

強化AI專題實作力：MATLAB GPU Coder 助力模型高效部署至 Python平台

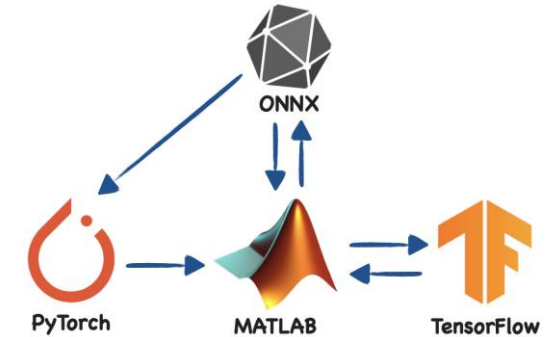
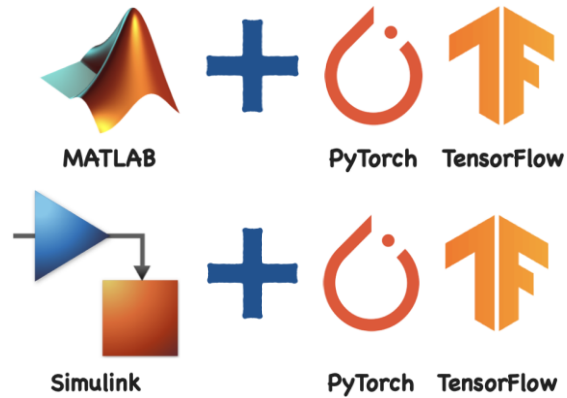
Judy Yang
Application Engineer
TeraSoft, Inc.

Common use case: combine MATLAB with python for AI

Co-execute

Exchange Models

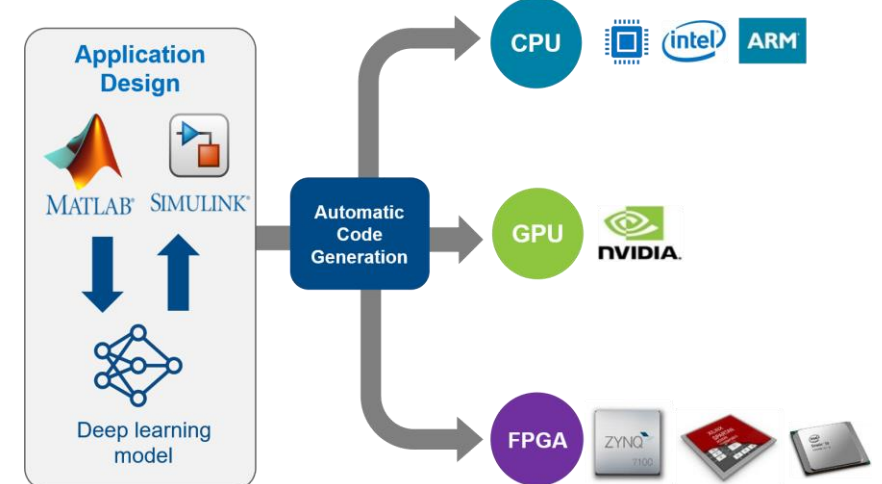
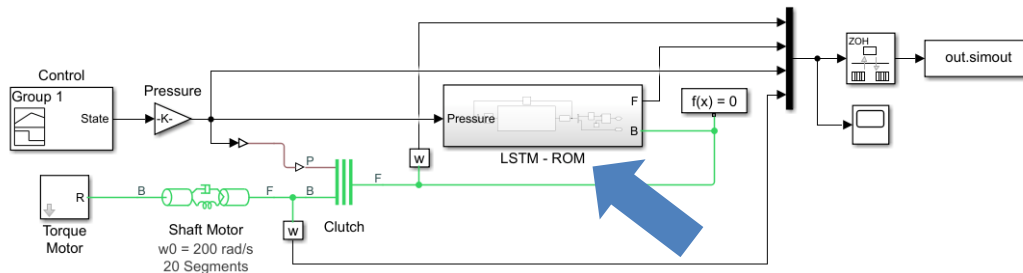
Techniques:



