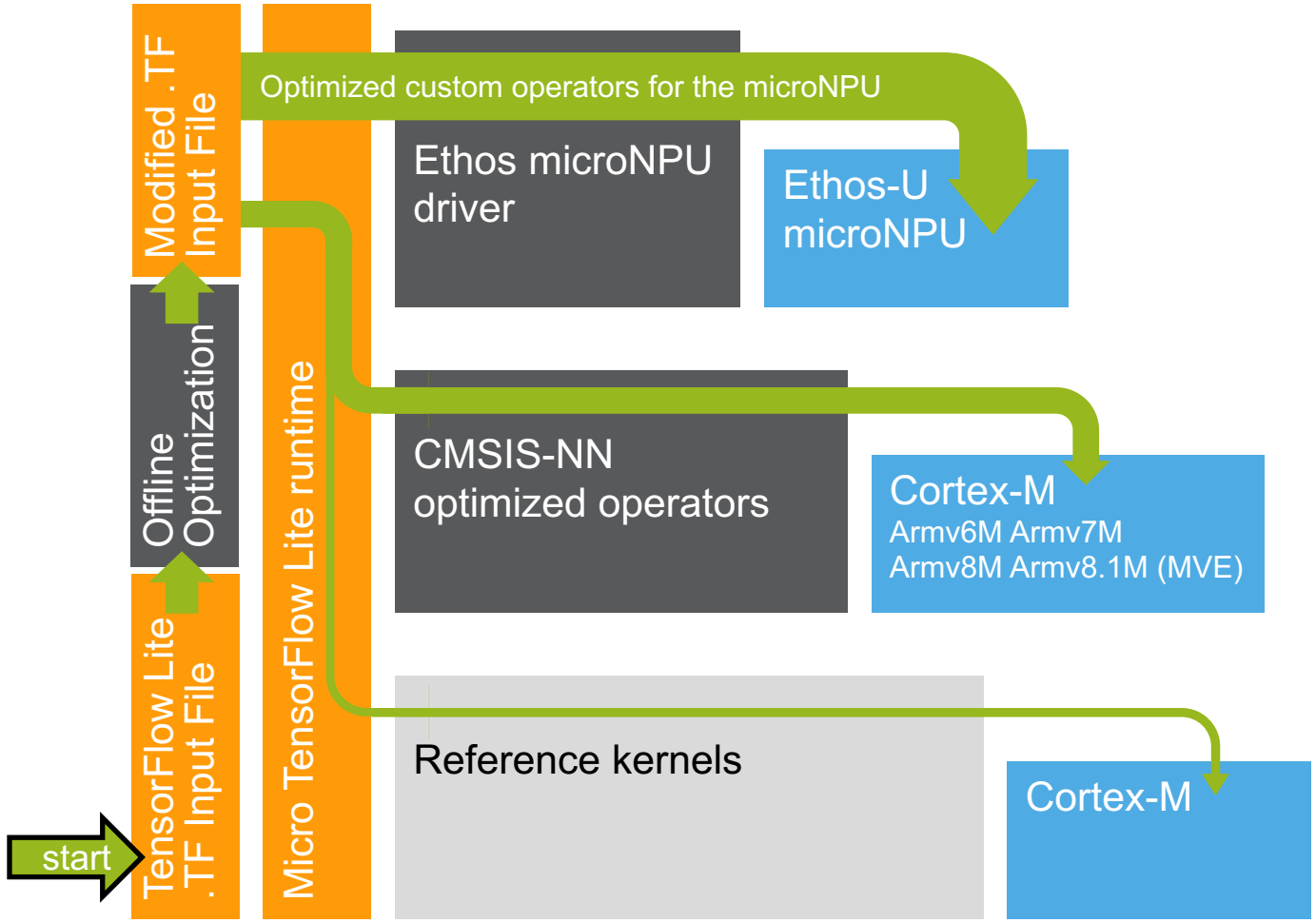
- Key ML function support
  - Aiming for best-in-class performance for Cortex-M CPUs (compared to other libraries)
  - Available now through open source license
- Consistent interface to all Cortex-M CPUs
  - Extending to Arm v8-M
- Open-source, via Apache 2.0 license
  - https://github.com/ARM-software/CMSIS_5







