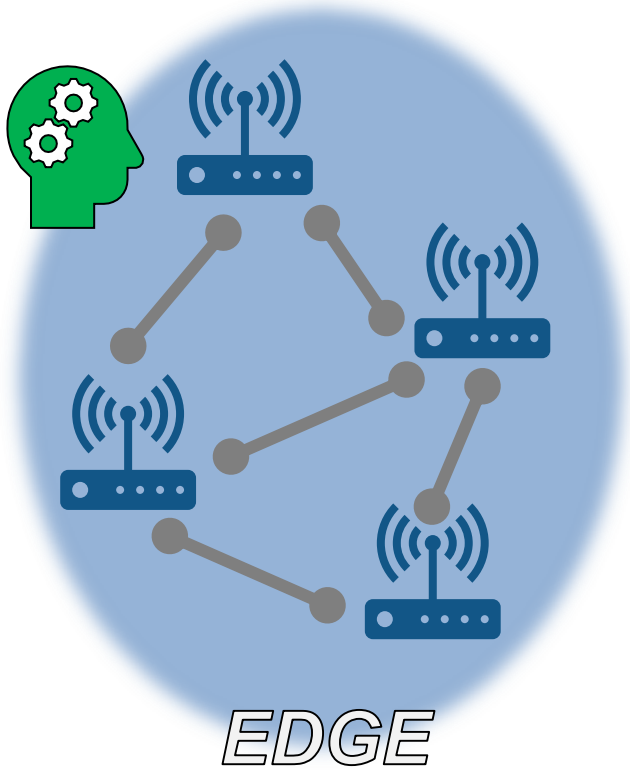
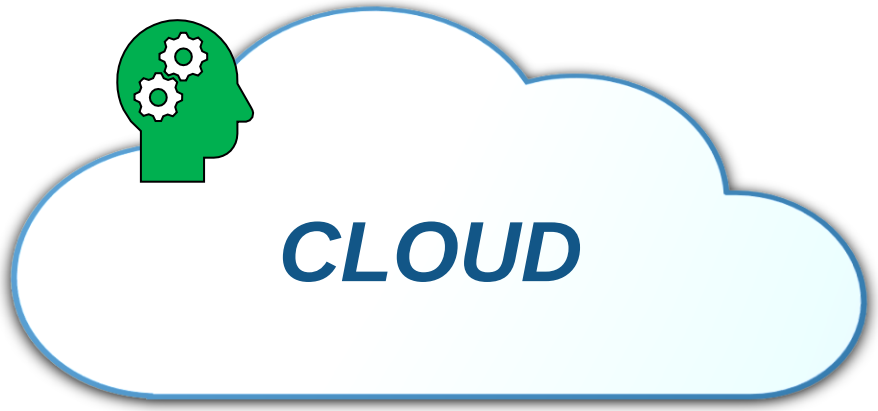
Bringing AI to Real-Time Insights using FPGA, GPU and CPU

Cynthia Cudicini – Technical Manager – MathWorks France
ccudicin@mathworks.com

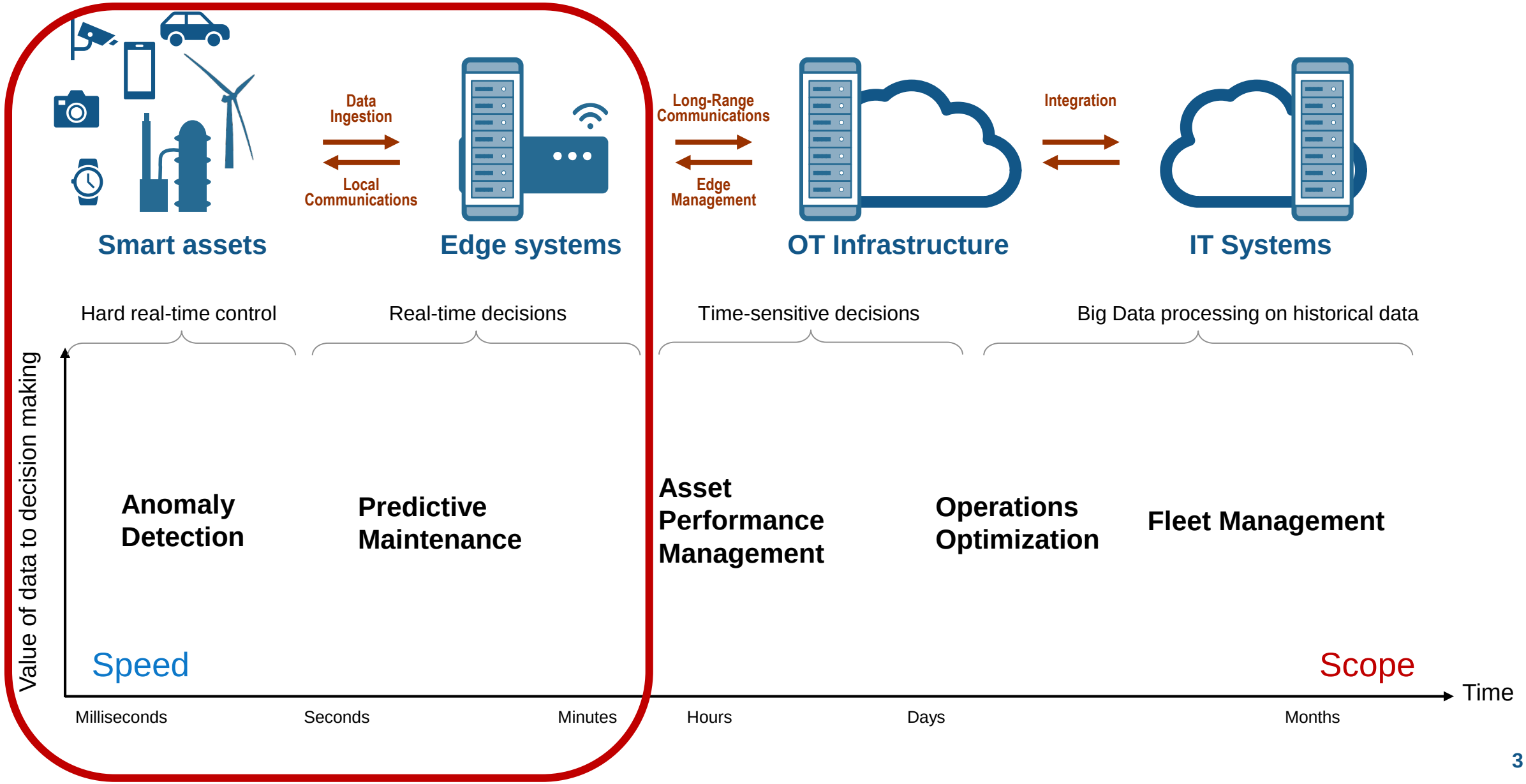
Artificial Intelligence Opportunities in « Internet of Everything »



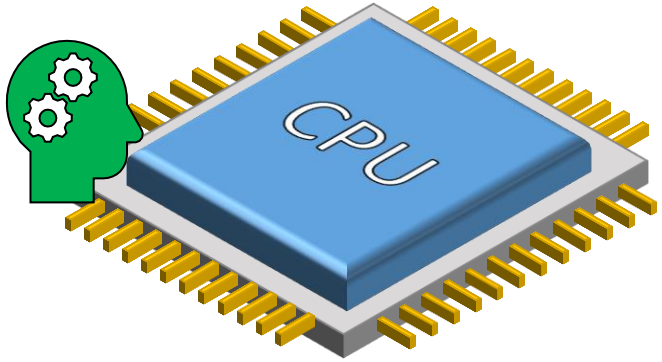


Do the right thing at the right place

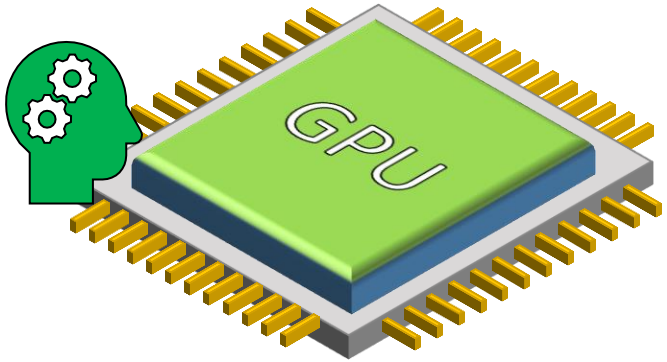
Artificial Intelligence Opportunities in « Internet of Everything »



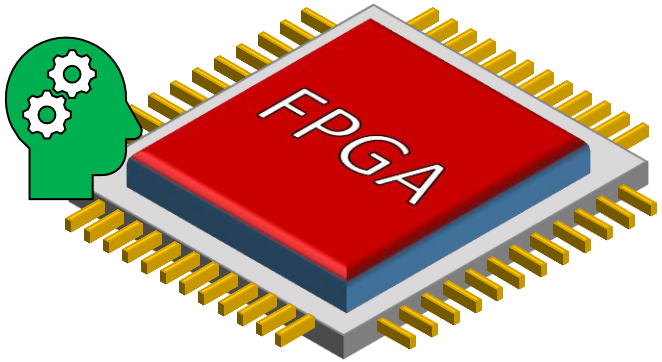
Embedded Targets & Mitigations



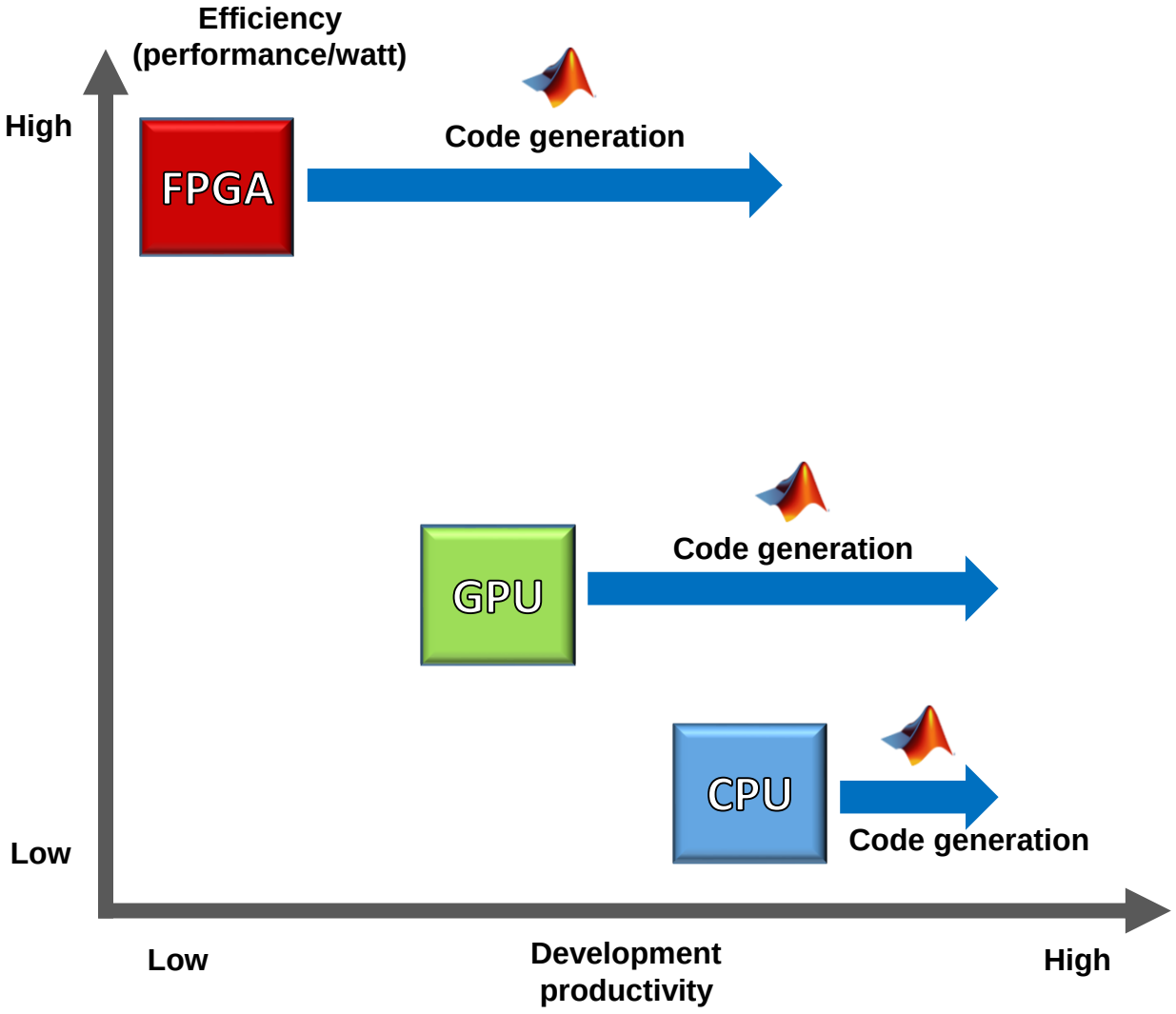
- C/C++ programing language
- Sequential processing



