Deploy MATLAB and Simulink to NVIDIA GPUs using GPU Coder

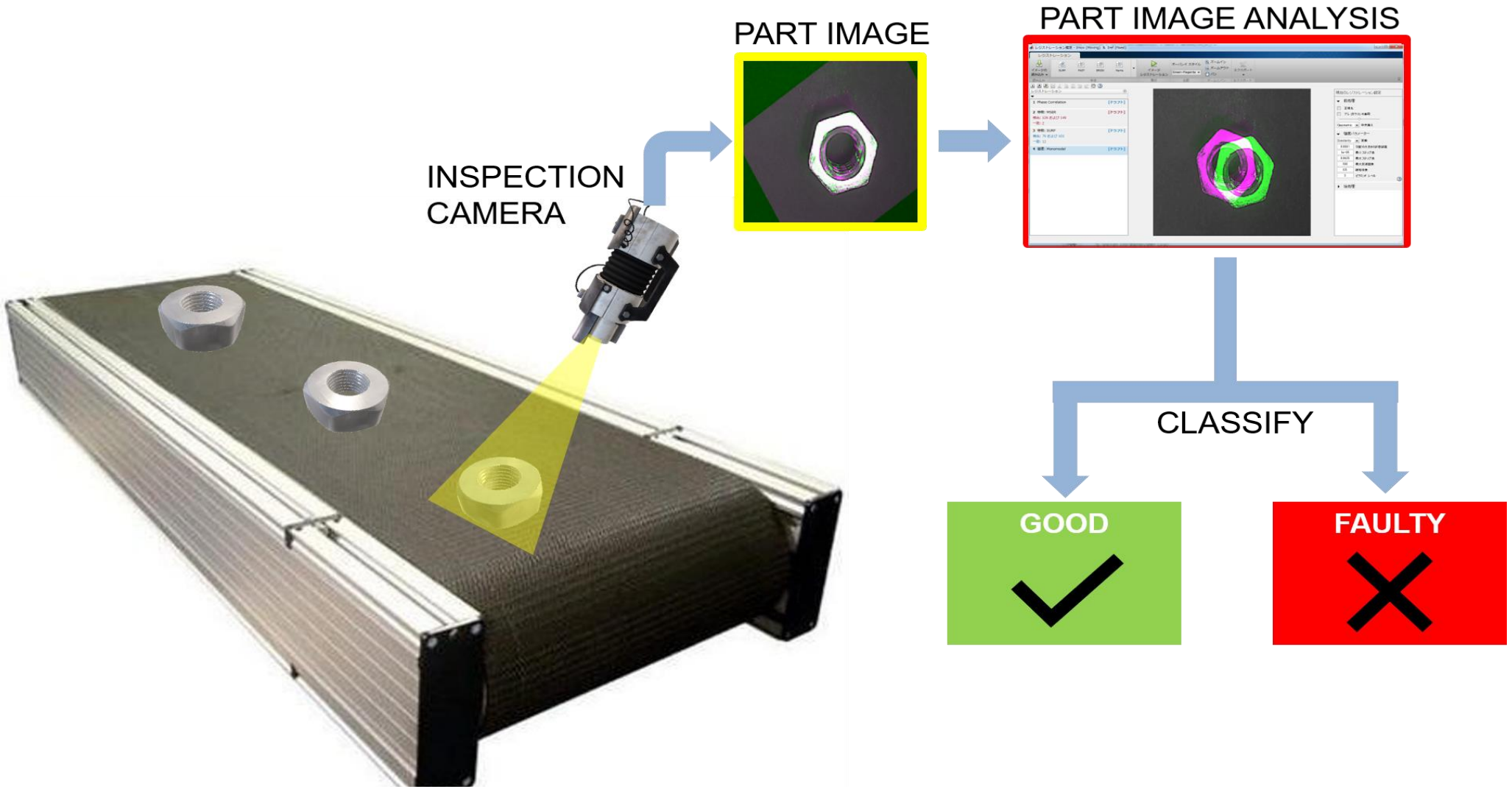
Bill Chou

Code Generation
MathWorks
bchou@mathworks.com

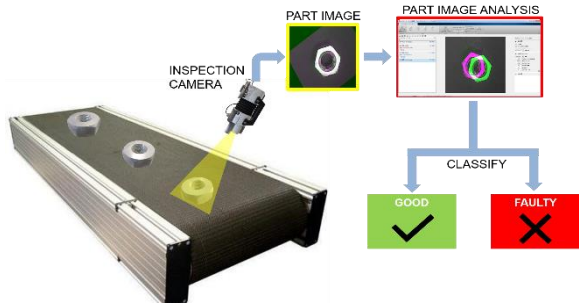
Agenda

- Why translate MATLAB to CUDA?
- What is GPU Coder?
- Workflow for using GPU Coder
- Deep Learning workflow
- Wrap-up

Automated Optical Inspection



Automated Optical Inspection



tb_myNDNet_Jetson_Intro.mlx * | tb_myNDNet_Jetson.mlx | myNDNet_Preprocess.m | myNDNet_Postprocess.m | targetFunction.m | main_nutsDet.cpp

Grab one frame from input video stream, run inference, identify defect in image

```
load('NDNet.mat');
img = imread('testImg.png');

wi = uint32(320);
he = uint32(240);
ch = uint32(3);
sz = [he wi ch];

imgOcv = mat2ocv(img);
Weights = convnet.Layers(67).Weights;

tic;
out = targetFunction(imgOcv,Weights,false);
toc

tic;
out = targetFunction_mex(imgOcv,Weights,false);
toc

out = ocv2mat(out,sz);

figure;
s1 = subplot(1,2,1);
s2 = subplot(1,2,2);
imshow(img, 'Parent',s1);
title(s1, 'Input');

imshow(out, 'Parent',s2);
```

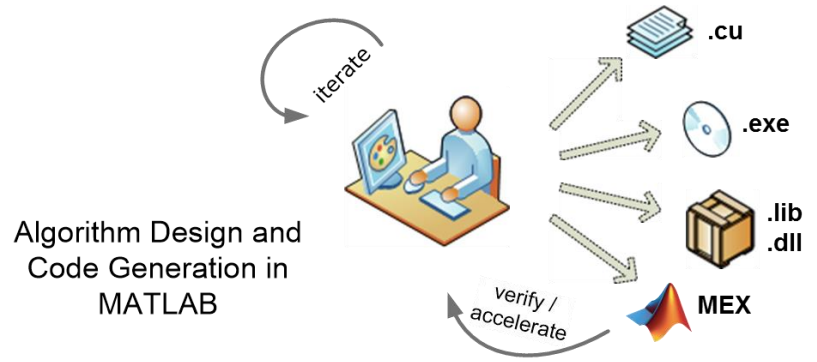
Why Use CUDA code generation ?

Pains: Hand code

- **Cannot code in CUDA**
- Time consuming
- Manual Coding Errors
- Multiple implementations
- Expensive

Solution: GPU Coder

- Automatically convert to CUDA
- Get to optimized CUDA faster
- Eliminate manual coding errors
- Maintain Single “Truth”
- Stay within MATLAB/Simulink at a higher level