Engineering + AI

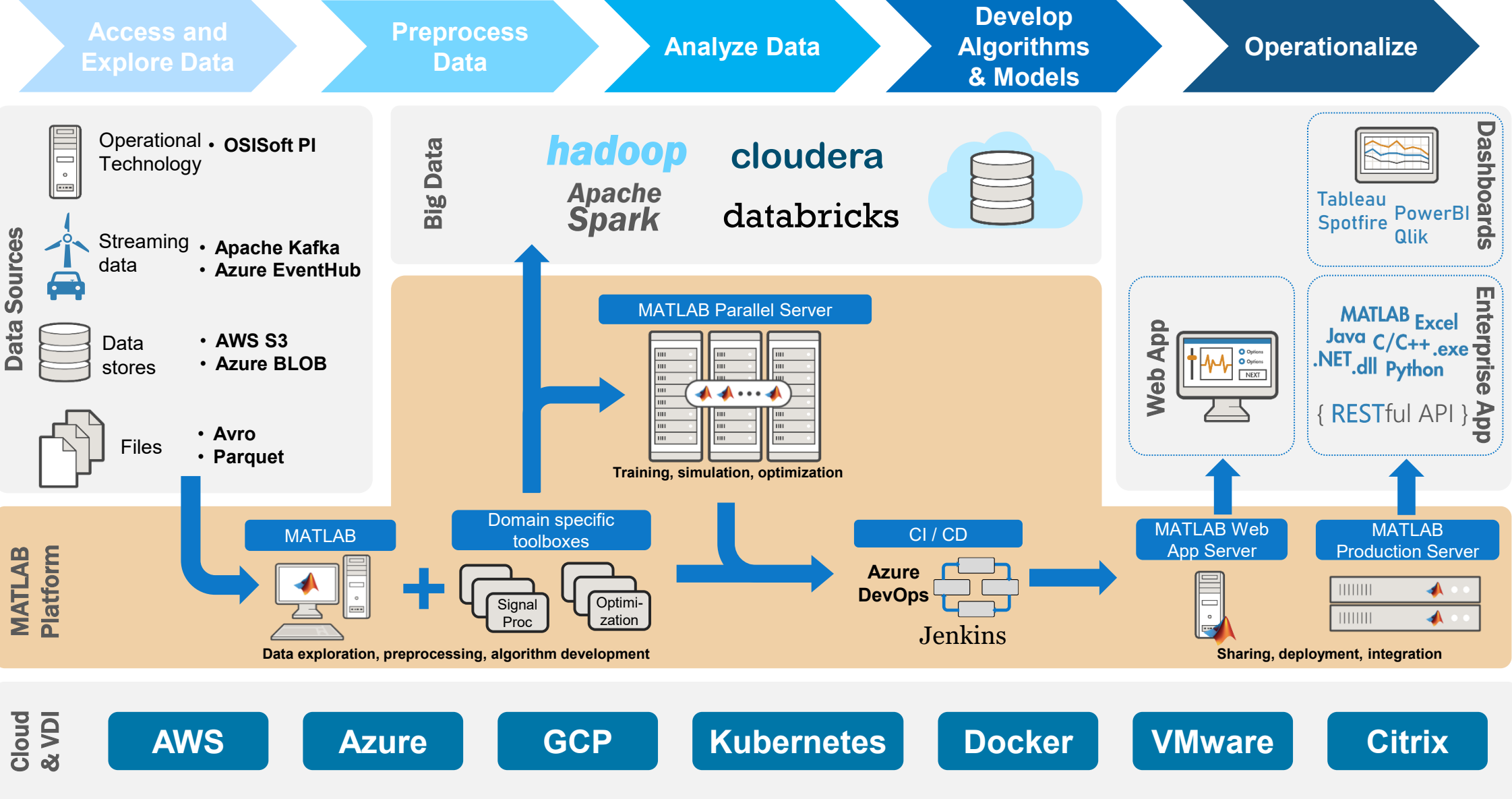
Deploy to Embedded Processors

Applications:



Requires Deep Learning Toolbox

Comprehensive end-to-end solutions for complex IT environments



MATLAB®校園授權-Campus Wide License

你擁有的不只是一個軟體!

校園授權享有的資源，比你想像多更多



各領域線上課程

程式基礎/人工智慧技巧課程，上完領證書

MATLAB雲端硬碟

5G 儲存空間，資料存取無障礙



線上版MATLAB

在網頁瀏覽器上使用

手機板MATLAB

在手機 / 平板上用 MATLAB，支援 ios/Android



叢集 & 分散式運算

提升平行運算速度



支援低成本硬體

Arduino /Raspberry Pi/ Beagle /LEGO

MATLAB線上作業評分系統

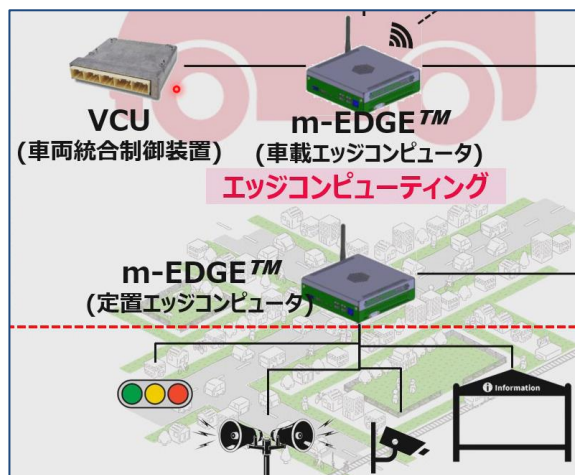
作業自動評分系統



軟體下載安裝

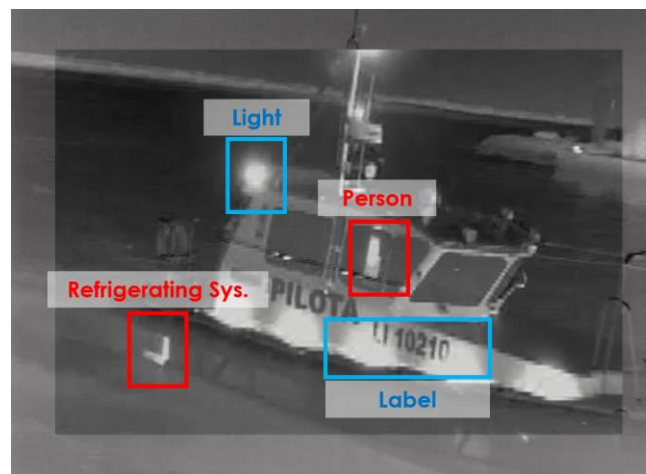
不限下載電腦台數/維護期內更新

Wide Range of Applications with GPU Coder



[DENSO Ten](#)

Vehicle detection and traffic flow analysis



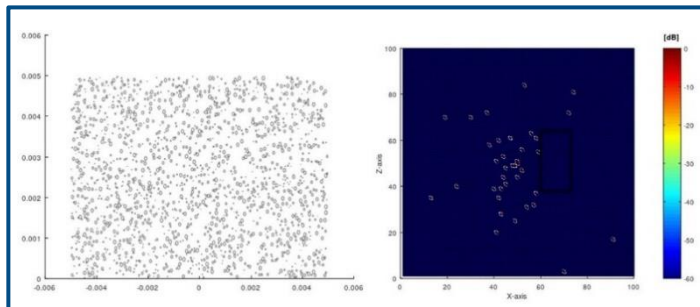
[Drass](#)

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

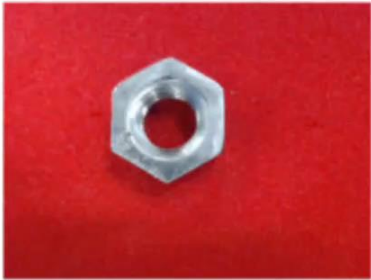
Automated Optical Inspection Example

Current Folder: tb_myNDNet_Jetson.mlx | myNDNet_Preprocess.m | myNDNet_Postprocess.m | targetFunction.m | +