TOOLS & TFLM OPERATOR SUPPORT – CMSIS-NN AND ETHOS MICRONPU



eIQ TensorFlow

SECURE CONNECTIONS
FOR A SMARTER WORLD

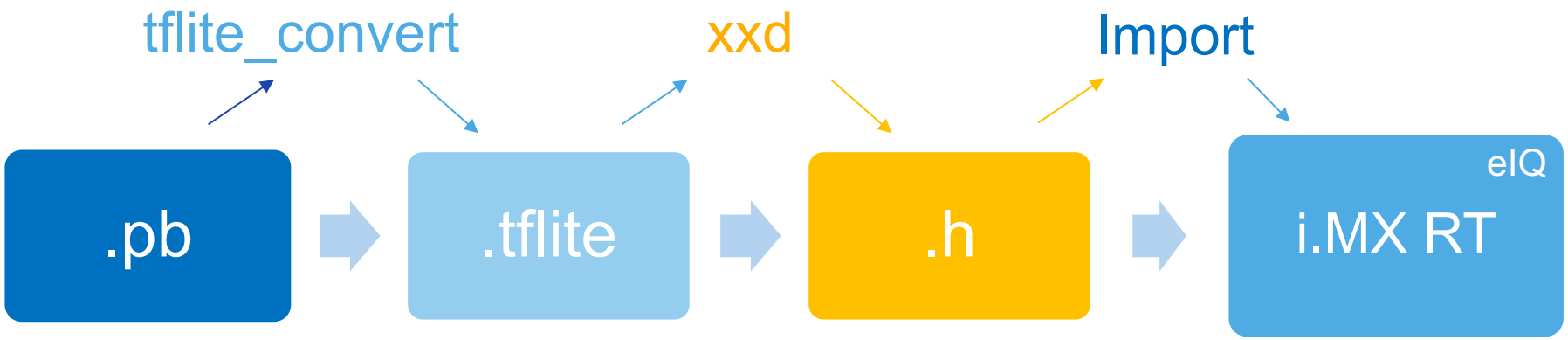
TENSORFLOW LITE INFERENCE ENGINE

- Developed by Google
  - TensorFlow → Training and Inference
  - TensorFlow Lite eIQ → NXP's implementation of TF Lite for MCUs
  - TensorFlow Lite Micro → TensorFlow's implementation of TF Lite for MCUs

- Can only be used with TensorFlow models
- Use tflite_convert utility (provided by TensorFlow) to convert a TensorFlow model to a .tflite binary
- TFLite flat buffer binary is read from memory by TFLite inference engine running on i.MX RT

TENSORFLOW LITE CONVERSION PROCESS

1. Transform a TensorFlow .pb model to TFLite flat buffer file.
2. Convert TFLite flat buffer file to C array in a **.h header**
3. Copy .h header file into eIQ TensorFlow Lite SDK example



TENSORFLOW LITE CODE FLOW

- Import model

```
#include “mobilenet_model.h”
model = tflite::FlatBufferModel::BuildFromBuffer(mobilenet_model, mobilenet_model_len);
```

- Get input

```
/* Extract image from camera to data buffer. */
CSI2Image(data, Rec_w, Rec_h, pExtract, true);
/* Resize image to input tensor size. */
ResizeImage(interpreter->tensor(input), data, Rec_h, Rec_w, image_height, image_width, image_channels, &s);
```

- Run inference

```
interpreter->Invoke();
```

- Get Results

```
std::vector<std::pair<float, int>> top_results;
GetTopN<float>(interpreter->typed_output_tensor<float>(0), output_size, s->number_of_results, threshold, &top_results, true);
auto result = top_results.front(); //Get results
const float confidence = result.first;  //Get confidence level
const int index = result.second;   //Get highest class
```

GEMMLOWP ASSEMBLY-CODED DSP OPTIMIZATION BENEFITS FOR TENSORFLOW LITE

GCC Arm® 8-2018-q4

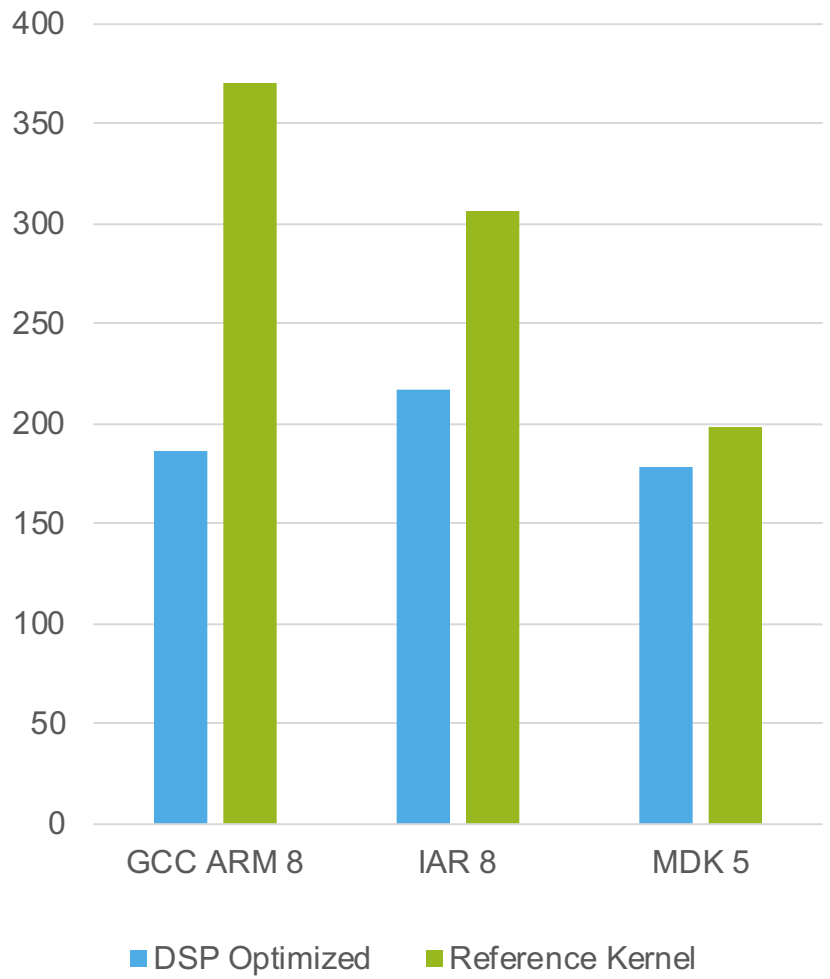
	DSP Optimized (-O2)	Reference Kernel (-O2)
Label Image	186 ms	370 ms
CIFAR-10	61 ms	229 ms