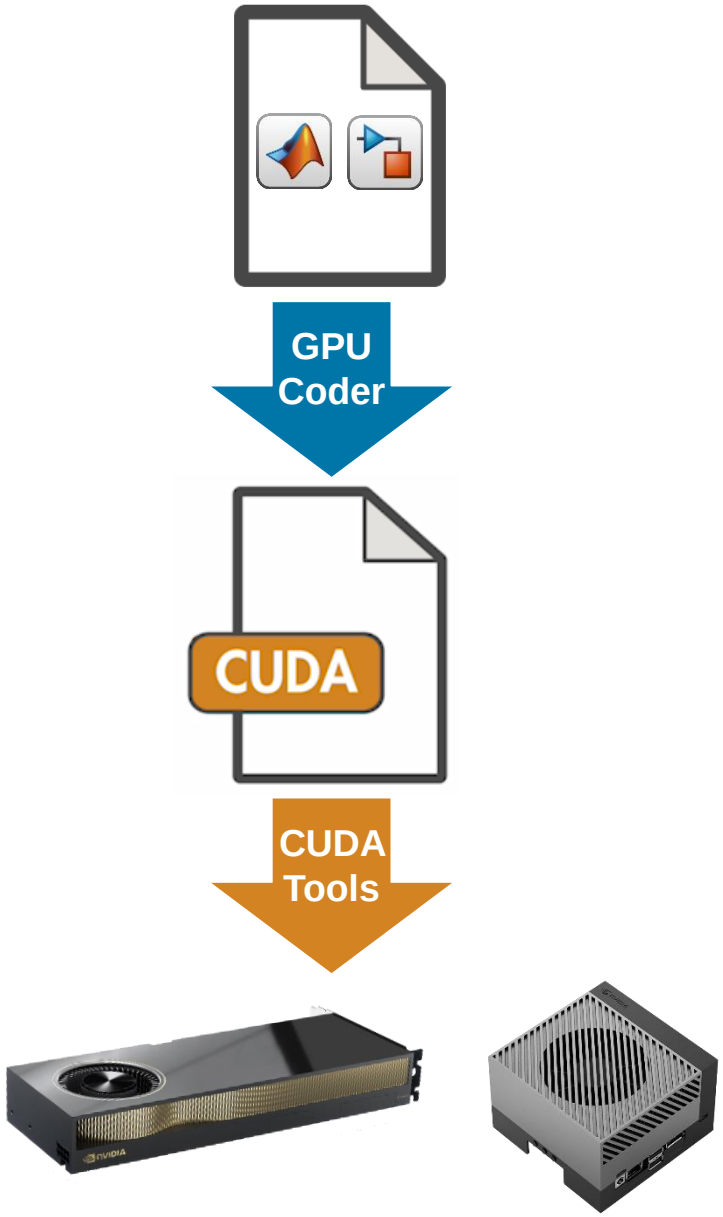


Agenda

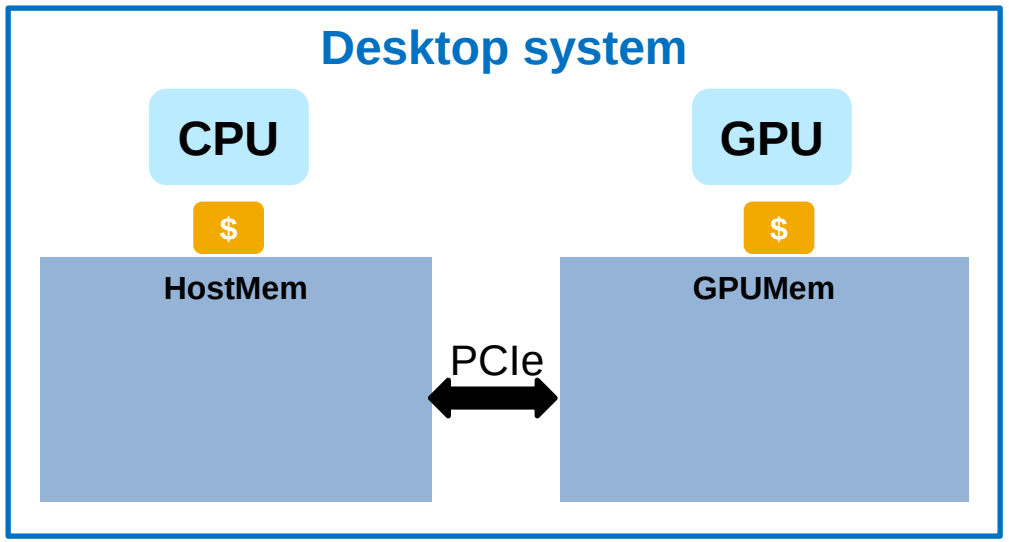
- Why translate MATLAB to CUDA?
- What is GPU Coder?
- Workflow for using GPU Coder
- Deep Learning workflow
- Wrap-up

What is GPU Coder?

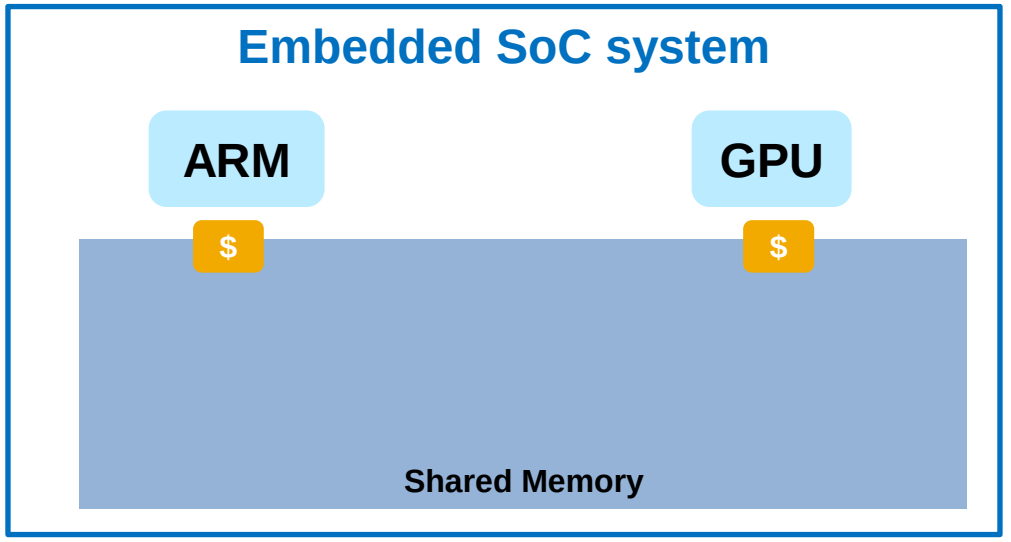
- Tool that automatically converts MATLAB to CUDA
- Generated CUDA is portable across all modern NVIDIA GPUs
- Automates deployment to cloud, desktop, and embedded GPUs
  - Including desktop RTX and embedded Jetson AGX Orin
- Can call optimization libraries like TensorRT
- Profile the generated CUDA to find opportunities to optimize performance

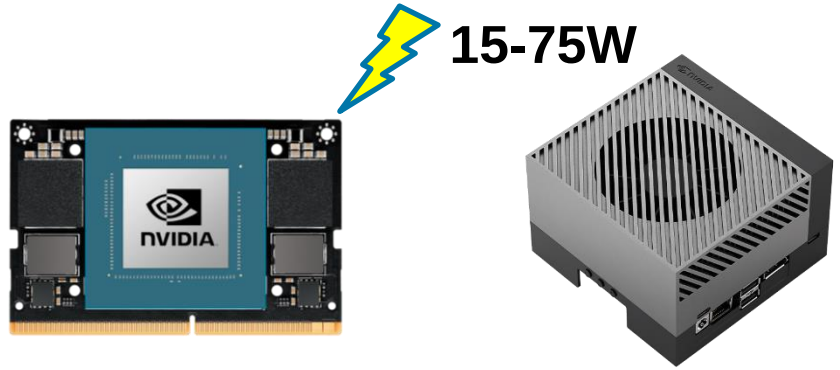


Types of GPUs



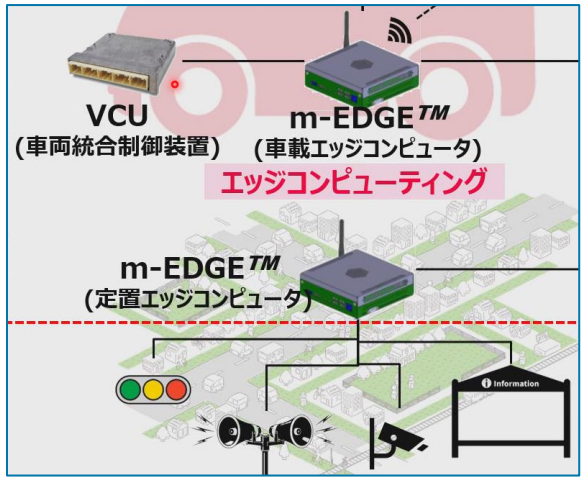
Desktop GPUs
(and Cloud GPUs)



