

MATLAB EXPO

Deploying Deep Learning on Embedded Devices – When FPGAs Make Sense

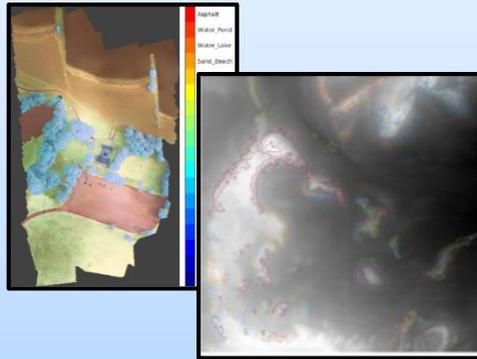
Jack Erickson

HDL Technical Marketing

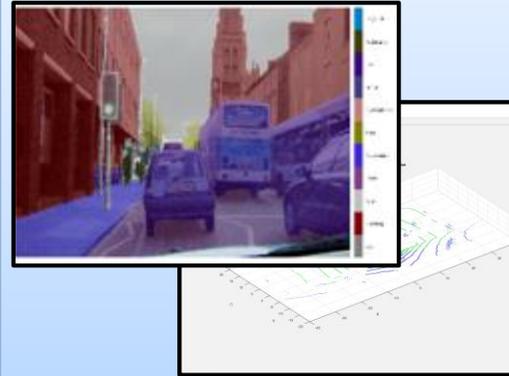


Deep Learning Inferencing on Embedded Devices

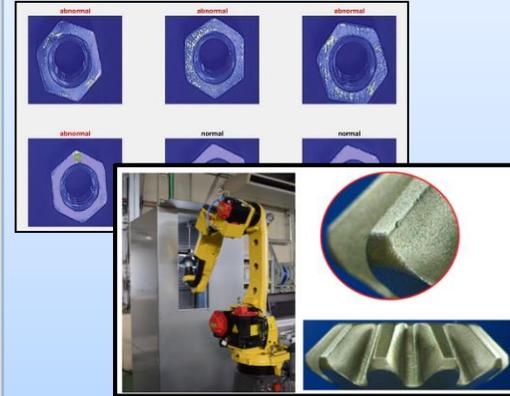
Airborne Image Analysis



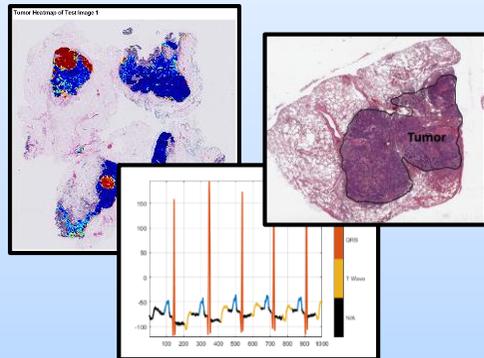
Autonomous Driving



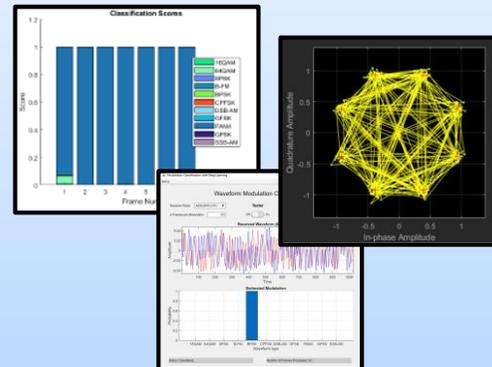
Industrial Inspection



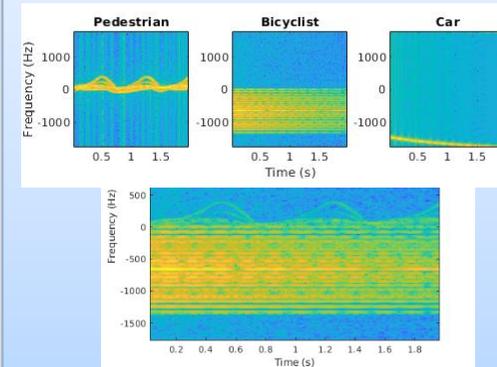
Medical Image Analysis



Wireless Modulation Classification



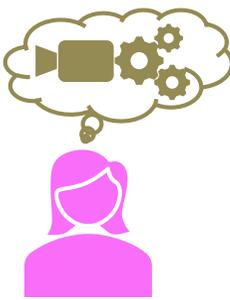
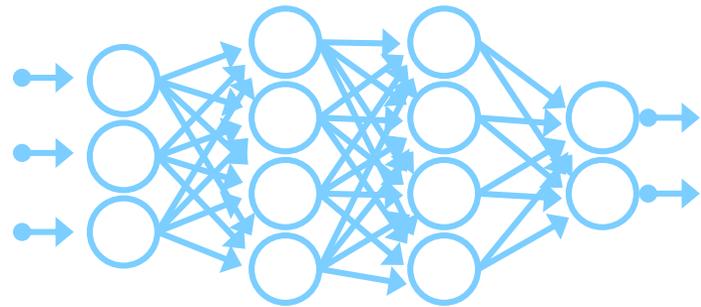
Radar Signature Classification



System Requirements Drive Network Design

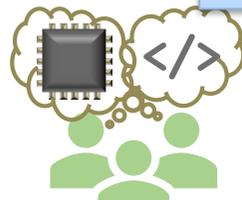


Deep Learning Practitioner



Systems Engineer

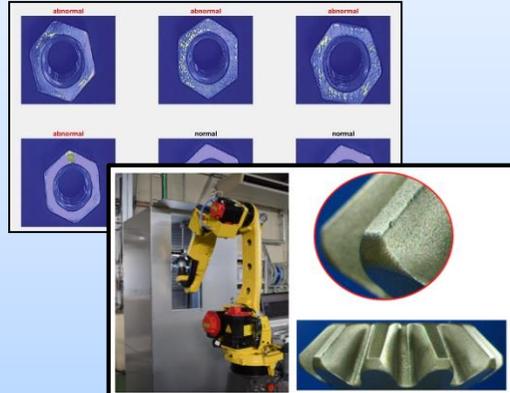
Camera specs
Accuracy
Latency
Cost
Power



