2020 embedded vision summit®

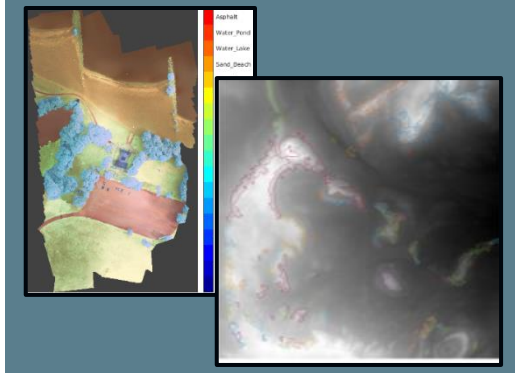
Deploying Deep Learning Application on FPGAs with MATLAB

Jack Erickson
Technical Marketing
September 2020

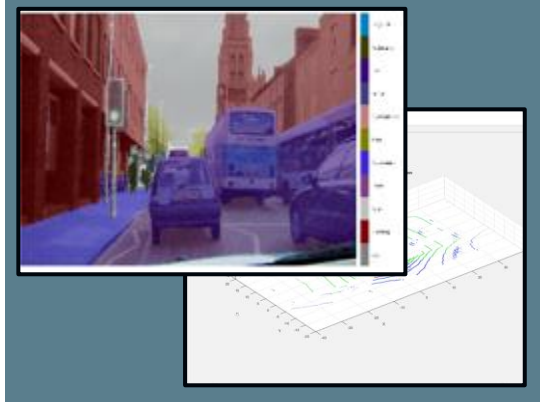
MathWorks®

Deep Learning Deployment on Embedded Devices

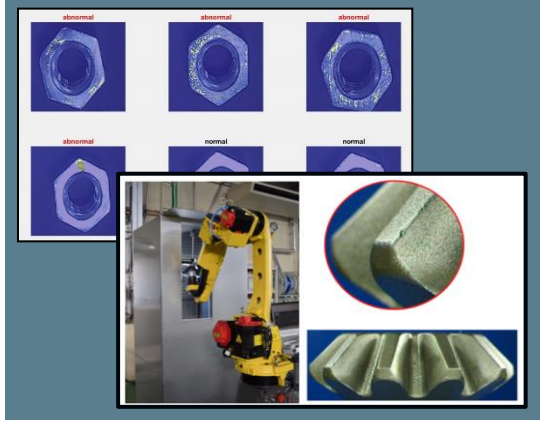
Airborne Image Analysis



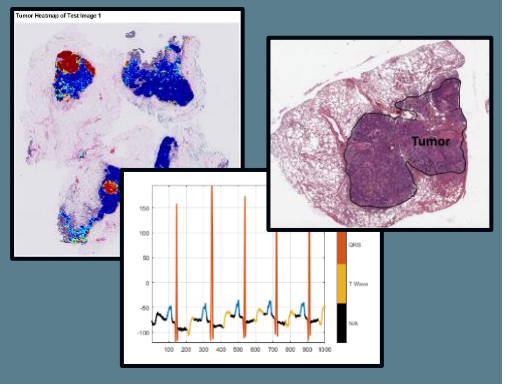
Autonomous Driving



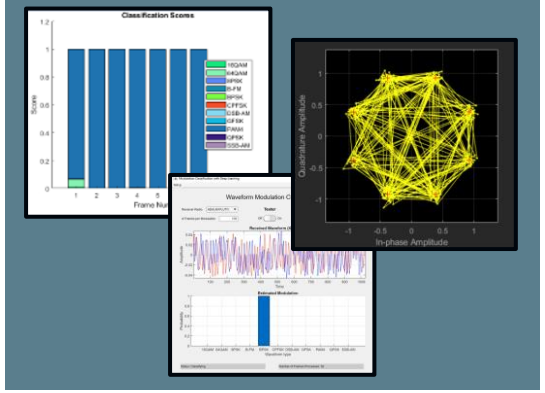
Industrial Inspection



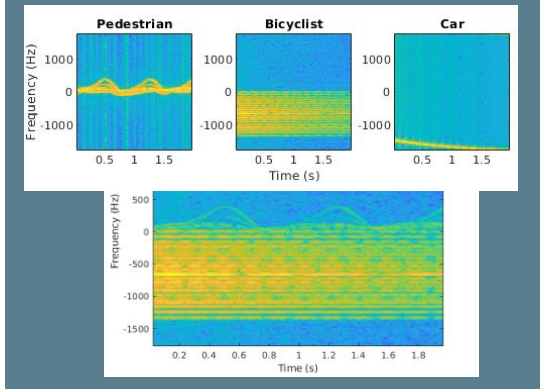
Medical Image Analysis



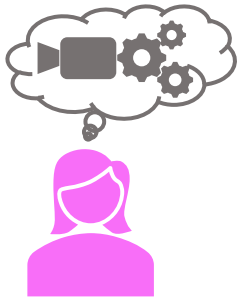
Wireless Modulation Classification



Radar Signature Classification



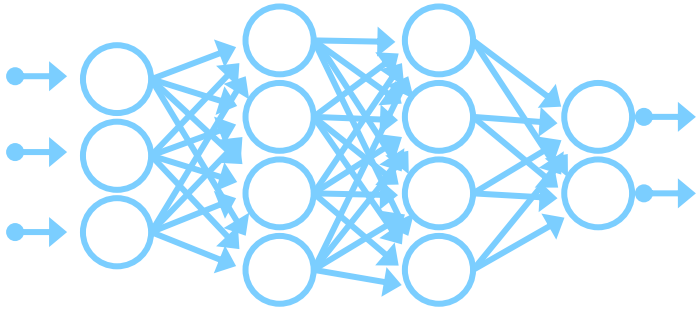
System Requirements Drive Network Design

