embedded VISION SUMMIT 2018

Deploying CNN based vision solutions on a $3 microcontroller

Au-Zone Technologies Inc.

Greg Lytle
May 22, 2018

Au-Zone Technologies Inc.

Au-Zone Technologies is a leading provider of development tools and enabling IP used for the design of intelligent embedded vision products and solutions.

Our architecture agnostic development tools (eCV SDK and DeepView) enable our customers quickly to develop and securely deploy machine learning solutions on a range of embedded hardware.

Design Challenge

Toaster Oven Example

Camera

Processor & Memory

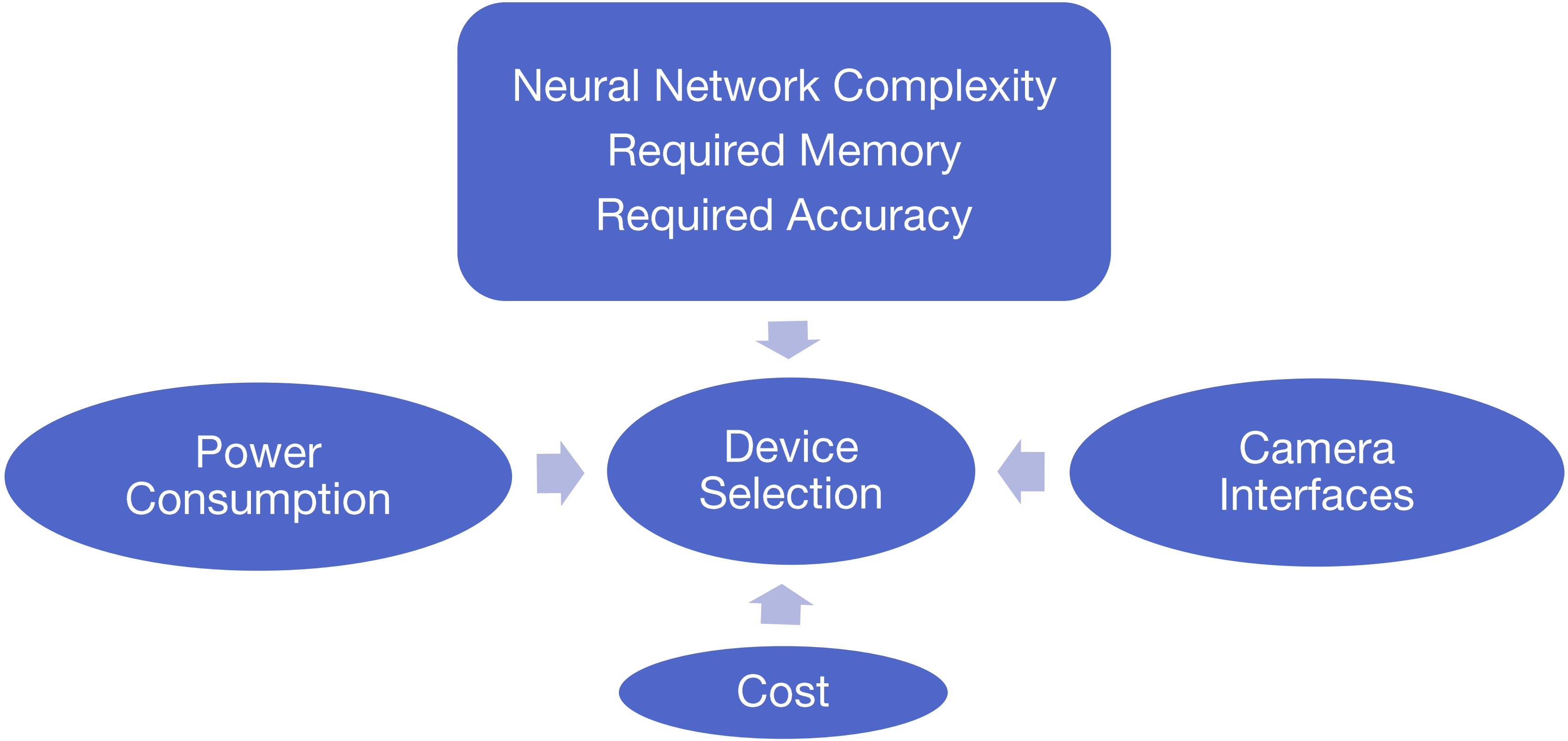
I/O

Display

Requirements:

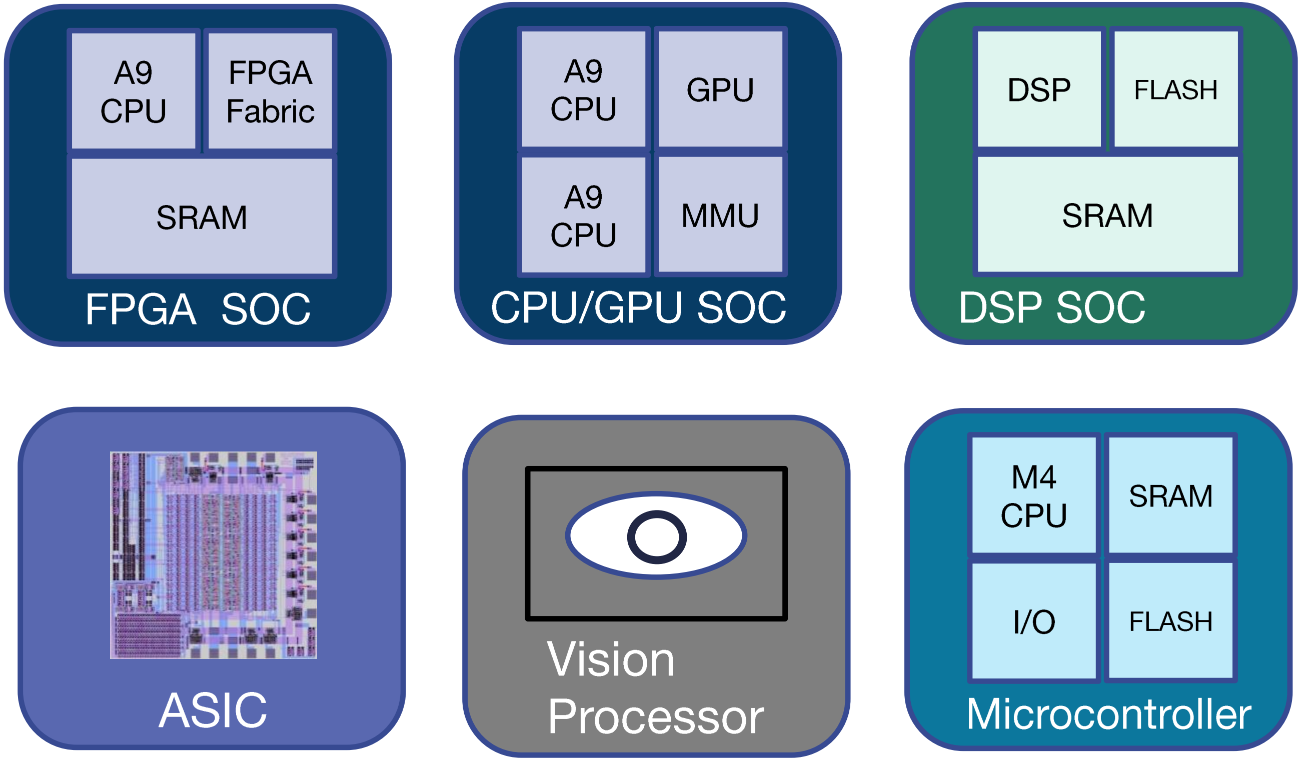
- Embedded image capture and classification
- Image Sensor
- LCD & Buttons
- Compute & Memory for vision algorithm and application
- Low BOM cost

Design Considerations



Architecture Options

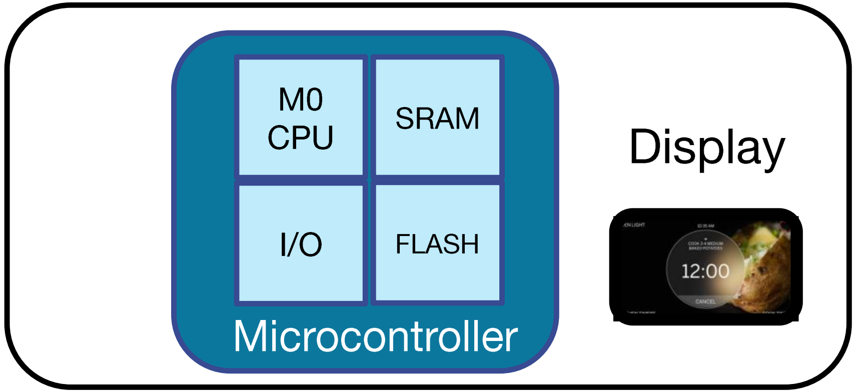
Many Options?

