MATLAB EXPO 2018

Master Class: Deep Learning

Del Prototipo a su Despliegue en Entornos Embarcados

Lucas García


Deep Learning Demo
Image Classification

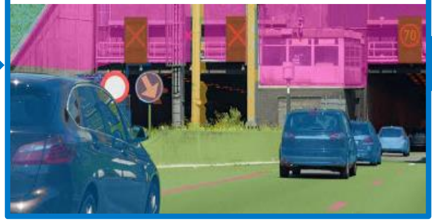
Why MATLAB for Deep Learning?

- MATLAB is Productive
- MATLAB Integrates with Open Source
- MATLAB is Fast

MATLAB Deep Learning Framework

Access Data

Design + Train



Deploy

- **Manage** large image sets
- **Automate** image labeling
- **Easy access** to models

- **Acceleration** with GPU’s
- **Scale** to clusters

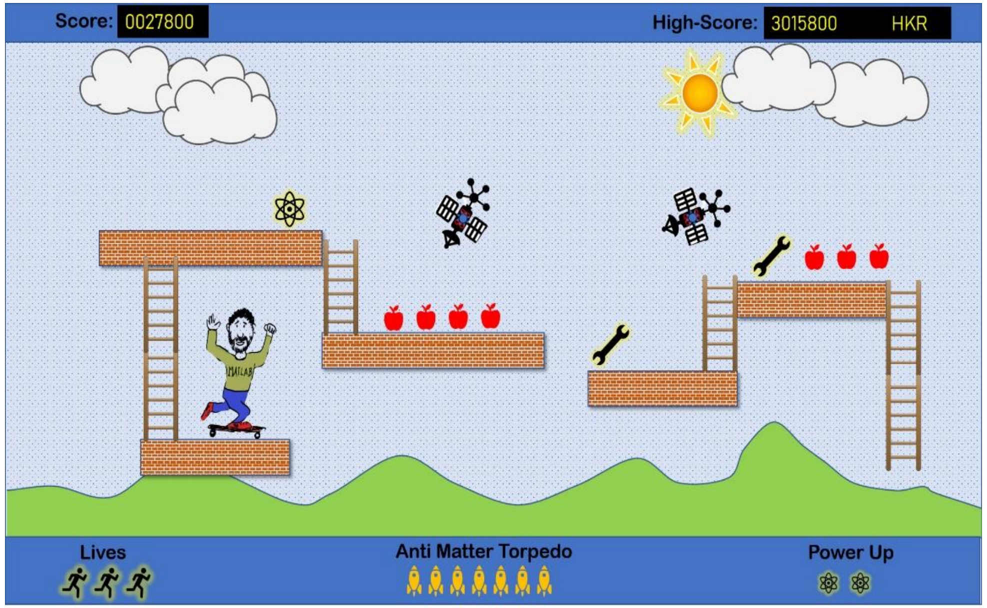
- **Automate compilation to GPUs and CPUs using GPU Coder:**
  - **5x faster** than TensorFlow
  - **2x faster** than MXNet

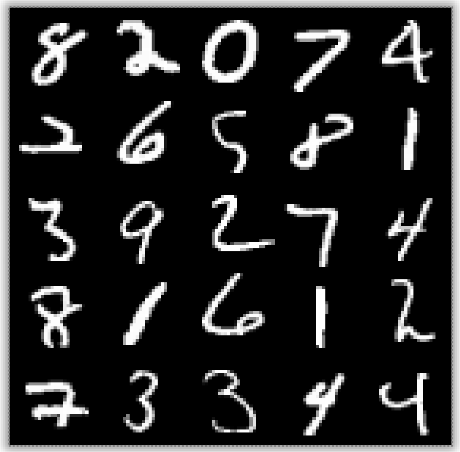
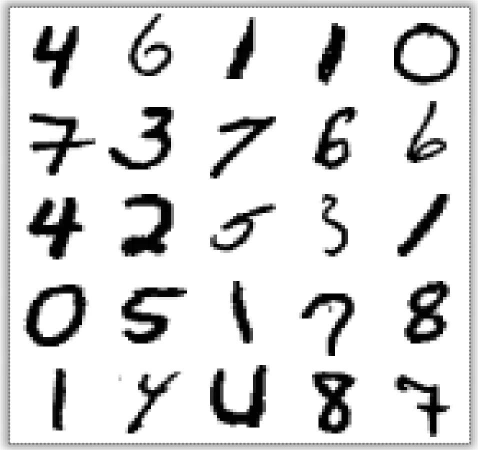
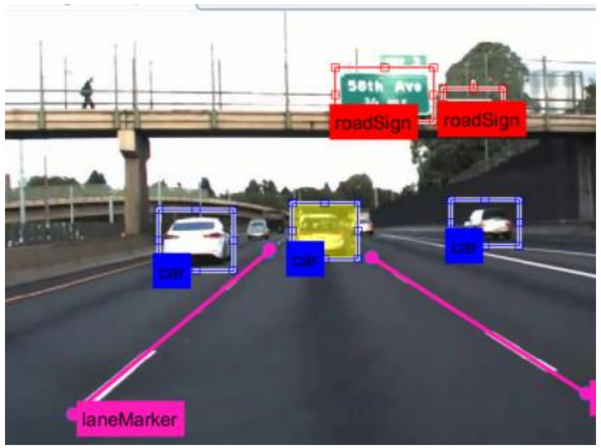
Deep Learning Applications

Voice assistants (speech to text)

Teaching character to beat video game

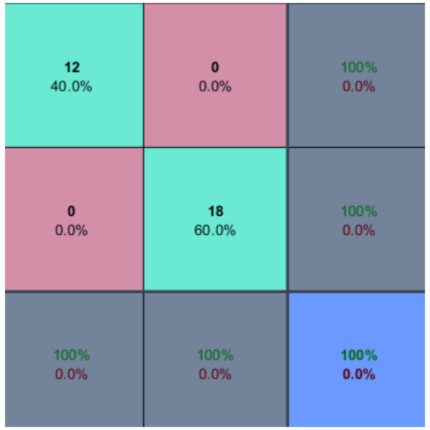
Automatically coloring black-and-white images



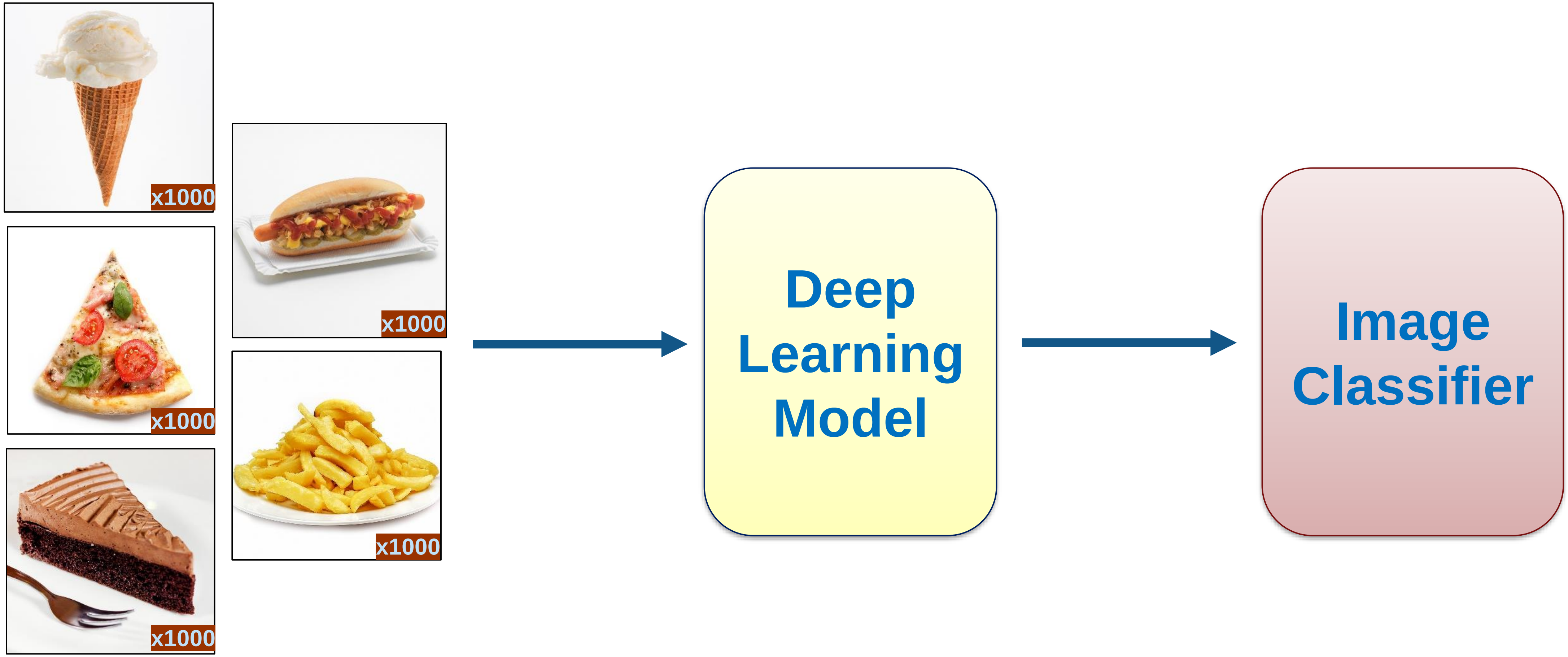
What is Deep Learning?



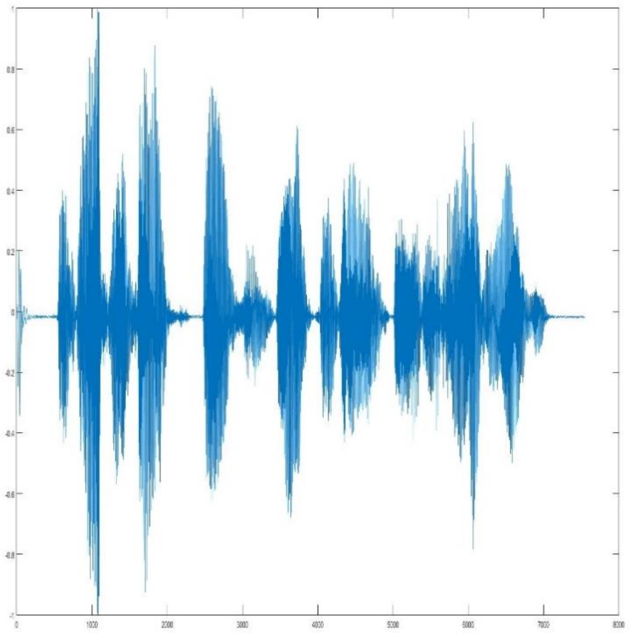


Deep Learning

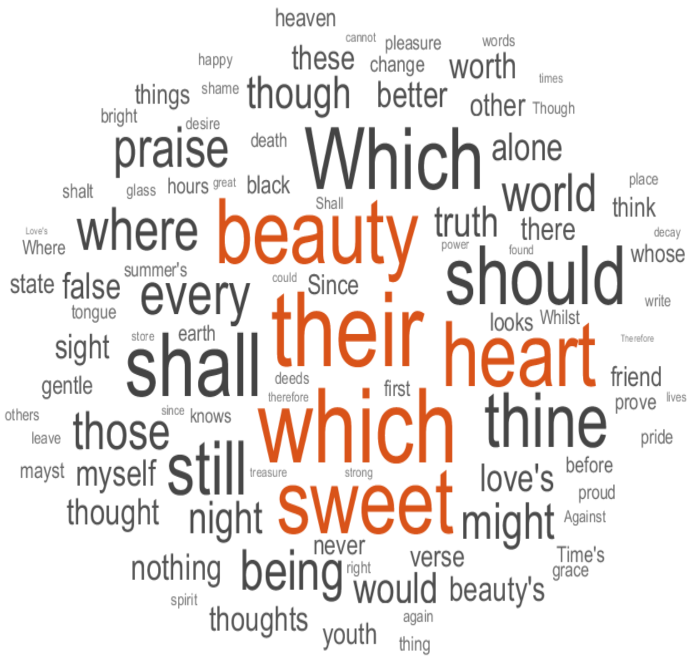
Model learns to perform classification tasks directly from data.



Data Types for Deep Learning



Signal



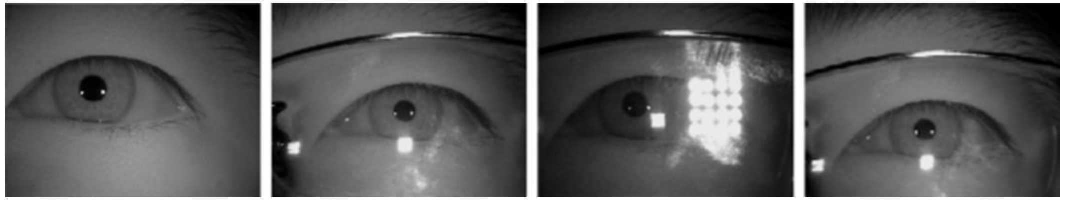
Text

Image

Deep Learning is Versatile

Detection of cars and road in autonomous driving systems

Rain Detection and Removal[1]

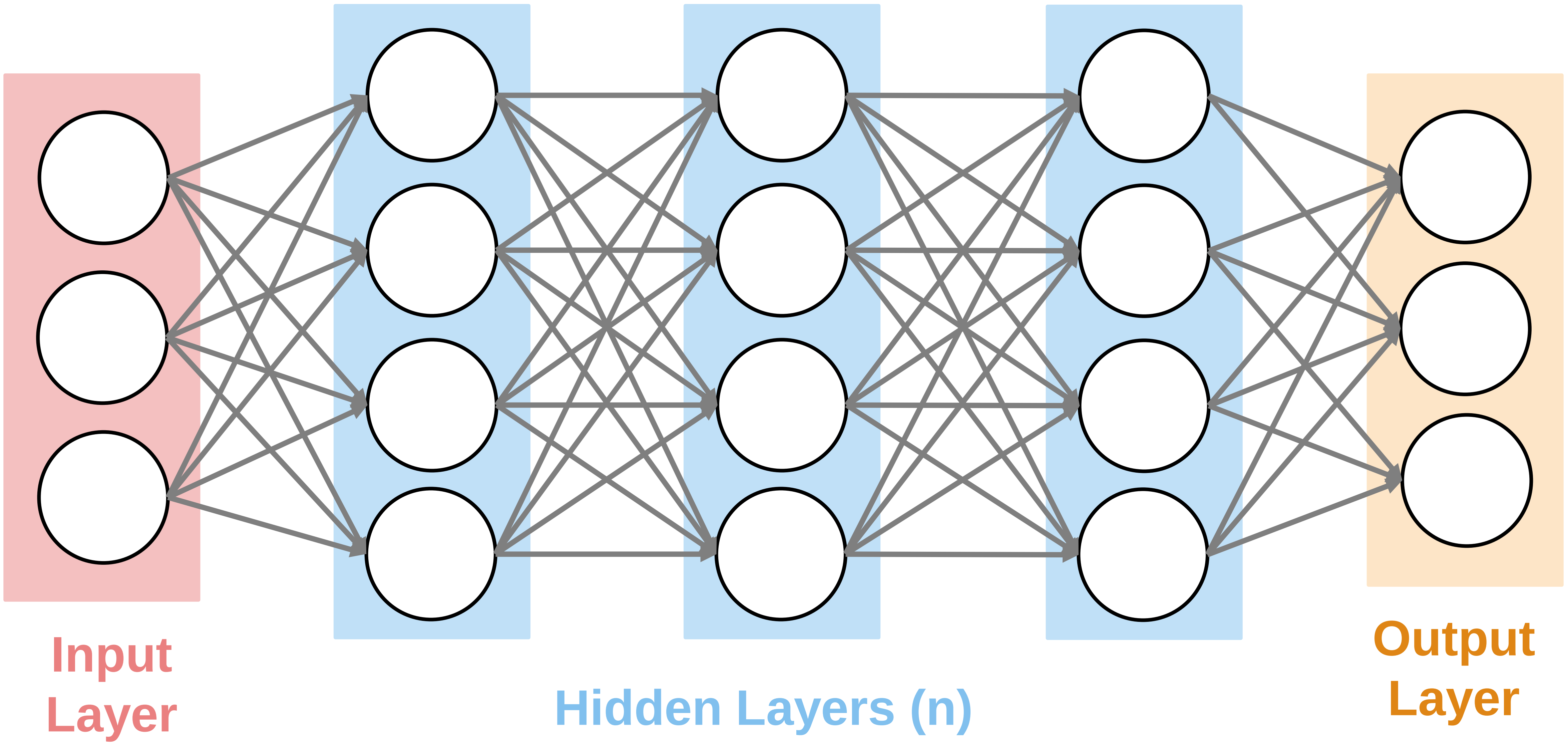


Iris Recognition – 99.4% accuracy[2]