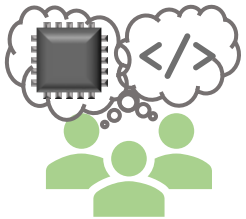


Systems Engineer

Camera specs
Accuracy
Latency
Cost
Power

Deep Learning Practitioner





Hardware/Software Engineers

Challenges of Deploying Deep Learning to FPGA Hardware: Convolution

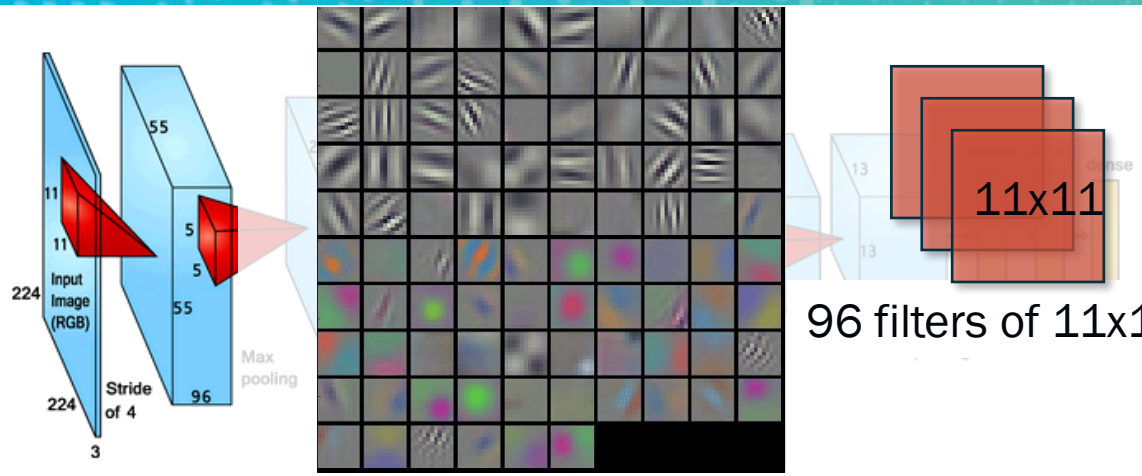
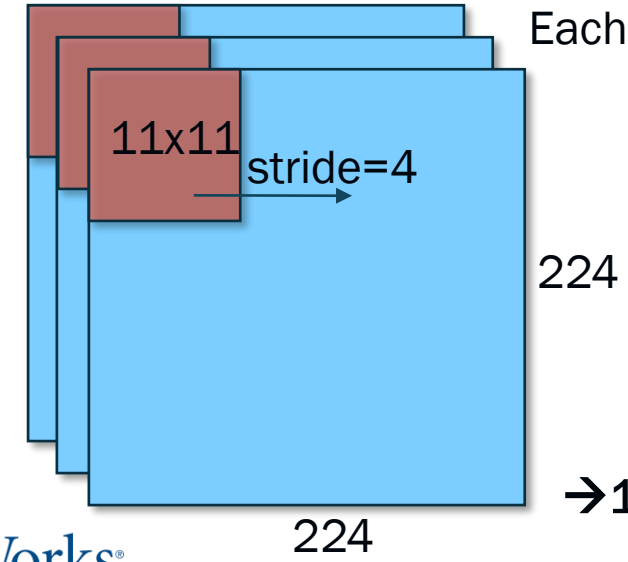
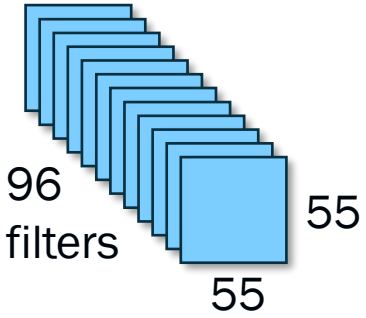


Figure 1. Illustration of AlexNet.

96 filters of 11x11x3 of 32-bit parameters →**140k bytes**



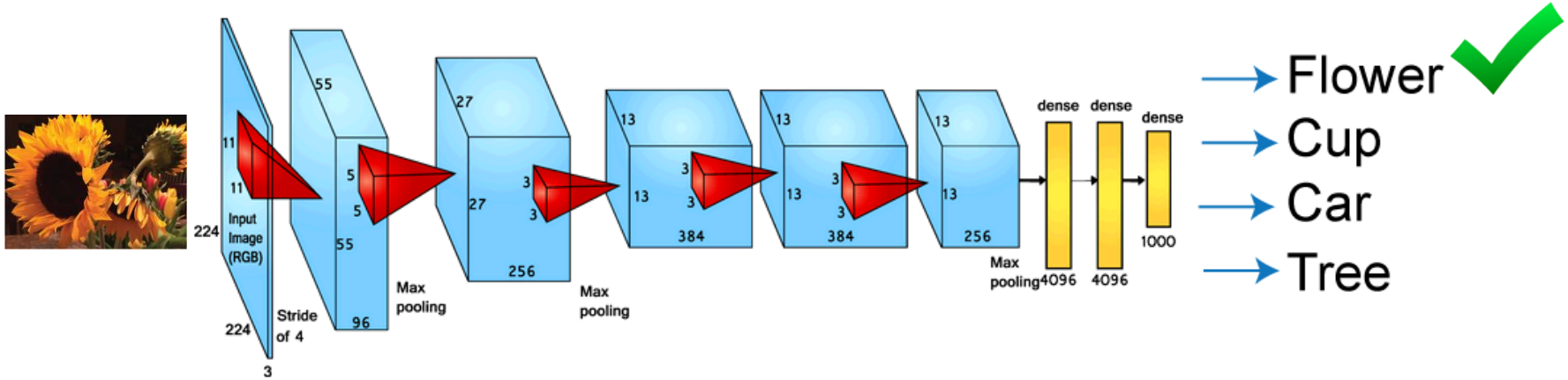
Each stride is an 11x11x3 matrix multiply-accumulate



→**1.16M bytes of activations**

→**105M** floating-point multiply operations!