IAR EW 8.32.3

	DSP Optimized	Original
Label Image	217 ms	307 ms
CIFAR-10	67 ms	159 ms

Keil MDK 5.27

	DSP Optimized	Original
Label Image	178 ms	198 ms
CIFAR-10	64 ms	87 ms



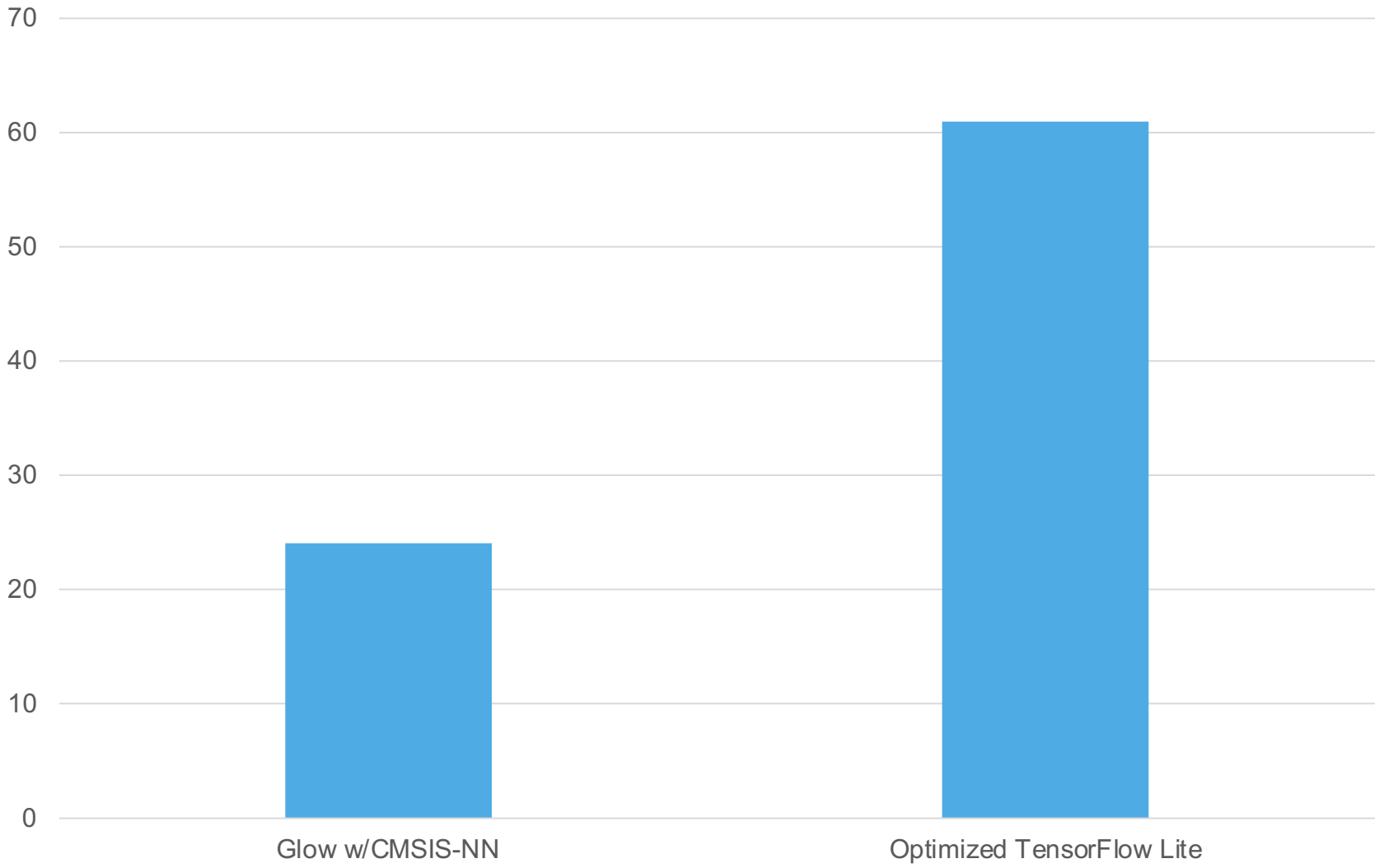
eIQ Glow

SECURE CONNECTIONS
FOR A SMARTER WORLD

GLOW (COMING IN JULY)

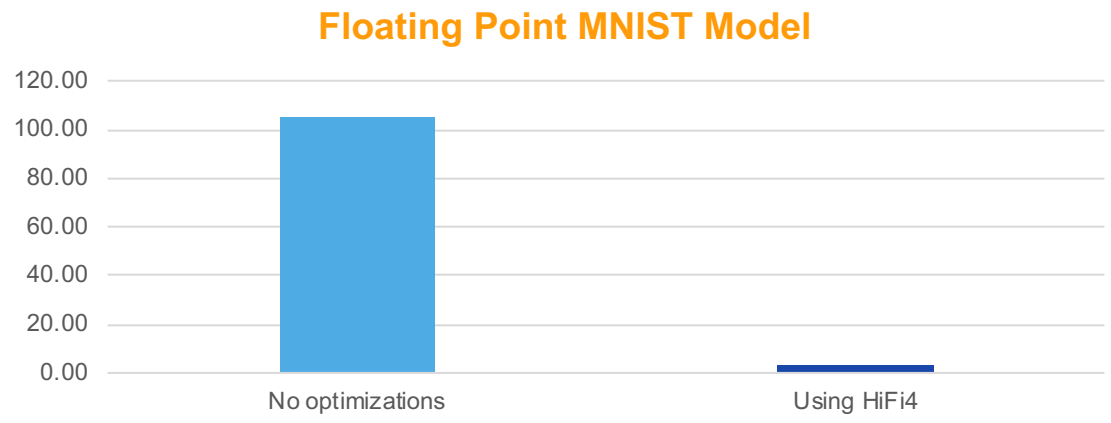
- Developed by Facebook
- Glow is a compiler that turns a model into an machine executable binary for the target device
  - Both the model and the inference engine are compiled into the binary that is generated.
  - Integrate the generated binary into an SDK software project
  - Can make use of compiler optimizations
  - Supports ONNX (universal model format) and Caffe2 models
- Cutting-edge inference technology