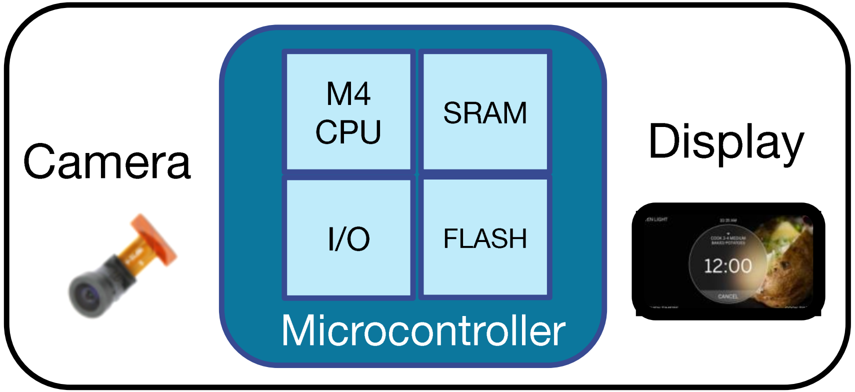


Market Drivers

- Is this practical?
- Why should you care?

Image Classification enabled





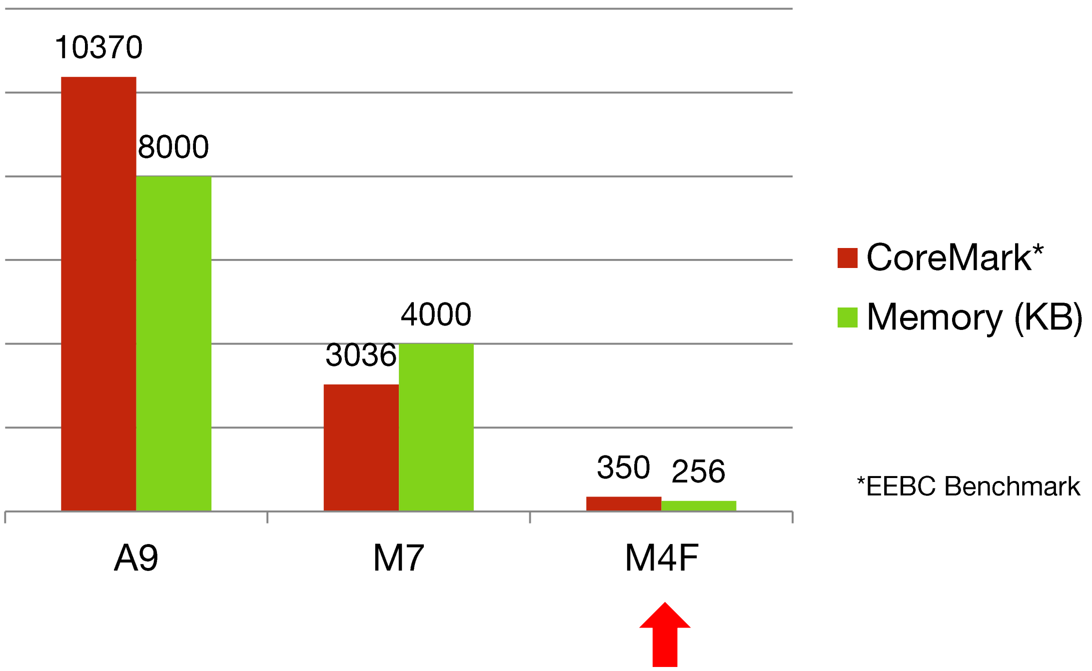
Low Incremental Cost

Constraints of Cortex M vs Cortex A

Much Lower raw processor performance

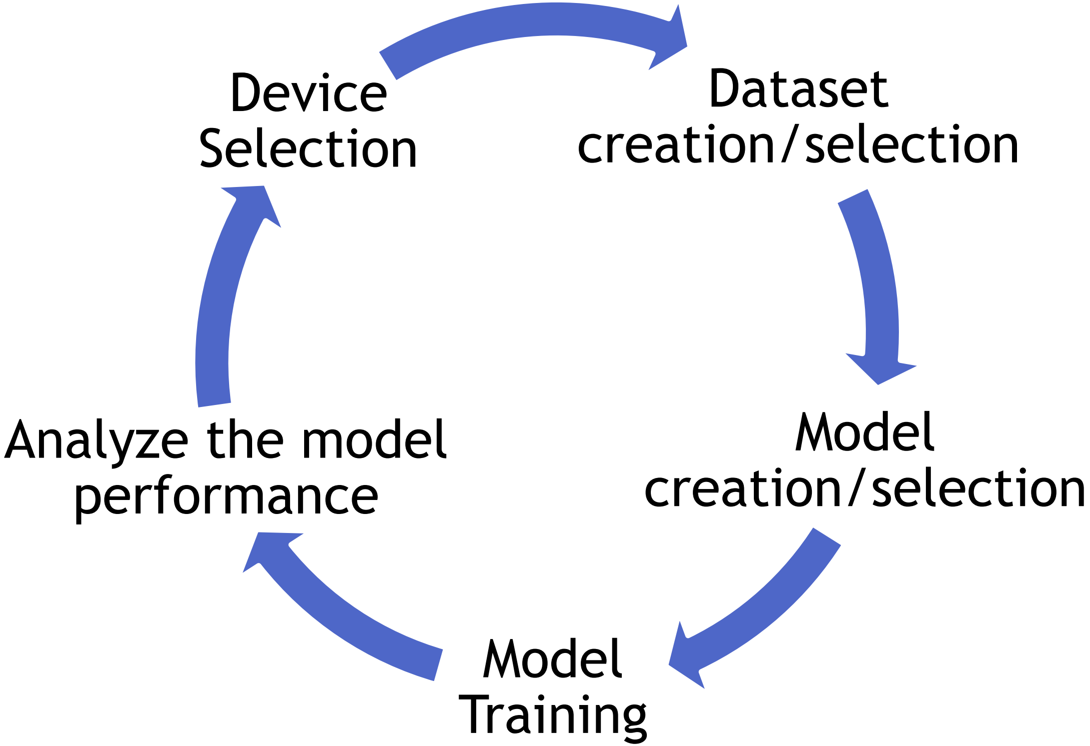
Significant Memory Constraints

CPU Performance & Memory

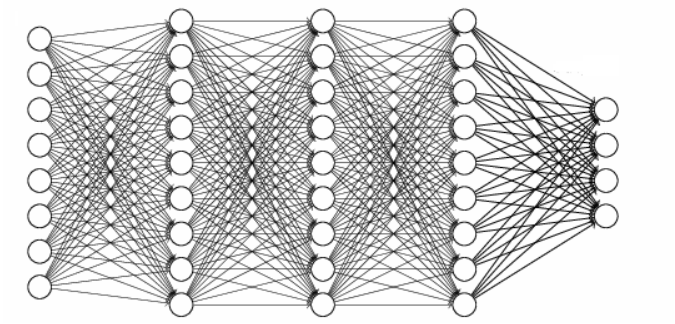


Network Design

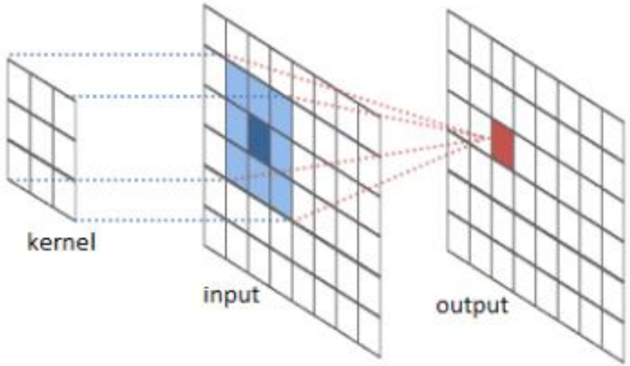
Network Model Design Flow



Network Layer Types

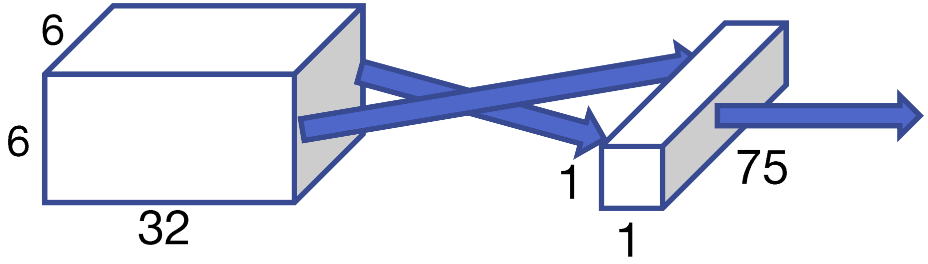


Fully Connected Layer



Convolutional Layer

Fully Connected Operation

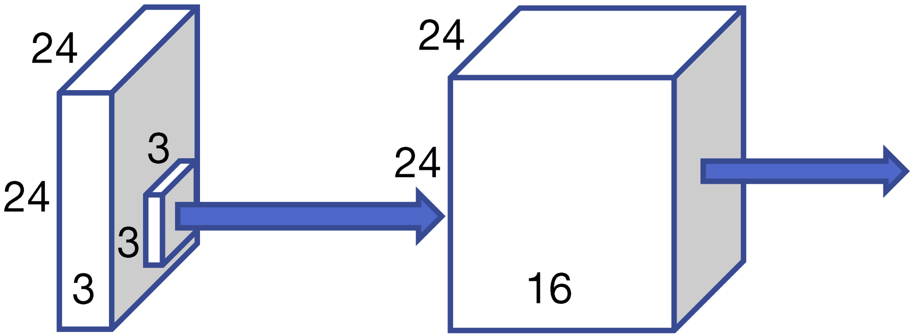


One Weight required for every connection

8-bit quantization saves on FLASH size with low impact on error

Parameter	Resource	Size	8bit	32 bit float	Typical
Input Buffer	RAM	6 x 6 x 32	1.1 KB	4.6 KB	4.6 KB
Weights	FLASH	6 x 6 x 32 x 1 x 1 x 75	86.4 KB	345.6 KB	86.4 KB