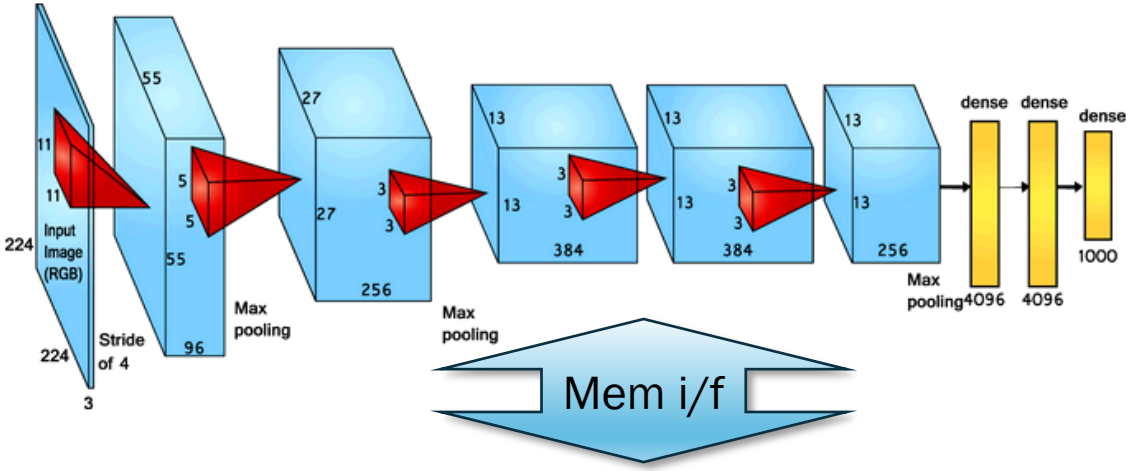
Challenges of Deploying Deep Learning to FPGA Hardware



	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

Acquire data → Resize



→ Flower
→ Cup
→ Car
→ Tree

Output / display

	input	conv 1	conv 2	conv 3	conv 4	conv 5	fc6	fc7	fc8	Total
Parameters (Bytes)										
Activations (Bytes)										
FLOPs										

Optimize

- Network /layers
- Fixed-point quantization
- Processor micro-architecture

A Collaborative AI Workflow

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

System Design

- Integration with complex systems
- System simulation
- System verification and validation

Deployment

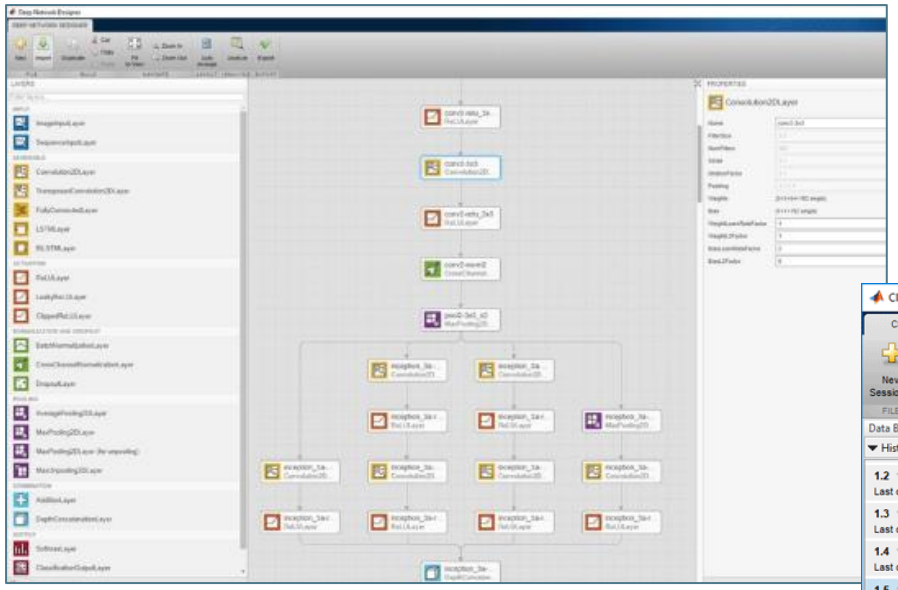
- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Iteration and Refinement

