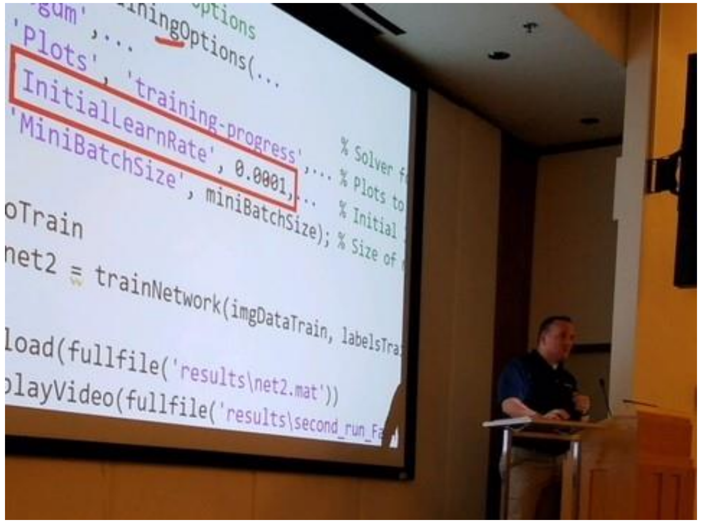
Hands-on Deep Learning Workshop



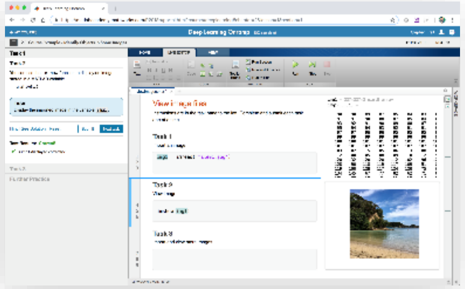
```
gum' ,...   ...ningOptions(...
'Plots', 'training-progress',... % Solver f
'InitialLearnRate', 0.0001,...   % Plots to
'MiniBatchSize', miniBatchSize); % Initial
                                 % Size of
oTrain
net2 = trainNetwork(imgDataTrain, labelsTra
load(fullfile('results\net2.mat'))
playVideo(fullfile('results\second_run_Fa
```

Matthias Sommer

Campus-Wide License at University of Bern

University & lab computers

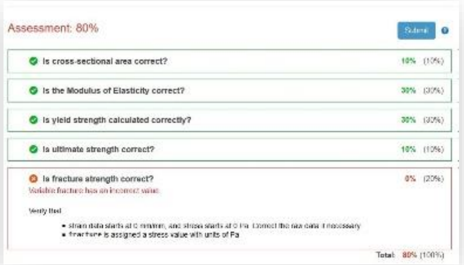
Personal Computers & Mobile Devices



Self-paced online learning

Online access

- License covers all faculty, staff, students and their devices
- Access on campus, in lab and field, and at home
- Get MATLAB via https://ch.mathworks.com/ or IT store of the University of Bern



Assessment: 80%

✓ Is cross-sectional area correct? 10% (10%)

✓ Is the Modulus of Elasticity correct? 30% (30%)

✓ Is yield strength calculated correctly? 30% (30%)

✓ Is ultimate strength correct? 10% (10%)

☒ Is fracture strength correct? 0% (20%)

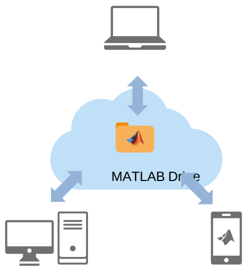
Variable fracture has an incorrect value.

Verify that:

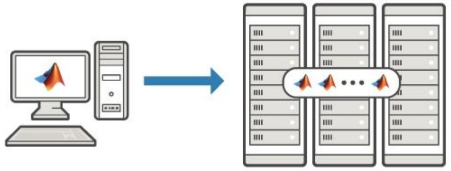
- strain data starts at 0 mm/mm, and stress starts at 0 Pa. Correct the raw data if necessary
- fracture is assigned a stress value with units of Pa

Total: 80% (100%)

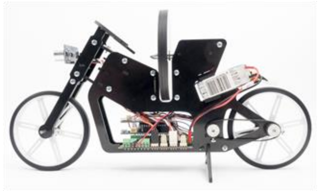
Auto-graded homework



Cloud Storage & Sharing



Clusters & HPC



Low-cost hardware support

Self-Paced, Online Training for MATLAB & Simulink

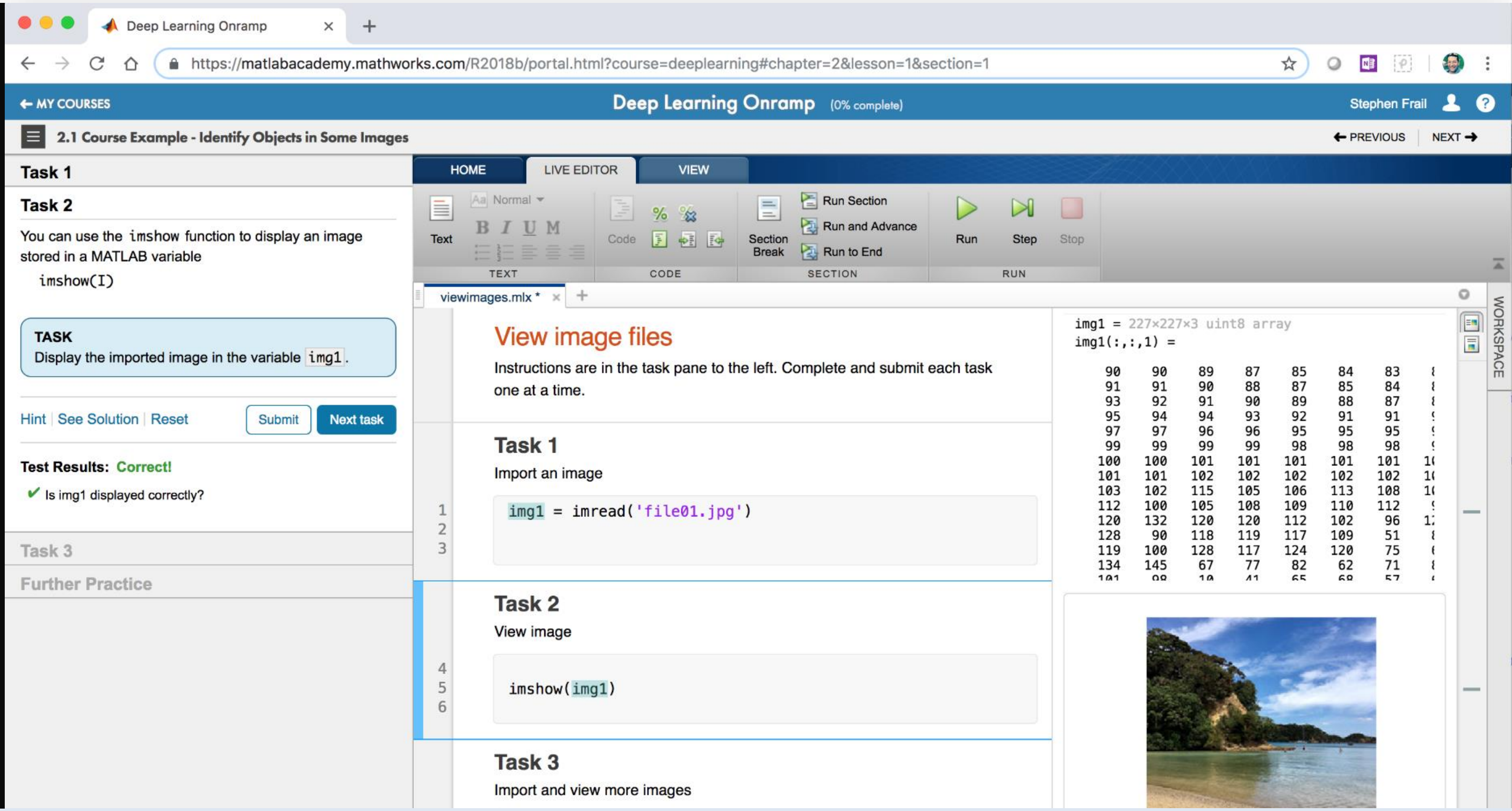
Campus-Wide Online Training

Hands-on MATLAB and Simulink experience

Measurable progress report and completion certificate

Interactive lessons with immediate feedback

24/7 availability



Task 1

Task 2

You can use the `imshow` function to display an image stored in a MATLAB variable

```
imshow(I)
```

TASK
Display the imported image in the variable `img1`.

Hint | See Solution | Reset   Submit   Next task

Test Results: Correct!

✓ Is img1 displayed correctly?

Task 3

Further Practice

View image files

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

Import an image

```
img1 = imread('file01.jpg')
```

```
img1 = 227×227×3 uint8 array
img1(:,:,1) =

    90    90    89    87    85    84    83
    91    91    90    88    87    85    84
    93    92    91    90    89    88    87
    95    94    94    93    92    91    91
    97    97    96    96    95    95    95
    99    99    99    99    98    98    98
   100   100   101   101   101   101   101
   101   101   102   102   102   102   102
   103   102   115   105   106   113   108
   112   100   105   108   109   110   112
   120   132   120   120   112   102    96
   128    90   118   119   117   109    51
   119   100   128   117   124   120    75
   134   145    67    77    82    62    71
```

Task 2

View image

```
imshow(img1)
```

Task 3

Import and view more images

MATLAB Onramp

Simulink Onramp

Deep Learning Onramp

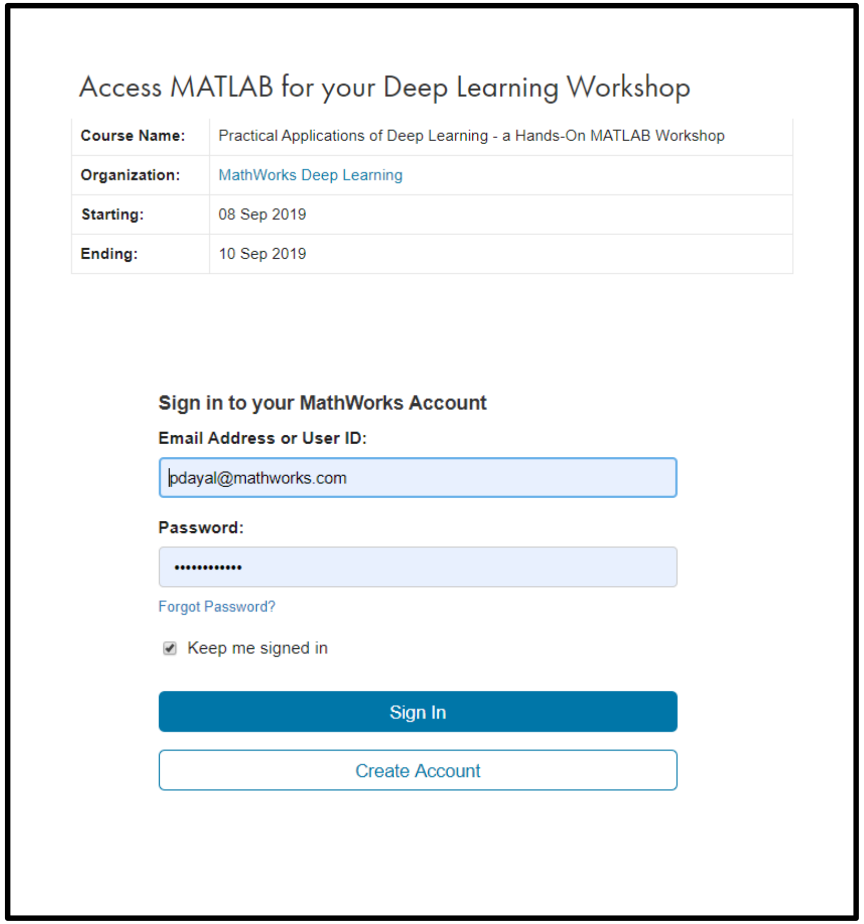
Stateflow Onramp

Machine Learning Onramp

11 hours of FREE content available for everyone

https://matlabacademy.mathworks.com/

Set-Up Instructions



Access MATLAB for your Deep Learning Workshop

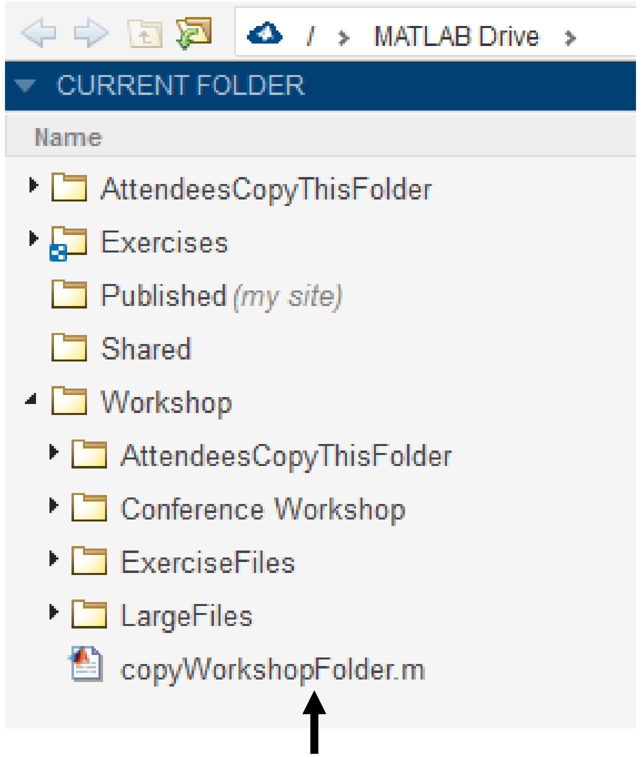
Course Name:	Practical Applications of Deep Learning - a Hands-On MATLAB Workshop
Organization:	MathWorks Deep Learning
Starting:	08 Sep 2019
Ending:	10 Sep 2019

Sign in to your MathWorks Account

Email Address or User ID: pdayal@mathworks.com

Password: •••••••••••

1 - Log into MATLAB Online



- AttendeesCopyThisFolder
- Exercises
- Published *(my site)*
- Shared
- Workshop
  - AttendeesCopyThisFolder
  - Conference Workshop
  - ExerciseFiles
  - LargeFiles
  - copyWorkshopFolder.m

R-click and select ‘run’

2 - Run ‘copyWorkshopFolder.m’

Deep Learning Demo

Image Classification

Agenda

Introduction

Exercise 1: Deep learning in 6 lines of code

Deep Learning Fundamentals

Exercises 2 and 3: Exploring pretrained networks/Classifying handwritten digits

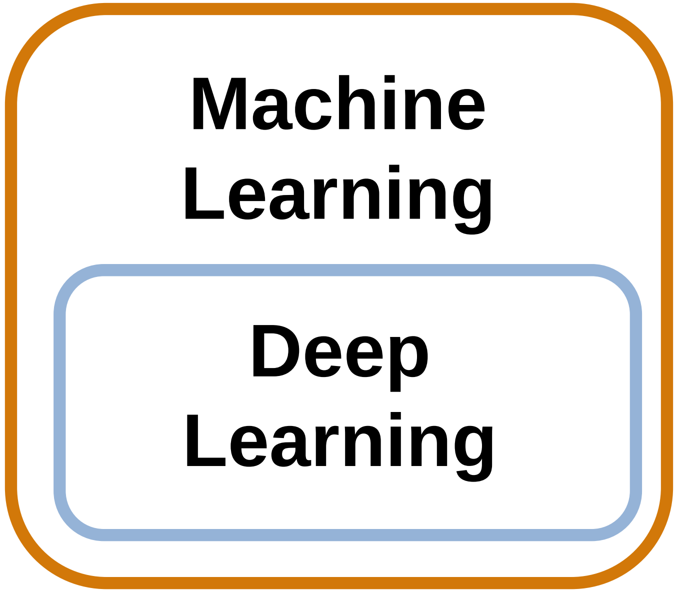
Exercise 4: Transfer Learning – OR – Signal Classification Exercise

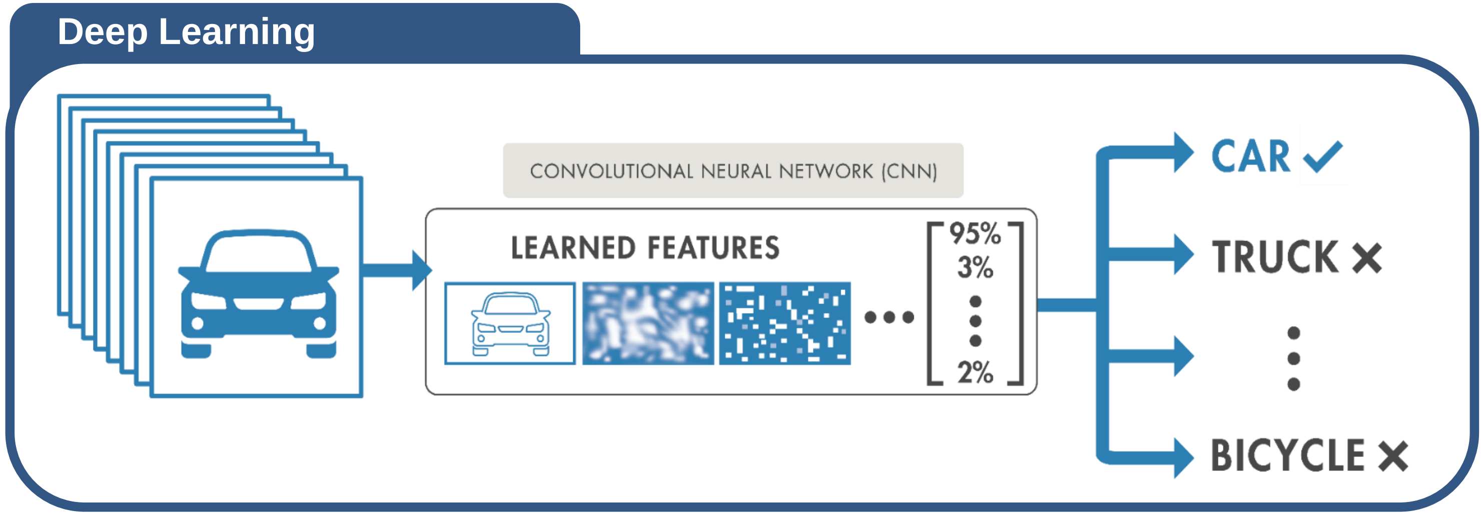
Demo: Deploying Deep Networks– OR – Improving Network Accuracy

Conclusion

What is Deep Learning?