1. *Deep Joint Rain Detection and Removal from a Single Image" Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan*
2. *Source: An experimental study of deep convolutional features for iris recognition Signal Processing in Medicine and Biology Symposium (SPMB), 2016 IEEE Shervin Minaee ; Amirali Abdolrashidiy ; Yao Wang; An experimental study of deep convolutional features for iris recognition*

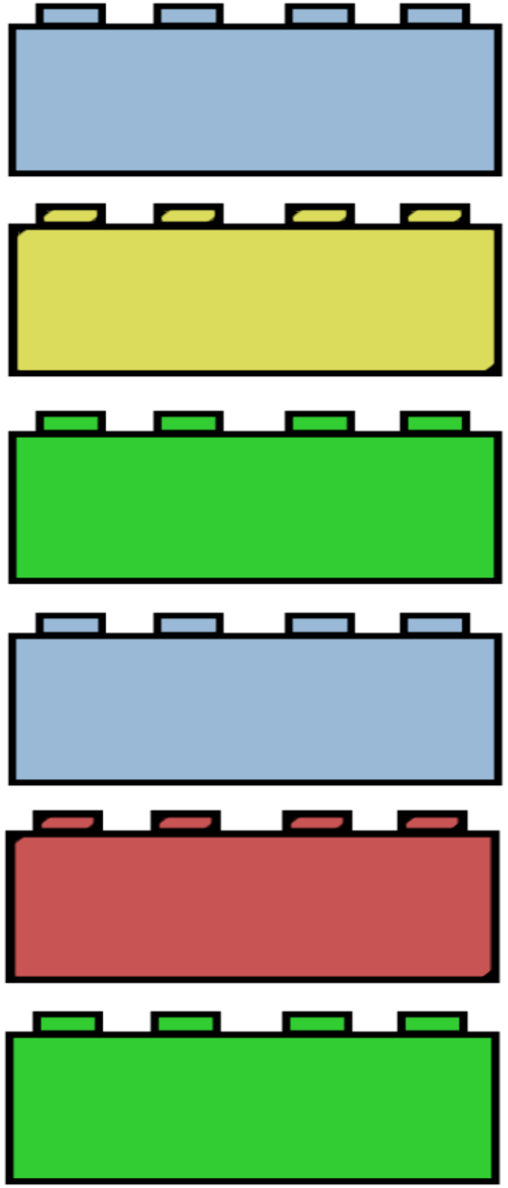
How is deep learning performing so well?

Deep Learning Uses a Neural Network Architecture



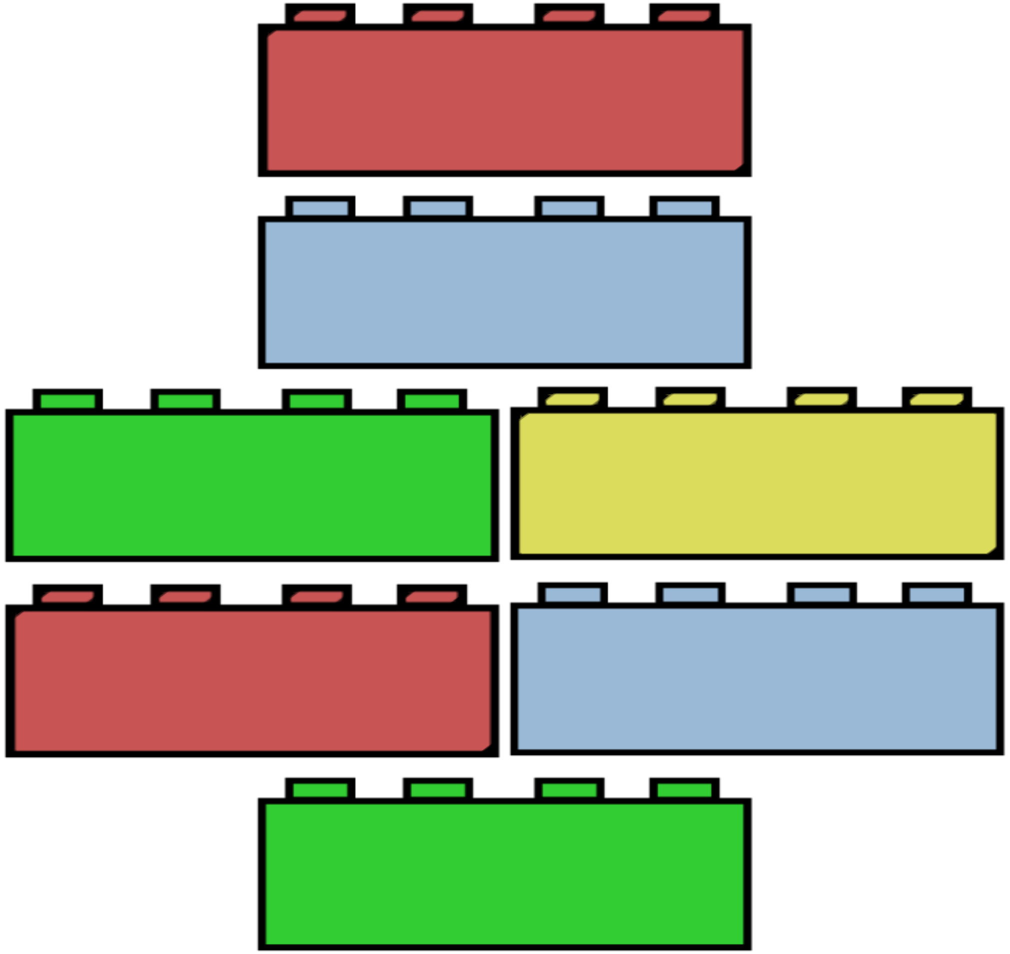
Thinking about Layers

- Layers are like blocks
  - Stack them on top of each other
  - Replace one block with a different one
- Each hidden layer processes the information from the previous layer

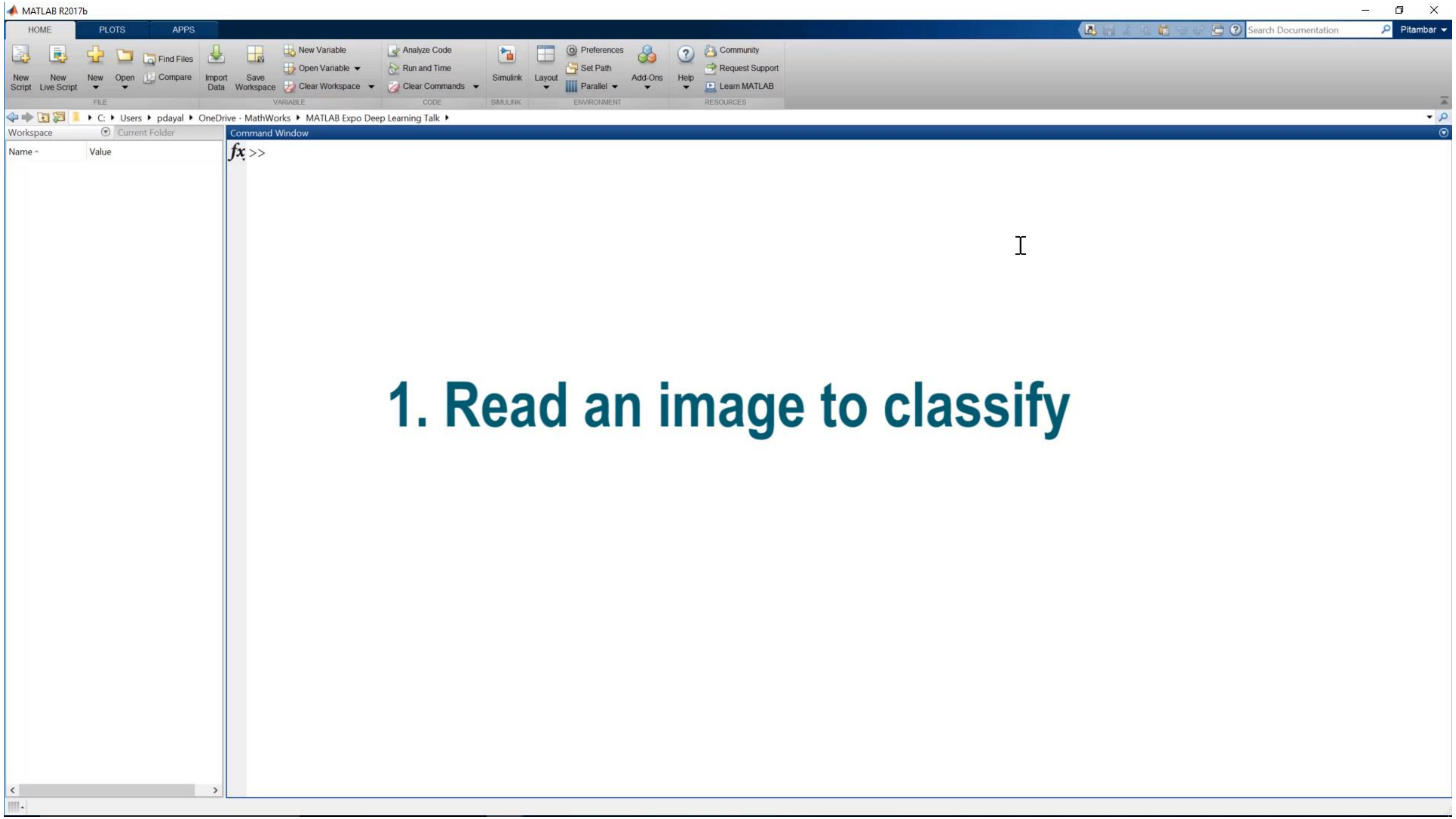


Thinking about Layers

- Layers are like blocks
  – Stack them on top of each other
  – Replace one block with a different one
- Each hidden layer processes the information from the previous layer
- Layers can be ordered in different ways



Deep Learning in 6 Lines of MATLAB Code



Why MATLAB for Deep Learning?

- MATLAB is Productive
- MATLAB integrates with Open Source
- MATLAB is Fast

“I love to label and preprocess my data”

~ Said no engineer, ever.

Caterpillar Case Study

- World's leading manufacturer of construction and mining equipment.

- Similarity between these projects?
  - Autonomous haul trucks
  - Pedestrian detection
  - Equipment classification
  - Terrain mapping

Computer Must Learn from Lots of Data

- ALL data must first be labeled to create these autonomous systems.

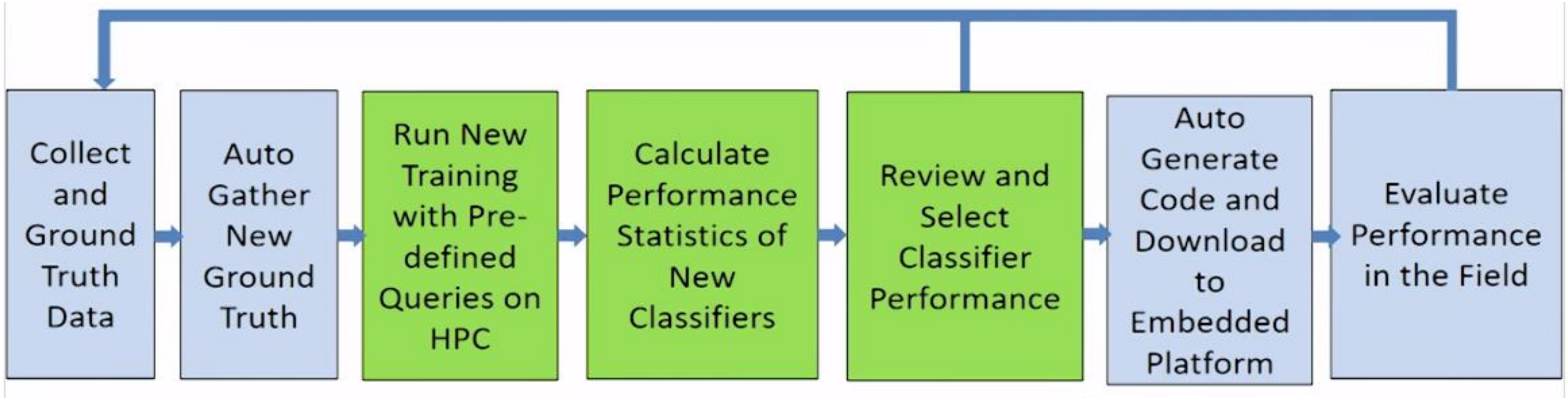
“We were spending way too much time ground-truthing [the data]”

--Larry Mianzo, Caterpillar

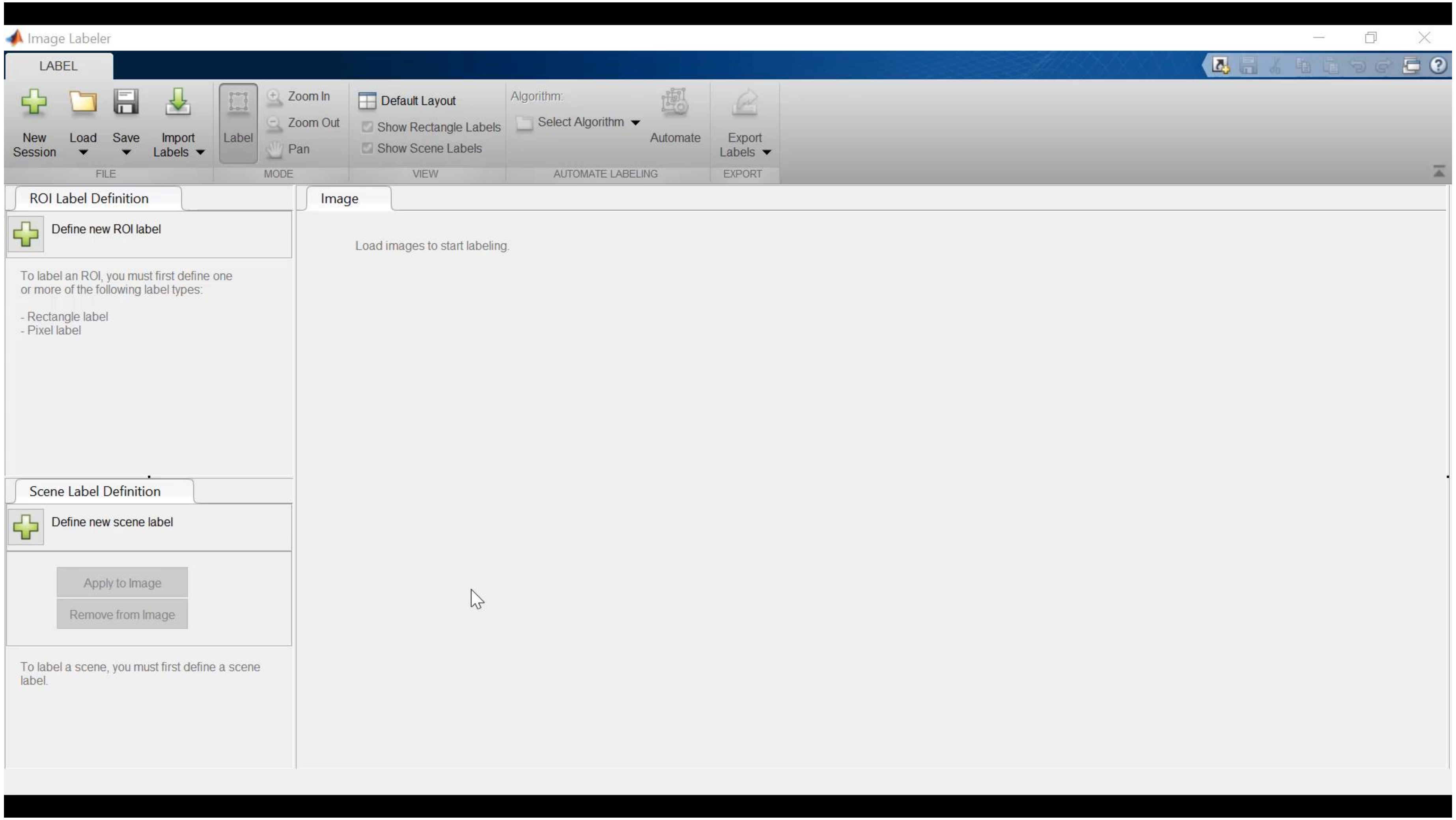
How Did Caterpillar Do with Our Tools?