CNN and post-processing

Read new image captured by webCam. The images used to train network were pre-processed image to fit it in input image size of network. Original image captured by webCam is like this;

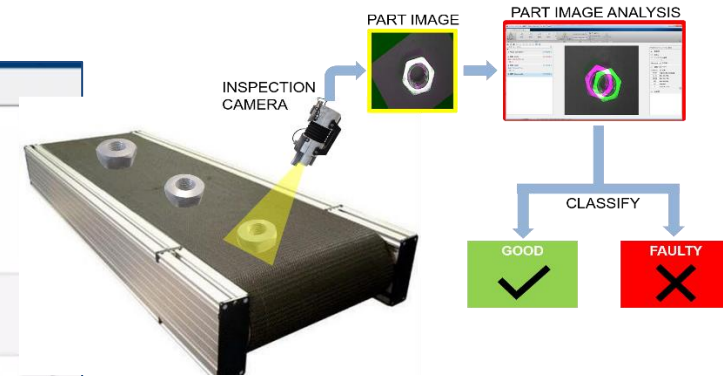
```
50 img = imread('testImg.png');  
51 figure, imshow(img)
```



52 % So, we need to extract where the nuts are as an ROI before passing image
53 % to the network.

Pre-processing for captured image from webcam

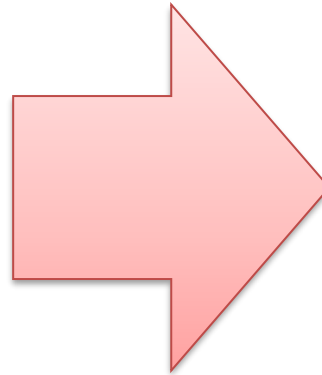
myNDNet_Preprocess is to extract ROI by using traditional image processing. Since generated C code from this m code will be



Why Use CUDA code generation ?