Output	RAM	1 x 1 x 75	.075 KB	3.00 KB	3 KB
Total Memory			87.6 KB	351 KB	7.6 KB RAM 86.4 KB FLASH
MAC Operations	CPU	(6 x 6 x 32) x (1 x 1 x 75)	86.4 K		86.4K

Convolution Operation



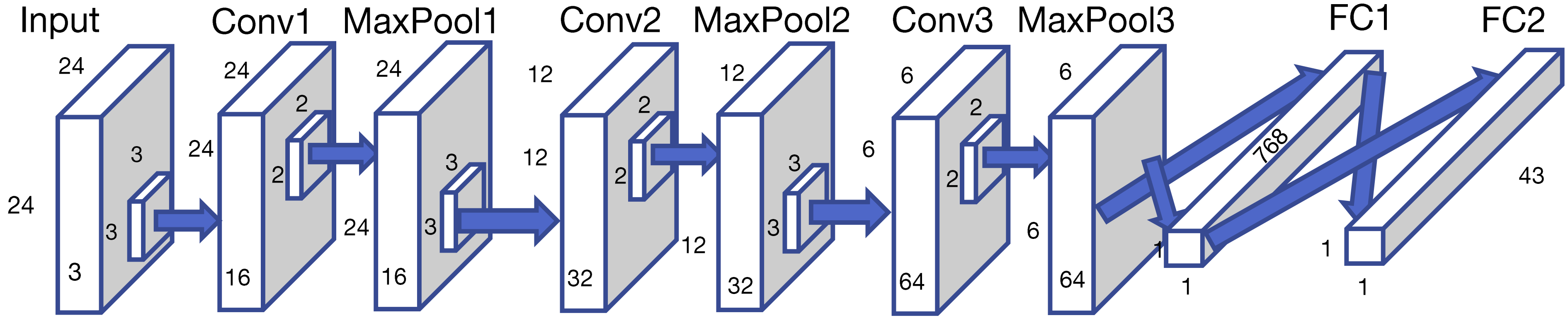
Parameter	Resource	Size	8bit	32 bit float	Typical
Input Buffer	RAM	26 x 26 x 3. (24x24x3 plus padding)	2.1 KB	8.1 KB	8.1 KB
3x3 Kernel	FLASH	3 x 3 x 3 x 16	0.42 KB	1.73 KB	0.42 KB
Output	RAM	24 x 24 x 16	9.2 KB	36.8 KB	36.8 KB
Total Mem			11.6 KB	46.7 KB	0.42 KB FLASH 44.9 KB RAM
MAC Ops	CPU	(24 x 24 x 3) x (3 x 3 x 16)	248.8 K		

Comparisons using GTSRB Data Set

GTSRB –

- German Traffic Signs in the wild
- 39K Training Samples
- 12.6K Testing Samples
- 43 classes of signs
- Cropped images 24 x 24 x 3

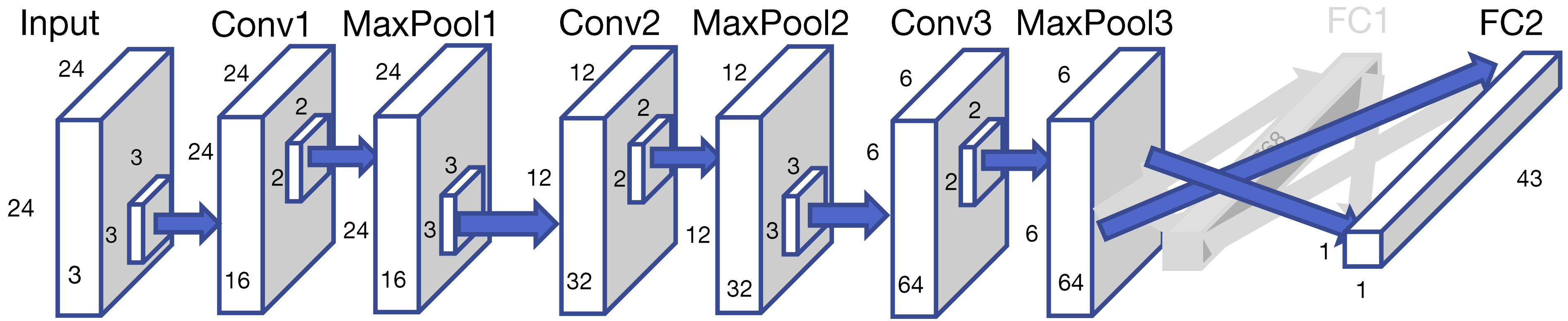
CNN with Fully Connected Layers



Model FLASH	2 MB	MAC Operations	~2M
Minimum RAM	74 KB	Weights	488K
Minimum Cache	2 KB	Layers	9
Optimum Cache	74 KB	Accuracy	94.9