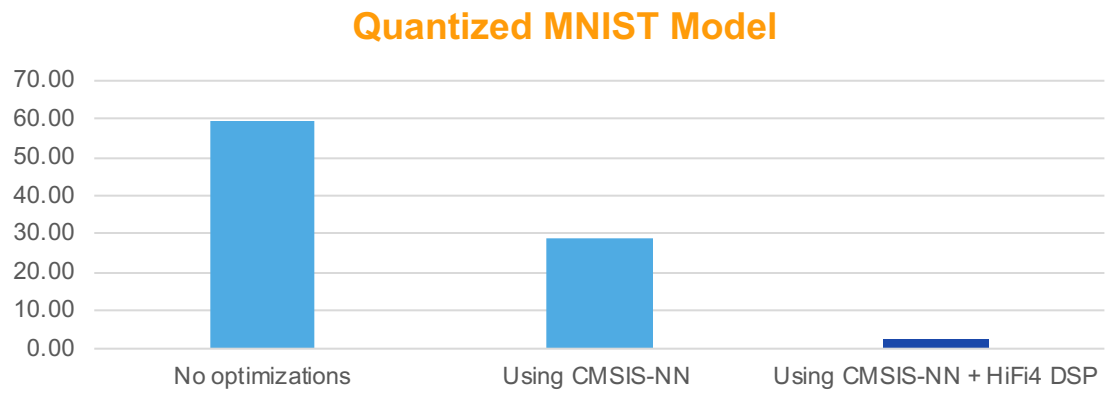
PERFORMANCE COMPARISON USING CIFAR-10 MODEL ON RT1050



OPTIMIZATIONS FOR GLOW

- NXP developed optimizations for Glow on i.MX RT devices
- Operations can be dispatched to the HiFi4 DSP on RT685
  - HiFi4 DSP increases performance up to 34x
- Operations can also use CMSIS-NN library optimizations for all Glow supported devices

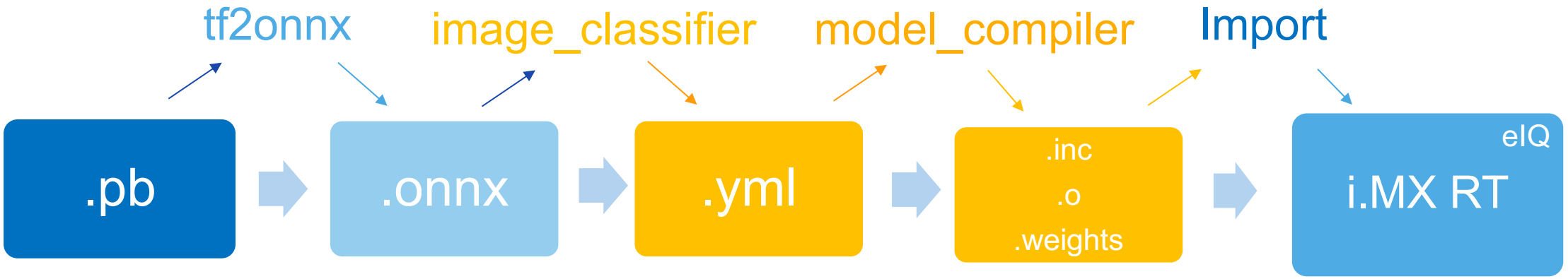




Glow Inference Time on RT685 (in milliseconds)	MNIST Model	CIFAR10 Model
Floating Point Model	104.63	213.78
Floating Point Model using HiFi4 DSP	3.02	13.36
Quantized Model	59.77	165.37
Quantized Model using CMSIS-NN	28.52	89.95
Quantized Model using CMSIS-NN + HiFi4 DSP	2.50	6.70

GLOW

1. Transform model to the universal ONNX format.
2. Optimize model with profiler to create profile.yml file for quantization
3. Compile with Glow model_compiler to generate compiled files and weights.
4. Copy binary files into eIQ Glow SDK example.



ADD COMPILED CODE TO PROJECT

- Add <network_name>.o compiled file to project settings
- Include <network_name>.h file
- Set input data