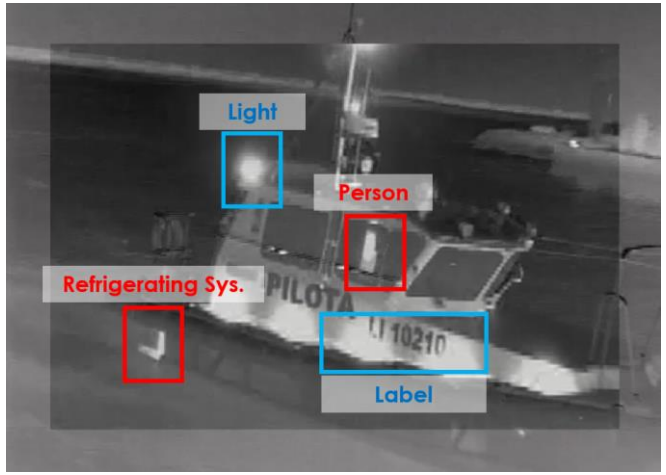


Embedded GPUs

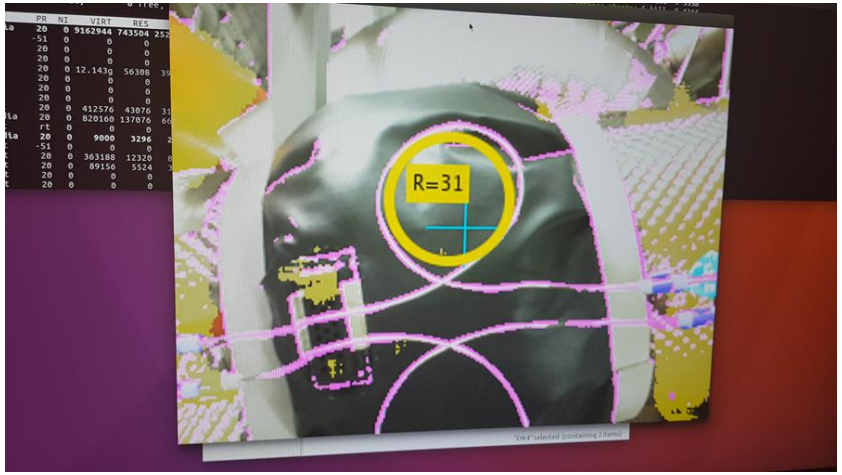
Wide Range of Applications



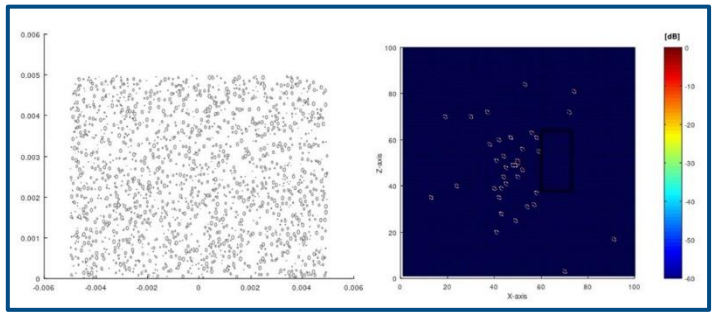
DENSO Ten
Vehicle detection and traffic flow analysis



Drass
Maritime object tracking on NVIDIA GPU

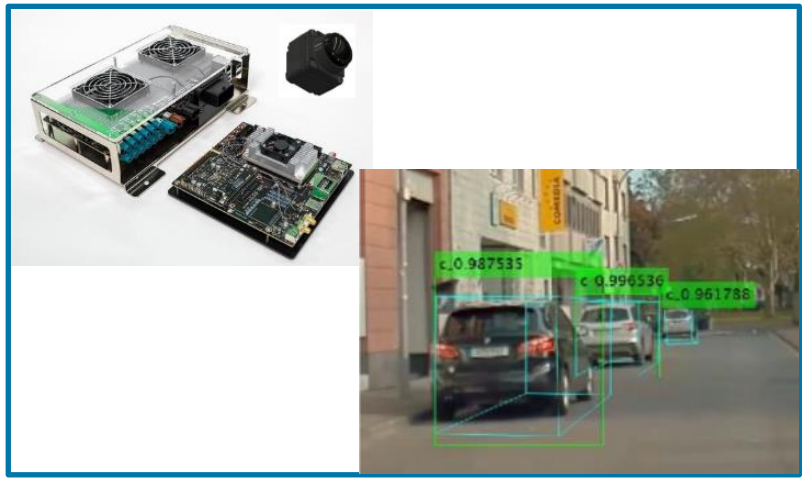


Airbus
Aircraft inspection on Jetson



M&R Technology
Medical Ray-Tracing Algorithms

JINGCE Electronics
Display measurement instrument on Jetson



Continental
Computer Vision Stack for AD

Drass Develops Deep Learning System for Real-Time Object Detection in Maritime Environments

Challenge

Help ship operators monitor sea environments and detect objects, obstacles, and other ships

Solution

Create an object-detection deep learning model that can be deployed on ships and run in real time

Results

- Data labeling automated
- Development time reduced
- Flexible and reproducible framework established

First day of object detection tests with optronic system prototype.

"From data annotation to choosing, training, testing, and fine-tuning our deep learning model, MATLAB had all the tools we needed—and GPU Coder enabled us to rapidly deploy to our NVIDIA GPUs even though we had limited GPU experience."

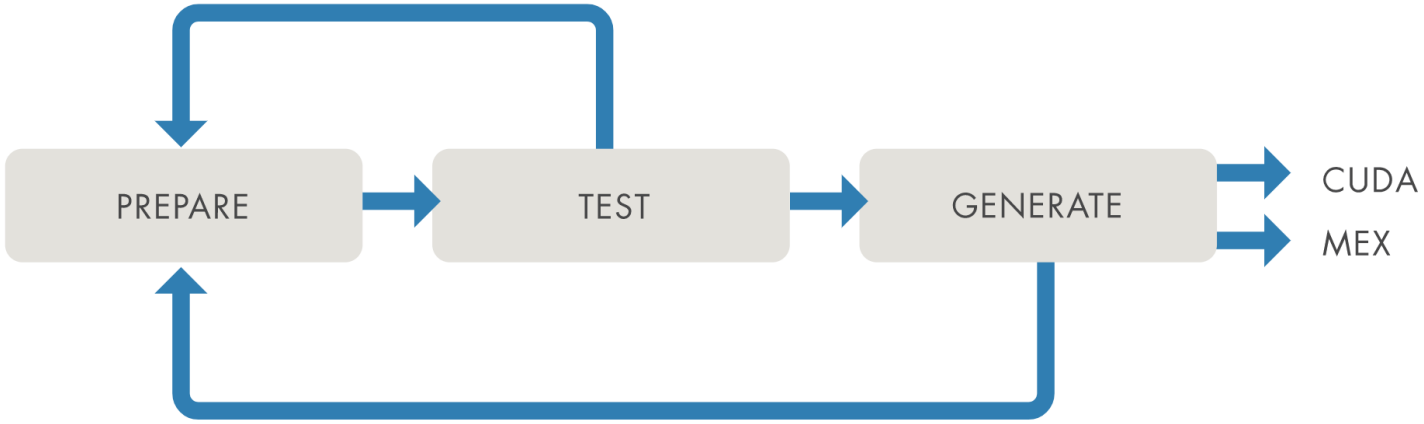
- Valerio Imbriolo, Drass Group

Link to user story

Agenda

- Why translate MATLAB to CUDA?
- What is GPU Coder?
- Workflow for using GPU Coder
- Deep Learning workflow
- Wrap-up