(example for 32-bit float implementation)

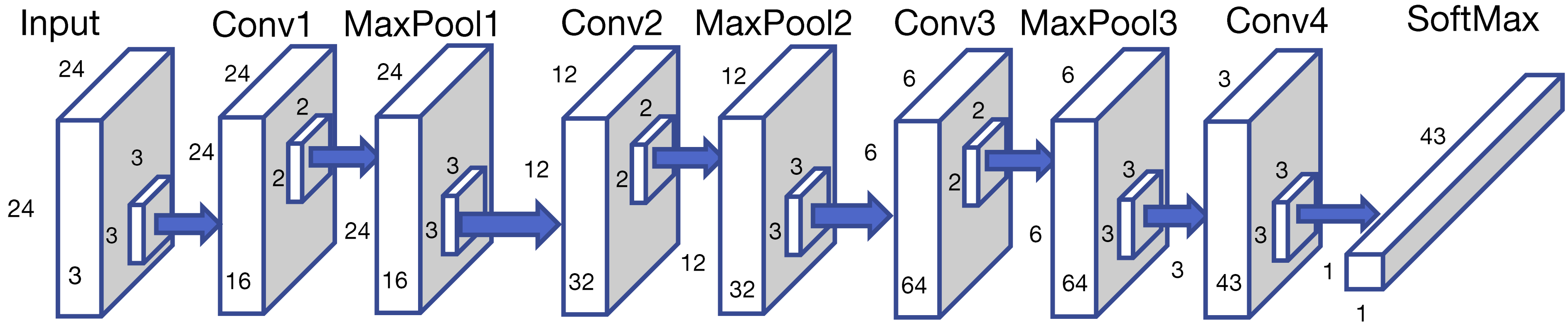
Reduced complexity



Model FLASH	2 M → 195 KB	MAC Operations	~2M
Minimum RAM	74 KB	Weights	488K→ 48K
Minimum Cache	2 KB	Layers	9 → 8
Optimum Cache	74 KB	Accuracy	94.9 → 92.5%

(example for 32-bit float implementation)

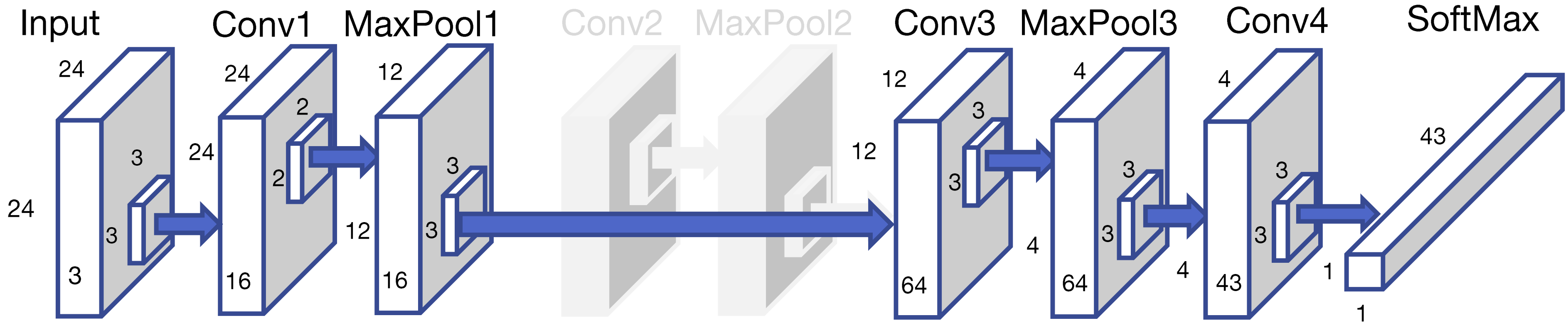
Fully Convolutional Network (FCN)



Model Size FLASH	109 KB	MAC Operations	~2 M
Minimum RAM	72 KB	Weights	26 K
Minimum Cache	2 KB	Layers	9
Optimum Cache	74 KB	Accuracy	93.3%

(example for 32-bit float implementation)

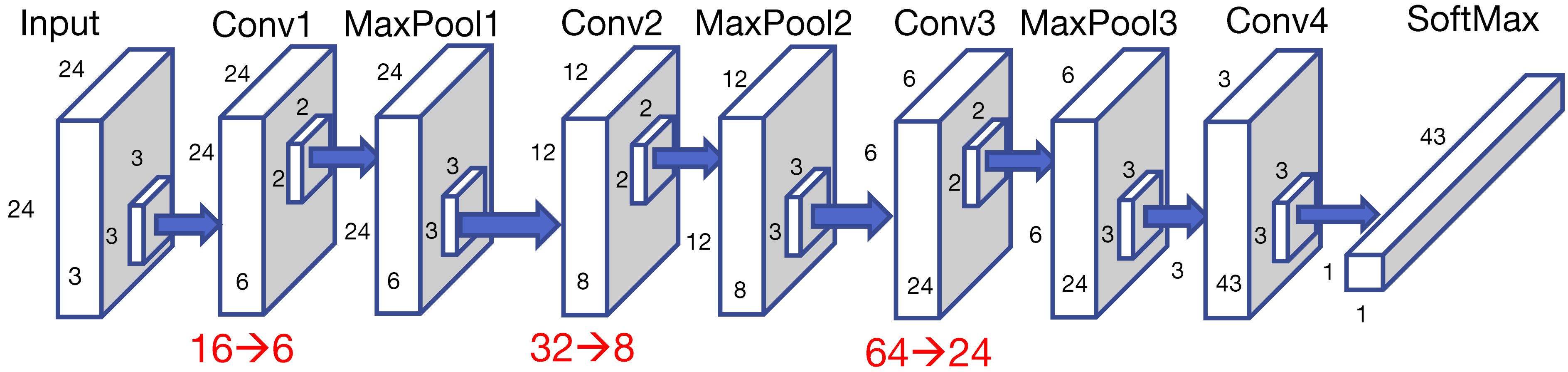
Convolutional Network



Model FLASH	109 →54 KB	MAC Operations	~2M
Minimum RAM	72 KB	Weights	26K→13K
Minimum Cache	2 K→.57 KB	Layers	9→7
Optimum Cache	74 K→37 KB	Accuracy	93.3 → 92.7%

(example for 32-bit float implementation)

Convolutional Network FCN_20



Model FLASH	109 → 20 KB	MAC Operations	~2M→ 222K
Minimum RAM	72 → 28 KB	Weights	26 → 4K
Minimum Cache	2 → 0.2 KB	Layers	9
Optimum Cache	74 → 8 KB	Accuracy	93.3 → 91.5%

(example for 32-bit float Implementation)

Target Implementation

Implementation

How to test and deploy on target hardware?