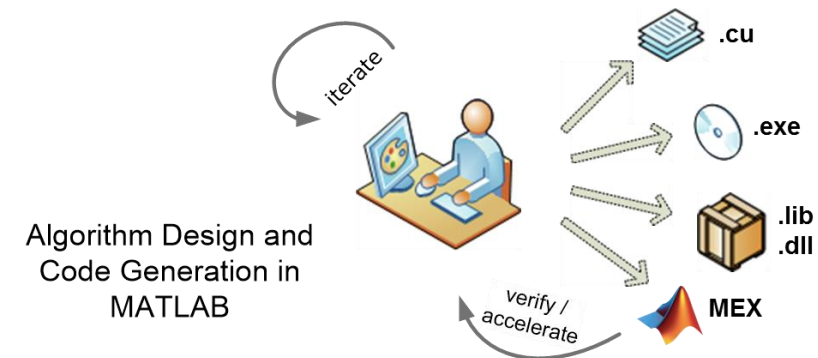
Pains: Hand code

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

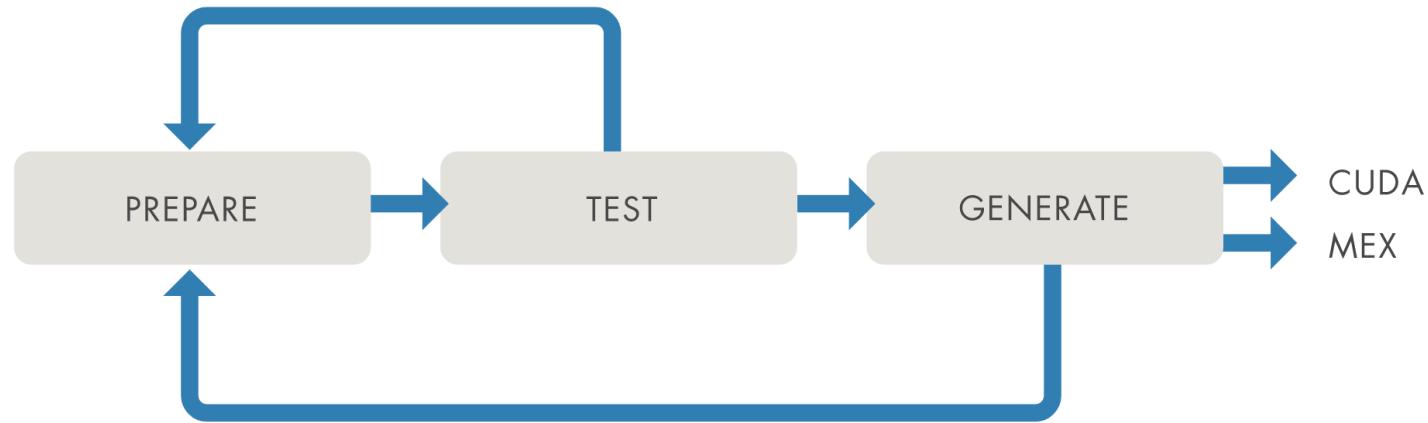


Solution: GPU Coder

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



Working with GPU Coder: Three-Step Workflow



Prepare your MATLAB algorithm for code generation

- Make implementation choices