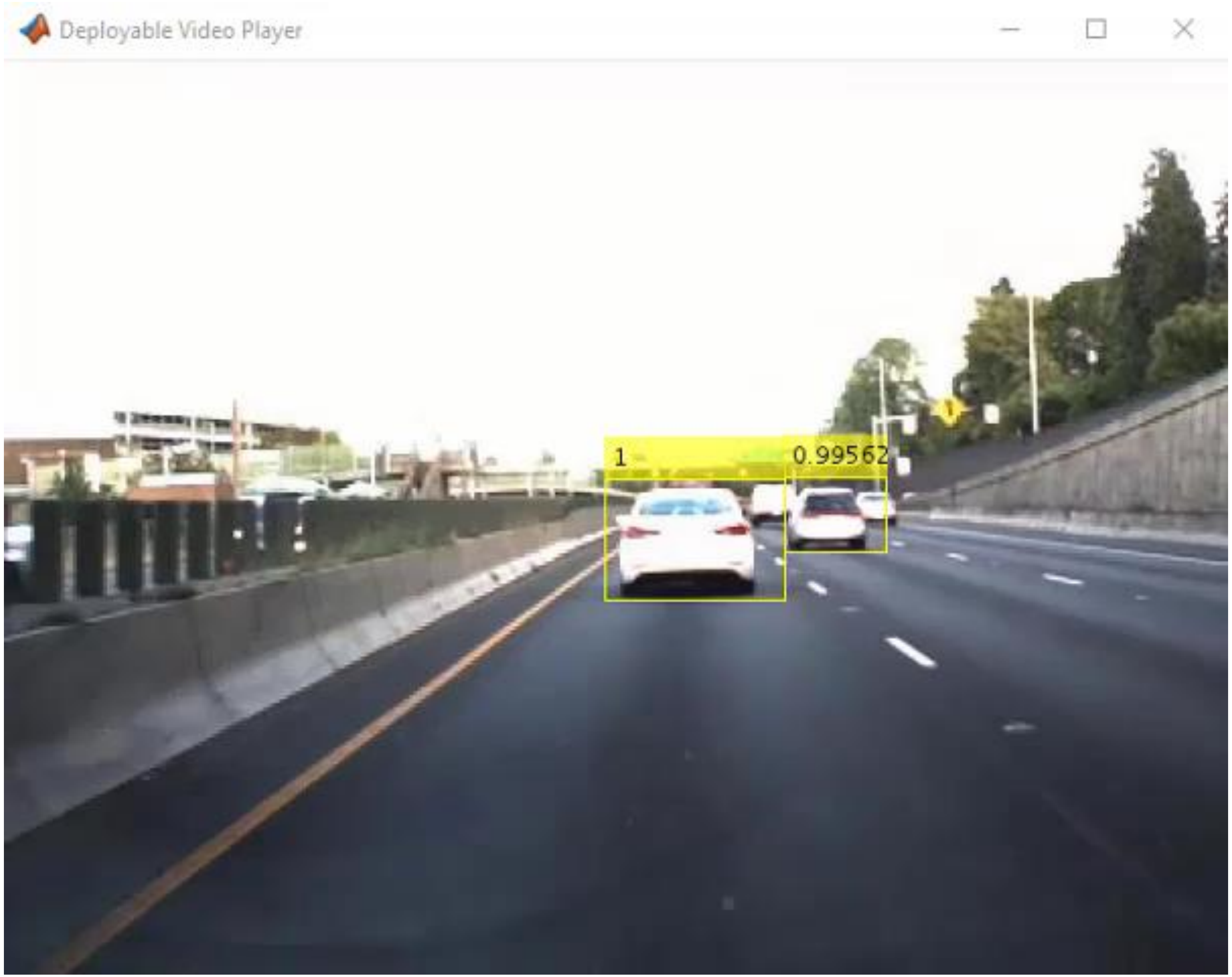
- Semi-automated labeling process
  - *"We go from having to label 100 percent of our data to only having to label about 80 to 90 percent"*

- Used MATLAB for entire development workflow.
  - *"Because everything is in MATLAB, development time is short"*



How Does MATLAB Come into Play?




Deployable Video Player
1
0.99562

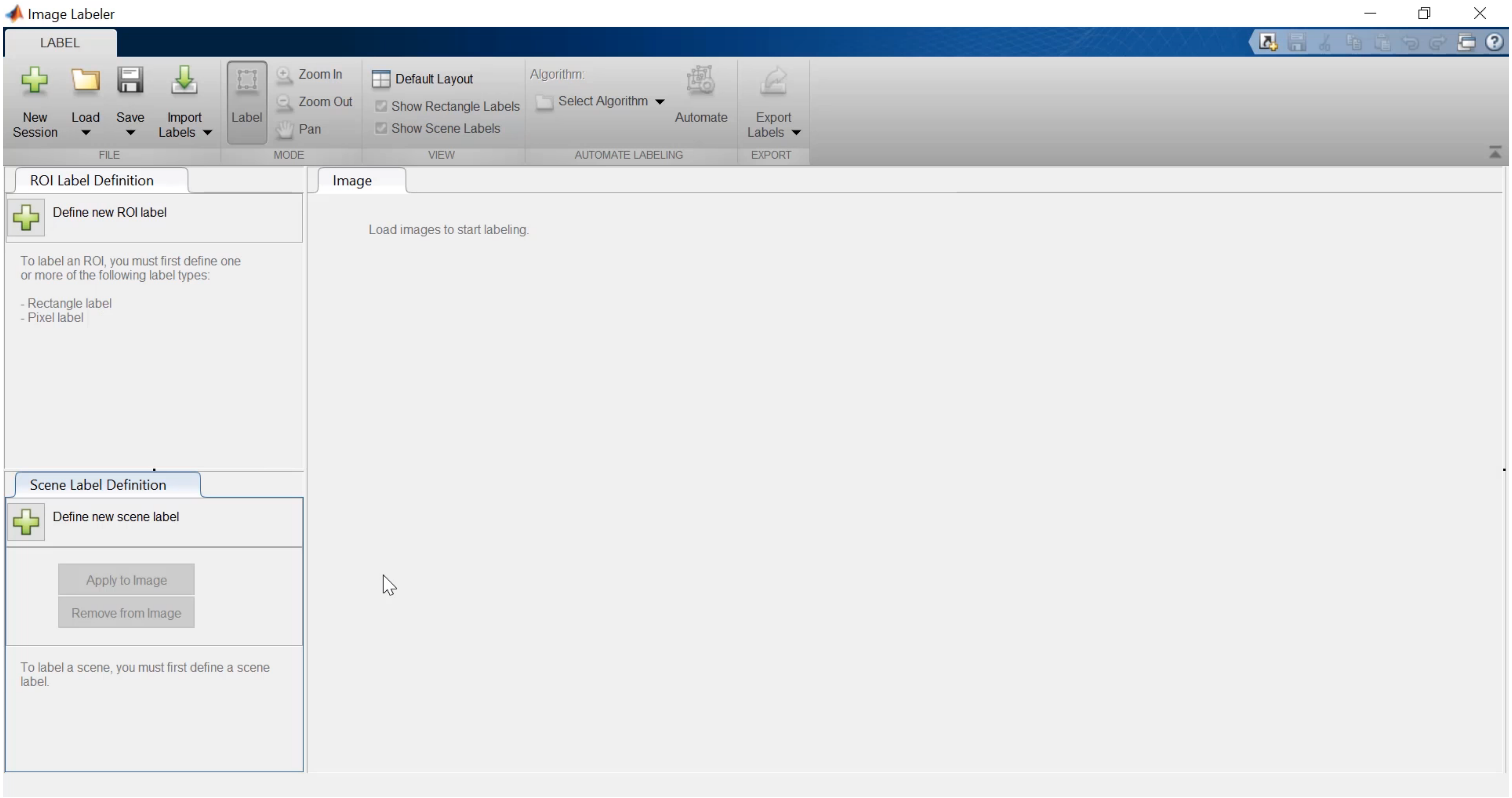


Image Labeler
LABEL
New Session
Load
Save
Import Labels
FILE
Label
Zoom In
Zoom Out
Pan
MODE
Default Layout
Show Rectangle Labels
Show Scene Labels
VIEW
Algorithm:
Select Algorithm
Automate
AUTOMATE LABELING
Export Labels
EXPORT
ROI Label Definition
Define new ROI label
To label an ROI, you must first define one or more of the following label types:
- Rectangle label
- Pixel label
Scene Label Definition
Define new scene label
Apply to Image
Remove from Image
To label a scene, you must first define a scene label.
Image
Load images to start labeling.


RICE HOUSE

MATLAB is Productive

- Image Labeler App semi-automates labeling workflow
- Bootstrapping
  - Improve automatic labeling by updating algorithm as you label more images correctly.

- Easy to load metadata even when labeling manually

Why MATLAB?

- MATLAB is Productive
- MATLAB Integrates with Open Source
- MATLAB is Fast

Used MATLAB and Open Source Together

1. *Deep Joint Rain Detection and Removal from a Single Image" Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan*

- Used Caffe and MATLAB together
- Achieved significantly better results than an engineered rain model.

- Use our tools where it makes your workflow easier!

MATLAB Integrates with Open Source Frameworks

- Access to many pretrained models through add-ons
- Users wanted to import latest models

- Import models directly from TensorFlow or Caffe
  - Allows for improved collaboration

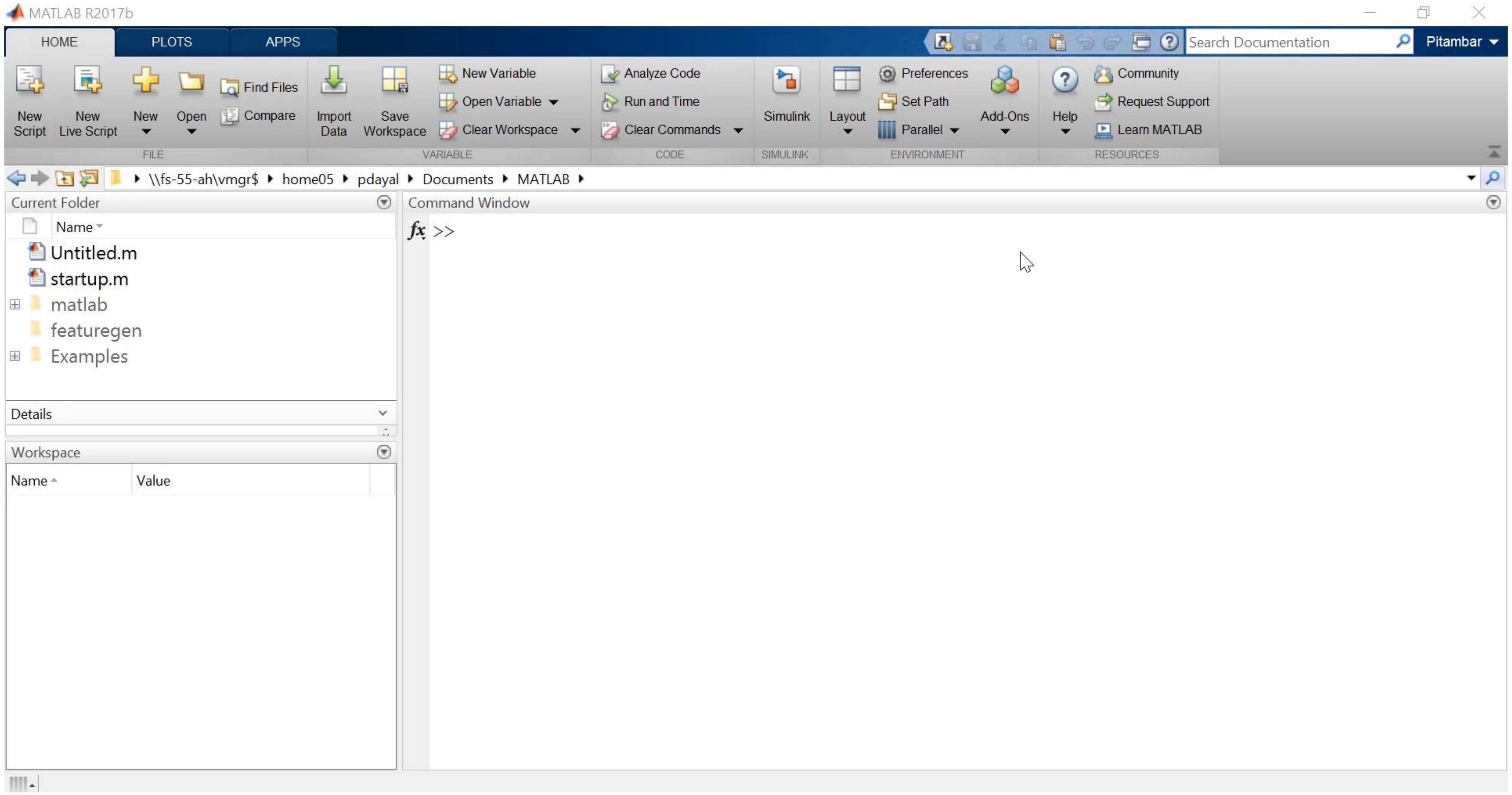
KERAS IMPORTER

Importer for TensorFlow-Keras Models

Caffe

MODELS

Keras-TensorFlow Importer



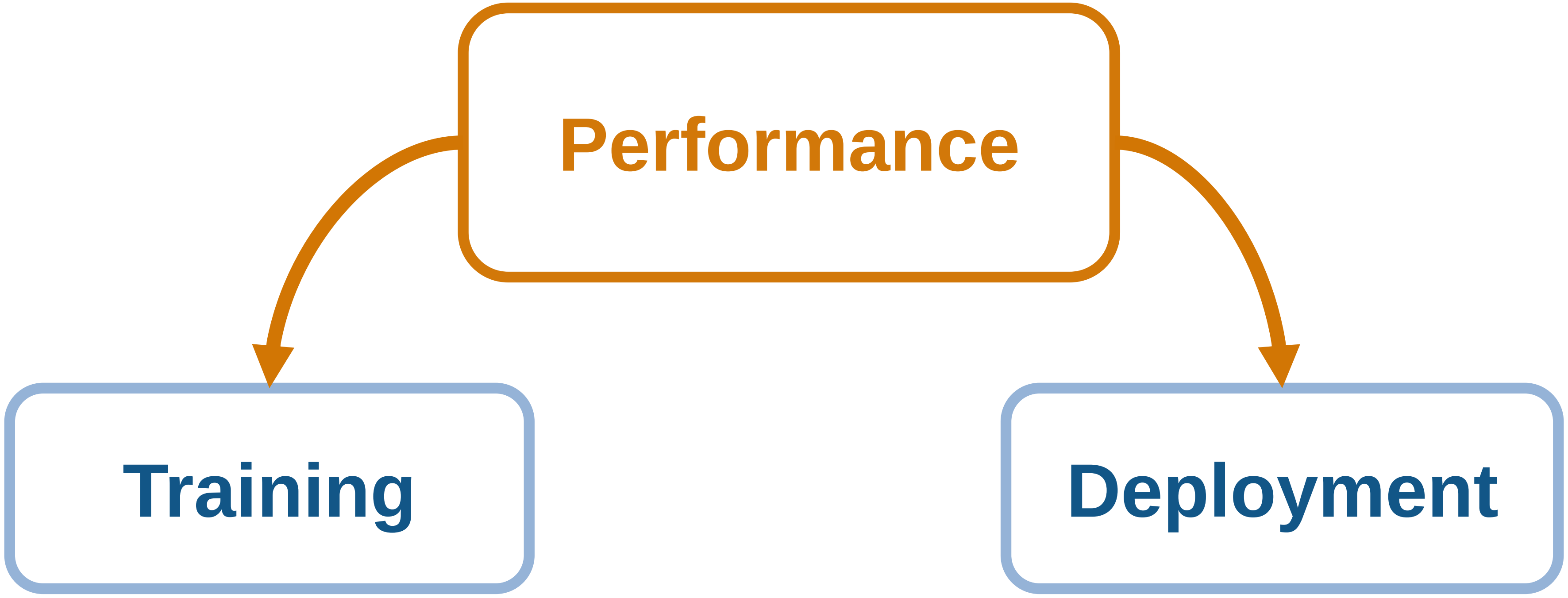
MATLAB Integrates with Open Source Frameworks

- MATLAB supports entire deep learning workflow
  – Use when it is convenient for your workflow
- Access to latest models
- Improved collaboration with other users

Why MATLAB?