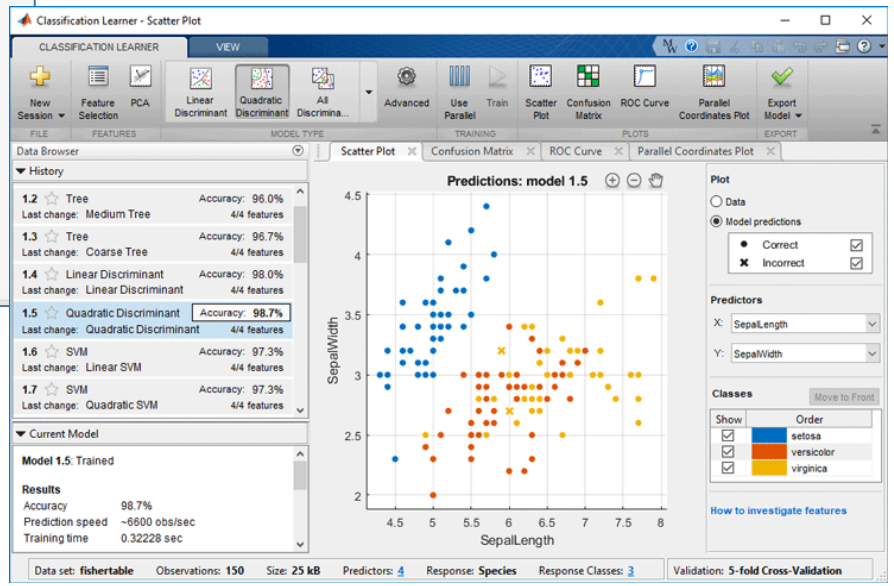
Design and Analyze Your Networks in MATLAB

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



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

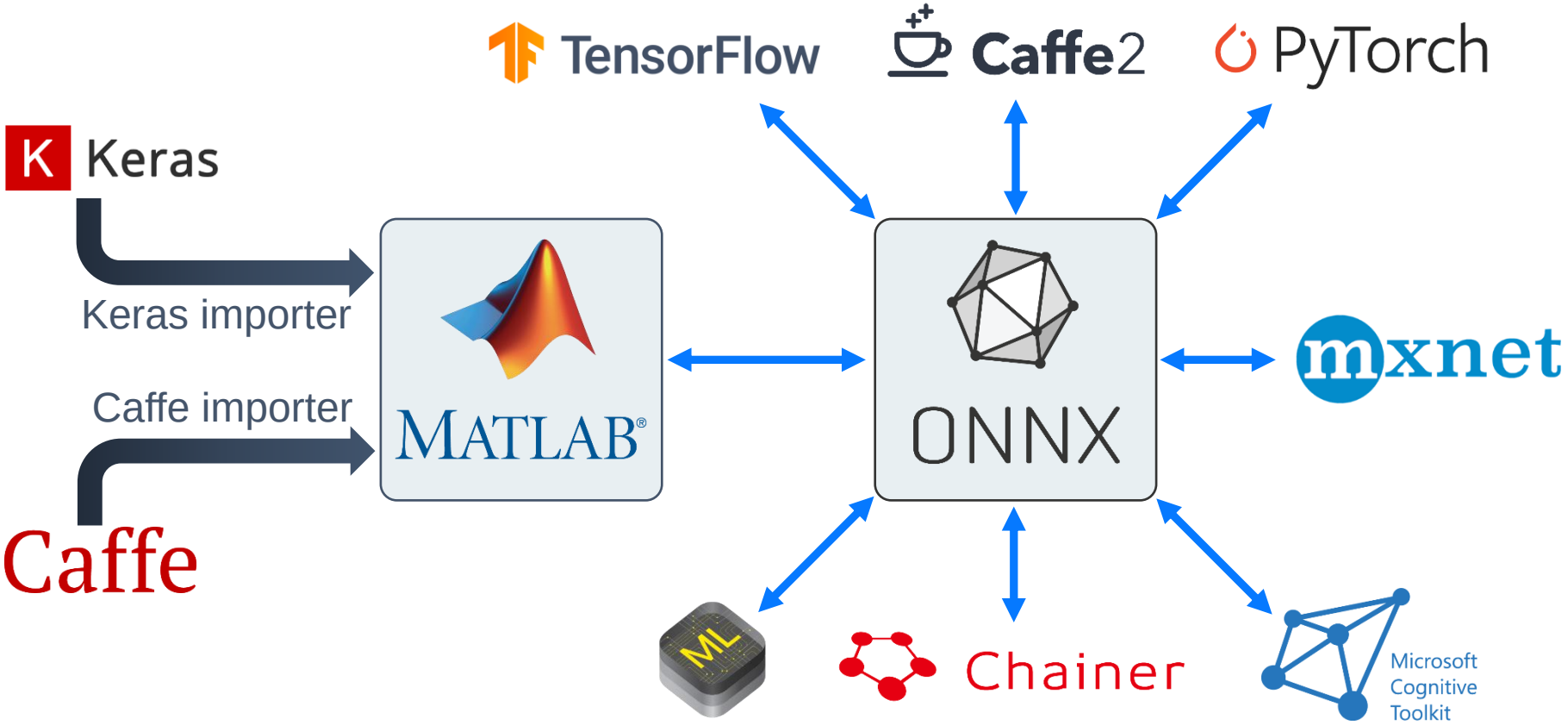


Classification Learner app to try different classifiers and find the best fit for your data set