- Use supported language features

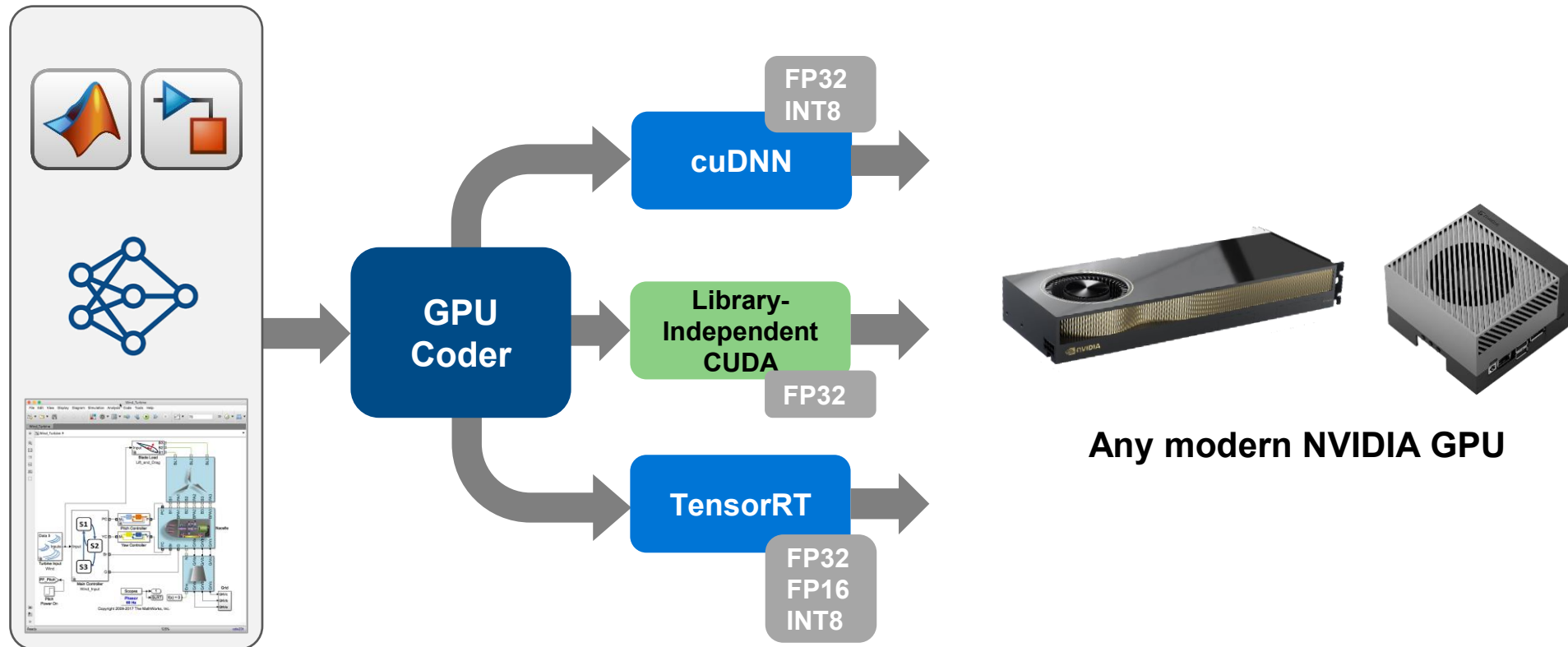
Test if your MATLAB code is ready for code generation

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

Generate source code or MEX for final use

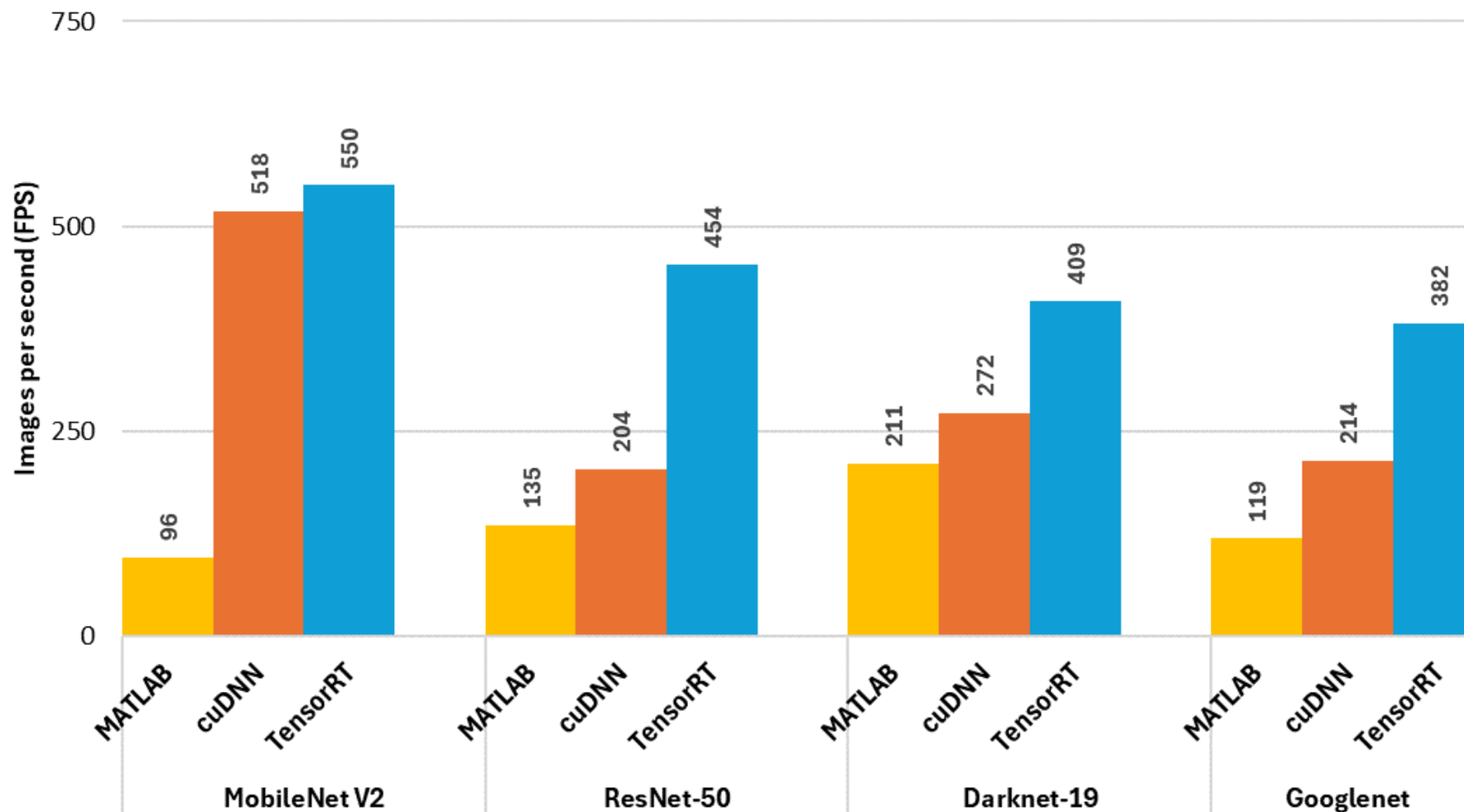
- Iterate your MATLAB code and use profiling tools to help optimize further
- Implement as source, executable, or library