- MATLAB is Productive
- MATLAB Integrates with Open Source
- MATLAB is Fast

MATLAB is Fast



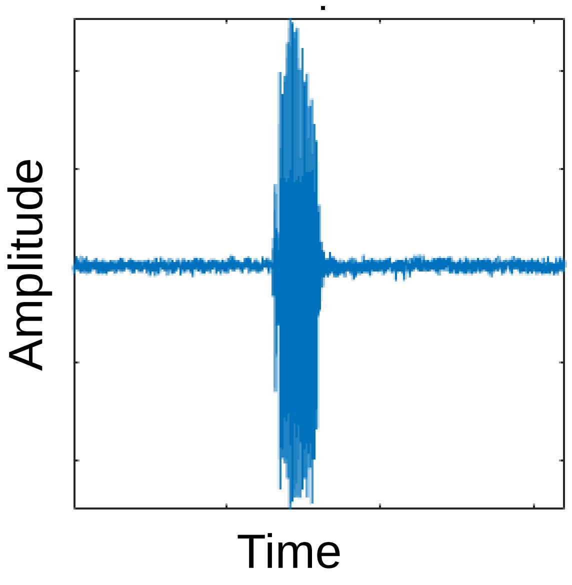
What is Training?

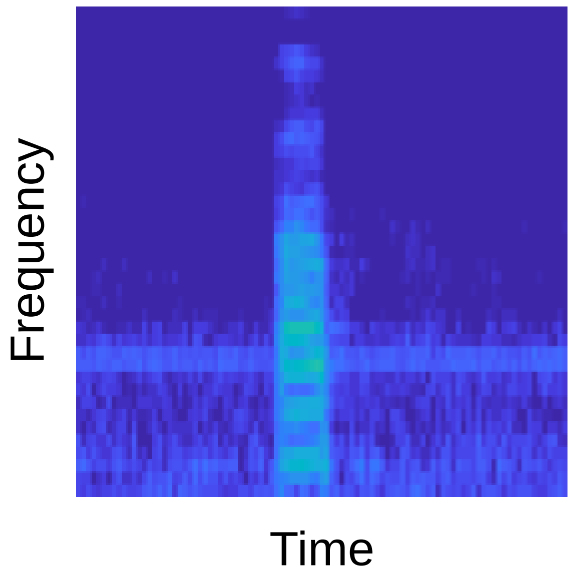
Feed labeled data into neural network to create working model

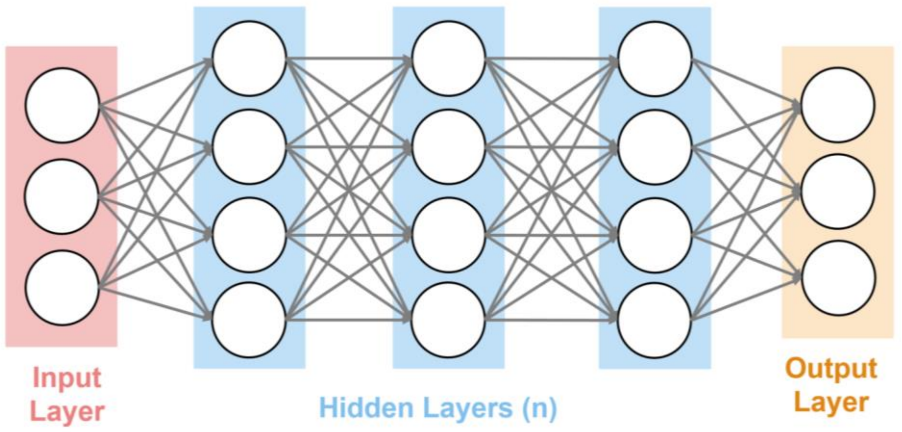
Convolution Neural Network → **Image Classifier Model**

Speech Recognition Example

Audio signal → Spectrogram → Image Classification algorithm

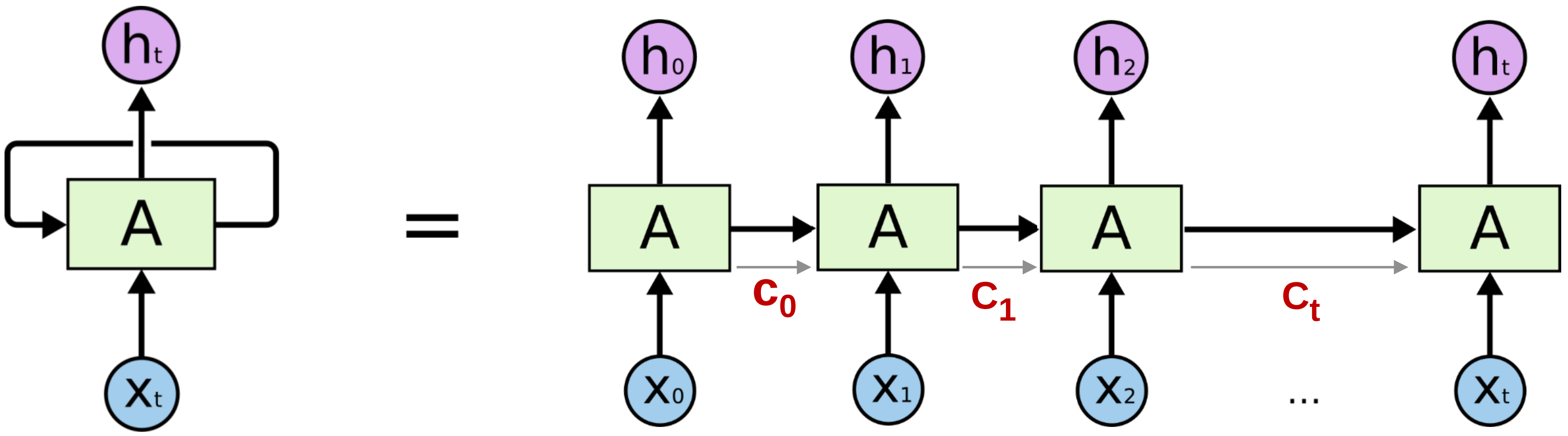






Another Network for Signals - LSTM

- LSTM = Long Short Term Memory (Networks)
  - Signal, text, time-series data
  - Use previous data to predict new information
- I live in France. I speak ___________.



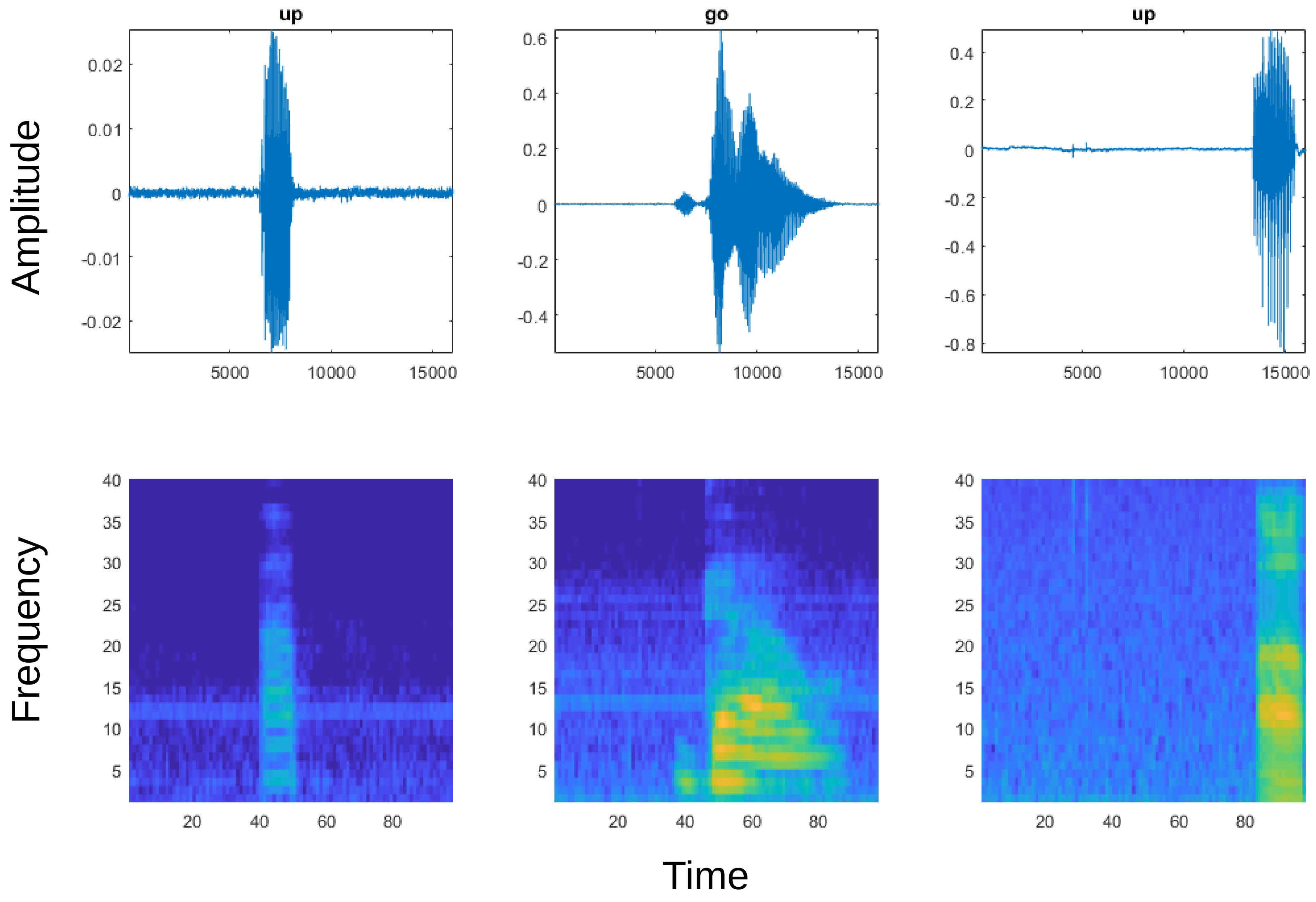
1. Create Datastore

- Datastore creates reference for data
- Do not have to load in all objects into memory

Name	Date modified
_background_noise_	2/12/2018 9:32 AM
Data	2/12/2018 9:39 AM
go	2/12/2018 9:34 AM
left	2/12/2018 9:35 AM
no	2/12/2018 9:36 AM
off	2/12/2018 9:37 AM
on	2/12/2018 9:38 AM
right	2/12/2018 9:31 AM
up	2/12/2018 9:31 AM
yes	2/12/2018 9:32 AM

```
datafolder = fullfile(tempdir,'speech_commands_v0.01');

addpath(fullfile(matlabroot,'toolbox','audio','audiodemos'))
ads = audioexample.Datastore(datafolder, ...
    'IncludeSubfolders',true, ...
    'FileExtensions','.wav', ...
    'LabelSource','foldernames', ...
    'ReadMethod','File')
```

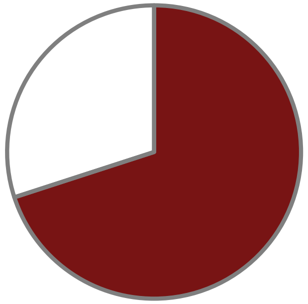
2. Compute Speech Spectrograms



3. Split datastores

Training

70%



- Trains the model
- Computer “learns” from this data

Validation

15%

- Checks accuracy of model during training

Test

15%

- Tests model accuracy
- Not used until validation accuracy is good

4. Define Architecture and Parameters

```matlab
layers = [
    imageInputLayer(imageSize)

    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2,'Padding',[0,1])

    dropoutLayer(dropoutProb)
    convolution2dLayer(3,64,'Padding','same')
    batchNormalizationLayer
    reluLayer

    dropoutLayer(dropoutProb)
```

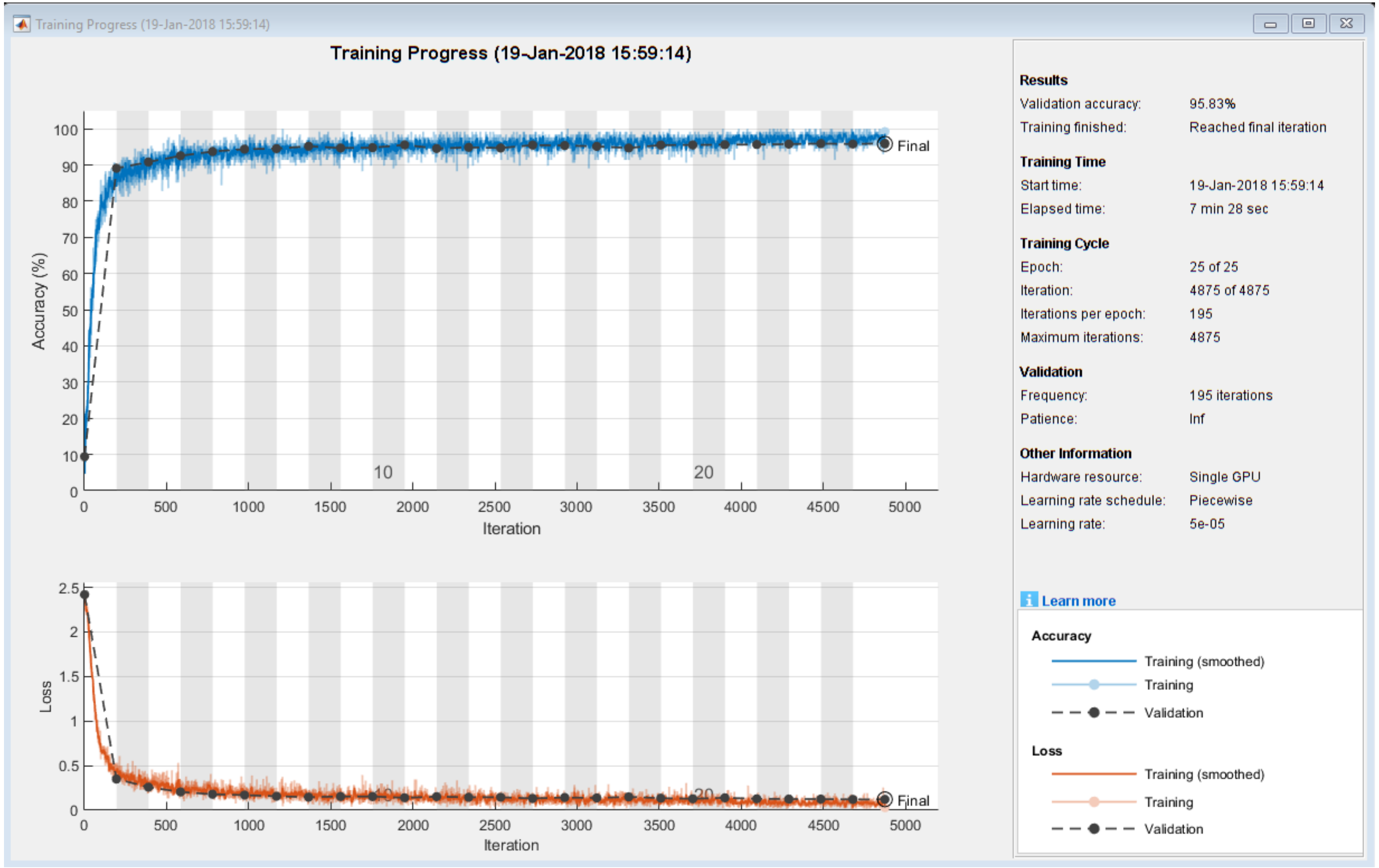
```matlab
    convolution2dLayer(3,64,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2,'Padding',[0,1])

    dropoutLayer(dropoutProb)
    convolution2dLayer(3,64,'Padding','same')
    batchNormalizationLayer
    reluLayer

    dropoutLayer(dropoutProb)
    convolution2dLayer(3,64,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer([1 13])

    fullyConnectedLayer(numClasses)
    softmaxLayer
    weightedCrossEntropyLayer(classNames,classWeights)];
```

Neural Network Architecture

```matlab
miniBatchSize = 128;
validationFrequency = floor(numel(YTrain)/miniBatchSize);
options = trainingOptions('adam', ...
    'InitialLearnRate',5e-4, ...
    'MaxEpochs',25, ...
    'MiniBatchSize',miniBatchSize, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress', ...
    'Verbose',false, ...
    'ValidationData',{XValidation,YValidation}, ...
    'ValidationFrequency',validationFrequency, ...
    'ValidationPatience',Inf, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.1, ...
    'LearnRateDropPeriod',20);
```