- CUDA/ OpenCL programing language
- Partly parallel processing



- VHDL/Verilog programing language
- Partly parallel processing



Example: Generates CUDA from MATLAB - saxpy

Scalarized MATLAB

```
for i = 1:length(x)
    z(i) = a .* x(i) + y(i);
end
```

Vectorized MATLAB

```
z = a .* x + y;
```

GPU Coder

CUDA

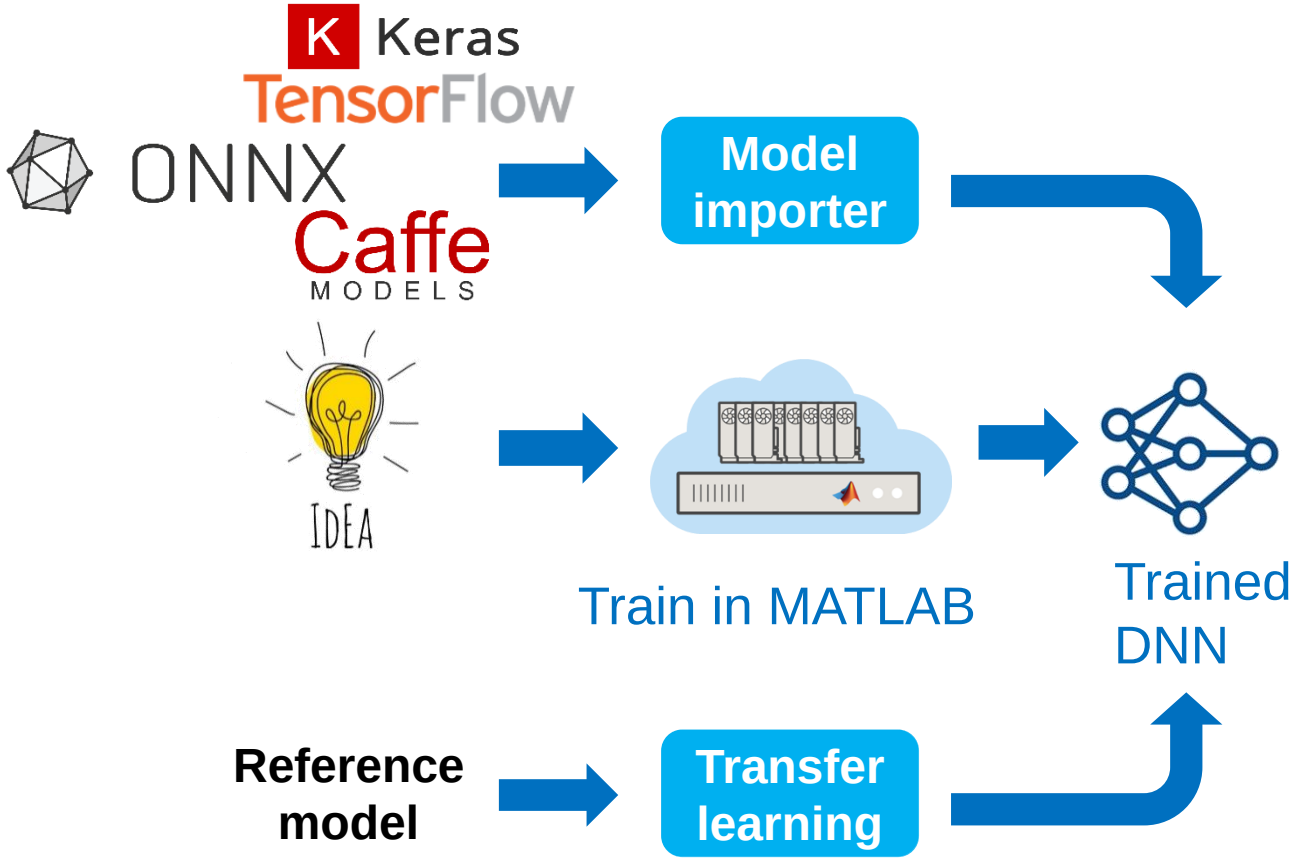
```
cudaMalloc(&gpu_z, 8388608UL);
cudaMalloc(&gpu_x, 4194304UL);
cudaMalloc(&gpu_y, 4194304UL);
cudaMemcpy((void *)gpu_y, (void *)y, 4194304UL, cudaMemcpyHostToDevice);
cudaMemcpy((void *)gpu_x, (void *)x, 4194304UL, cudaMemcpyHostToDevice);
saxpy_kernel1<<<dim3(2048U, 1U, 1U), dim3(512U, 1U, 1U)>>>(gpu_y, gpu_x, a,
  gpu_z);
cudaMemcpy((void *)z, (void *)gpu_z, 8388608UL, cudaMemcpyDeviceToHost);
cudaFree(gpu_y);
cudaFree(gpu_x);
cudaFree(gpu_z);
```

CUDA kernel for GPU parallelization

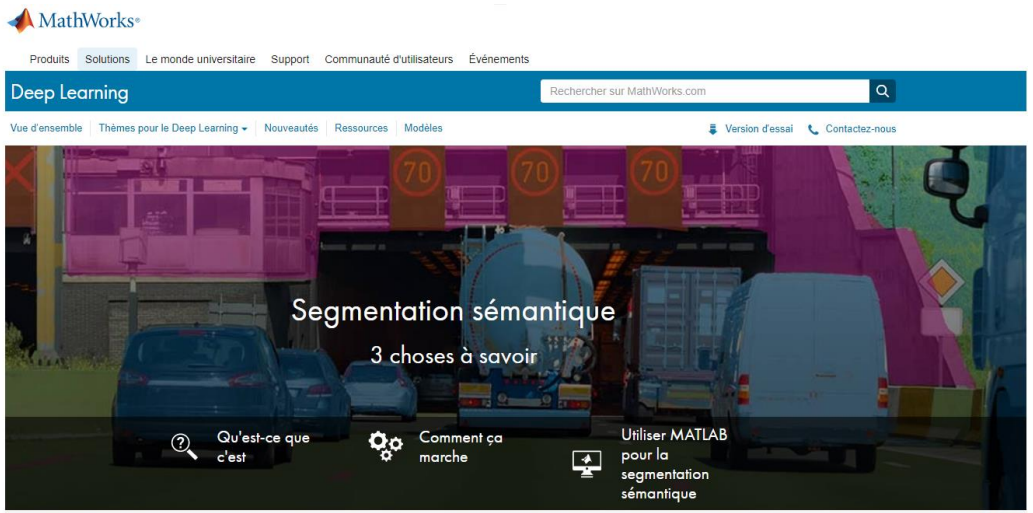
```
static __global__ __launch_bounds__(512, 1) void saxpy_kernel1(const real32_T *y,
  const real32_T *x, real32_T a, real_T *z)
{
  int32_T i;

  i = (int32_T)(((((gridDim.x * gridDim.y * blockIdx.z + gridDim.x * blockIdx.y)
                  + blockIdx.x) * (blockDim.x * blockDim.y * blockDim.z) +
                 threadIdx.z * blockDim.x * blockDim.y) + threadIdx.y *
                blockDim.x) + threadIdx.x);
  if (!(i >= 1048576)) {
    z[i] = (real_T)(a * x[i] + y[i]);
  }
}
```

Deep Neural Network Design and Training



- **Design in MATLAB**
  - **Manage** large data sets
  - **Automate** data labeling
  - **Easy access** to models
- **Training in MATLAB**
  - **Acceleration** with GPU's
  - **Scale** to clusters



Quelques lignes de code MATLAB® suffisent pour développer des modèles de Deep Learning, sans être nécessairement un expert du domaine. Découvrez comment MATLAB peut vous aider à exécuter des tâches de Deep Learning.

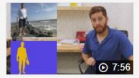
- Accédez facilement aux derniers modèles, y compris **GoogLeNet**, **VGG-16**, **VGG-19**, **AlexNet**, **ResNet-50**, **ResNet-101** et **Inception-v3**.
- Accélérez les algorithmes sur **NVIDIA® GPU**, le cloud et les ressources de datacenter sans aucun élément de programmation particulier.
- Créez, modifiez et analysez des architectures de réseaux de neurones profonds à l'aide des applications MATLAB et des outils de **visualisation**.
- Automatisez la labélisation de la vérité terrain (**ground-truth labeling**) des données image, vidéo et audio à l'aide d'applications.
- Travaillez avec des modèles **Caffe** et **TensorFlow-Keras.**
- MATLAB supporte **ONNX™** pour vous permettre de collaborer avec des collègues à l'aide d'infrastructures comme **PyTorch** et **MxNet**.

MATLAB pour le Deep Learning

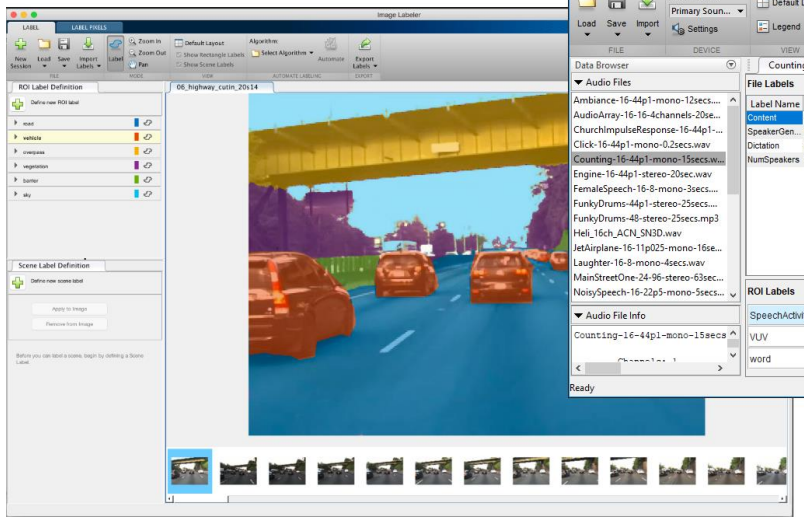
Qu'est-ce que la segmentation sémantique ?

