Deploy the AI/ML model in Microchip FPGA

A Leading Provider of Smart, Connected and Secure Embedded Control Solutions

Allen Chang

Software overlay based deployment of ML solutions in PolarFire FPGAs

Agenda

- **Microchip FPGA**
- **Smart Embedded Vision**
- **VectorBlox SDK**
- **VectorBlox Solutions for**
  - PolarFire FPGAs
  - PolarFire SoC FPGAs
- **CoreVectorBlox IP**
- **Roadmap**

Microchip FPGA Portfolio

The World's Most Power-Efficient FPGAs

MICROCHIP

Everyone is Moving Intelligence to the Edge

Gartner

"**The Intelligent Edge exists** when real-time insights and decisions happen outside the data center."

5G & Data Networks

Industrial Edge

AI/ML in the IoT

Medical Imaging

Autonomous Vehicles

Space Compute

Aerospace & Defense

Gartner

"**75% of enterprise-generated data** will be created & processed outside a centralized data center or cloud by 2025."

Gartner

"**ML algorithm growth** will be especially strong at the intelligent edge."

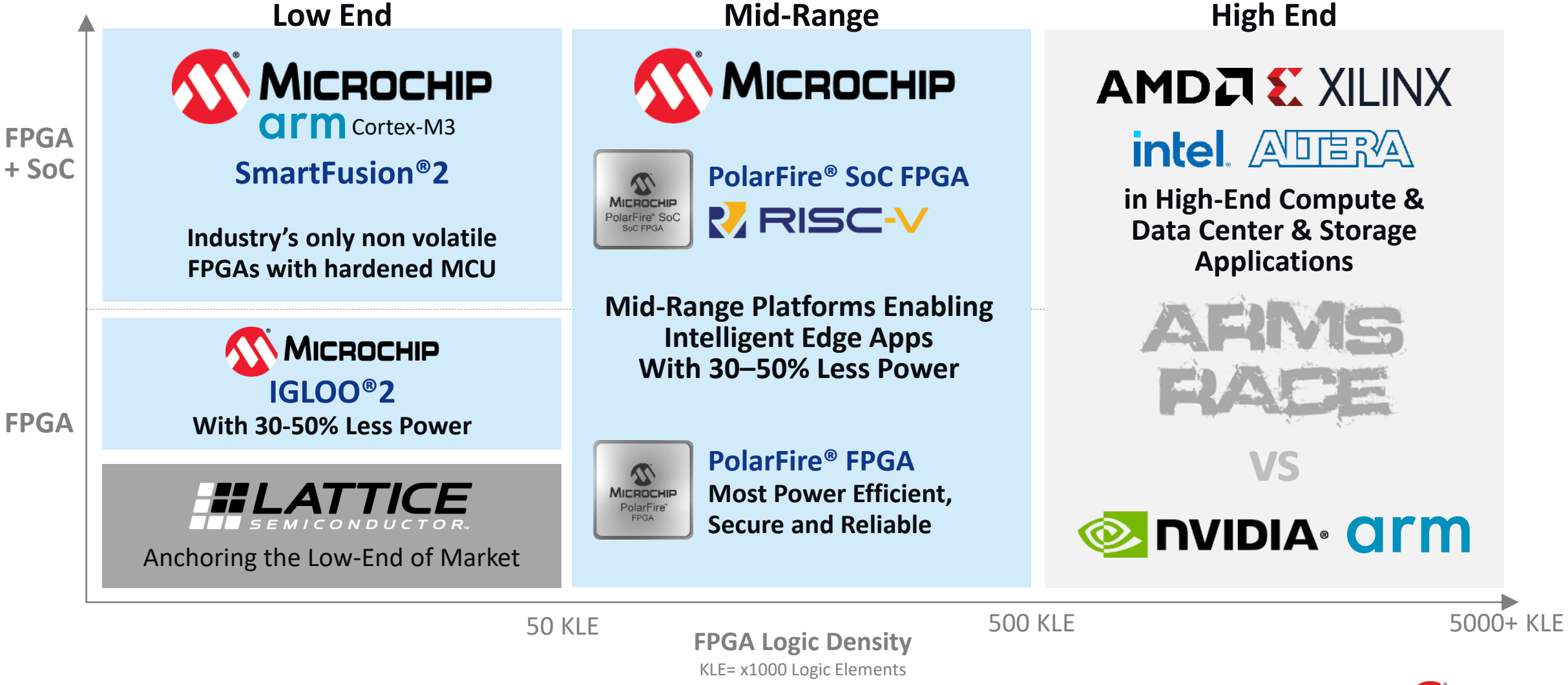
BAIN & COMPANY

"The market for intelligent edge computing is as much as **$127 billion** in spending on embedded silicon forecast through 2027."

MICROCHIP

Microchip FPGA Leadership

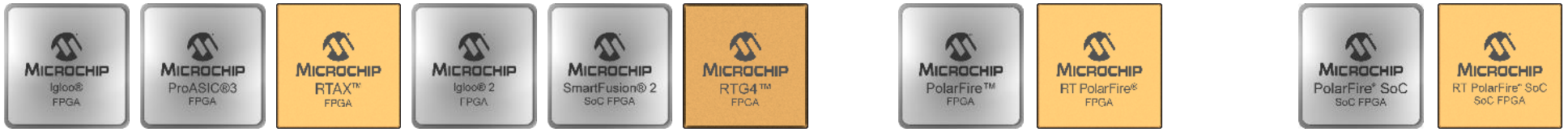
Most Power Efficient, Secure Mid-Range FPGAs



MICROCHIP

From the Intelligent Edge to the Depths of Space

Power & Thermal Efficiency, Pervasive Security, Exceptional Reliability



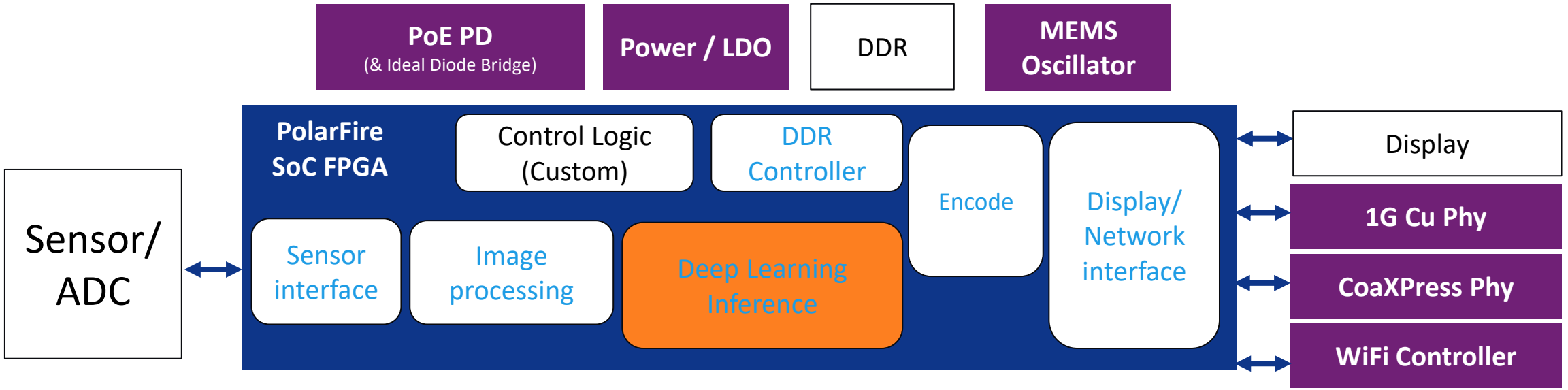
Family →	SmartFusion, ProASIC3, IGLOO	SmartFusion2, IGLOO2	PolarFire	PolarFire SoC
Description	Smallest Packages CPLD Replacements	Low Density FPGAs with more resources & lowest power	Mid-Range Density FPGAs Lowest Power, Smallest Form Factors	Mid-Range Density SoCs Hard 5-Core RISC-V 64-bit CPU Realtime Linux
Logic Elements	100—30K	5K—150K	50K—480K	25K—460K
Transceiver Rates	—	1—5 Gbps	250 Mbps—12.7 Gbps	250 Mbps—12.7 Gbps
I/O Speeds	400 Mbps LVDS	667 Mbps DDR3 750 Mbps LVDS	1600 Mbps DDR4 1.6 Gbps LVDS / 1.5 Gbps MIPI	1600 Mbps DDR4 / LPDDR4 1.6 Gbps LVDS / 1.5 Gbps MIPI
DSP (18x18 Multipliers)	—	240	1480	1420
Max RAM	144 Kb	5 Mb	33 Mb	32 Mb
Processor Options	Hard 100 MHz ARM Cortex-M3	Hard 166 MHz ARM Cortex-M3 Soft RISC-V	Soft RISC-V Soft ARM Cortex-M1 Hard Crypto Processor	64-bit Multi-CPU Cluster Hard 5-Core RISC-V CPU (600MHz) Hard Crypto Processor / 2MB L2 Cache
On-board Flash	Up to 512 KB code store	Up To 512 KB code store	56 KB secure NVM	56 KB secure NVM / 128 KB Boot Flash

MICROCHIP

Smart Embedded Vision

MICROCHIP

SEV: A Microchip One-Stop-Shop



- **Sensor Interfaces (Rx)**
  - MIPI CSI-2 Receive (1.5 Gbps / lane)
  - SLVS-EC (4.7 Gbps / lane)
  - JESD204B (12.5G/lane)
  - LVDS (1.6G/lane)
- **Display Interfaces**
  - MIPI Tx (1.0 Gbps/ lane)
  - MIPI Tx (2.5 Gbps/ lane)
  - MIPI DSI Tx (1080p60)
  - MIPI DSI Tx (40k60)
- **Image Processing**
  - Basic ISP: Color Space, Image Enhancement, Edge Detection, Exposure Control…
  - Advanced: Defect Pixel Correction, Histogram
- **Soft DDR Controllers**
  - DDR4/ DDR3/ LPDDR3
  - LPDDR4 (characterization)
- **Deep Learning Inference**
  - VectorBlox SDK v1.4
- **Encode (Decode)**
  - H.264 (1080p60)
  - H.264 (4K60)
  - JPEGXS
  - mJPEG Encode, mJPEG Decode
- **Transport Interfaces**
  - CoaXPress 6.25 / 12.5G
  - SDI (HD/3G/12G)
  - 10G MAC / 10G PHY
  - USXGMII (1/2.5/5/10G)
  - HDMI 2.0 Rx (4K30) / Tx (4K60)
  - DisplayPort 1.4 Tx (4K60)
  - USB 2.0, USB3.1 Gen1 & Gen2
  - Aurora 8/10b, 64/66b

In Development / 3rd Party

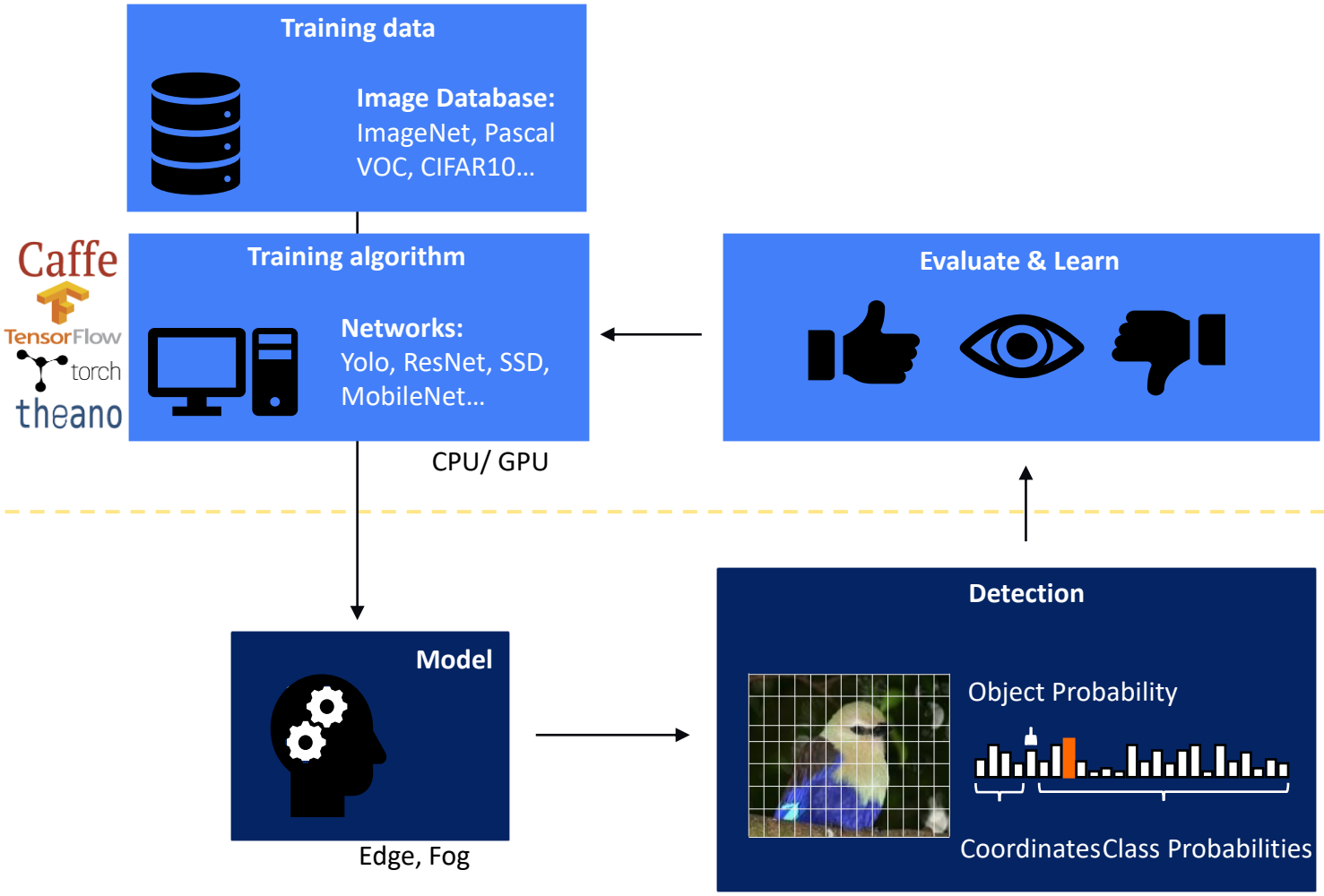
MICROCHIP

What does VectorBlox Enable?