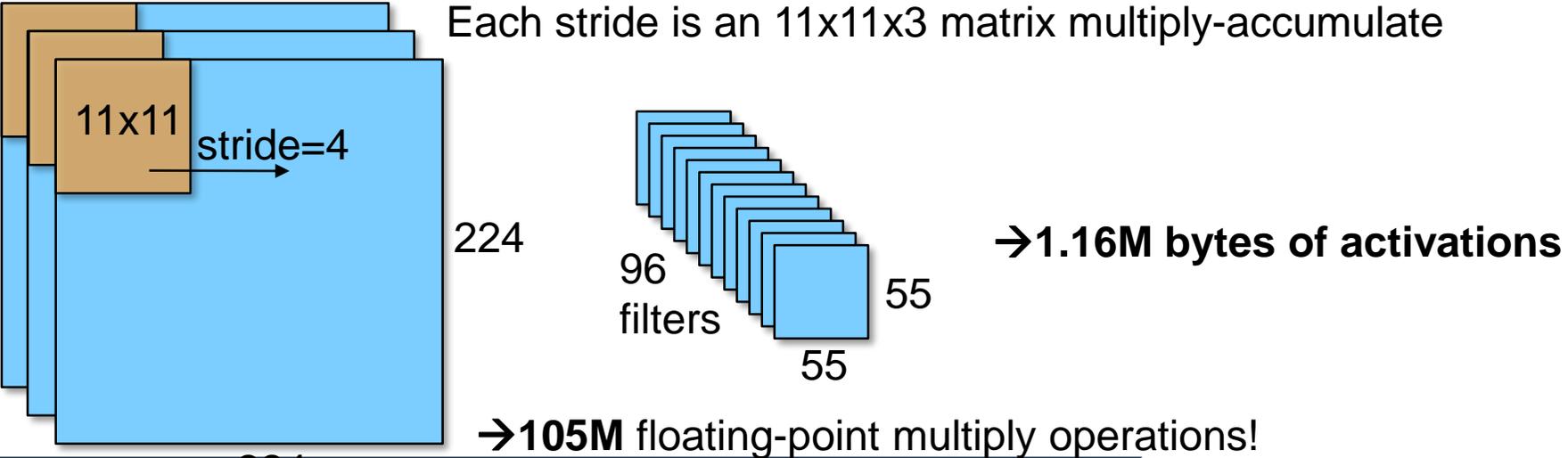
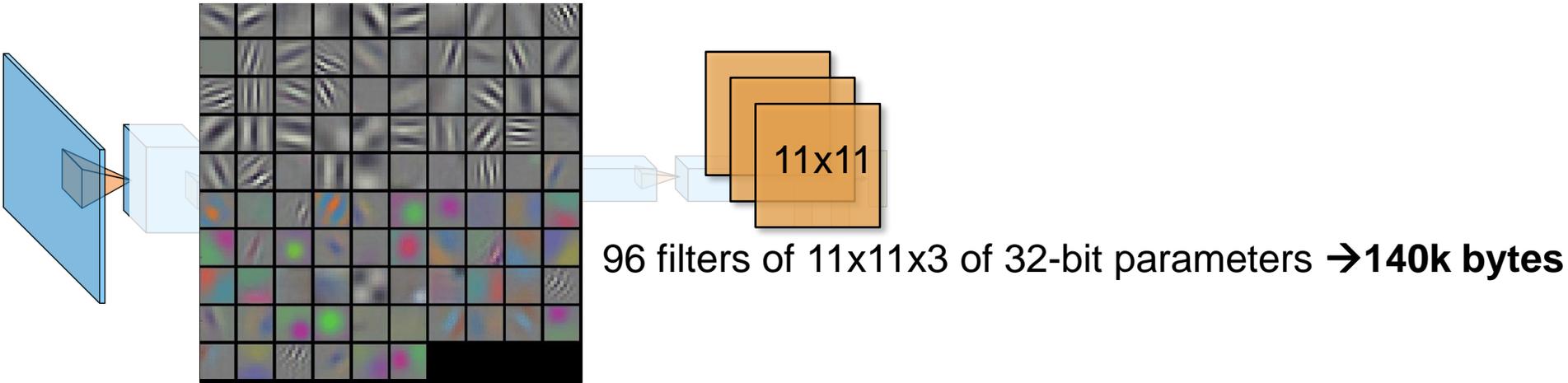
Hardware/Software Engineers



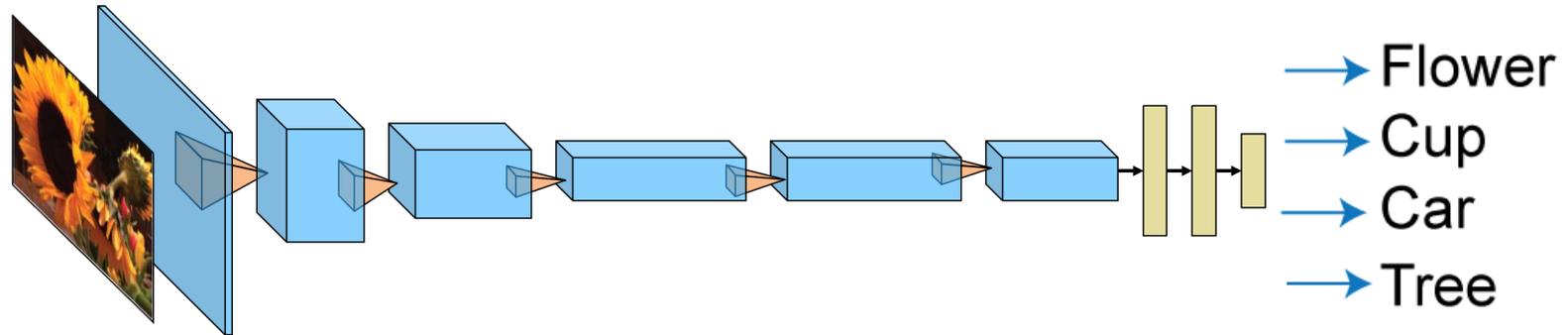
Industrial Inspection



Challenges of Deploying Deep Learning to FPGA Hardware: Convolution

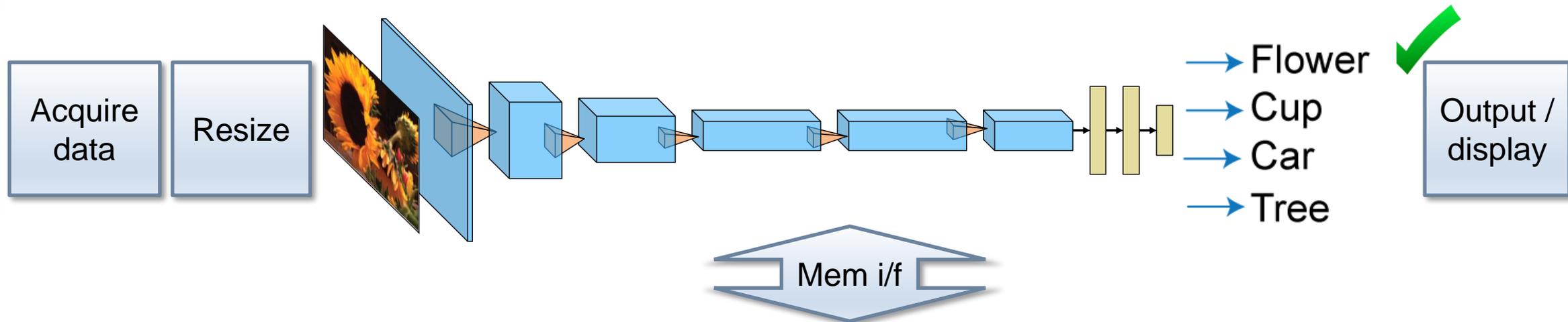


Challenges of Deploying Deep Learning to FPGA Hardware



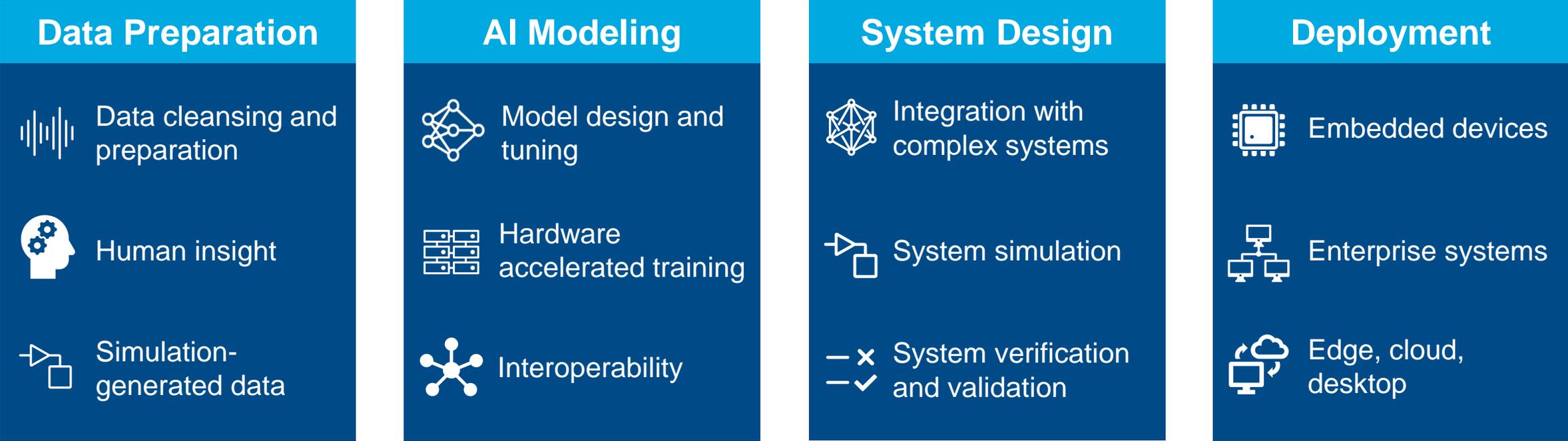
	input	conv 1	conv 2	conv 3	conv 4	conv 5	fc6	fc7	fc8	Total	
Parameters (Bytes)	n/a	140K	1.2M	3.5M	5.2M	1.8M	148M	64M	16M	230 M	➡ Off-chip RAM
Activations (Bytes)	588K	1.1M	728K	252K	252K	168K	16K	16K	4K	3.1 M	➡ Block RAM
FLOPs	n/a	105M	223M	149M	112M	74M	37M	16M	4M	720 M	➡ DSP Slices