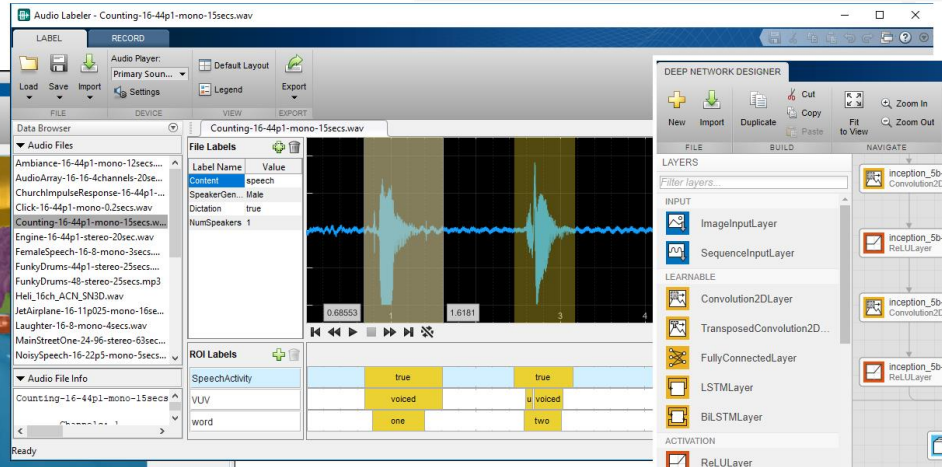
La segmentation sémantique est un algorithme de Deep Learning qui associe une étiquette ou une catégorie à chaque pixel d'une image. Elle permet de reconnaître un ensemble de pixels qui forment des catégories distinctes. Par exemple, un véhicule autonome doit identifier des véhicules, des piétons, des panneaux de signalisation, des trottoirs et autres éléments de l'environnement routier.

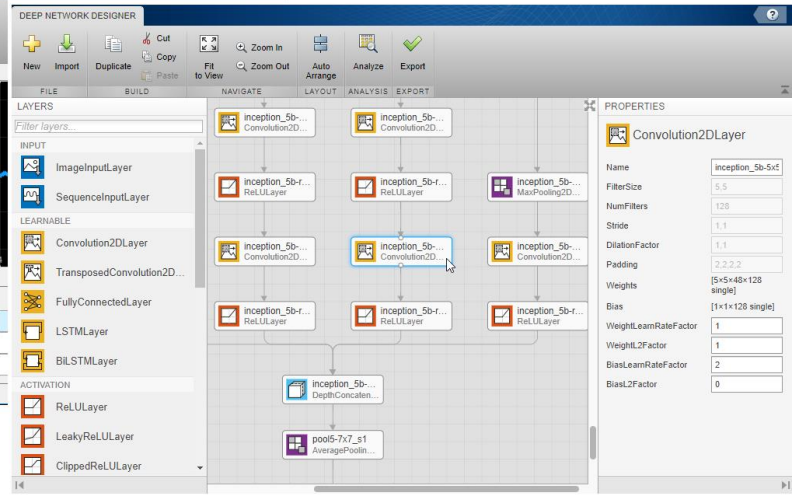
La segmentation sémantique intervient dans de nombreuses applications telles que la conduite autonome, l'imagerie médicale et les contrôles industriels.



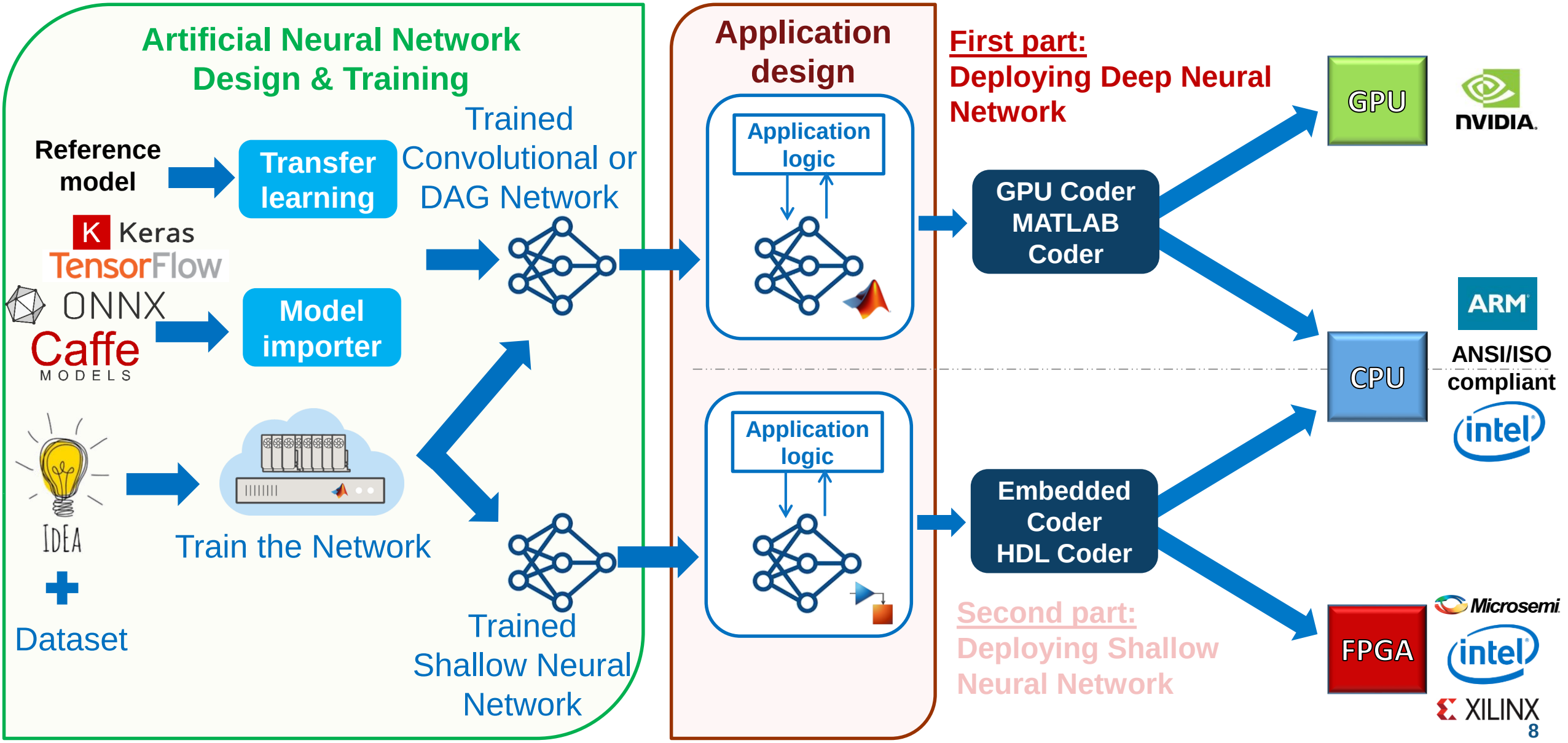
Vue d'ensemble de la segmentation sémantique