MATLAB Interoperates with Other AI Frameworks

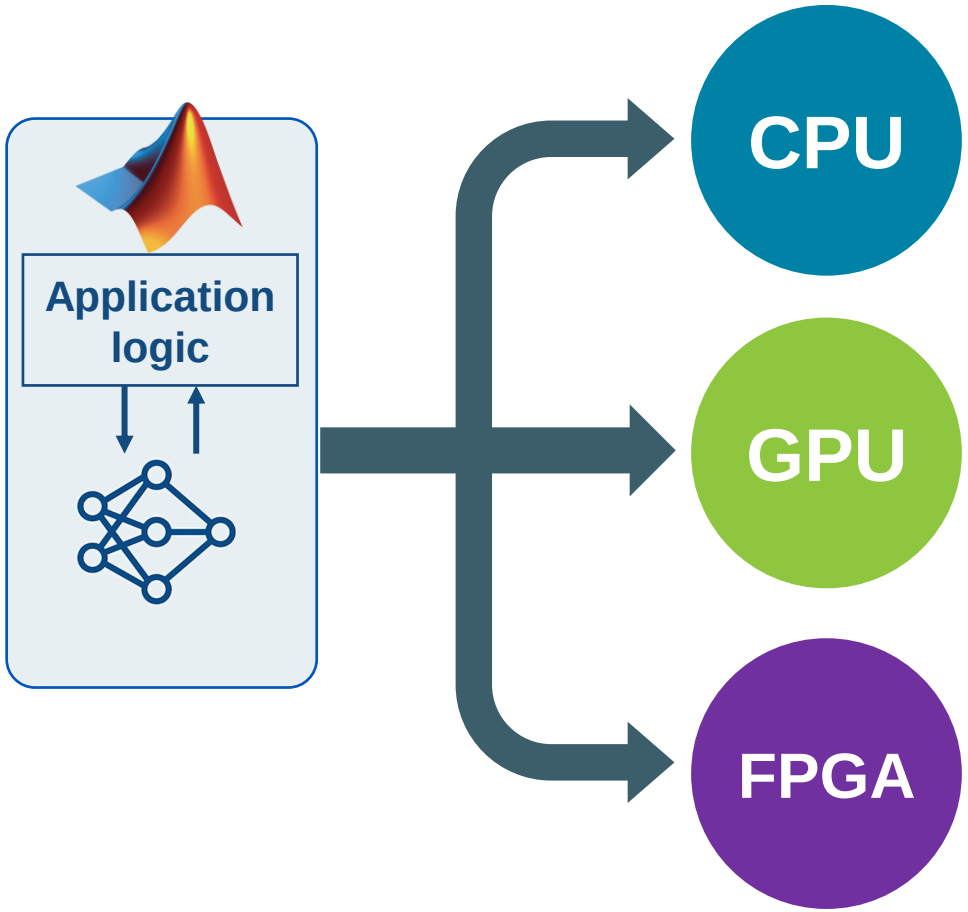
AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



MathWorks

Deploy from MATLAB to a Variety of Hardware Platforms



Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

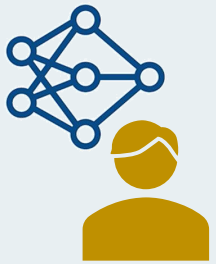
FPGA Deployment from MATLAB

MathWorks

Get Started Prototyping on FPGA with Deep Learning HDL Toolbox™

2020 embedded vision summit

Application logic



```
%% Create Target Object
hTarget = dlhdl.Target('Xilinx','Interface','Ethernet','IPAddress','10.10.10.15');

%% Create WorkFlow Object
hW = dlhdl.Workflow('network', snet, 'Bitstream', 'zcu102_single','Target',hTarget);

%% Compile the Lanenet series Network
dn = hW.compile;

%% Program Bitstream onto FPGA and Download Network Weights
hW.deploy;

%% Run prediction for one frame
outputs = hW.predict(inputImg);
```

Layer control instructions

Weights & Activations

Download

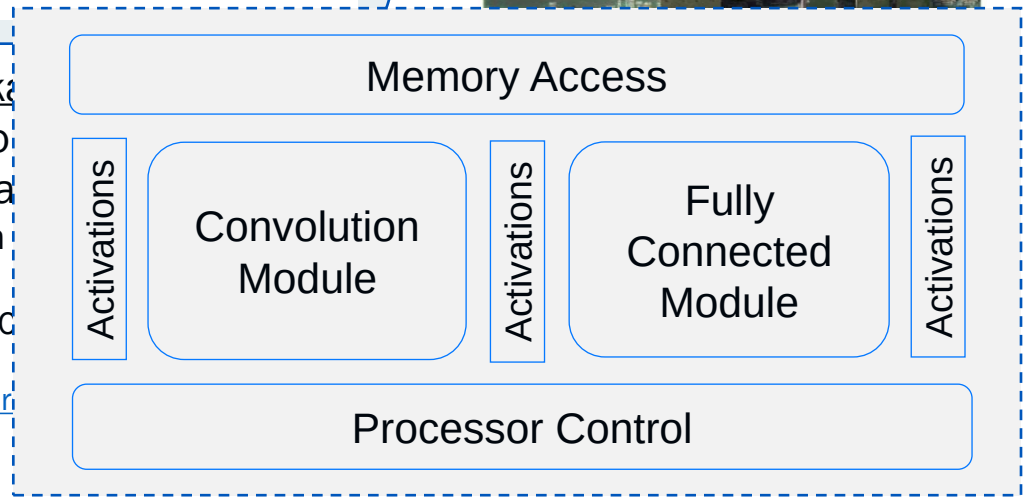
FPGA Bitstream

Hardware support packa

Deep learning processo

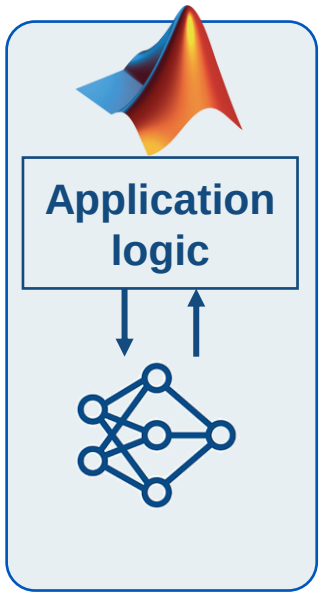
external memory interfa

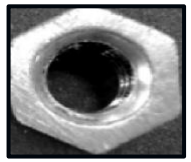
- Int8 or single precision
- Supported boards:
  - Xilinx: ZCU102 or ZC
  - Intel: Arria10 SoC
- http://mathworks.com/har