Deploying Deep Learning to FPGA Hardware Requires Collaboration



	input	conv 1	conv 2	conv 3	conv 4	conv 5	fc6	fc7	fc8	Total
Parameters (Bytes)	<div style="text-align: center;"> <p>Optimize</p> <ul style="list-style-type: none"> • Network / layers • Fixed-point quantization • Processor micro-architecture  </div>									
Activations (Bytes)										
FLOPs										

AI-Driven System Design



Iteration and Refinement

Design and Analyze Your Networks in MATLAB

AI Modeling



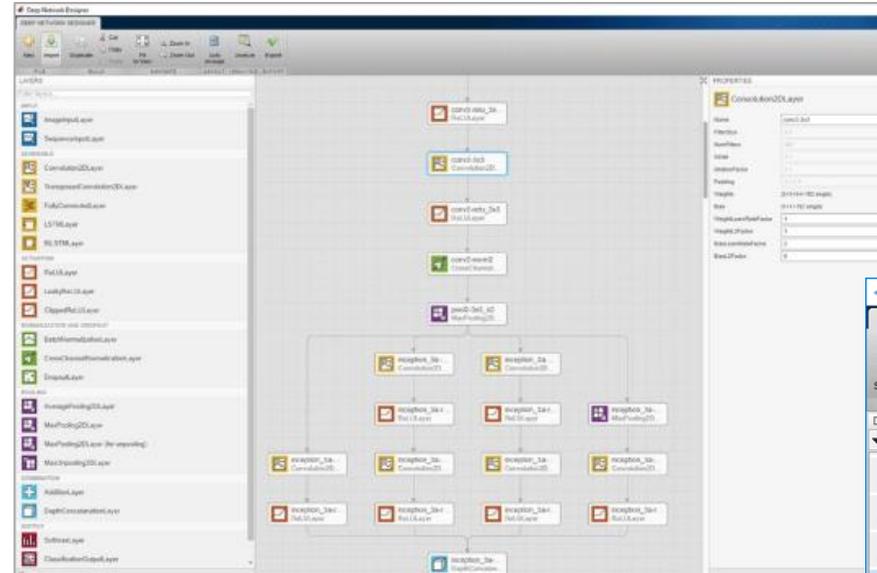
Model design and tuning



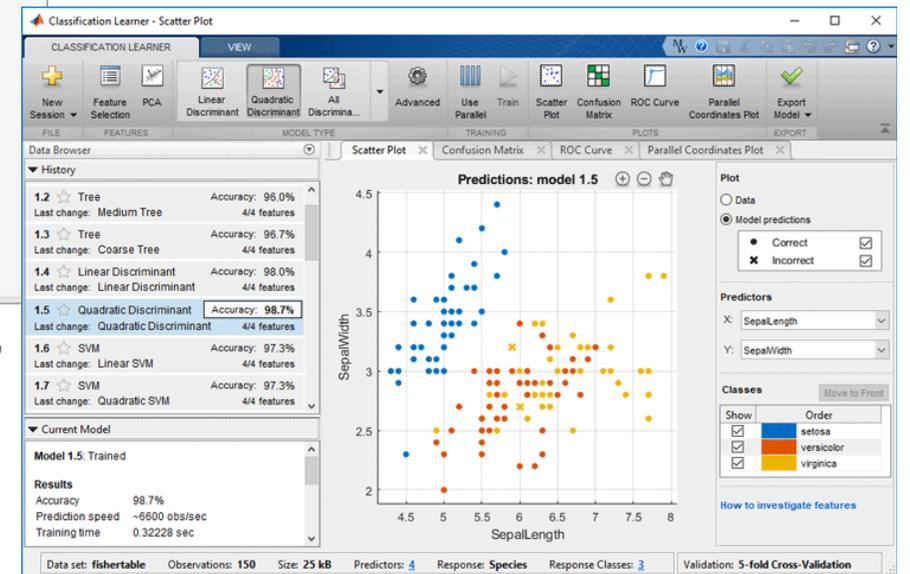
Hardware accelerated training



Interoperability



Deep Network Designer app to build, visualize, and edit deep learning networks

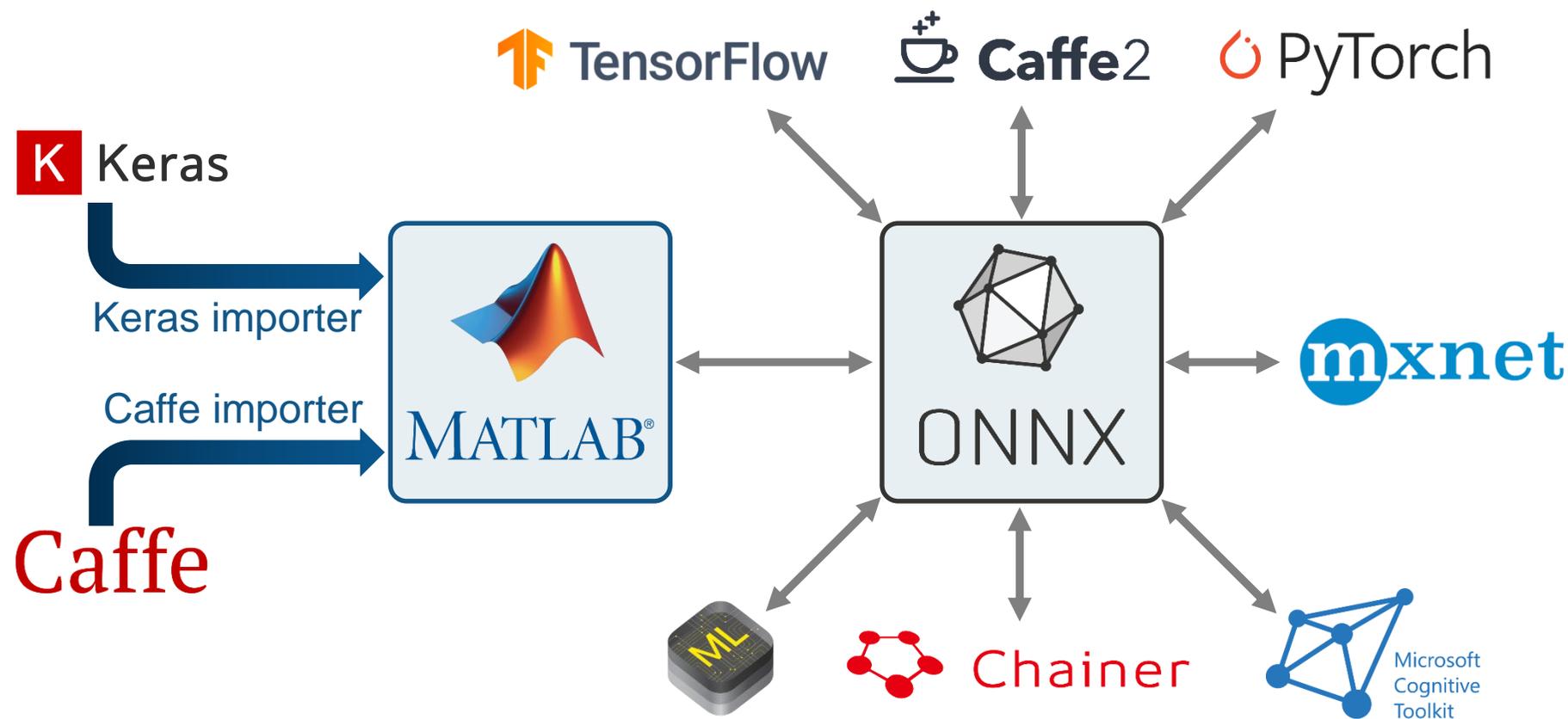


Classification Learner app to try different classifiers and find the best fit for your data set