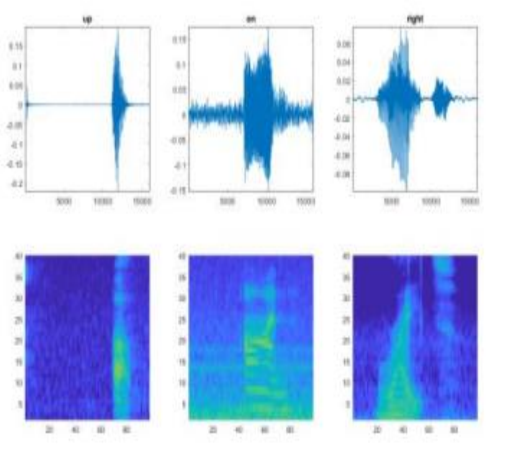
- Subset of machine learning with automatic feature extraction
  - Learns features and tasks directly from data
- Accuracy can surpass traditional ML Algorithms





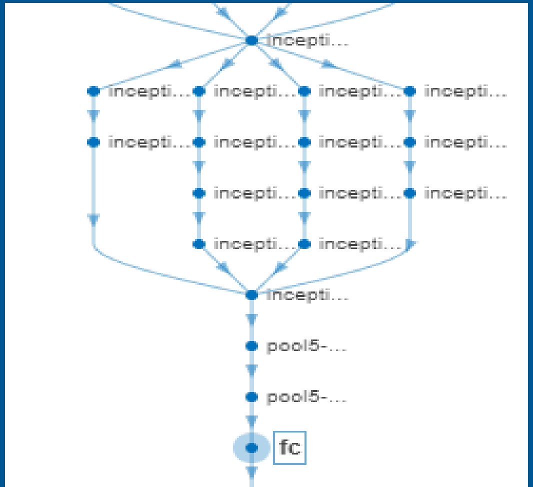
Deep Learning Workflow

PREPARE DATA



The data must be labeled and preprocessed to give accurate results

TRAIN MODEL



Build a neural network that learns from your dataset

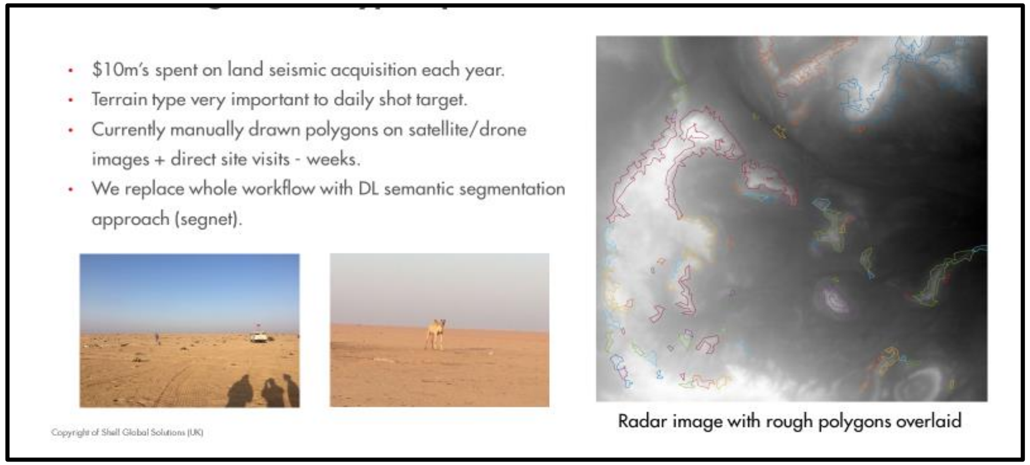
DEPLOY SYSTEM

```
cudaMalloc(&gpu_inputdata, 6183480LL);
cudaMemcpy((void *)gpu_inputdata, (void
c_DeepLearningNetwork_predict_k<<<dim
cudaMemcpy(obj->inputData, gpu_inp
obj->predict();
cudaMemcpy(gpu_out, obj->
d_DeepLearningNetwork_pr
```

Integrate your trained model onto embedded hardware or cloud

Deep Learning Examples

Shell



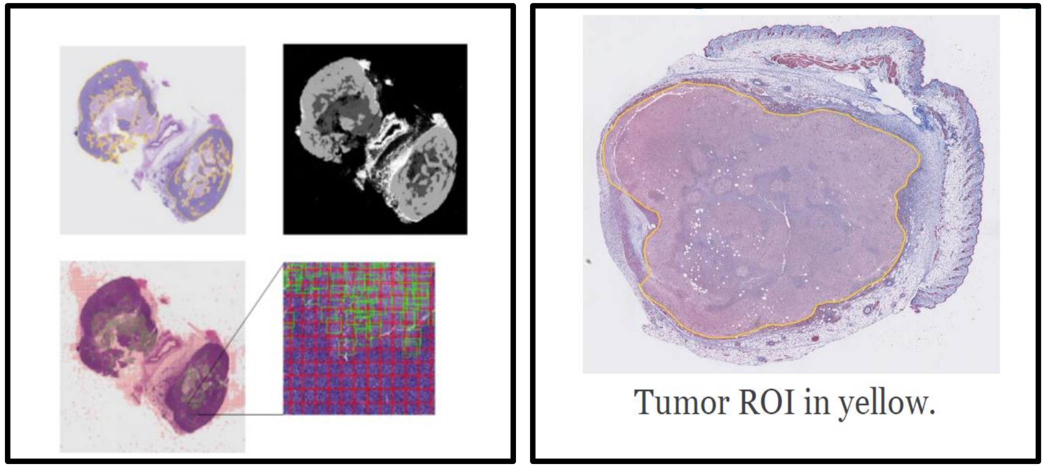
- $10m's spent on land seismic acquisition each year.
- Terrain type very important to daily shot target.
- Currently manually drawn polygons on satellite/drone images + direct site visits - weeks.
- We replace whole workflow with DL semantic segmentation approach (segnet).

Radar image with rough polygons overlaid

Copyright of Shell Global Solutions (UK)

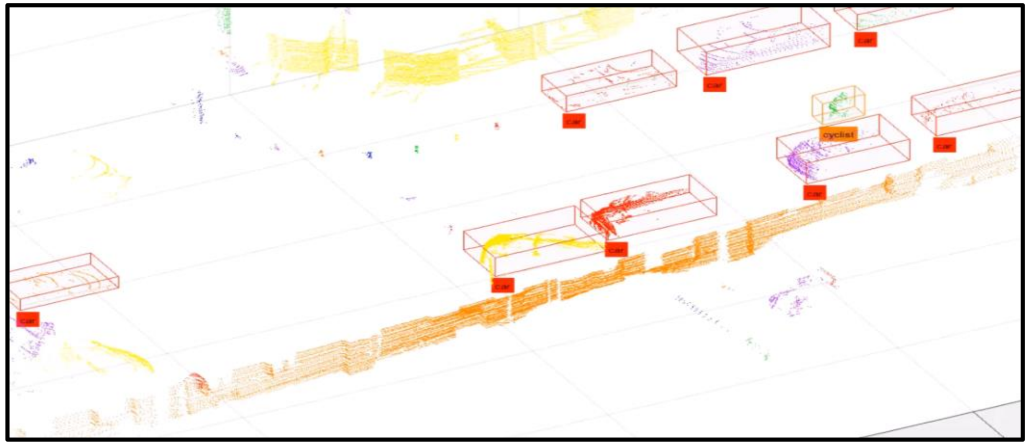
Terrain Recognition with Hyperspectral Data

Genentech



CNNs for Digital Pathology Analysis

Veoneer



LiDAR-Based Sensor Verification

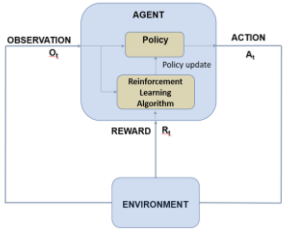
Caterpillar

Equipment Classification

MATLAB’s deep learning workflows were designed for engineers and scientists in many domains

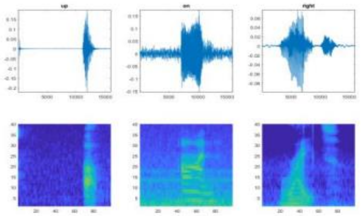
Computer Vision

Image Processing

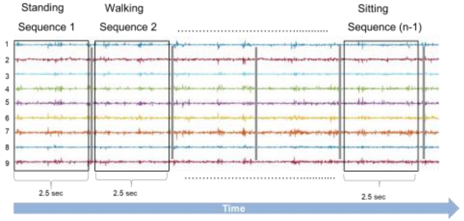


Control Design

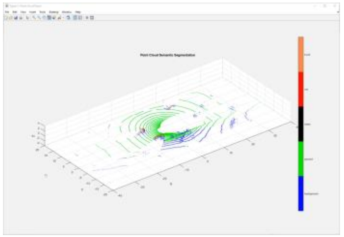
Text Analytics



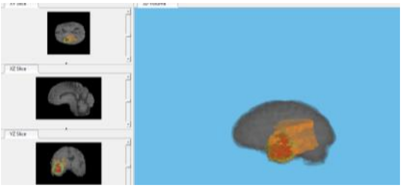
Audio Processing



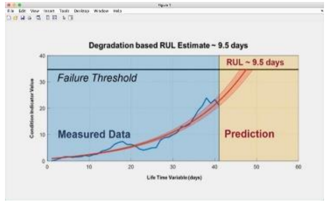
Sensor Data Analysis



Lidar Processing



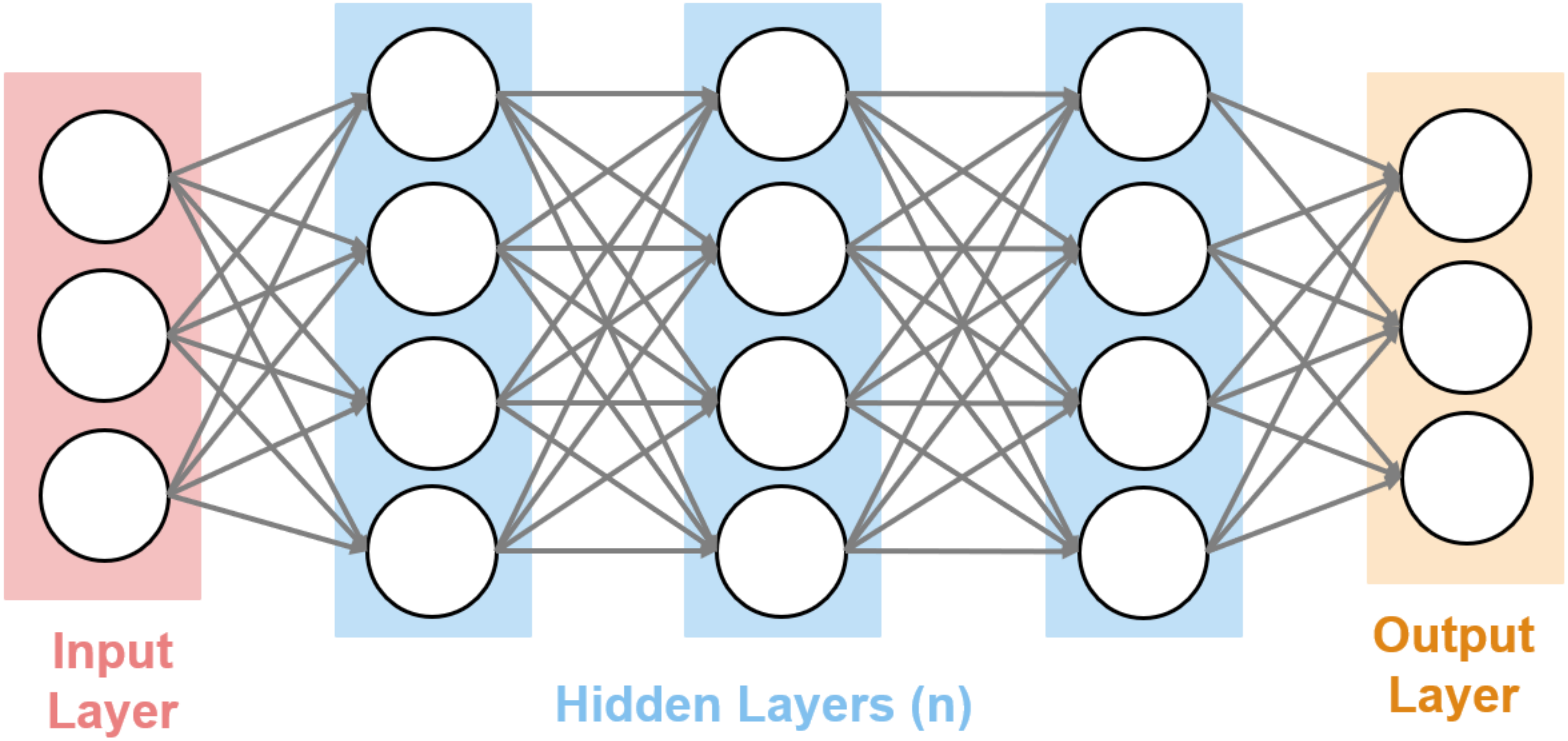
N-D Volumes



Sensor Data Analysis

Deep Learning models are Neural networks (for all data types)

- Deep neural networks have many layers
- Data is passed through the network, and the layer parameters are updated (training)



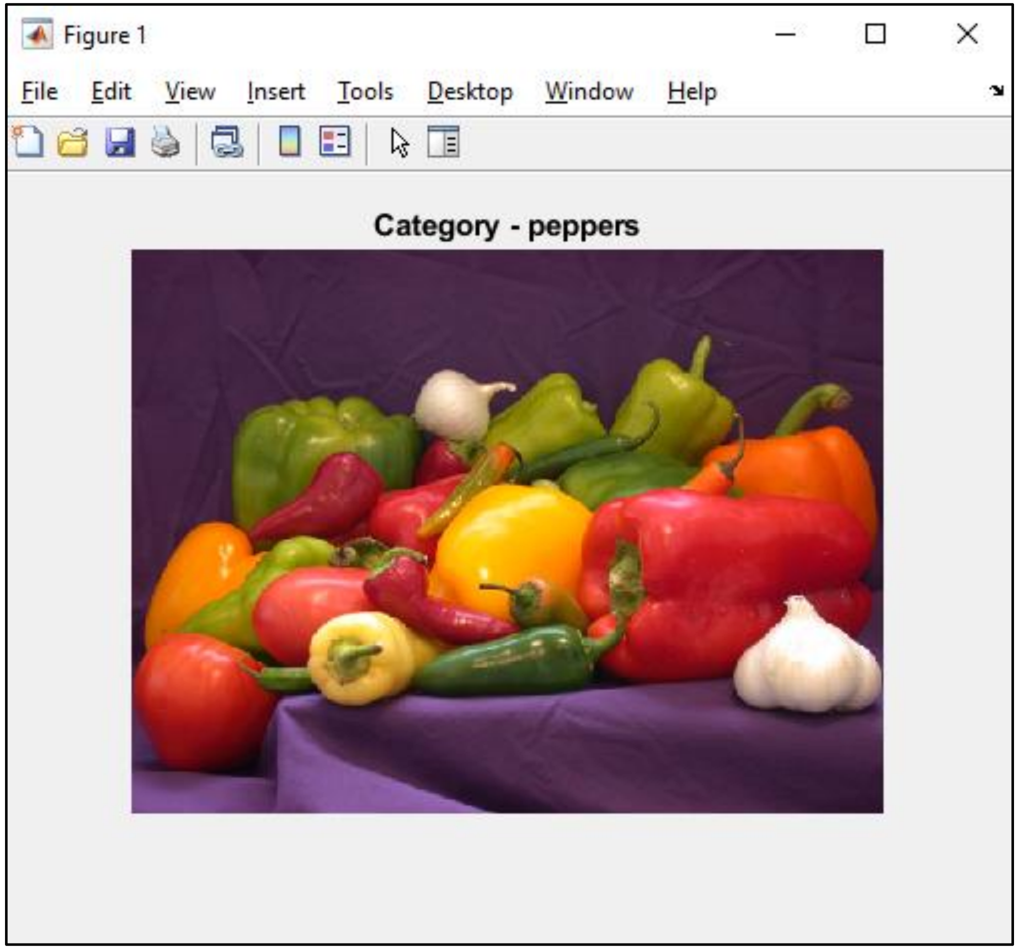
Exercise 1 – Deep Learning in 6 Lines of Code

Purpose:

- Ensure MATLAB Online is running properly
- Use a neural network to classify an image

To Do:

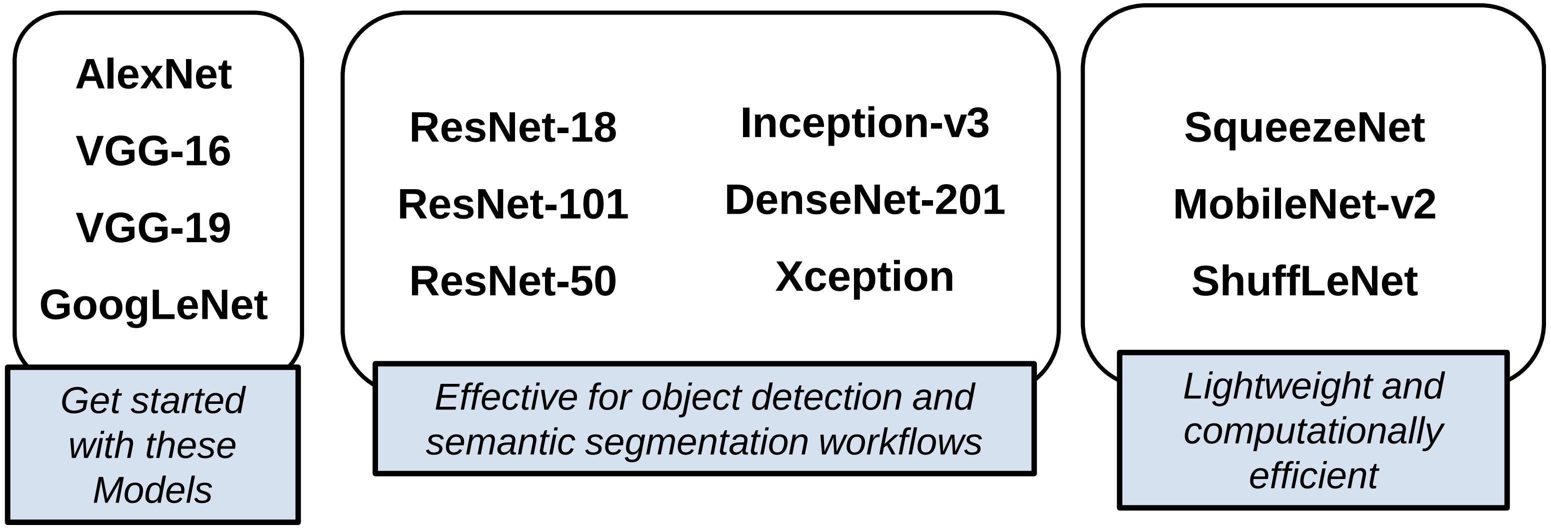
1. Open work_deeplearningin6lines.mlx
2. Follow along with instructor



Category - peppers

We Can Build Networks from Scratch or Use Pretrained Models

- Pretrained models have predefined layer orders and parameter values
- Can be used for inference without training



AlexNet

VGG-16

VGG-19

GoogLeNet

Get started with these Models

ResNet-18

ResNet-101

ResNet-50

Inception-v3

DenseNet-201

Xception

Effective for object detection and semantic segmentation workflows

SqueezeNet

MobileNet-v2

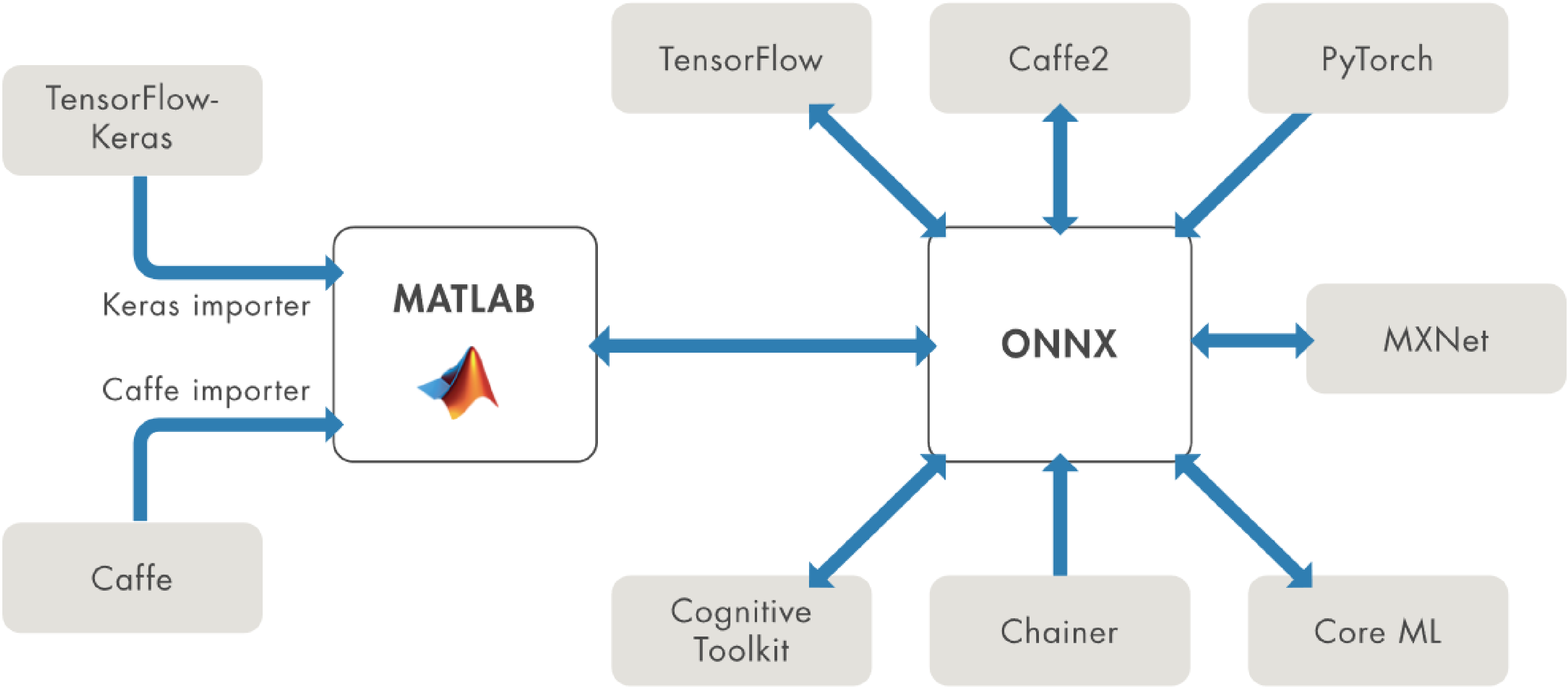
ShuffLeNet

Lightweight and computationally efficient

Full list of models available at: *https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html*

Where Can I Access Pretrained models

- Many are built into MATLAB
- Others can found on the web and imported into MATLAB