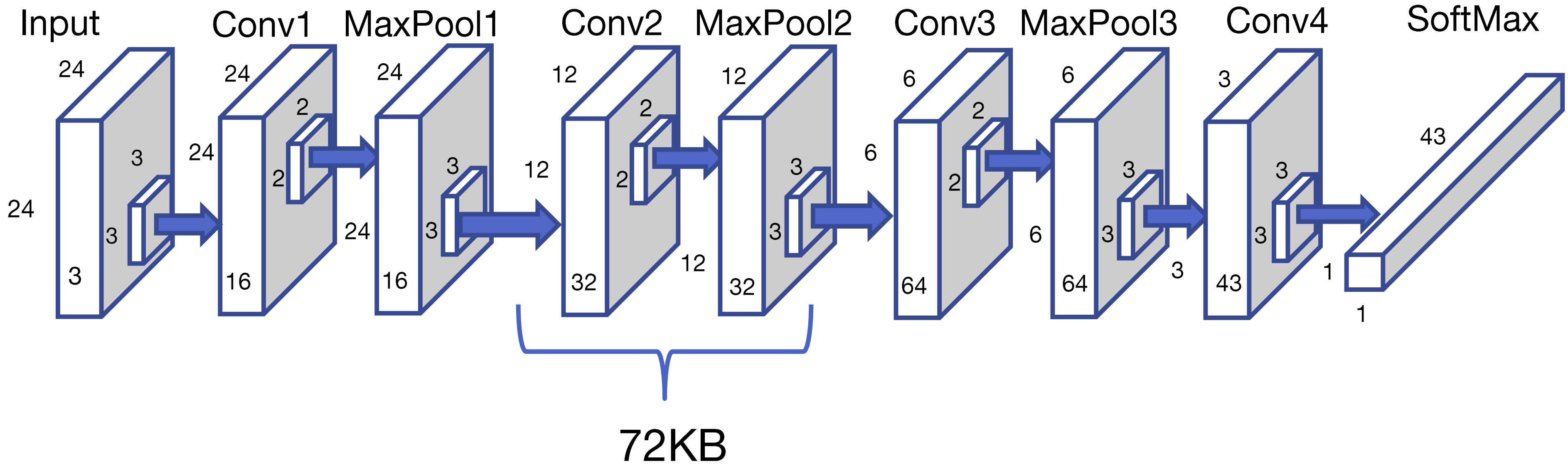
Requires optimized inference implementation

Vendor framework with generic network support	Custom C code for selected network
• Design flow to benchmark and iterate quicklywith different network designs • Profile tools for fitting designs • Support for network update	• High level of optimization is possible • Additional effort to prototype and implement

Challenges to address

- Fixed memory allocation at compile time
- Available contiguous memory on target microcontroller?
- Buffer management for best RAM utilization
- Efficient image sensor pipeline
- Validation of accuracy on the target

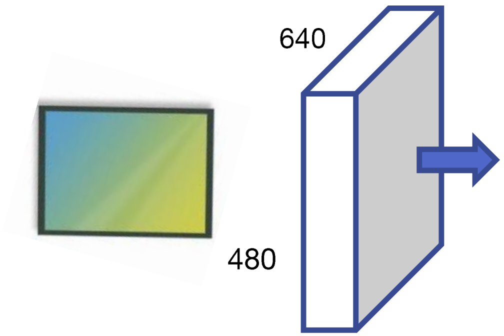
Fully Convolutional (FCN) – RAM Optimization



Minimum RAM is determined by largest layer for linear network

Non-linear networks require RAM buffer management

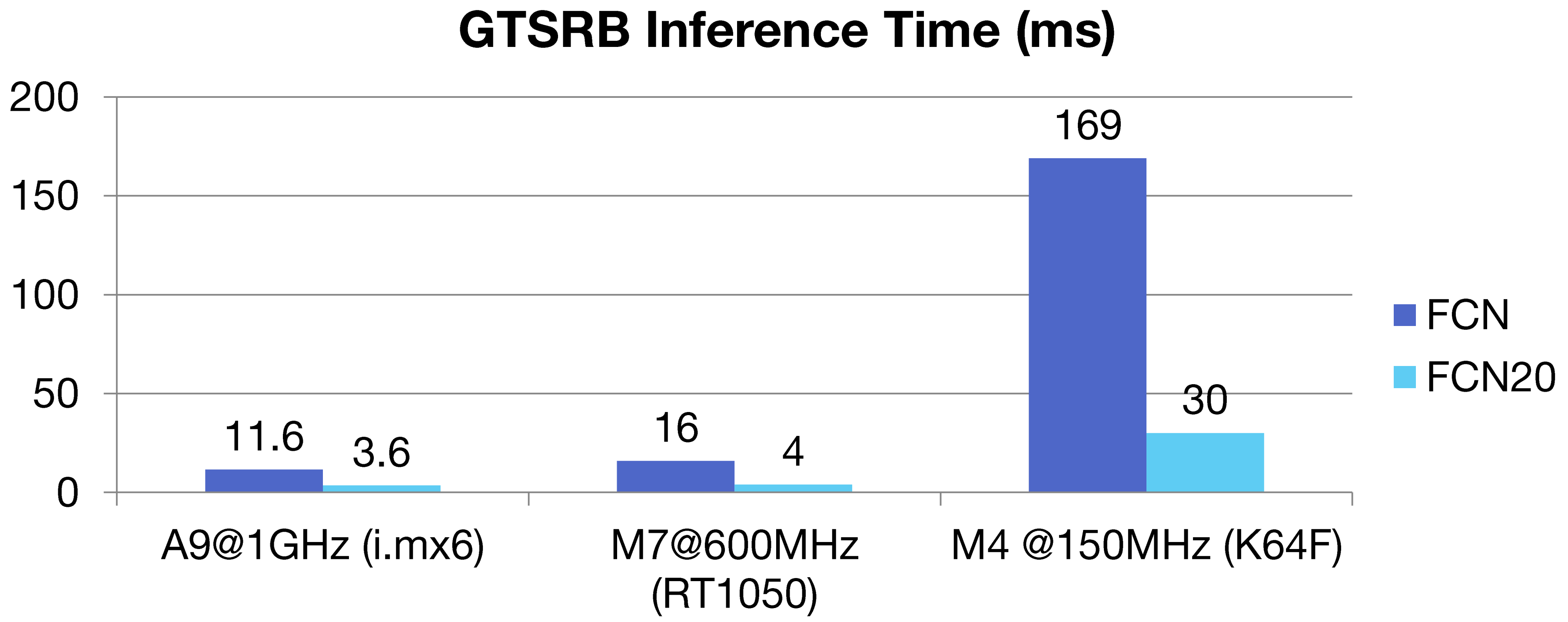
Image Sensor Pipeline



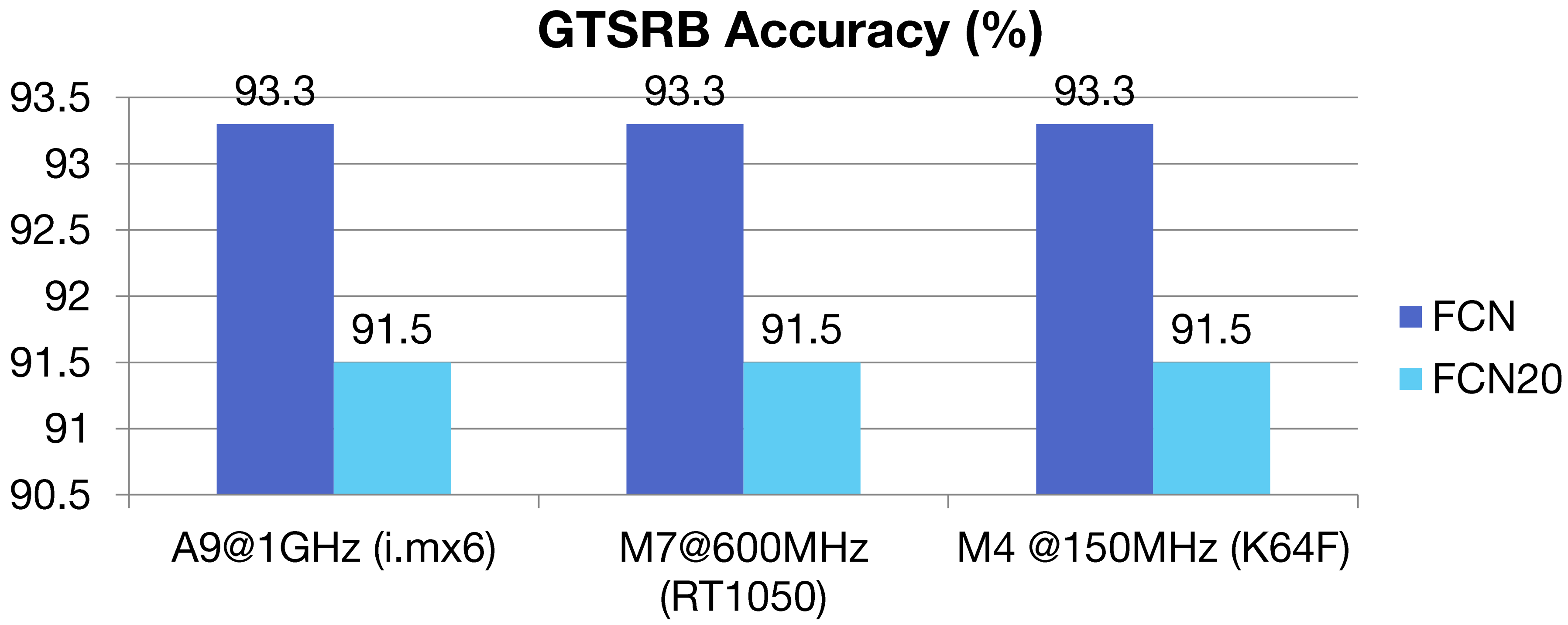
Sensor	H	V	Pixels	16bit (RGB565)
VGA	640	480	307 K	614 KB
QVGA	320	240	77 K	154 KB
QQVGA	160	120	19K	38 KB