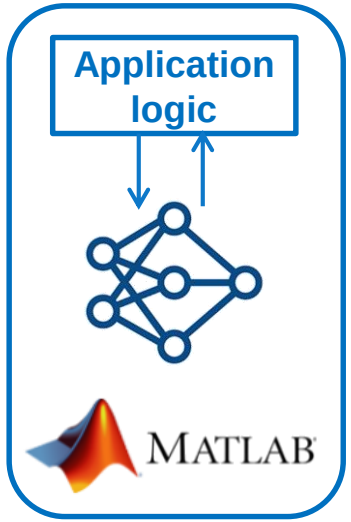




MathWorks Workflows: Neural Network to Embedded Targets



Deploying to CPUs

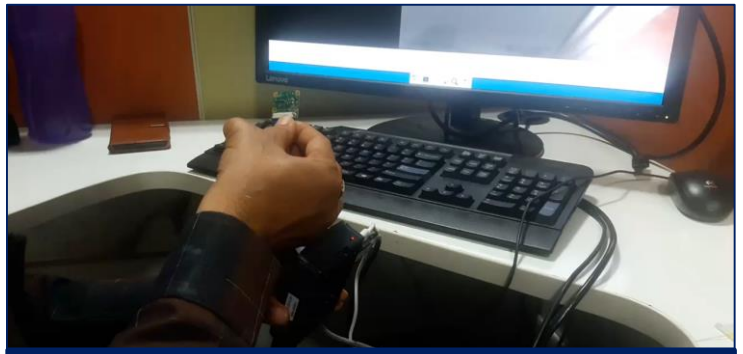


MATLAB Coder

Intel MKL-DNN Library

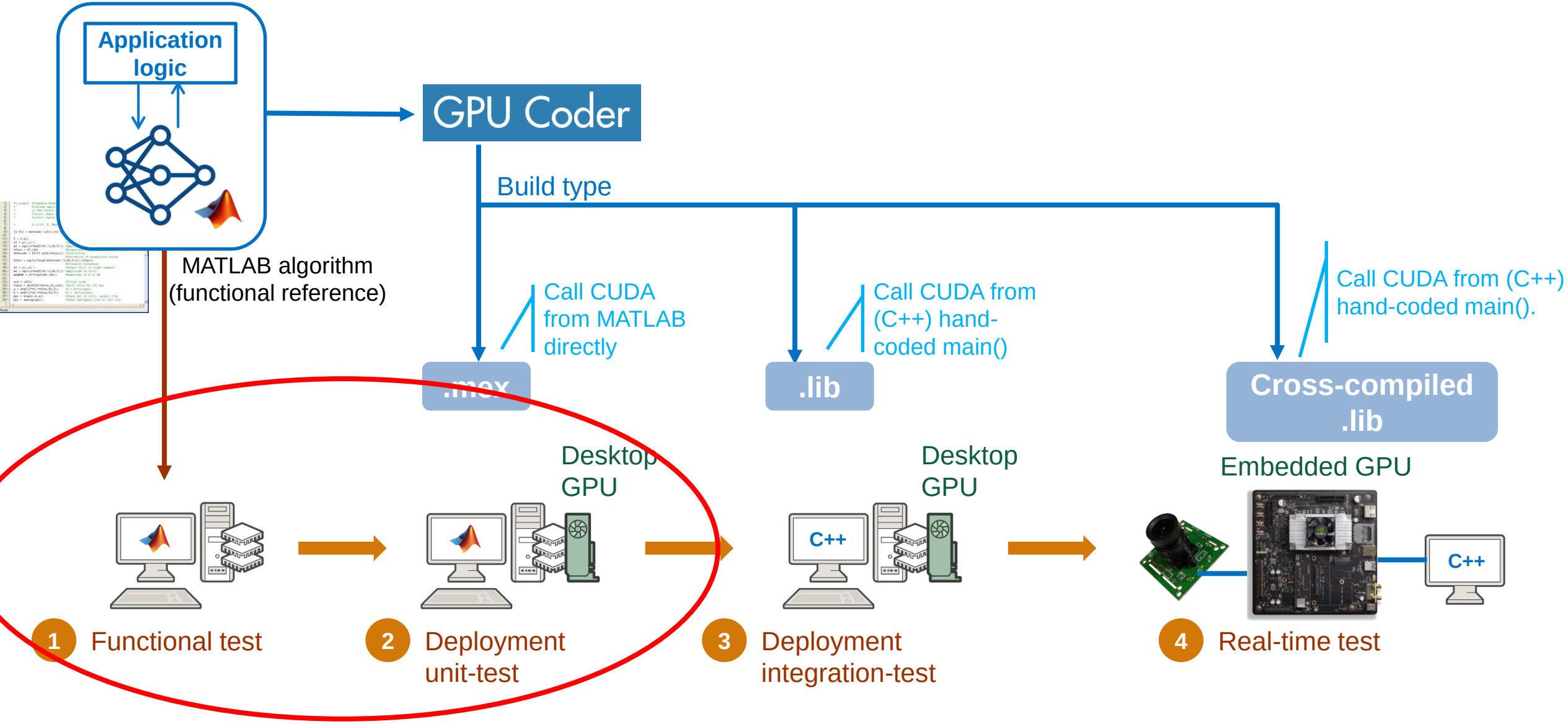
ARM Compute Library

Desktop CPU

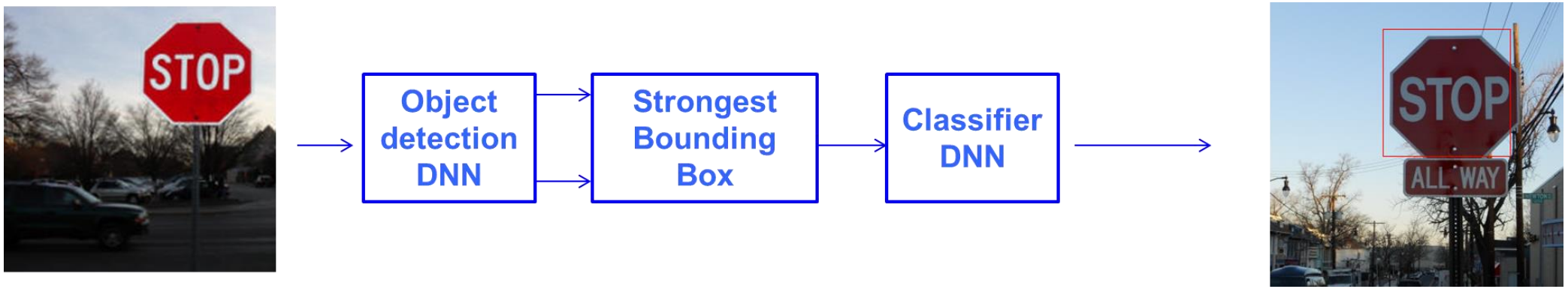


Raspberry Pi board

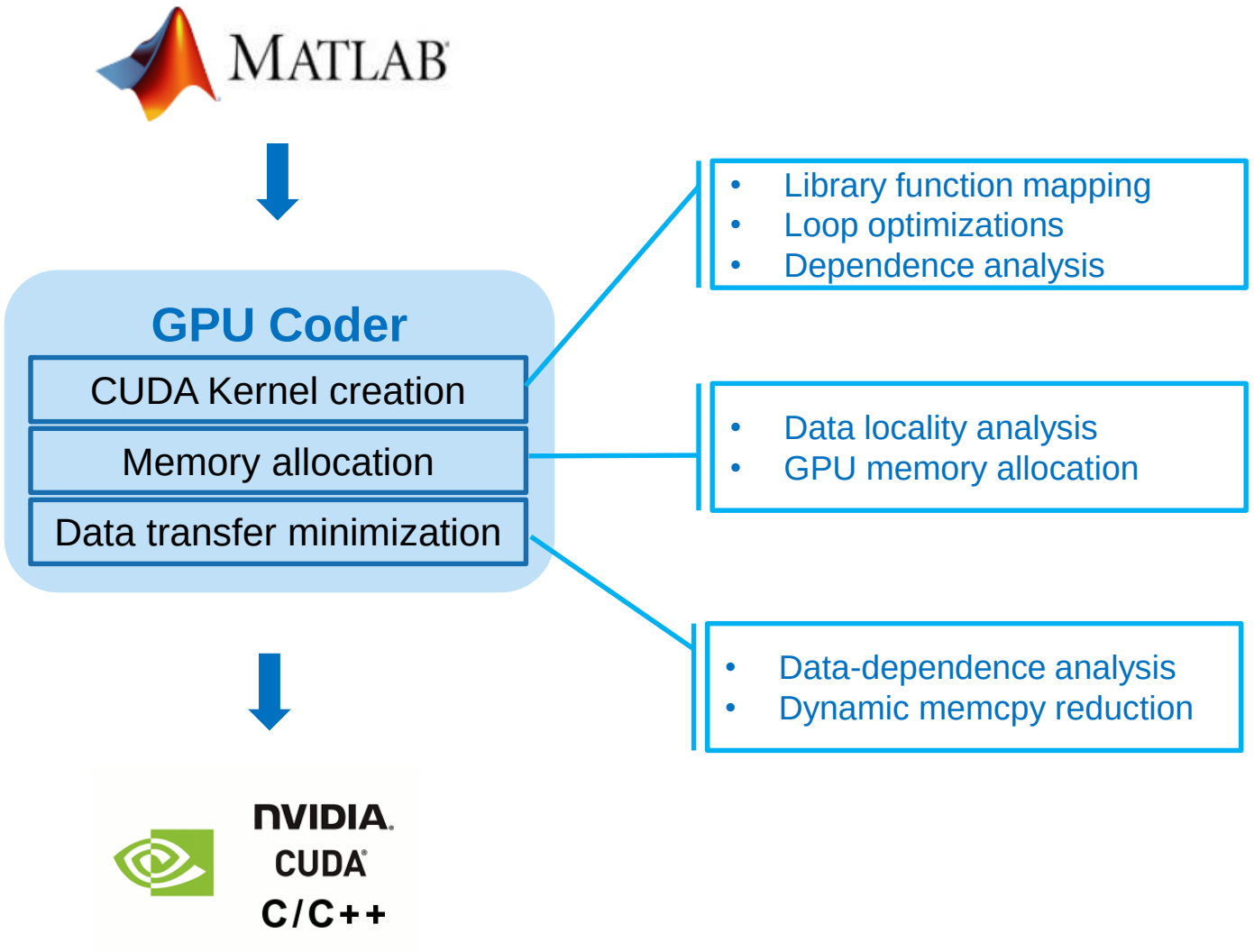
Deploying to GPUs



Example: Traffic Sign Detection and Recognition



GPU Coder Helps You Deploy to GPUs Faster



GPU Coder automatically extracts parallelism from MATLAB

1. Scalarized MATLAB
   ("for-all" loops)

```
%% Pixel processing on the height/width of an image
for i = 1:height
    for j = 1:width
        tmpVal = (width*height);
        for x = 1:width
            disValTmp = temp(x,i);
            dist = ((j-x)^2 + disValTmp^2 );
            if (dist < tmpVal)
                tmpVal = single(dist);
            end
        end
        out(i,j) = tmpVal;
    end
end
```

2. Vectorized MATLAB
   (math operators and library functions)

```
%% Parallel element-wise math to compute
%  Restoration with inverse Koschmieder's law
factor=1.0./(1.0-(diff_im));
restoreOut(:,:,1)= (input(:,:,1)-diff_im).*factor;
restoreOut(:,:,2)= (input(:,:,2)-diff_im).*factor;
restoreOut(:,:,3)= (input(:,:,3)-diff_im).*factor;
```

Infer CUDA kernels from MATLAB loops

3. Composite functions in MATLAB
   (maps to cuBlas, cuFFT, cuSolver, cuDNN, TensorRT)

```
C = A * B; % cuBLAS

y = fft(in); % cuFFT

z = A \ B; % cuSolver
```

```
predictions = detectionnet.activations(img_rz,56,'OutputAs','channels'); % cuDNN or TensorRT
```

Library replacement

Optimizing CPU-GPU Data Movement is a Challenge

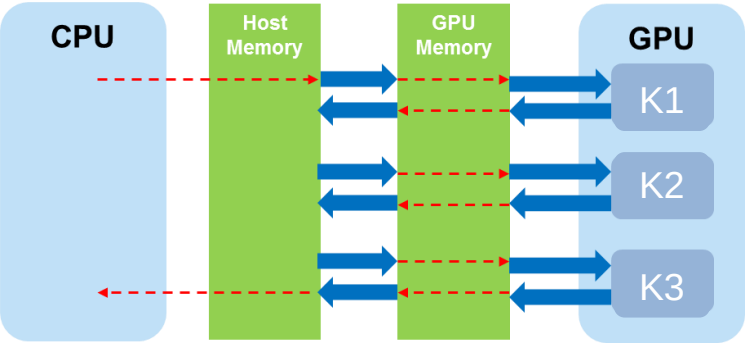
```
A = …
…
for i = 1:N
   … A(i)
end
…

imfilter
…
```

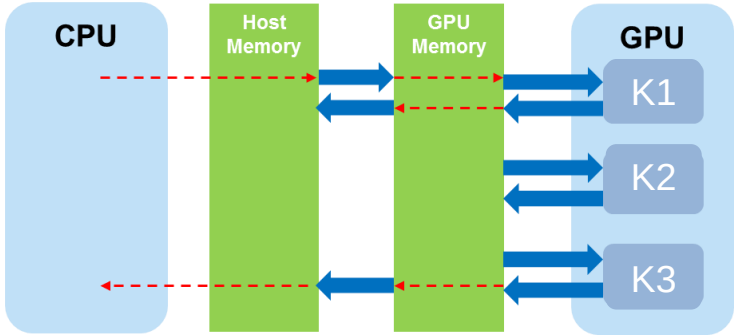
```
A = …
…
cudaMemcpyHtoD(gA, a);
kernel1<<<…>>>(gA)
cudaMemcpyDtoH(…)
…
cudaMemcpyHtoD(…)
imfilter_kernel(…)
cudaMemcpyDtoH(…)
…
```

Where is the ideal placement of memcpy?