Working with GPU Coder: Three-Step Workflow



Prepare your MATLAB algorithm for code generation

- Make implementation choices
- Use supported language features

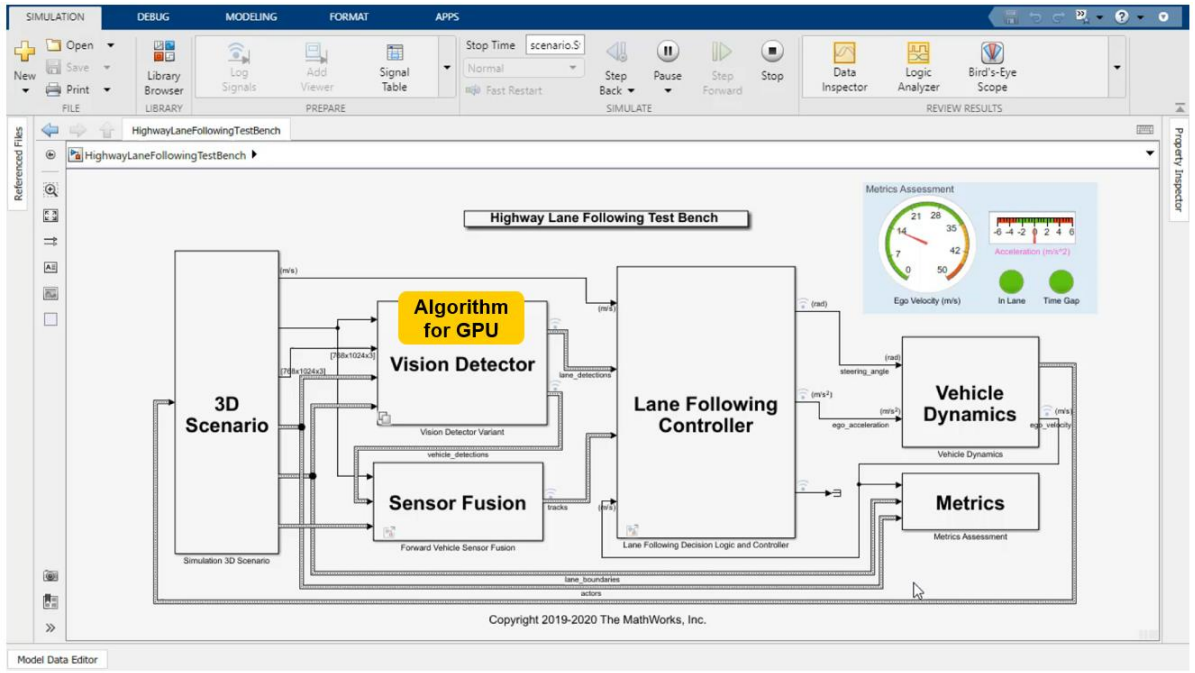
Test if your MATLAB code is ready for code generation

- Validate that MATLAB program generates code
- Accelerate execution of user-written algorithm

Generate source code or MEX for final use

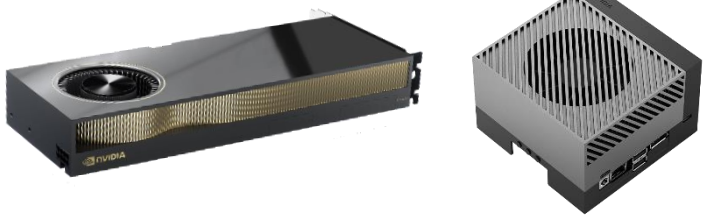
- Iterate your MATLAB code and use profiling tools to help optimize further
- Implement as source, executable, or library