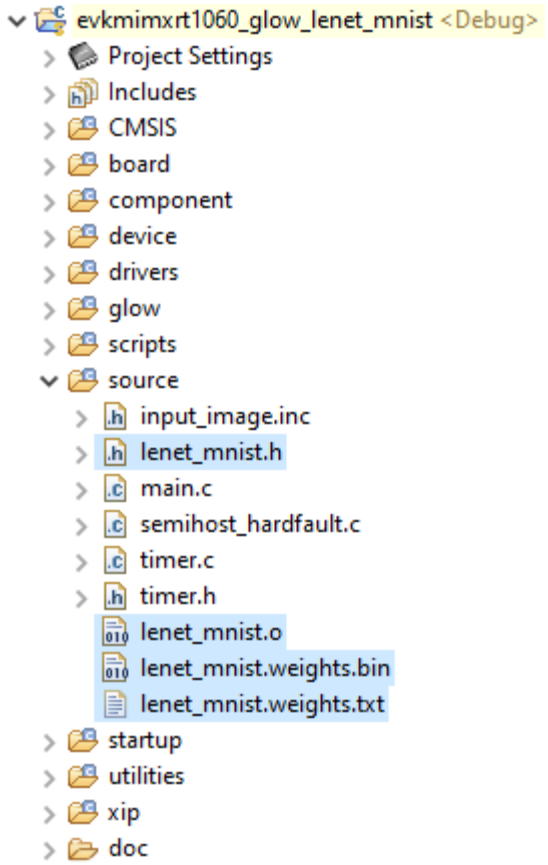
```
// Load input data
memcpy(bundleInpAddr, ((char *)INPUT_DATA_START) + idx * INPUT_IMAGE_SIZE, INPUT_IMAGE_SIZE);
```

- Run model

```
// Perform inference and compute inference time.
start_time = get_time_in_us();
mnist(constantWeight, mutableWeight, activations);
stop_time = get_time_in_us();
duration_ms = (stop_time - start_time) / 1000;
```

- Get result

```
// Get classification top1 result and confidence
float *out_data  = (float *)(bundleOutAddr);
float max_val    = 0.0;
uint32_t max_idx = 0;
for (int i = 0; i < OUTPUT_NUM_CLASS; i++)
{
    if (out_data[i] > max_val)
    {
        max_val = out_data[i];
        max_idx = i;
    }
}
```



GLOW MEMORY USAGE

- Glow does not use dynamically allocated memory (heap).
- All the memory requirements of a compiled model can be found in the auto-generated header file.

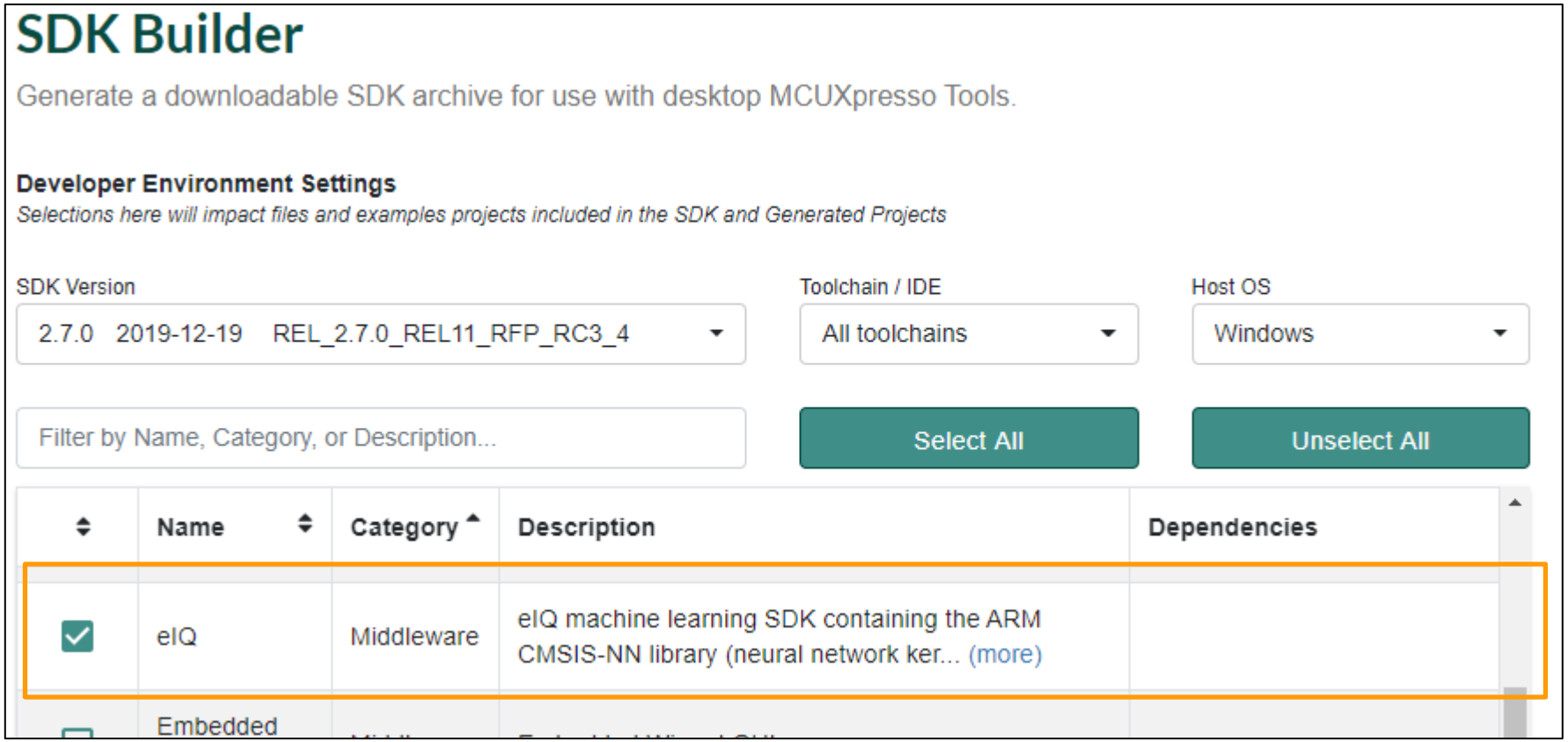
```
// Memory sizes (bytes).
#define CIFAR10_CONSTANT_MEM_SIZE     34176     // Stores model weights. Can be stored in Flash or RAM.
#define CIFAR10_MUTABLE_MEM_SIZE      12352     // Stores model inputs/outputs. Must be in RAM.
#define CIFAR10_ACTIVATIONS_MEM_SIZE  71680     // Store model scratch memory required for intermediate computations. Must be in RAM.
```