Model Parameters

5. Train Network

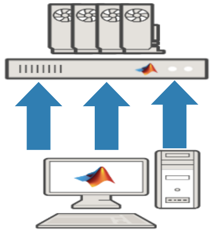


Deep Learning on CPU, GPU, Multi-GPU and Clusters

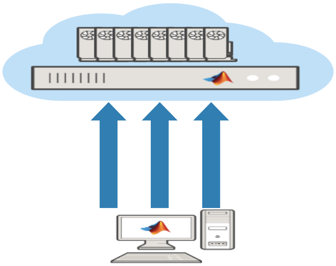
Single CPU

Single CPU
Single GPU

Single CPU, Multiple GPUs



On-prem server with GPUs



Cloud GPUs (AWS)

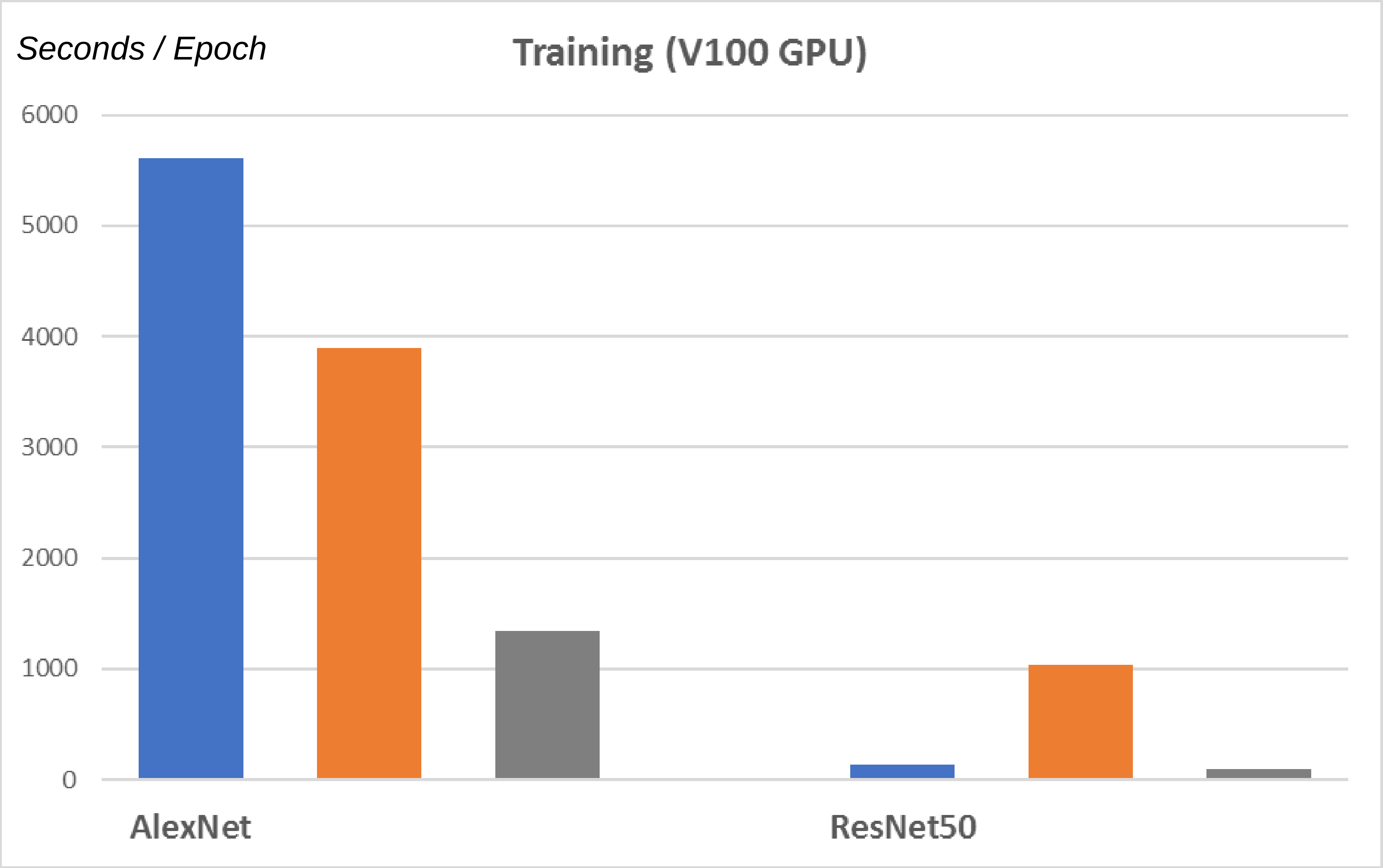
How to target?

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
'ExecutionEnvironment', 'auto' );
```

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
'ExecutionEnvironment', 'multi-gpu' );
```

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
'ExecutionEnvironment', 'parallel' );
```

Training Performance



TensorFlow
MATLAB
MXNet
Batch size 32

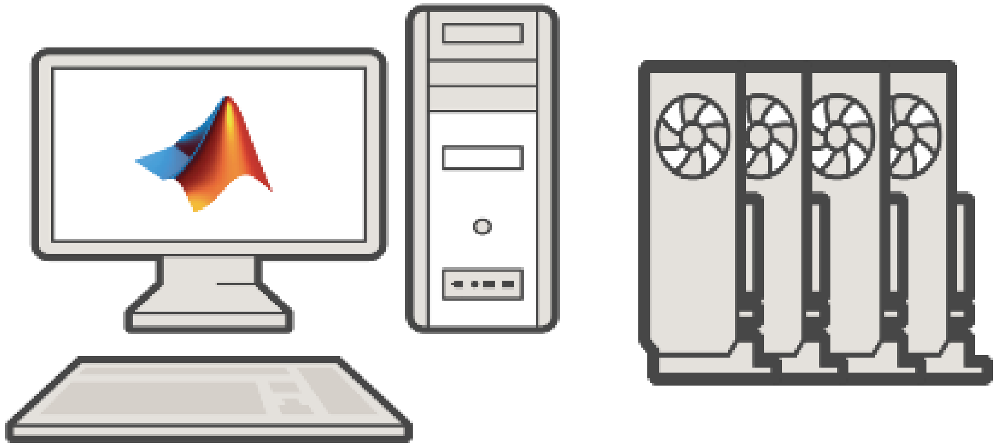
Training is an Iterative Process

```
miniBatchSize = 128;
validationFrequency = floor(numel(YTrain)/miniBatchSize);
options = trainingOptions('adam', ...
    'InitialLearnRate',5e-4, ...
    'MaxEpochs',25, ...
    'MiniBatchSize',miniBatchSize, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress', ...
    'Verbose',false, ...
    'ValidationData',{XValidation,YValidation}, ...
    'ValidationFrequency',validationFrequency, ...
    'ValidationPatience',Inf, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.1, ...
    'LearnRateDropPeriod',20);
```

Parameters adjusted according to performance

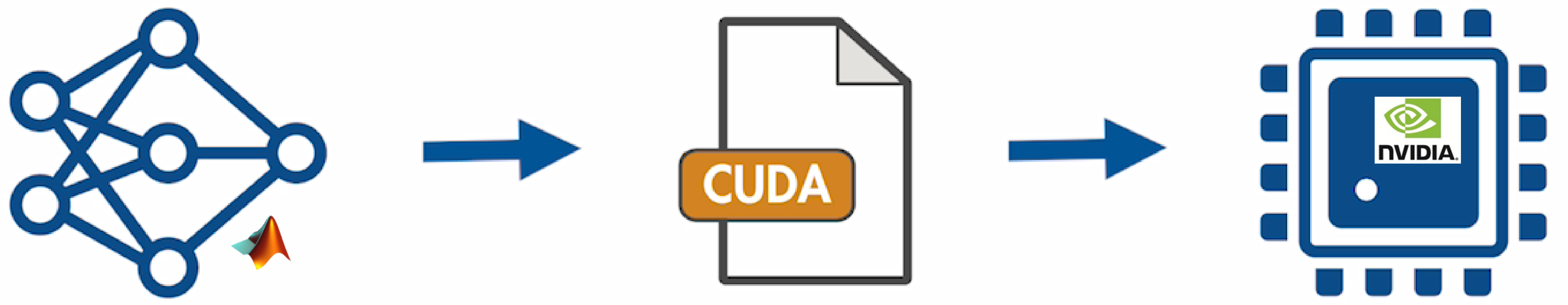
MATLAB is Fast for Deployment

- Target a GPU for optimal performance
- NVIDIA GPUs use CUDA code
- We only have MATLAB code. Can we translate this?



GPU Coder

- Automatically generates **CUDA** Code from MATLAB Code
  - can be used on NVIDIA GPUs



- CUDA extends C/C++ code with constructs for parallel computing

GPU Coder for Deployment

MATLAB

GPU Coder

Accelerated implementation of parallel algorithms on GPUs & CPUs

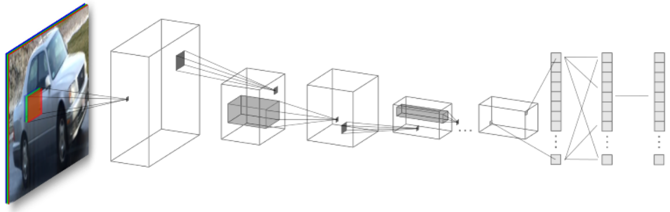
Intel MKL-DNN Library

NVIDIA CUDA C/C++

ARM Compute Library

Deep Neural Networks

Deep Learning, machine learning



5x faster than TensorFlow
2x faster than MXNet

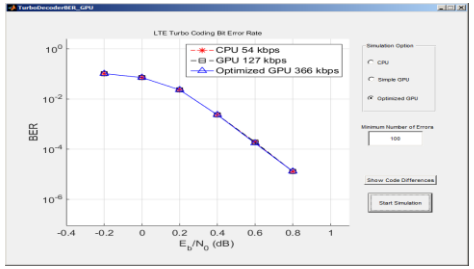
Image Processing and Computer Vision

Image filtering, feature detection/extraction

60x faster than CPUs for stereo disparity

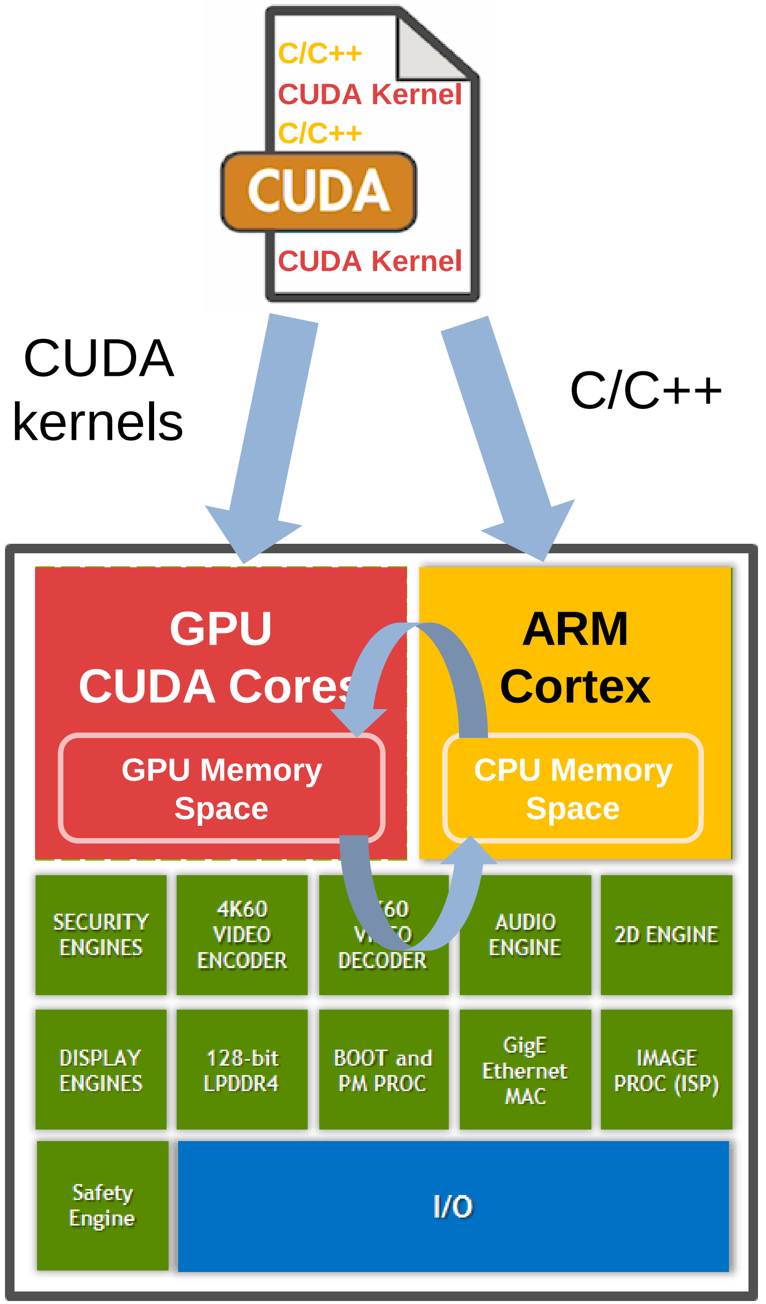
Signal Processing and Communications

FFT, filtering, cross correlation,



20x faster than CPUs for FFTs

GPUs and CUDA



Challenges of Programming in CUDA for GPUs

- Learning to program in CUDA
  - Need to rewrite algorithms for parallel processing paradigm
- Creating CUDA kernels
  - Need to analyze algorithms to create CUDA kernels that maximize parallel processing
- Allocating memory
  - Need to deal with memory allocation on both CPU and GPU memory spaces
- Minimizing data transfers
  - Need to minimize while ensuring required data transfers are done at the appropriate parts of your algorithm

GPU Coder Helps You Deploy to GPUs Faster

MATLAB

GPU Coder

- CUDA Kernel creation
  - Library function mapping
  - Loop optimizations
  - Dependence analysis
- Memory allocation
  - Data locality analysis
  - GPU memory allocation
- Data transfer minimization
  - Data-dependence analysis
  - Dynamic memcpy reduction

NVIDIA CUDA C/C++

GPU Coder Generates CUDA from MATLAB: saxpy

Scalarized MATLAB

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

GPU Coder

Vectorized MATLAB

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

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]);
  }
}
```

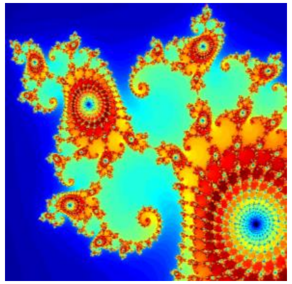
Loops and matrix operations are directly compiled into kernels

Generated CUDA Optimized for Memory Performance

Kernel data allocation is automatically optimized

```matlab
z = z0;
for n = 0:maxIterations
    z = z.*z + z0;
    inside = abs( z )<=2;
    count = count + inside;
end
count = log( count );
```

GPU Coder



Mandelbrot space

CUDA kernel for GPU parallelization

```c
static __global__ __launch_bounds__(512, 1) void kernel3(creal_T *z0, real_T
  *count, creal_T *z)
{
  real_T z_im;
  real_T y[1000000];
  int32_T threadIdX;
  threadIdX = (int32_T)(blockDim.x * blockIdx.x + threadIdx.x);
  if (!(threadIdX >= 1000000)) {
    z_im = z[threadIdX].re * z[threadIdX].im + z[threadIdX].im * z[threadIdX].re;
    z[threadIdX].re = (z[threadIdX].re * z[threadIdX].re - z[threadIdX].im *
                       z[threadIdX].im) + z0[threadIdX].re;
    z[threadIdX].im = z_im + z0[threadIdX].im;
    y[threadIdX] = hypot(z[threadIdX].re, z[threadIdX].im);
    count[threadIdX] += (real_T)(y[threadIdX] <= 2.0);
  }
}
```