- Use Binning or Scaling to reduce Image sensor buffer
- Use DMA to minimize slow memory buffer copies
- Use a monochrome sensor if color is not required for application

Benchmark Results



Benchmark Results

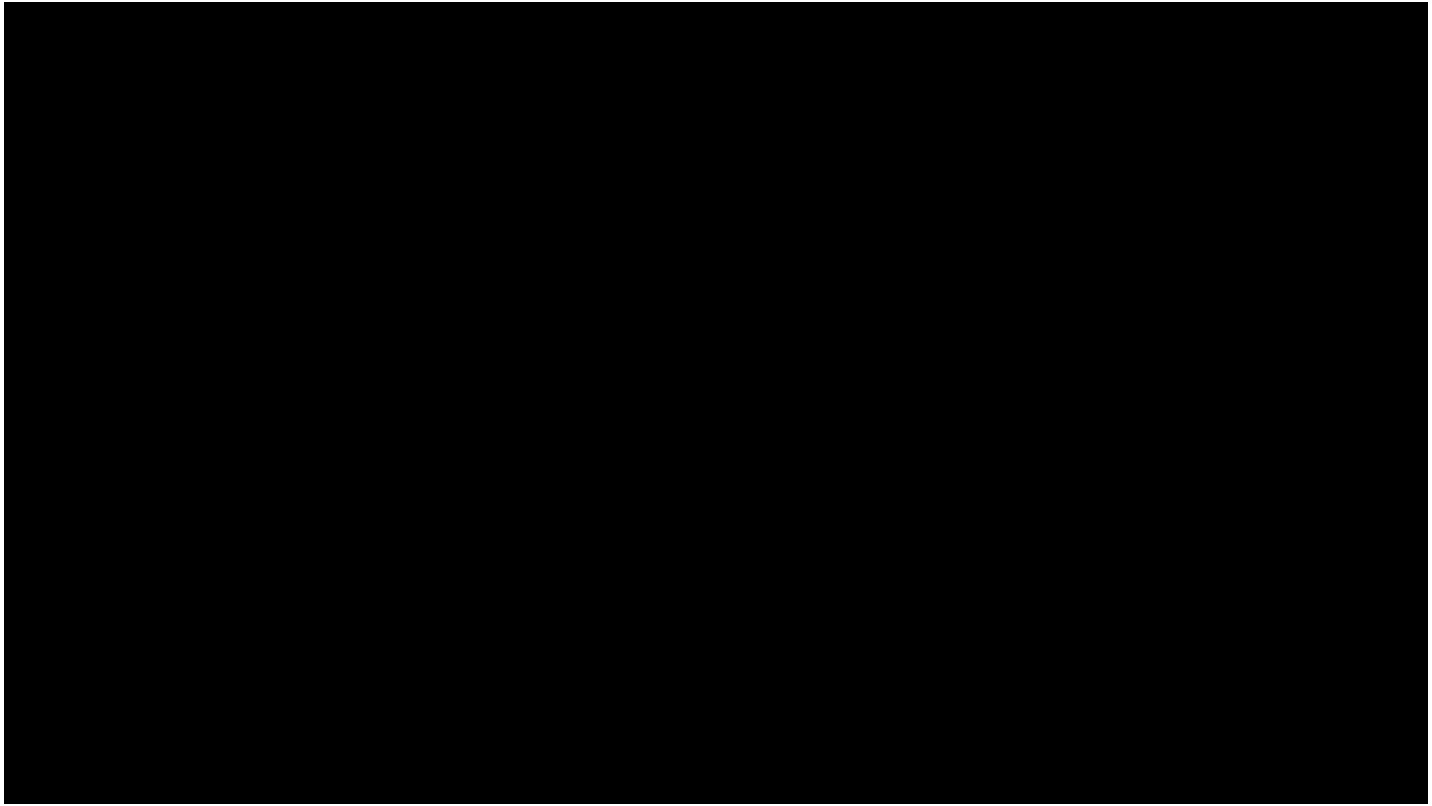


Areas for further Optimization

- Implement networks with 8-bit /16-bit arithmetic for devices that do not have hardware FPU support

- Cloud integration for distributed edge and network-based solutions where connectivity is practical

Validation on target demo



Conclusion

1. IoT-embedded vision using CNNs is practical on very low-cost microcontrollers
2. An appropriate network design is required for design and target fit
3. Memory optimization techniques should be carefully implemented
4. Specific part selection is driven by performance/accuracy and other system requirements
5. Optimized embedded inference code and tools can accelerate implementation