PolarFire Video Kit

Smart Camera

The Deep Learning Construct



Network Training

- In the data centre or workstation
- Large compute capacity
- No power or space constraints

Inference

- Need Low Latency
- Power & space constrained
- Requires security and reliability

Many Challenges in FPGA Development

Models developed in the software domain

Network

RTL

VHDL VERILOG

Translating models into RTL requires expertise

Many frameworks used by AI/ML developers

TensorFlow

MathWorks

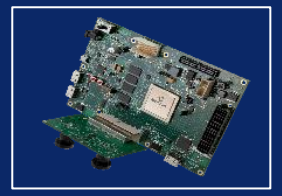
ONNX

Caffe2 PYTORCH Chainer

mxnet CNTK PaddlePaddle

Custom Implementation for each framework

Need out-of-box evaluation platforms



Smart Camera

PCIe Interface

Learning curve for new developers

Enable evaluation before choosing hardware

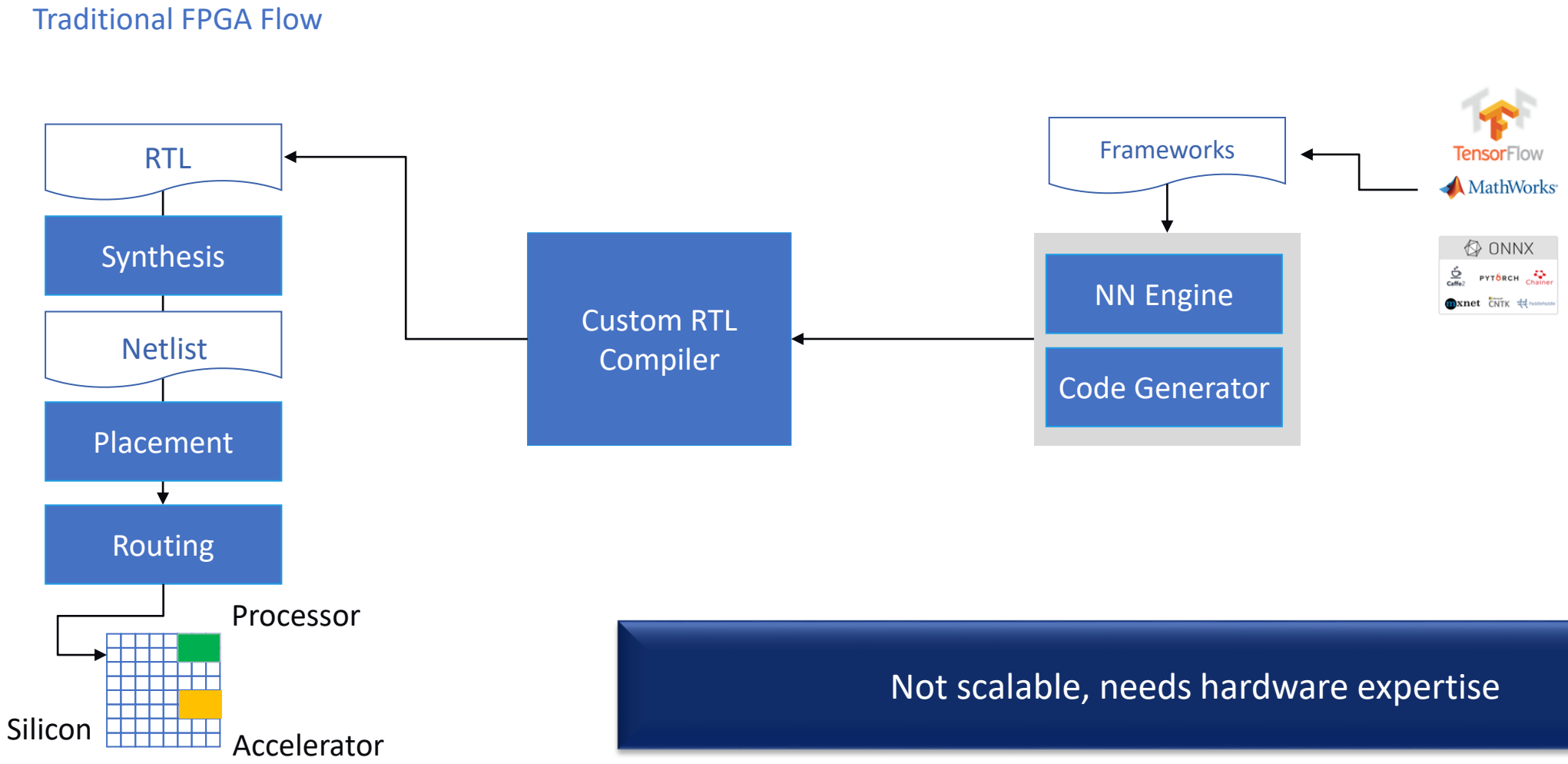
Network

Simulator

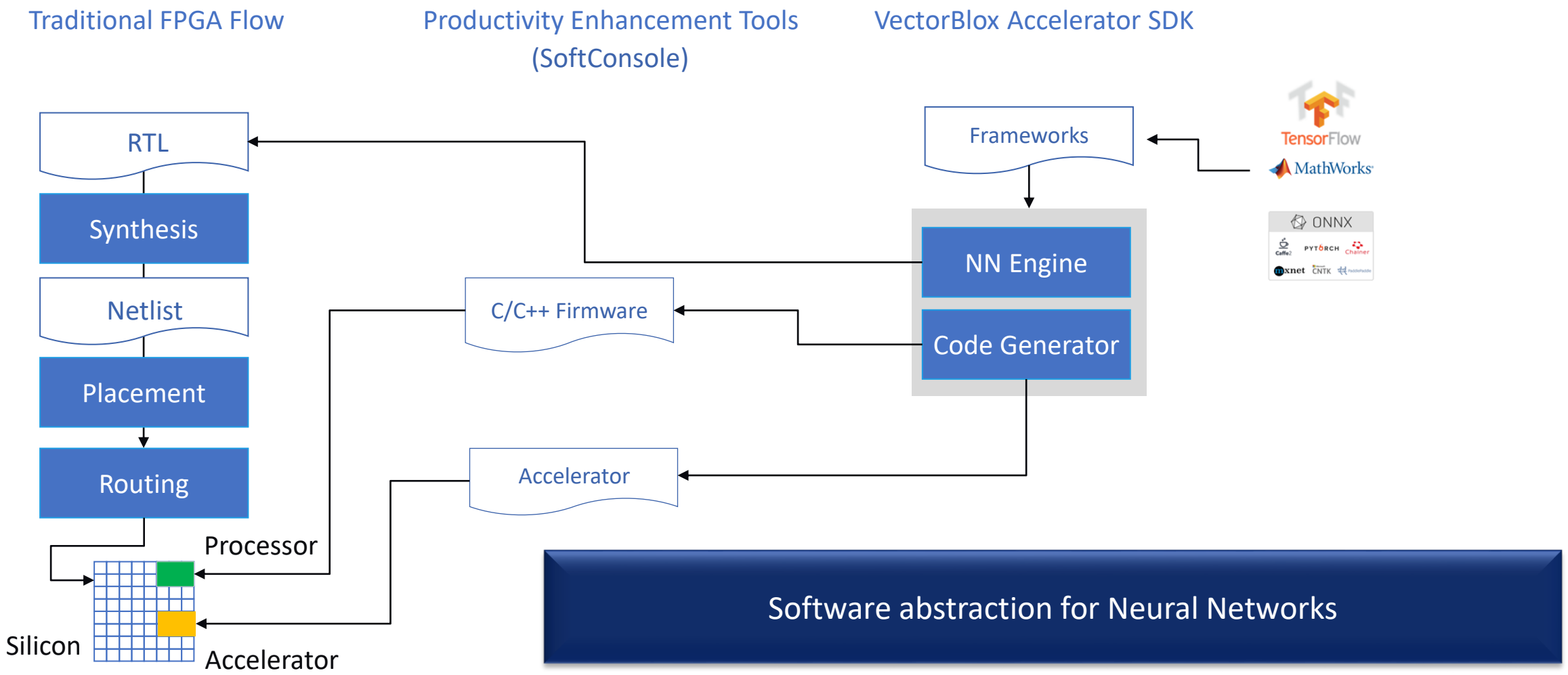
FPGA

Validate CNNs before investing in hardware

Traditional FPGA Design Flow Challenges



Traditional FPGA Design Flow Challenges

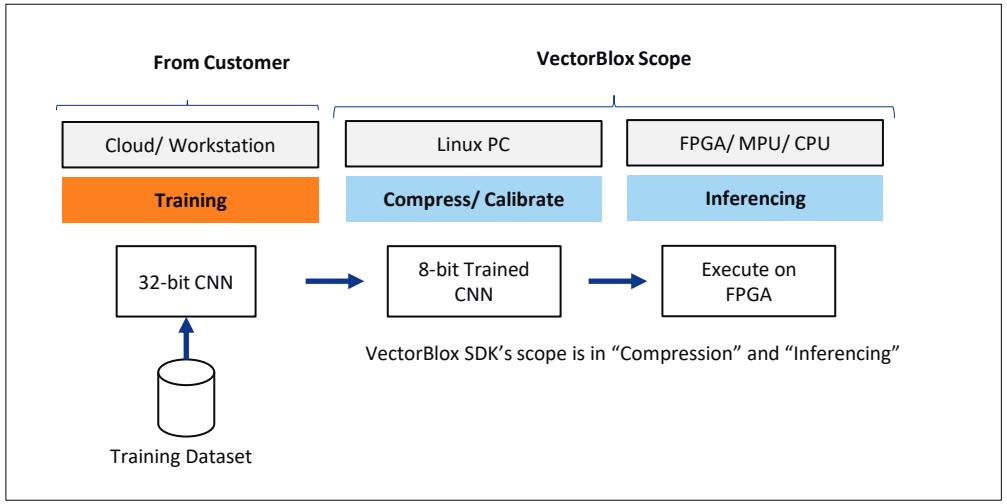


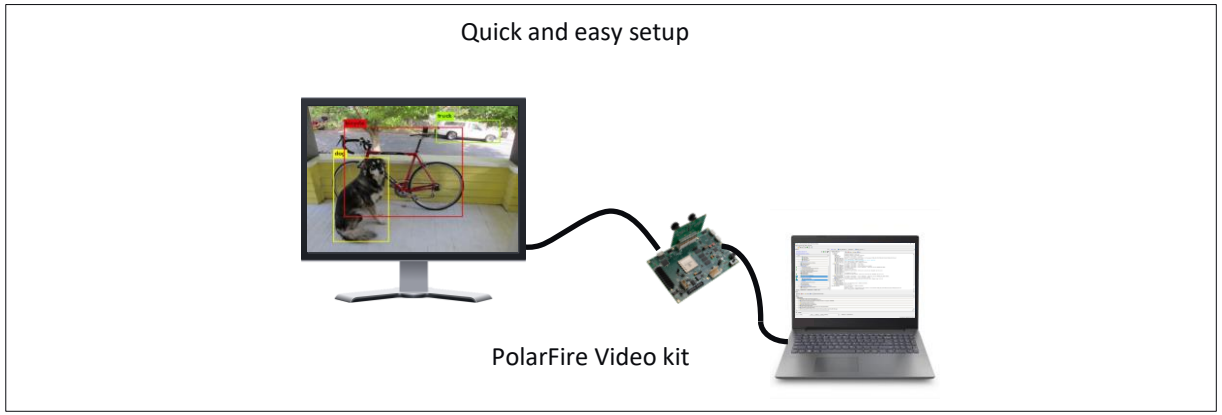
MICROCHIP

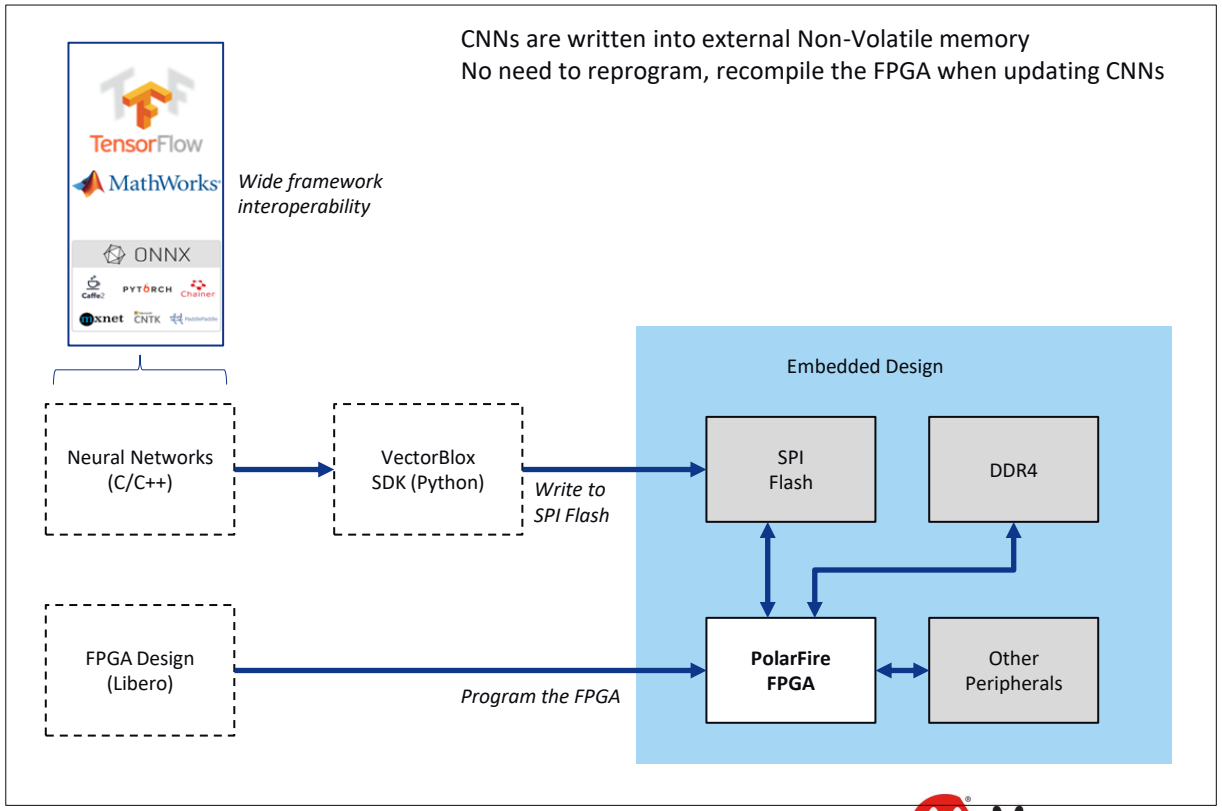
VectorBlox in Short

VectorBlox SDK and NN IP enables

- Software developers to run Neural Networks (NN) without prior FPGA knowledge
- Utilization of most popular NN software frameworks
- Simulation in software without procuring hardware





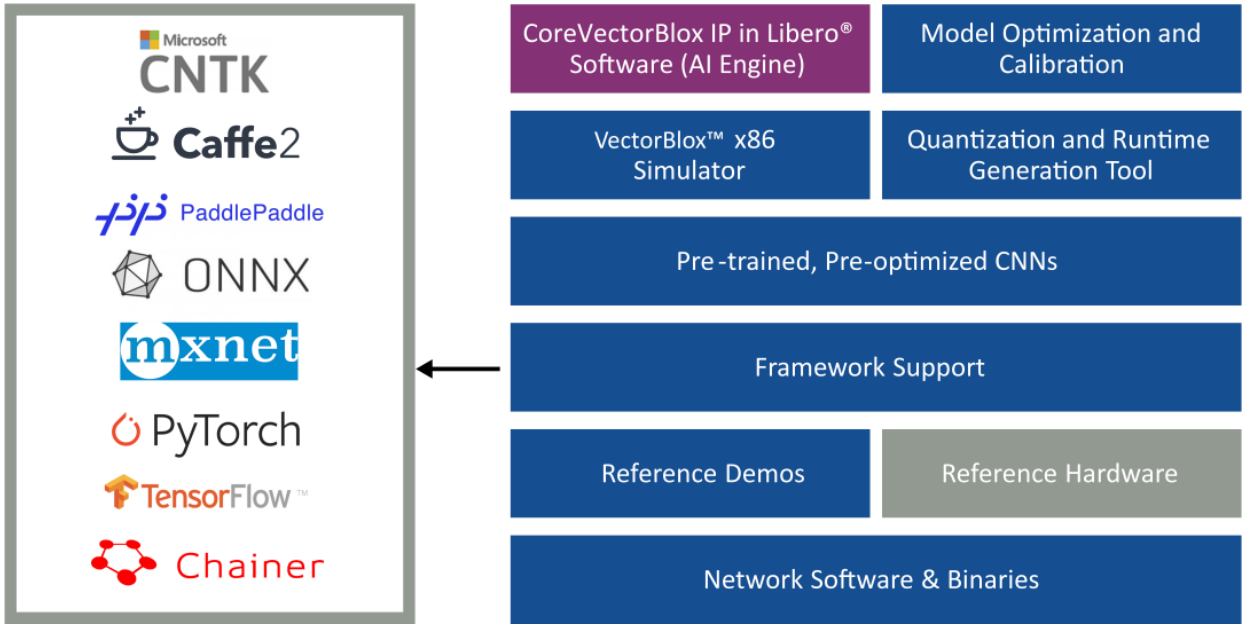


VectorBlox Software Development Kit

VectorBlox SDK is supported in Ubuntu & WSL

VectorBlox™ Accelerator SDK

- Enables developers to code in C/C++ and use power efficient neural networks
  - No prior FPGA design experience required.
  - Works on Linux and Windows.
- Executes models in TensorFlow and ONNX
  - Offers the widest framework interoperability
- Includes a bit accurate simulator
  - Users can validate the accuracy of the hardware while in the software environment.
- Pre-trained Neural Networks demos included
  - Users can load different network models at run time on supported hardware