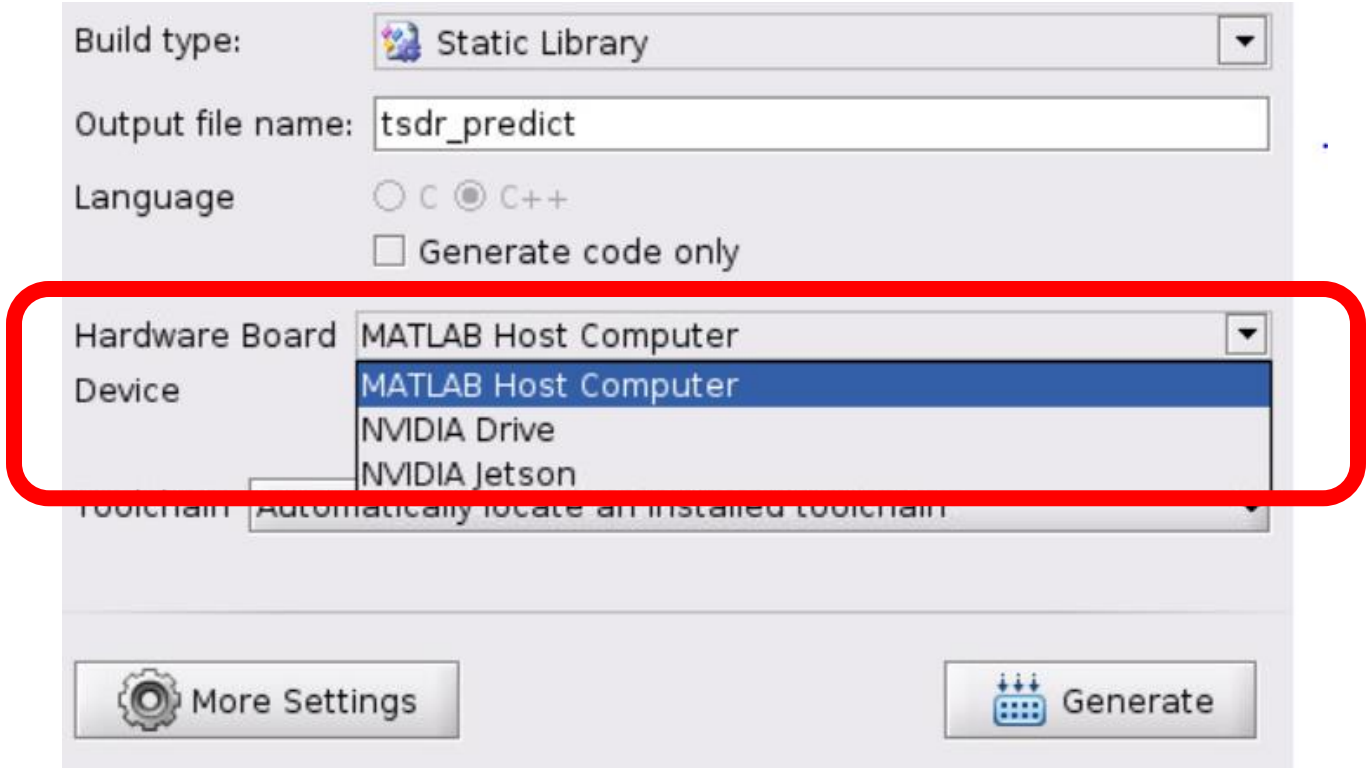
Naïve placement



Optimized placement



NVIDIA Hardware Support Package (HSP)

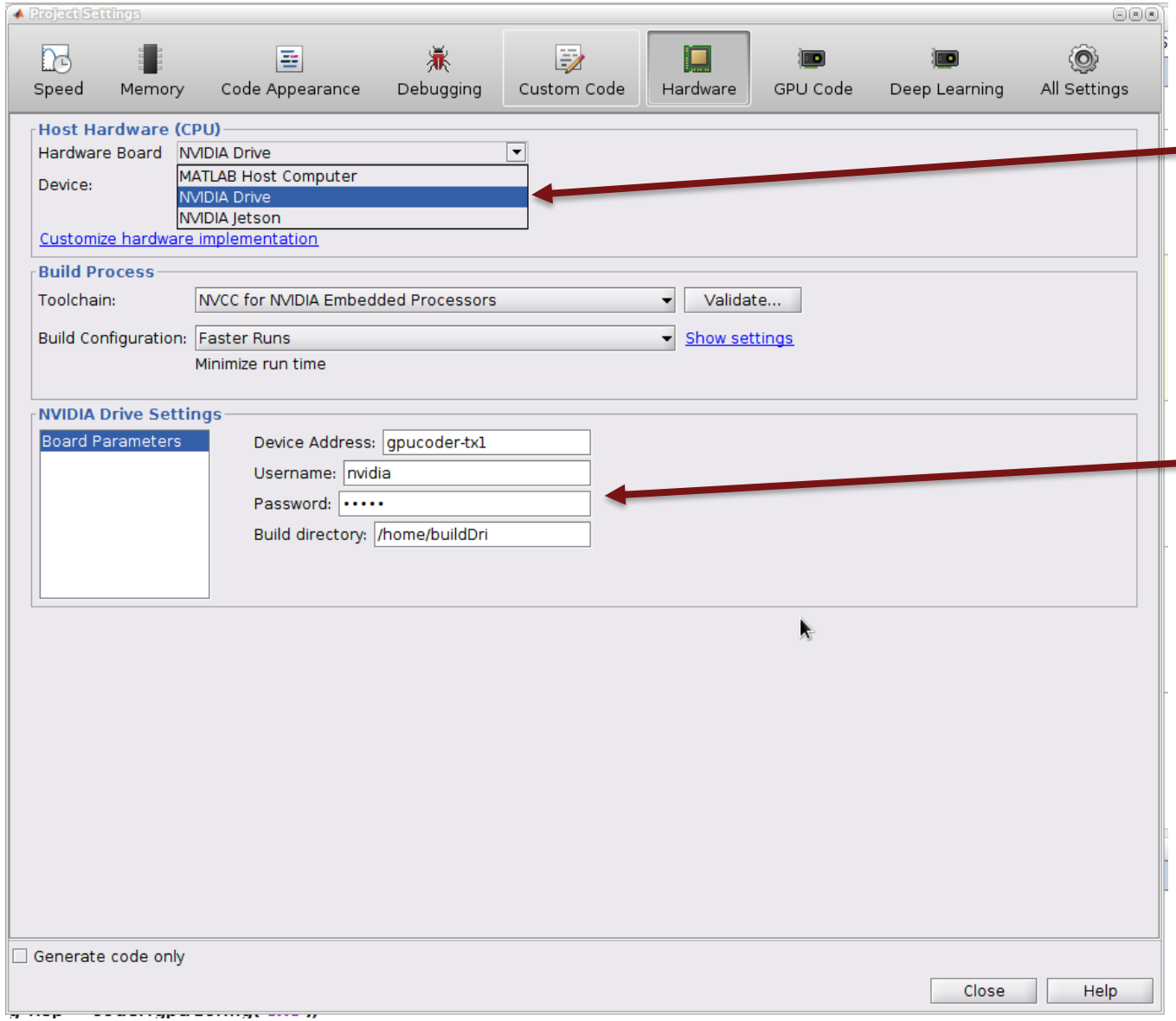


Simple out-of-box targeting to NVIDIA boards:

Jetson

Drive

Using NVIDIA HSP through GUI



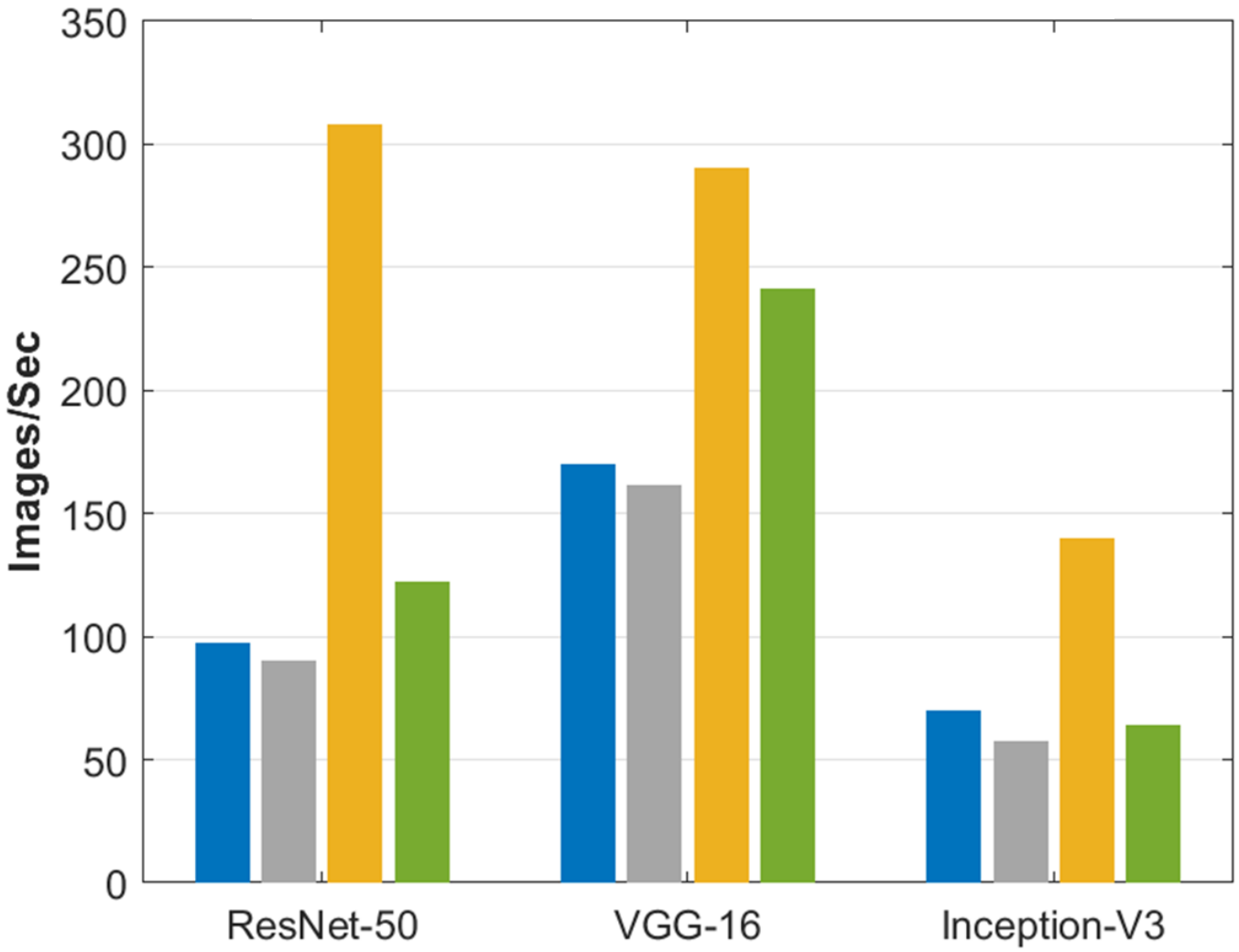
Choose your target board

Provide target info

Performance of Generated Code?