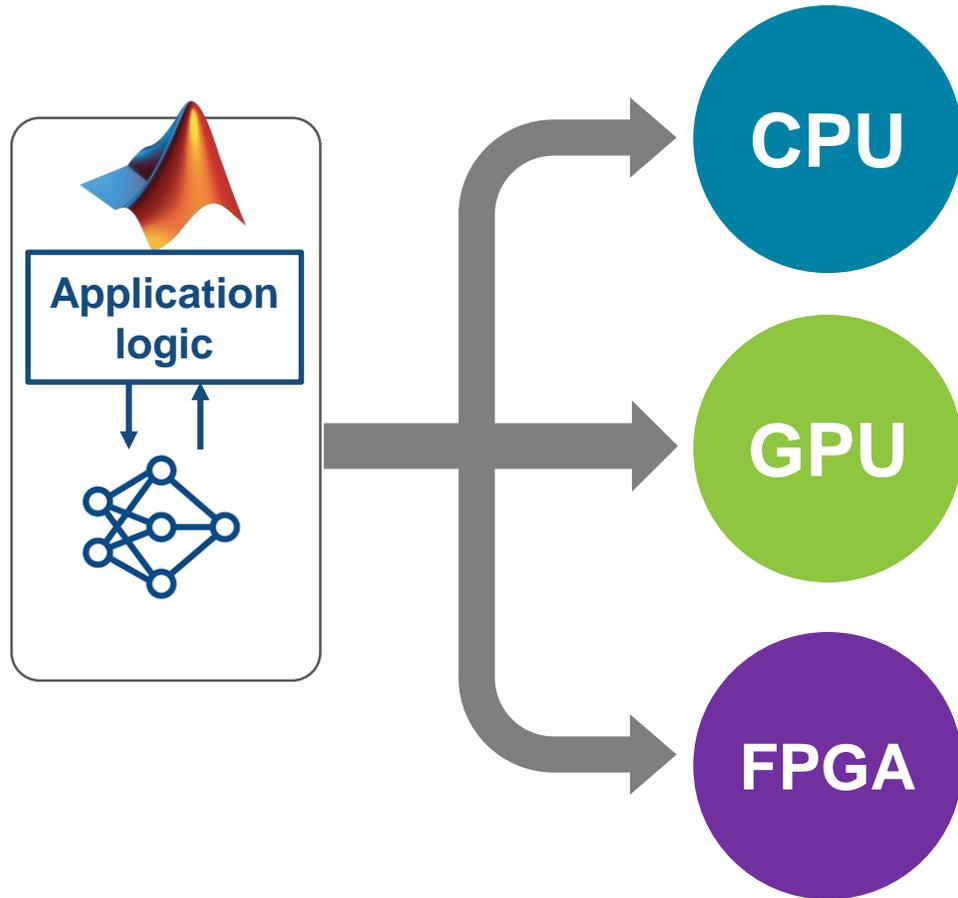
MATLAB Interoperates with Other AI Frameworks