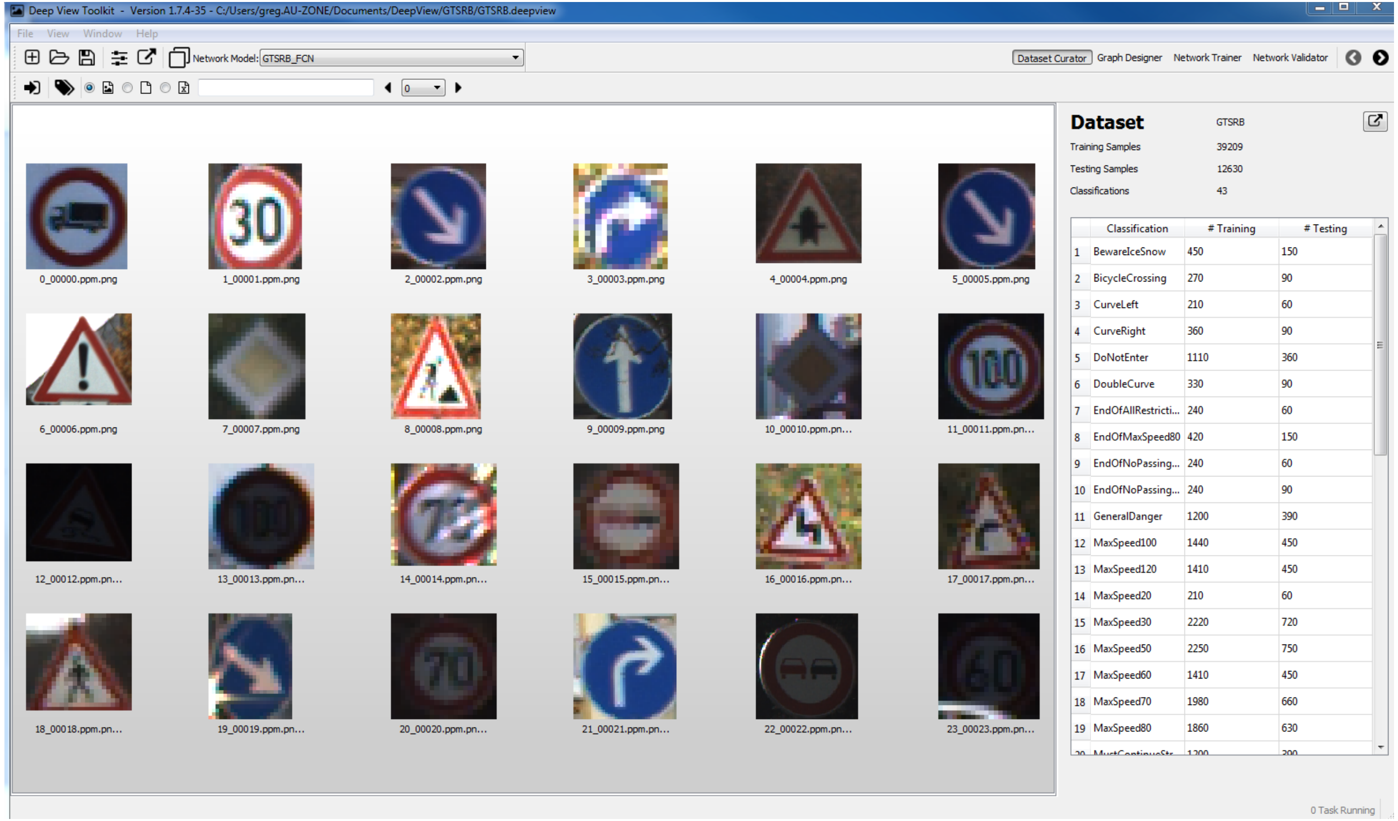
Further Resources

- Drop by Au-Zone booth 802 for further demos and details

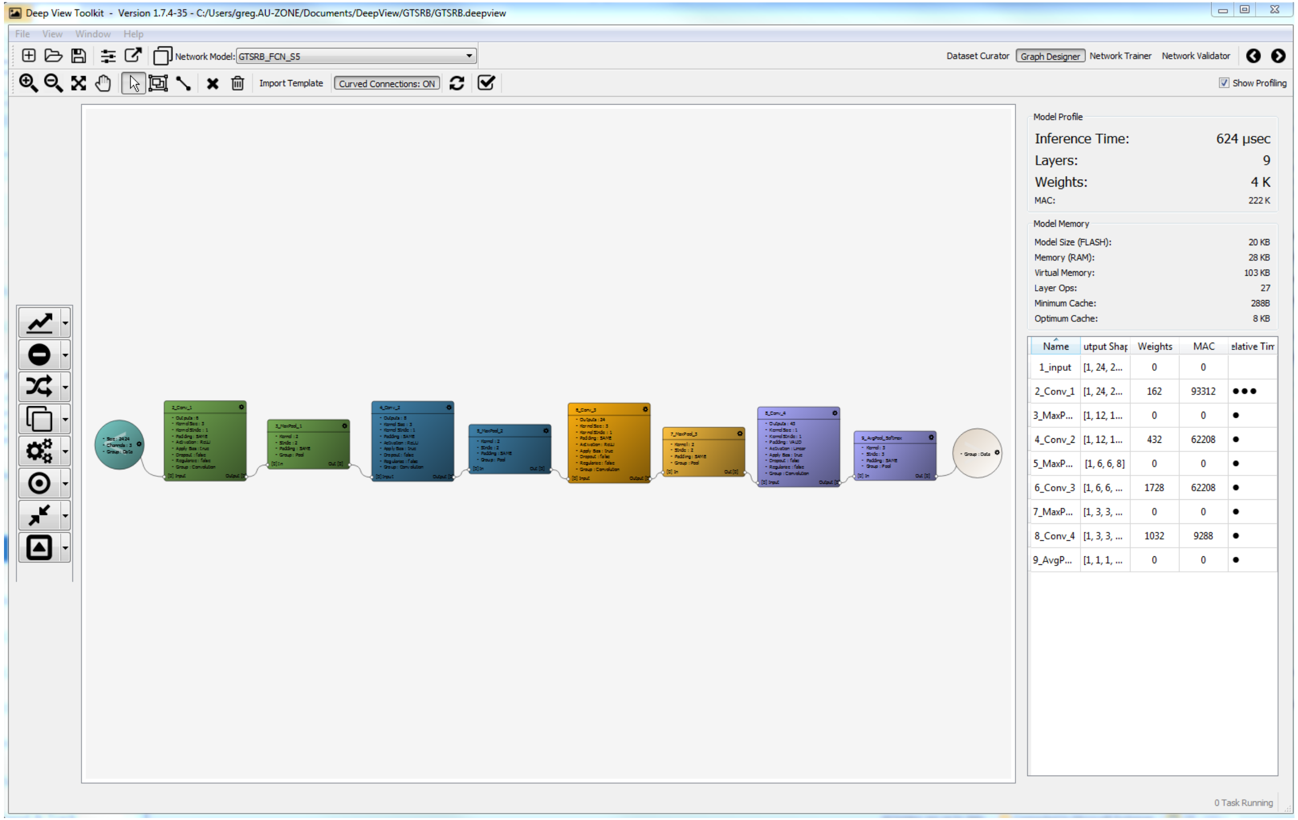
- Software Frameworks and Toolsets for Deep Learning-based Vision Processing
- Demonstration of the DeepView ML Toolkit for Embedded Platforms
- www.embeddedml.com/deepview
- ARM CMSIS NN
- NXP i.MX RT Series: Crossover Processor
- NXP Kinetis® K Series:M4 Core MCU