Open Neural Network Exchange

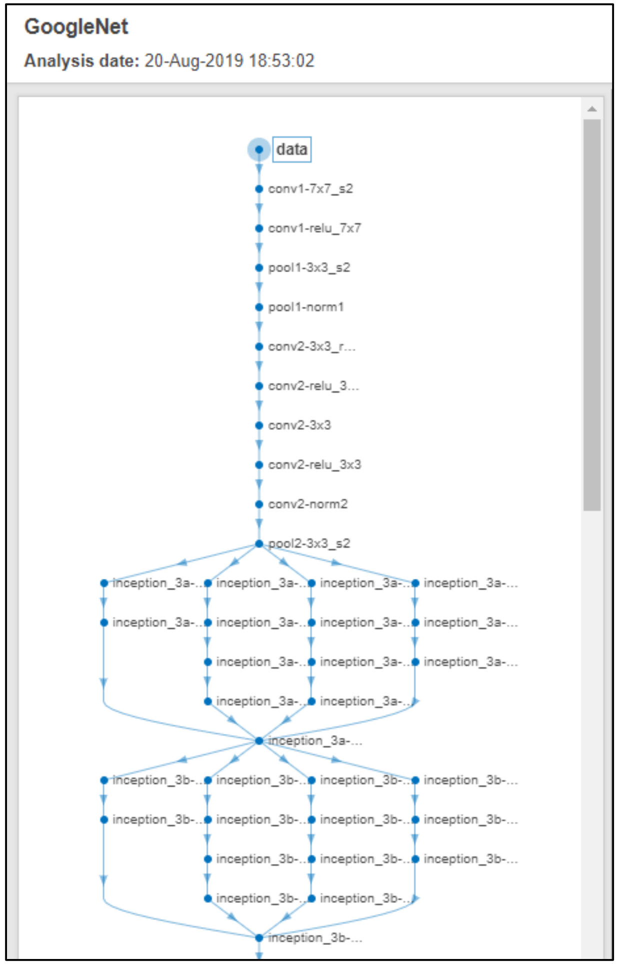
Exercise 2 – Pretrained Models

Purpose:

- Learn how to use pre-trained models in MATLAB.
- See how different network architectures affect results.
- Use datastores to access data efficiently

To Do:

1. Open work_pretrainednetworks.mlx.

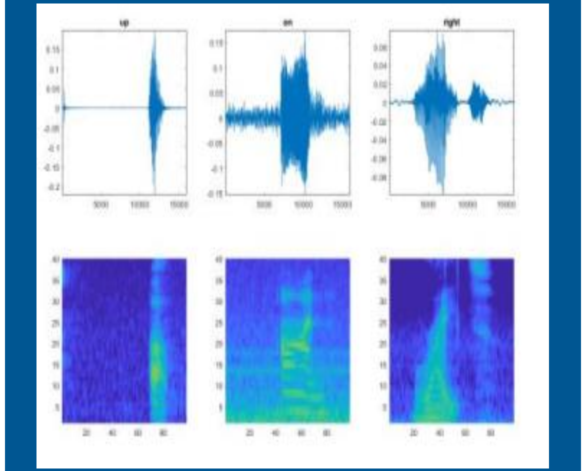


GoogleNet

Analysis date: 20-Aug-2019 18:53:02

Pretrained models aren’t always enough, we may have to build and train networks from scratch

PREPARE DATA



TRAIN MODEL

- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

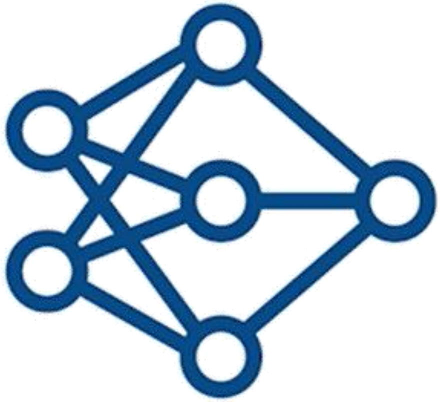
DEPLOY SYSTEM

```
cudaMalloc(&gpu_inputdata, 6183480LL)
cudaMemcpy((void *)gpu_inputdata, (void
c_DeepLearningNetwork_predict_k<<<dim
cudaMemcpy(obj->inputData, gpu_inp
obj->predict();
cudaMemcpy(gpu_out, obj->
d_DeepLearningNetwork_pr
```

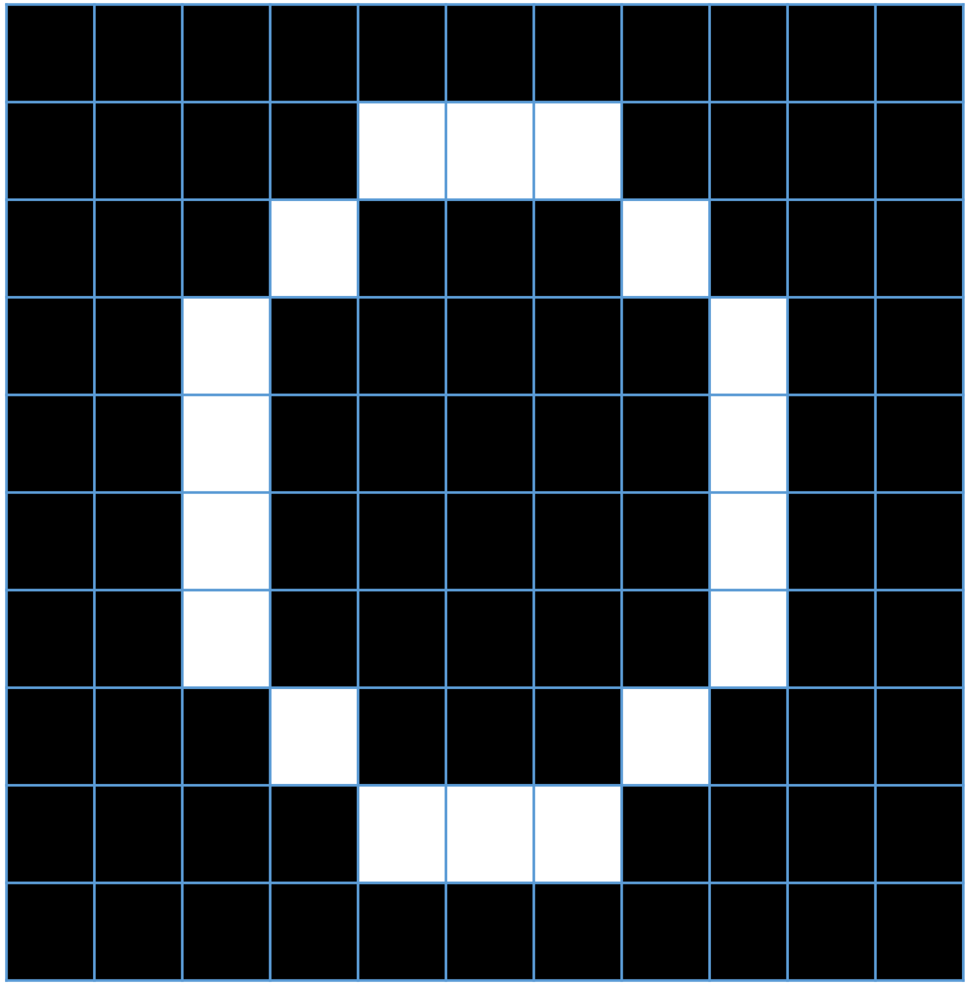
Creating Layer Architectures

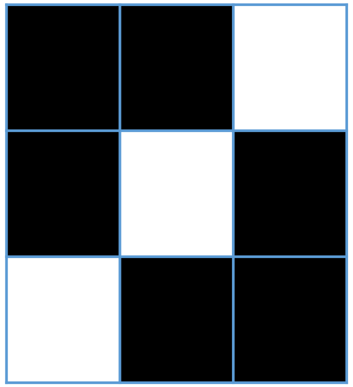
- Convolution Neural Networks – CNN
- Special layer combinations that make them adept at classifying images

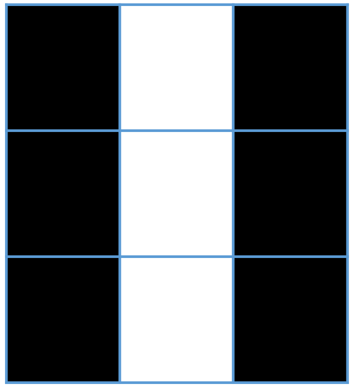
- Convolution Layer
- Max Pooling Layer
- ReLU Layer

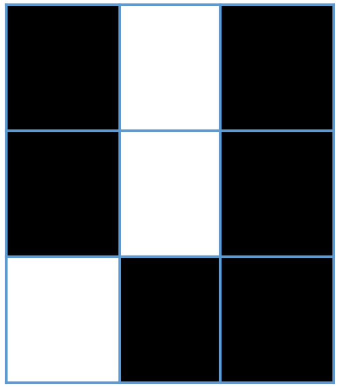


Convolution Layers Search for Patterns

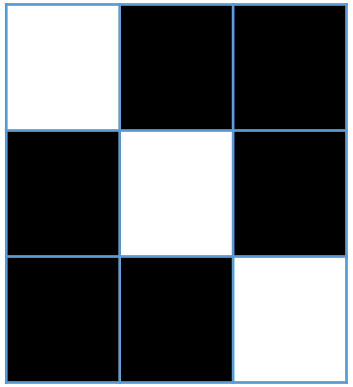


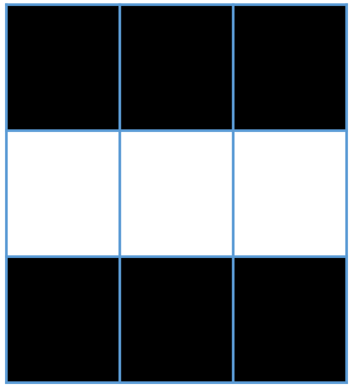


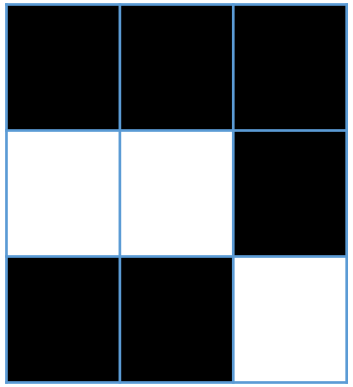




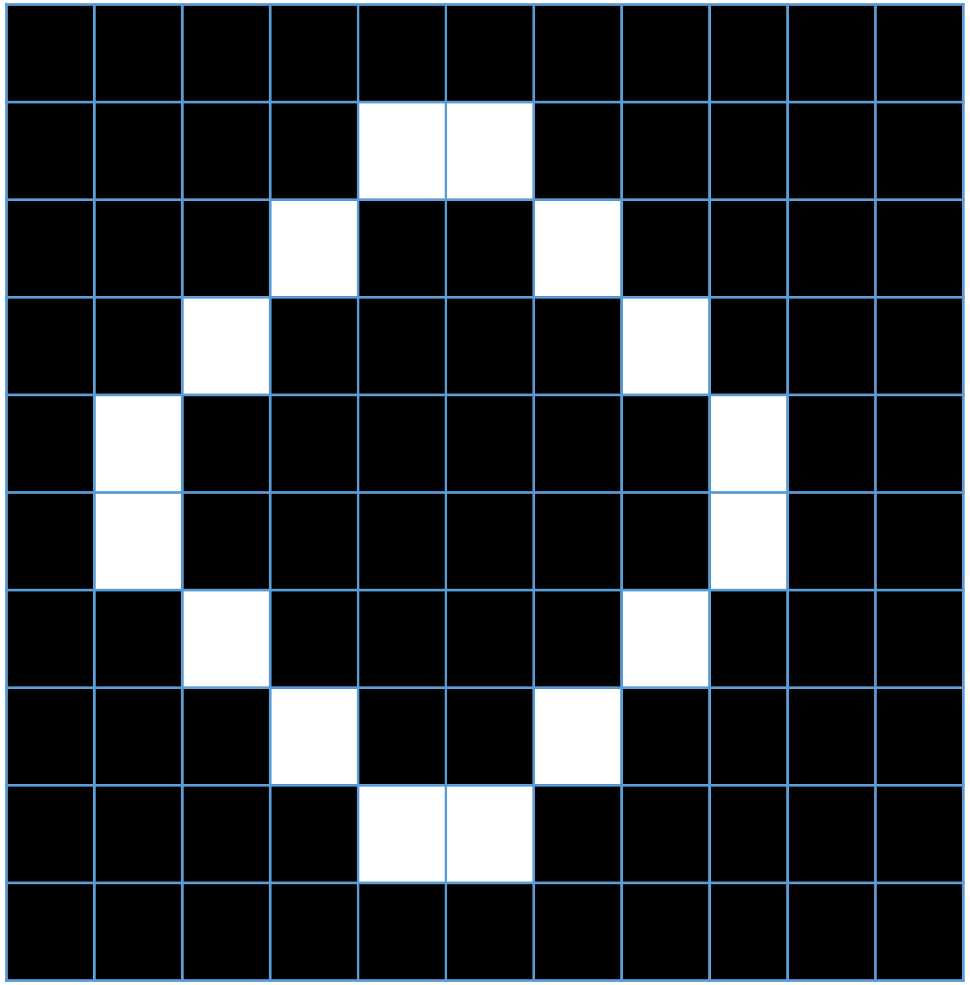
These patterns would be common in the number 0

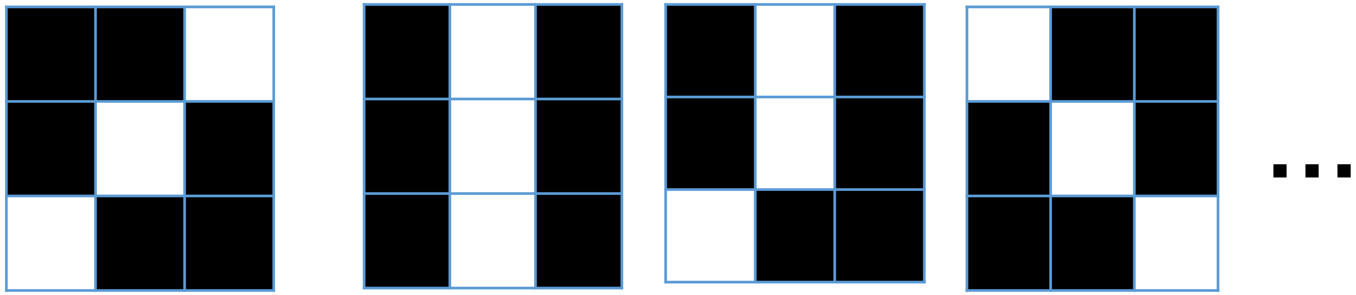






All patterns are compared to the patterns on a new image.



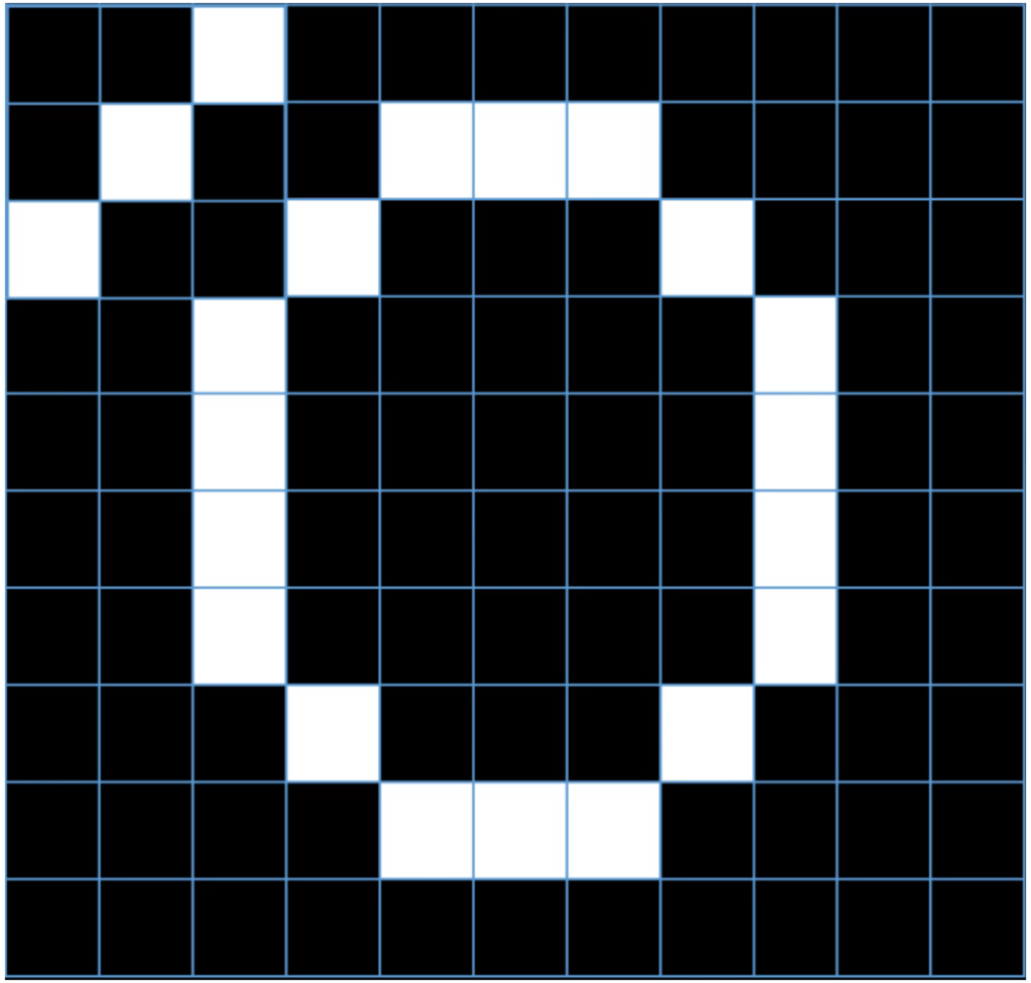


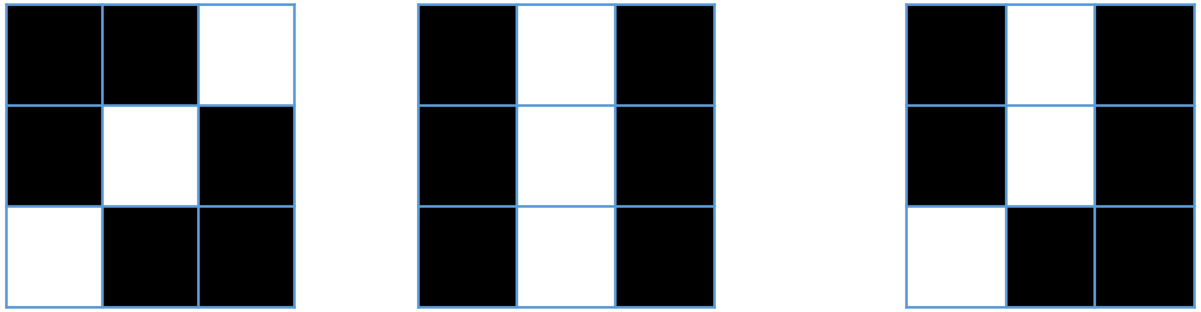
- Pattern starts at left corner

  Perform comparison

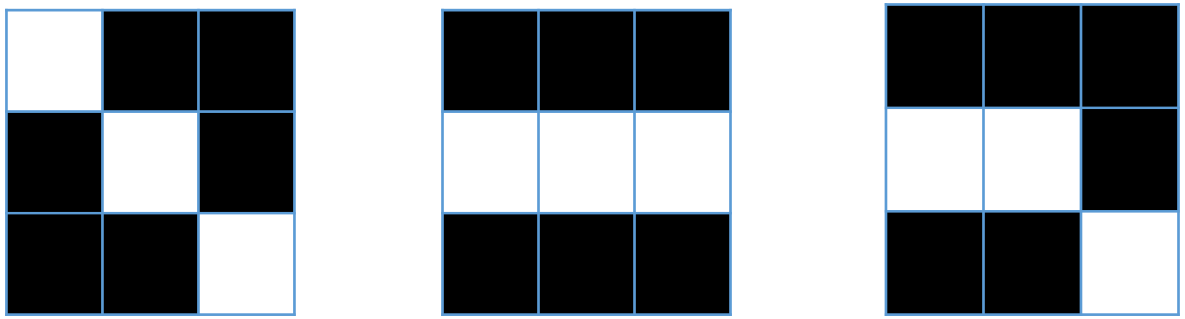
  Slide over one pixel
- Reach end of image
- Repeat for next pattern