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

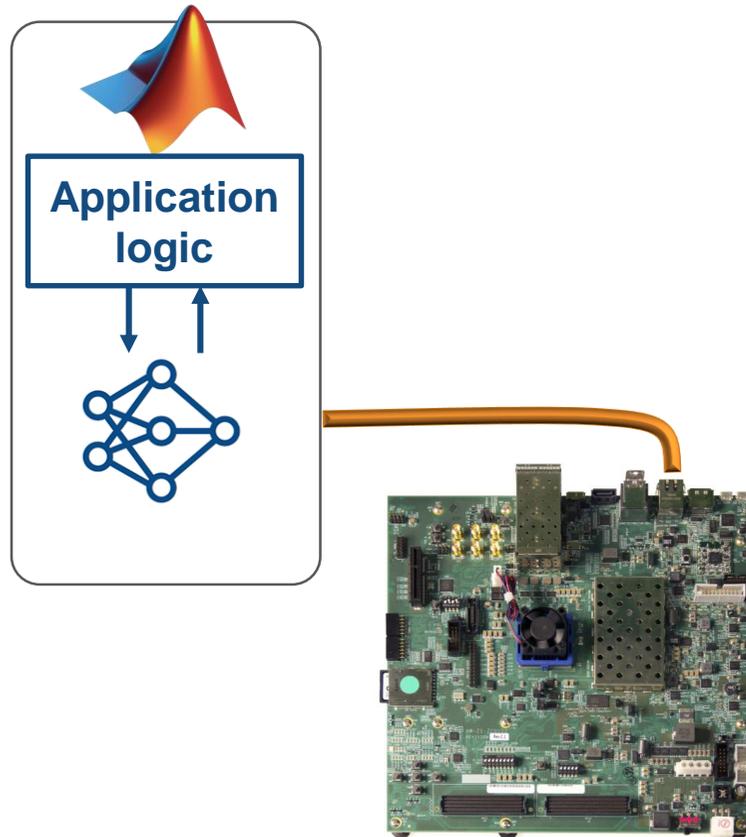


Deploy from MATLAB to a Variety of Hardware Platforms



Deployment	
	Embedded devices
	Enterprise systems
	Edge, cloud, desktop

FPGA Deployment from MATLAB

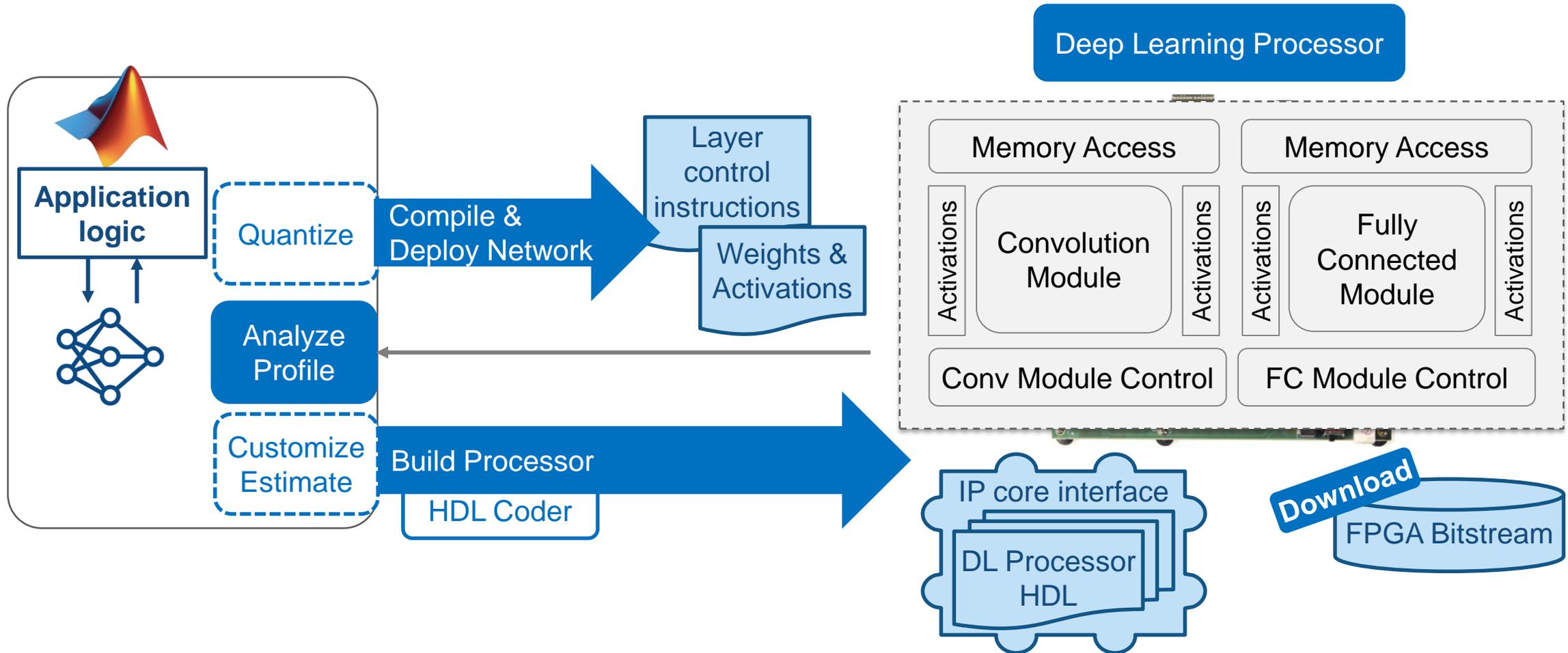


Deep Learning HDL Toolbox™

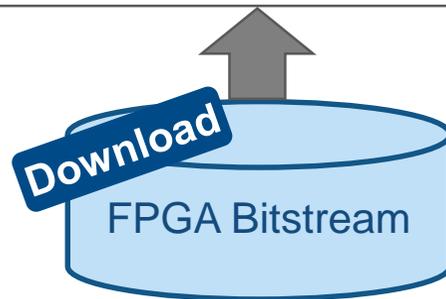
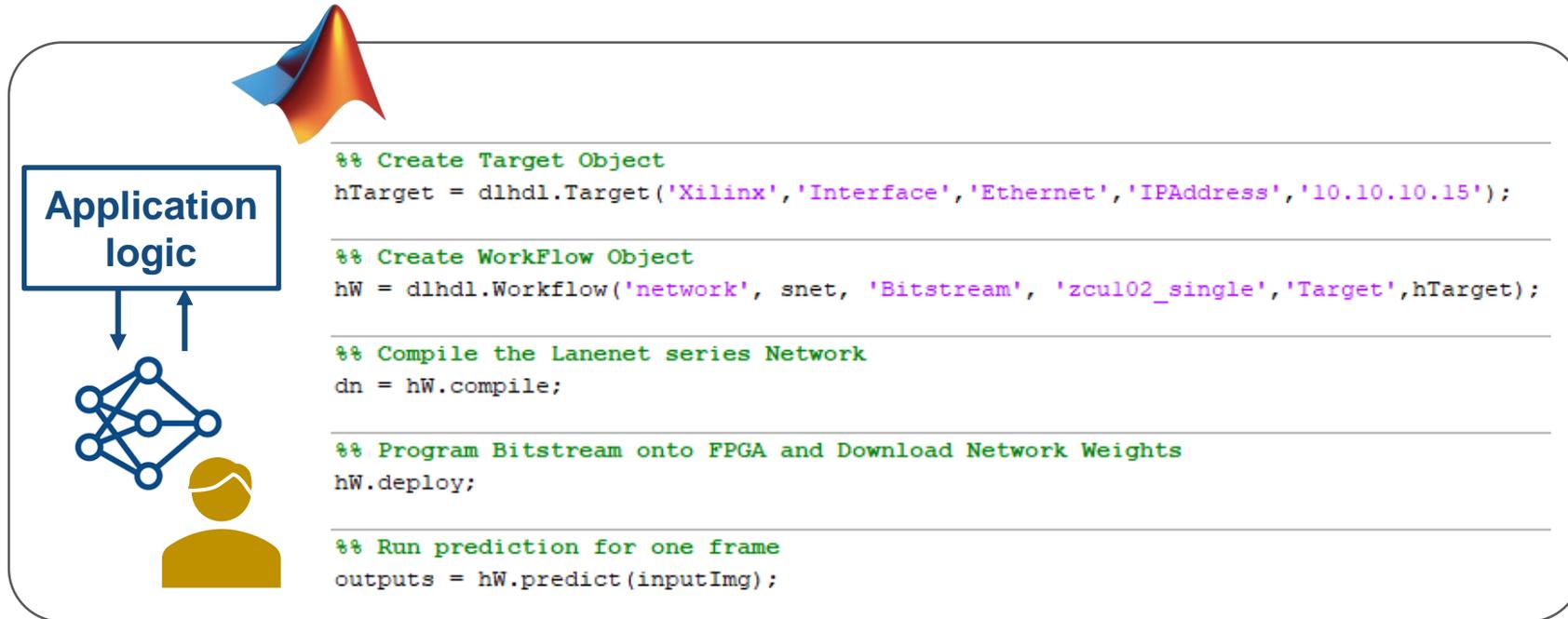
- Prototype network on FPGA
 - Assess memory usage, latency, and accuracy
 - Adjust network and iterate
 - Quantize to fixed-point
 - Generate customized deep learning processor HDL
- ...all from within MATLAB!