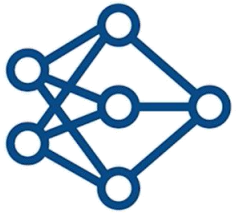
MathWorks

Defect Detection Example

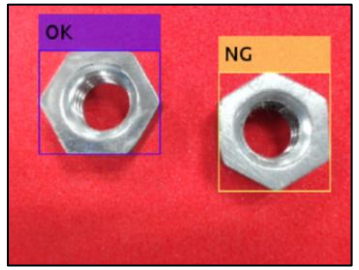




Pre-processing: Extract regions and resize



Inference: Predict using trained network → FPGA



Post-processing: Annotate and label

Run Deep Learning on FPGA from MATLAB

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

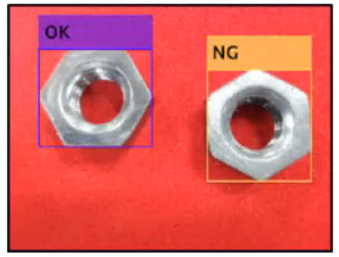
Search Documentation Sign In

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

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

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

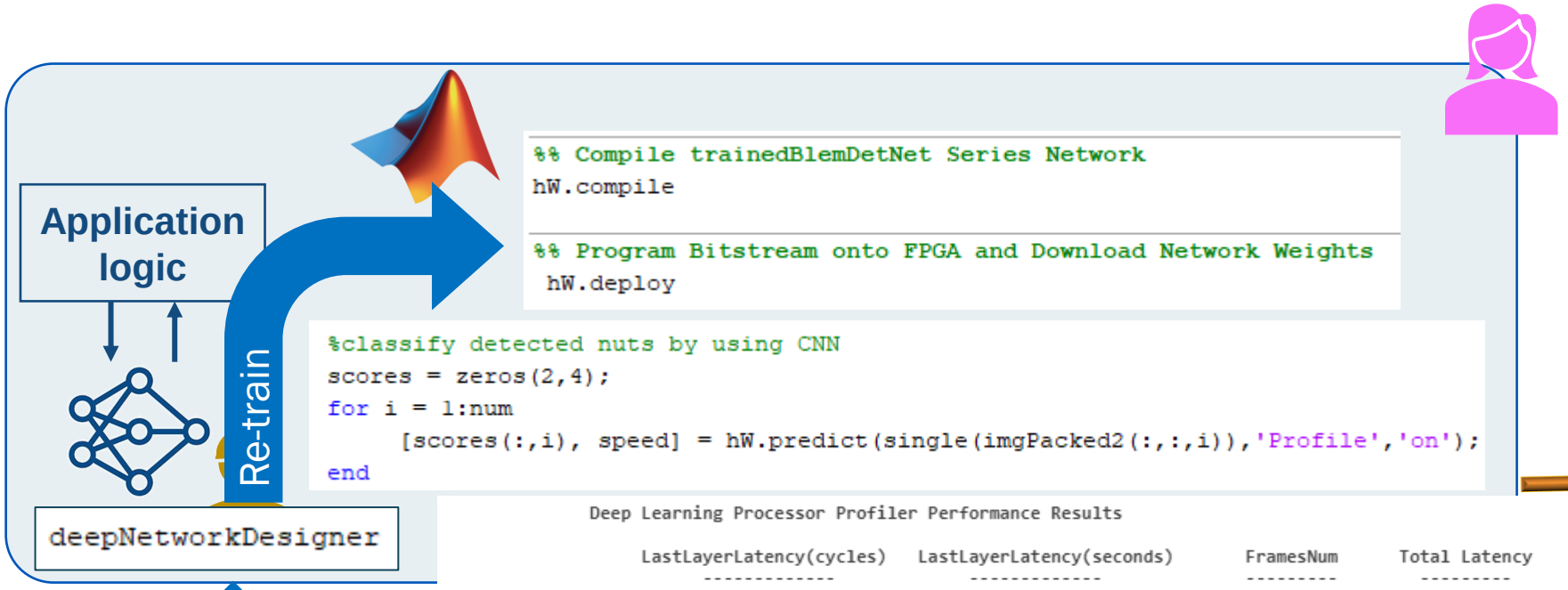
```
unzip('dnnfpga_defectdetection.zip');
[newDir, origDir] = cloneSetupDir('dnnfpga_defectdetection');
cd(newDir);
```

Command Window

>>

UTF-8 script Ln 10 Col 31

Prototyping: Design Exploration and Customization




```
%% Compile trainedBlemDetNet Series Network
hW.compile

%% Program Bitstream onto FPGA and Download Network Weights
 hW.deploy
```

```
%classify detected nuts by using CNN
scores = zeros(2,4);
for i = 1:num
     [scores(:,i), speed] = hW.predict(single(imgPacked2(:,:,i)),'Profile','on');
end
```