- CNN inference (ResNet-50, VGG-16, Inception V3) on Titan V GPU
- CNN inference (ResNet-50) on Jetson TX2
- CNN inference (ResNet-50 , VGG-16, Inception V3) on Intel Xeon CPU

Single Image Inference on Titan V using cuDNN



TensorFlow (1.13.0)

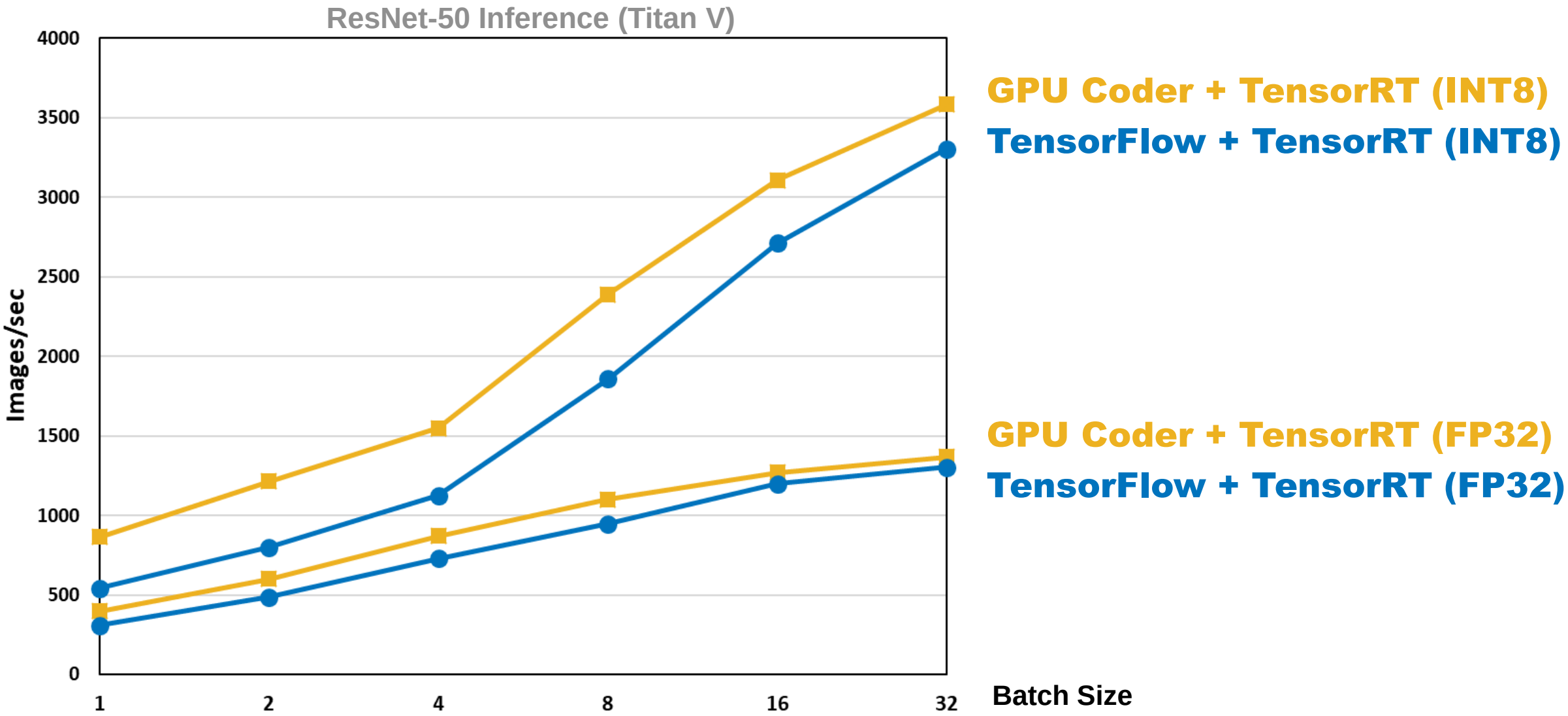
MXNet (1.4.0)

GPU Coder (R2019a)

PyTorch (1.0.0)

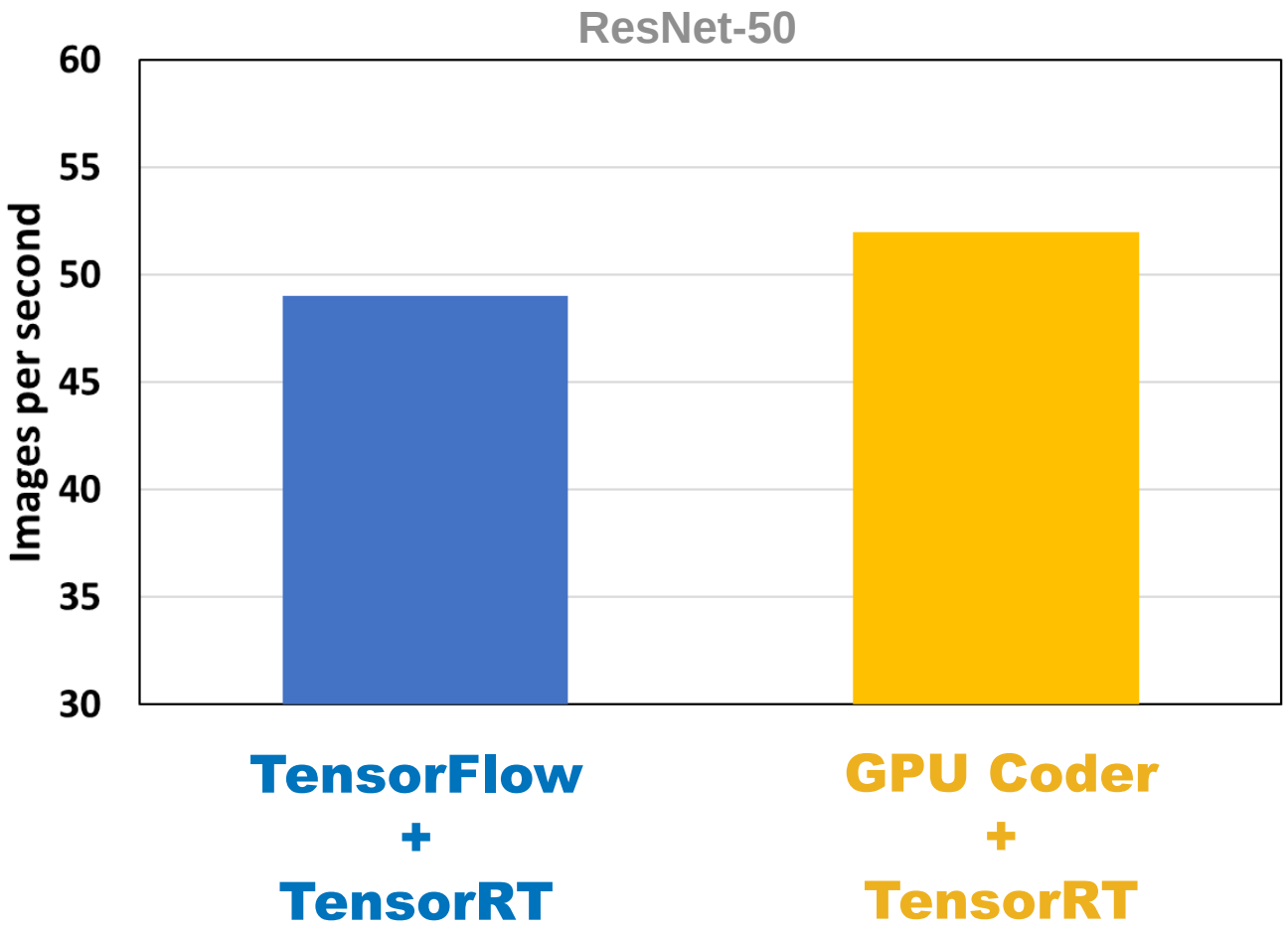
Intel® Xeon® CPU 3.6 GHz - NVIDIA libraries: CUDA10 - cuDNN 7 - Frameworks: TensorFlow 1.13.0, MXNet 1.4.0 PyTorch 1.0.0

Even Stronger Performance with INT8 using TensorRT



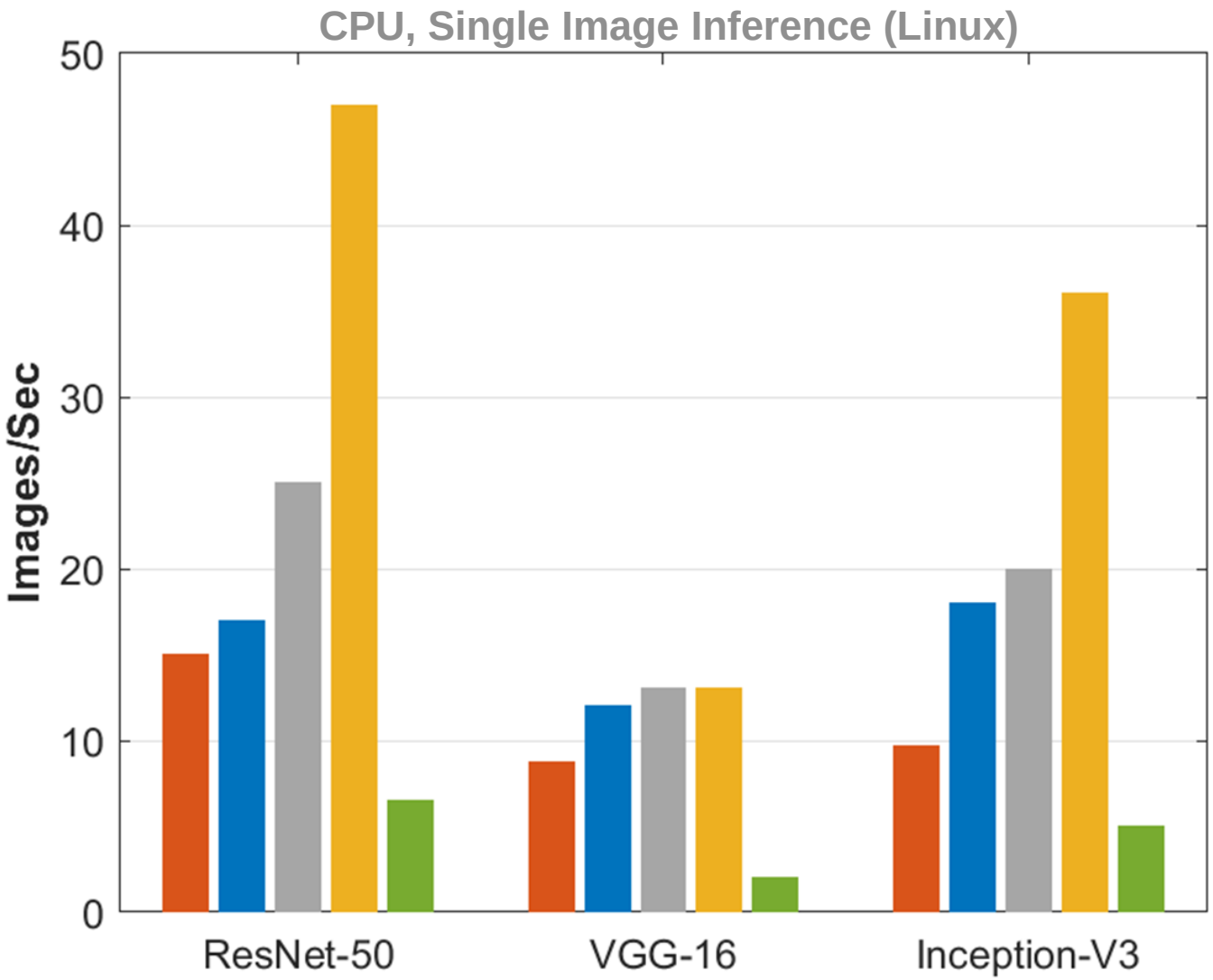
Intel® Xeon® CPU 3.6 GHz - NVIDIA libraries: CUDA10 - cuDNN 7.4.1 – TensorRT 5.0.2.6 - Frameworks: TensorFlow 1.13.0_rc0