Deep Learning HDL Toolbox Components



Get Started Prototyping on FPGA with Deep Learning HDL Toolbox™



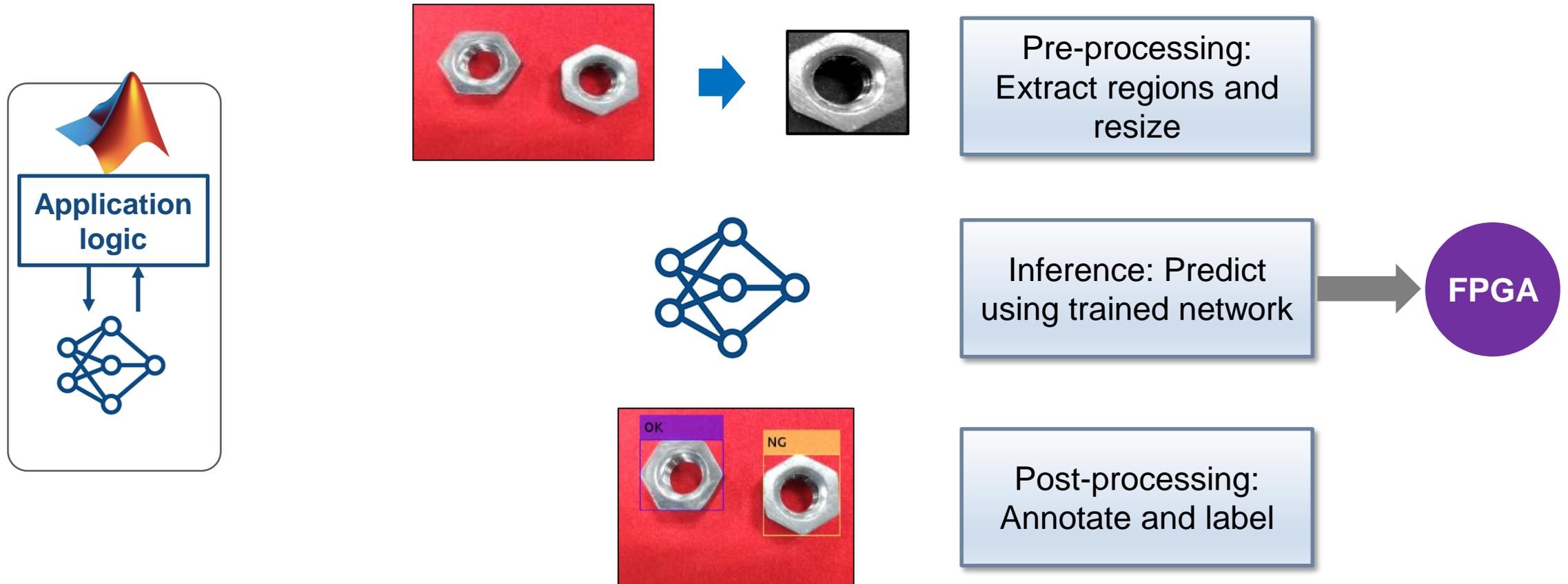
Hardware support package

Deep learning processor with I/O and external memory interfaces

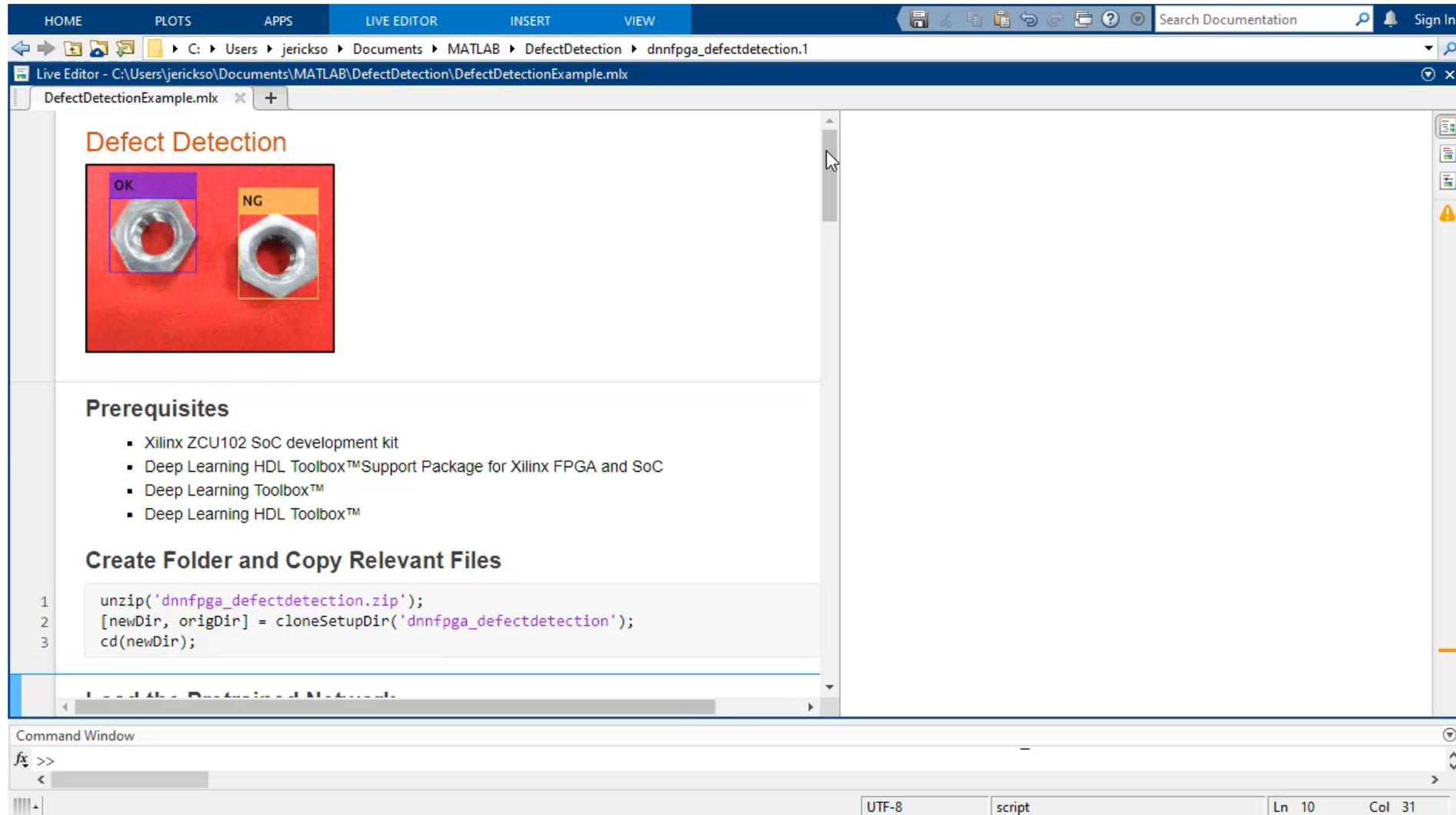
- Int8 or single
- Supported boards:
 - Xilinx: ZCU102 or ZC706
 - Intel: Arria10 SoC
- <http://mathworks.com/hardware-support.html>



Defect Detection Example



Run Deep Learning on FPGA from MATLAB

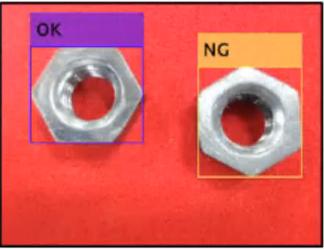


HOME PLOTS APPS LIVE EDITOR INSERT VIEW Search Documentation Sign In

C:\Users\jerickso\Documents\MATLAB\DefectDetection\dnnfpga_defectdetection.1

Live Editor - C:\Users\jerickso\Documents\MATLAB\DefectDetection\DefectDetectionExample.mlx

Defect Detection



Prerequisites

- Xilinx ZCU102 SoC development kit
- Deep Learning HDL Toolbox™ Support Package for Xilinx FPGA and SoC
- Deep Learning Toolbox™
- Deep Learning HDL Toolbox™

Create Folder and Copy Relevant Files

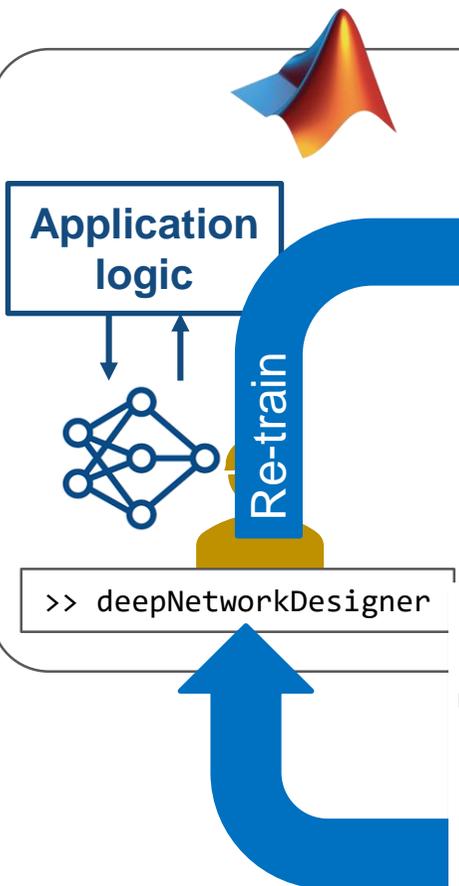
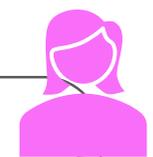
```
1 unzip('dnnfpga_defectdetection.zip');  
2 [newDir, origDir] = cloneSetupDir('dnnfpga_defectdetection');  
3 cd(newDir);
```

Command Window

```
fx >>
```

UTF-8 script Ln 10 Col 31

Profile FPGA Prototype and Iterate in MATLAB



```

% Load the modified and trained network
net2 = load('trainedBlemDetNet.mat');
snet_blemdetnet = net2.convnet;
% Use the new network in the workflow object
hW = dlhdl.Workflow('Network',snet_blemdetnet,'Bitstream','zcu102_single','Target');
hW.compile
hW.deploy

scores = zeros(2,4);
for i = 1:num
    [scores(:,i), speed] = hW.predict(single(imgPacked2(:,:,i)), 'Profile', 'on');
end
    
```

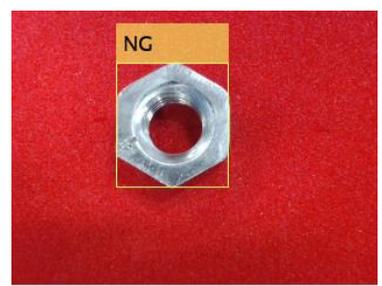
Layer control instructions

Weights & Activations

Deep Learning Processor Profiler Performance Results

	LastLayerLatency(cycles)	LastLayerLatency(seconds)	FramesNum	Total Latency	Frames/s
Network	12213262	0.05551	1	12213302	18.0
conv_module	3292045	0.01496			
conv1	412728	0.00188			
norm1	173252	0.00079			
pool1	58636	0.00027			
conv2	656582	0.00298			
norm2	128169	0.00058			
pool2	53269	0.00024			
conv3	780456	0.00355			
conv4	600050	0.00273			
conv5	408977	0.00186			
pool5	20059	0.00009			
fc_module	8921217	0.04055			
fc6	1759800	0.00800			
fc7	7030644	0.03196			
fc8	130771	0.00059			

* The clock frequency of the DL processor is: 220MHz



Design Exploration and Customization

The screenshot shows the MATLAB Live Editor interface with the following code in the editor:

```
11 hT = dlhdl.Target('Xilinx','Interface','Ethernet','IPAddress','10.10.10.15')  
  
12 hW = dlhdl.Workflow('Network',snet_defnet,'Bitstream','zcu102_single','Target',h  
  
13 hW.compile  
  
14 hW.deploy  
  
15 unzip('testImages.zip')  
16  
17 filename=[pwd,'/testImages/ng1.png'];  
18 img=imread(filename);  
19 predictDefect(hW, img);
```

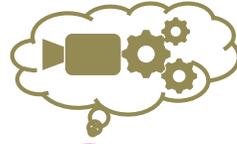
The right pane displays the following performance results table:

Network	LastLayerLatency(cycles)	LastLayerLatency(seconds)
conv_module	12213262	0.05551
conv1	3292045	0.01496
conv2	412728	0.00188
conv3	173252	0.00079
conv4	58636	0.00027
conv5	656582	0.00298
conv6	128169	0.00058
conv7	53269	0.00024
conv8	780456	0.00355
conv9	600050	0.00273
conv10	408977	0.00186
conv11	20059	0.00009
conv12	8921217	0.04055
conv13	1759800	0.00800
conv14	7030644	0.03196
conv15	130771	0.00059

* The clock frequency of the DL processor is: 220MHz

The prediction image shows a red background with a metal nut in the center, labeled 'NG' in a yellow box.