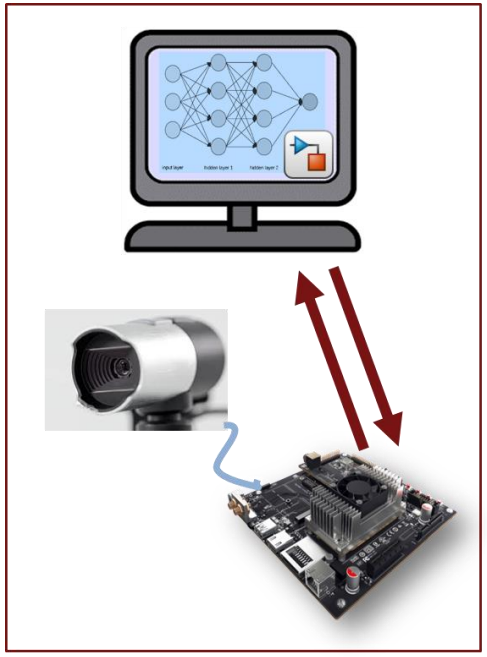
GPU Coder for Simulink Workflows



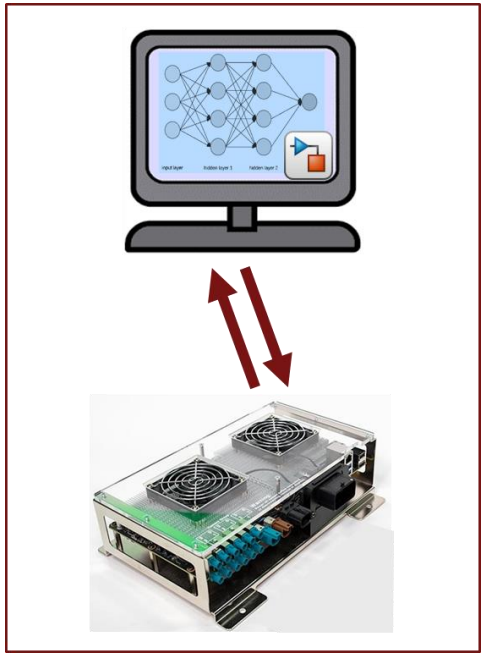
Deploy Standalone Application

GPU Coder



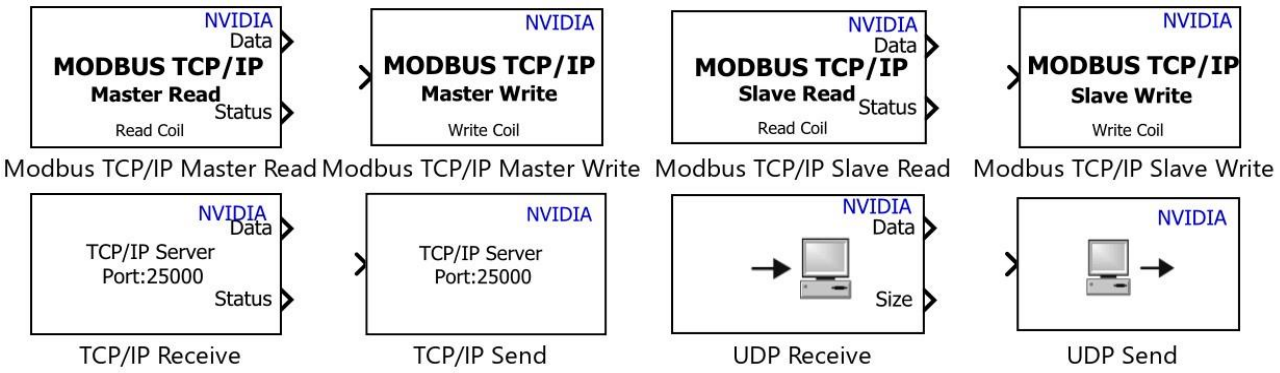


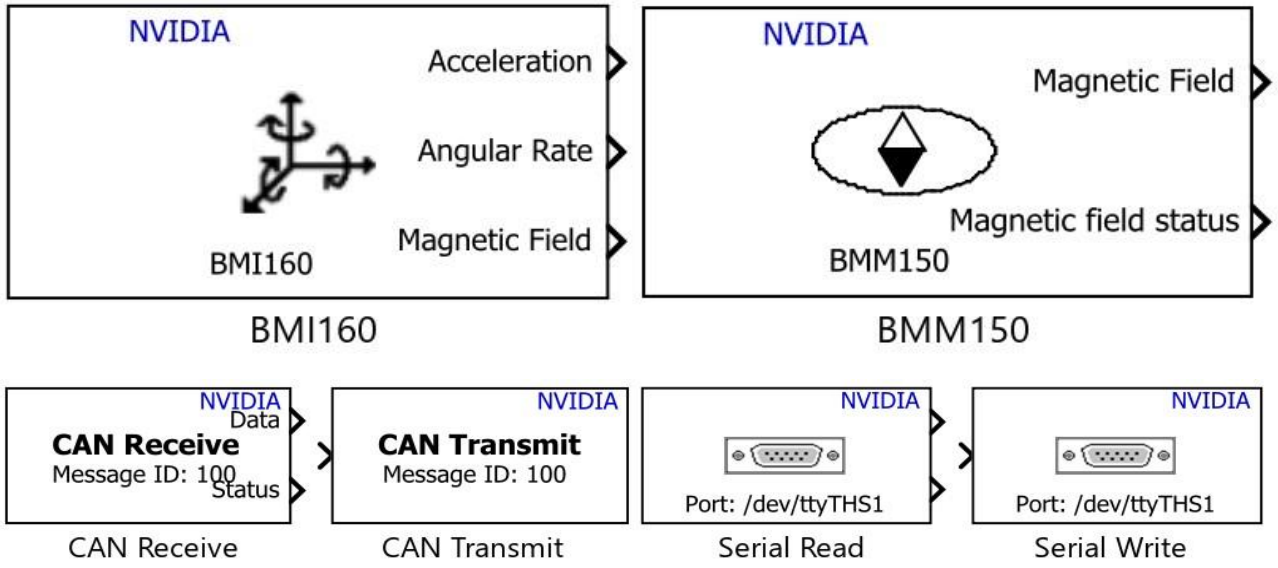
Access Peripheral from Simulink

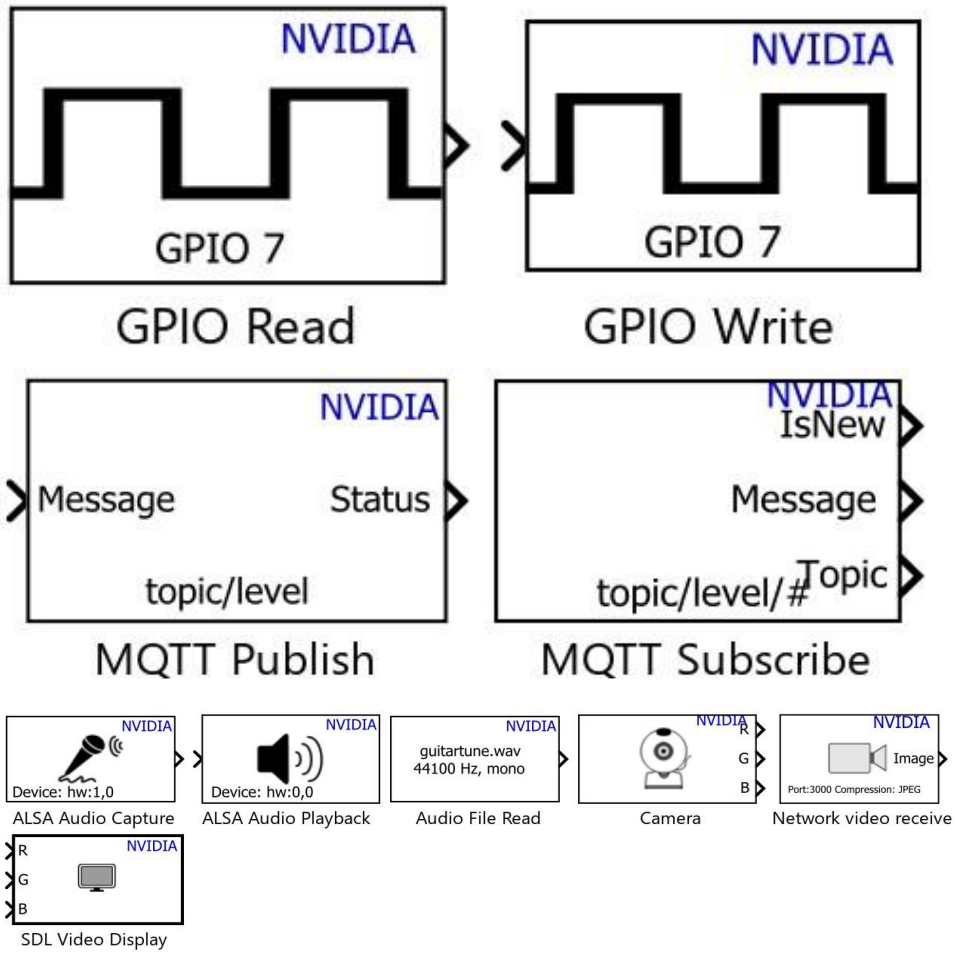


Software-in-Loop, Processor-in-Loop, External Mode

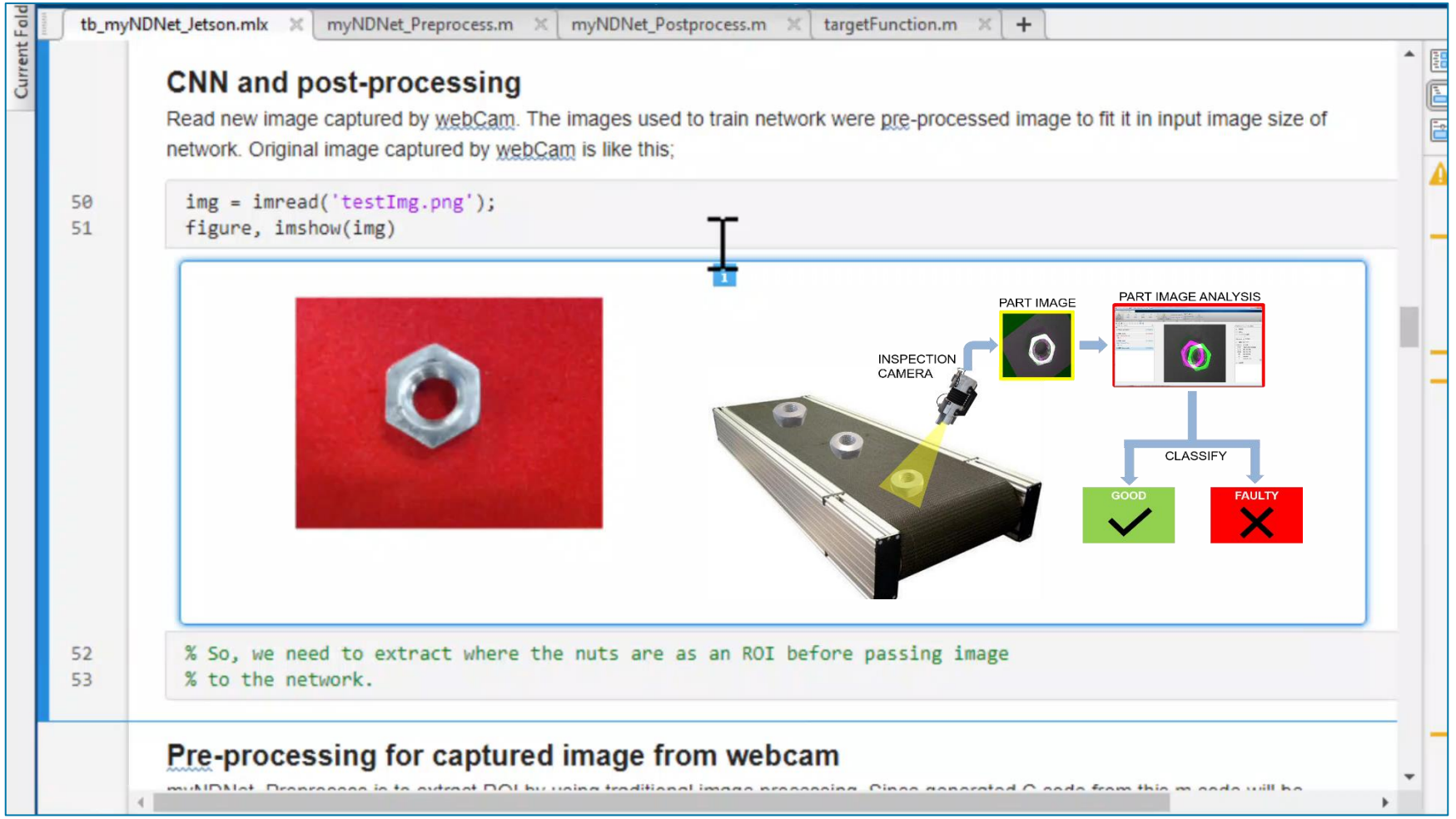
NVIDIA Peripheral Support – block library