MPF300-VIDEO-KIT-NS

MPFS250-VIDEO-KIT

MICROCHIP

*demo only

VectorBlox™ Accelerator SDK

CoreVectorBlox IP in Libero® Software *(AI Engine)*

Model Optimization

VectorBlox™ Simulator

Calibration and Runtime Generation

Pre-trained, Pre-optimized CNNs

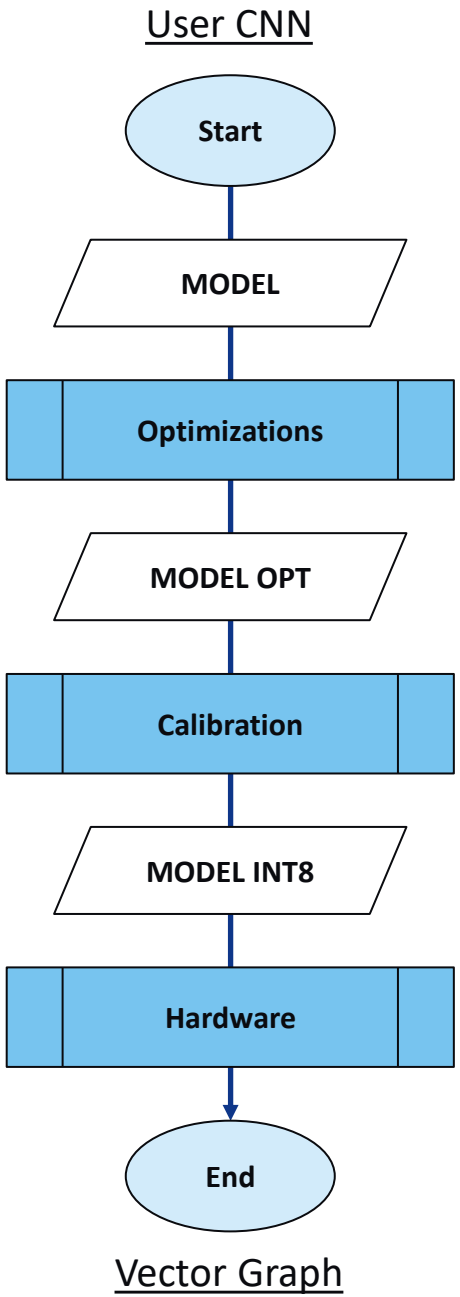
Framework Support

Reference Demos

Reference Hardware

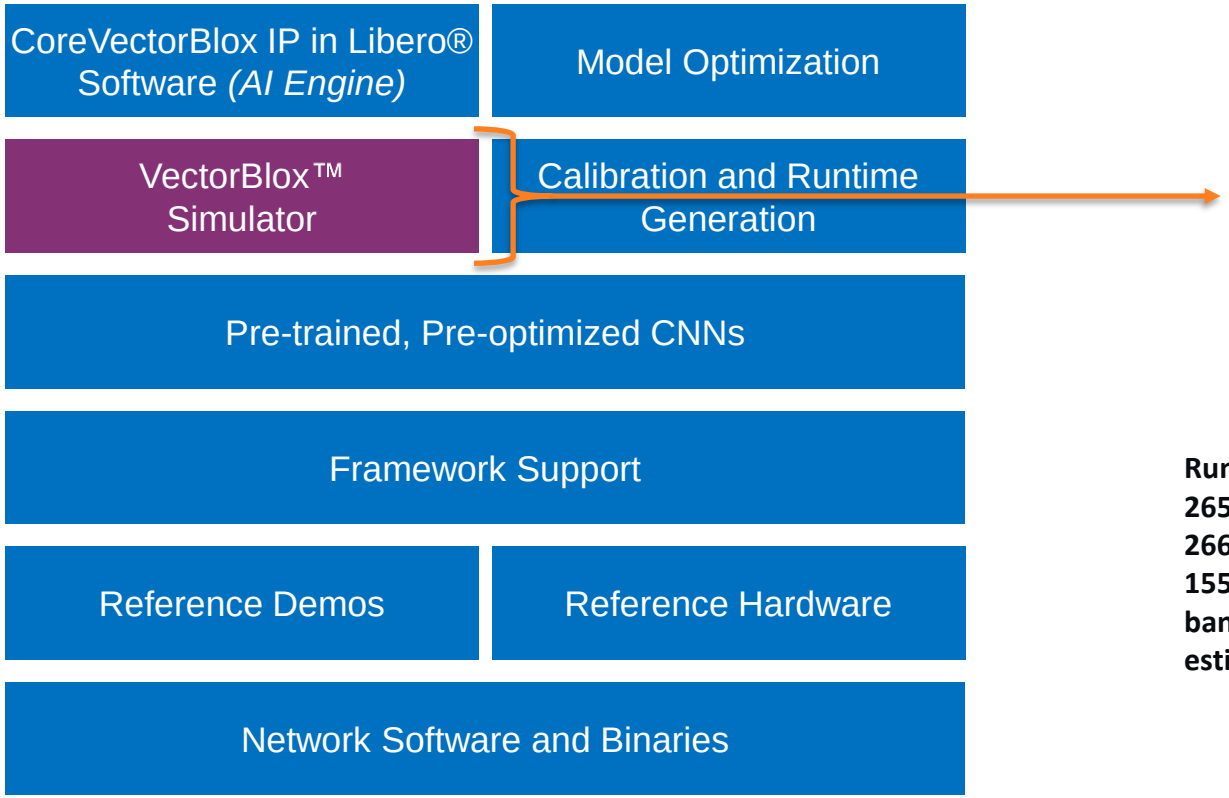
Network Software and Binaries

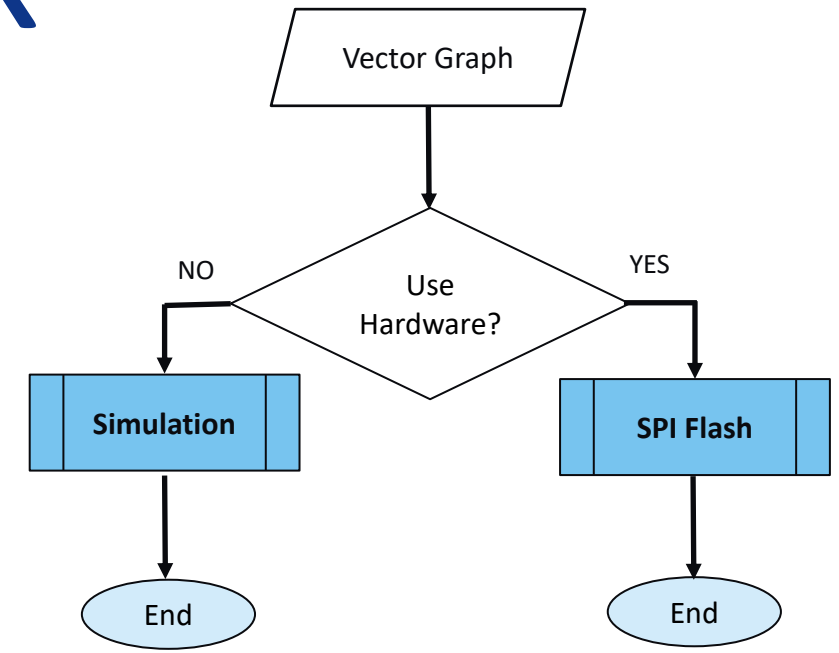
Model Optimization and Calibration



MICROCHIP

VectorBlox™ Accelerator SDK



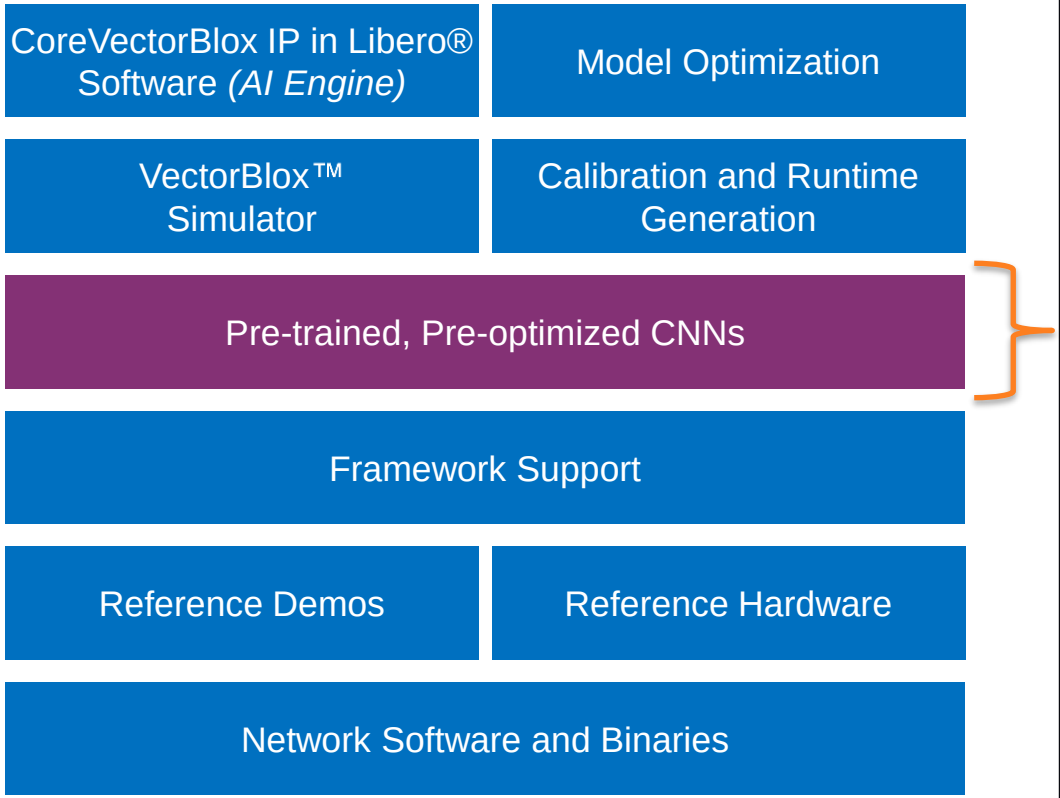


Running Simulation...
265 toy poodle 0.6151123
266 miniature poodle 0.11071777
155 Shih-Tzu 0.07495117
bandwidth per run = 32002180
estimated 0.0400075 seconds at 100MHz

Using the Simulator

MICROCHIP

VectorBlox™ Accelerator SDK

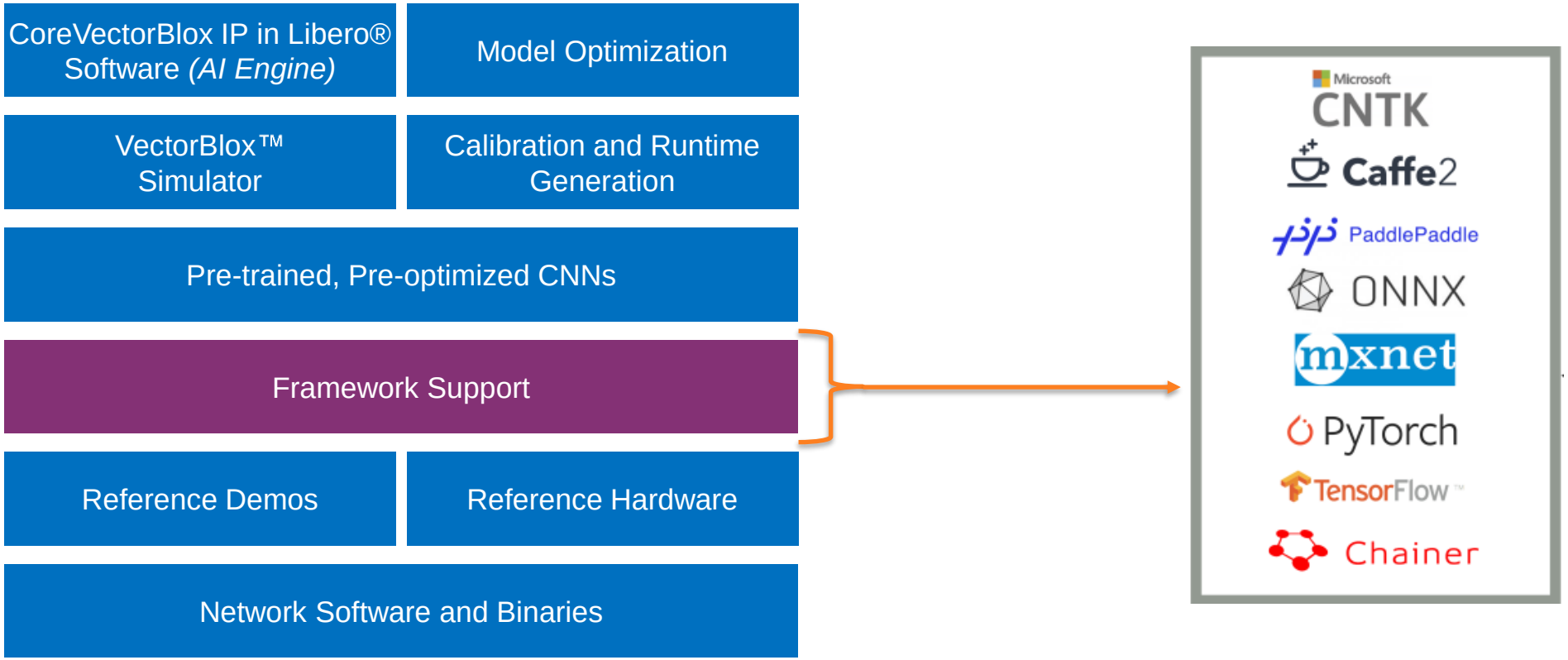


CNNs are supported with Tutorials

Tutorial Name	Source Framework	Task	Accuracy Metric	Accuracy Score	V1000 kcycles
mobilenet-v1-1.0-224	caffe	Classification	topk	65.4	1623
mobilenet-v2	caffe	Classification	topk	70	2037
mobilenet-v1-1.0-224-tf	tensorflow	Classification	topk	68.8	1613
mobilenet-v2-1.0-224	tensorflow	Classification	topk	69	1710
mobilenet-v2-1.4-224	tensorflow	Classification	topk	74.8	2625
resnet-50-tf	tensorflow	Classification	topk	75.2	8786
squeezenet1.0	caffe	Classification	topk	58.2	2326
squeezenet1.1	caffe	Classification	topk	58.4	1301
Sphereface	caffe	face_compare			3784
onnx_resnet18-v1	onnx	Classification	topk	71.4	3304
mnist	onnx	Classification			21
torchvision_resnet50	pytorch	Classification	topk	75.2	8909
torchvision_wide_resnet50_2	pytorch	Classification	topk	76.2	