Deep Learning Code Generation



Performance of Generated Code from GPU Coder

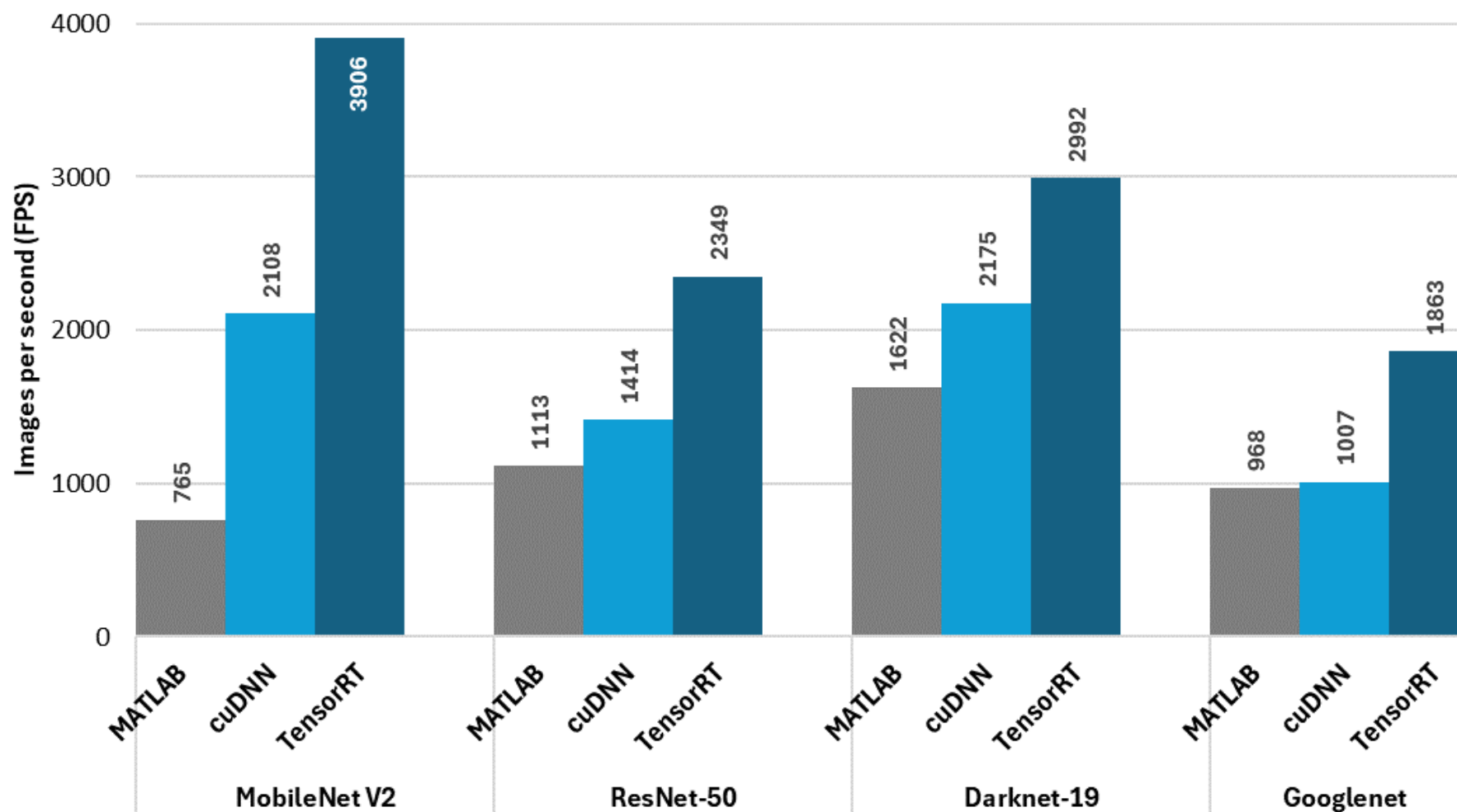
Batch Size 1



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

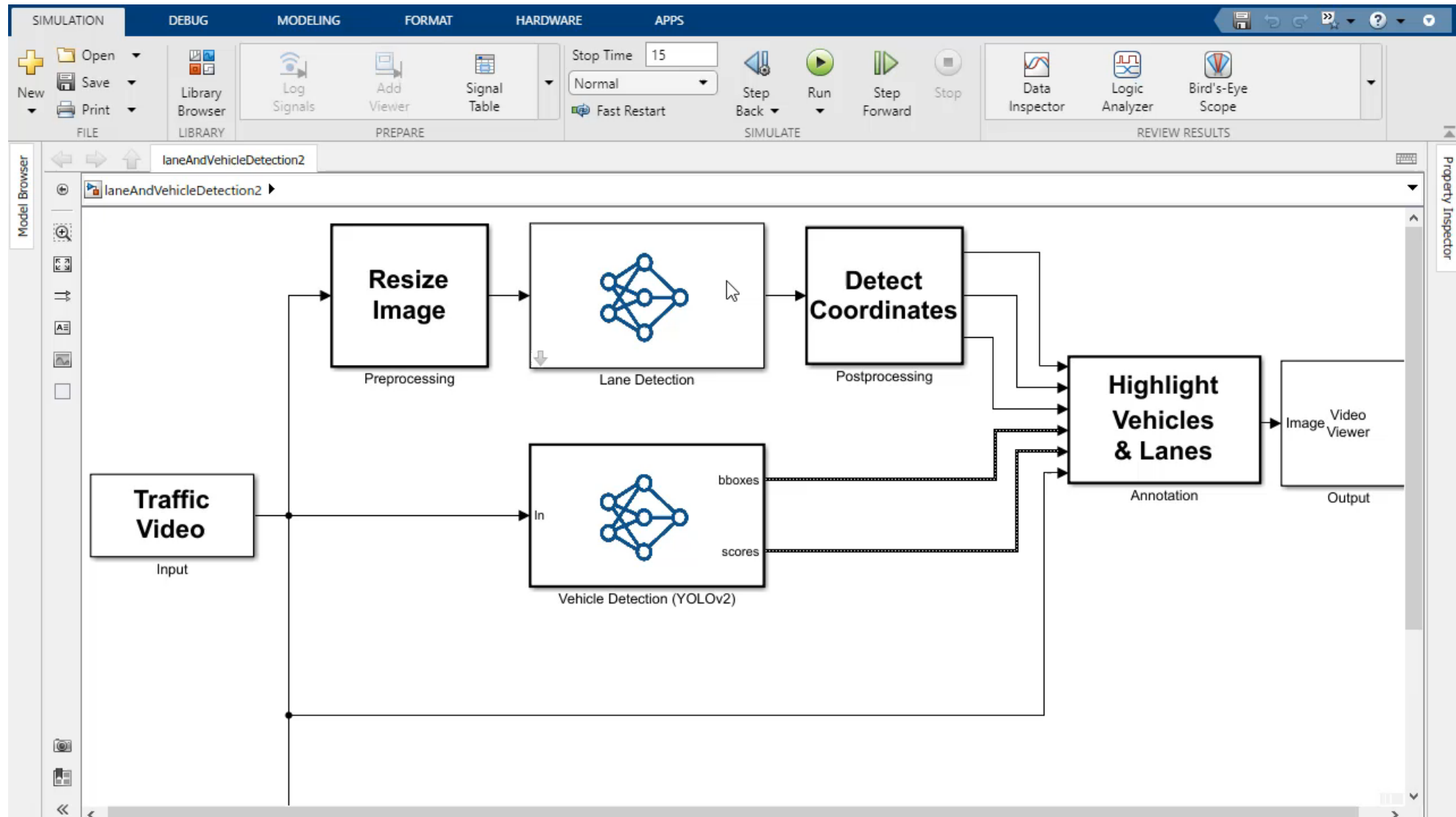
Performance of Generated Code from GPU Coder

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

Run Simulation on Desktop GPU



Example: Image Classification by ResNet-50

```
1 function out = resnet_predict(in) %#codegen
2 % Copyright 2020-2024 The MathWorks, Inc.
3
4 persistent dlnet;
5
6 dlIn = dlarray(in, 'SSC');
7 if isempty(dlnet)
8     % Call the function resnet50 that returns a dlnetwork object
9     % for ResNet-50 model.
10    dlnet = imagePretrainedNetwork('resnet50');
11 end
12
13 dlOut = predict(dlnet, dlIn);
14 out = extractdata(dlOut);
15
16 end
```

ell pepper

ucumber

mon

corn squash

amper

- `gpuDeviceCount` % 顯示 GPU 數量
- `gpuDevice(1)` % 顯示第一張 GPU 的詳細資訊

```
ans =
```

```
CUDADevice with properties:
```

```
      Name: 'NVIDIA RTX A500 Laptop GPU'  
      Index: 1 (of 1)  
ComputeCapability: '8.6'  
      DriverModel: 'WDDM'  
      TotalMemory: 4294508544 (4.29 GB)  
AvailableMemory: 3464757248 (3.46 GB)  
DeviceAvailable: true  
      DeviceSelected: true
```

```
Show all properties.
```