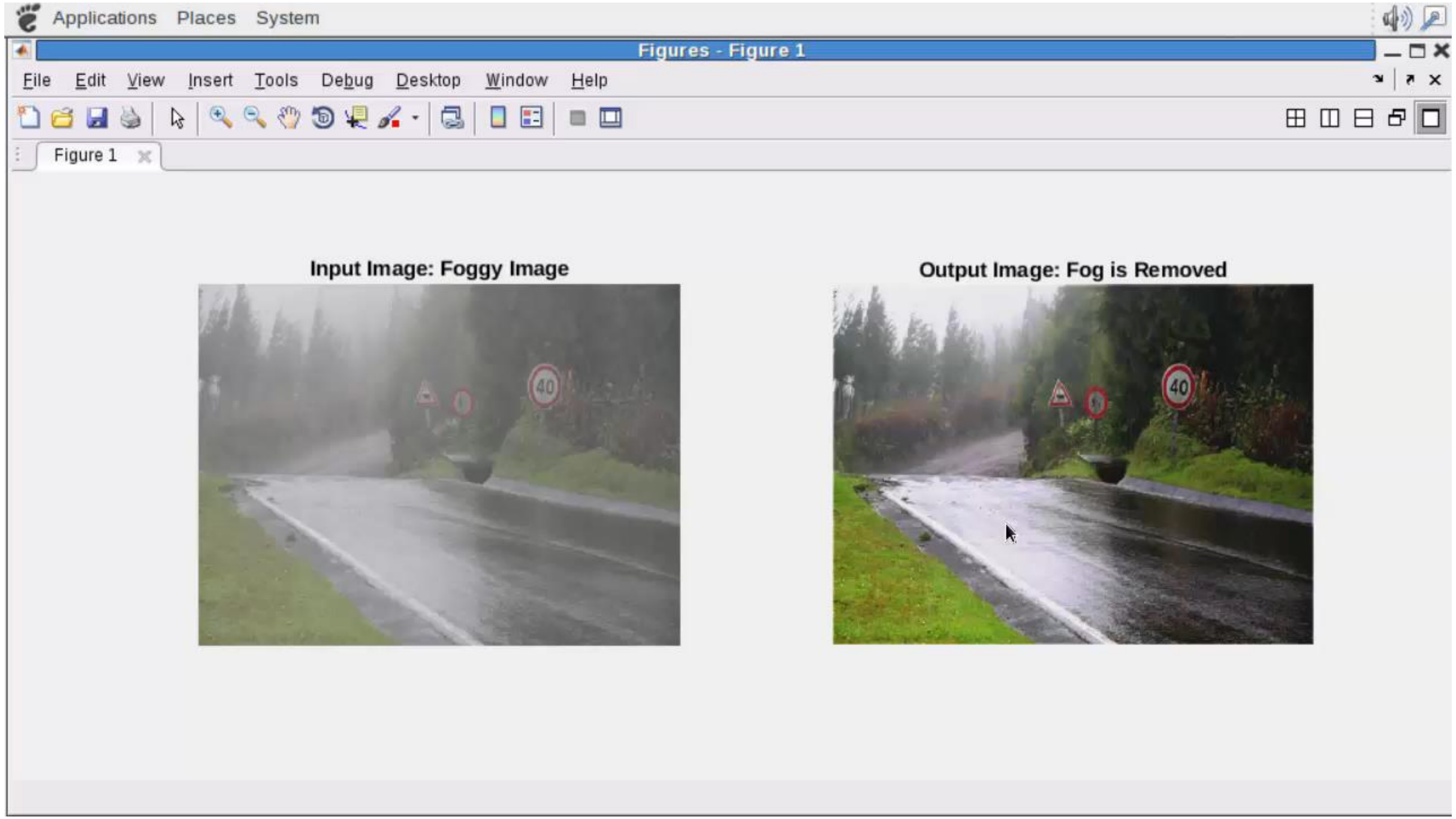
CUDA

...
...

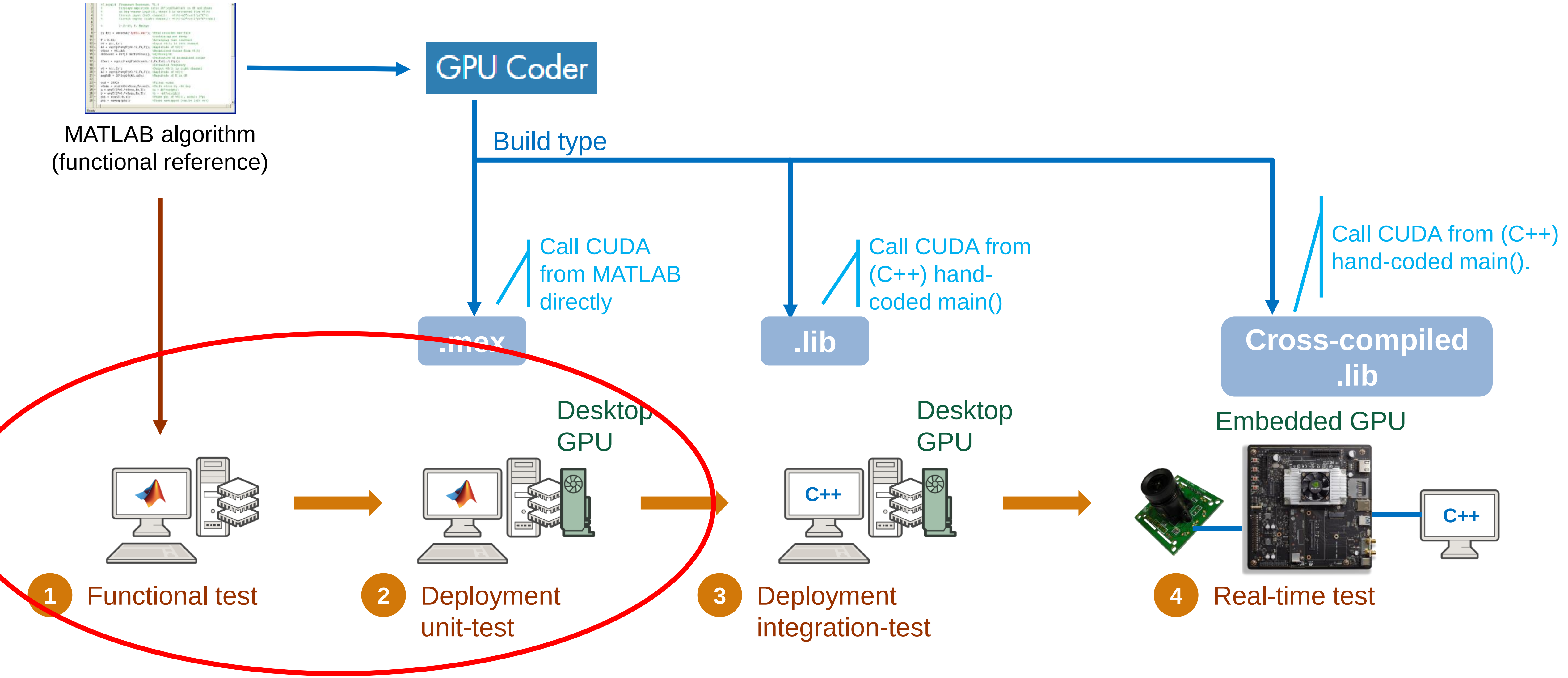
```c
cudaMalloc(&gpu_xGrid, 8000000U);
cudaMalloc(&gpu_yGrid, 8000000U);

/*  mandelbrot computation */
cudaMemcpy(gpu_yGrid, yGrid, 8000000U, cudaMemcpyHostToDevice);
cudaMemcpy(gpu_xGrid, xGrid, 8000000U, cudaMemcpyHostToDevice);
kernel1<<<dim3(1954U, 1U, 1U), dim3(512U, 1U, 1U)>>>(gpu_yGrid, gpu_xGrid,
  gpu_z, gpu_count, gpu_z0);
for (n = 0; n < (int32_T)(maxIterations + 1.0); n++) {
  kernel3<<<dim3(1954U, 1U, 1U), dim3(512U, 1U, 1U)>>>(gpu_z0, gpu_count,
    gpu_z);
}

kernel2<<<dim3(1954U, 1U, 1U), dim3(512U, 1U, 1U)>>>(gpu_count);
cudaMemcpy(count, gpu_count, 8000000U, cudaMemcpyDeviceToHost);
cudaFree(gpu_yGrid);
```

...
...

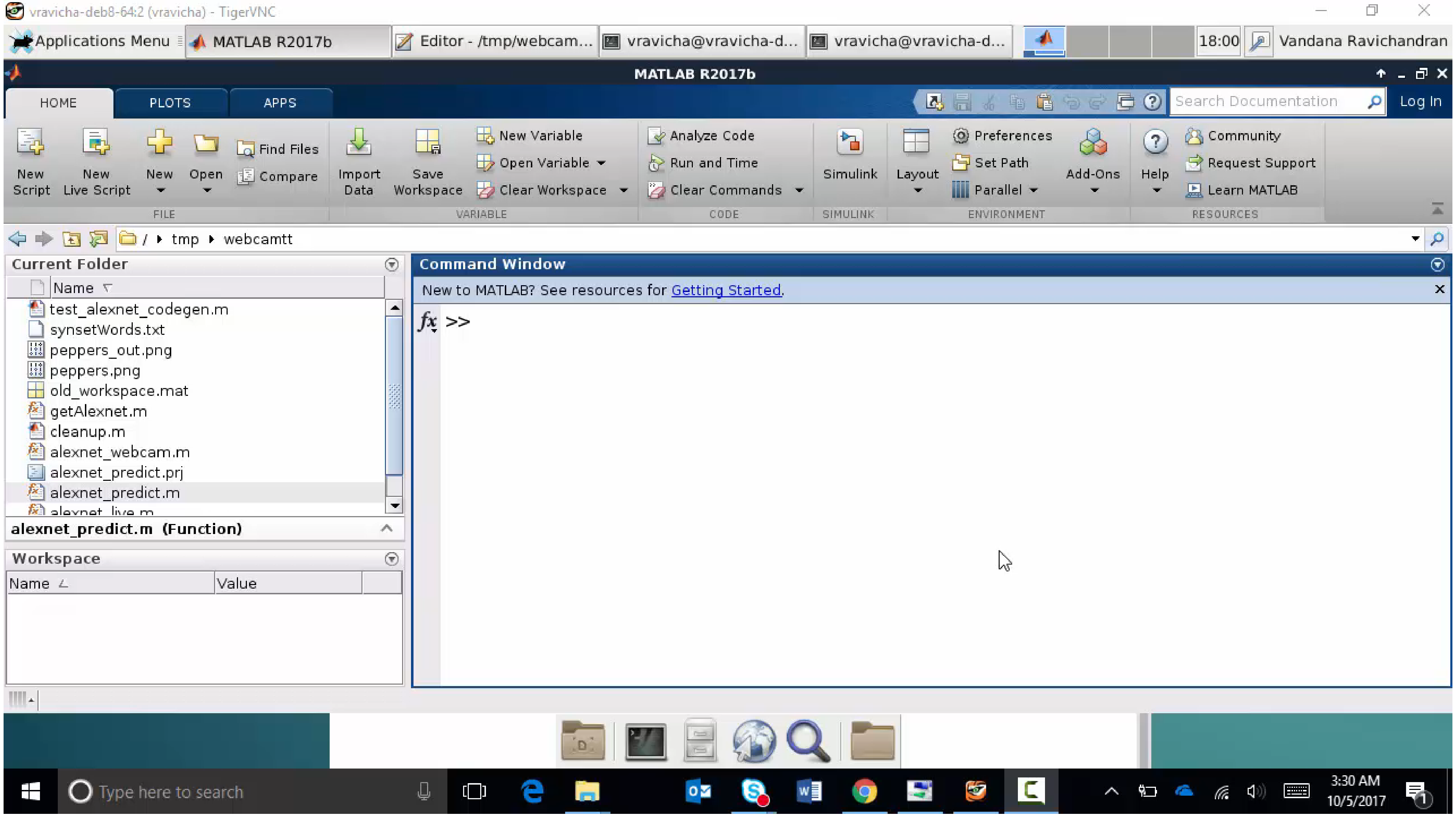
Example: Fog Rectification



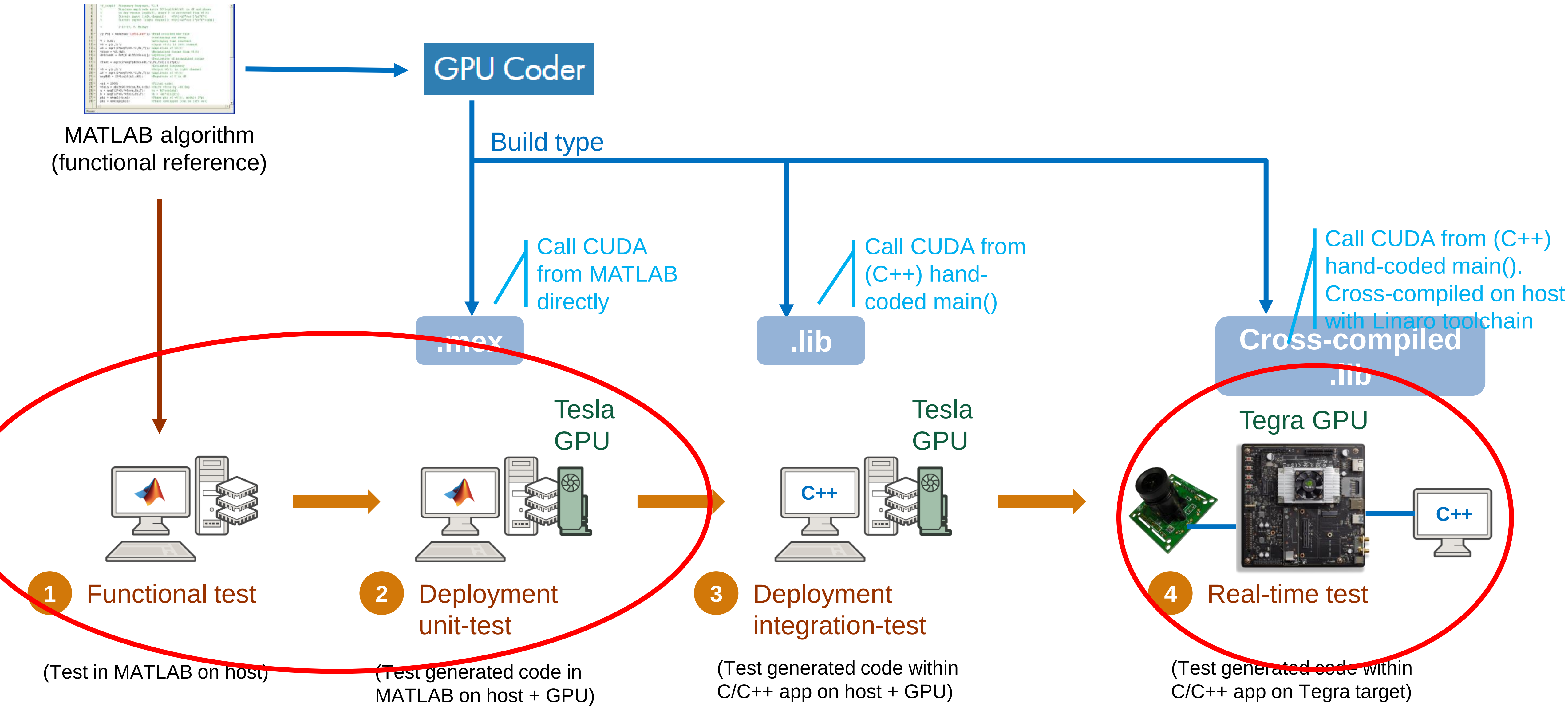
Algorithm Design to Embedded Deployment Workflow