yolov2-tiny-voc	darknet	Object Detection	VOCmAP	54.06	8907
yolov2-tiny	darknet	Object Detection	COCOmAP	27.02	9056
yolov3-tiny	darknet	Object Detection	COCOmAP	39.71	10147
yolov2-voc	darknet	Object Detection	VOCmAP	69.99	
yolov3	darknet	Object Detection			
yolov2	darknet	Object Detection			43905
BlazeFace	pytorch	Classification			470

MICROCHIP

VectorBlox™ Accelerator SDK



Supports TensorFlow and ONNX (Caffe, CNTK, mxnet, PyTorch etc.)

VectorBlox™ Accelerator SDK

CoreVectorBlox IP in Libero® Software *(AI Engine)*

Model Optimization

VectorBlox™ Simulator

Calibration and Runtime Generation

Pre-trained, Pre-optimized CNNs

Framework Support

Reference Demos

Reference Hardware

Network Software and Binaries

CNNs are supported with Tutorials

Smart Camera Reference Design

MICROCHIP

VectorBlox™ Accelerator SDK

CoreVectorBlox IP in Libero® Software *(AI Engine)*

Model Optimization

VectorBlox™ Simulator

Calibration and Runtime Generation

Pre-trained, Pre-optimized CNNs

Framework Support

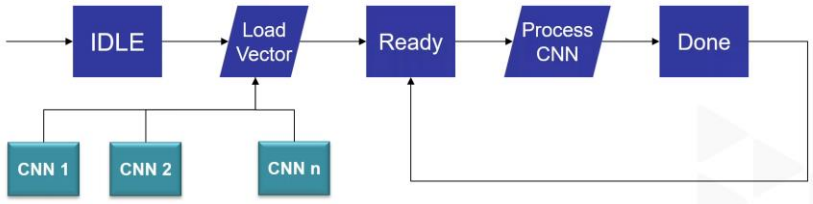
Reference Demos

Reference Hardware

Network Software and Binaries

```
struct model_desc_t models [ ] = {
                        { “MobileNet V1”, 0x259000, 224, IMAGENET, 10, 0x6926fe18, 30},
                        { “Tiny Yolo V3 COCO”, 0xed5000, 416, YOLOV3, 30, 0xf640b794, 30},
};
vbx_cnn_t* the_vbx_cnn;
int logo_width = 420;
int logo_height = 264;
unit32_t* logo = 0;

void clear_frame (uint32_t* draw_frame) {
        //clear relevant sections.
        //clear center
        draw_rectangle ((1920-1080)/2,0,1080, 1080, 0,
```



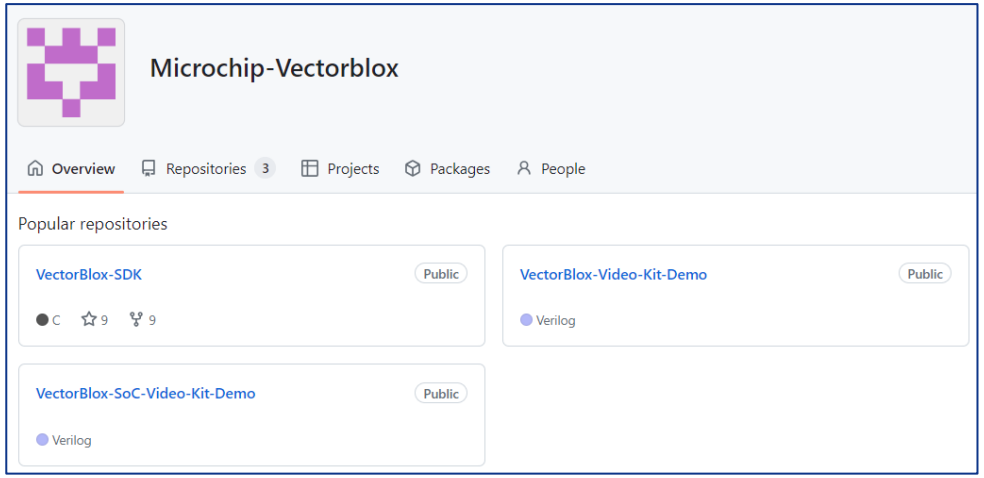
Switch CNNs without “Re-programming” or “Place and Route”