Convolution Layers Search for Patterns

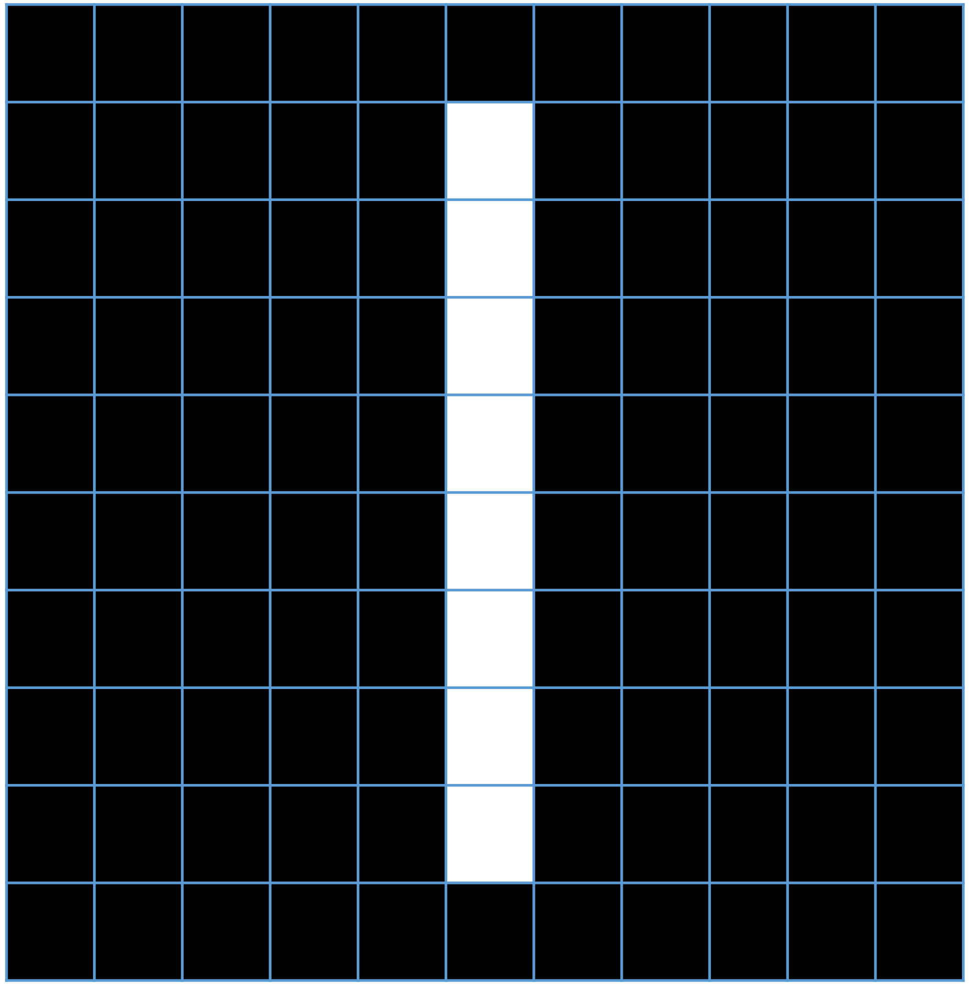




These patterns would be common in the number 0

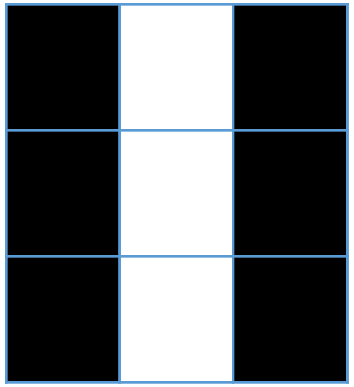


Good pattern matching in convolution improves chances that object will classify properly



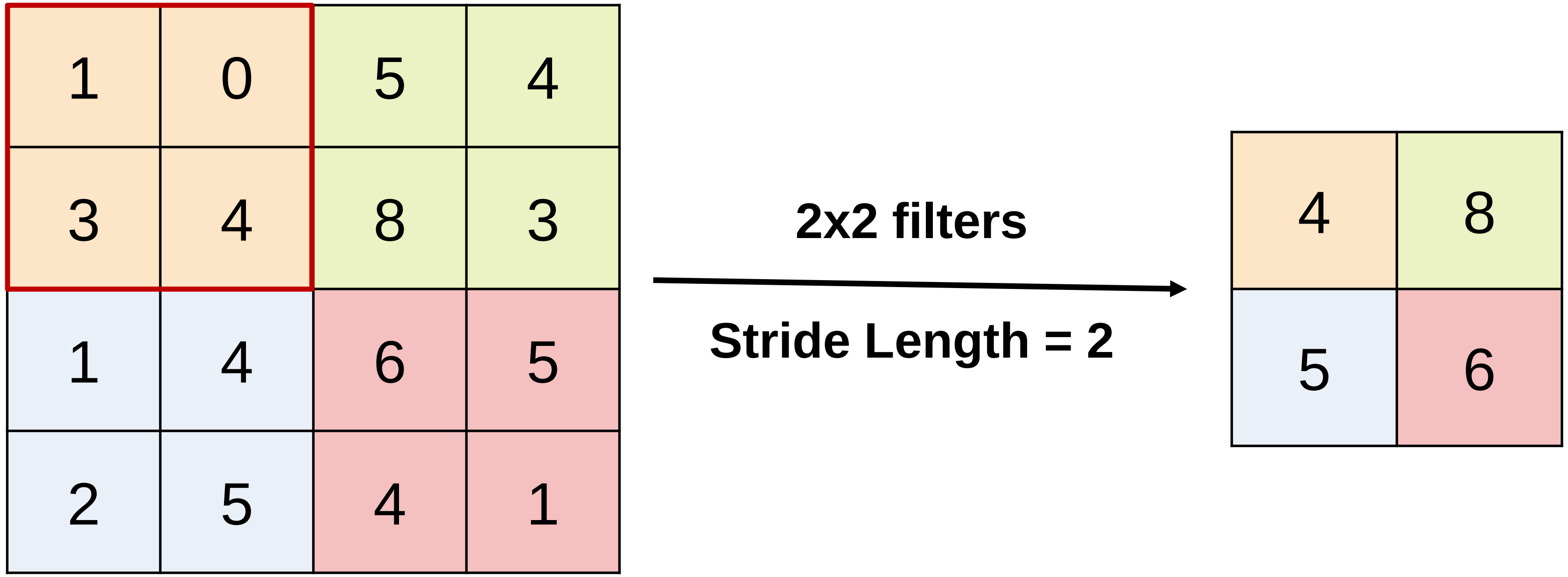
- This image would not match well against the patterns for the number zero

- It would only do very well against this pattern



Max Pooling is a down-sampling operation

Shrink large images while preserving important information



1	0	5	4
3	4	8	3
1	4	6	5
2	5	4	1

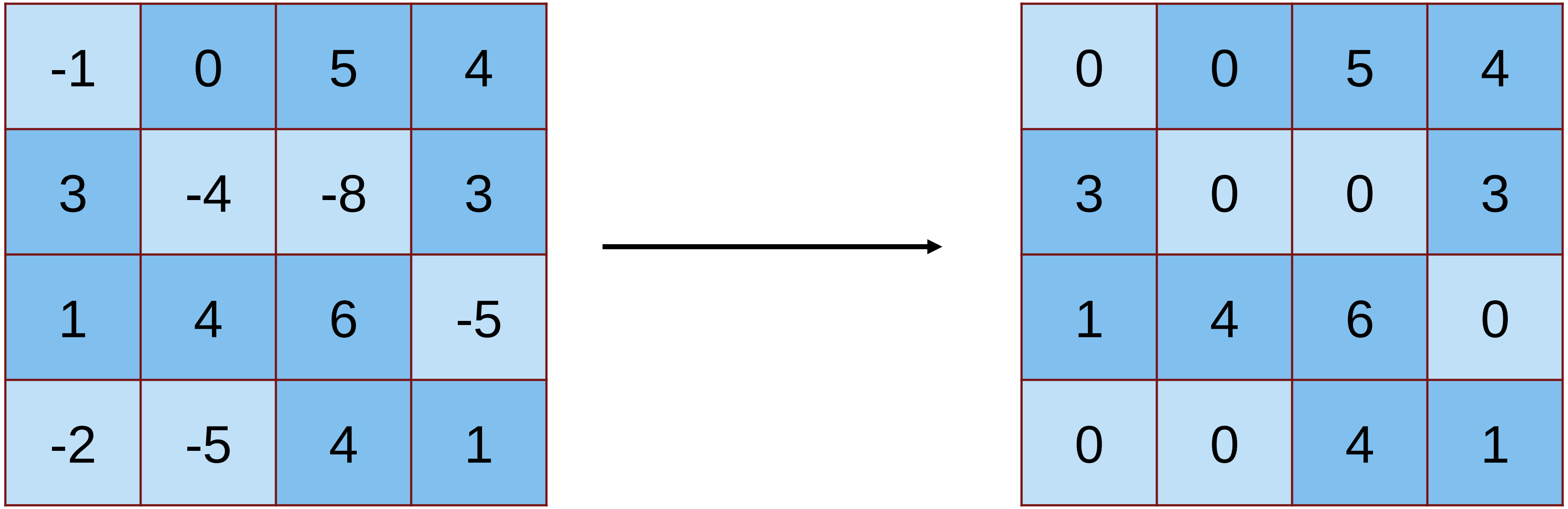
2x2 filters

Stride Length = 2

4	8
5	6

Rectified Linear Units Layer (ReLU)

Converts negative numbers to zero

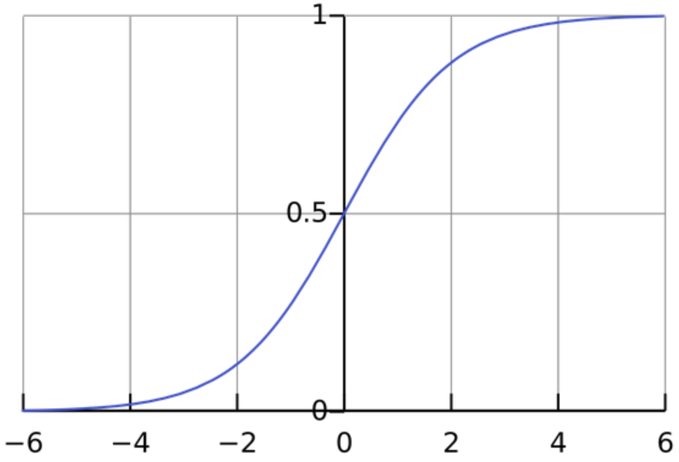


-1	0	5	4
3	-4	-8	3
1	4	6	-5
-2	-5	4	1

0	0	5	4
3	0	0	3
1	4	6	0
0	0	4	1

Classification Problems End with 3 Layers

- Fully Connected Layer
  - Looks at which high-level features correspond to a specific category
  - Calculates scores for each category (highest score wins)



- Softmax Layer
  - Turns scores into probabilities.

- Classification Layer
  - Categorizes image into one of the classes that the network is trained on

Note: Regression problems end with a fully connected layer and regression layer

How Do I know Which Layers to Use?

Feature Extraction - Images

- 2D and 3D convolution
- Transposed convolution (…)

Activation Functions

- ReLU
- Tanh (…)

Sequence Data

Signal, Text, Numeric

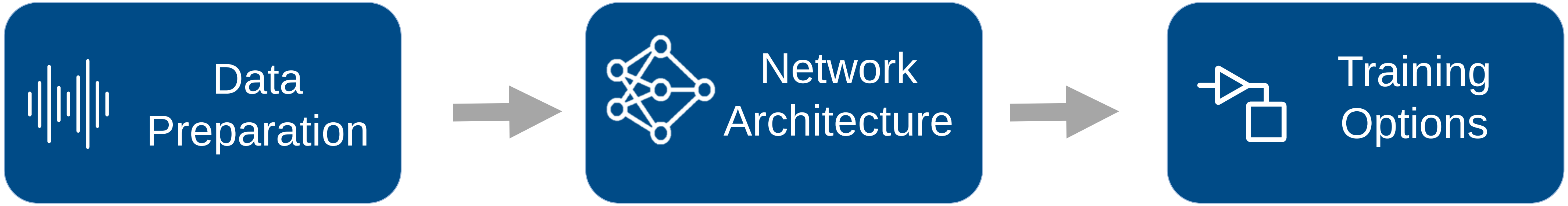
- LSTM
- BiLSTM
- Word Embedding (…)

Normalization

- Dropout
- Batch normalization
- (…)

Tip: Research papers can provide guidelines for creating architecture

3 Components to Train any Network



Data Preparation

Network Architecture

Training Options

"How much data do I need?"

It depends…but **A LOT**

Define inputs and layers for deep learning model

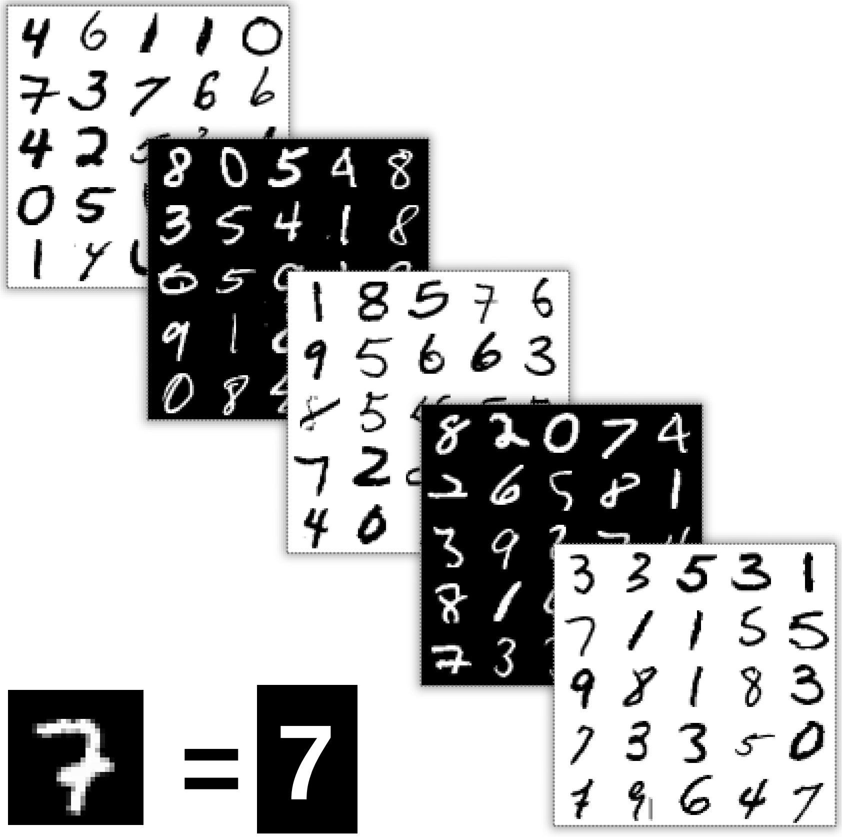
Influence training time and accuracy

- Solver Type
- Initial Learn Rate
- Minibatch Size
- Max Epochs
- …

Exercise 3 - MNIST

Purpose:

- Learn how to create and train deep neural network
- Use MATLAB’s Deep Network Designer
- Explore hyperparameters

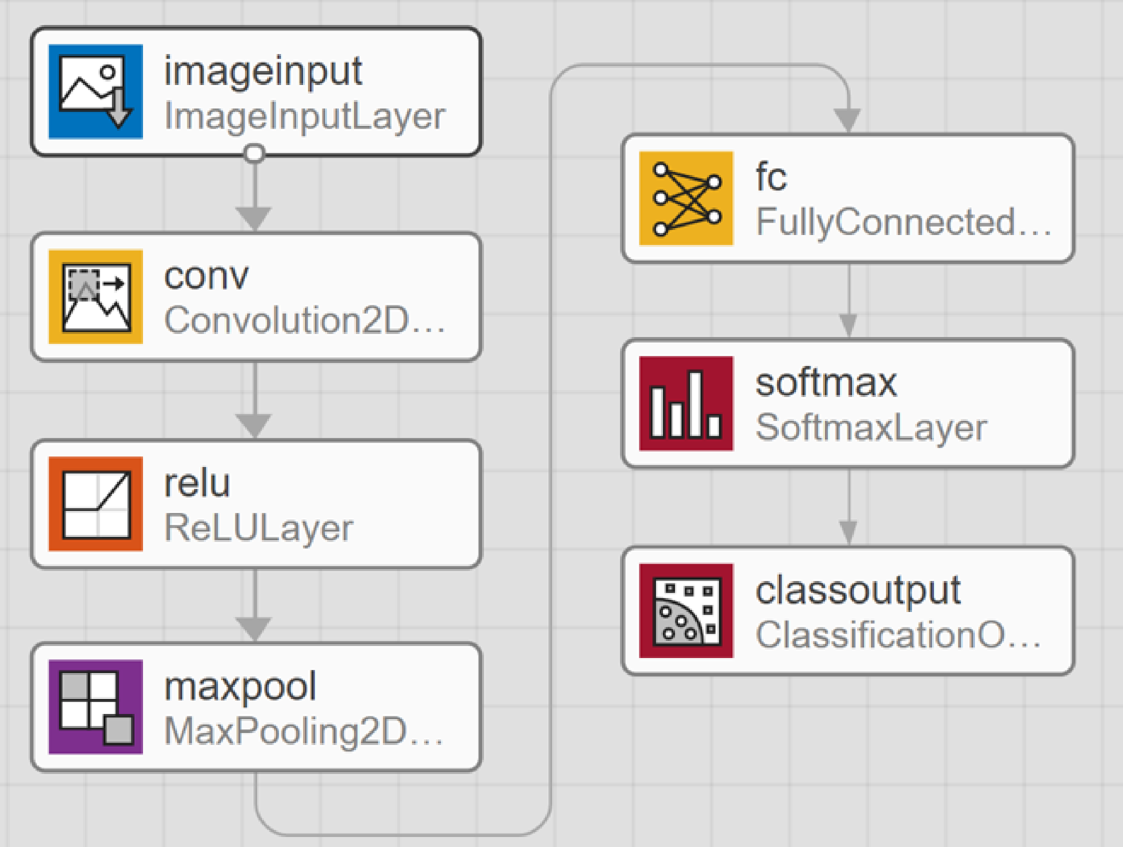


Details

- Dataset consists of handwritten digits 0-9
- 60,000 training images
- 10,000 test images

Sources: *http://yann.lecun.com/exdb/mnist/*
https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results

Network to Create for Part 1 of MNIST



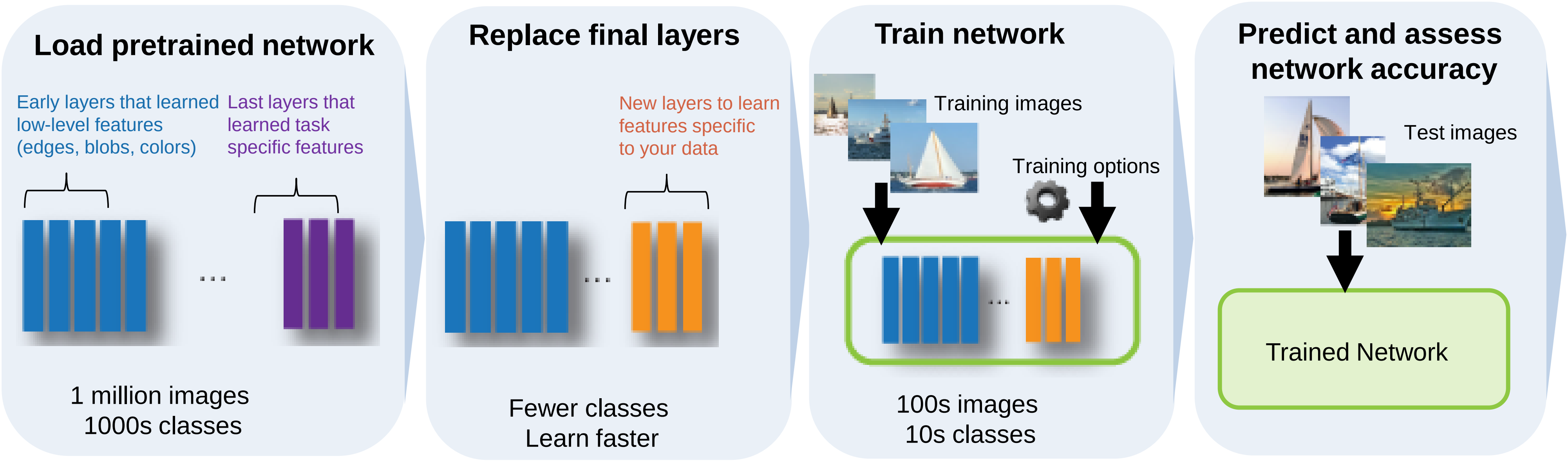
Use Default Parameters Except for the Following Layers

<u>Imageinput</u>
InputSize = 28,28,1

<u>Conv</u>
FilterSize = 5,5
NumFilters = 20
Padding = (0,0,0,0)

<u>Maxpool</u>
PoolSize = 2,2
Stride = 2,2

Transfer Learning Workflow

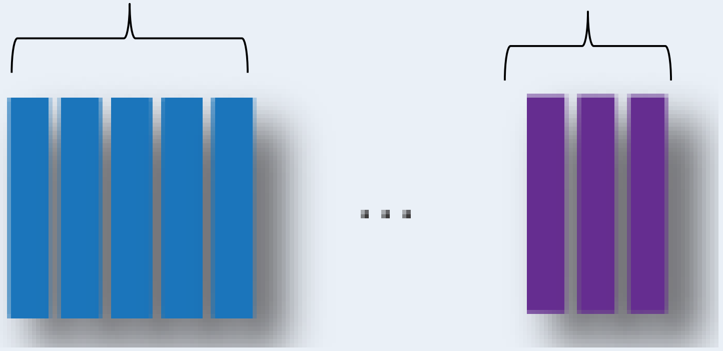

Load pretrained network
Early layers that learned low-level features (edges, blobs, colors)
Last layers that learned task specific features
...
1 million images
1000s classes
Replace final layers
New layers to learn features specific to your data
...
Fewer classes
Learn faster
Train network
Training images
Training options
...
100s images
10s classes
Predict and assess network accuracy
Test images
Trained Network