Single Image Inference on Jetson TX2



NVIDIA libraries: CUDA9 - cuDNN 7 – TensorRT 3.0.4 - Frameworks: TensorFlow 1.12.0

CPU Performance



MATLAB

TensorFlow

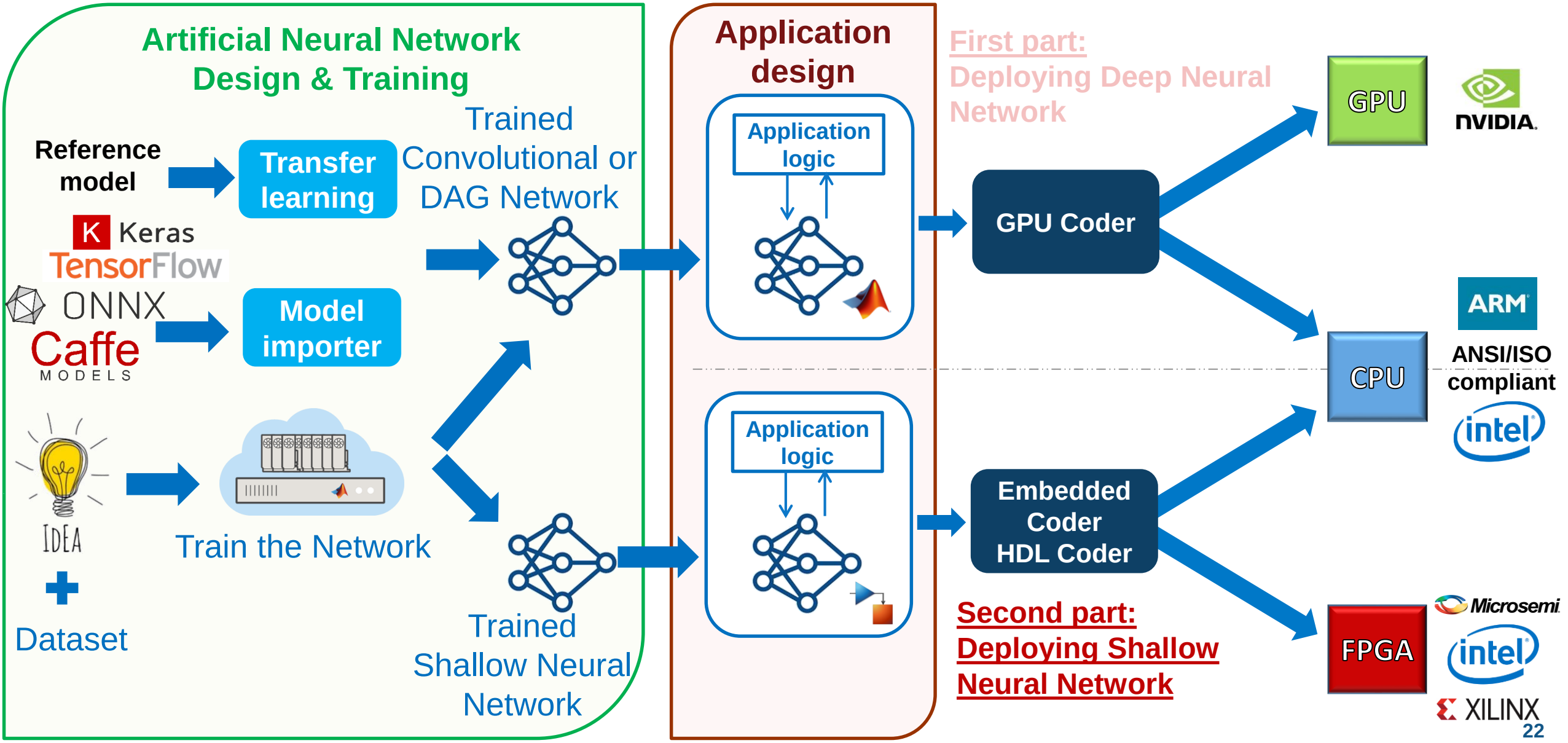
MXNet

MATLAB Coder

PyTorch

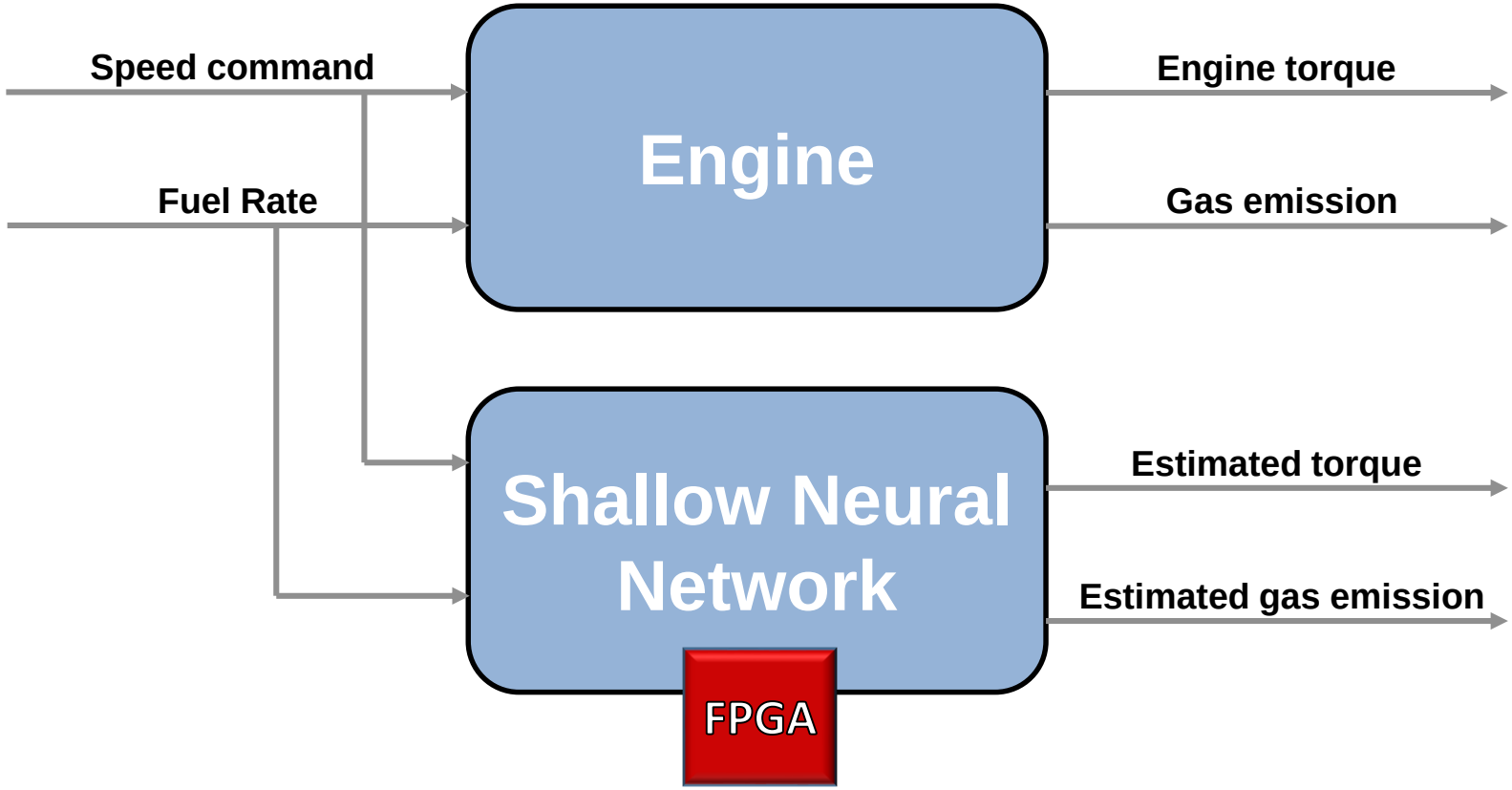
Intel® Xeon® CPU 3.6 GHz - Frameworks: TensorFlow 1.6.0, MXNet 1.2.1, PyTorch 0.3.1

MathWorks workflows: Neural Network to embedded targets

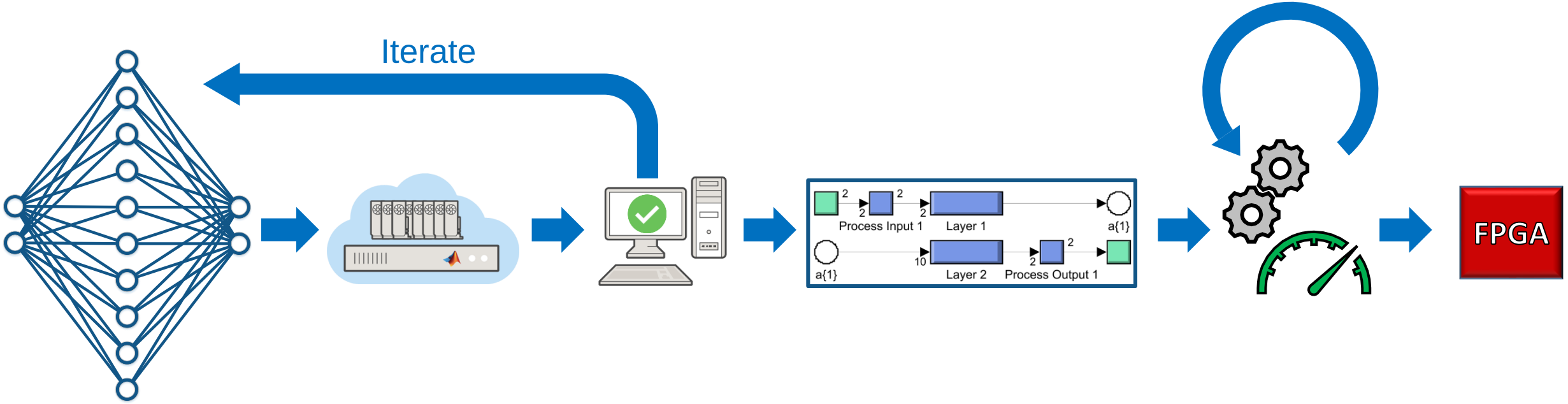


Demo: Shallow Network Deployment on Zynq Platform

Neural network as gas emission estimator (sensorless)



Shallow Network Deployment on FPGA/SoC Platform - Workflow



Create the Network structure

Train the Network

Test the Network

Export to Simulink

Fine-tune & optimize for the target

Generate code

MATLAB R2019a

HOME | PLOTS | APPS | PROJECT | PROJECT SHORTCUTS

New Shortcut | Organize Groups | Bringing AI to Real-Time Insights using SoC - GPU and CPU | FPGA in the Loop verification | Generate optimized HDL | NN on FPGA

MANAGE | GENERAL

C: ▸ myData ▸ NN2FPGA ▸ work ▸

Project - NN2FPGA

Views: Files, Dependency Analysis

All | Project (11) | Modified (2) | Layout: Tree

Name	Status	Git	Classification
data	✔	■	
models	✔	■	
utilities	✔	·	

Labels

Git

Ahead of /origin/master

Details

Current Folder | Workspace

Name	Git
hdl_prj	·
slprj	·
fxpdemo_neuralnet_..	
fxpdemo_neuralnet_..	
gm_fxpdemo_neural.	