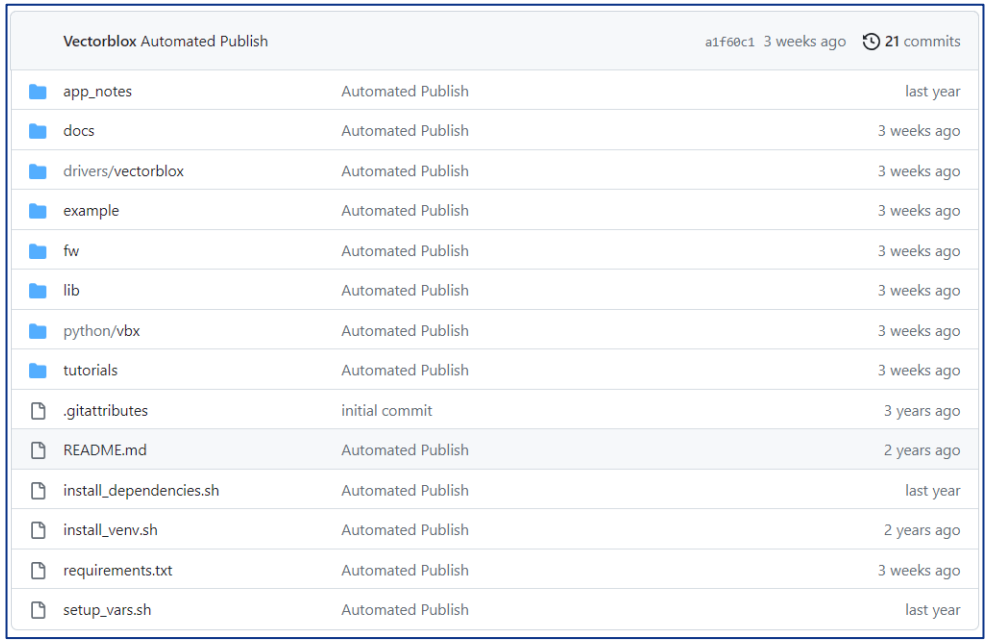
Hardware Supported in VectorBlox

MPF300-VIDEO-KIT-NS

MPFS250-VIDEO-KIT

VectorBlox GitHub And Model-Zoo





Tutorials

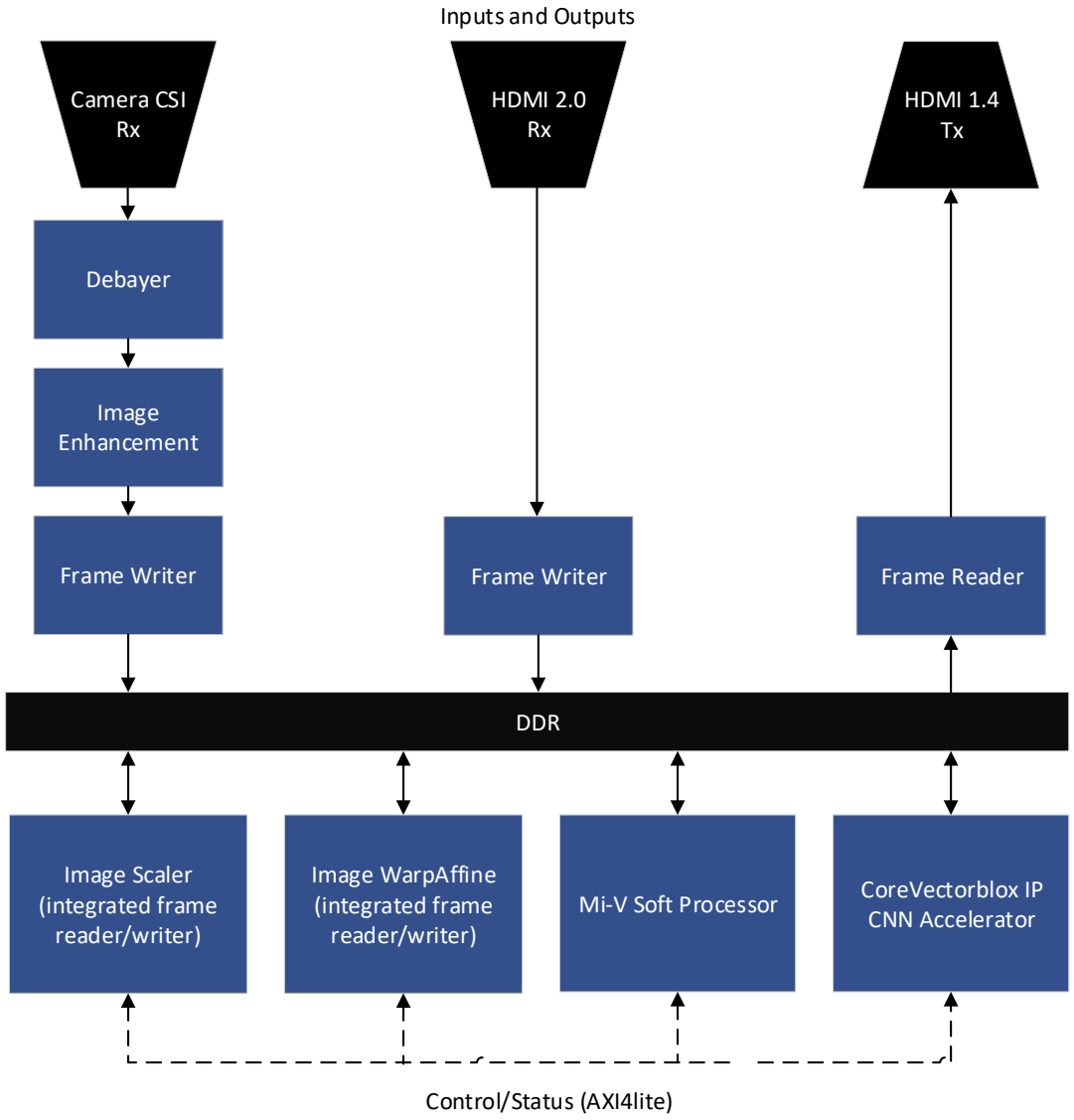
This directory contains scripts that will generate vectoblox compatible Binary Large OBjects (BLOBs) for networks from various sources. The scripts are intended as examples to be read and understood by users. Users can then modify the scripts to generate their own networks.

Below is a list of included tutorials. Frames per Second (FPS) assumes the cores are running at 137 MHz (V1000), 143 MHz (V500) and 154 (V250).

Tutorial Name	Source Framework	Task	Accuracy Metric	Accuracy Score FP32 / 8-bit	V1000 fps	V500 fps	V250 fps
ultralytics.yolov5n.relu	pytorch	Object Detection	mAP(COCO)	32.43/32.1	46.42	32.42	16.48
ultralytics.yolov5s.relu	pytorch	Object Detection	mAP(COCO)	48.9/48.61	18.36	10.22	4.86
ultralytics.yolov5m.relu	pytorch	Object Detection	mAP(COCO)	54.6/54.38	6.68	3.56	1.85
lpd_eu_v42	pytorch	License Plate Detection					
lpr_eu_v3	pytorch	License Plate Recognition					
yolo-v3-tf	tensorflow	Object Detection			2.35	1.22	
yolov2-voc	darknet	Object Detection	mAP(VOC)	74.79/74.13	6.13	3.38	1.61
scrfd_500m_bnkps	onnx	Face Detection			79.37	63.25	33.25
genderage	onnx	Gender Age Estimation			862.19	850.06	584.32

MICROCHIP

Reference Design



- **Inputs (1080p60)**
  - MIPI CSI-2 Camera
  - HDMI
- **Output (1080p60)**
  - HDMI
- **Supported in**
  - PolarFire Video Kit
  - PolarFire SoC Video Kit

PolarFire Video Kit

PolarFire SoC Video Kit

MICROCHIP

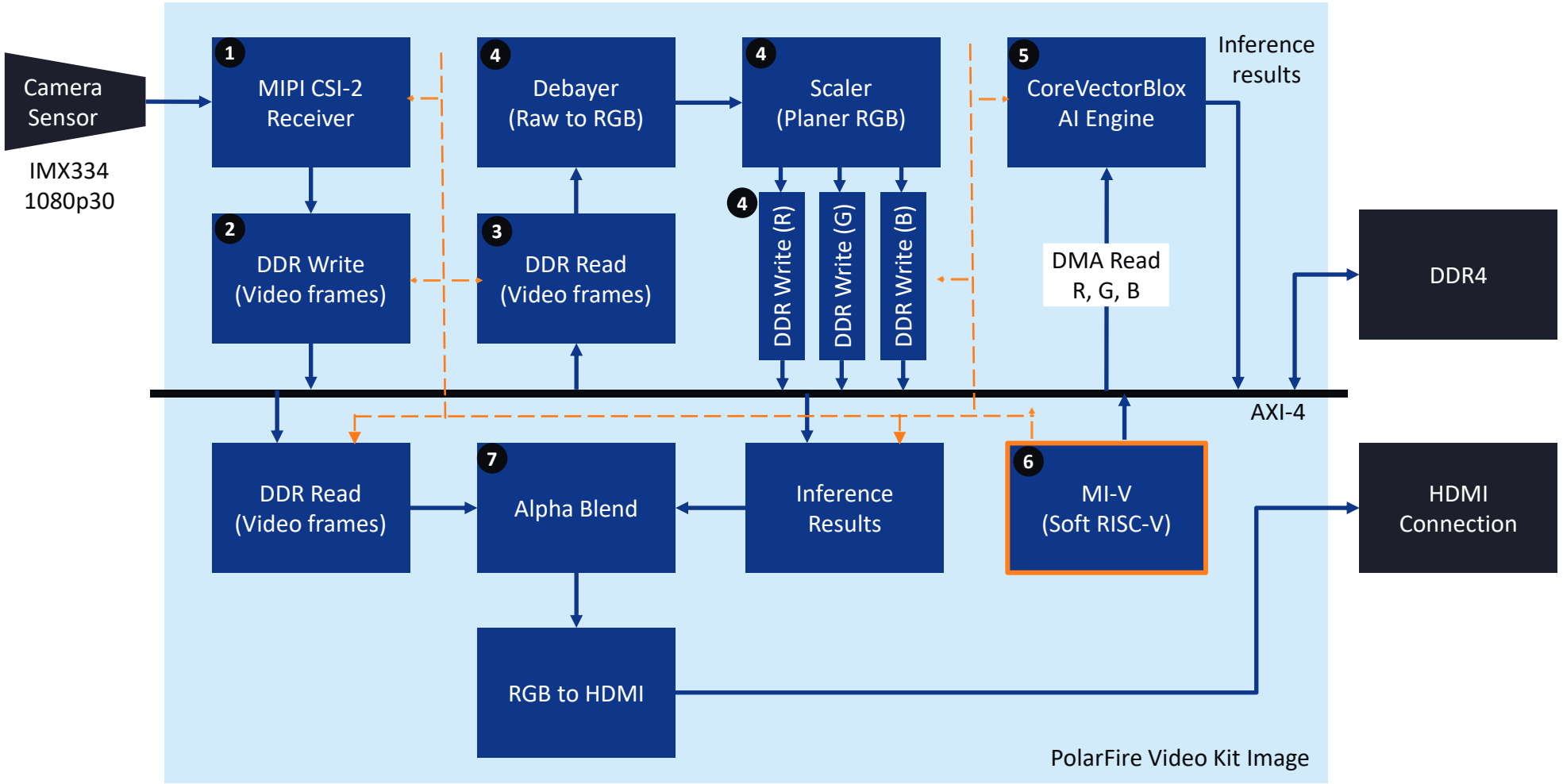
FPS Performance Summary

Demo Mode	FPS*
FaceDemo w/ Gender Age	31
Face Demo	32
License Plate Reader	16
MobileNet	50
YoloV5	38
YoloV4	

Measured using V1000 on the PolarFire® SoC Video Kit

MICROCHIP

Switching CNNs in Real-Time

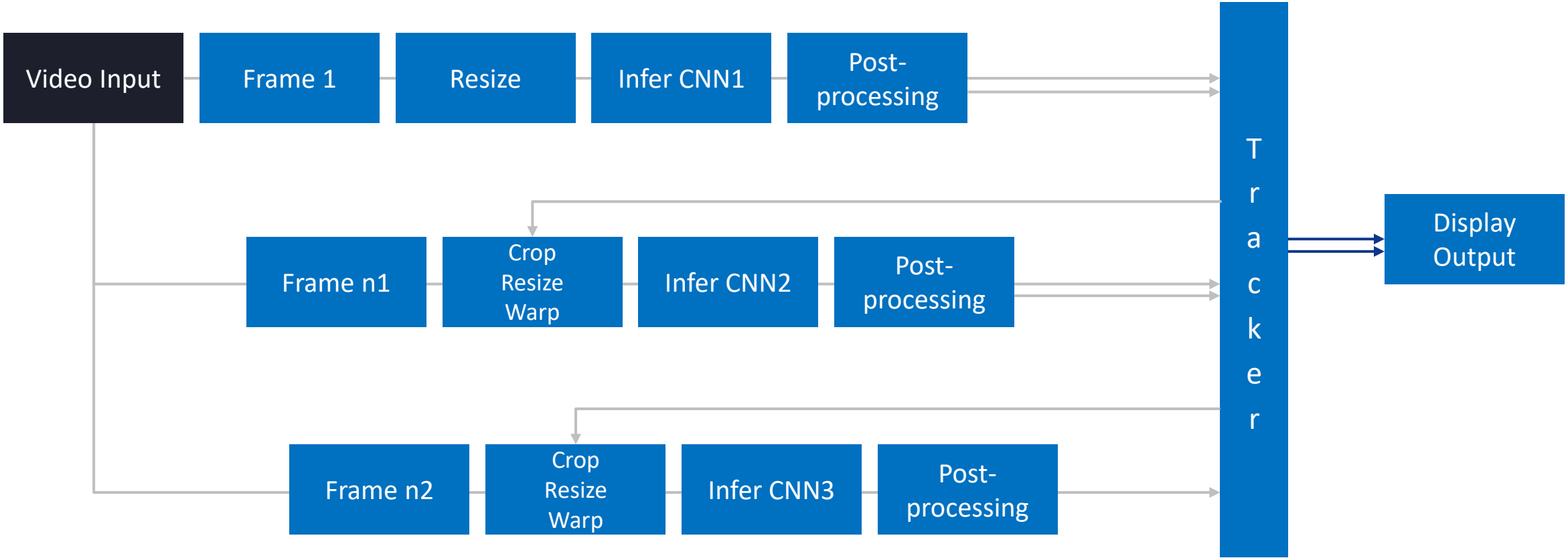


1. Video frame is received via MIPI CSI-2
2. And stored in the DDR via AXI-4 interconnect
3. Before inference – the frame is read back from the DDR
4. Converted from RAW to RGB, RGB to planer R, G, B and written back into DDR
5. CoreVectorBlox engine runs inference on R, G, B arrays and writes results back into DDR
6. Mi-V (PolarFire FPGA) sorts probabilities, creates an overlay frame with bounding boxes, classification results, fps etc., and stores the frame in DDR
    1. PolarFire® SoC FPGA uses the Linux running on the RISC V cores for this task
7. The original video frame is read and blended with the overlay frame and sent out to an HDMI display