Using GPU Coder with Automated Optical Inspection Example



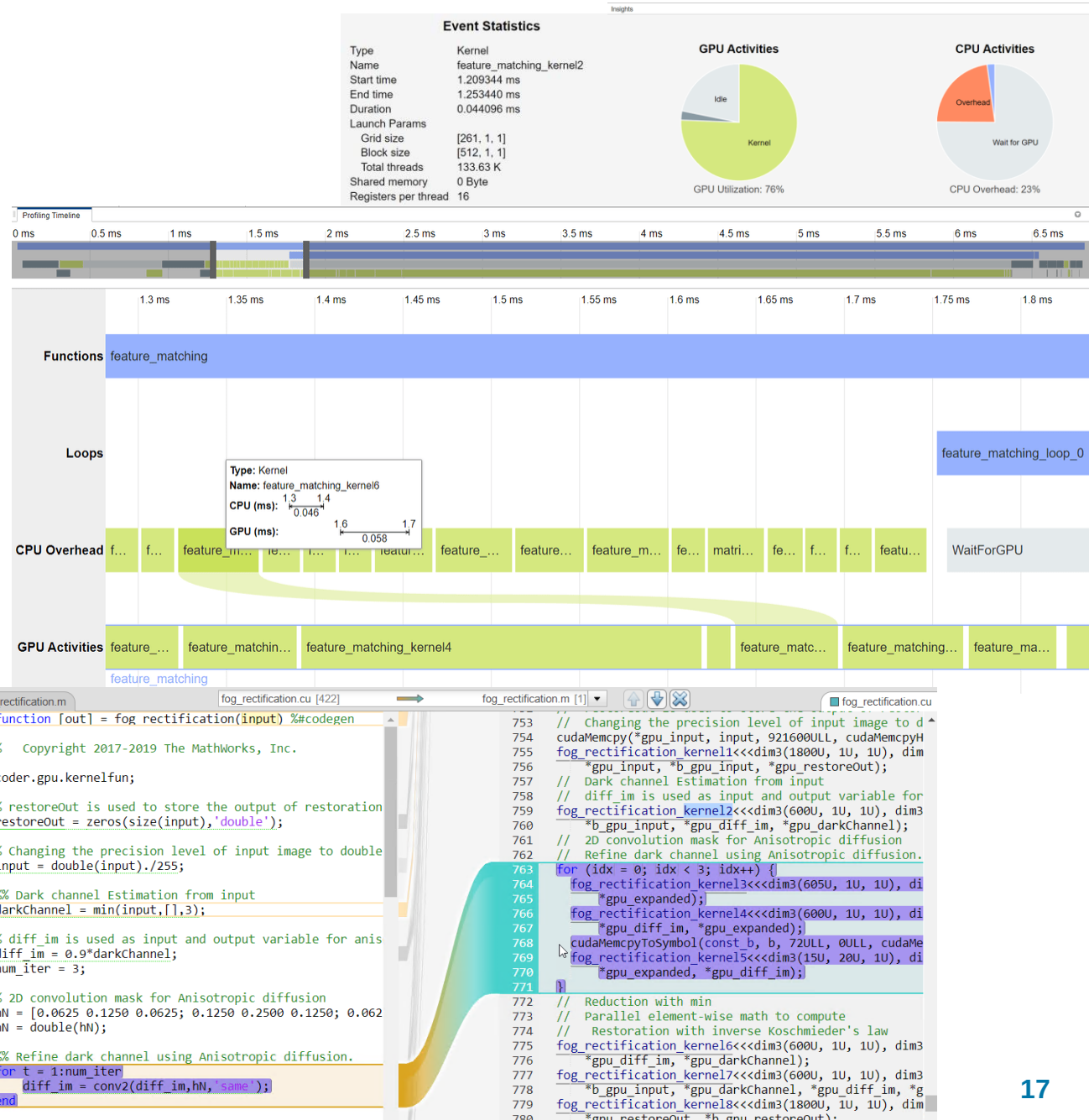
Example in MATLAB

- Generate MEX file and accelerate in MATLAB
- Generate DLL/LIB/EXE and deploy
  - Desktop
  - Cloud
  - Embedded
- Generate source code and integrate into existing project
- Connected with Jetson board using MATLAB Coder Support Package for NVIDIA Jetson and NVIDIA DRIVE Platforms
- File transfer to/from Jetson
- Peripheral support
  - GPIO, audio input/output, USB/CSI camera, webcam, LiDAR sensors, …
- Processor-in-loop (PIL) testing

How to Optimize Further?

Profiling and Bidirectional Traceability **Tools**

- Use the GPU Coder Performance Analyzer to profile the generated CUDA code
  - Identify bottlenecks & opportunities to optimize performance
- Use bidirectional traceability to map to/from CUDA code back to MATLAB code
  - Helps you to understand how CUDA kernels are created from your MATLAB algorithms



Using the GPU Coder Performance Analyzer

PATooldemoUsingSURF.mlx | SurfDetect1.m | PATooldemoUsingSURF.m

Performance Analyzer

Surf Features are an important detection mechanism in computer vision problems to help facilitate registration and object detection algorithms. Let's take a look. In this peppers example, each feature is marked in the image and contains a coordinate represented by the position on the image, a scale represented by the circle diameters, and orientation represented by the vector in the circles, and by the sign of Laplacian. This implementation designed for deployment takes a little over 1 second to run on a modern laptop. We will generate CUDA code for this algorithm to demonstrate how the Performance Analyzer Tool can help you modify your MATLAB code to achieve better performance.

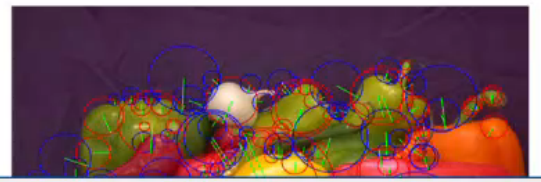
```
1  imageFile = 'peppers.png';
2  img = imread(imageFile);
3  tic;
4  intPoints = SurfDetect(img);
5  toc
```

```
Elapsed time is 1.240863 seconds.
```

```
6
7  % display a Surf Point structure
8  intPoints(3)
```

```
ans = struct with fields:
              x: 83.8932
              y: 203.8722
          scale: 2.0504
    orientation: 2.8450
      laplacian: 1
```

```
9  DrawIpoints(imageFile,intPoints);
```



CUDA Kernel Optimizations Heuristics

- Memory Allocation
  - Supports various GPU memory spaces (local, global, shared, constant)

- Data Transfer Minimization
  - Analyzes data dependency between the CPU and GPU partitions to determine minimum set of locations where data must be copied between CPU and GPU using `cudaMemcpy`

  - Data shared between the CPU and GPU is allocated on GPU memory using `cudaMalloc` or `cudaMallocManaged` (for unified memory, GPU Coder determines minimum number of `cudaDeviceSync` calls)