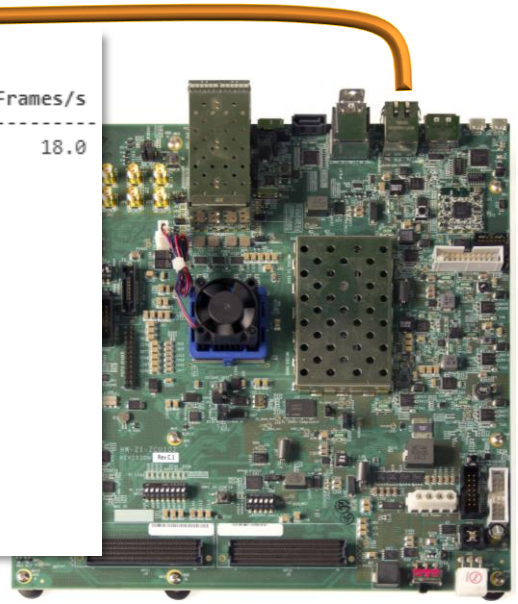
Deep Learning Processor Profiler Performance Results

	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

Layer control instructions

Weights & Activations



Design Exploration and Customization

HOME | PLOTS | APPS | LIVE EDITOR | INSERT | VIEW

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

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

DefectDetectionExample.mlx

Create Target Object

```
hT = dlhdl.Target('Xilinx','Interface','Ethernet', 'IPAddress', '10.10.10.15')
```

Create Workflow Object for trainedDefNet Network

```
hW = dlhdl.Workflow('Network',snet_defnet,'Bitstream','zcu102_single','Target',h
```

Compile trainedDefNet Series Network

```
hW.compile
```

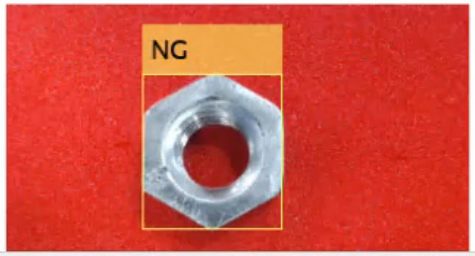
Program Bitstream onto FPGA and Download Network Weights

```
hW.deploy
```

Run Prediction for One Image

```
unzip('testImages.zip')

filename=[pwd,'/testImages/ng1.png'];
img=imread(filename);
predictDefect(hW, img);
```

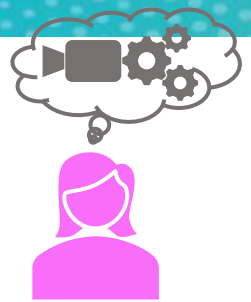
```
              Deep Learning Processor Profiler Performance Results

                   LastLayerLatency(cycles)   LastLayerLatency(seconds)
                         -------------             -------------
Network                     12213262                  0.05551
    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
```



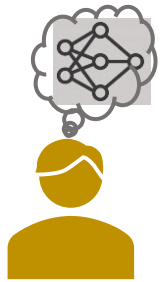
Command Window

>>

Optimizing Deep Learning Applications Requires Collaboration

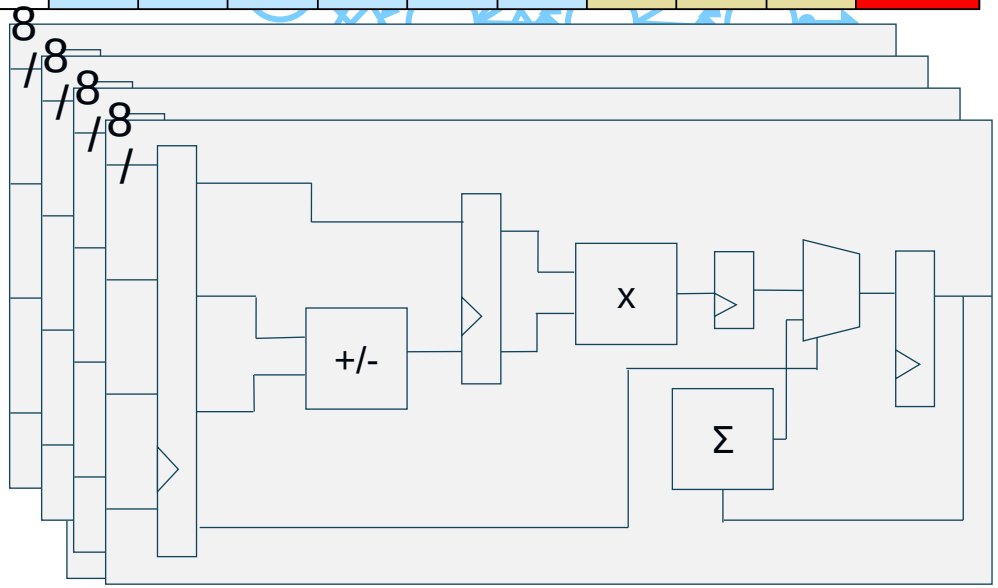


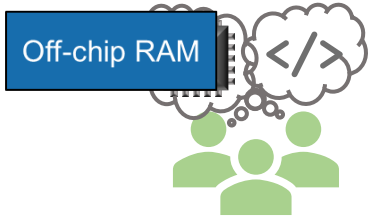
Systems Engineer



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





Hardware/Software Engineers

MathWorks

INT8 Quantization

HOME | PLOTS | APPS | LIVE EDITOR | INSERT | VIEW

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

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

DefectDetectionExample.mlx *

```
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','Targe
```