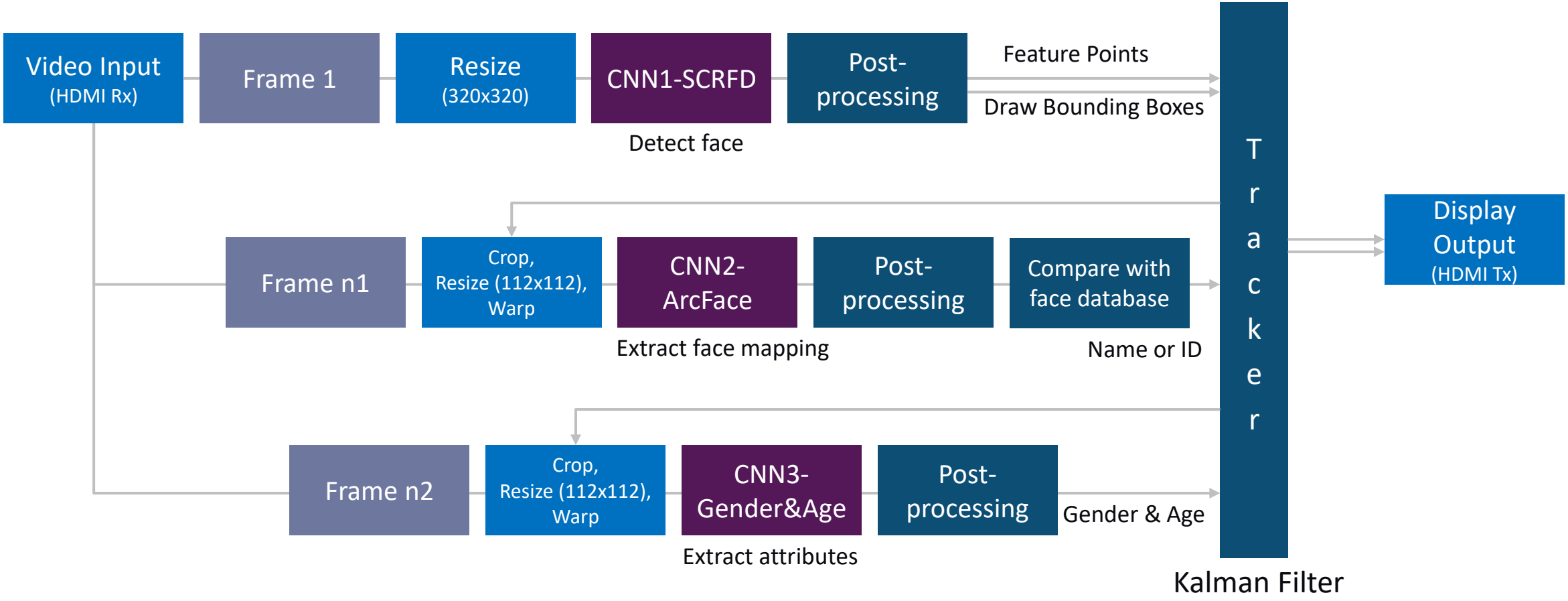
Switching CNNs in Real-Time

Facial Recognition

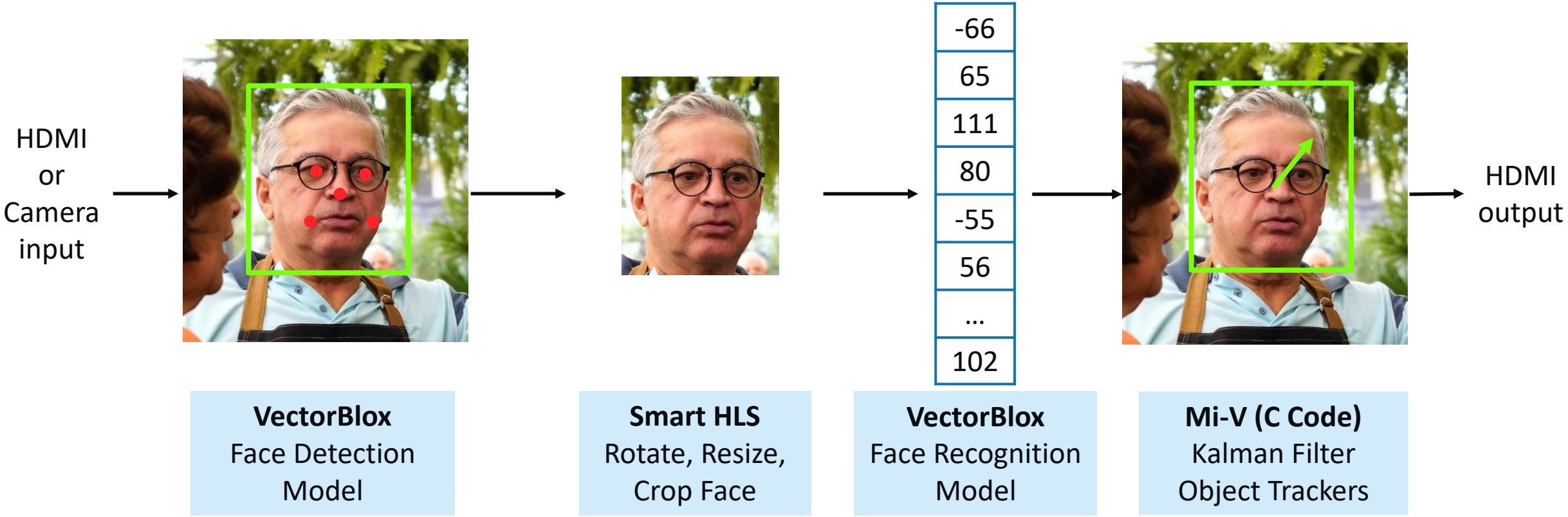
Solving Complex Problems with CNN Pipelining



Solving Complex Problems with CNN Pipelining

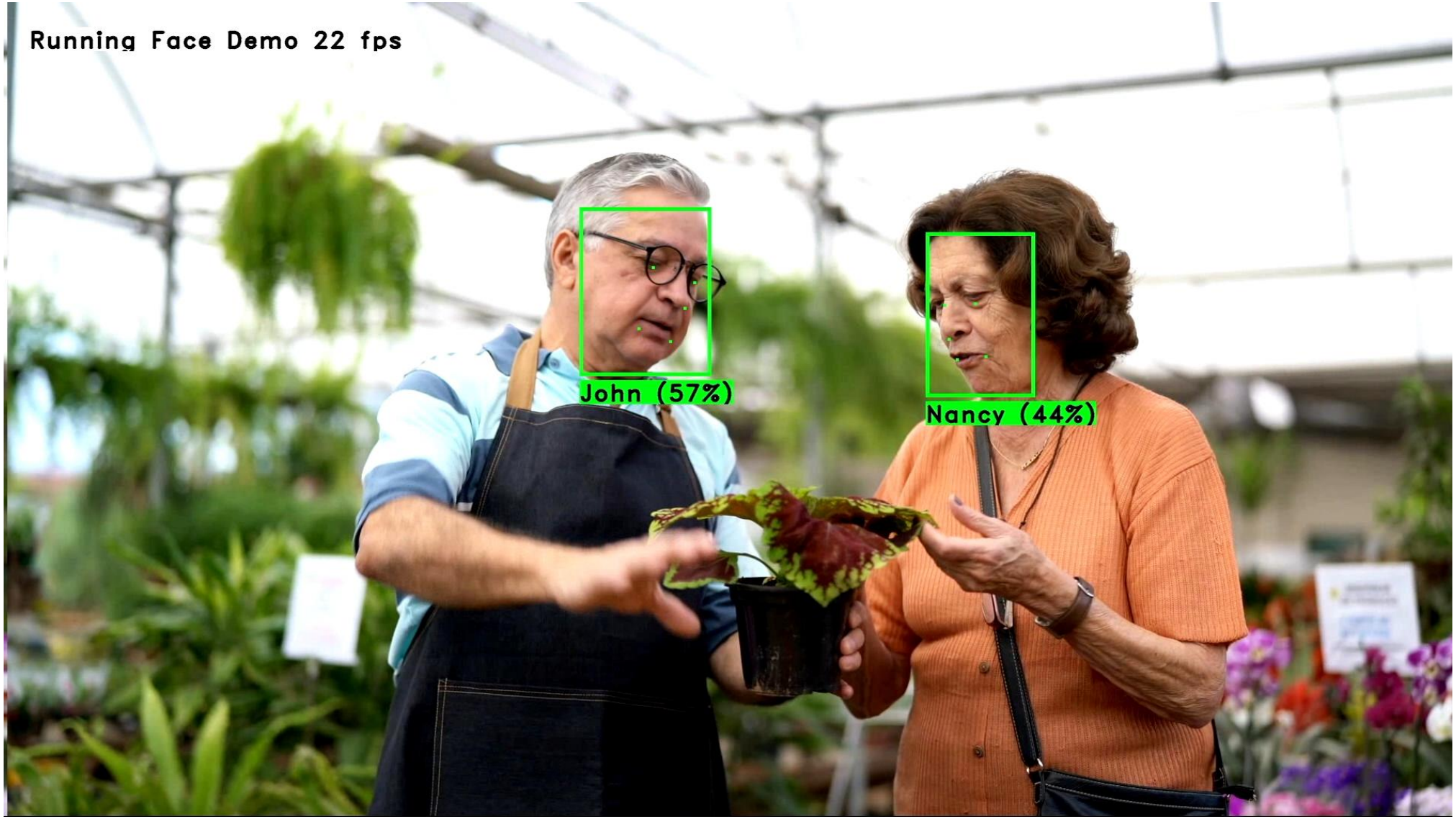


Face Recognition Demo



- Face detection and recognition models share VectorBlox accelerator
- CNNs in the pipeline (e.g., license plate detection and recognition)
- Object tracking common for video applications

Face Recognition - SCRFD, ArcFace

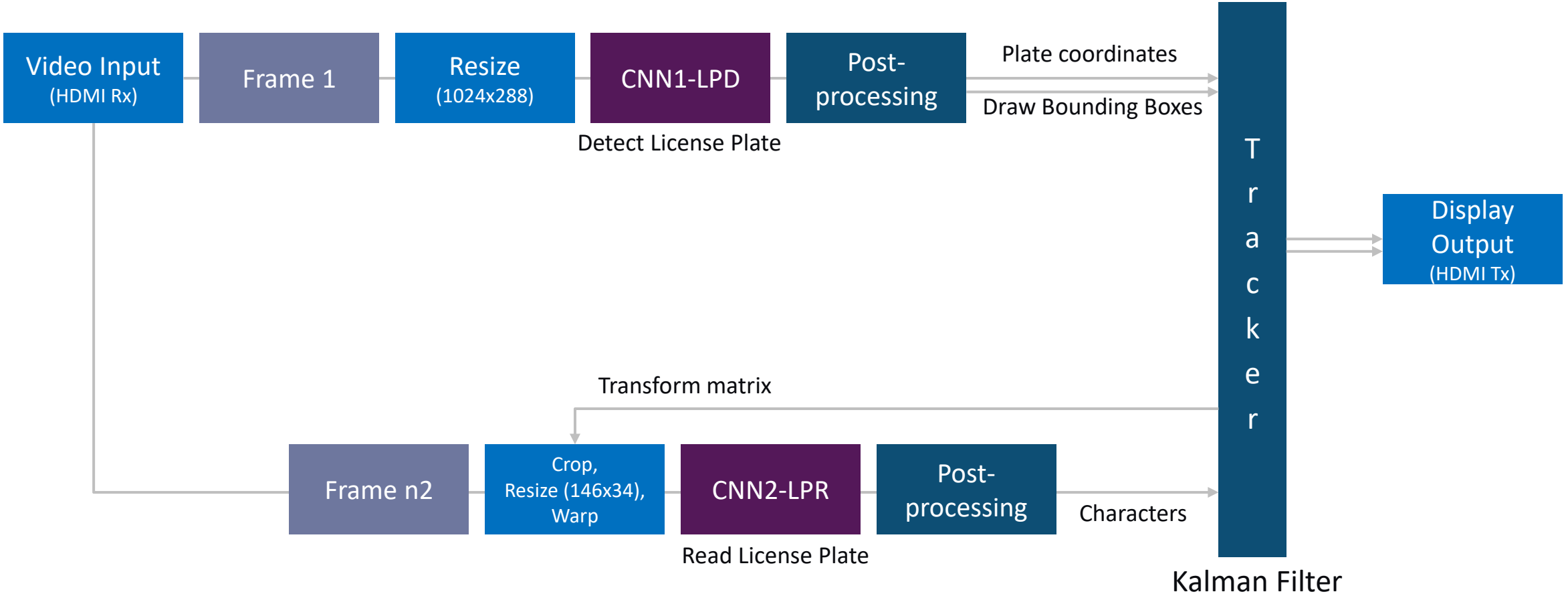


Reference Design - Resource Utilization

CoreVectorBlox Configuration	V1000 (MPF300)	V500 (MPF300)	V250 (MPF300)
FPGA Resources Total LUT/FF/DSP/uSRAM/LSRAM	158K/151K/348/1056/503	143K/130K/220/832/439	126K/111K/140/721/404
FPGA Resources Core LUT/FF/DSP/uSRAM/LSRAM	59K/69K/292/531/146	44K/48K/164/307/82	26K/28K/84/165/47
Face Detection Network Runtime	13 ms	16 ms	30 ms
Face Recognition Network Runtime	8 ms	10 ms	19 ms
Face Attribute Network Runtime	1 ms	1 ms	2 ms
Total FPS (including post-processing)	23 FPS (44 ms)	20 FPS (49 ms)	14 FPS (73 ms)