Optimizations for Generated CUDA Code

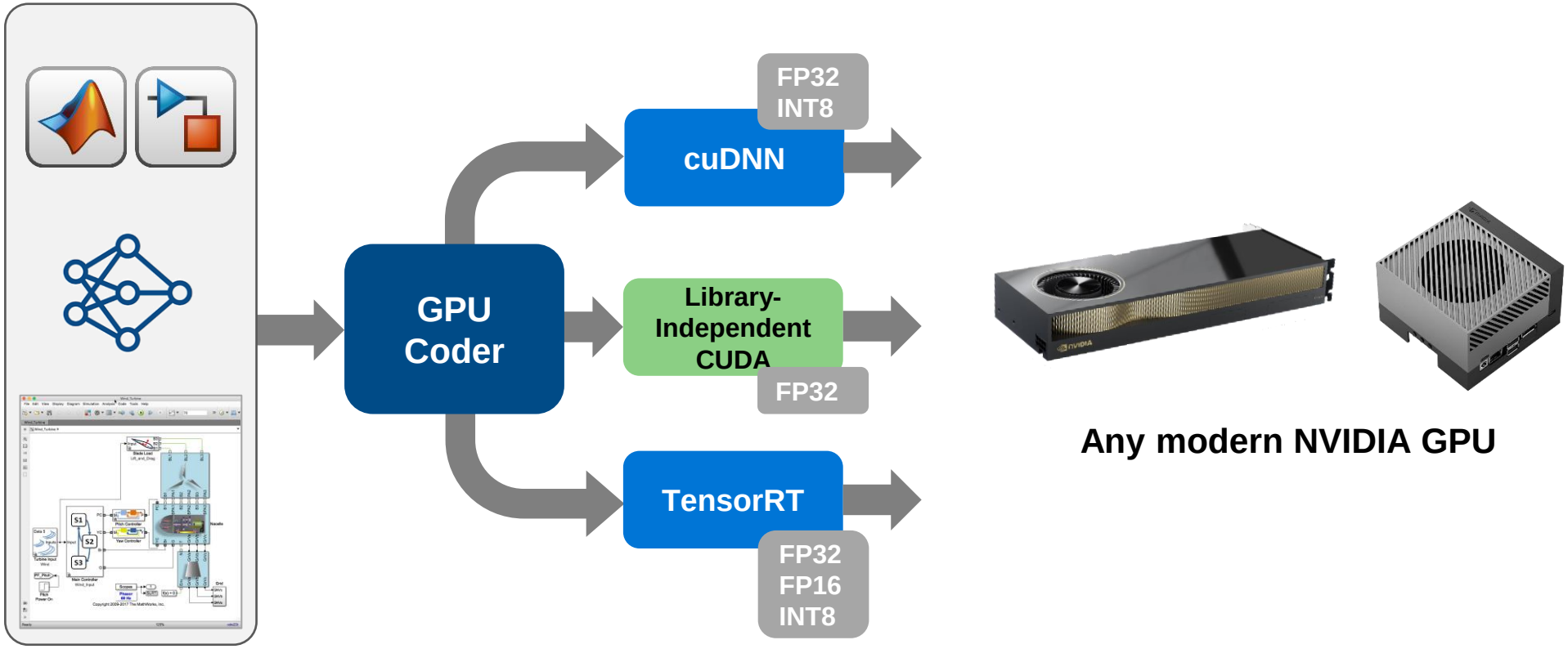
- Accelerated library support
  - cuFFT, cuBLAS, cuSolver, Thrust, cuDNN, and TensorRT

- Design patterns (apply manually):
  - Stencil processing can be used for operations such as convolution, median filtering, and finite element methods
  - Matrix-matrix processing can be used for operations such as sum of absolute differences (SAD) and sum of squared differences (SSD)
  - Reductions can be used for operations such as vector summation

Agenda

- Why translate MATLAB to CUDA?
- What is GPU Coder?
- Workflow for using GPU Coder
- Deep Learning workflow
- Wrap-up