Verify GPU code generation environment

- `coder.checkGpuInstall`

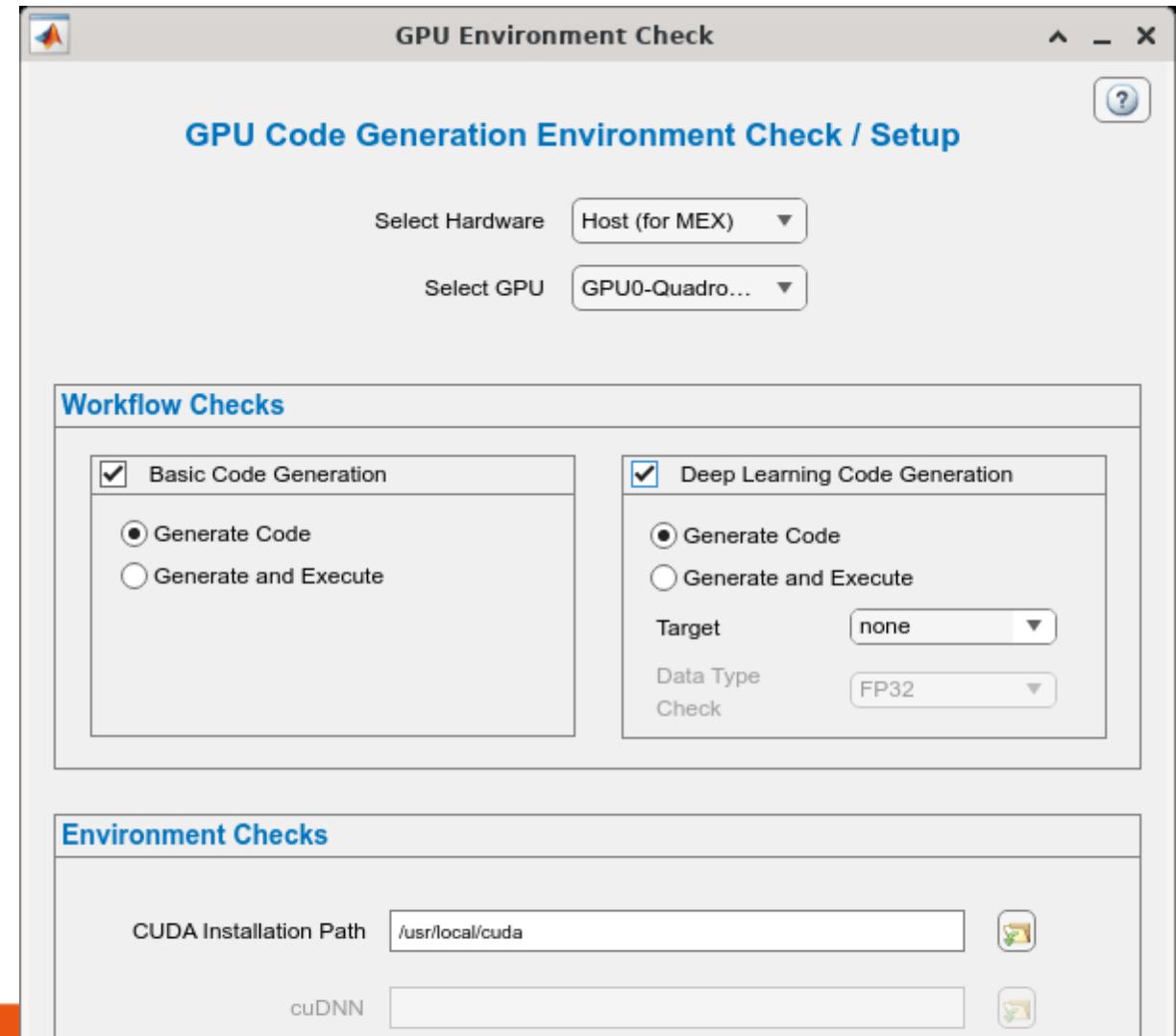
```
Compatible GPU          : PASSED
CUDA Environment       : PASSED
    Runtime            : PASSED
    cuFFT              : PASSED
    cuSOLVER           : PASSED
    cuBLAS             : PASSED
Host Compiler          : PASSED
Basic Code Generation   : PASSED
Basic Code Execution    : PASSED
```

```
results =
```

```
struct with fields:
```

```
    gpu: 1
    cuda: 1
    cudnn: 0
    tensorrt: 0
```

- `gpuCoderSetup`



The screenshot shows the 'GPU Environment Check / Setup' dialog box. It has a title bar with a question mark icon. The main title is 'GPU Code Generation Environment Check / Setup'. Below the title, there are two dropdown menus: 'Select Hardware' set to 'Host (for MEX)' and 'Select GPU' set to 'GPU0-Quadro...'. The dialog is divided into two main sections: 'Workflow Checks' and 'Environment Checks'. The 'Workflow Checks' section contains two sub-sections: 'Basic Code Generation' and 'Deep Learning Code Generation'. Both are checked. Under 'Basic Code Generation', there are two radio buttons: 'Generate Code' (selected) and 'Generate and Execute'. Under 'Deep Learning Code Generation', there are two radio buttons: 'Generate Code' (selected) and 'Generate and Execute'. Below these, there are two dropdown menus: 'Target' set to 'none' and 'Data Type' set to 'FP32'. The 'Environment Checks' section contains two input fields: 'CUDA Installation Path' set to '/usr/local/cuda' and 'cuDNN' (empty). Both input fields have a folder icon to their right.

Generate GUDA code from GPU Coder

The screenshot displays the GPU Coder Report Viewer interface. The top window is titled "GPU Coder - resnet_predict.prj". Below it, the "GPU Coder Report Viewer" window shows a report for the project located at "C:\Users\Judy\Documents\MATLAB\Examples\R2024b\deeplearning_shared\CodeGenerationForDeepLearningNetworksExample\codegen\dl\resnet_predict\html\rep...".

The interface includes a "REPORT" tab and a "GPU Code" sidebar. The main area is divided into three sections:

- NAVIGATE:** Contains buttons for "Back", "Forward", "Go To", "Find", "Trace Code", "Edit In MATLAB", "Package Code", and "Export Report Information".
- TRACE:** A section for tracing the code execution.
- EDIT:** A section for editing the code.
- SHARE:** A section for sharing the report.

The left sidebar shows a tree view of the project files, including "MATLAB Source" and "Generated Code". The "Generated Code" section is expanded, showing a list of files such as "resnet_predict.cu", "resnet_predict.h", "resnet_predict_data.cu", "resnet_predict_data.h", "resnet_predict_initialize.cu", "resnet_predict_initialize.h", "resnet_predict_internal_types.h", "resnet_predict_rtutil.cu", "resnet_predict_rtutil.h", "resnet_predict_spec.h", "resnet_predict_terminate.cu", "resnet_predict_terminate.h", "resnet_predict_types.h", "rtwtypes.h", "shared_layers_export_macros.hpp", and "tmwtypes.h".

The main content area displays the C++ code for "resnet_predict.cu". The code includes headers, function definitions, and the implementation of the "resnet_predict" function. The code is as follows:

```
19 #include "resnet_predict.h"
20
21 static boolean_T dlnet_not_empty;
22
23 // Function Definitions
24 //
25 // Copyright 2020-2024 The MathWorks, Inc.
26 //
27 // Arguments      : const float cpu_in[150528]
28 //                  float cpu_out[1000]
29 // Return Type    : void
30 //
31 void resnet_predict(const float cpu_in[150528], float cpu_out[1000])
32 {
33     float(*gpu_in)[150528];
34     float(*gpu_out)[1000];
35     if (!isInitialized_resnet_predict) {
36         resnet_predict_initialize();
37     }
38     b_checkCudaError(mwCudaMalloc(&gpu_out, 4000ULL), __FILE__, __LINE__);
39     b_checkCudaError(mwCudaMalloc(&gpu_in, 602112ULL), __FILE__, __LINE__);
40     if (!dlnet_not_empty) {
41         // Call the function resnet50 that returns a dlnetwork object
42     }
```

At the bottom of the window, a "Summary" section indicates that the code generation was successful. The summary includes the following information:

- Generated on:** 12-Jun-2025 23:01:22
- Build type:** Dynamic Library
- Toolchain:** NVIDIA CUDA (w/Microsoft Visual C++ 2017) | nmake (64-bit Windows)
- Build:** Faster Runs
- Configuration:** (The configuration details are partially obscured by a scrollbar.)