License Plate Reading

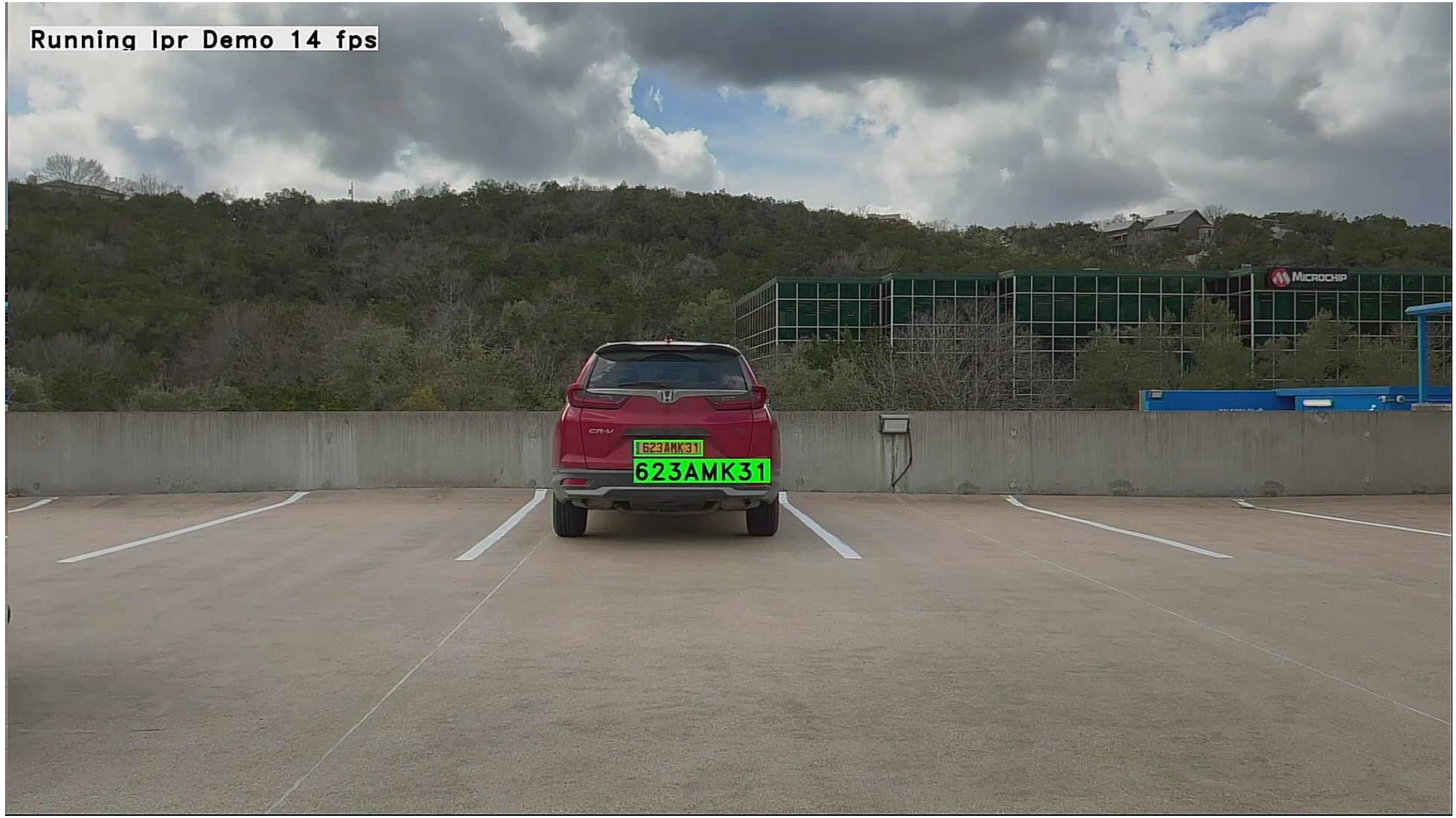
Solving Complex Problems with CNN Pipelining



FPGA fabric

Soft Processor

License Plate Recognition



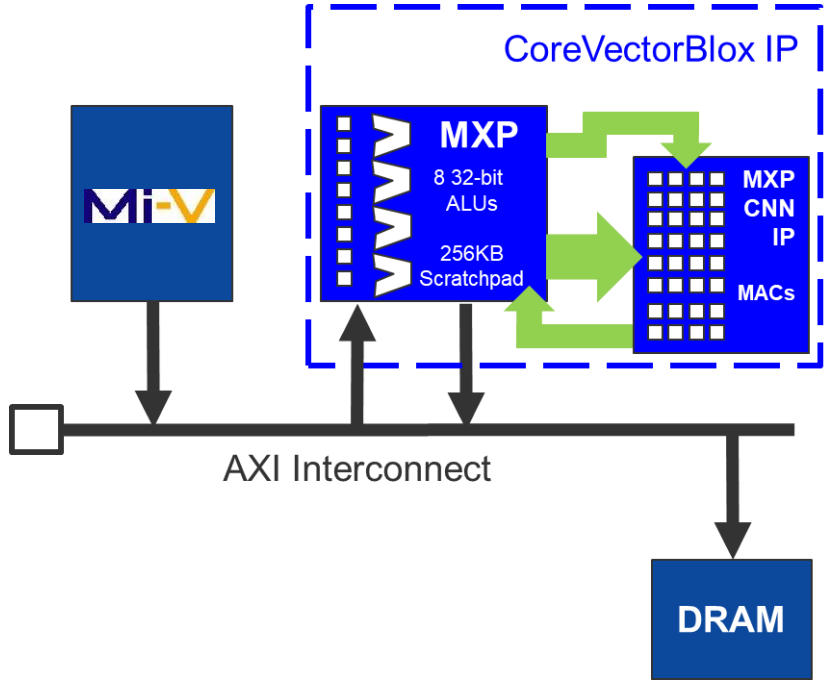
MICROCHIP

Reference Design - Resource Utilization

CoreVectorBlox Configuration	V1000 (MPF300)	V500 (MPF300)	V250 (MPF300)
FPGA Resources Total LUT/FF/DSP/uSRAM/LSRAM	160K/153K/353/1061/507	145K/132K/225/837/443	128K/112K/145/726/406
FPGA Resources Core LUT/FF/DSP/uSRAM/LSRAM	60K/69K/292/531/148	45K/48K/164/307/84	26K/28K/84/16/47
Detection Network Runtime	51.83 ms	83.42 ms	188.34 ms
Reading Network Runtime	4.16 ms	6.74 ms	12.11 ms
Total FPS (including post-processing)	14 FPS (56 ms)		

CoreVectorBlox Neural Network Engine

- **Multi-core Software Architecture**
  - Mi-V runs complete software-based application
- **VectorBlox Matrix Processor (MXP)**
  - Elementwise tensor operations
    (add, sub, xor, shift, mul, dotprod …)
  - Up to 8 32-bit ALUs
  - Mixing precisions ok (int8, int16, int32)
  - 256KB Scratchpad Memory and DMA Controller
- **VectorBlox CNN**
  - Tensor multiply-accumulate operations
  - Fixed at either int8 precision
  - Layer enhancements added with software update



Overlay implementation allows several different networks to run on the same FPGA design without the need to resynthesize

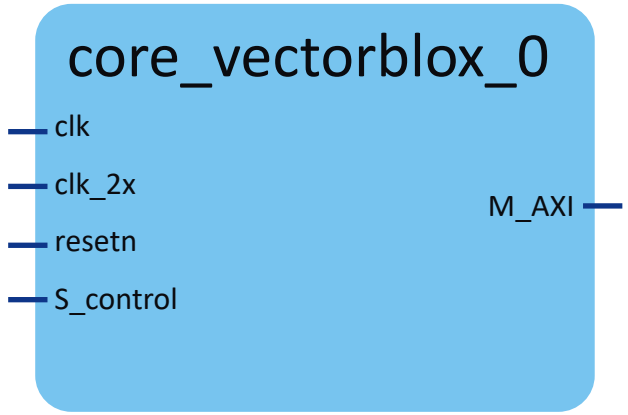
MICROCHIP

CoreVectorBlox v1.4.4 Utilization & Performance

Resources	V250	V500	V1000
Peak GOPs	79	146	279
ResNet-50 (fps)	4	8	14
Tiny Yolo v3 (fps)	5	10	28
MobileNet v1 (fps)	19	38	81
MobileNet v2 (fps)	17	34	76

Resource Utilization (LUT4)			
MPF100 LEs	25%	43%	
MPF200 LEs	14%	24%	32%
MPF300 LEs	9%	16%	21%
MPF500 LEs	6%	10%	13%

Fabric Clock	153 MHz	143 MHz	129 MHz



Configuration Options

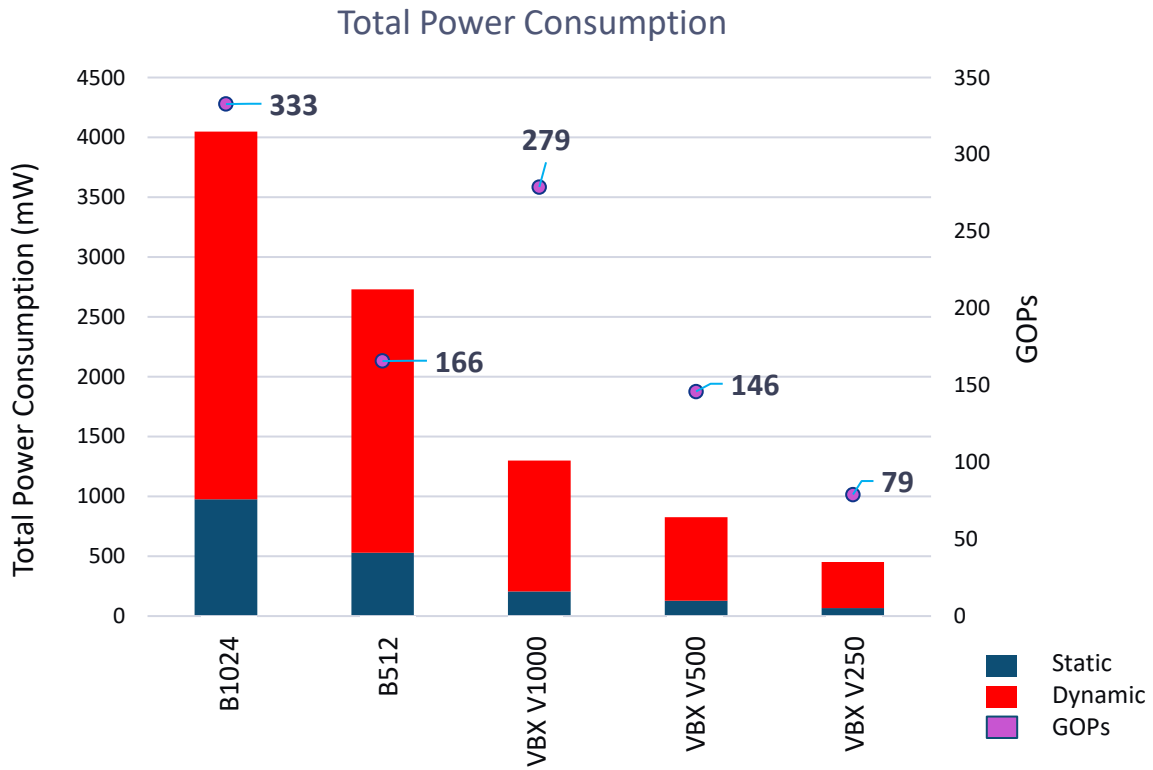
Size Configuration: V250, V500, V1000

Memory Master Width: 256b, 512b

- **Configurable performance based on size and power budget**

2-3x More Power Efficient Inference

Core Name	Peak GOPs	Dynamic Power (mW)	Static Power* (mW)	Total Power (mW)	Total Power (mW/GOP)
VectorBlox V1000	279	1094	206	1300	**5.1**
VectorBlox V500	146	698	127	825	6.4
VectorBlox V250	79	387	65	452	7.1
Comp A B1024	332.8	3072	976	4048	12.2
Comp A B512	166.4	2201	528	2729	16.4