Compile trainedBlemDetNet Series Network

```
28      hW.compile
```

Program Bitstream onto FPGA and Download Network Weights

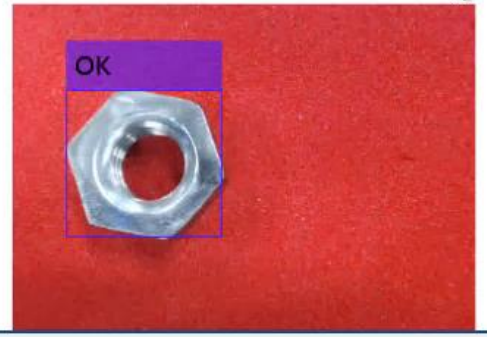
```
29      hW.deploy
```

Run Prediction for One Image

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

int8 Quantization

```
er 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
s: 220MHz
```



Command Window

>>

UTF-8 | script | Ln 30 | Col 38

Deep Learning HDL Toolbox

Iterate and Converge on Deep Learning FPGA Deployment from MATLAB

Application logic

```
% Create target object
hTarget = dlhdl.Target(…)

% Create workflow object, using the target
hW = dlhdl.Workflow(…);

% Compile the network
hW.compile;

% Program the bitstream and deploy the compiled network and weights
hW.deploy;

% Run prediction
[score, speed] = hW.predict(img, 'Profile', 'on');
```

```
>> deepNetworkQuantizer
```

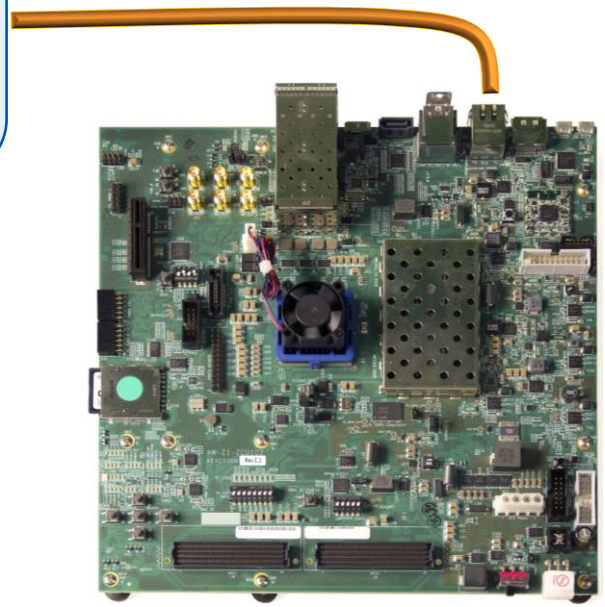
Quantize

Generate HDL

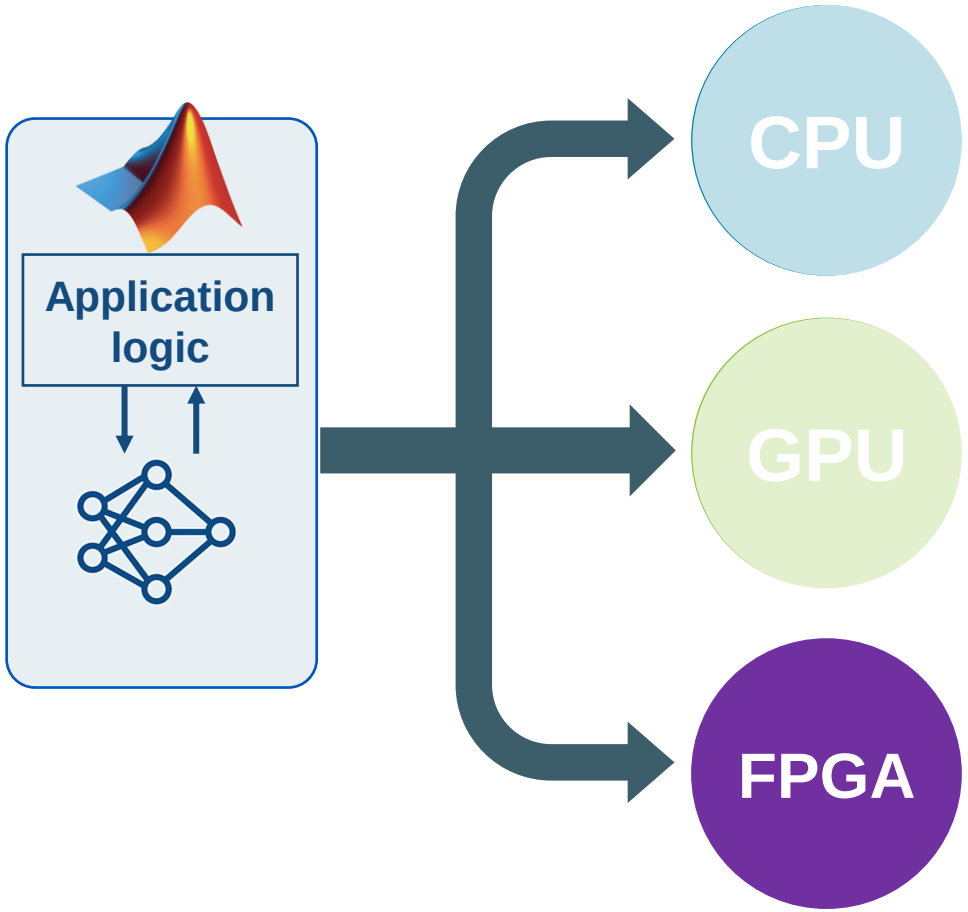
Layer control instructions

Weights & Activations

Parameters	Speed
140 MB	18 fps
84 MB	45 fps
68 MB	139 fps



Deploy from MATLAB to a Variety of Hardware Platforms



Deep Learning HDL Toolbox

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

Resource Slide

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