Transfer Learning Workflow – Step 1

Load pretrained network

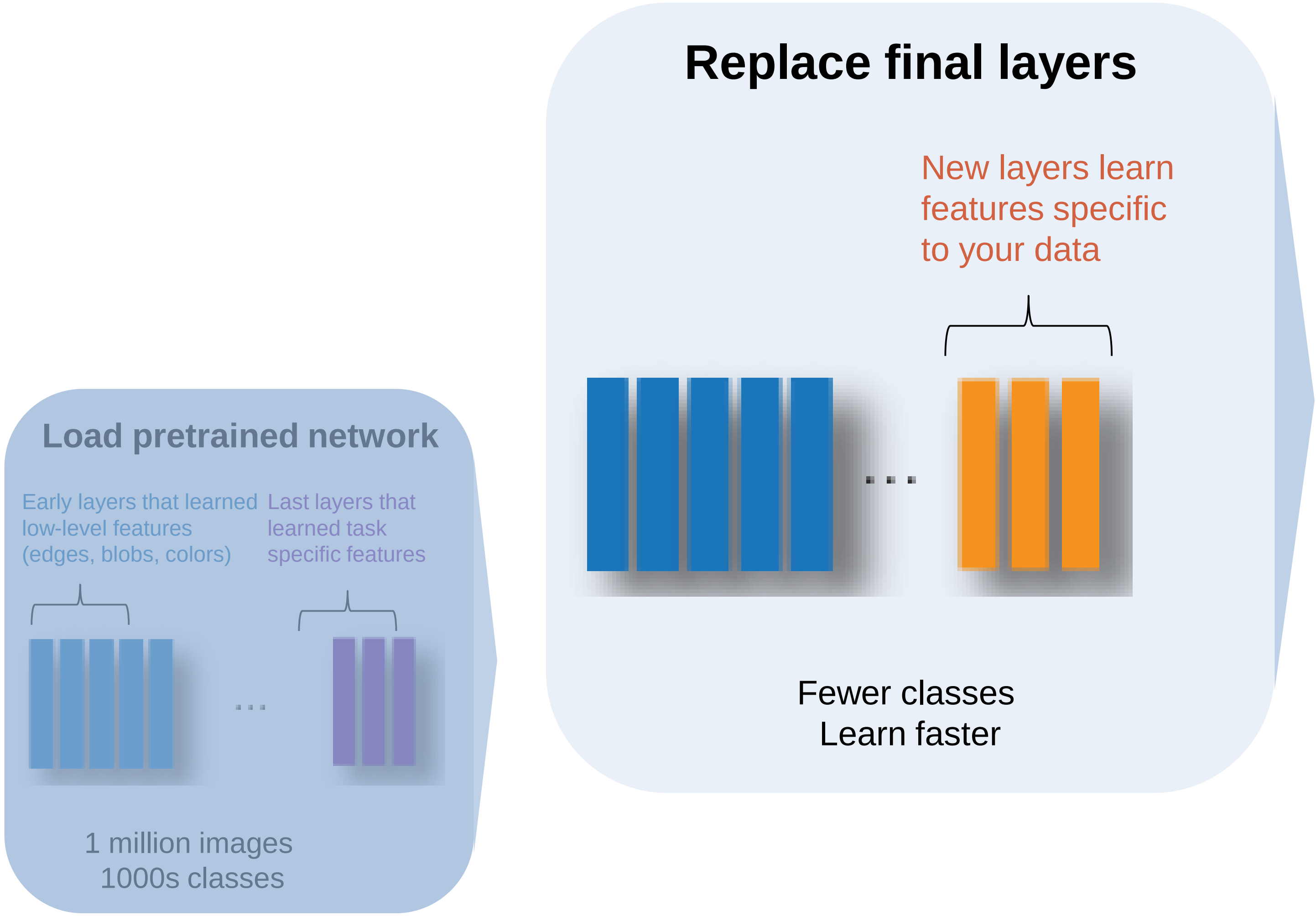
Early layers learn low-level features (edges, blobs, colors)

Last layers learn task-specific features



1 million images
1000s classes

Transfer Learning Workflow – Step 2



Load pretrained network

Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

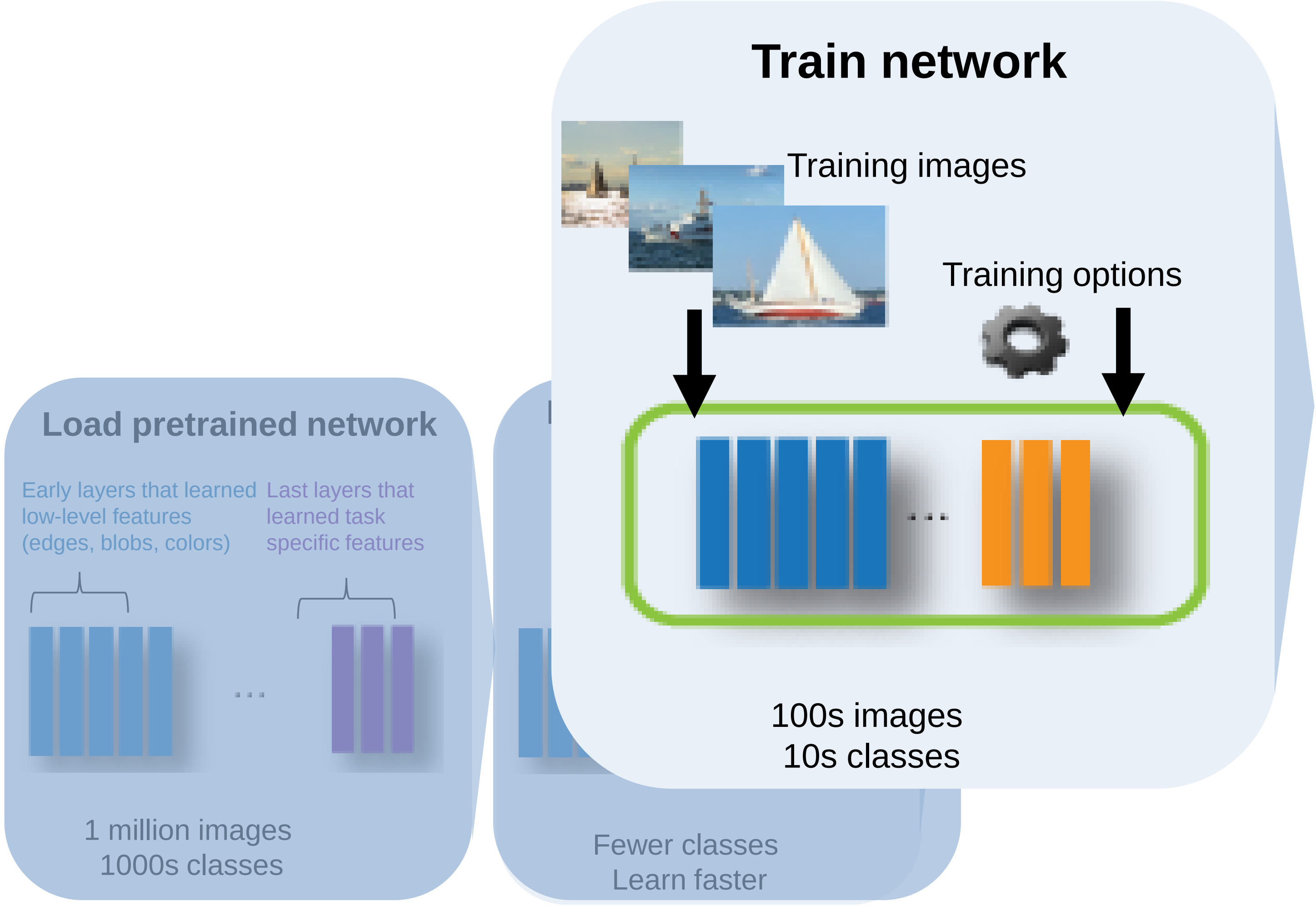
1 million images
1000s classes

Replace final layers

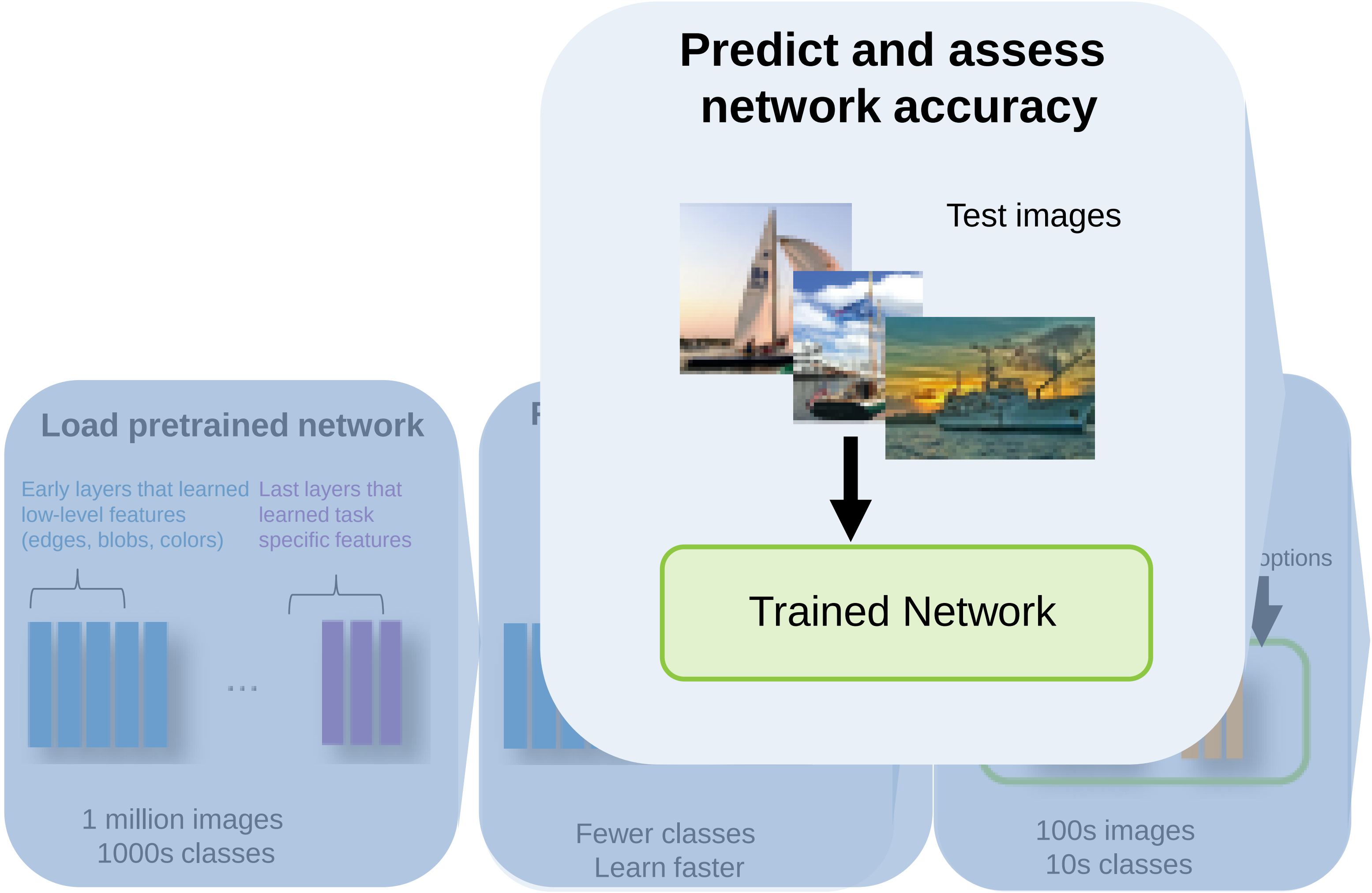
New layers learn features specific to your data

Fewer classes
Learn faster

Transfer Learning Workflow – Step 3



Transfer Learning Workflow – Step 4



Predict and assess network accuracy

Test images

Trained Network

Load pretrained network

Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

...

1 million images
1000s classes

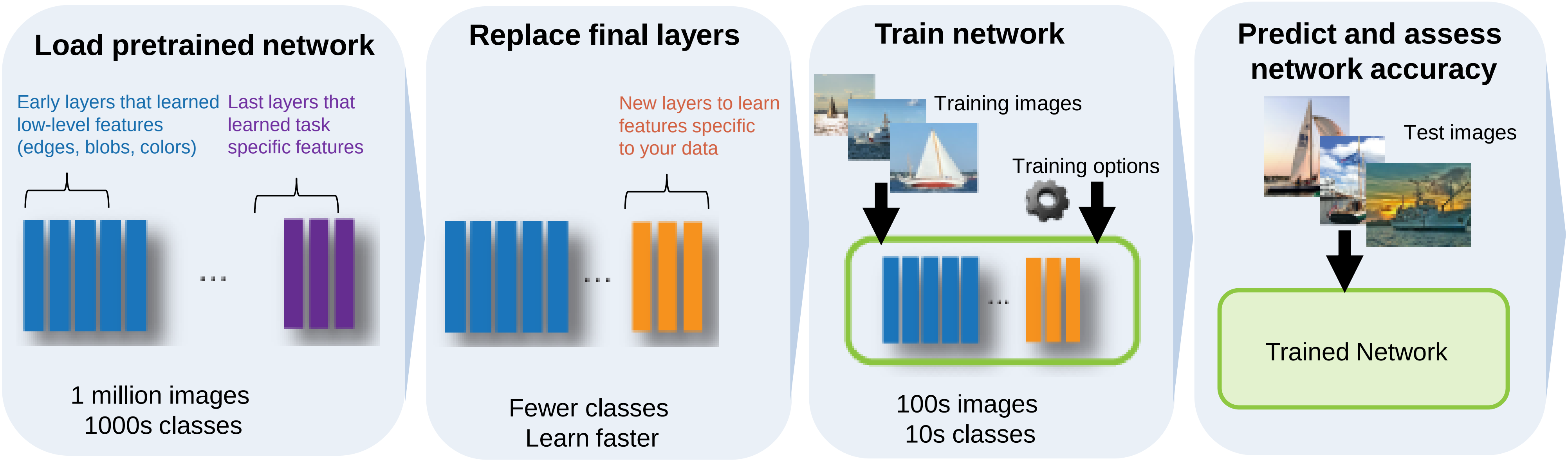
F

Fewer classes
Learn faster

options

100s images
10s classes

Transfer Learning Workflow


Load pretrained network
Early layers that learned low-level features (edges, blobs, colors)
Last layers that learned task specific features
...
1 million images
1000s classes
Replace final layers
New layers to learn features specific to your data
...
Fewer classes
Learn faster
Train network
Training images
Training options
...
100s images
10s classes
Predict and assess network accuracy
Test images
Trained Network

Exercise 4 – Transfer Learning

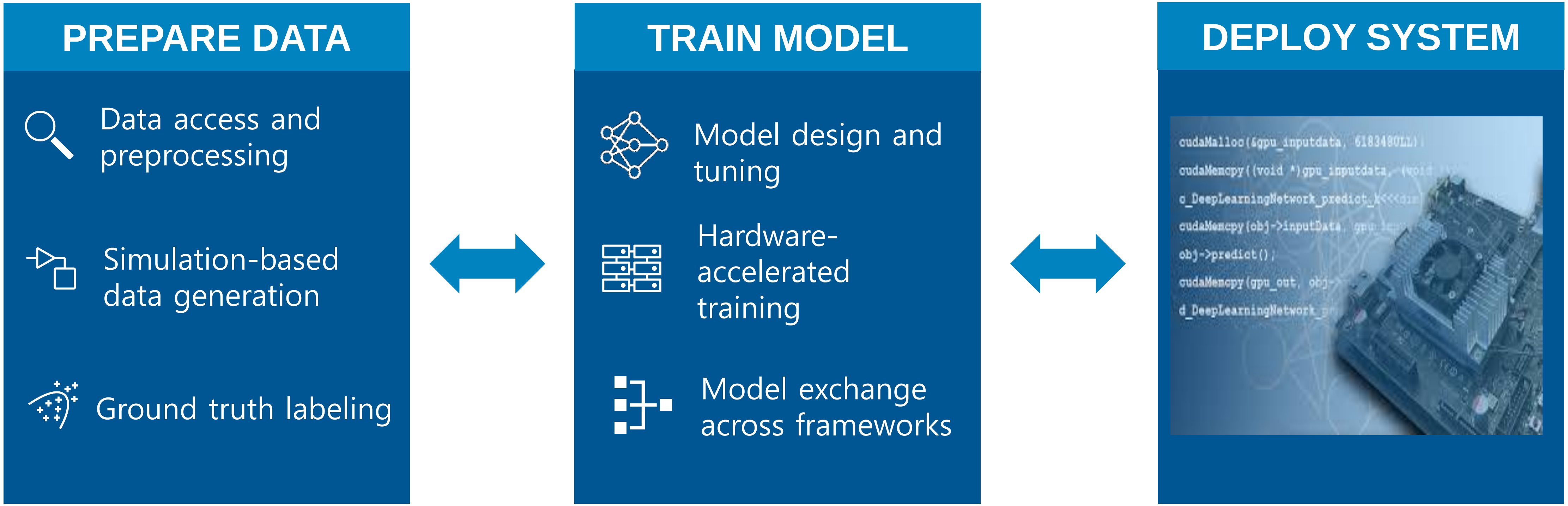
Purpose:

- Use transfer learning to leverage a pretrained model to classify 5 types of food
- Visualize activations within a network

Questions to consider once you understand training

- How do I prepare a dataset for training and testing?
- How do I create ground truth data?
- How do I deploy to hardware or the cloud?

Deep Learning Workflow – Prepare Data



PREPARE DATA

- Data access and preprocessing
- Simulation-based data generation
- Ground truth labeling

TRAIN MODEL

- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

DEPLOY SYSTEM

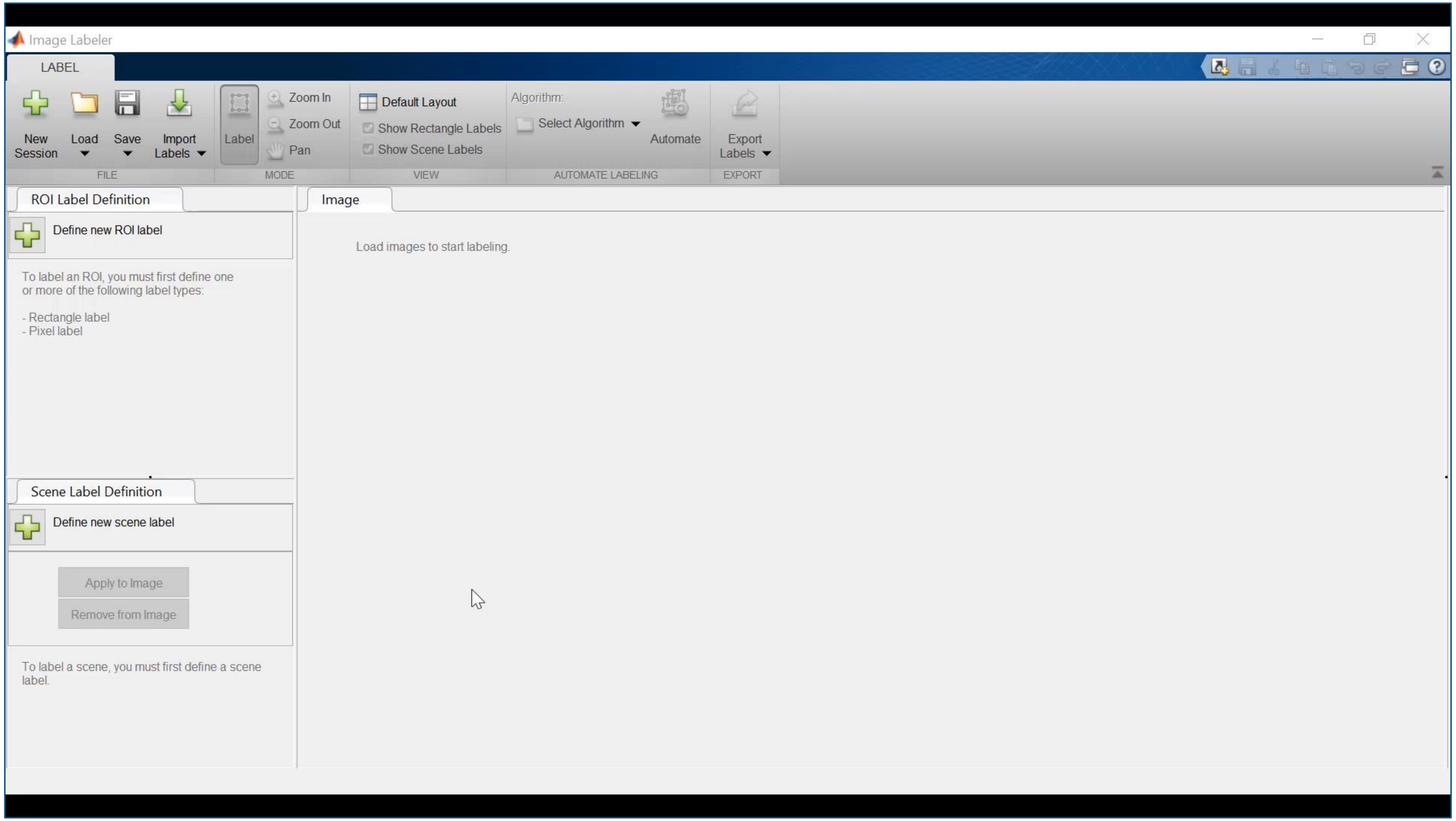
```
cudaMalloc(&gpu_inputdata, 6183480ULL)
cudaMemcpy((void *)gpu_inputdata, (void
c_DeepLearningNetwork_predict_k<<<dim
cudaMemcpy(obj->inputData, gpu_inp
obj->predict();
cudaMemcpy(gpu_out, obj-
d_DeepLearningNetwork_pr
```

How do I label my data?