Getting eIQ

SECURE CONNECTIONS
FOR A SMARTER WORLD

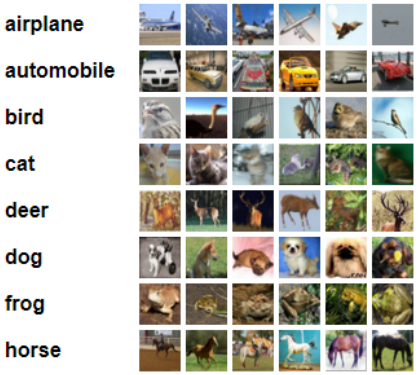
eIQ IN MCUXPRESSO SDK

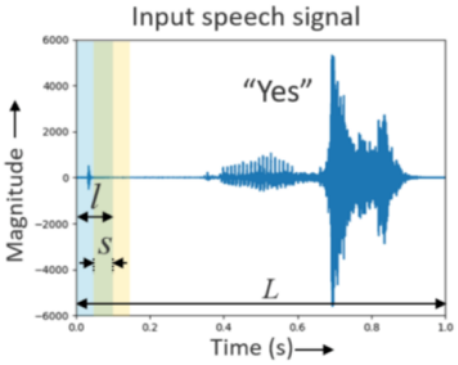
- eIQ for the i.MX RT family is included as part of MCUXpresso SDK (https://mcuxpresso.nxp.com/en/welcome)
- Make sure eIQ selected in MCUXpresso SDK builder:

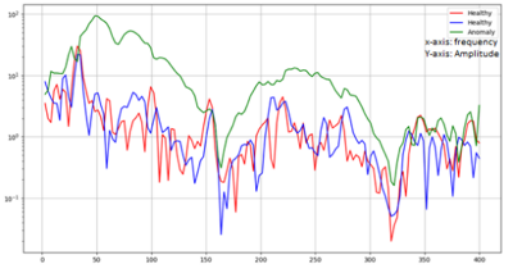


eIQ RT1060 SDK examples available:

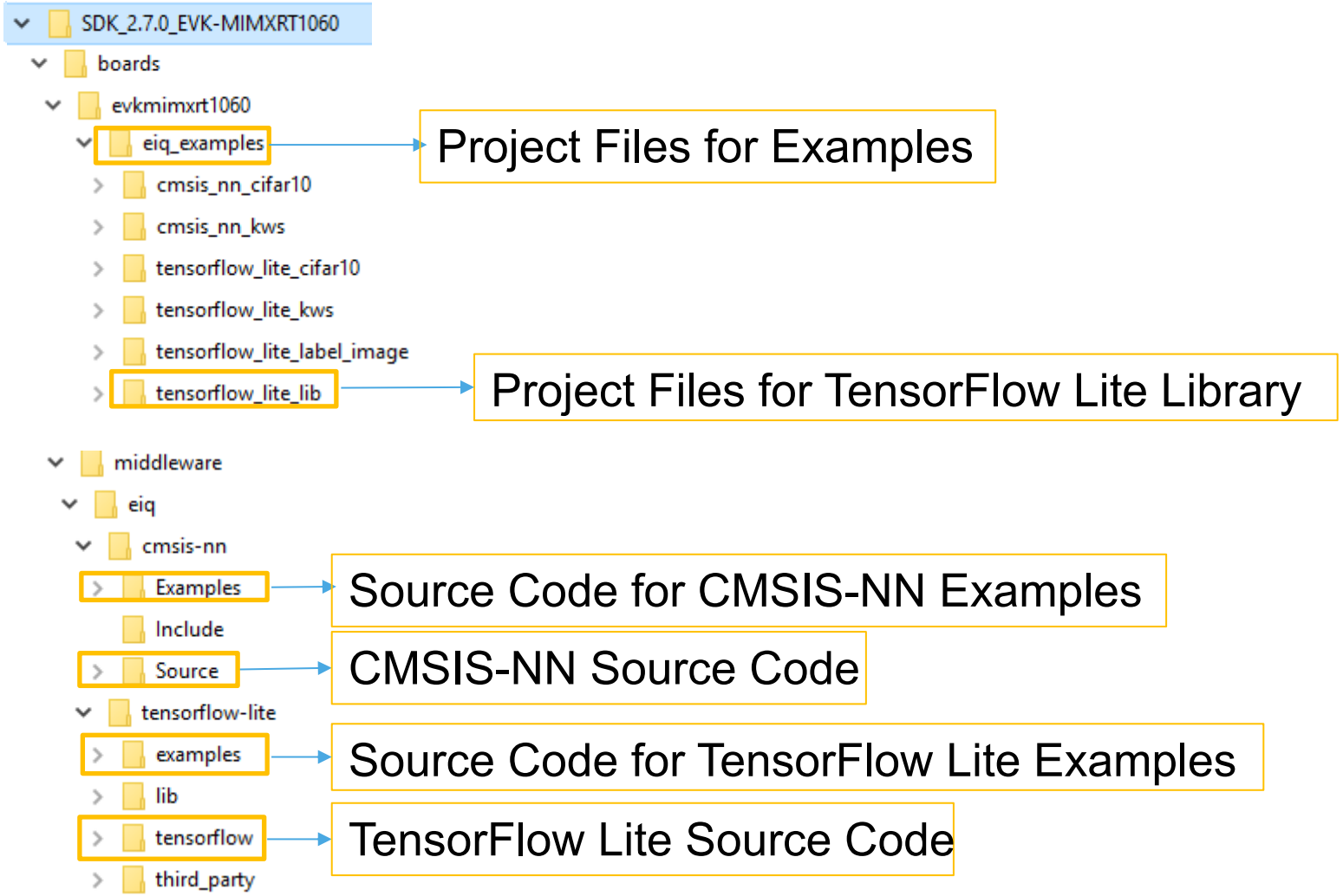
	CIFAR-10	Keyword Spotting (KWS)	Label Image	Anomaly Detection
Description	Classifies 32x32 image from camera input into one of 10 categories	Detects specific keywords from microphone input	Classifies 128x128 image from camera input into one of 1000 categories using Mobilenet model	Use FRDM-STBC-AGM01 sensor board for accelerometer anomaly analysis (Select “agm01” board)
TensorFlow Lite Example	✓	✓	✓	✓
CMSIS-NN Example	✓	✓		