Collaborate to Quantize Network



Systems Engineer

Accuracy



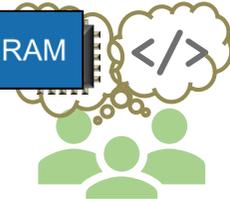
- Latency
- Cost
- Power



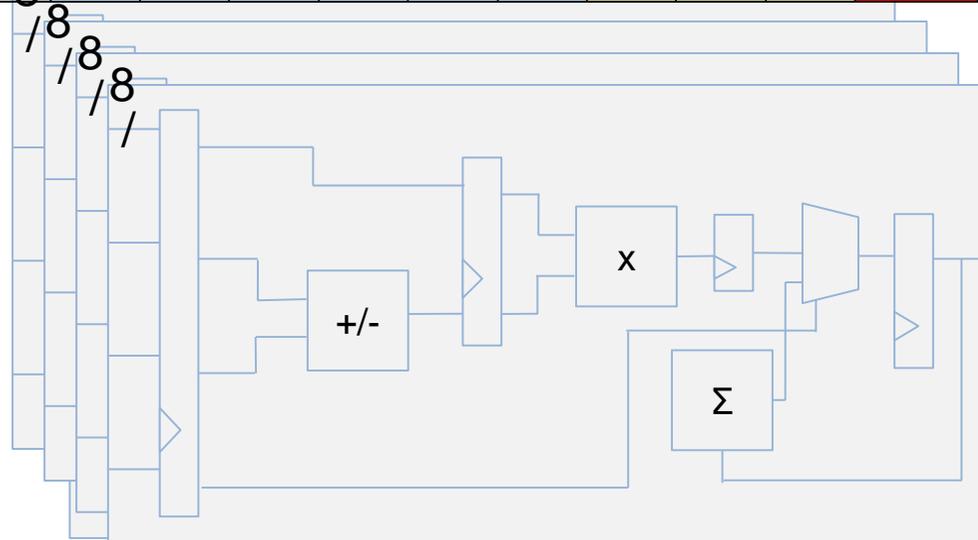
Deep Learning Practitioner

	input	conv 1	conv 2	conv 3	conv 4	conv 5	fc6	fc7	fc8	Total
Parameters (Bytes)	n/a	35K	0.4M	0.9M	1.3M	0.5M	37M	16M	4M	58 M

Off-chip RAM



Hardware/Software Engineers



Int8 Quantization

The screenshot shows the MATLAB Live Editor interface with the following code and workflow steps:

```
22     websave('trainedBlemDetNet.mat',url);
23 end
24 net2 = load('trainedBlemDetNet.mat');
25 snet_blemdetnet = net2.convnet
26 analyzeNetwork(snet_blemdetnet)
```

Create Workflow Object for trainedBlemDetNet Network

```
27 hw = dlhdl.Workflow('Network',snet_blemdetnet,'Bitstream','zcu102_single','Target');
```

Compile trainedBlemDetNet Series Network

```
28 hw.compile
```

Program Bitstream onto FPGA and Download Network Weights

```
29 hw.deploy
```

Run Prediction for One Image

```
30 filename=[pwd,'/testImages/ok1.png'];
31 img=imread(filename);
32 predictDefect(hw, img);
```

Performance Results

LastLayerLatency(seconds)	FramesNum	Total Latency	Frames/s
0.02222	1	4887512	45.0
0.00571			
0.00212			
0.00087			
0.00072			
0.00181			
0.00019			
0.01651			
0.01643			
0.00007			

Frequency: 220MHz

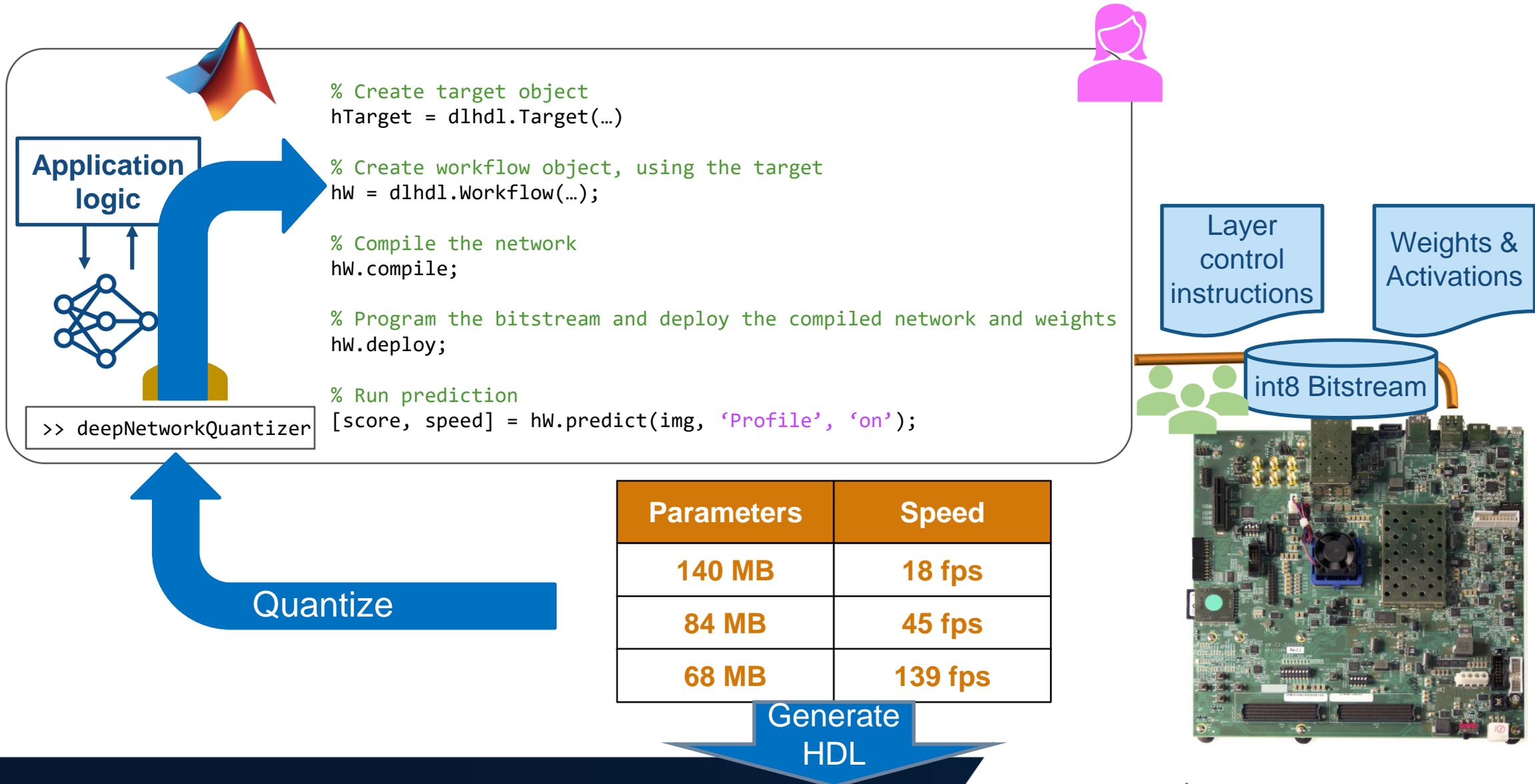
The prediction results show a bounding box around a nut in a red background, labeled 'OK'.