Image Labeler
+ Video labeler

Signal Labeler
+ Audio Labeler

Big-Image
Labeler



To label an ROI, you must first define one or more of the following label types:

- Rectangle label
- Pixel label

To label a scene, you must first define a scene label.

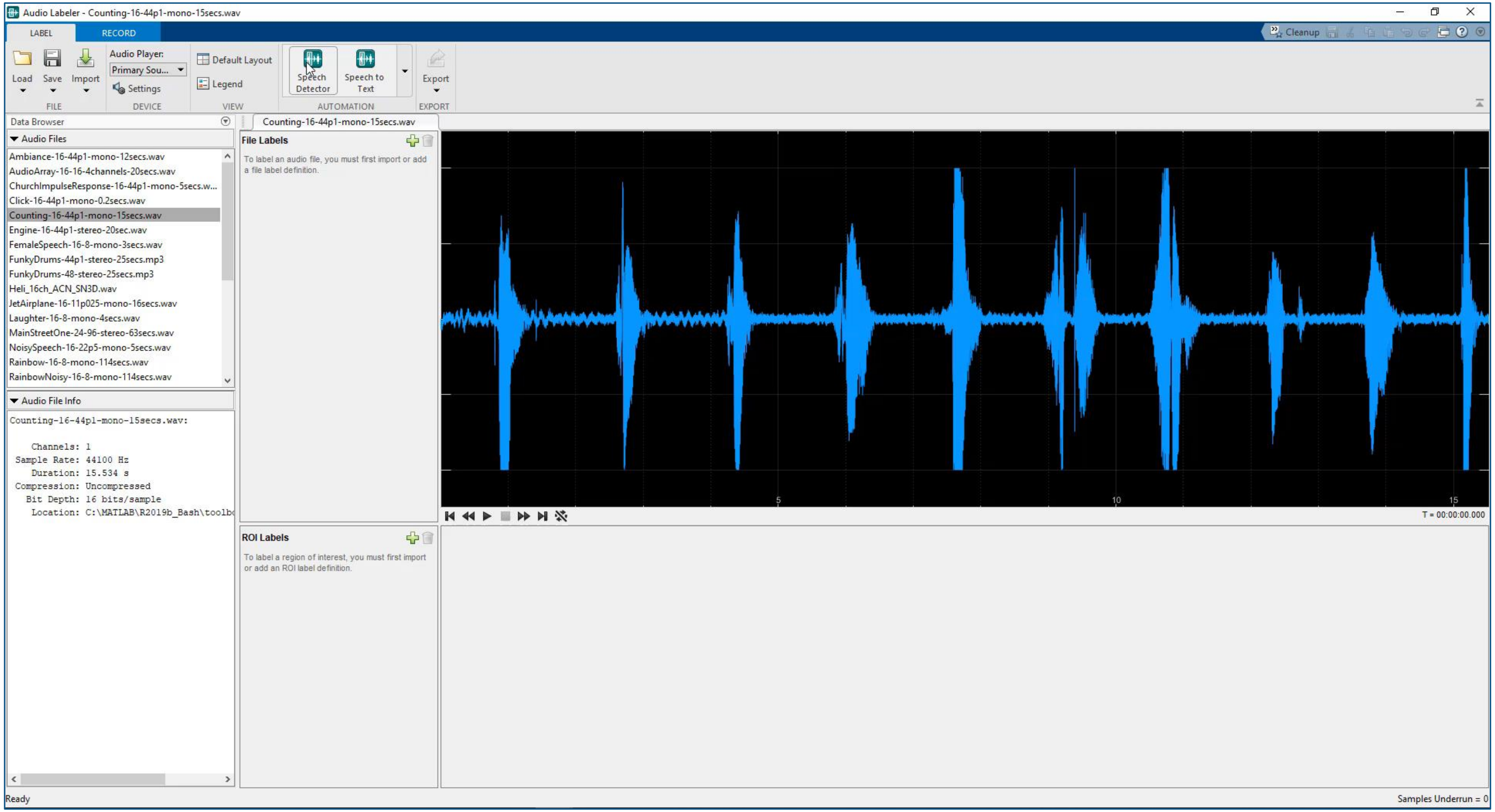
Load images to start labeling.

How do I label my data?

Image Labeler + Video labeler

Signal Labeler + Audio Labeler

Big-Image Labeler



Audio Files

- Ambiance-16-44p1-mono-12secs.wav
- AudioArray-16-16-4channels-20secs.wav
- ChurchImpulseResponse-16-44p1-mono-5secs.w...
- Click-16-44p1-mono-0.2secs.wav
- Counting-16-44p1-mono-15secs.wav
- Engine-16-44p1-stereo-20sec.wav
- FemaleSpeech-16-8-mono-3secs.wav
- FunkyDrums-44p1-stereo-25secs.mp3
- FunkyDrums-48-stereo-25secs.mp3
- Heli_16ch_ACN_SN3D.wav
- JetAirplane-16-11p025-mono-16secs.wav
- Laughter-16-8-mono-4secs.wav
- MainStreetOne-24-96-stereo-63secs.wav
- NoisySpeech-16-22p5-mono-5secs.wav
- Rainbow-16-8-mono-114secs.wav
- RainbowNoisy-16-8-mono-114secs.wav

Audio File Info

```
Counting-16-44p1-mono-15secs.wav:

    Channels: 1
 Sample Rate: 44100 Hz
    Duration: 15.534 s
 Compression: Uncompressed
   Bit Depth: 16 bits/sample
    Location: C:\MATLAB\R2019b_Bash\toolb
```

File Labels

To label an audio file, you must first import or add a file label definition.

ROI Labels

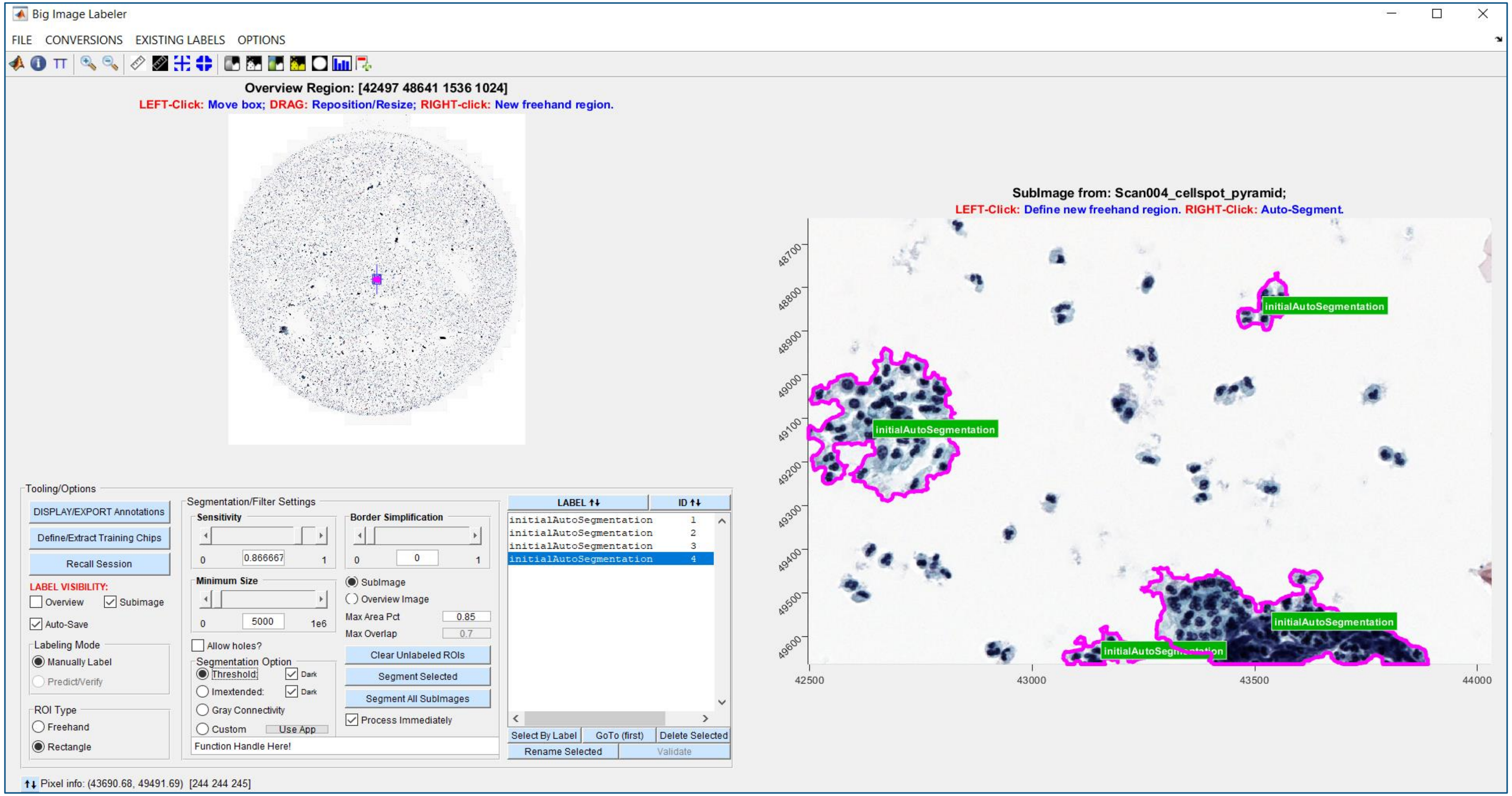
To label a region of interest, you must first import or add an ROI label definition.

How do I label my data?

Image Labeler
+ Video labeler

Signal Labeler
+ Audio Labeler

Big-Image
Labeler



Overview Region: [42497 48641 1536 1024]

LEFT-Click: Move box; DRAG: Reposition/Resize; RIGHT-click: New freehand region.

SubImage from: Scan004_cellspot_pyramid;

LEFT-Click: Define new freehand region. RIGHT-Click: Auto-Segment.

Tooling/Options

LABEL VISIBILITY:
□ Overview ☑ Subimage
☑ Auto-Save

Labeling Mode
● Manually Label
○ Predict/Verify

ROI Type
○ Freehand
● Rectangle

Segmentation/Filter Settings

Sensitivity: 0 – 0.866667 – 1

Minimum Size: 0 – 5000 – 1e6

□ Allow holes?

Segmentation Option
● Threshold: ☑ Dark
○ Imextended: ☑ Dark
○ Gray Connectivity
○ Custom

Function Handle Here!

Border Simplification: 0 – 0 – 1

● SubImage
○ Overview Image

Max Area Pct 0.85

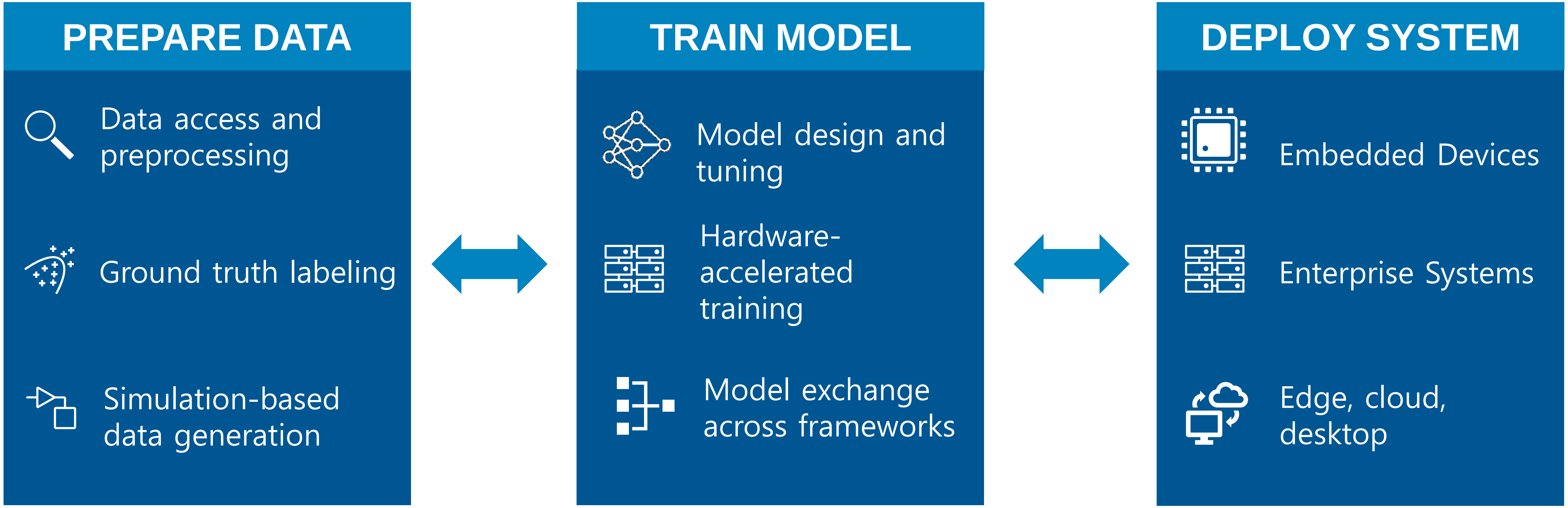
Max Overlap 0.7

☑ Process Immediately

LABEL ↑↓	ID ↑↓
initialAutoSegmentation	1
initialAutoSegmentation	2
initialAutoSegmentation	3
initialAutoSegmentation	4

↑↓ Pixel info: (43690.68, 49491.69) [244 244 245]

Deep Learning Workflow – Deploy System



PREPARE DATA

- Data access and preprocessing
- Ground truth labeling
- Simulation-based data generation

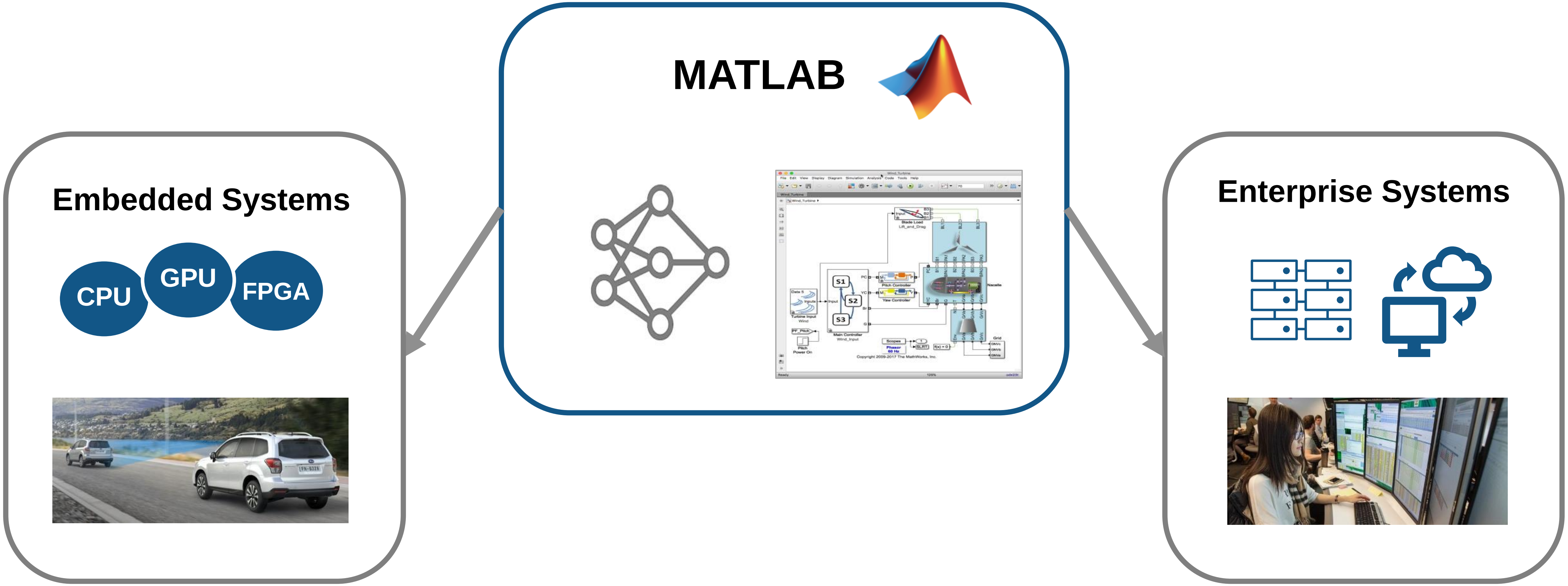
TRAIN MODEL

- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

DEPLOY SYSTEM

- Embedded Devices
- Enterprise Systems
- Edge, cloud, desktop

Deployment and Scaling for A.I.



MATLAB

Embedded Systems

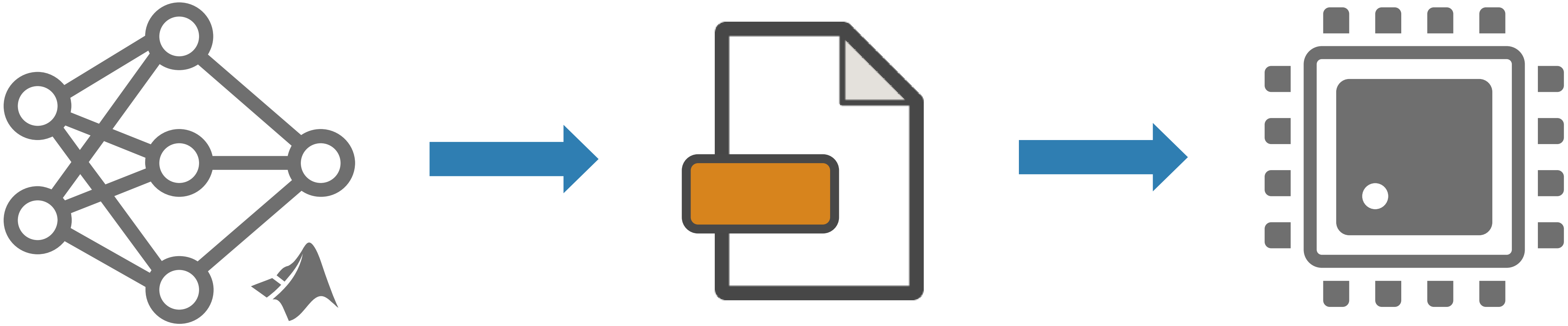
CPU

GPU

FPGA

Enterprise Systems

Embedded Deployment – Automatic Code Generation

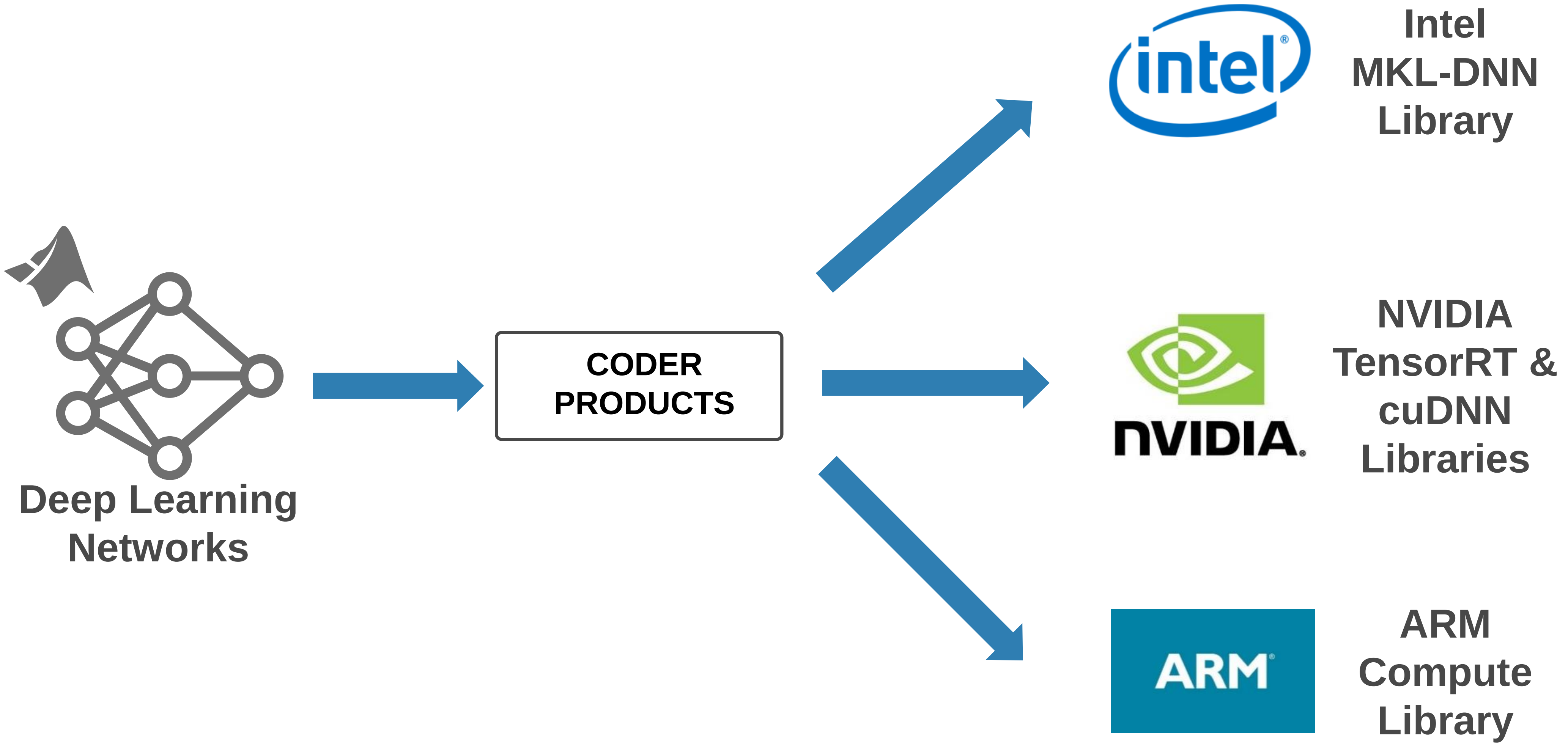


MATLAB Code

Auto-generated Code
(C/C++/CUDA)