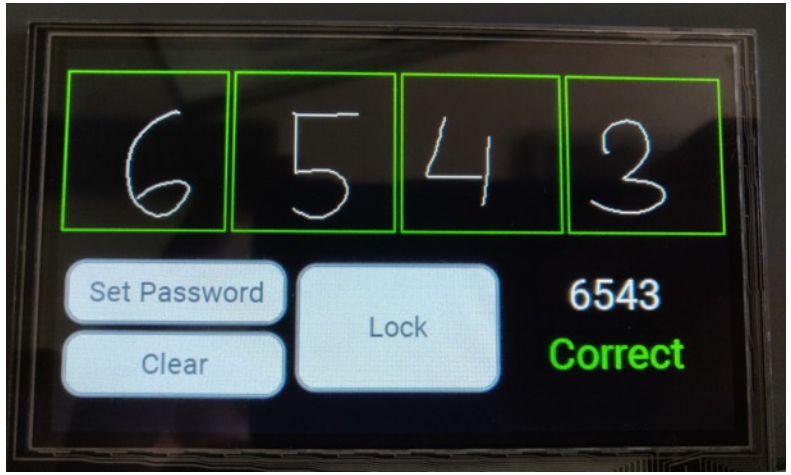


eIQ FOLDER STRUCTURE



eIQ APP NOTES

- Anomaly Detection with eIQ using K-Means clustering in TF-Lite (AN12766)
- Handwritten Digit Recognition using TensorFlow Lite (AN12603)



Coming Soon:

- Transfer Learning and Datasets

INFERENCE TIMES

- Benchmarking ongoing and optimizations still under development. Numbers subject to change
- Inference time heavily dependent on the particular model
  - Different images (if same size) will not affect inference time
- Each eIQ example reports inference time

Image Classification (ms)	CIFAR-10 (32x32 input)	Mobilenet (128x128 input)
RT685 w/ HiFi4 Glow	6.7	61
RT1060 Glow	24	74
RT1060 TensorFlow Lite	64	178

MEMORY REQUIREMENTS

- Flash: Model, inference engine code, and input data
- RAM: Intermediate products of the model layers
  - Size depends on amount of data, size, and type of the layers and is very model dependent