int8 Quantization

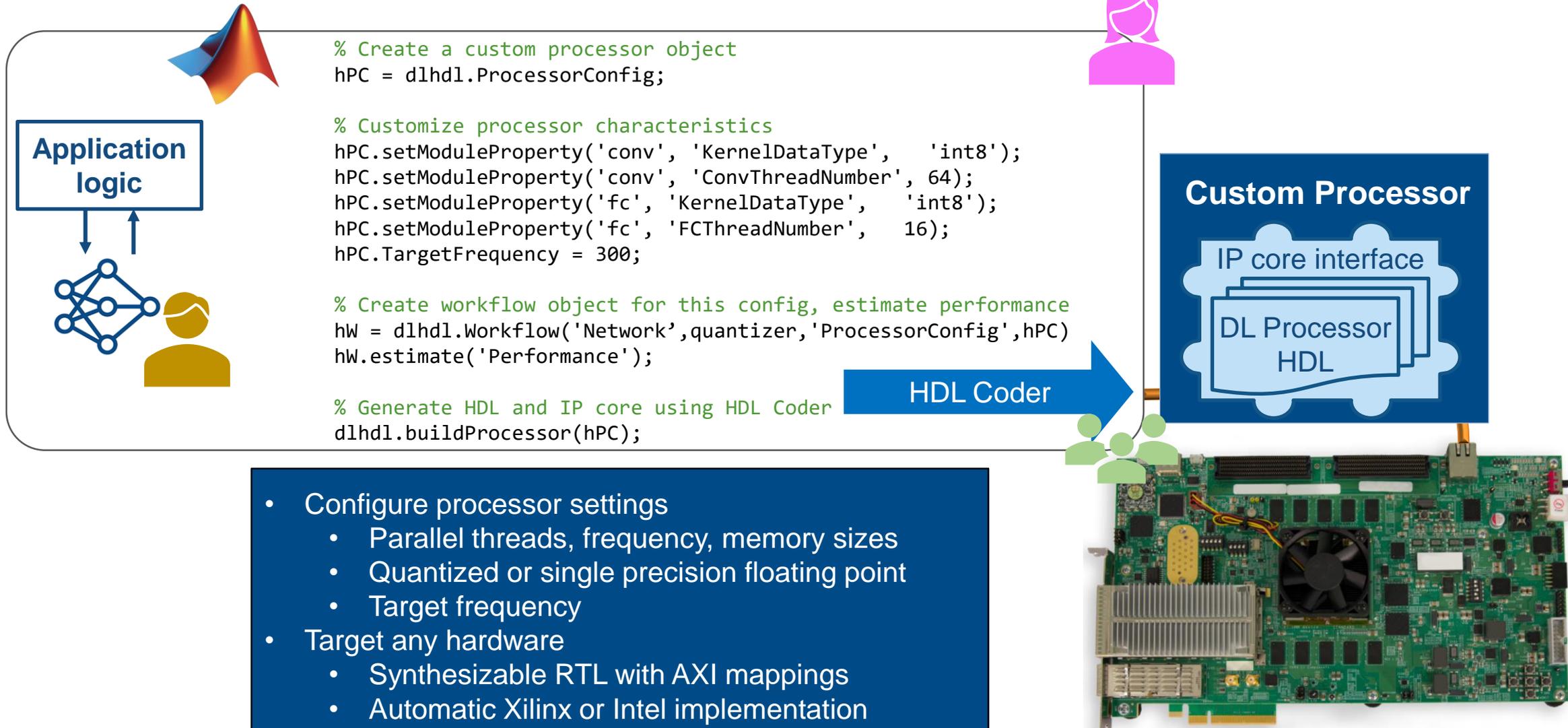
Command Window: `>>`

UTF-8 script Ln 30 Col 38

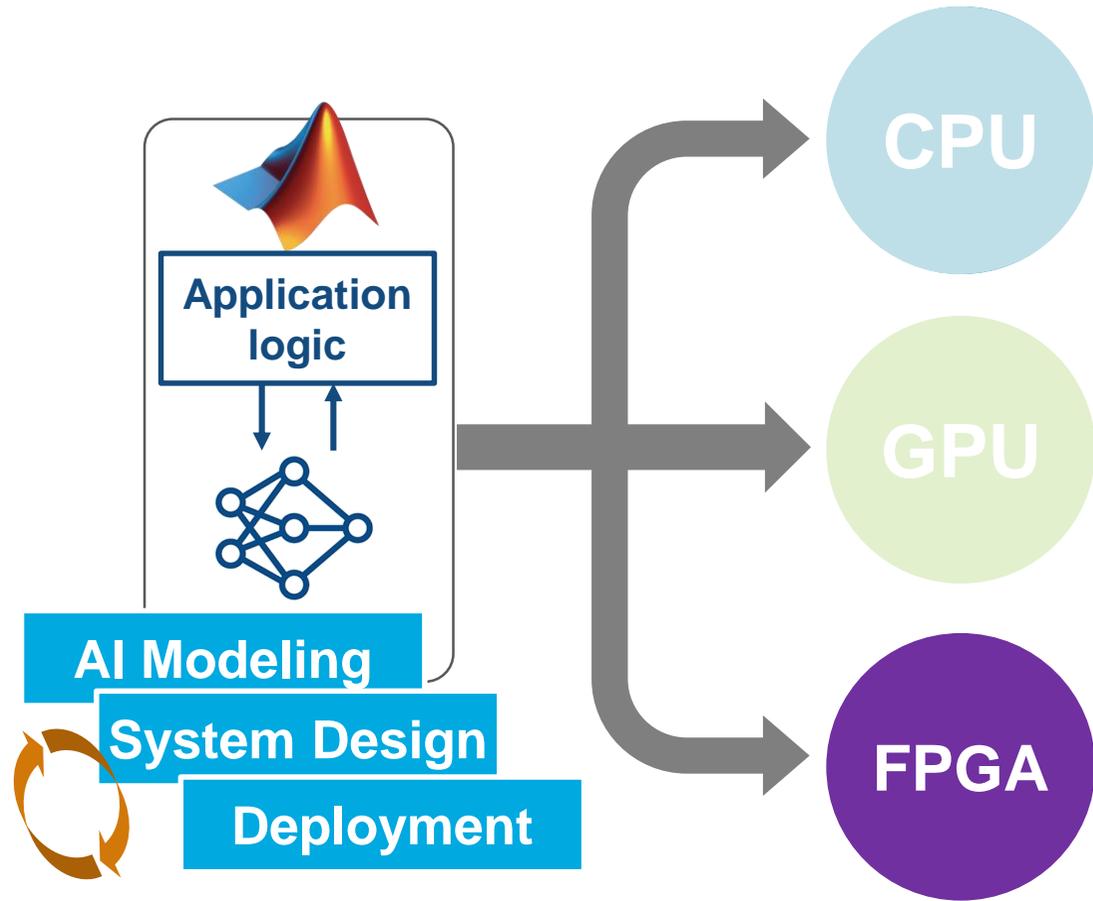
Converge on an FPGA-Optimized Deep Learning Network



Generate Custom Deep Learning Processor HDL and IP Core



Collaborate to Converge on Deep Learning FPGA Implementation



Deep Learning HDL Toolbox

-  Prototype from MATLAB
-  Tune for system requirements
-  Configure and generate RTL

Learn More

- Deep Learning Solutions in MATLAB
<https://www.mathworks.com/solutions/deep-learning.html>
- Deep Learning HDL Toolbox
<https://www.mathworks.com/products/deep-learning-hdl.html>
- Onramp: Deep Learning in MATLAB
<https://www.mathworks.com/learn/tutorials/deep-learning-onramp.html>
- MathWorks FPGA Solutions Page
<https://www.mathworks.com/solutions/fpga-asic-soc-development.html>