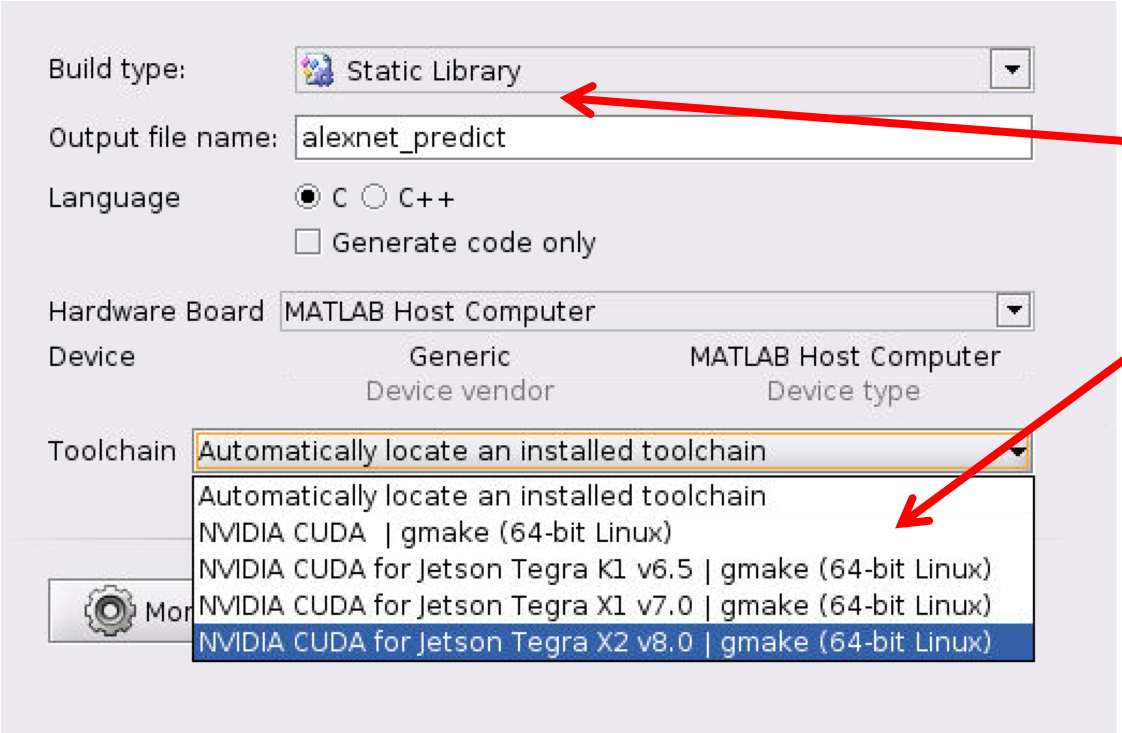
Demo: Alexnet Deployment with ‘mex’ Code Generation



Algorithm Design to Embedded Deployment on Tegra GPU



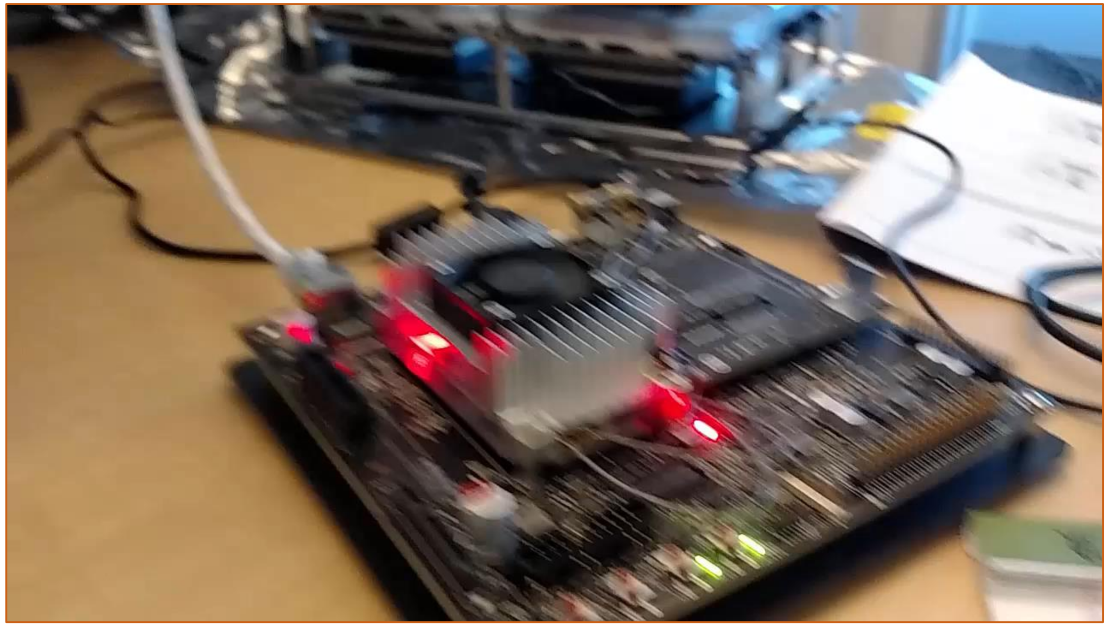
Alexnet Deployment to Tegra: Cross-Compiled with ‘lib’



Two small changes

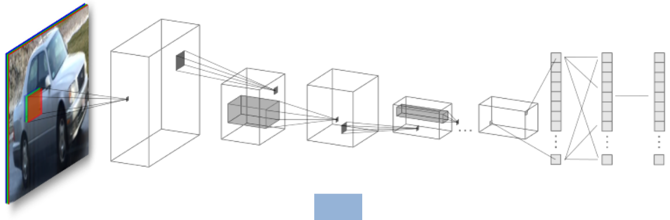
1. Change build-type to ‘lib’

2. Select cross-compile toolchain

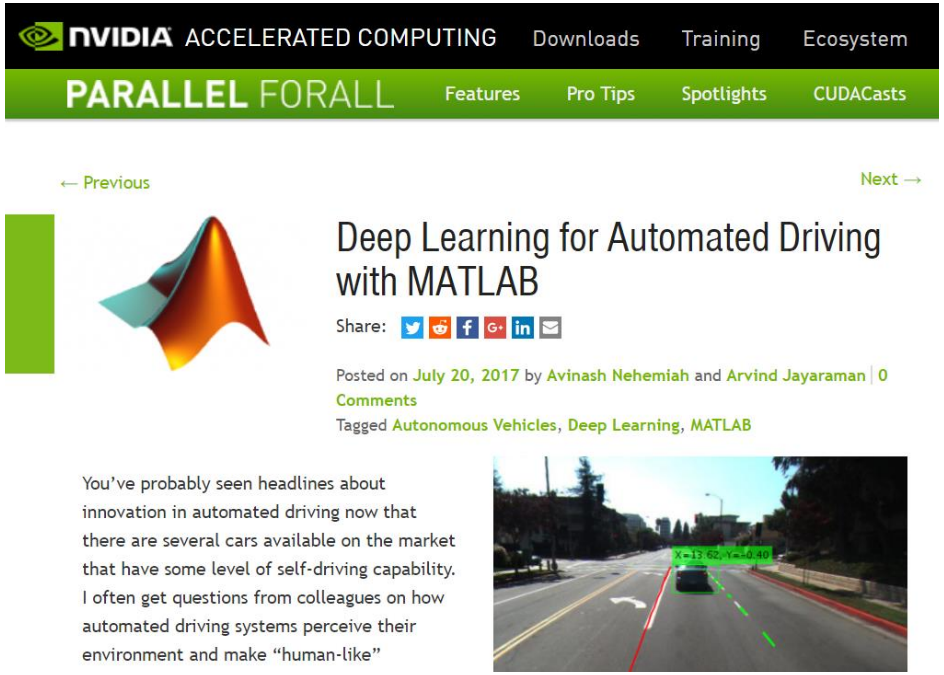


End-to-End Application: Lane Detection

Alexnet



Transfer Learning



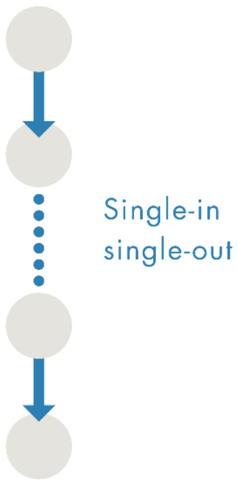
Output of CNN is lane parabola coefficients according to: $y = ax^2 + bx + c$

Image → **Lane detection CNN** → Left lane coefficients / Right lane coefficients → **Post-processing (find left/right lane points)** → Image with marked lanes

GPU coder generates code for whole application

Deep Learning Network Support (with Neural Network Toolbox)

SeriesNetwork



GPU Coder: R2017b

Networks:
- MNist
- Alexnet
- YOLO
- VGG
- Lane detection
- Pedestrian detection

DAGNetwork

GPU Coder: R2018a

Networks:
- GoogLeNet
- ResNet
- SegNet
- DeconvNet

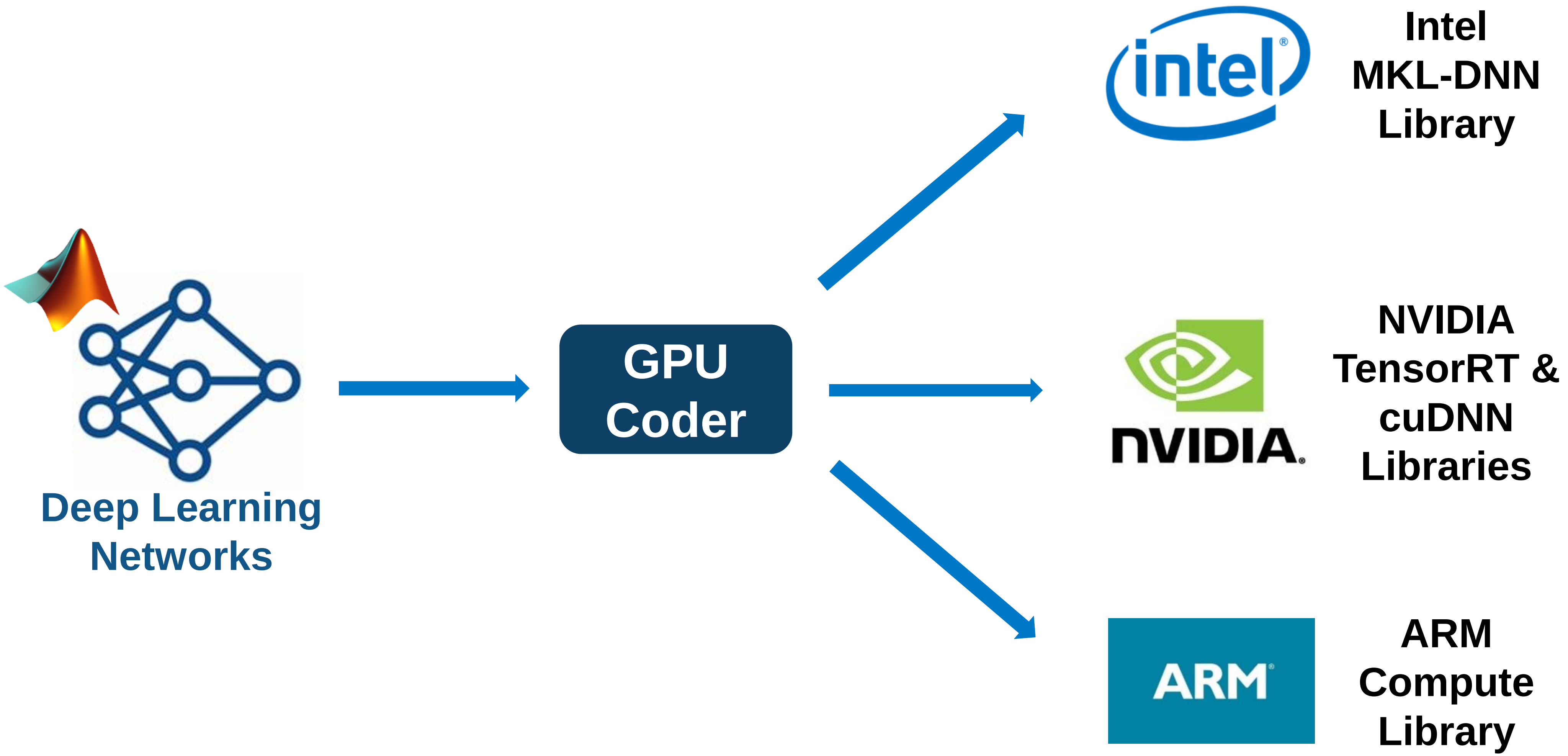
GoogLeNet, ResNet: Object detection
SegNet, DeconvNet: Semantic segmentation

Semantic Segmentation

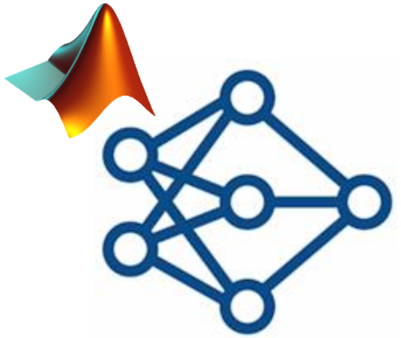
Running in MATLAB

Generated Code from GPU Coder

Deploying to CPUs



Deploying to CPUs

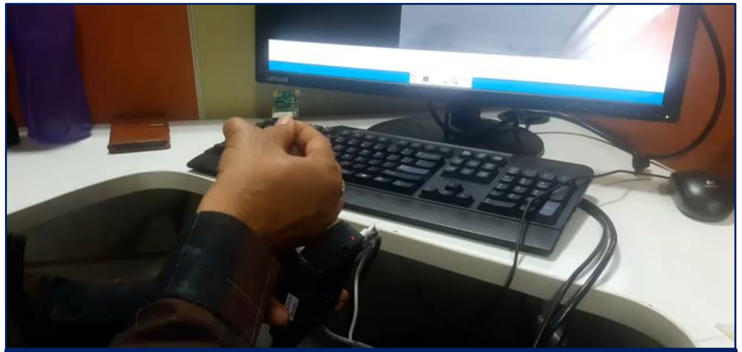


Deep Learning Networks

GPU Coder

Desktop CPU

NVIDIA TensorRT & cuDNN Libraries



Raspberry Pi board

How Good is Generated Code Performance

- Performance of image processing and computer vision

- Performance of CNN inference (Alexnet) on Titan XP GPU

- Performance of CNN inference (Alexnet) on Jetson (Tegra) TX2

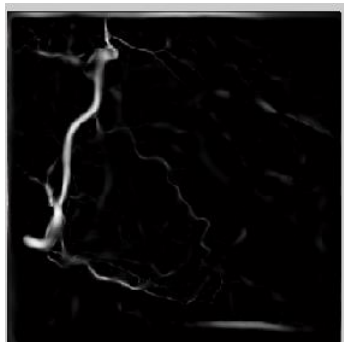
GPU Coder for Image Processing and Computer Vision