Python code

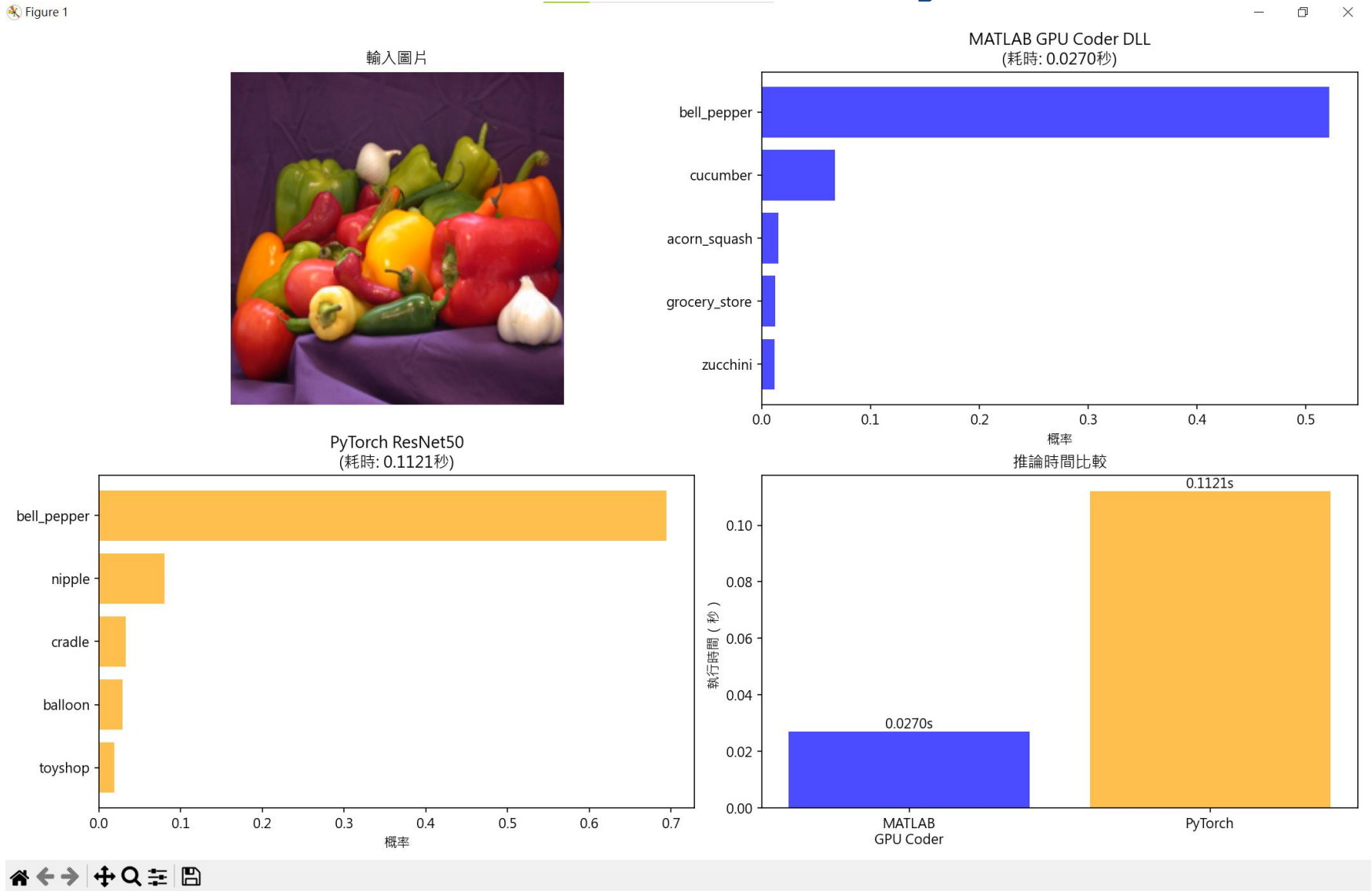
```
# === 設定CUDA環境路徑 ===
try:
    os.add_dll_directory(r"C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.6\bin")
    os.add_dll_directory(r"C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\cuDNN\bin")
    os.add_dll_directory(r"D:\ProgramFiles\MATLAB\R2024b\bin\win64") # 視你的版本而定
except:
    print("警告：無法設定CUDA路徑，DLL可能無法正常運作")
```

```
# === GPU Coder DLL 預測 ===
def setup_dll():
    dll_path = r"D:\GPU_Coder\codegen\dll\resnet_predict\resnet_predict.dll"
    if not os.path.exists(dll_path):
        print(f"[錯誤] 找不到DLL檔案: {dll_path}")
        sys.exit(1)

    dll = ctypes.CDLL(dll_path)

    # 簡化參數設定，使用基本的POINTER方式
    dll.resnet_predict.argtypes = [
        ctypes.POINTER(ctypes.c_float), # 輸入
        ctypes.POINTER(ctypes.c_float) # 輸出
    ]
    dll.resnet_predict.restype = None
    print("使用基本POINTER方法設定DLL參數")
```

速度比較: MATLAB GPU Coder DLL vs PyTorch: 4.15x



Key Takeaways

- MATLAB empowers rapid development of visual inspection algorithms
- GPU Coder automatically converts these algorithms into native CUDA C code
- Seamless integration with Python-based systems
- 對學校的價值
 - 👉 GPU Coder 讓師生專注研究，而不是卡在 CUDA 細節
 - 👉 最大化 GPU cluster 投資效益，推動跨領域研究與產學合作