Deployment
Target

Deploying Models for Inference



Deep Learning Networks

CODER PRODUCTS

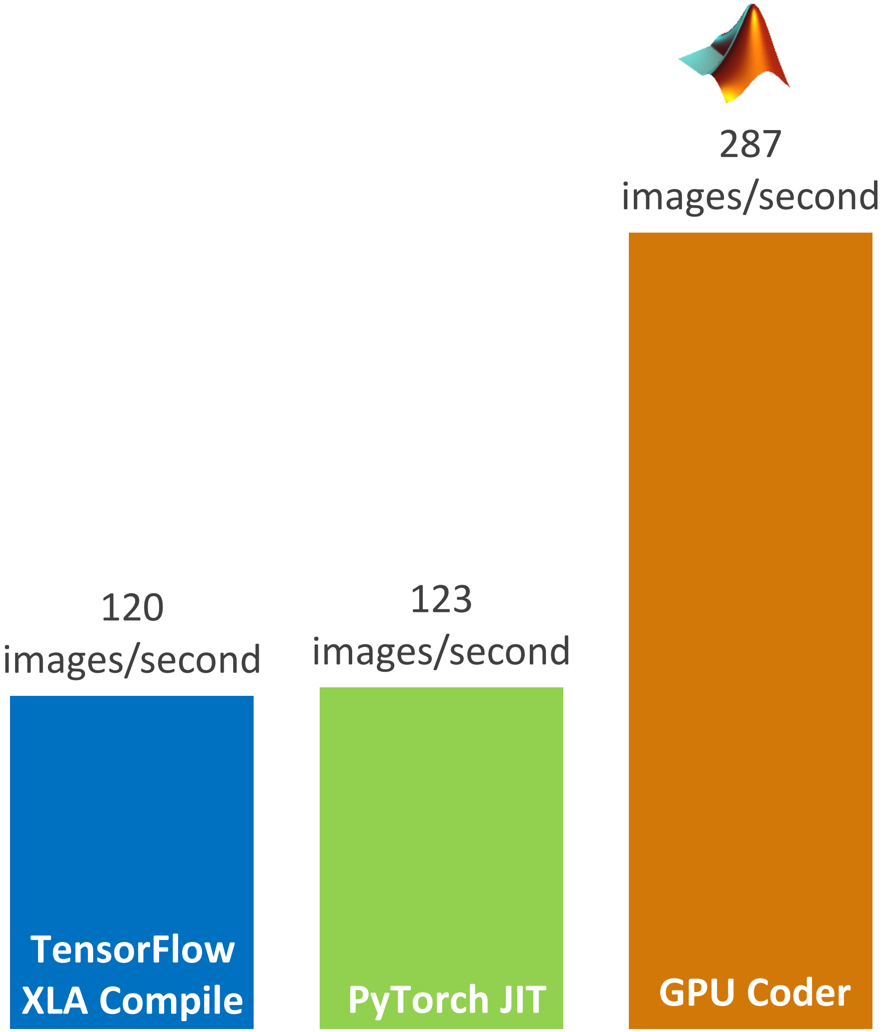
Intel MKL-DNN Library

NVIDIA TensorRT & cuDNN Libraries

ARM Compute Library

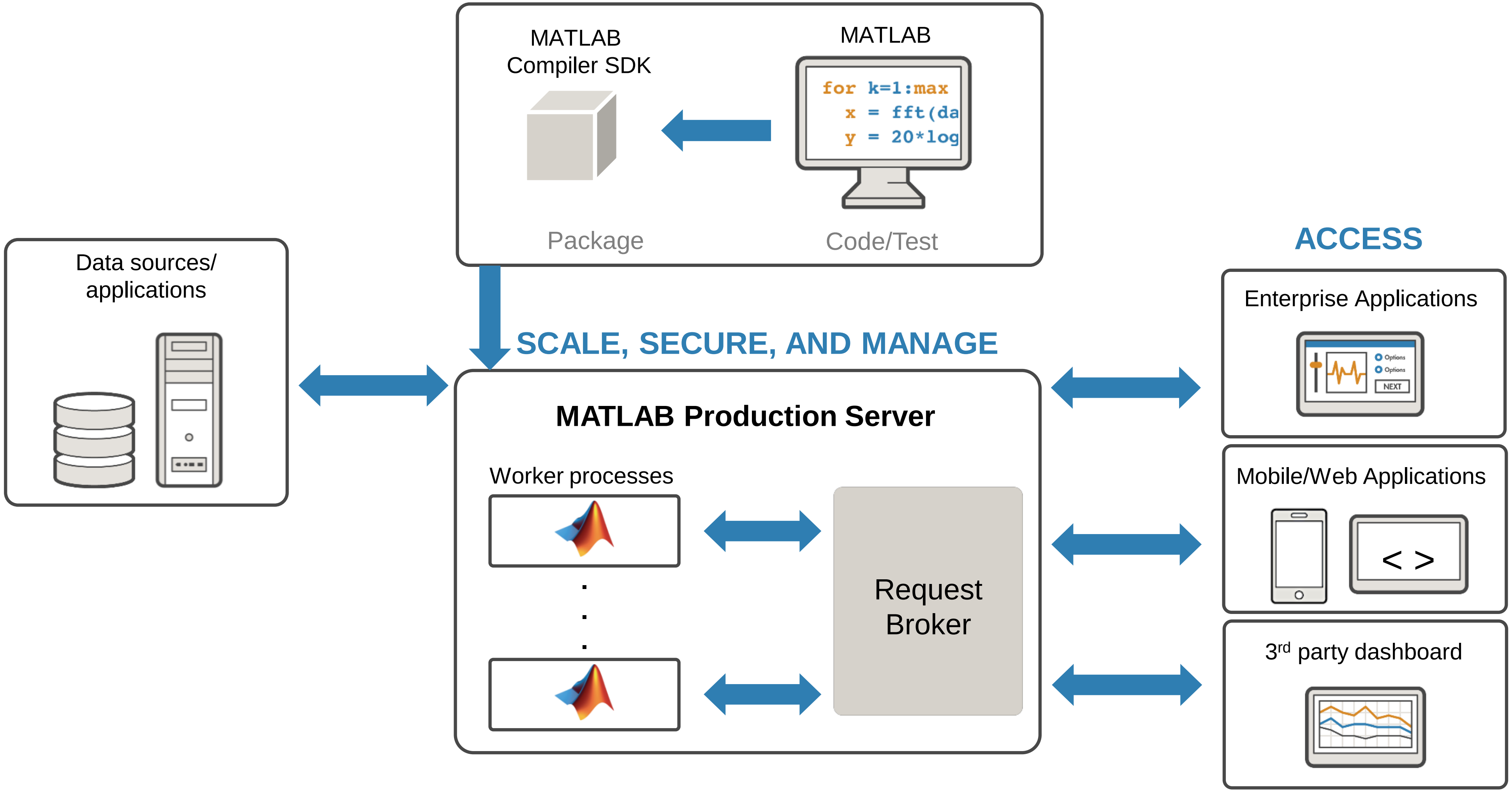
NVIDIA, the NVIDIA logo, and TensorRT are registered trademarks of NVIDIA Corporation
Intel logo is a registrered trademark of Intel Coroporation

GPU Coder is over 2x Faster Than Other Compiled Frameworks



Intel® Xeon® CPU 3.6 GHz with NVIDIA® Titan V GPU - NVIDIA libraries: CUDA® – cuDNN - Frameworks: TensorFlow™ 1.13.0, PyTorch 1.1.0 – ResNet-50 pre-trained model

Deployment to the Cloud and Data Centers with MATLAB Compiler and MATLAB Production Server

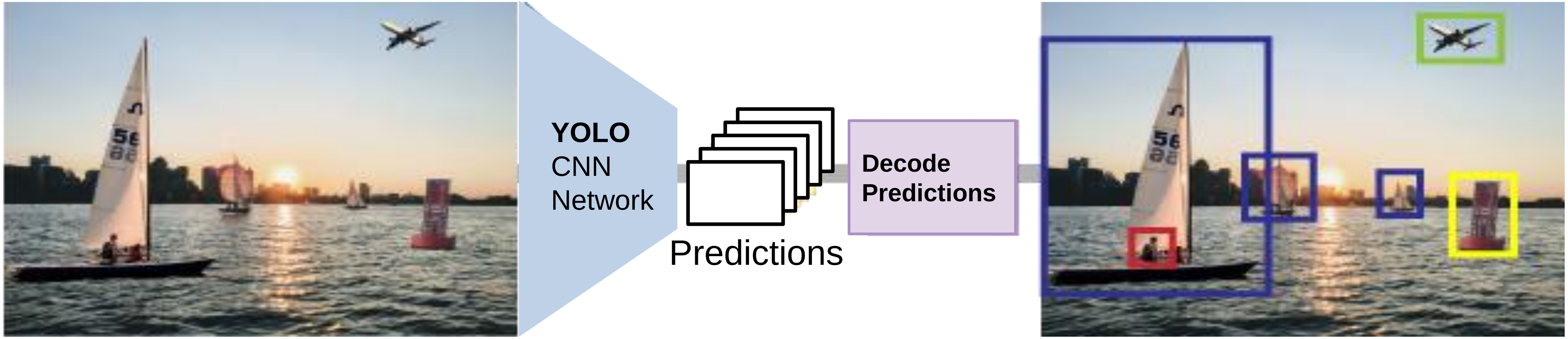


```
for k=1:max
  x = fft(da
  y = 20*log
```

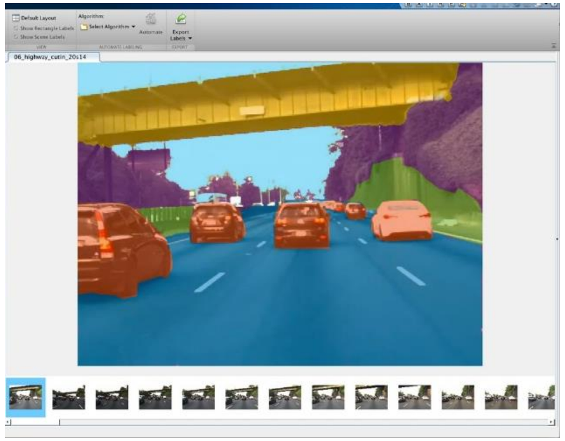
GPU Code Generation with ONNX and YOLOv2

YOLOv2

- **Y**ou **O**nly **L**ook **O**nce
- Real-time object detector
- 1000x faster than R-CNN
- Autonomous driving, traffic monitoring

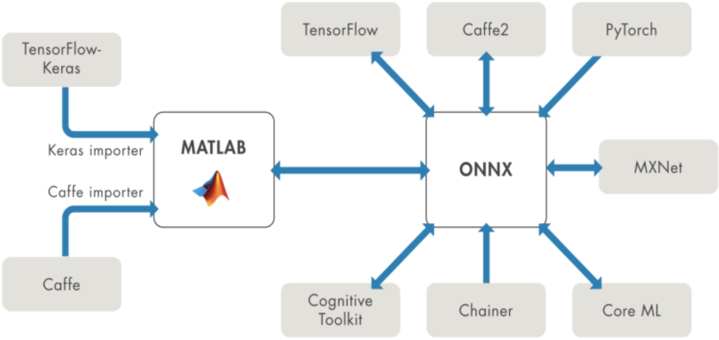


Why Use MATLAB



MATLAB supports the **data preparation, training, and deployment** workflow

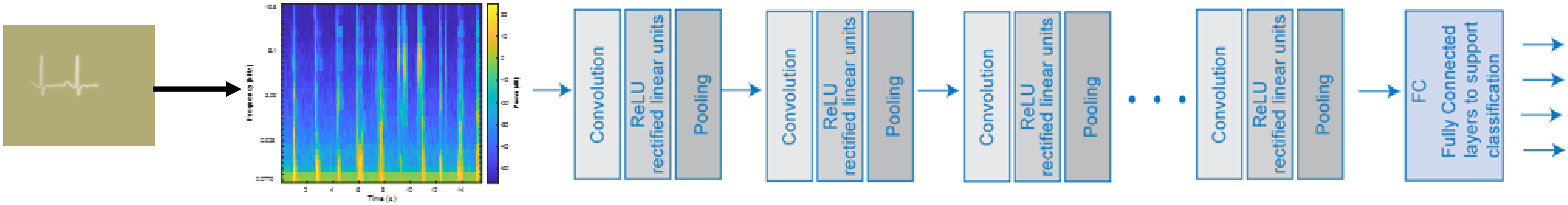
MATLAB has specialized DL tools designed for **scientists and engineers**



MATLAB **interoperates and enhances** Open Source frameworks

Common Network Architectures - Signal Processing

Convolutional Neural Networks (**CNN**)



Convolution

ReLU rectified linear units

Pooling

Convolution

ReLU rectified linear units

Pooling

Convolution

ReLU rectified linear units

Pooling

• • •

Convolution

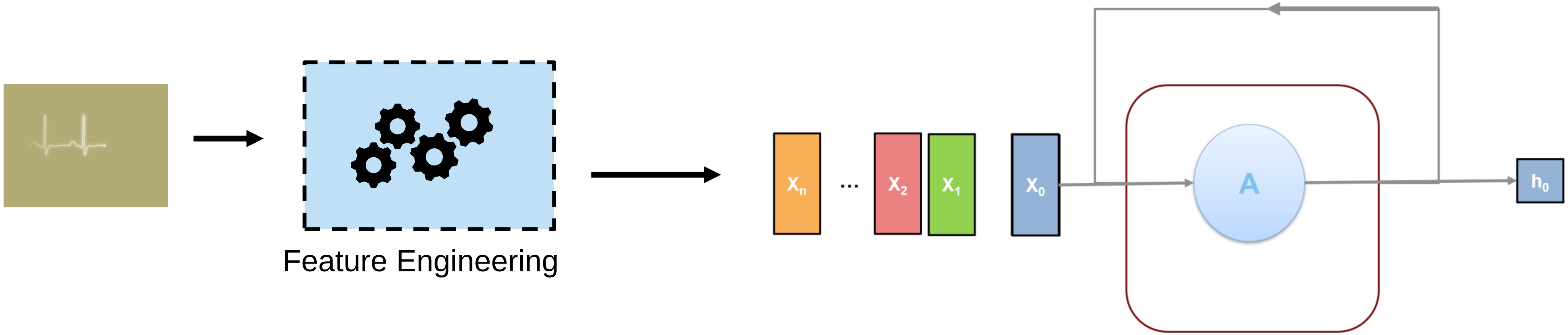
ReLU rectified linear units

Pooling

FC Fully Connected layers to support classification

Time-Frequency Transformation

Long Short Term Memory (**LSTM**) Networks

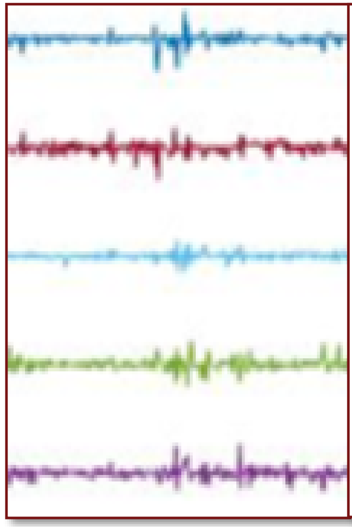


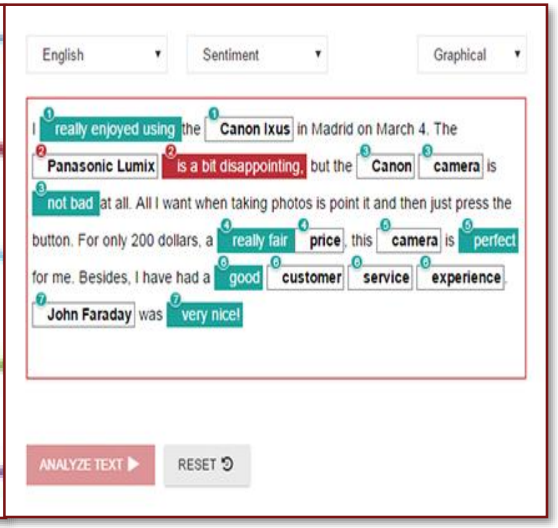
Feature Engineering

Selecting a Network Architecture

Image Data

Signal or Text Data





I really enjoyed using the Canon Ixus in Madrid on March 4. The Panasonic Lumix is a bit disappointing, but the Canon camera is not bad at all. All I want when taking photos is point it and then just press the button. For only 200 dollars, a really fair price, this camera is perfect for me. Besides, I have had a good customer service experience. John Faraday was very nice!

CNN

LSTM or CNN

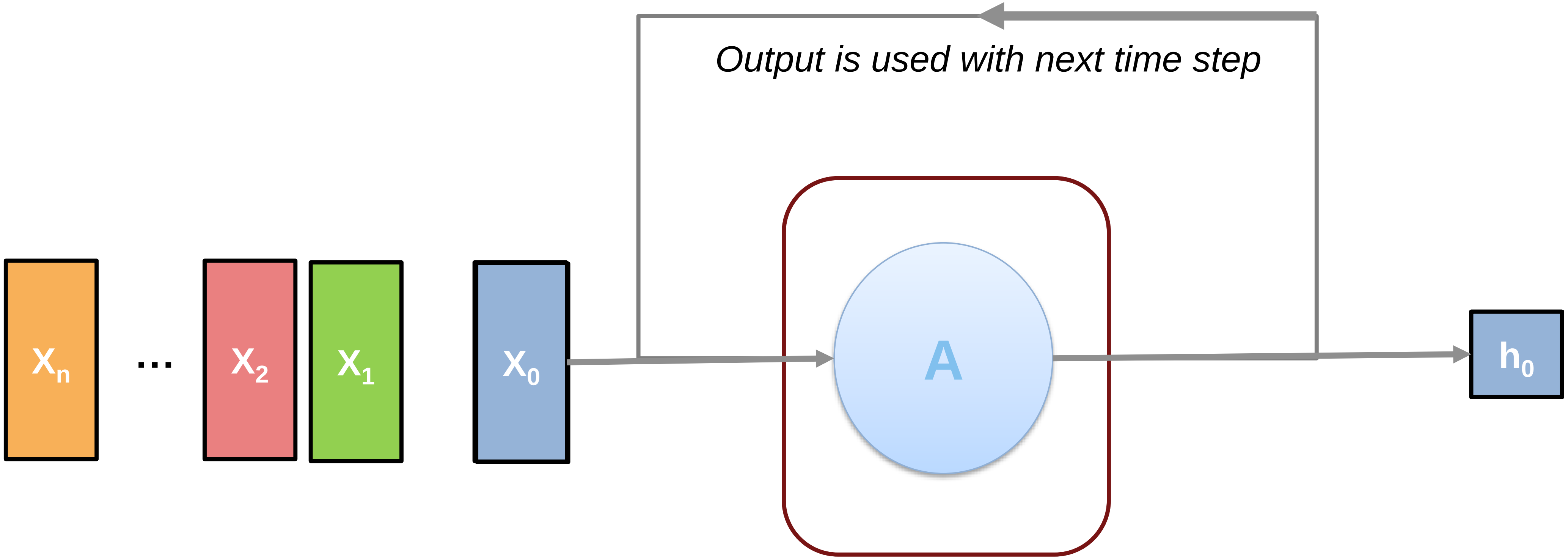
LSTM = Long Short Term Series Network (more detail in later slides)

I was born in France…

… I speak ___________ ?