Details

Select a file to view details

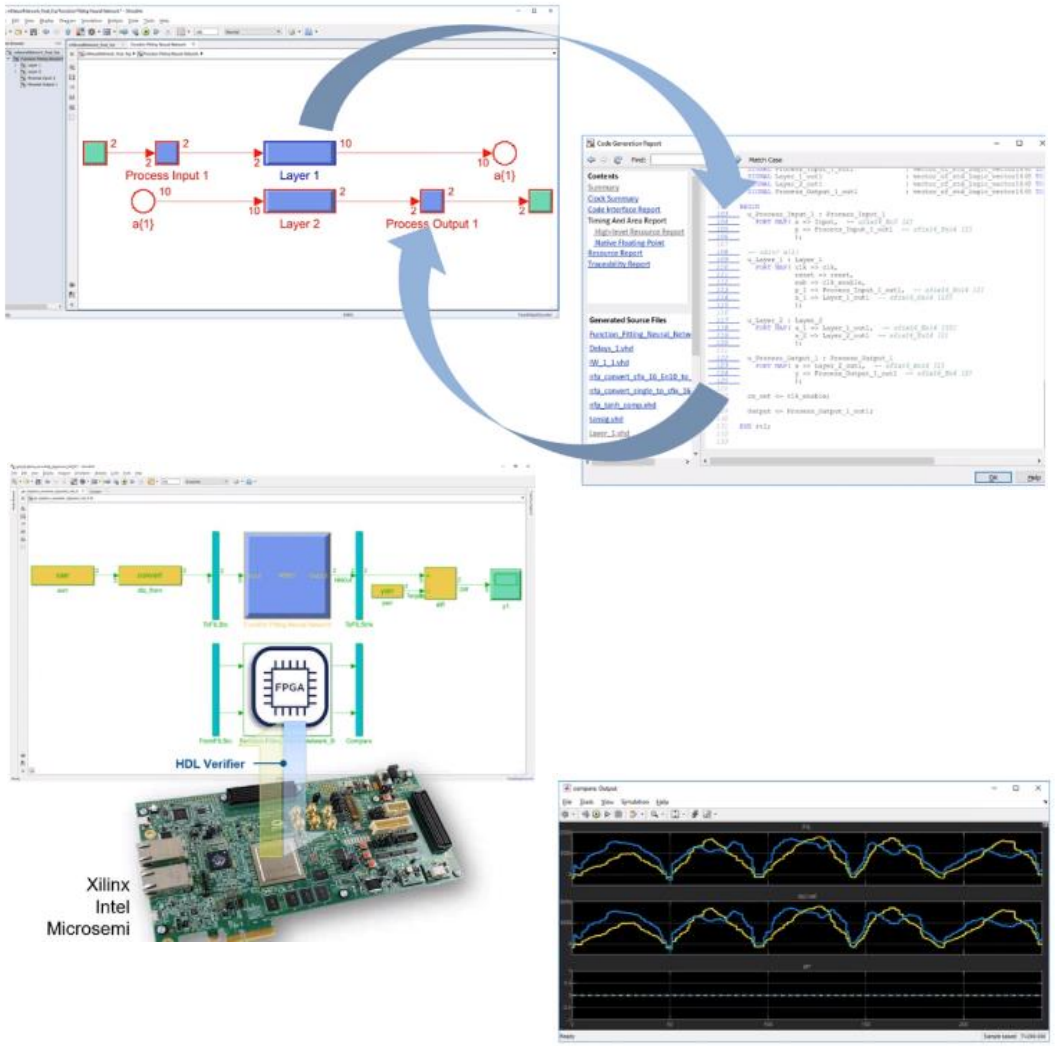
Command Window

```
IdleTimeout has been reached.
Parallel pool using the 'local' profile is shutting down.
>>
```

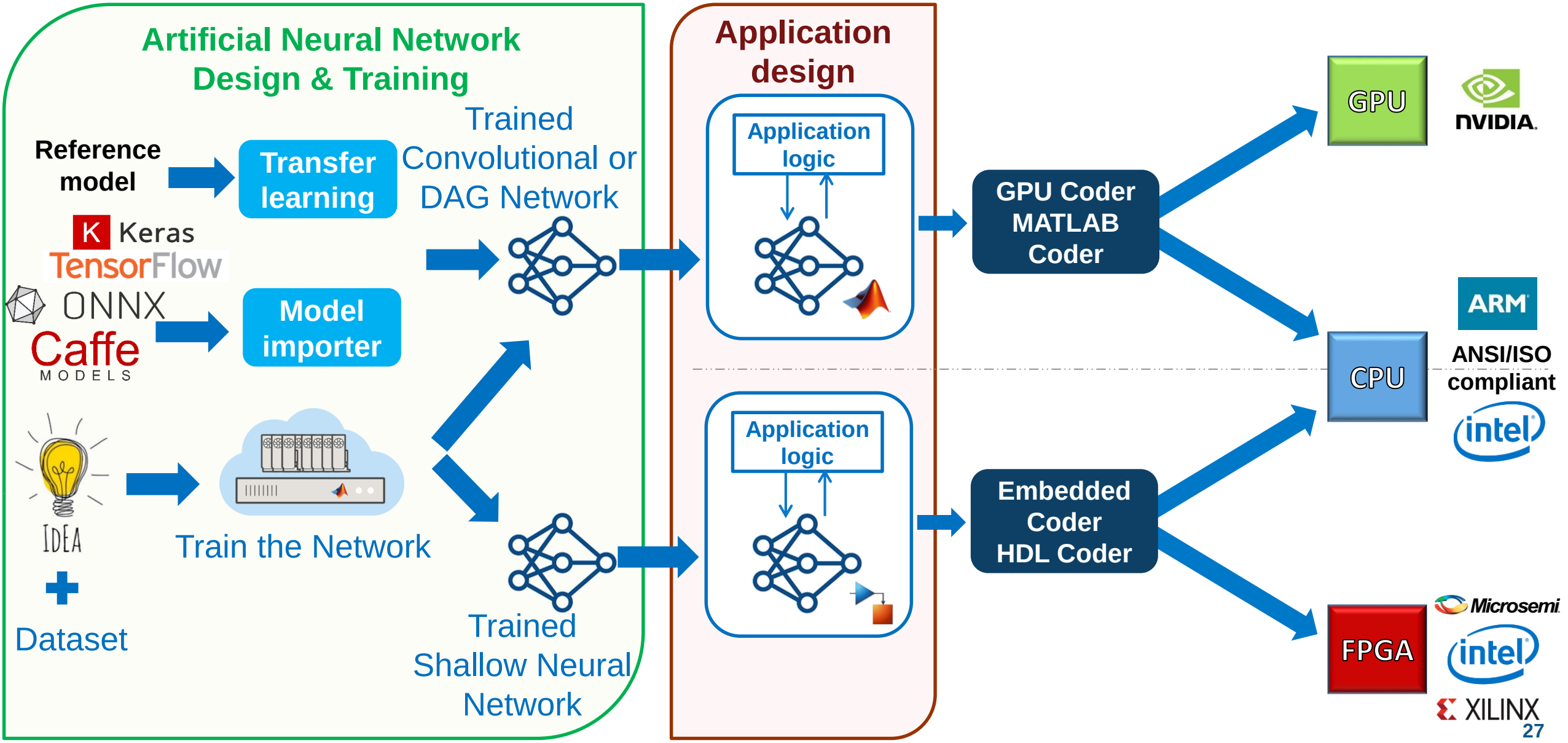
NLD 10:59

Shallow Network Deployment on FPGA/SoC Platform

- Automatic HDL code generation
- Freedom of choice to implement in floating-point or fixed-point
- Full bi-directional traceability
- Many area/speed optimization options
- Integrated verification
  - Reuse of existing MATLAB/Simulink testbenches
  - HDL code execution on FPGA
  - Automated generation of co-simulation infrastructure (Ethernet, JTAG, PCIe)



MathWorks workflows: Neural Network to embedded targets



Key Takeaways

- Comprehensive & integrated development environment from dataset to target
- Fast design space exploration and trade-off
- Target-independent functional reference for optimized implementation model
- Deploy « smart applications », not neural networks only
- More to come in a near future… Stay tuned!


MATLAB EXPO 2019
FRANCE
18 juin | Paris
Inscription : matlabexpo.fr

08:30	Enregistrement, petit-déjeuner et visite des stands		
	Sessions plénières		
09:15	**Mot de bienvenue et introduction** Jean-Manuel Quiroga, MathWorks		
09:30	**Keynote - Beyond the "I" in AI** Loren Shure, MathWorks		
10:00	**Keynote - Building Innovative Hardware in an Era of Artificial Intelligence** Igor Carron, CEO of LightOn		
10:30	**Les nouveautés MATLAB et Simulink** Cynthia Cudicini et Daniel Martins, MathWorks		
11:00	**Pause et visite des stands**		
	Calcul Scientifique et Data Analytics	**Modélisation et Simulation Système**	**Master Classes**
11:30	**Conception et simulation de systèmes de conduite autonome** Fulvio Martinelli, MathWorks	**Vers une conception intégrale à base de modèles ?** Sebastien Ratisseau, Zodiac Aero Electric	**Le Model-Based Design pour le développement de FPGA, ASIC et SOC** Fahd Morchid, MathWorks
12:00	**5G NR : Appréhender ce nouveau standard des télécommunications** Gerald Albertini, MathWorks	**Développement de systèmes de gestion de batterie (BMS)** Romain Romain Lachaux, MathWorks	
12:30	**Déjeuner, visite des stands, session "Women in Tech" et Tech'Talks**		
	Calcul Scientifique et Data Analytics	**Modélisation et Simulation Système**	**Master Classes**
14:00	**L'intelligence artificielle : les workflows de deep learning et de reinforcement learning** Valerie Leung, MathWorks	**Model-Based Design et certification : application au domaine médical** David Terrier, Fresenius Kabi	**La maintenance prédictive avec MATLAB et Simulink** Mathieu Cuenant et Kevin Roblet, MathWorks
14:35	**L'analyse d'images et le machine learning au service d'applications biologiques** Victor Racine, Quantacell	**Comment obtenir des crédits de certification avec Simulink ?** Daniel Martins, MathWorks	
15:10	**L'intelligence artificielle pour faciliter les activités d'inspection** Nicolas Castet, Airbus	**Table Ronde Certification** Brice Beltran, DGA Techniques Aéronautiques; James Bezamat, Safran Electronics & Defense; Luc Pelle, PMV Consulting Services; David Terrier, Fresenius Kabi et Patrick Munier, MathWorks	**Fusion de capteurs et pistage pour les systèmes autonomes** Gerald Albertini, MathWorks
15:40	**Visite des stands**		
16:10	**Déploiement des modèles temps-réel pour des applications de maintenance prédictive** Pierre Harouimi, MathWorks	**Ingénierie Système : La méthode PMM** Pascal Paper, Airbus	**Du modèle au matériel : prototypez rapidement vos systèmes en temps-réel** Olivier Berard, MathWorks
16:45	**Eradiquer les bugs : du mythe à la pratique** Olivier Bouissou, MathWorks	**Ingénierie système : des exigences à l'architecture et la simulation** Laurent Royer, MathWorks	
17:15	**Fin de la conférence**		


Merci !