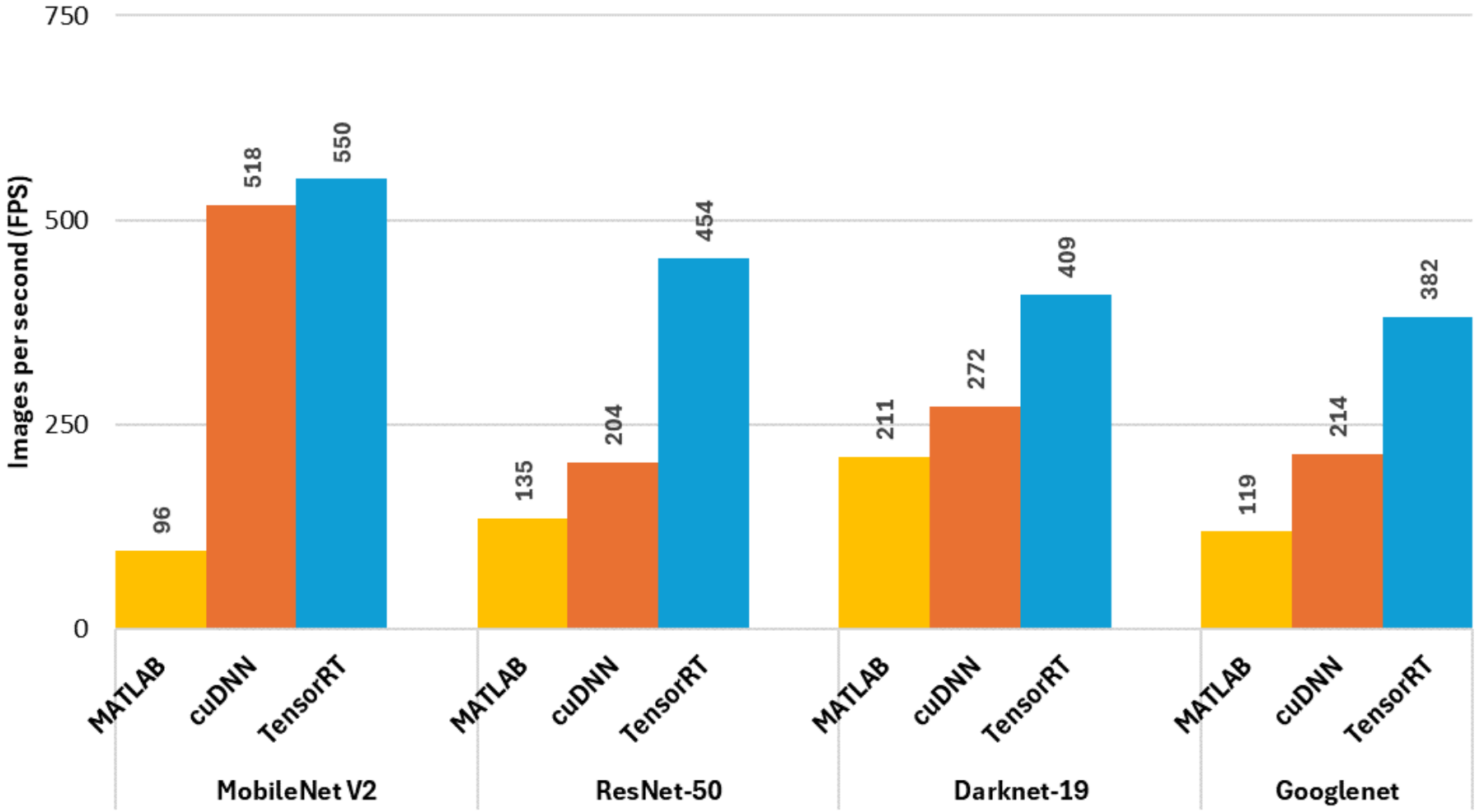
Deep Learning Code Generation



22

Performance of Generated Code from GPU Coder

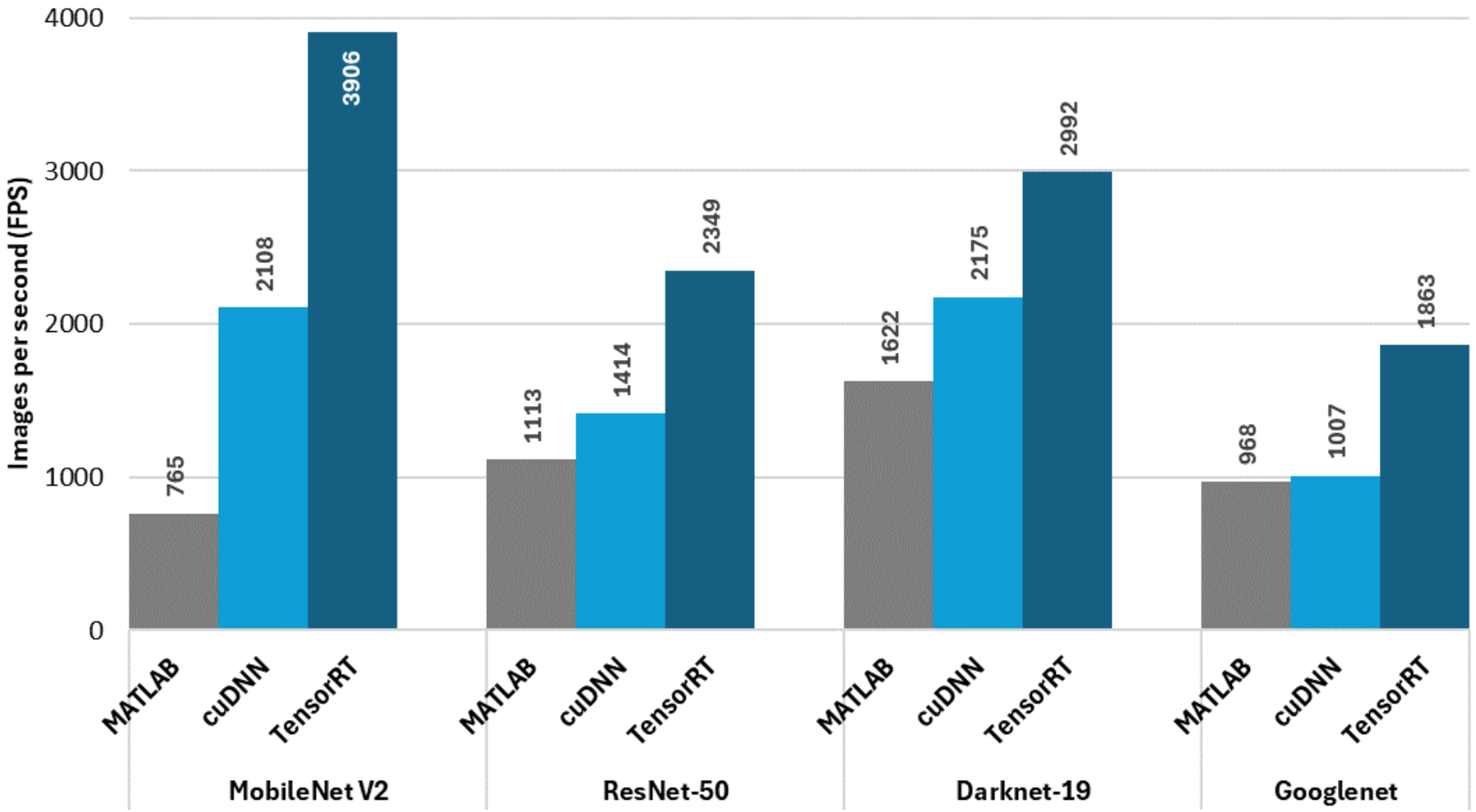
Batch Size 1



AMD® EPYC 7262 CPU 3.2 GHz | NVIDIA® RTX A5000 GPU | NVIDIA libraries: CUDA12.2 - cuDNN 8.9 – TensorRT 8.6.1

Performance of Generated Code from GPU Coder

Batch Size 8



AMD® EPYC 7262 CPU 3.2 GHz | NVIDIA® RTX A5000 GPU | NVIDIA libraries: CUDA12.2 - cuDNN 8.9 – TensorRT 8.6.1

Agenda

- Why translate MATLAB to CUDA?
- What is GPU Coder?
- Workflow for using GPU Coder
- Deep Learning workflow
- Wrap-up

Key Takeaways

- GPU Coder generates CUDA code from MATLAB & Simulink
  - Deploy signal processing, radar processing, image processing, and computer vision algorithms
  - Deploy complete AI applications (with Deep Learning Toolbox)
- Integrate handwritten CUDA code for use within MATLAB/Simulink and generated code
- Accelerate MATLAB & Simulink simulations using MEX-files

Shipping Examples

MathWorks® Products Solutions Academia Support Community Events

Help Center

Search Help Center | Help Center

CONTENTS

« Documentation Home
« Examples
« Code Generation

Category

Fixed-Point Designer
GPU Coder
- Get Started with GPU Coder 2
- Kernel Creation 13
- Performance 5
- Deep Learning with GPU Coder 28
- Deployment 21

HDL Coder
HDL Verifier
IEC Certification Kit
MATLAB Coder
Simulink Code Inspector
Simulink Coder
Simulink PLC Coder

Type

- All 66
- MATLAB 51
- Simulink 15

New Example

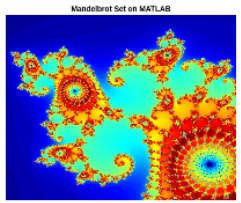
Added in R2024a

Documentation | Examples | Functions | Apps | Videos | Answers | Trial Software | Product Updates

GPU Coder — Examples

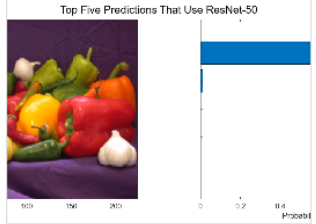
https://www.mathworks.com/help/gpucoder/examples.html

Get Started with GPU Coder



GPU Code Generation: The Mandelbrot Set

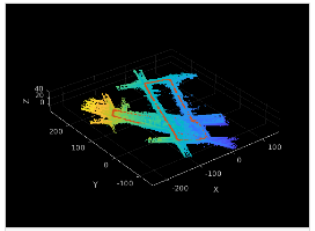
Generate CUDA® code from a simple MATLAB® function by using GPU Coder™. A Mandelbrot set implementation by using standard



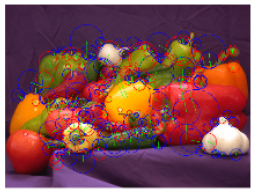
Code Generation for Deep Learning Networks

Get started with CUDA code generation for image classification networks such as `ResNet`.

Kernel Creation

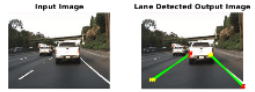


Build a Map from Lidar Data

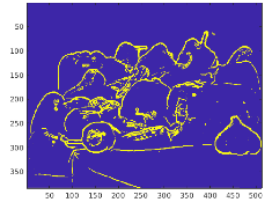


Feature Extraction Using

Feature Matching



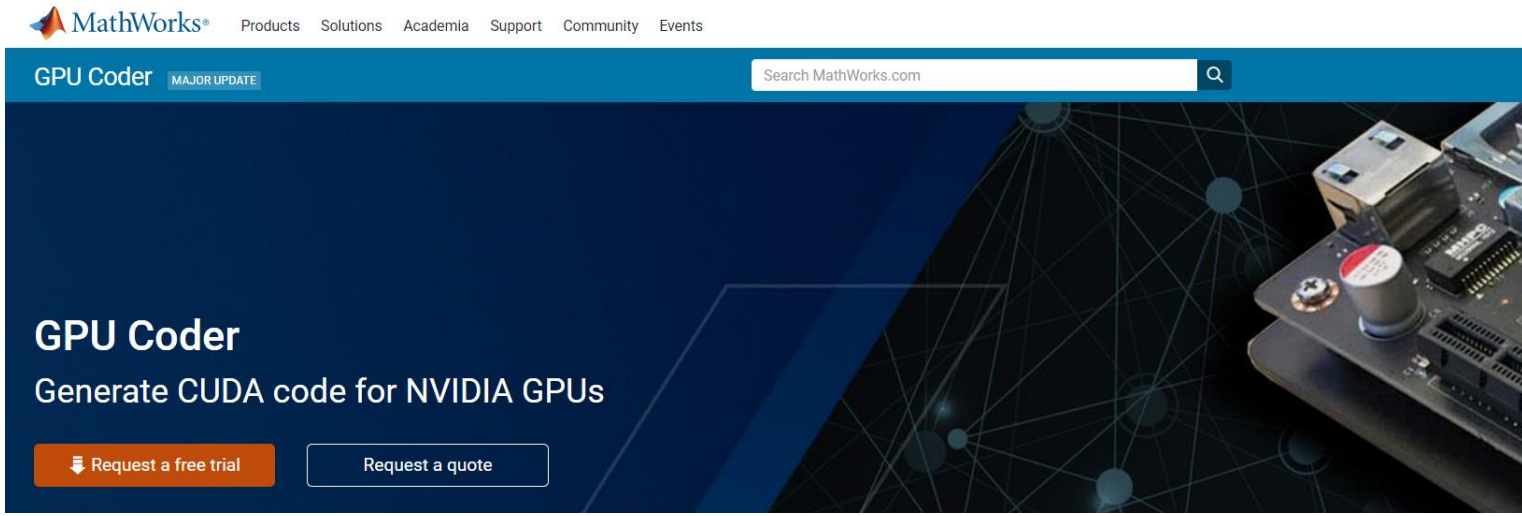
Lane Detection on the GPU



Edge Detection with Sobel

Next Steps

- To learn more, including playing with a trial license, visit:
  - www.mathworks.com/products/gpu-coder
- Read white paper on generating CUDA code from MATLAB
- Attend 4-hour hands-on workshop
  - Arrange with your account manager



Q&A