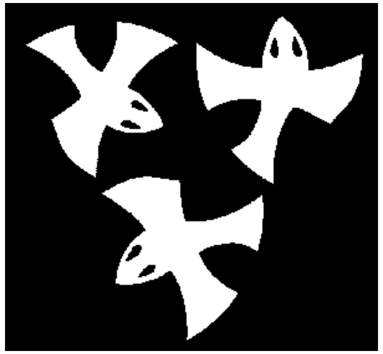
Fog removal

5x speedup

Frangi filter

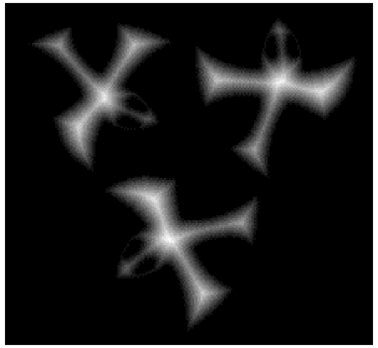
3x speedup





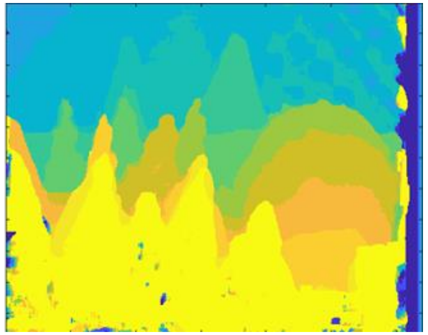
Distance transform

8x speedup



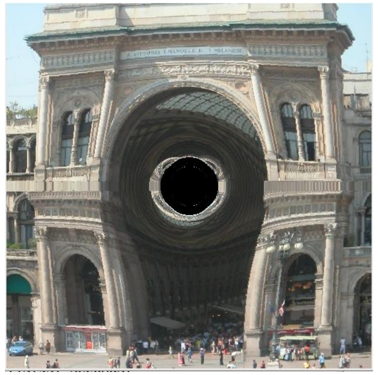
Stereo disparity

50x speedup



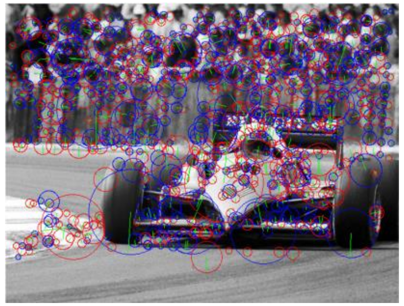
Ray tracing

18x speedup

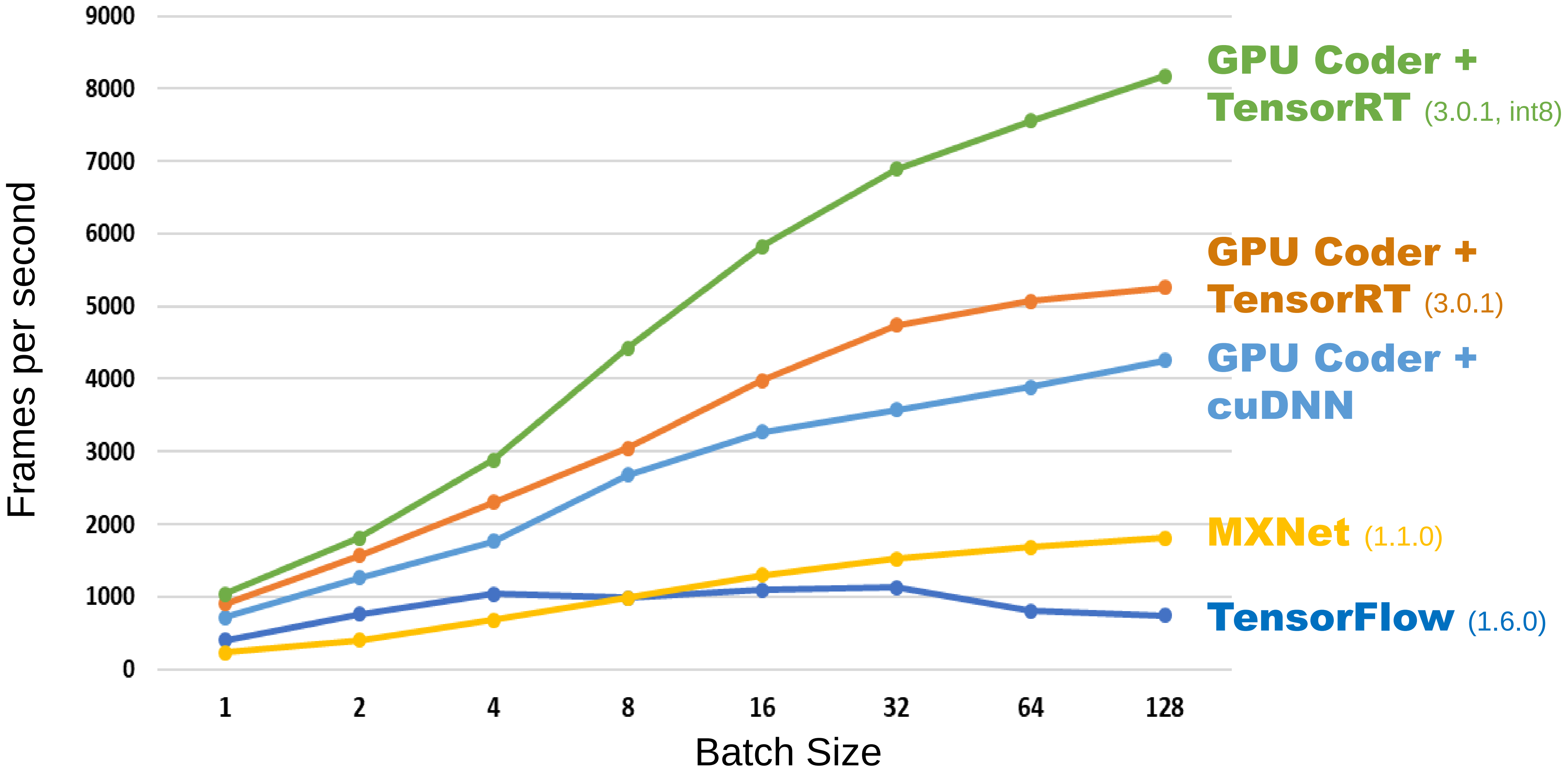


SURF feature extraction

700x speedup



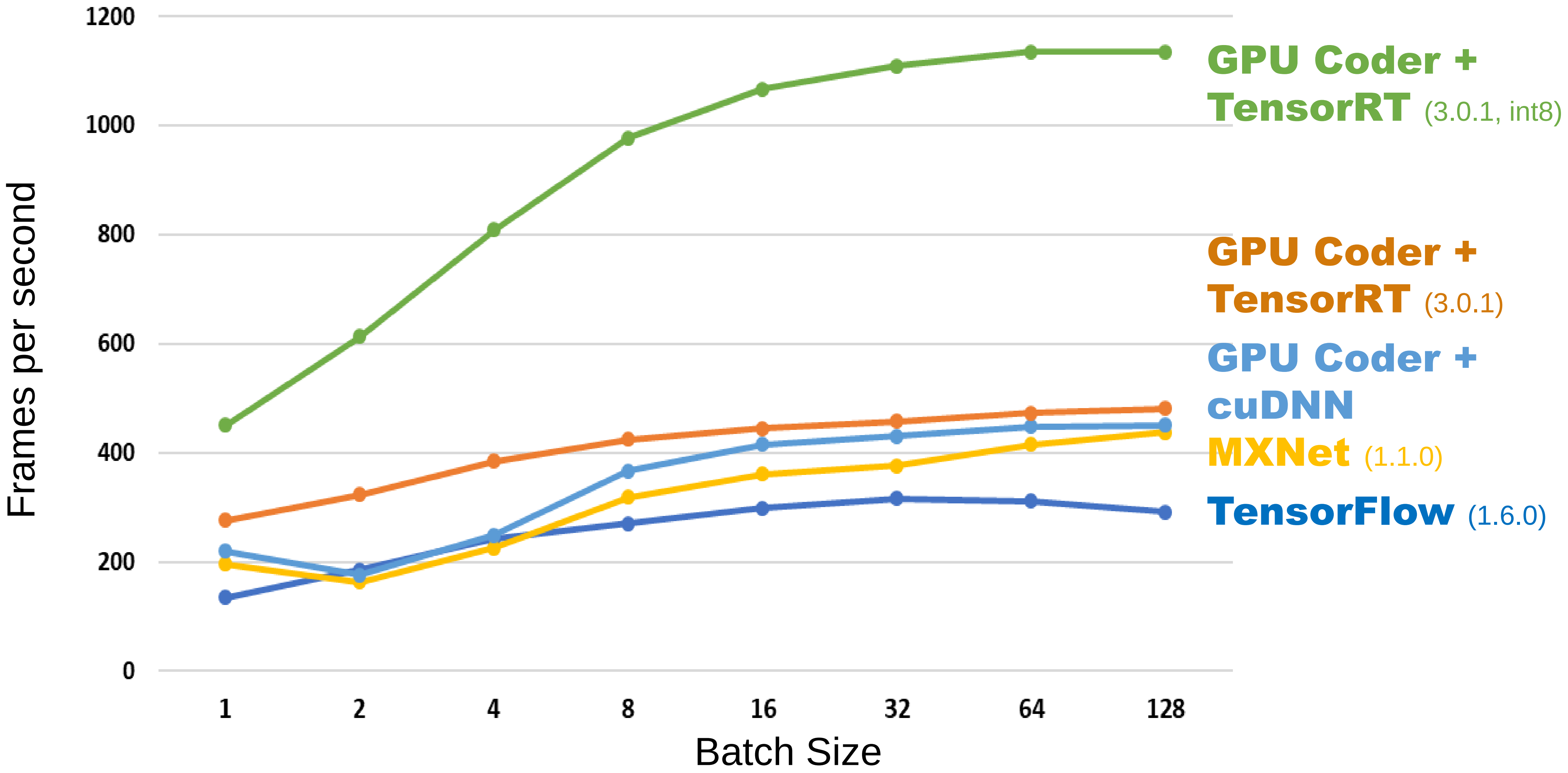
Alexnet Inference on NVIDIA Titan Xp



Testing platform

CPU	Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz
GPU	Pascal Titan Xp
cuDNN	v7

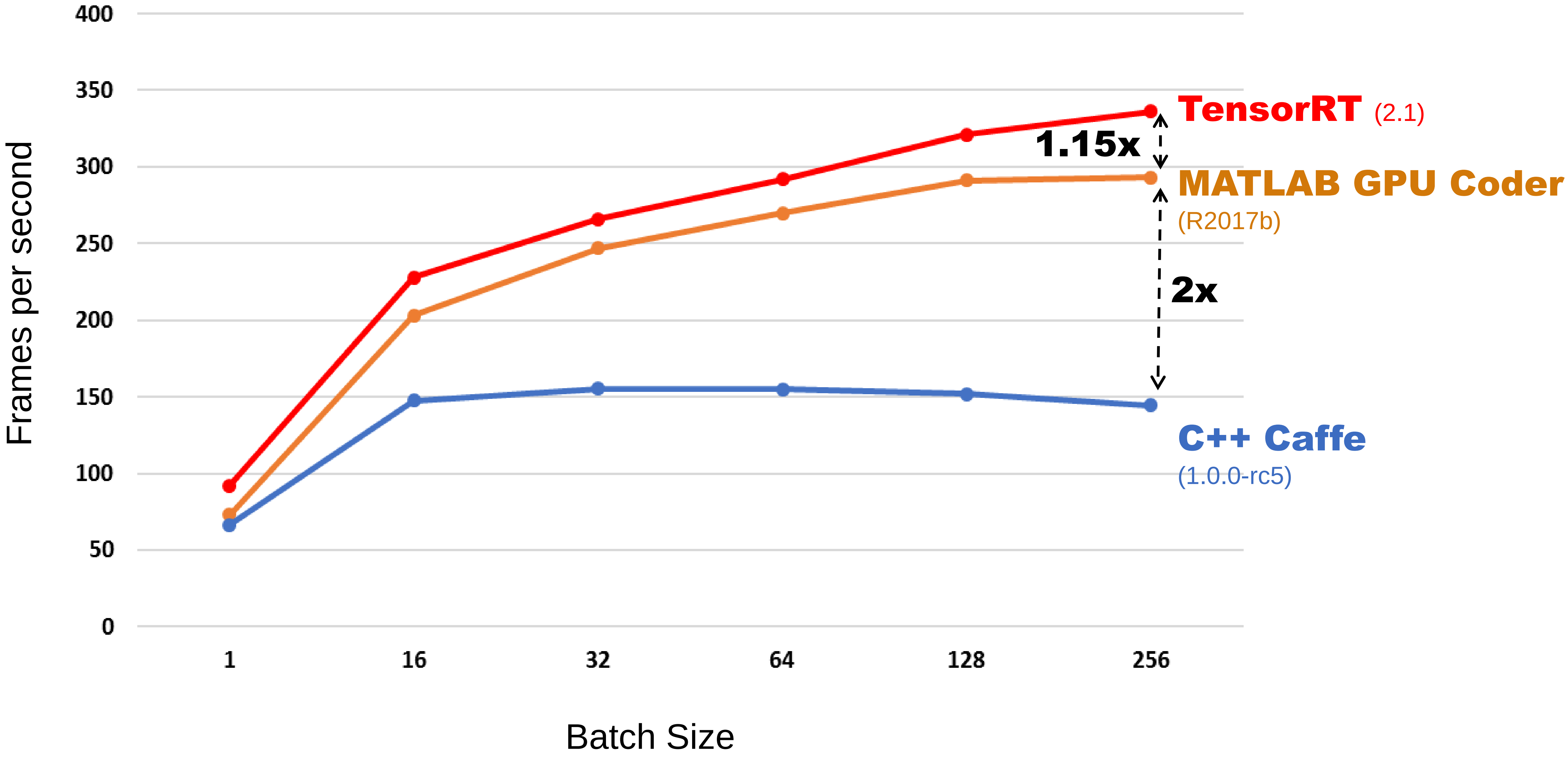
VGG-16 Inference on NVIDIA Titan Xp



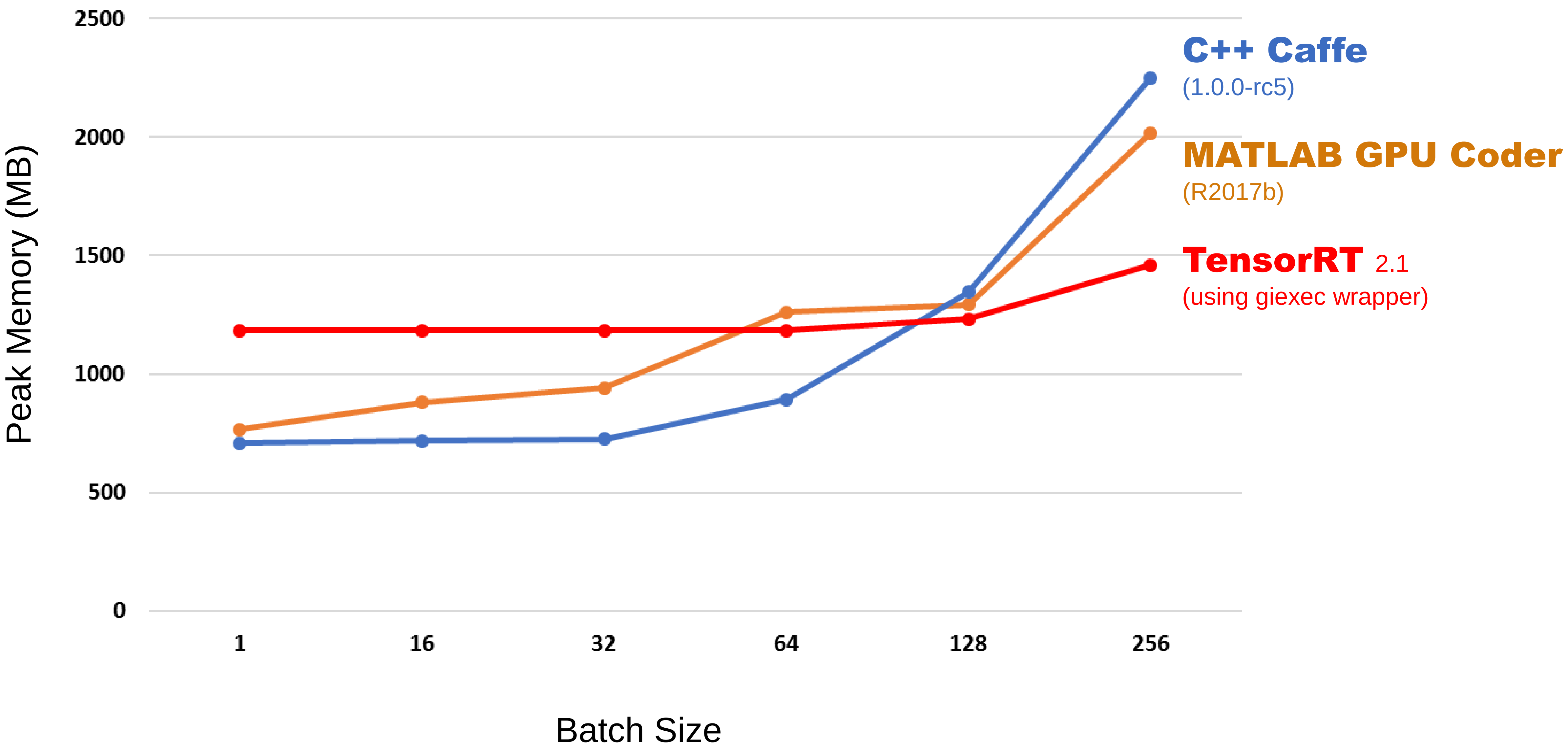
Testing platform

CPU	Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz
GPU	Pascal Titan Xp
cuDNN	v7

Alexnet Inference on Jetson TX2: Frame-Rate Performance



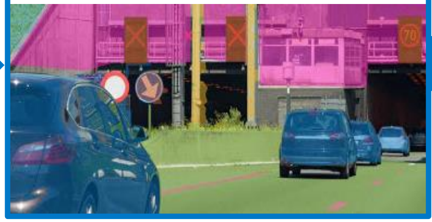
Alexnet Inference on Jetson TX2: Memory Performance



MATLAB Deep Learning Framework

Access Data

Design + Train



Deploy

- **Manage** large image sets
- **Automate** image labeling
- **Easy access** to models

- **Acceleration** with GPU’s
- **Scale** to clusters

- **Automate compilation to GPUs and CPUs using GPU Coder:**
  - **5x faster** than TensorFlow
  - **2x faster** than MXNet

Why MATLAB for Deep Learning?

- MATLAB is Productive
- MATLAB Integrates with Open Source *(Frameworks)*
- MATLAB is Fast *(Performance)*