- **2-3x more power efficient inferencing up to 280 GOPs**
- **Suitable for applications requiring**
  - Low power consumption, small enclosures and fan less designs

*Scaled for resource utilization

MICROCHIP

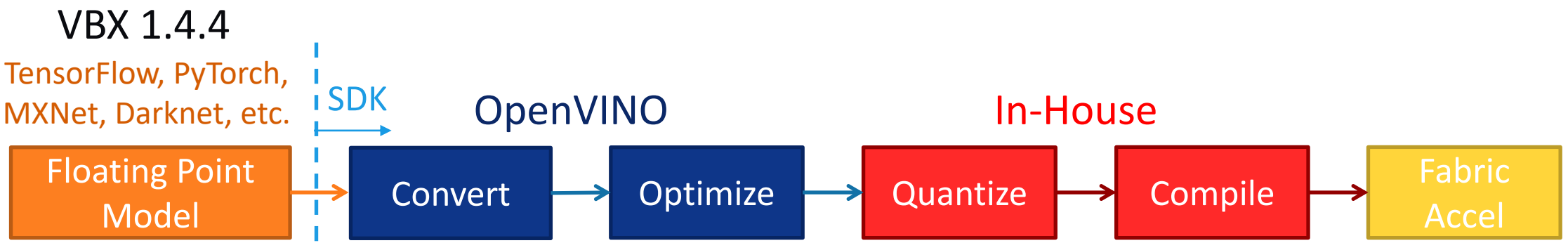
Supported Layers in CoreVectorBlox IP

Supported CNN Layers		
Abs	Add	AveragePool
BatchNormalization	Concat	Conv
Dropout	Gemm	Identity
LeakyRelu	MatMul	MaxPool
Mul	PRelu	Relu

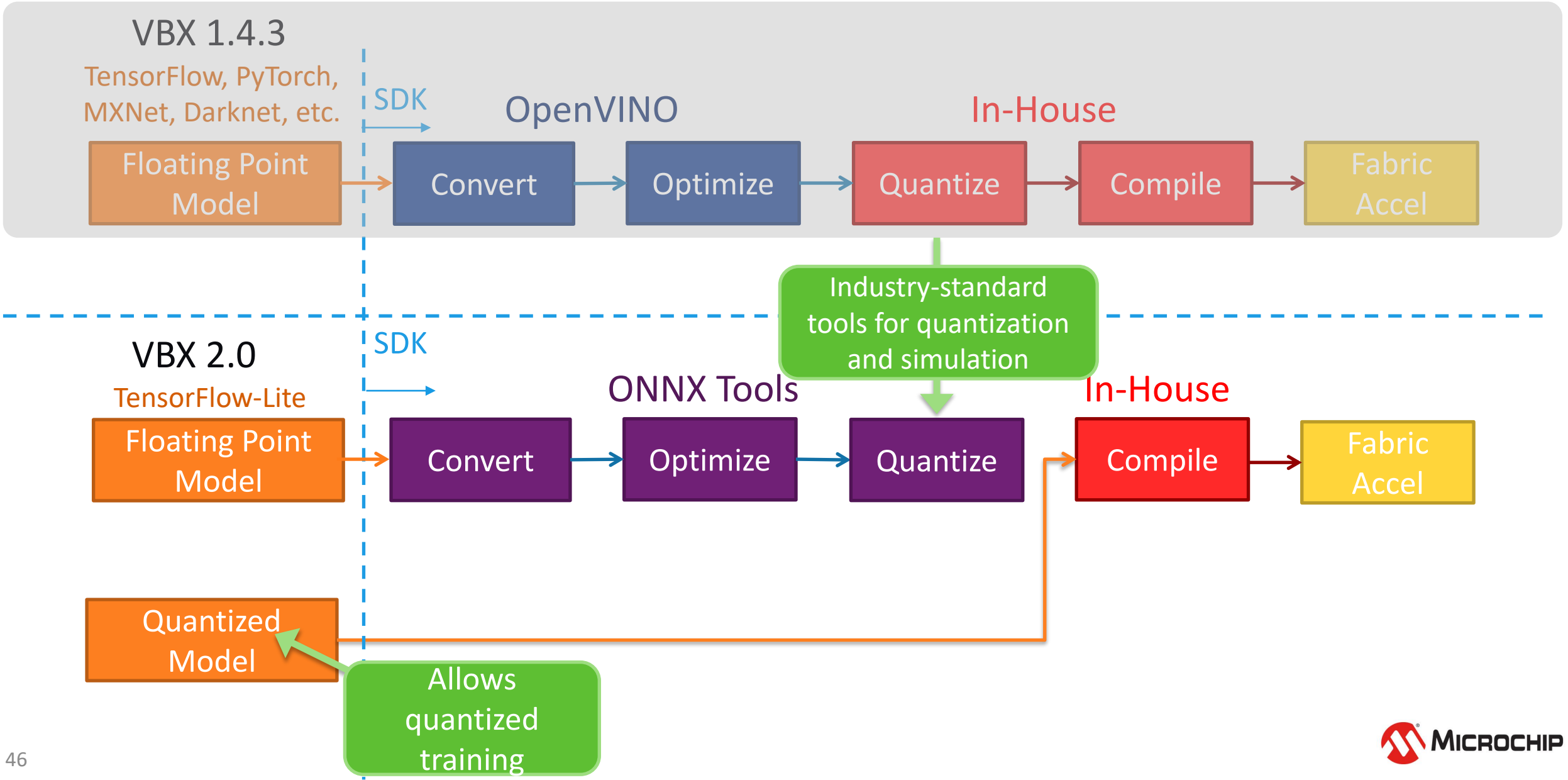
- Additional layers can be added via software update

VectorBlox Roadmap

VectorBlox Roadmap



VectorBlox Roadmap

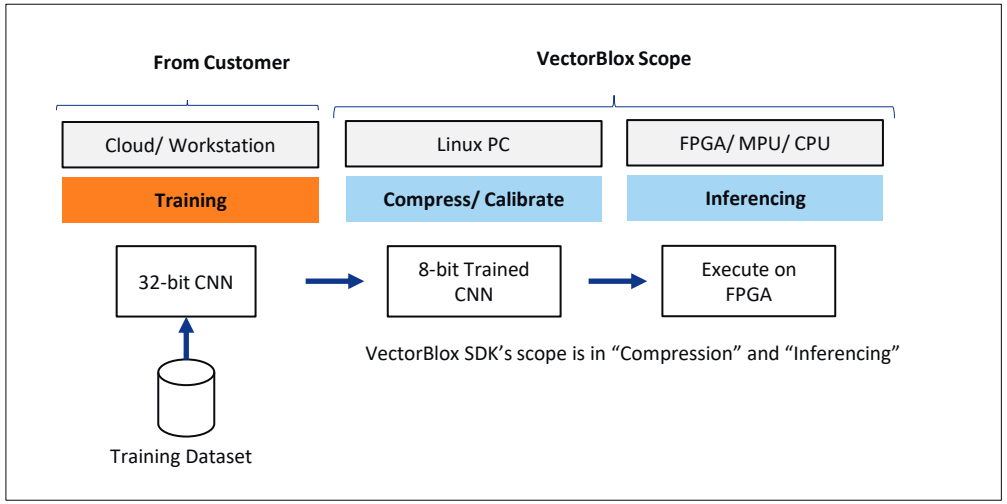


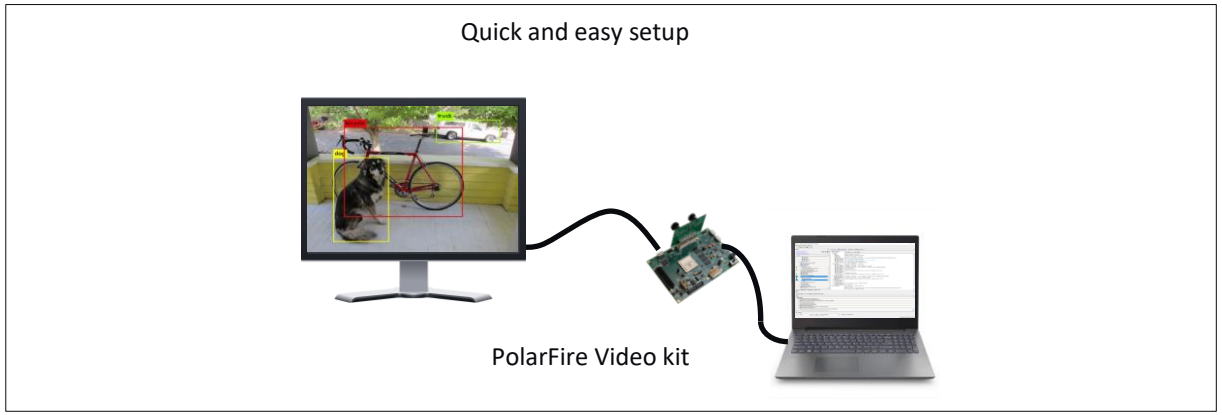
MICROCHIP

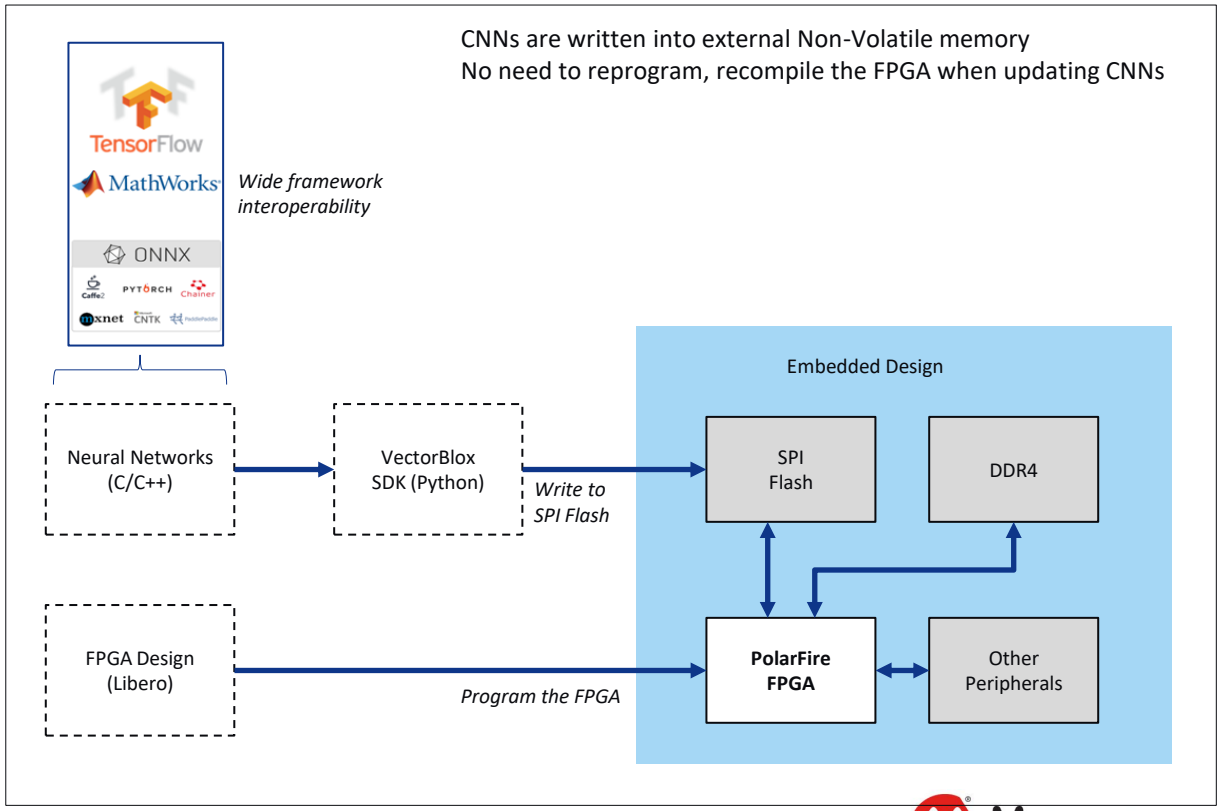
VectorBlox™ in Summary

VectorBlox SDK and NN IP enables

- Software developers to run power efficient Neural Networks (NN) without prior FPGA knowledge
- Utilization of most popular NN software frameworks
- Simulation in software without procuring hardware







MICROCHIP

Thank You