Recurrent Neural Networks

Take into account previous data when making new predictions



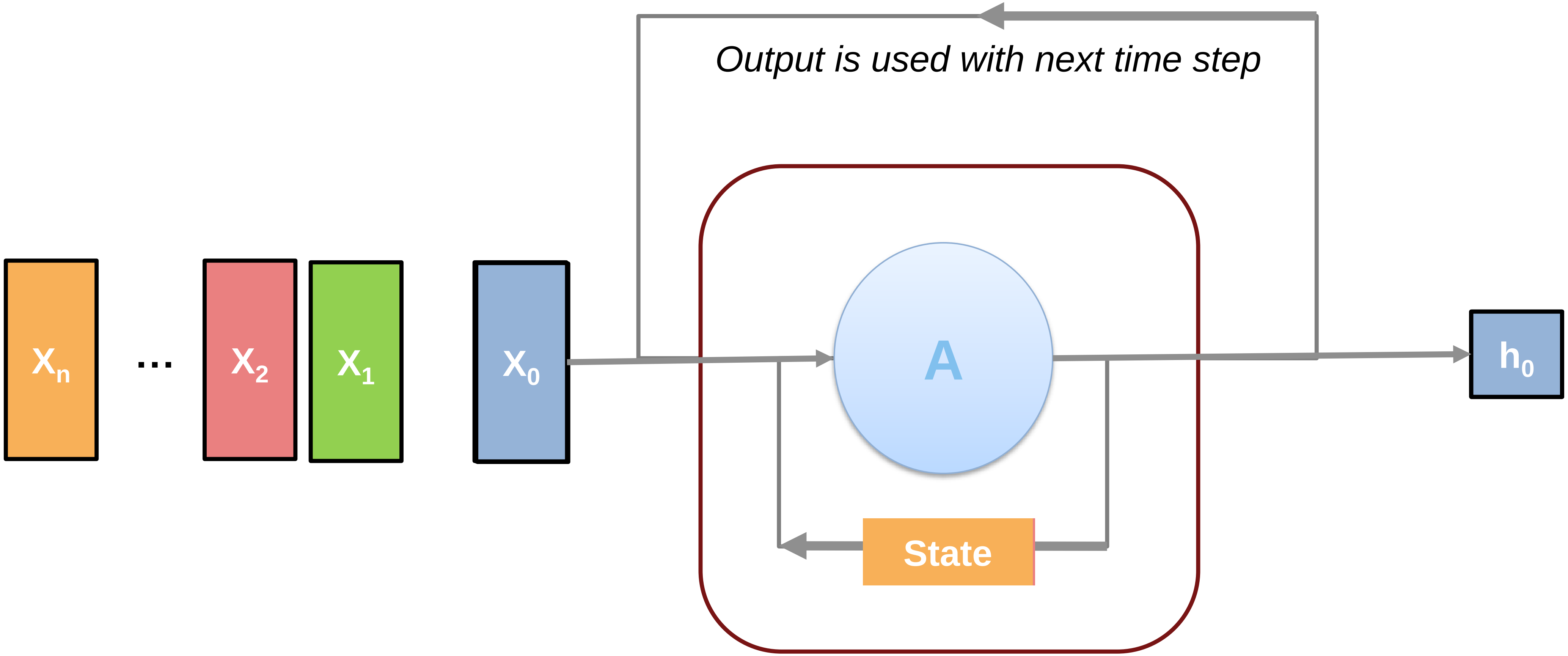
I was born in France…

[2000 words]

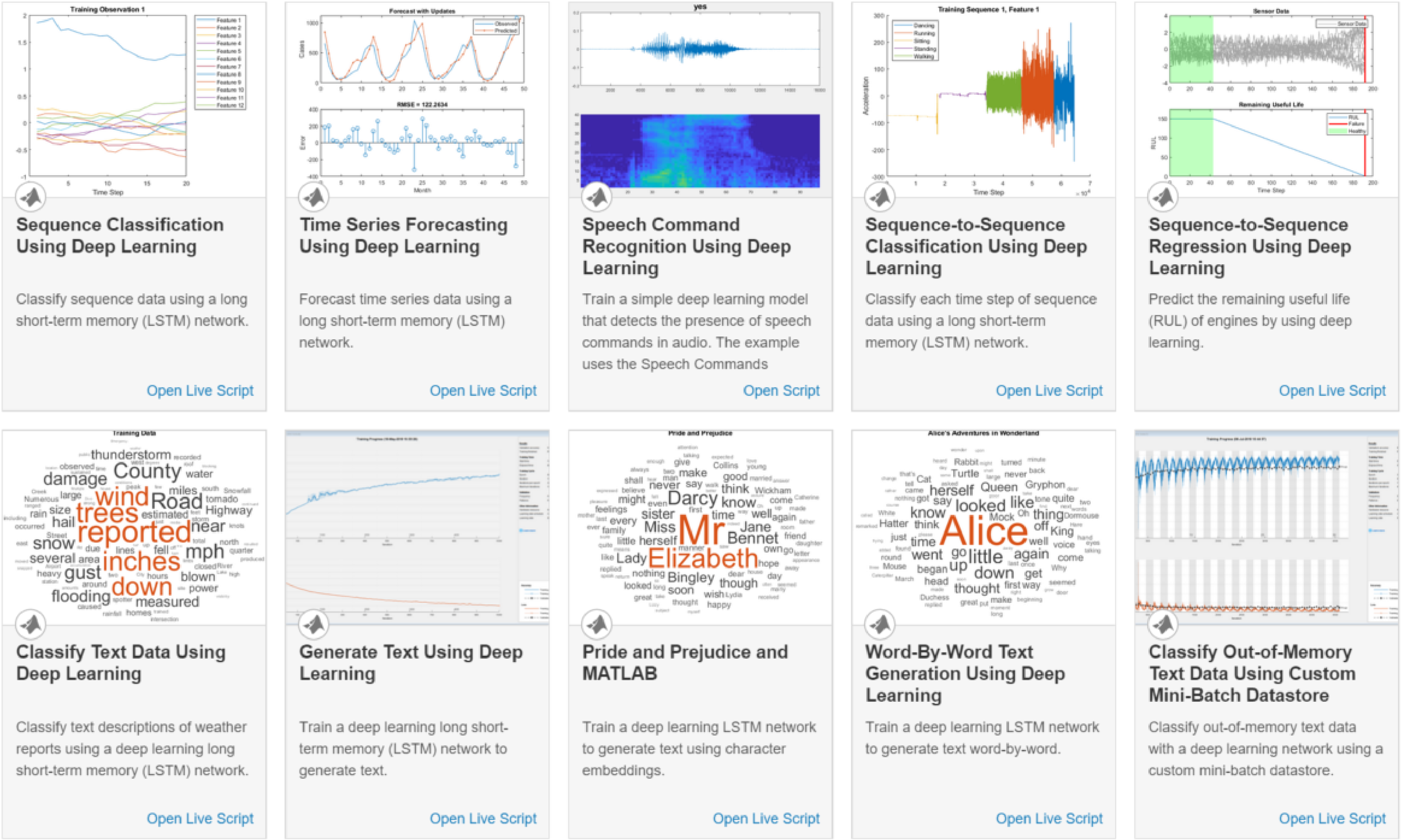
… I speak ____________ ?

Long Short-Term Memory Network

Recurrent Neural Network that carries a memory cell (state) throughout the process



Examples in MATLAB Documentation



Sequence Classification Using Deep Learning

Classify sequence data using a long short-term memory (LSTM) network.

Open Live Script

Time Series Forecasting Using Deep Learning

Forecast time series data using a long short-term memory (LSTM) network.

Open Live Script

Speech Command Recognition Using Deep Learning

Train a simple deep learning model that detects the presence of speech commands in audio. The example uses the Speech Commands

Open Script

Sequence-to-Sequence Classification Using Deep Learning

Classify each time step of sequence data using a long short-term memory (LSTM) network.

Open Live Script

Sequence-to-Sequence Regression Using Deep Learning

Predict the remaining useful life (RUL) of engines by using deep learning.

Open Live Script

Classify Text Data Using Deep Learning

Classify text descriptions of weather reports using a deep learning long short-term memory (LSTM) network.

Open Live Script

Generate Text Using Deep Learning

Train a deep learning long short-term memory (LSTM) network to generate text.

Open Live Script

Pride and Prejudice and MATLAB

Train a deep learning LSTM network to generate text using character embeddings.

Open Live Script

Word-By-Word Text Generation Using Deep Learning

Train a deep learning LSTM network to generate text word-by-word.

Open Live Script

Classify Out-of-Memory Text Data Using Custom Mini-Batch Datastore

Classify out-of-memory text data with a deep learning network using a custom mini-batch datastore.

Open Live Script

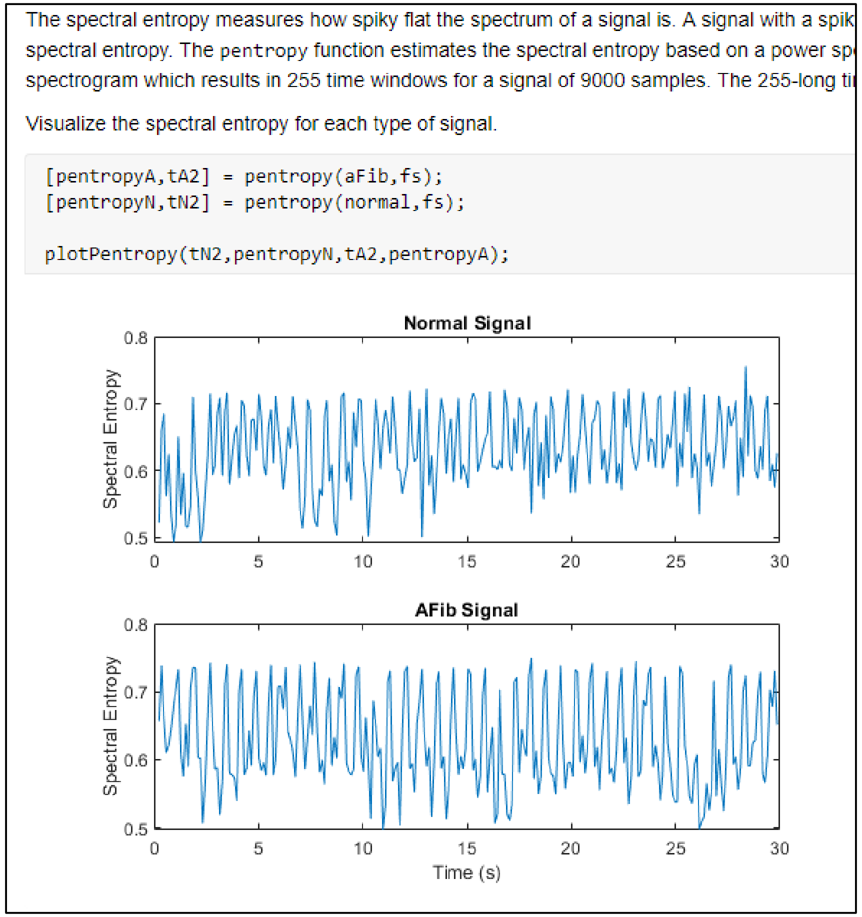
Exercise – ECG Signal Classification

Purpose:

- Use LSTM to classify ECG signal as normal heartbeat or Atrial Fibulation
- Perform preliminary feature engineering and view difference in results.

To Do:

1. Open work_ClassifyECGSignals.mlx.



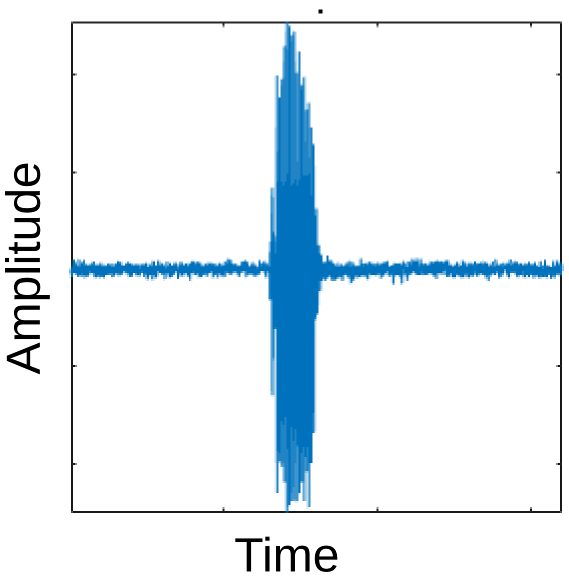
The spectral entropy measures how spiky flat the spectrum of a signal is. A signal with a spik
spectral entropy. The pentropy function estimates the spectral entropy based on a power sp
spectrogram which results in 255 time windows for a signal of 9000 samples. The 255-long ti

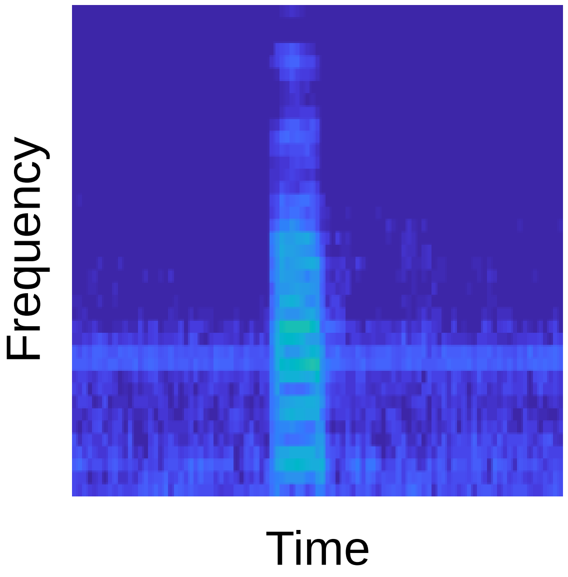
Visualize the spectral entropy for each type of signal.

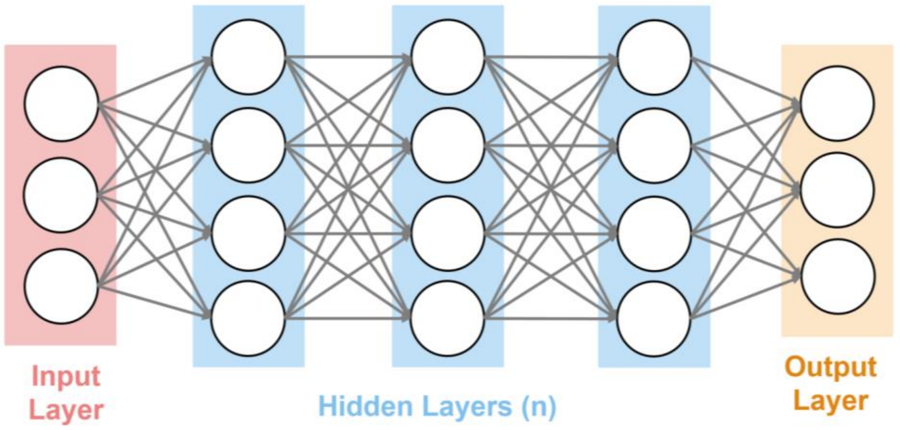
```
[pentropyA,tA2] = pentropy(aFib,fs);
[pentropyN,tN2] = pentropy(normal,fs);

plotPentropy(tN2,pentropyN,tA2,pentropyA);
```

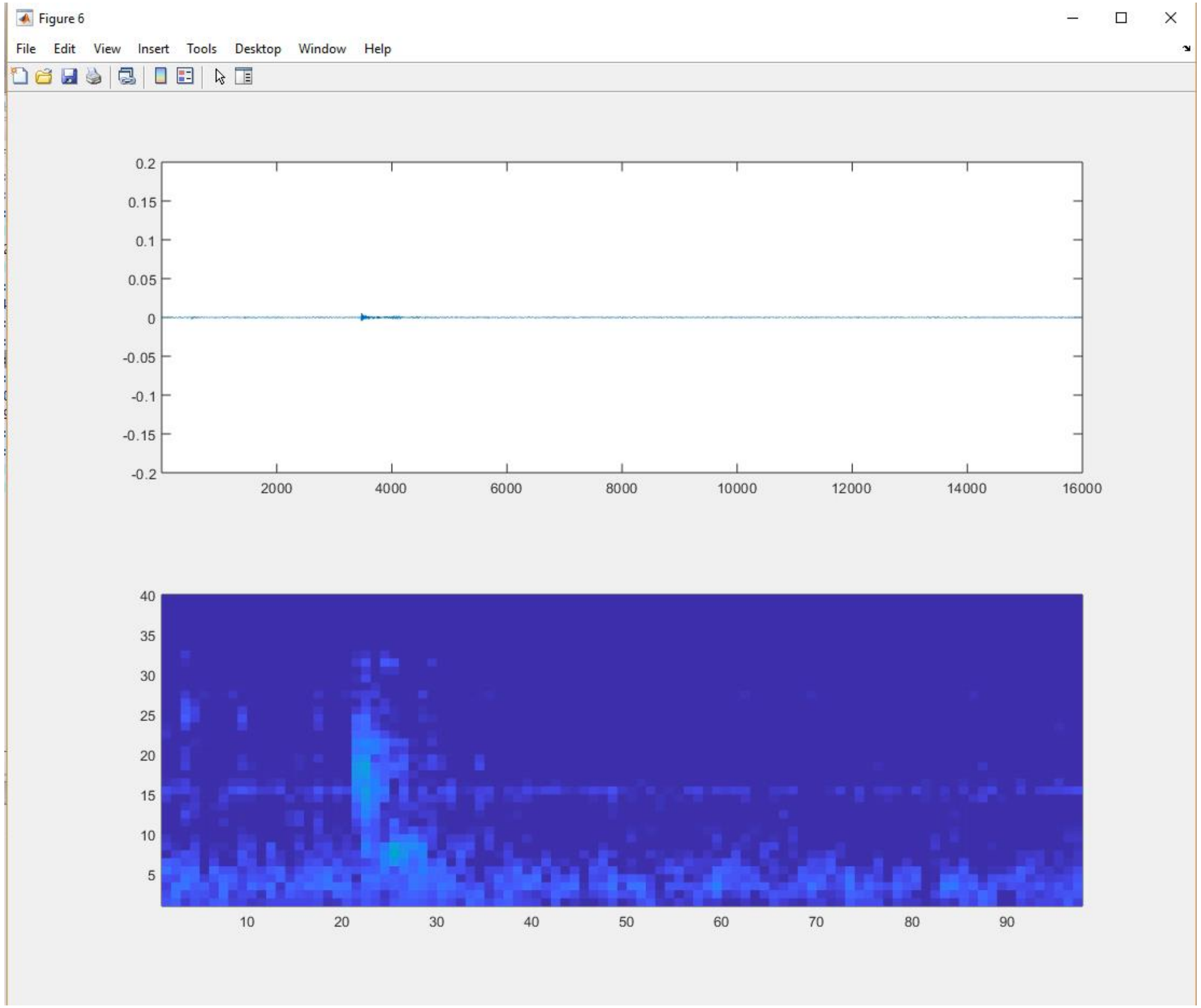
Speech Recognition Example

Audio signal → Spectrogram → Image Classification algorithm









Speech Recognition using CNNs

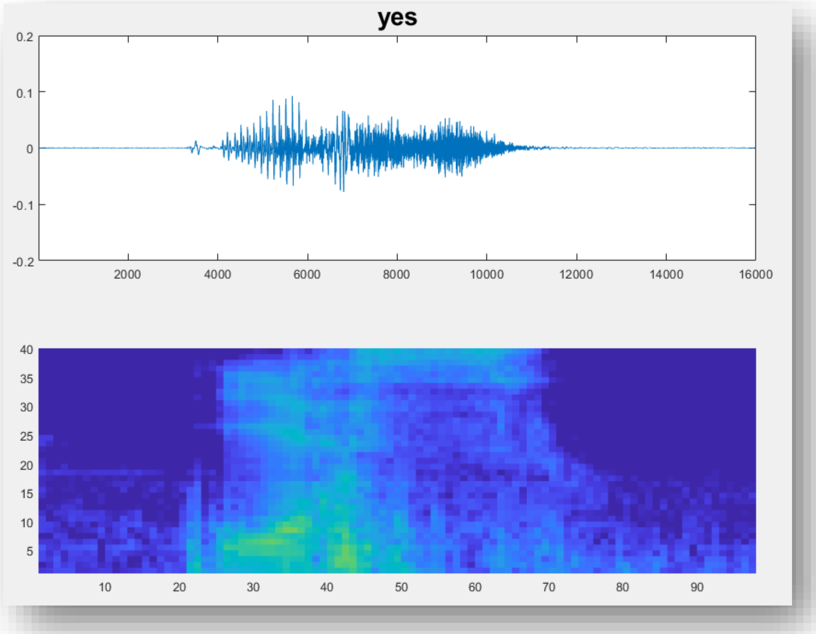
Exercise: Speech Command Recognition with Deep Learning

- Train a Convolutional Neural Network (CNN) to recognize speech commands

- Work with Google's speech command dataset

- Leverage:
  - `audioDatastore` (Read and manage large datasets)
  - `melSpectrogram` (Transform 1D signals into 2D images using perceptually-spaced frequency scaling)

- Prototype trained network in real-time on live audio


yes