Backup Slides

Data Set Curator and Augmentation

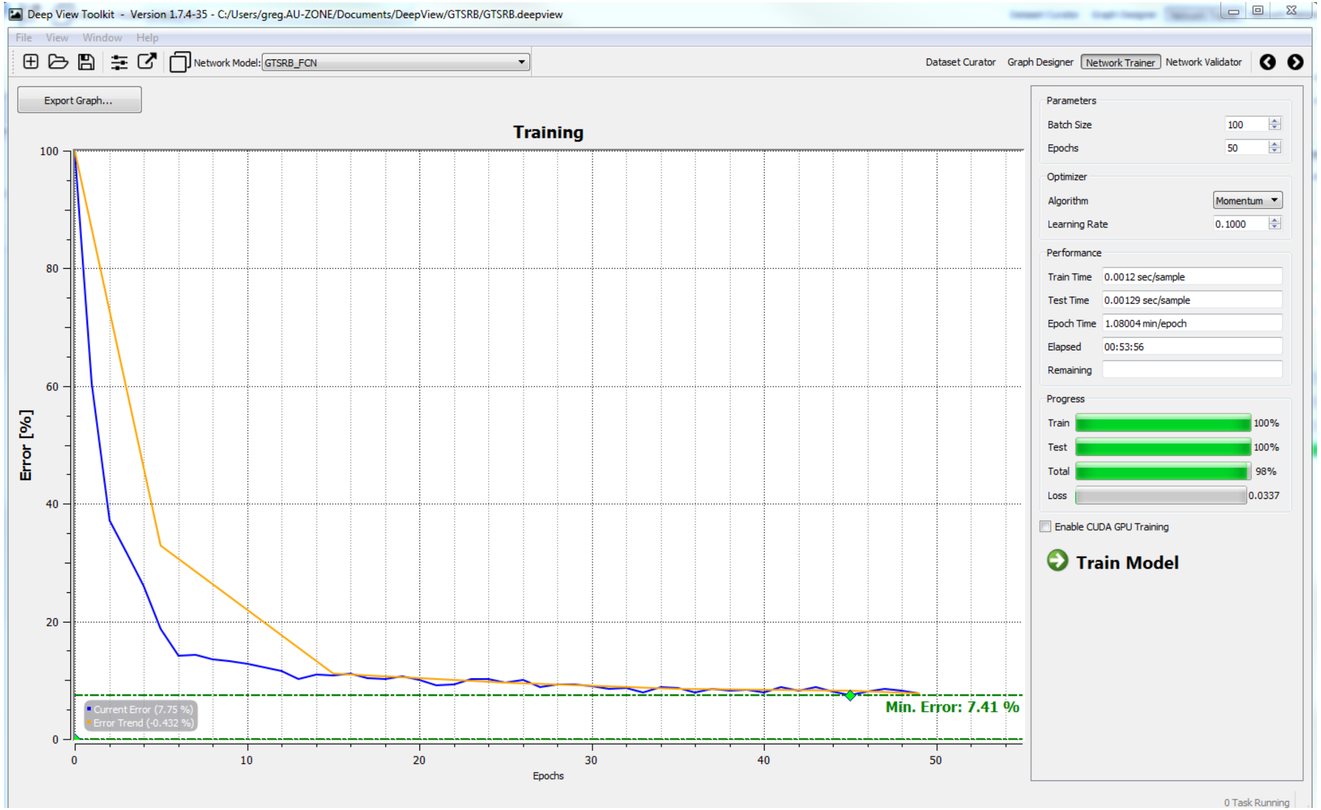


Network Design with profile details

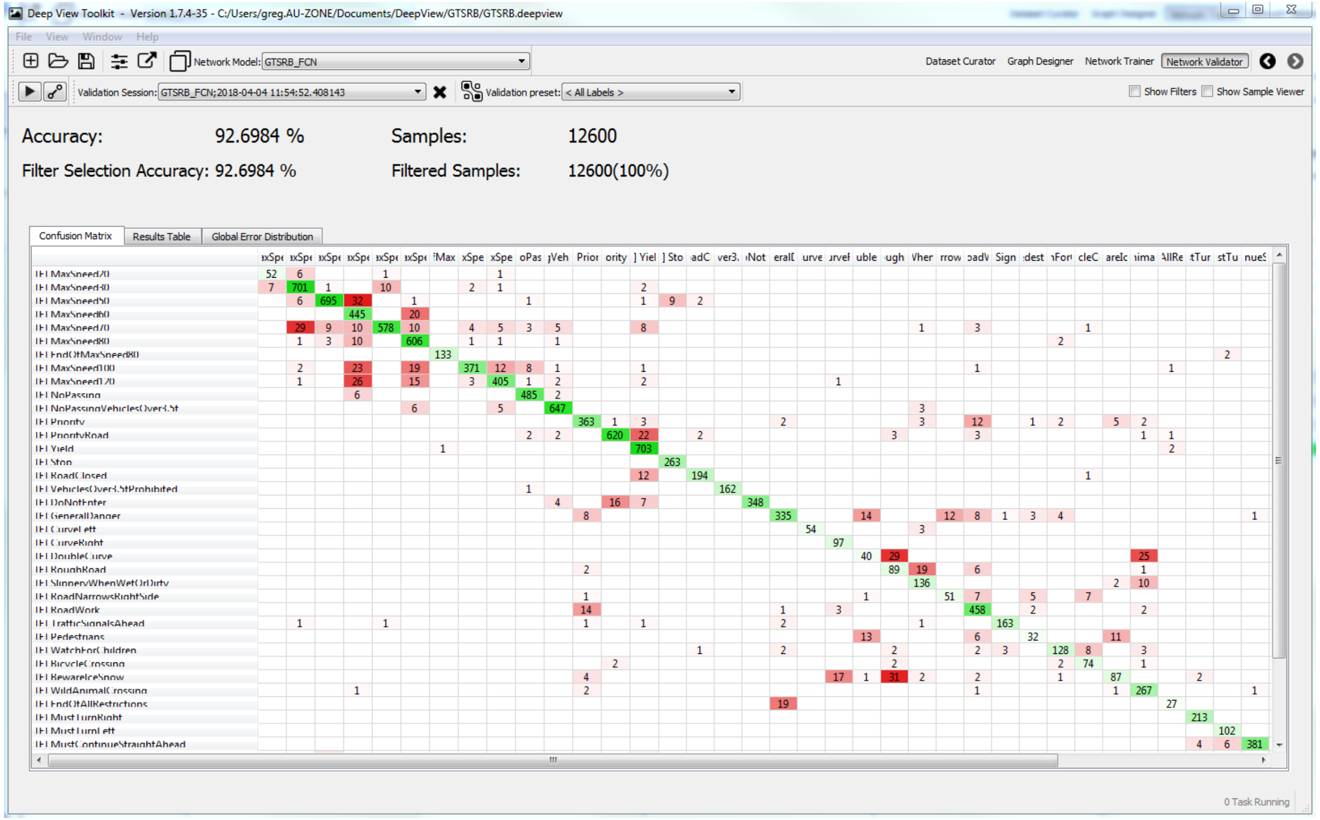


Au-Zone Technologies Inc.

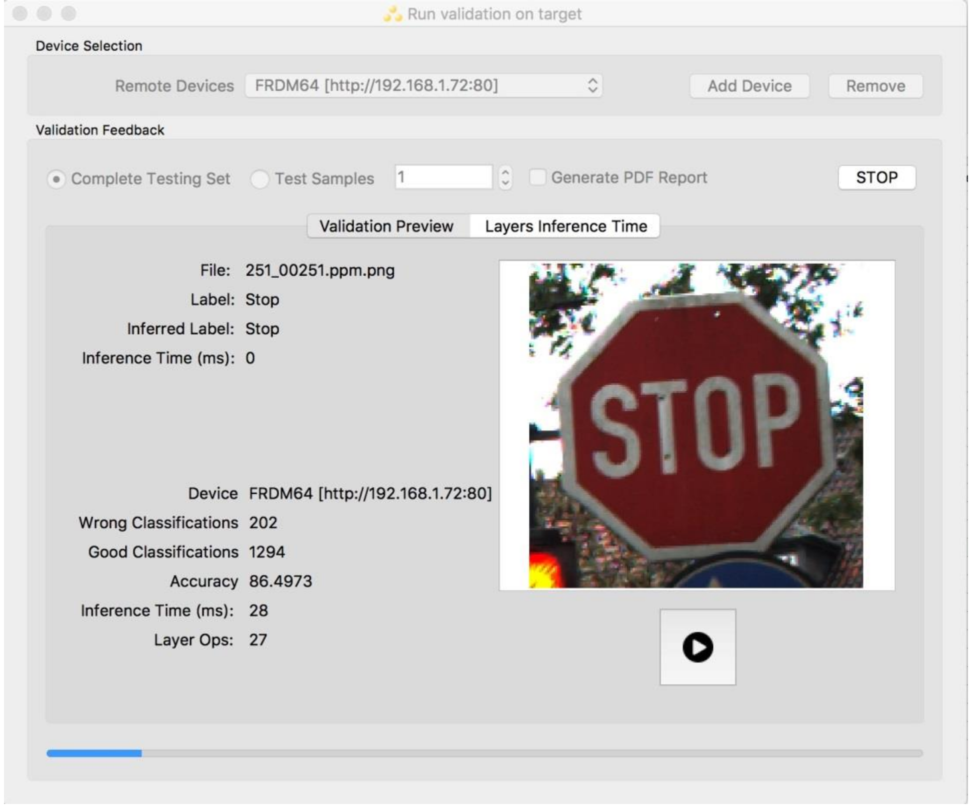
Network Training



Network Validation on Host



Network Validation on Target



Network Profile on Target