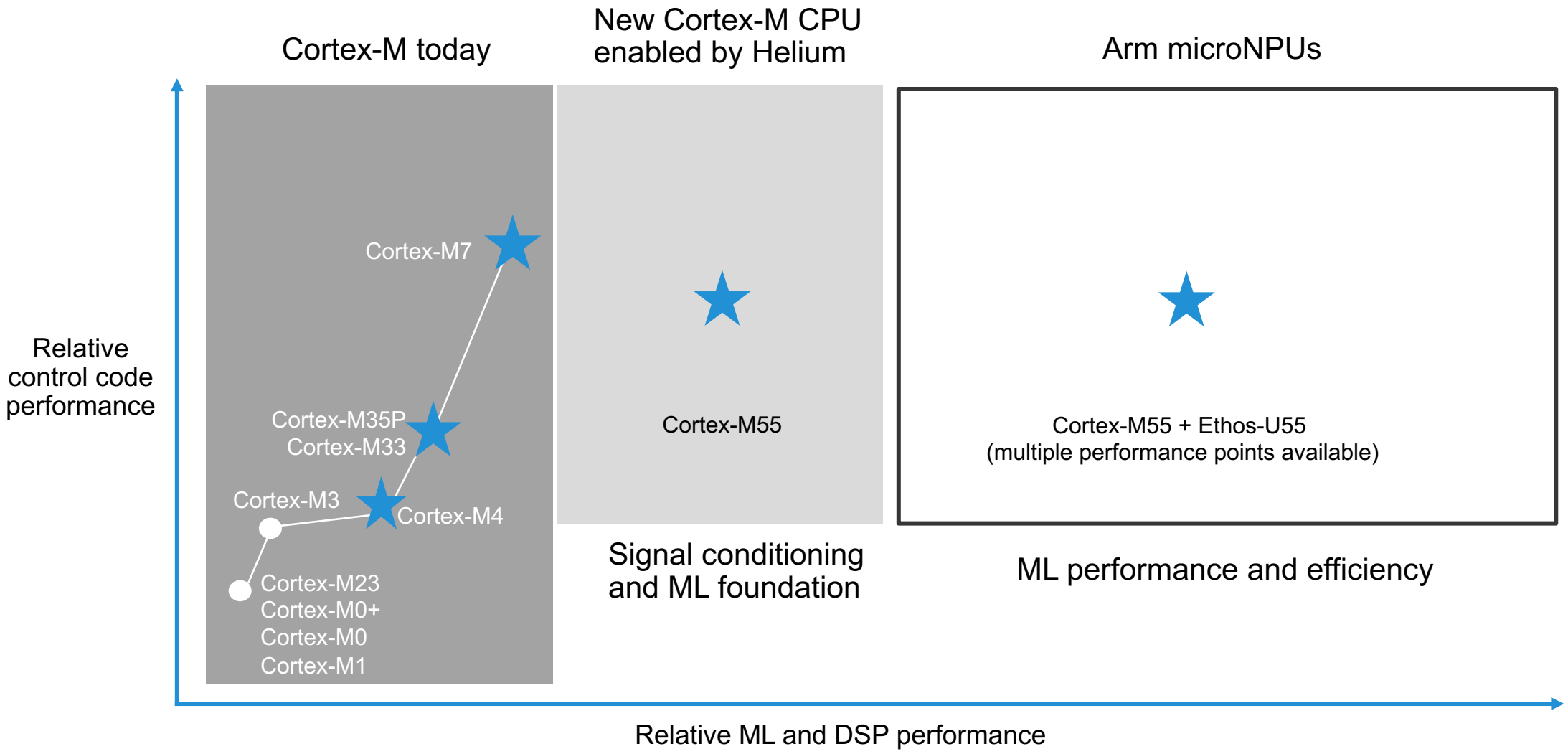
Benchmark and optimizations ongoing. Numbers subject to change:

Model	Inference Engine	Flash	RAM
CIFAR-10	CMSIS-NN	110KB	50KB
CIFAR-10	TensorFlow Lite	600KB (92KB model, 450KB engine)	320KB
CIFAR-10	Glow	69KB	131KB
Mobilenet v1	TensorFlow Lite	1.5MB (450KB model, 450KB engine)	2.5MB
Mobilenet v1	Glow	507KB	1MB

The future

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUSHING THE BOUNDARIES FOR REAL-TIME ON-DEVICE PROCESSING



Well suited for ML & DSP applications

CORTEX-M55 & ETHOS-U55: TRANSFORMING CAPABILITIES OF THE SMALLEST DEVICES

Boosting signal processing and ML performance for millions of developers

Signal processing → Machine learning →

Signal conditioning

Feature extraction

Decision algorithm

Cortex-M55
Up to 5x higher signal processing performance
(CFFT in int32)

Cortex-M55
Up to 15x higher ML performance*
(matrix multiplication in int8)

Cortex-M55 & Ethos-U55
Up to 480x higher ML performance*
(matrix multiplication in int8)

*Compared to existing Armv8-M implementations

Summary

SECURE CONNECTIONS
FOR A SMARTER WORLD

FURTHER READING

- NXP eIQ
- TensorFlow Lite
- Glow
- CMSIS-NN

Machine Learning Courses:

- Video series on Neural Network basics
- Arm Embedded Machine Learning for Dummies
- Google TensorFlow Lab
- Google Machine Learning Crash Course
- Google Image Classification Practical
- YouTube series on the basics of ML and TensorFlow (ML Zero to Hero Series)

Book:

You Look Like a Thing and I Love You: How Artificial Intelligence Works and Why It's Making the World a Weirder Place

GIT REPOS

- TensorFlow Lite
  - https://github.com/tensorflow/tensorflow/tree/v1.13.1/tensorflow/lite
- TensorFlow Lite for Microcontrollers
  - https://www.tensorflow.org/lite/microcontrollers

- CMSIS-NN
  - https://github.com/ARM-software/CMSIS_5/tree/master/CMSIS/NN
  - CIFAR-10: https://github.com/ARM-software/ML-examples/tree/master/cmsisnn-cifar10
  - KWS: https://github.com/ARM-software/ML-KWS-for-MCU

- Glow
  - https://github.com/pytorch/glow

NXP eIQ RESOURCES

- eIQ for i.MX RT is included in MCUXpresso SDK
  - https://mcuxpresso.nxp.com
  - TF-Lite and CMSIS-NN eIQ User Guides in SDK documents

- eIQ available for i.MX RT1050 and i.MX RT1060 today
  - Can also run on i.MX RT1064: https://community.nxp.com/docs/DOC-344225
- eIQ available for i.MX RT685 in July

- Transfer Learning Lab: https://community.nxp.com/docs/DOC-343827
- Anomaly Detection App Note: https://www.nxp.com/docs/en/application-note/AN12766.pdf
- Handwritten Digit Recognition: https://www.nxp.com/docs/en/application-note/AN12603.pdf

arm AI
Virtual Tech Talks Series

NXP

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה

arm AI

AI Virtual Tech Talks Series

Join our next virtual tech talk:

tinyML development with Tensorflow Lite for Microcontrollers and CMSIS-NN

Tuesday 30 June

Register here:

developer.arm.com/solutions/machine-learning-on-arm/ai-virtual-tech-talks

arm

arm AI
Virtual Tech Talk Series

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה