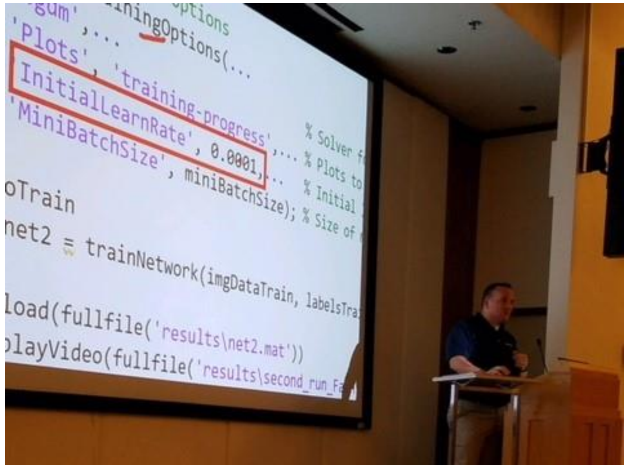
Hands-on Virtual Lab: Deep Learning



Reece Teramoto
Application Engineer


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

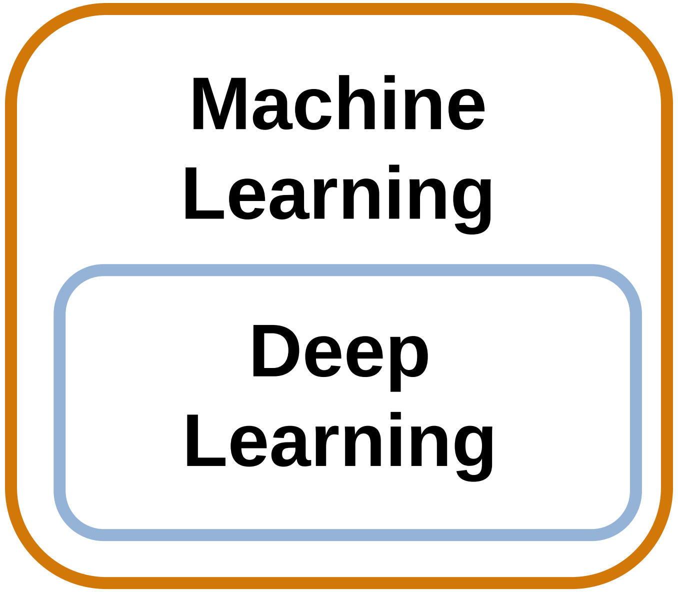
Exercise 4: Transfer Learning – OR – Signal Classification Exercise

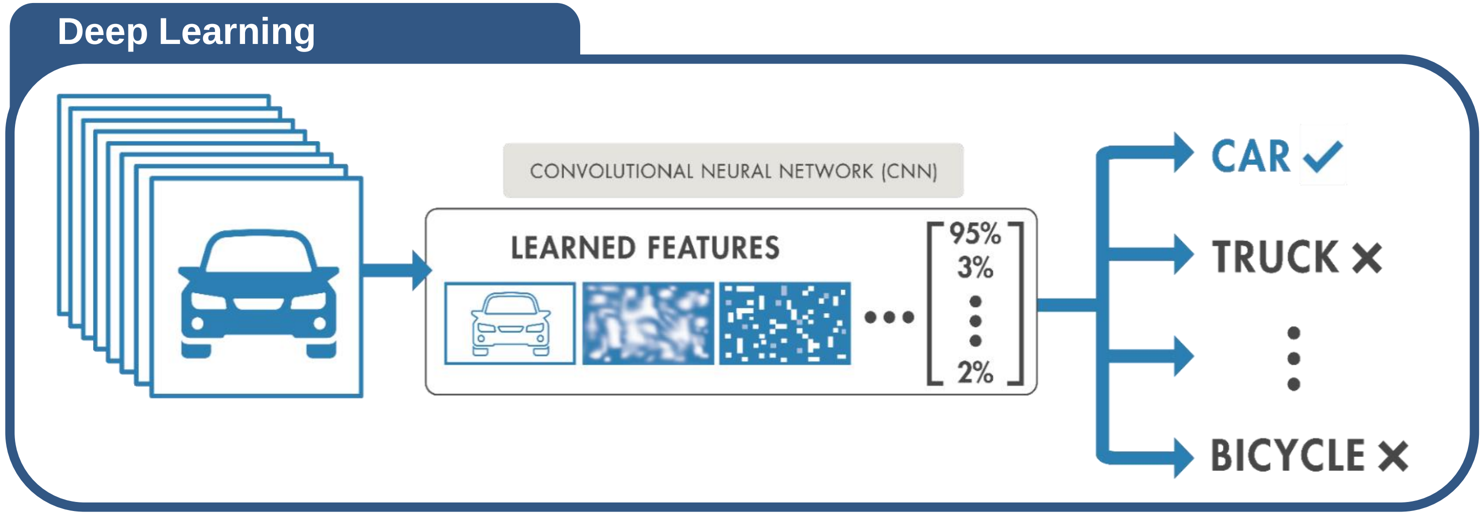
Optional: Deploying Deep Networks– OR – Improving Network Accuracy

Conclusion

What is Deep Learning?

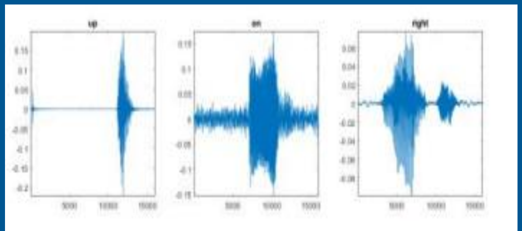
- Subset of machine learning with automatic feature extraction
  - Learns features and tasks directly from data
- Accuracy can surpass traditional ML Algorithms

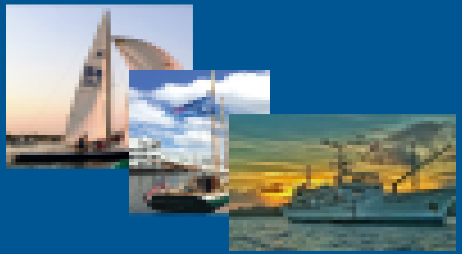




Deep Learning Workflow

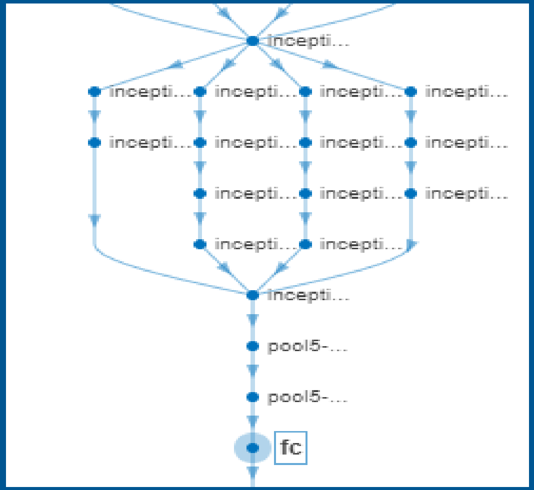
PREPARE DATA





The data must be labeled and preprocessed to give accurate results

TRAIN MODEL



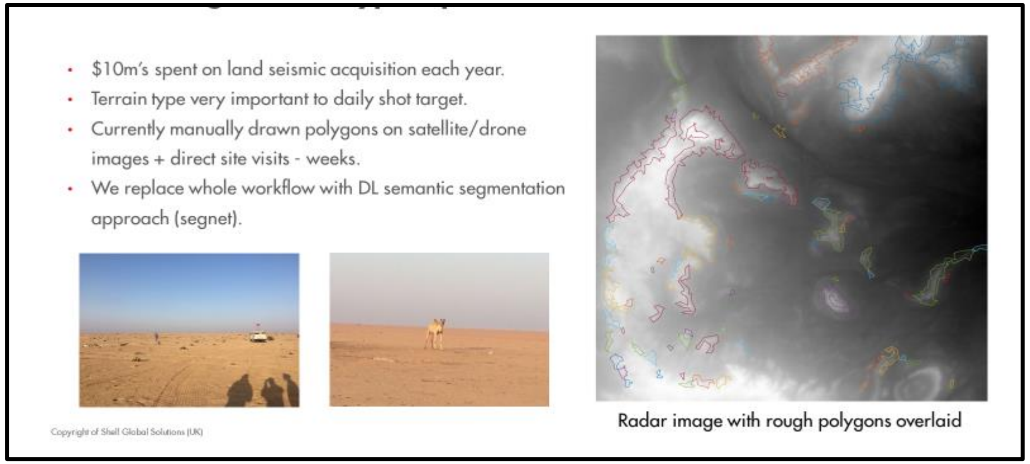
Build a neural network that learns from your dataset

DEPLOY SYSTEM

Integrate your trained model onto embedded hardware or cloud

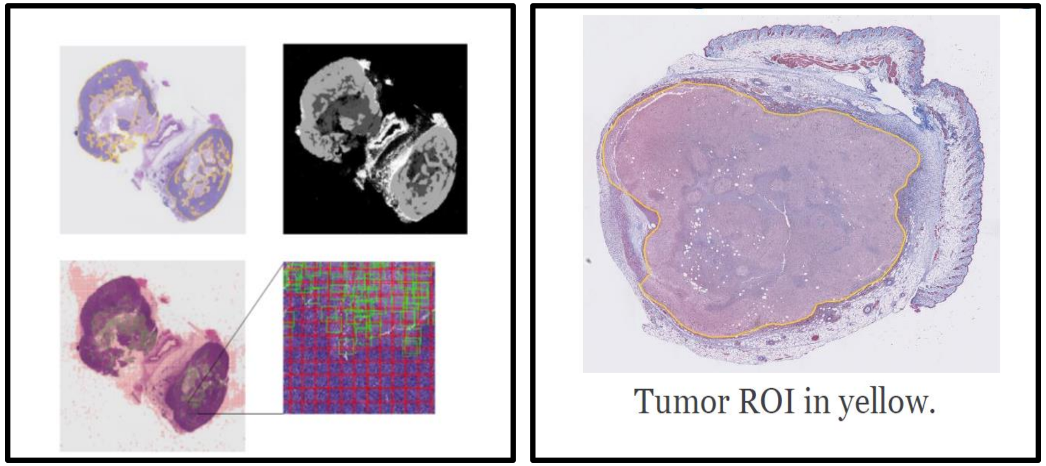
Deep Learning Examples

Shell



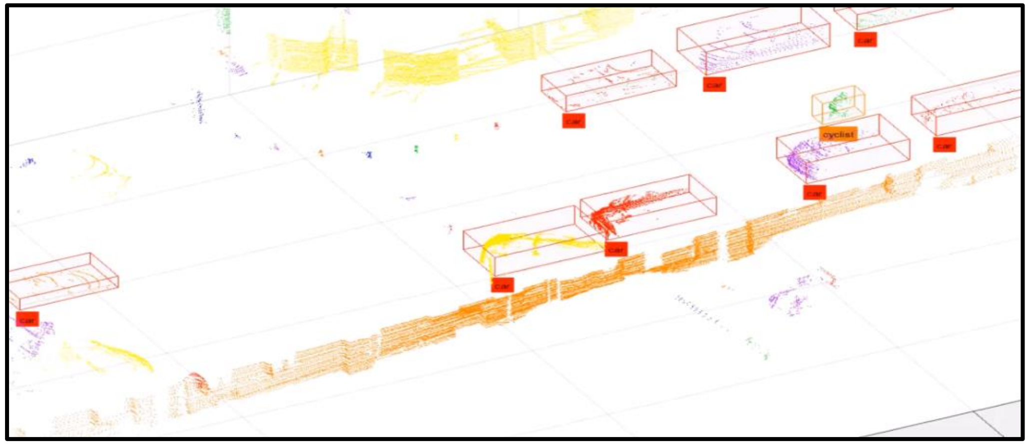
Terrain Recognition with Hyperspectral Data

Genentech



CNNs for Digital Pathology Analysis

Veoneer



LiDAR-Based Sensor Verification

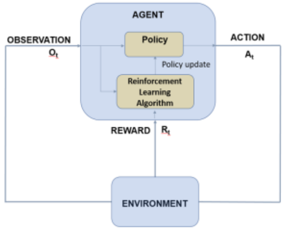
Caterpillar

Equipment Classification

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

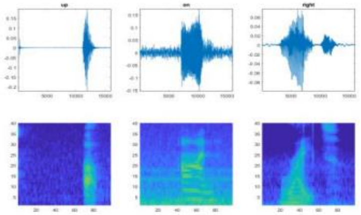
Computer Vision

Image Processing

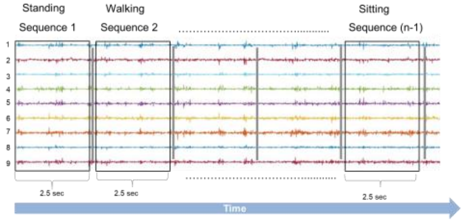


Control Design

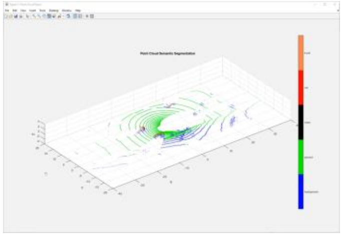
Text Analytics



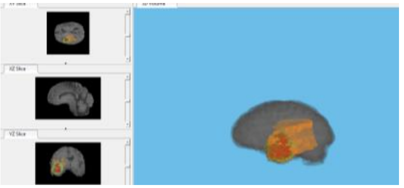
Audio Processing



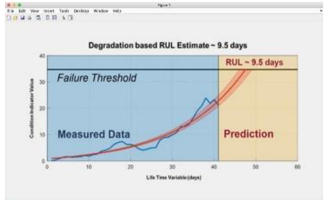
Sensor Data Analysis



Lidar Processing



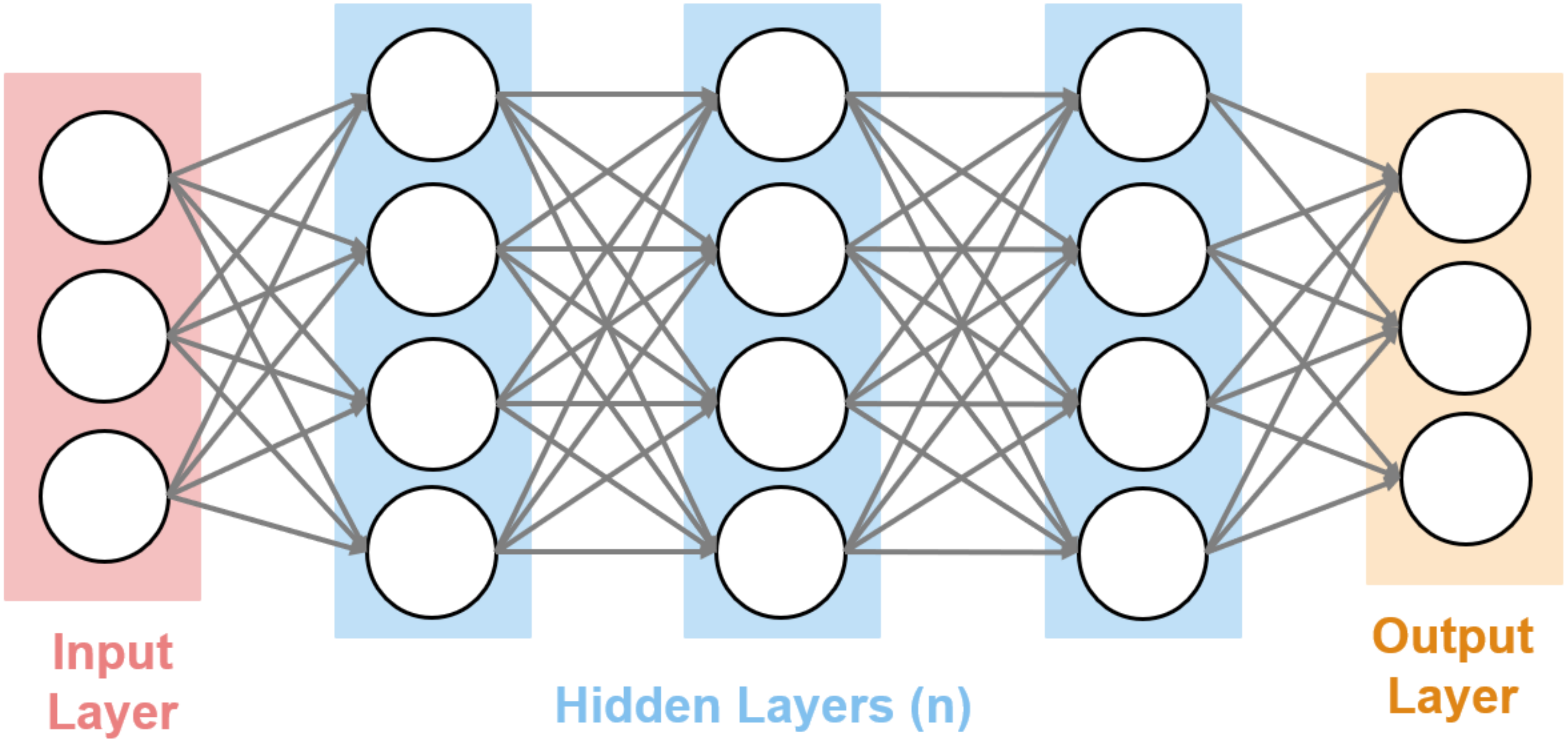
N-D Volumes



Sensor Data Analysis

Deep Learning Models are Neural networks

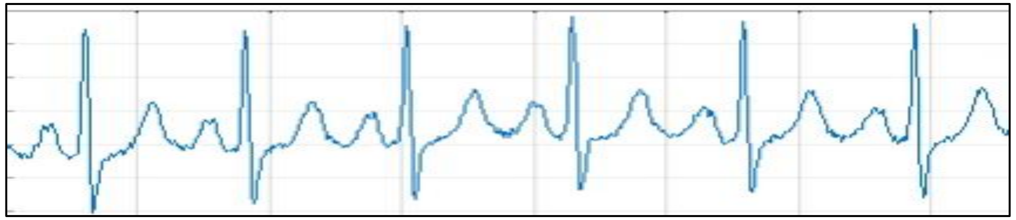
- Deep neural networks have many layers
- Data is passed through the network, and the layer parameters are updated (training)



Deep Learning Networks Take in Numeric Data

199	206	208	201	188	178	165	164	180
202	205	202	188	176	169	178	186	183
203	206	189	178	181	183	182	154	87
203	192	184	186	177	167	153	181	192
191	182	176	166	153	141	136	180	227
166	165	154	154	138	137	169	170	211
158	150	145	183	144	156	158	154	179
143	51	98	144	129	130	143	178	123
107	50	33	95	152	173	192	159	87
104	100	84	120	132	172	131	64	94
119	101	97	81	90	109	87	106	111
127	122	110	97	108	120	133	131	134
111	117	108	119	131	143	146	141	156
126	122	113	119	139	142	155	161	151
129	126	130	111	103	130	149	149	156
138	128	136	144	136	129	134	122	145
154	133	134	141	168	150	126	127	151

Images are a numeric matrix



131	158	185	212	239	266	293	320	347	374	376

Signals are numeric vectors

The Bird Flies = [0 13 5 6]

The Leaf Is Brown = [13 3 11 2]

Text is processed as numeric vectors

[Deep Learning Documentation]

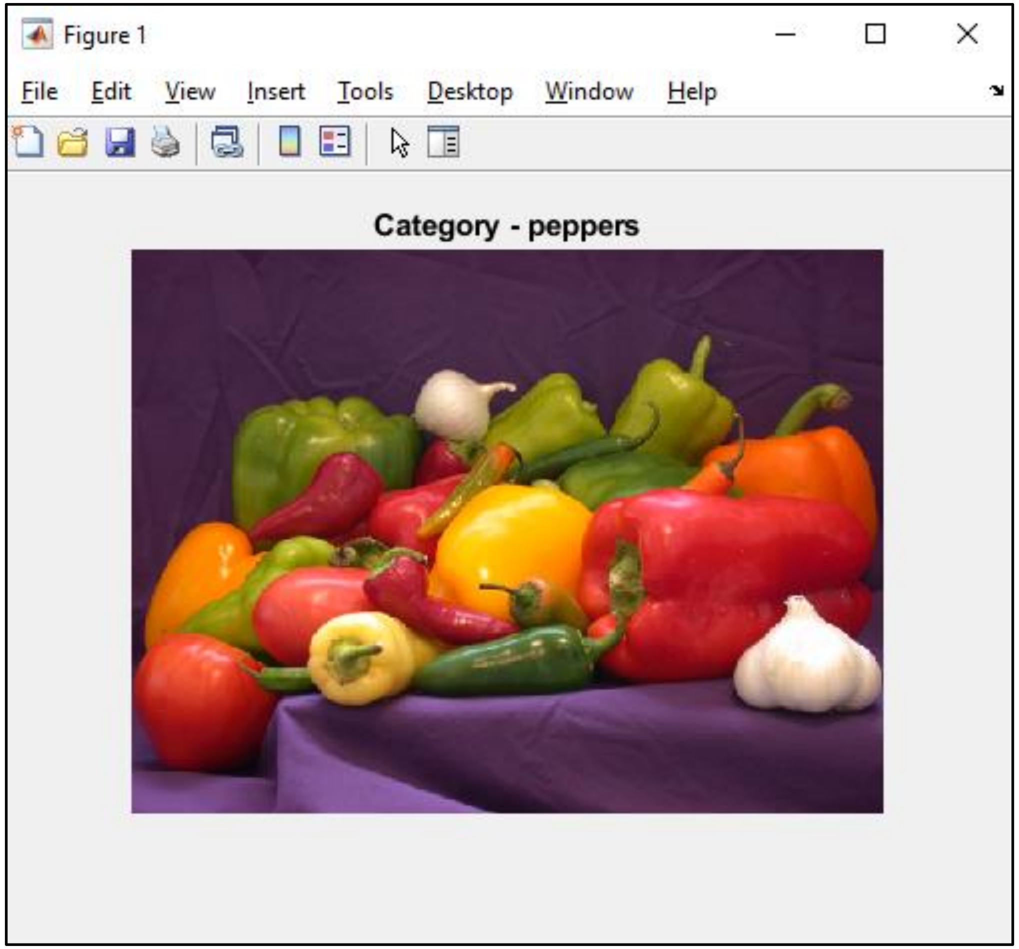
Exercise 1 – Deep Learning in 6 Lines of Code

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



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 **Inception-v3**
ResNet-101 **DenseNet-201**
ResNet-50 **Xception**

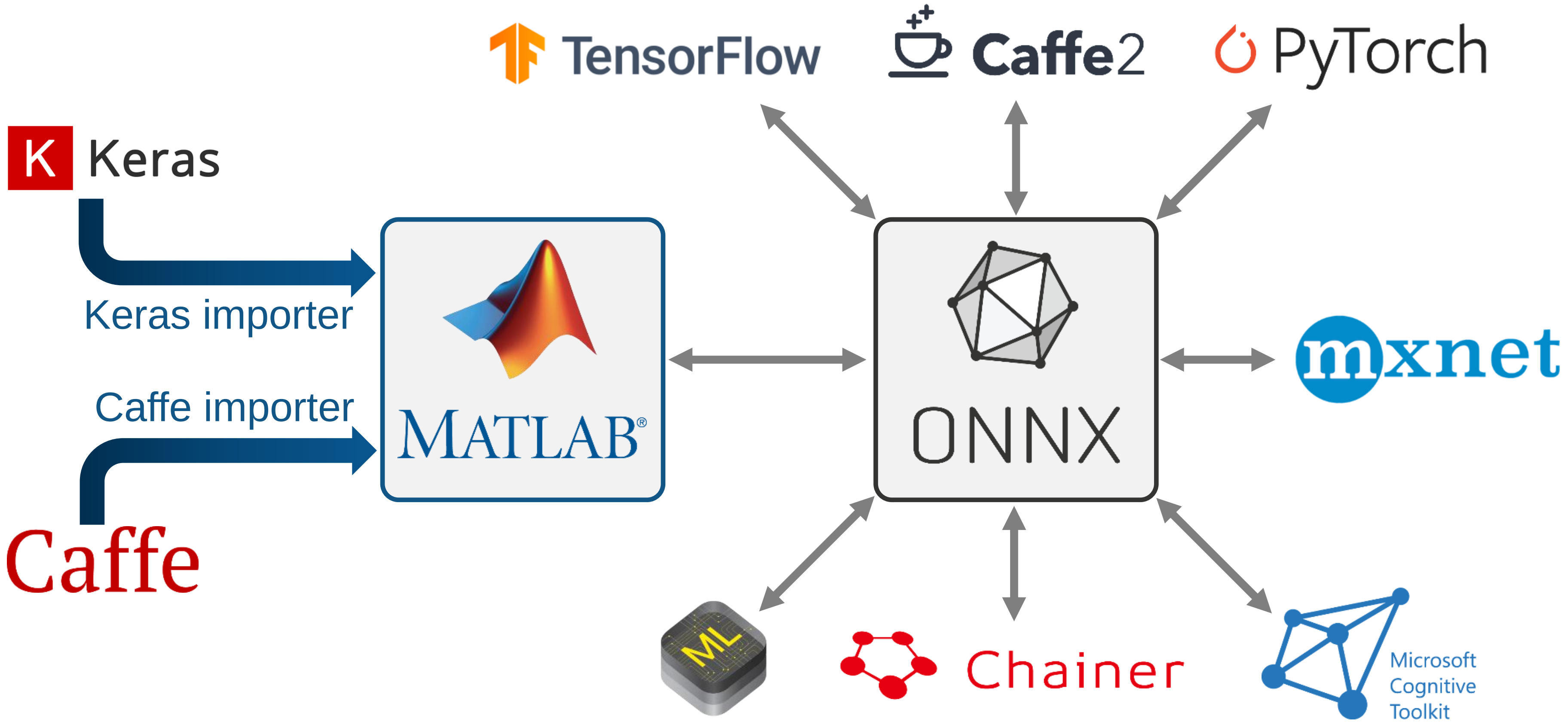
Effective for object detection and semantic segmentation workflows

SqueezeNet
MobileNet-v2
ShuffLeNet

Lightweight and computationally efficient

Full list of models available HERE

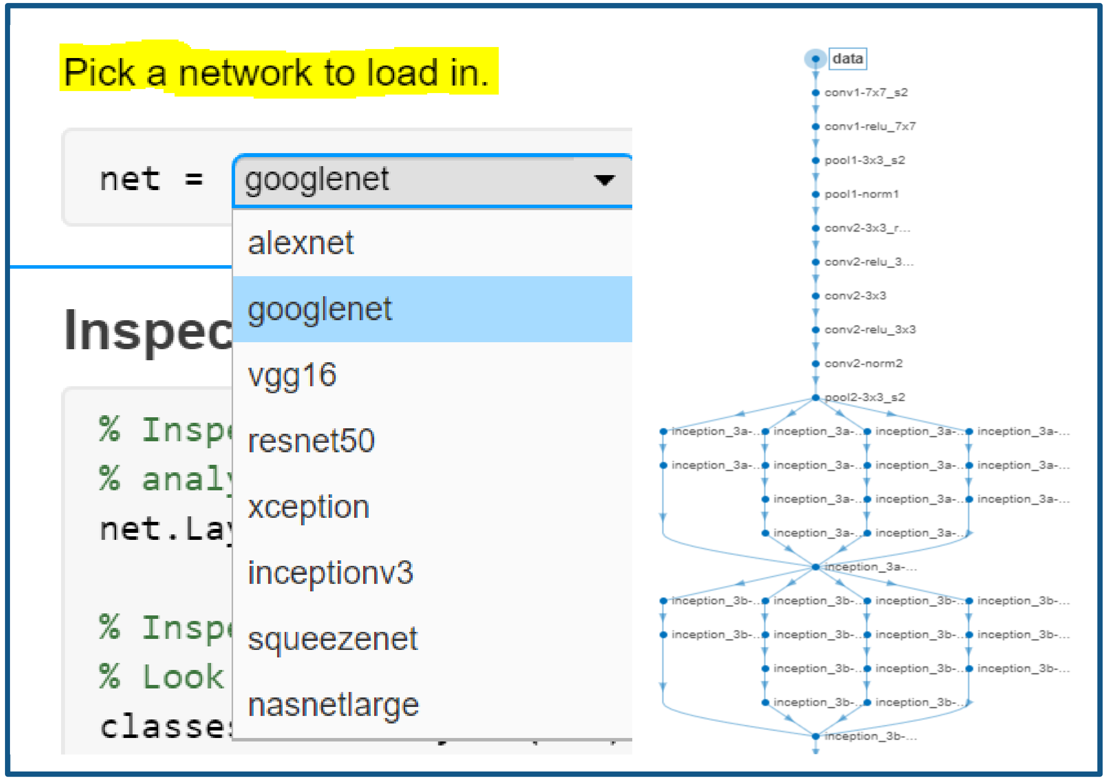
Access Pretrained Models from Within MATLAB or Import from the Web



Exercise 2 – Pretrained Models

Purpose:

- Classify Images using pretrained models.
- See how different network architectures affect results.
- Use datastores to access data efficiently

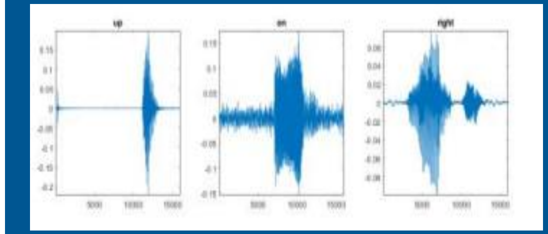


To Do:

1. Open work_pretrainednetworks.mlx.

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

PREPARE DATA



TRAIN MODEL

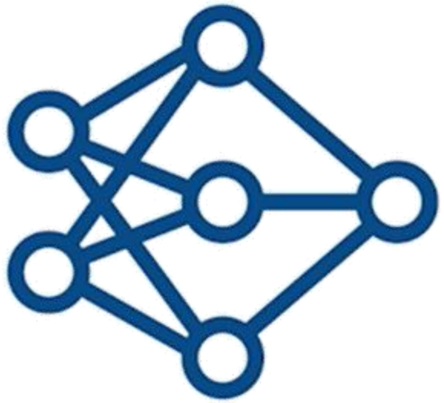
- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

DEPLOY SYSTEM

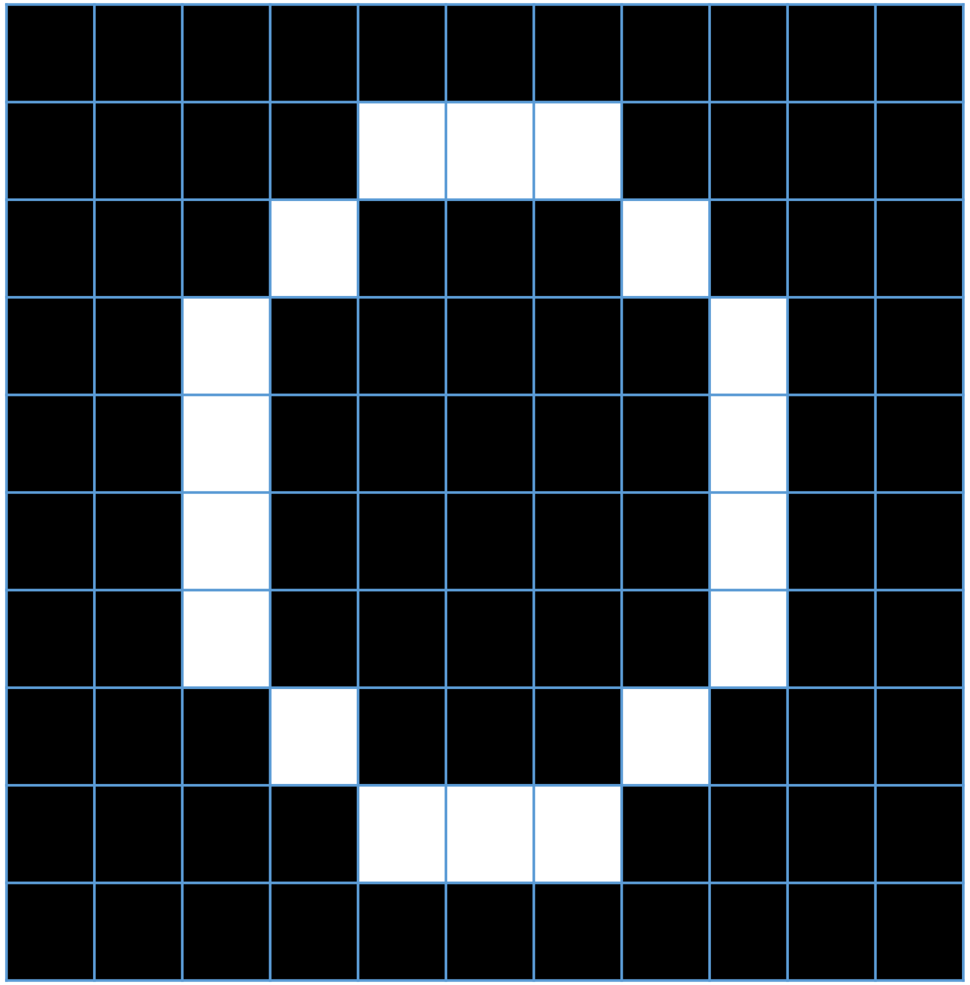
Creating Layer Architectures

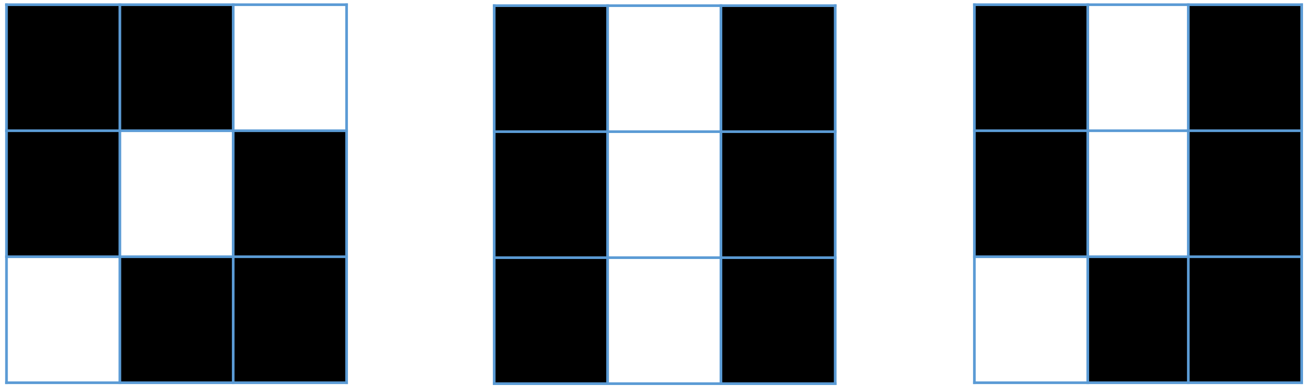
- Convolution Neural Networks – CNN
- Special layer combinations that make them adept at classifying images

- Convolution Layer
- ReLU Layer
- Max Pooling Layer

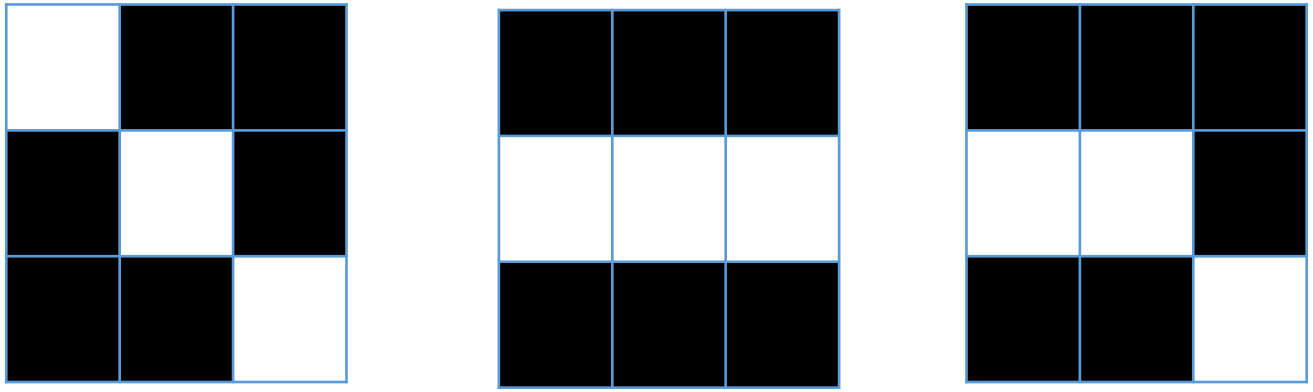


Convolution Layers Search for Patterns

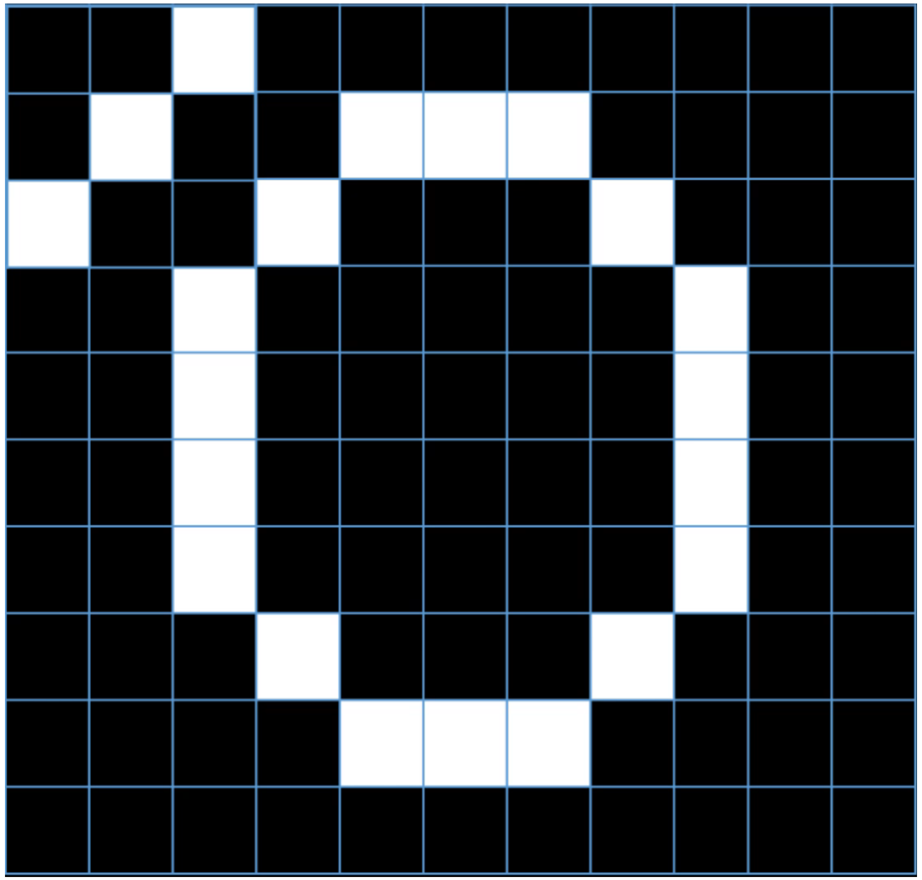


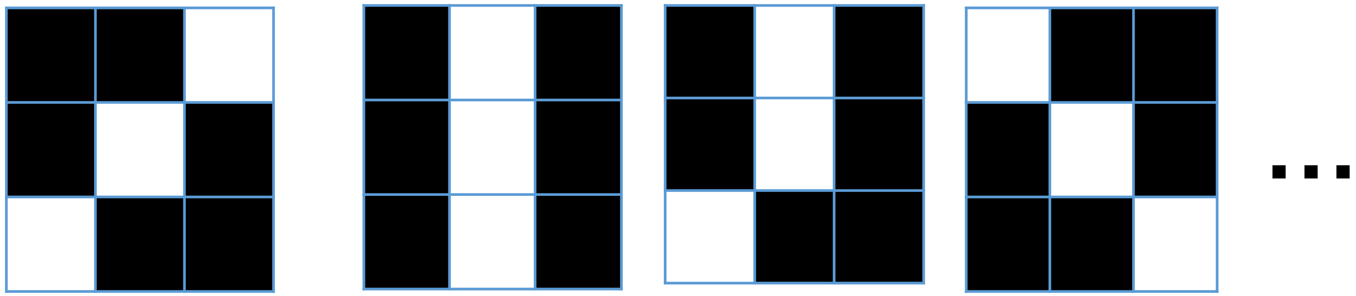


These patterns would be common in the number 0



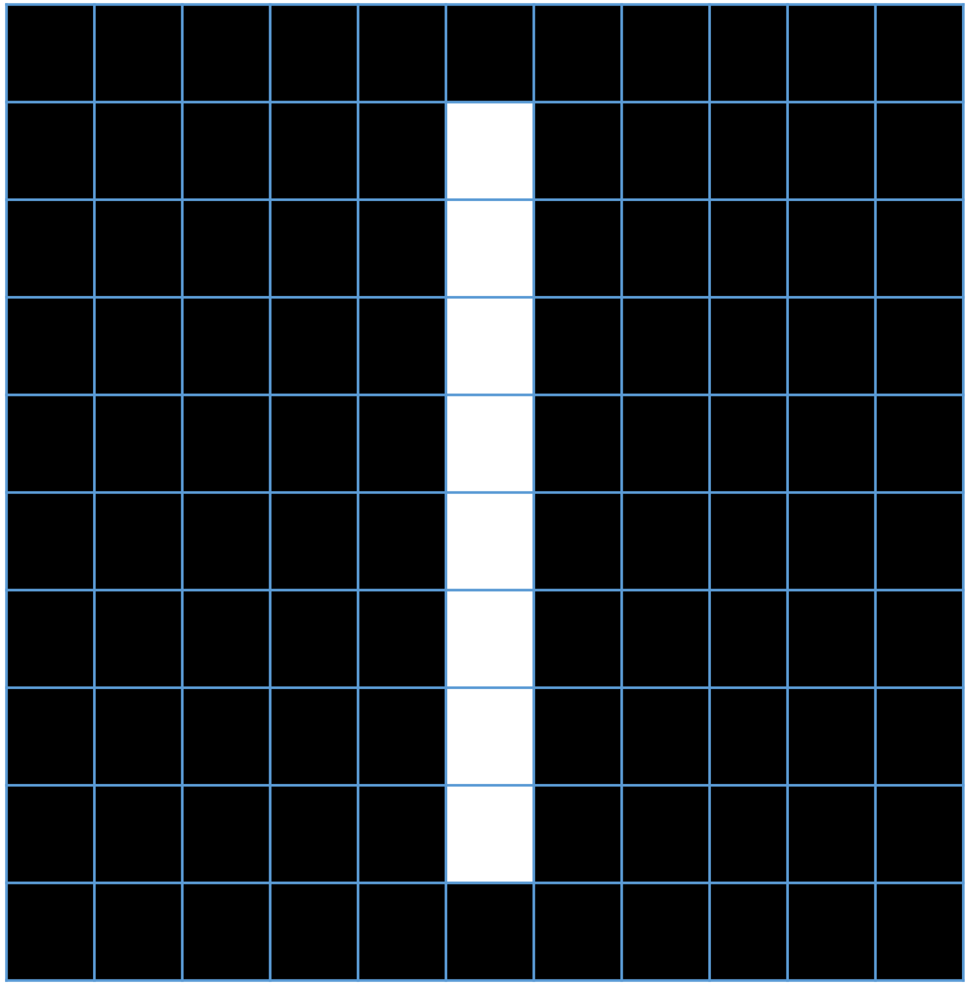
All patterns are compared to the patterns on a new image.





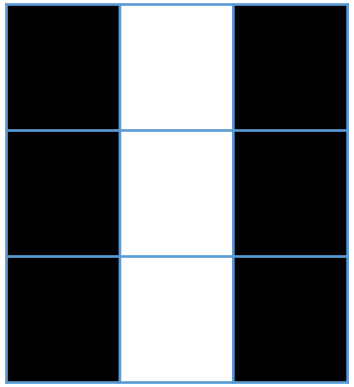
- Pattern starts at left corner
  - Perform comparison
  - Slide over one pixel
- Reach end of image
- Repeat for next pattern

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



Rectified Linear Units Layer (ReLU)

Converts negative numbers to zero

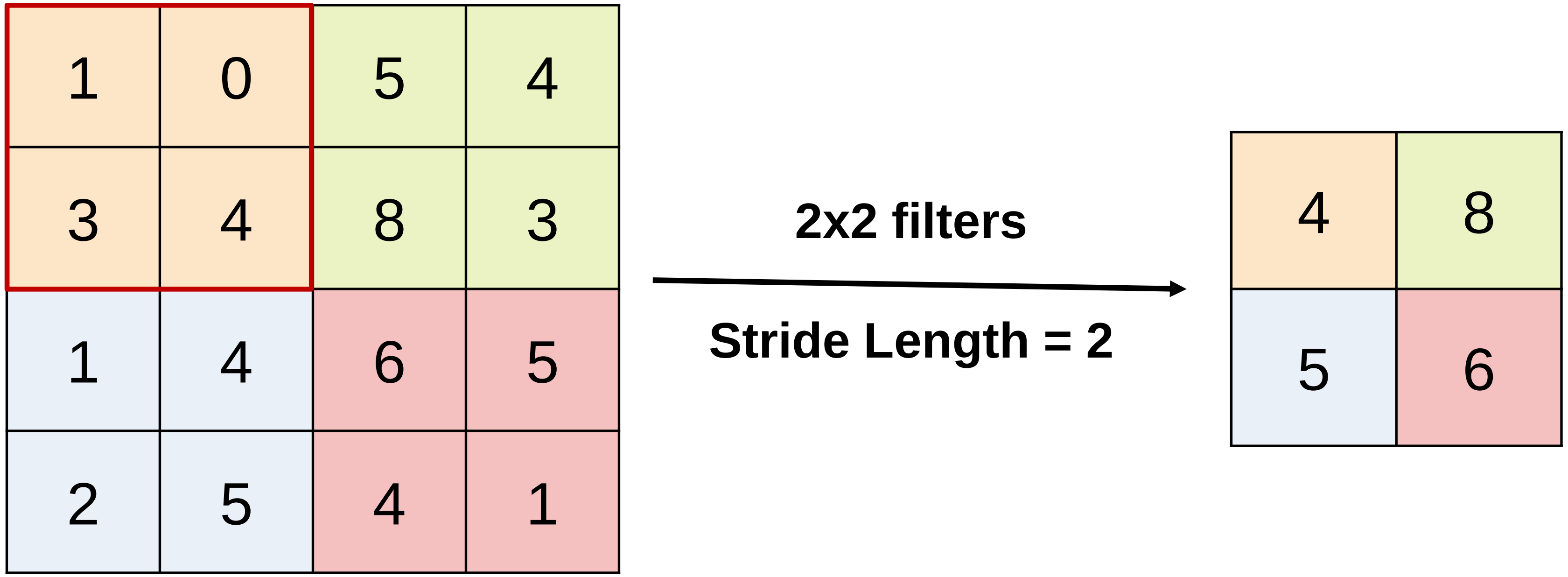
-1	0	5	4
3	-4	-8	3
1	4	6	-5
-2	-5	4	1

→

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

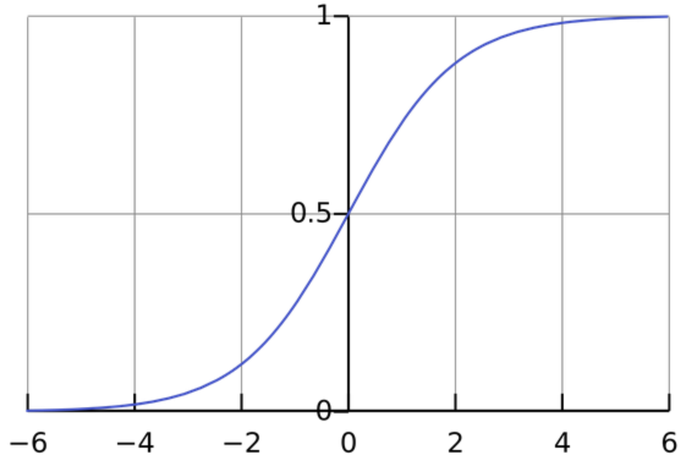
Max Pooling is a down-sampling operation

Shrink large images while preserving important information



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
- (…)

Research papers and doc examples can provide guidelines for creating architecture.

Documentation on Network Layers

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

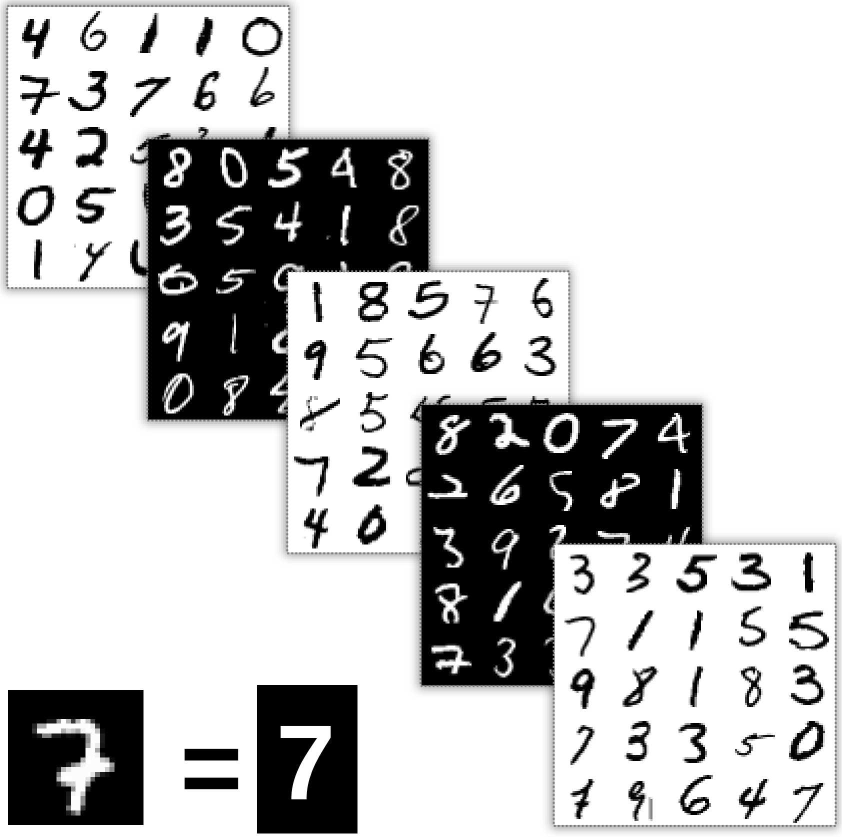
Influence training time and accuracy

- 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

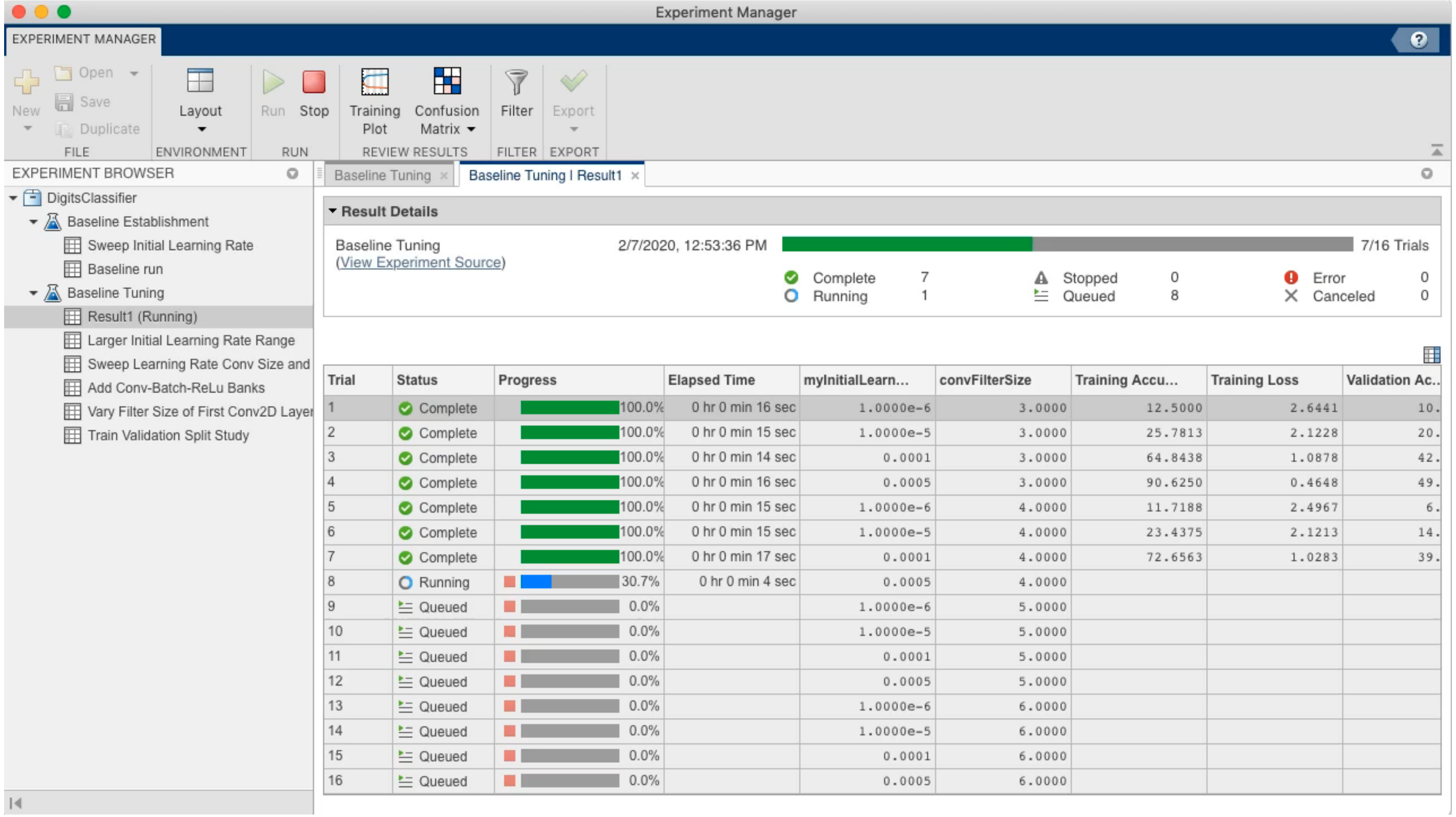


Details

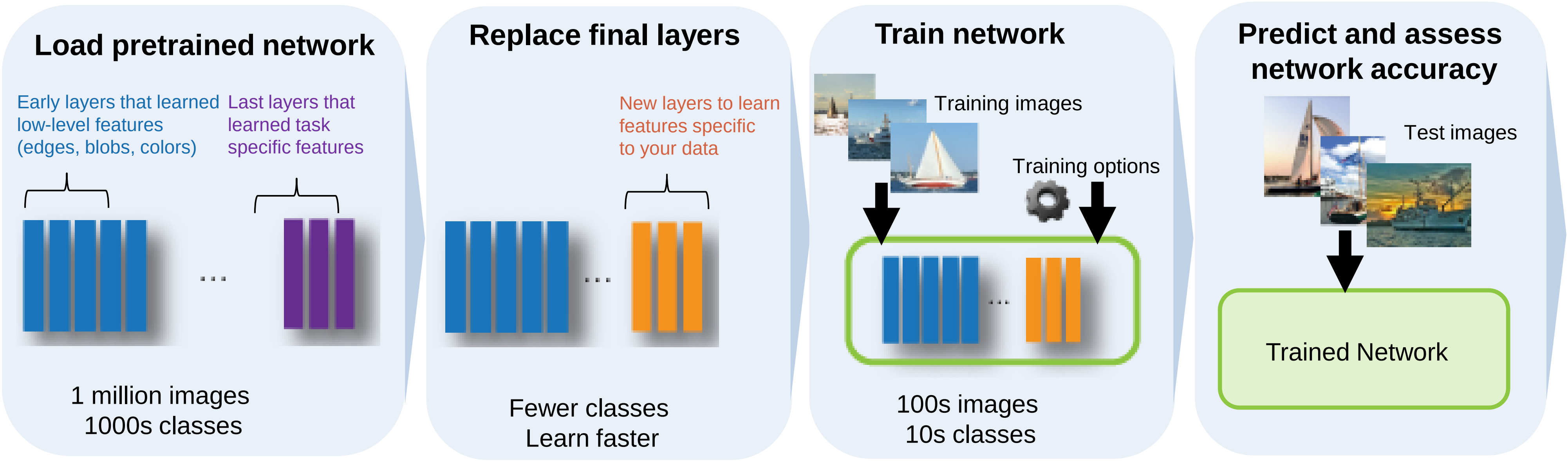
- Dataset consists of handwritten digits 0-9
- 60,000 training images
- 10,000 test images

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

Experiment Manager – Run, Track, and Analyze Multiple Deep Learning Experiments



Transfer Learning Workflow

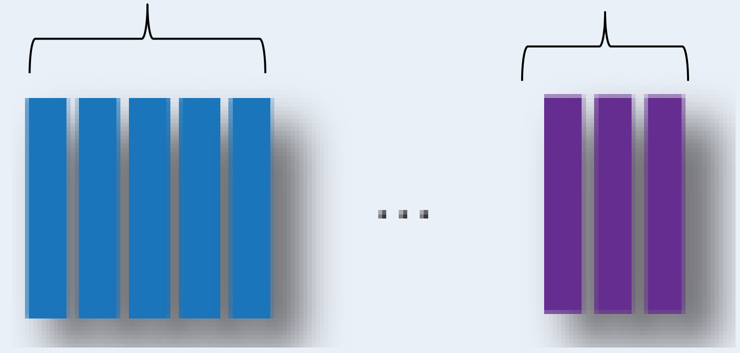


Transfer Learning Workflow – Step 1

Load pretrained network

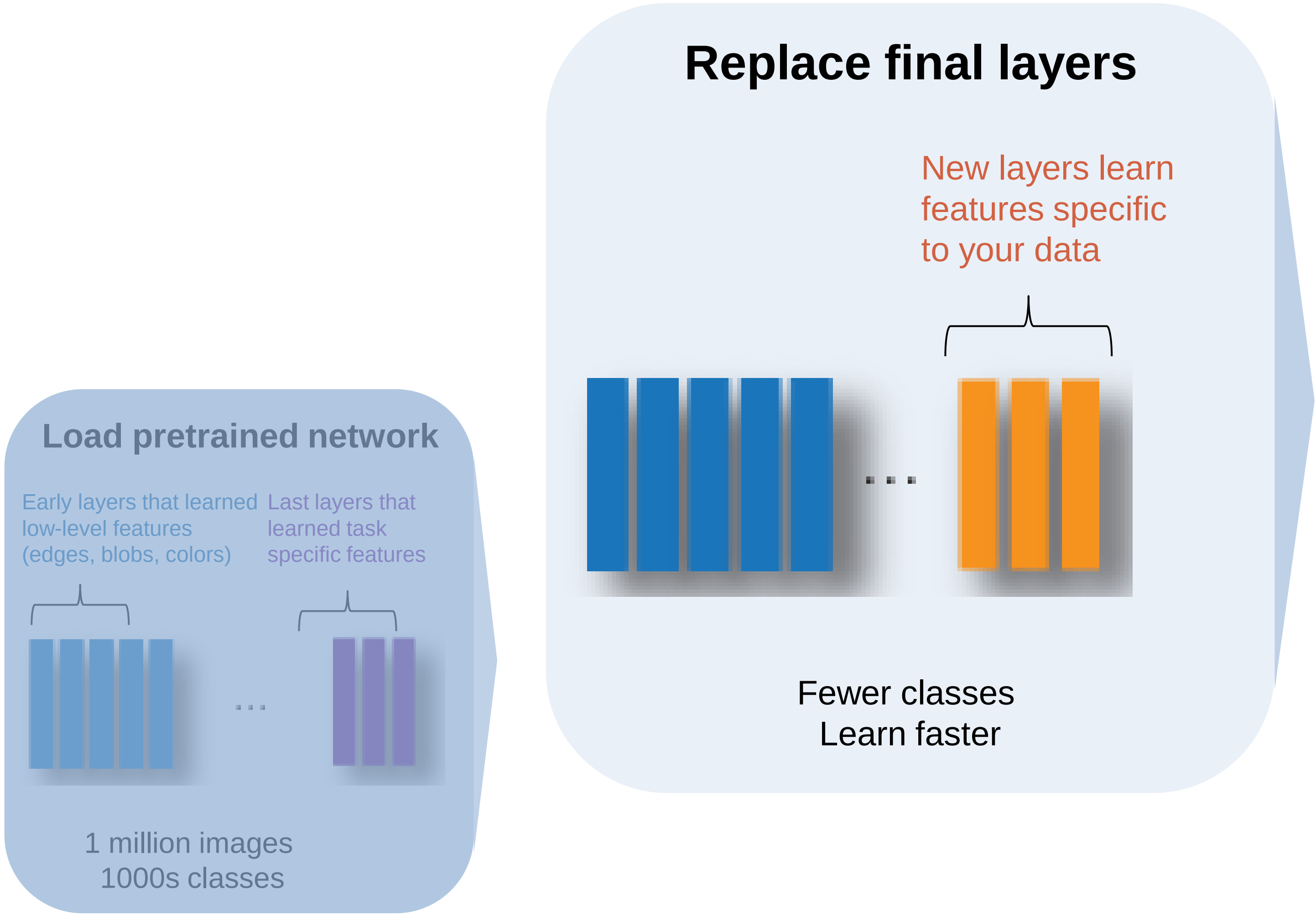
Early layers learn low-level features (edges, blobs, colors)

Last layers learn task-specific features

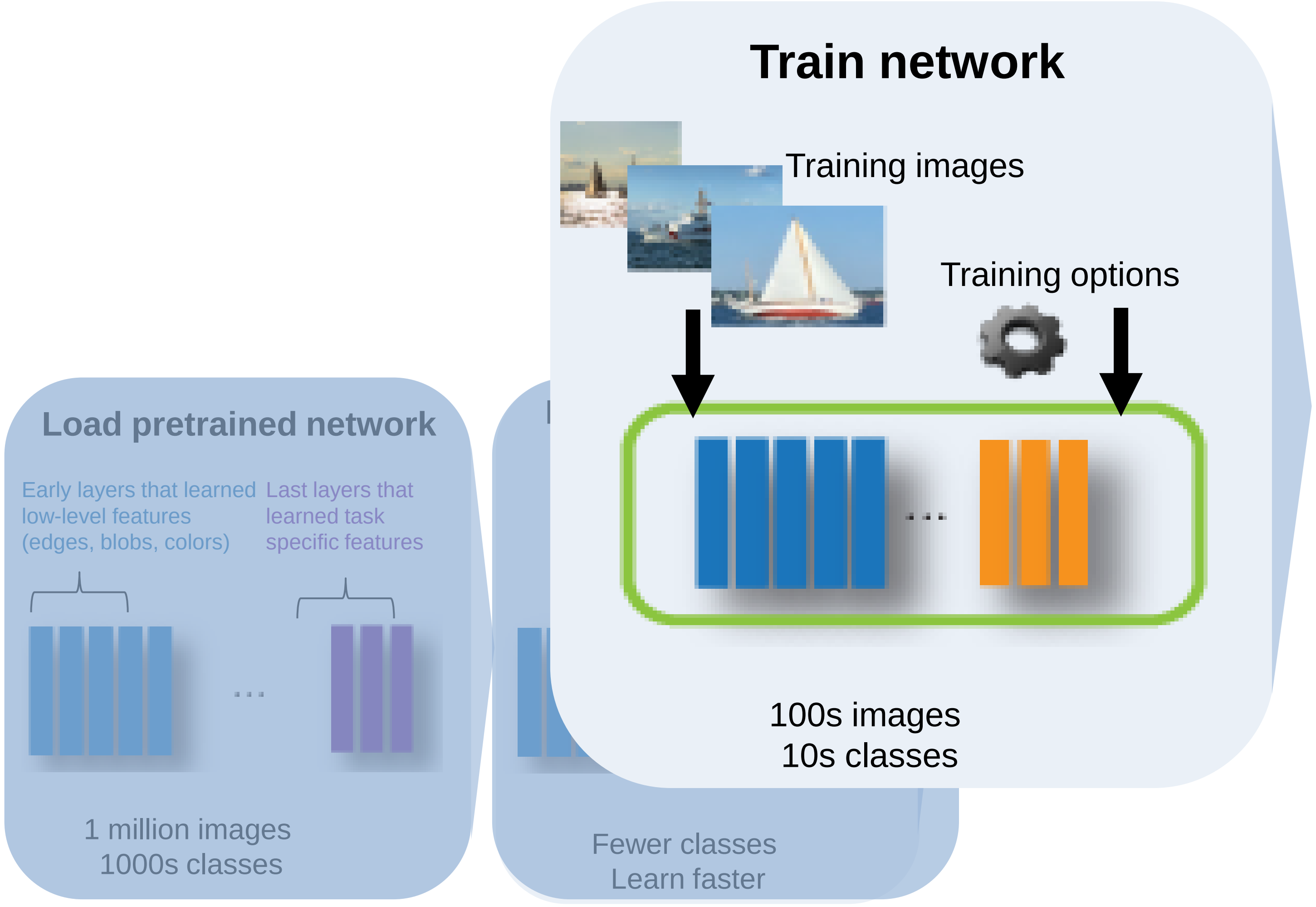


1 million images
1000s classes

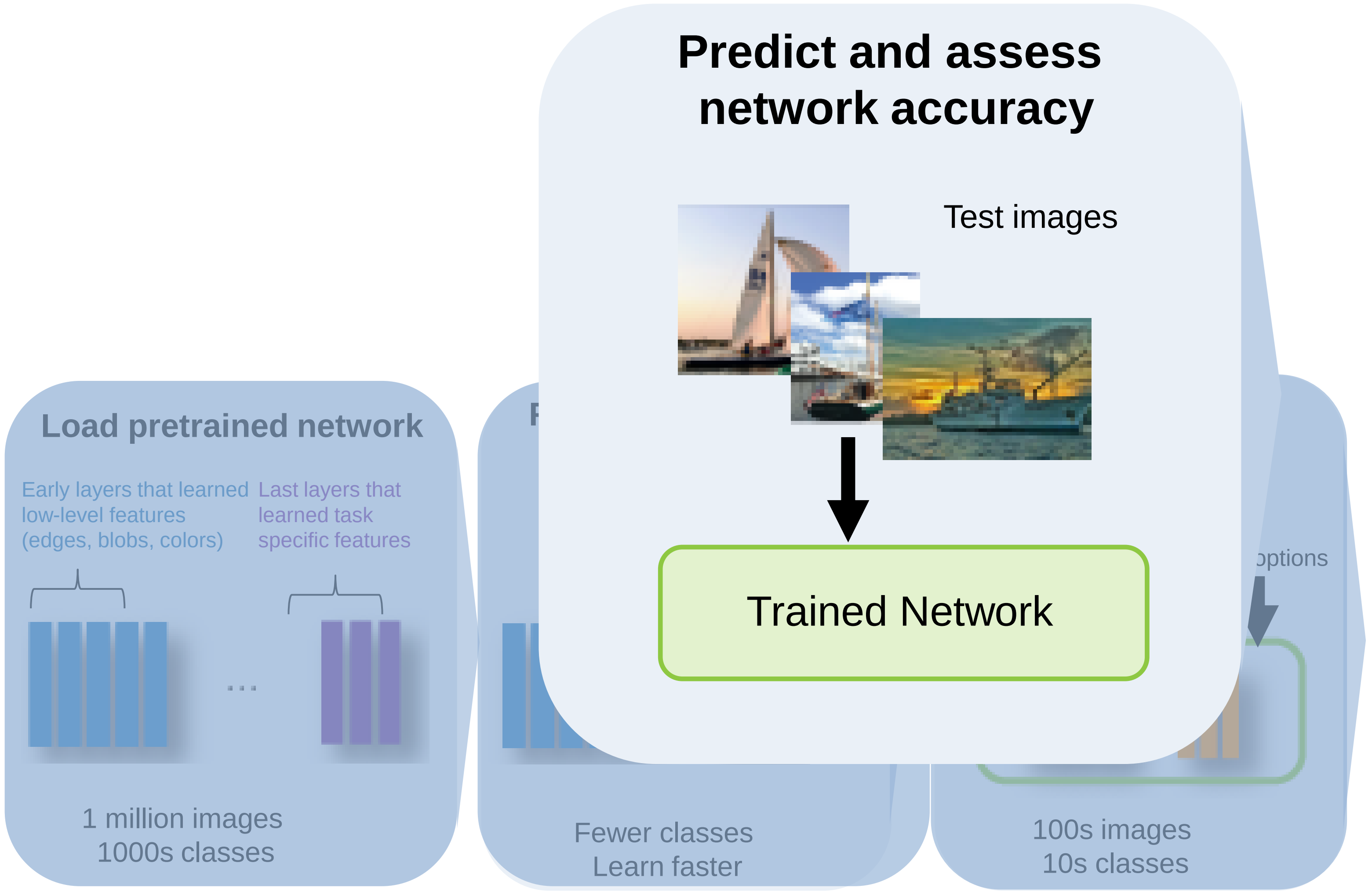
Transfer Learning Workflow – Step 2



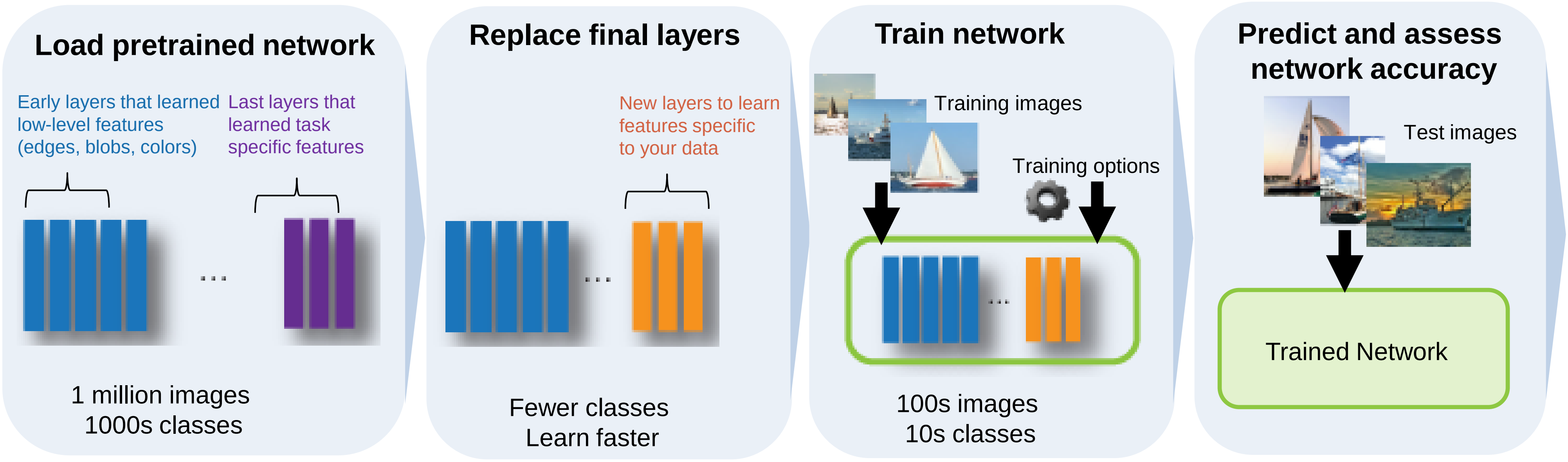
Transfer Learning Workflow – Step 3



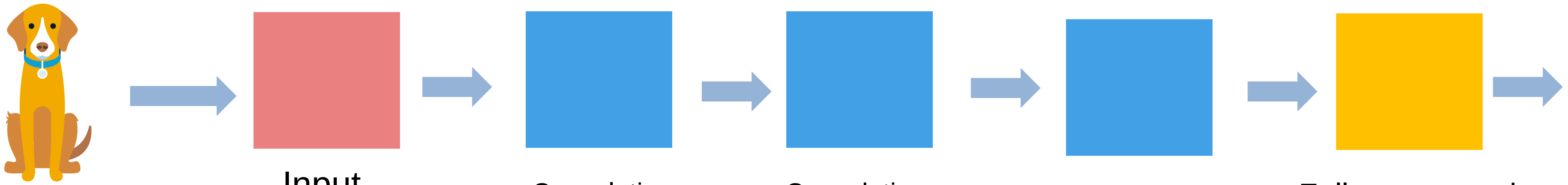
Transfer Learning Workflow – Step 4



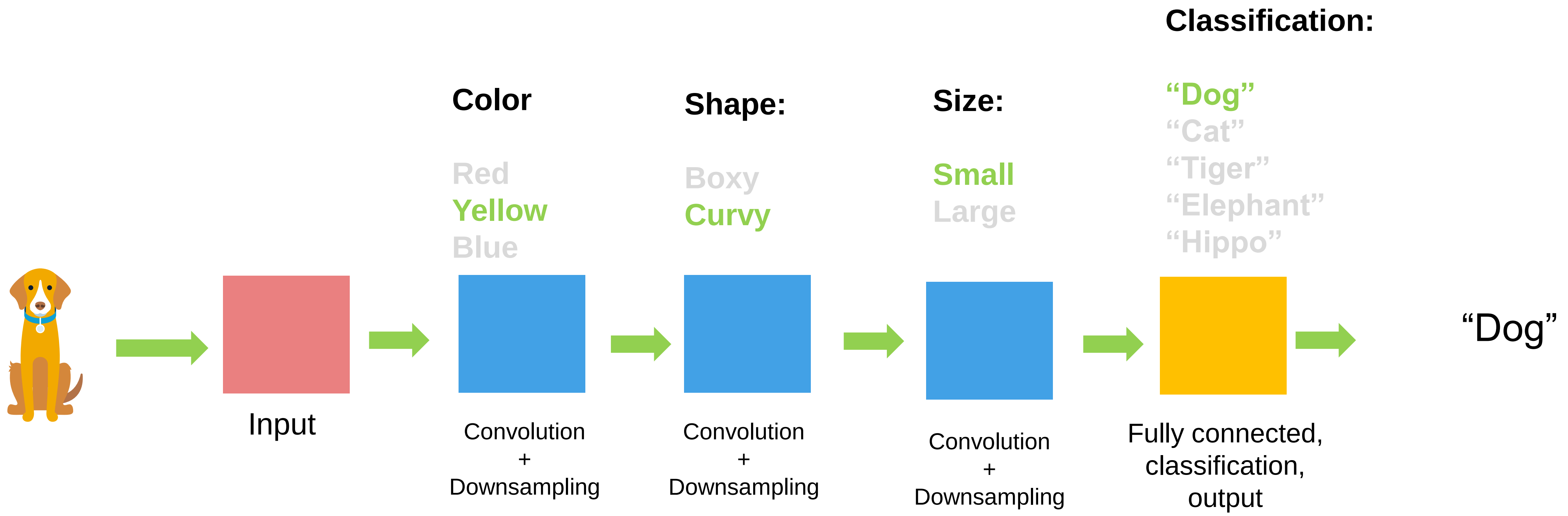
Transfer Learning Workflow



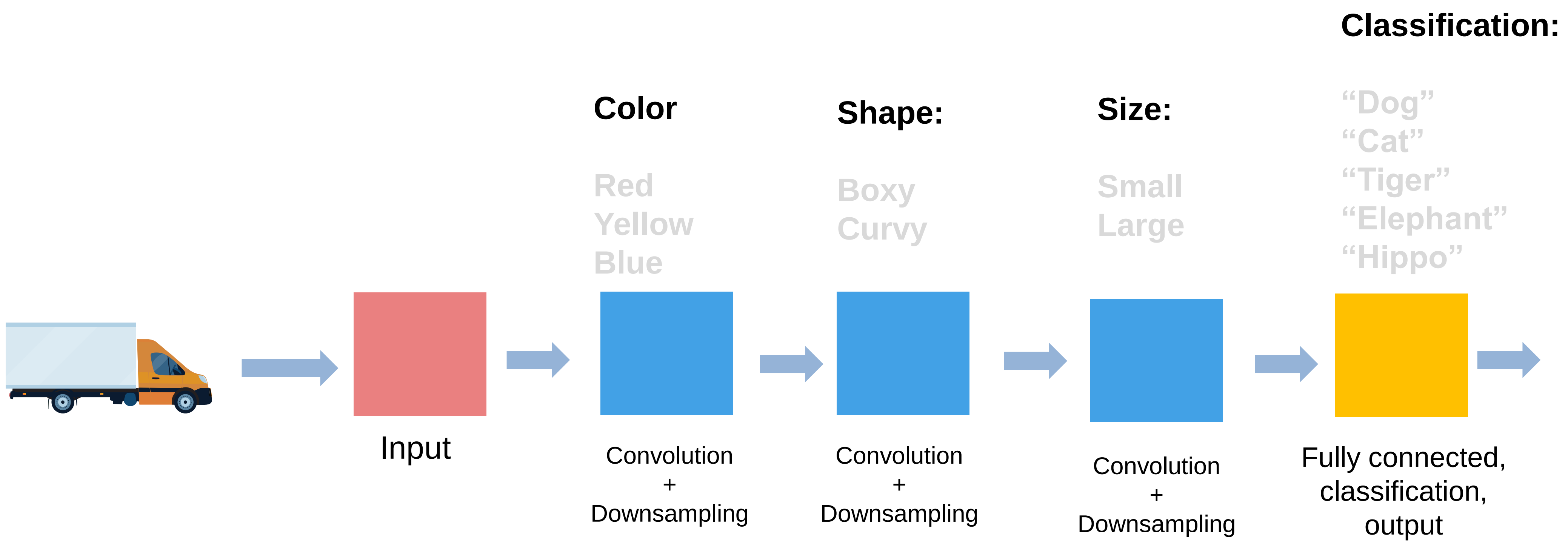
Transfer Learning



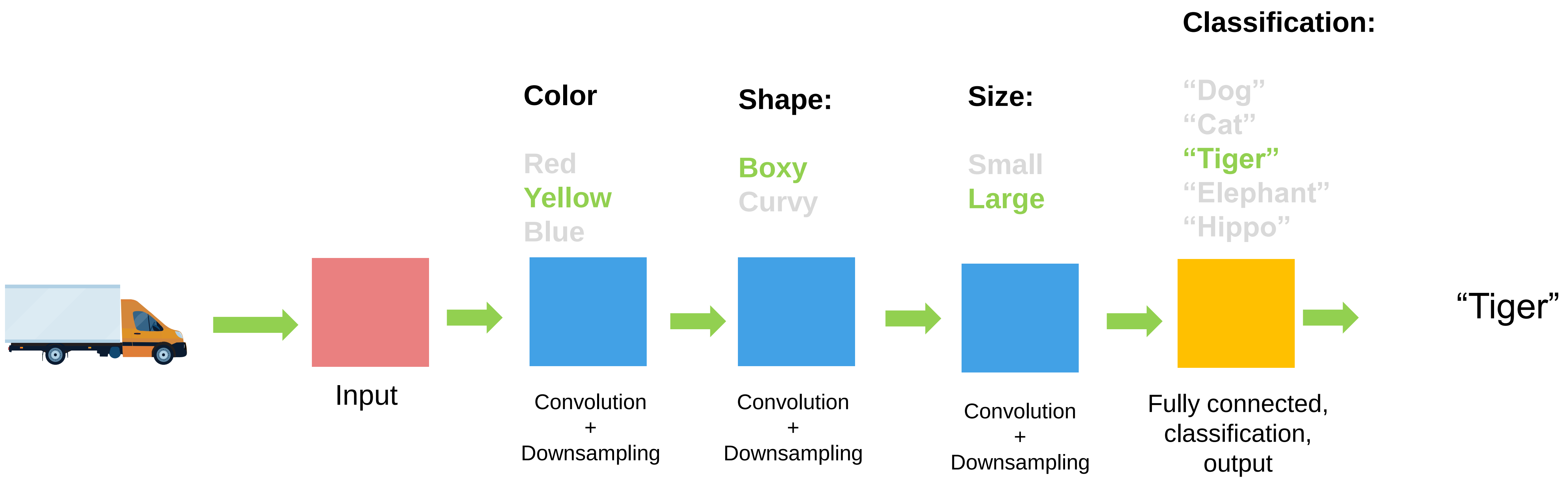
Transfer Learning



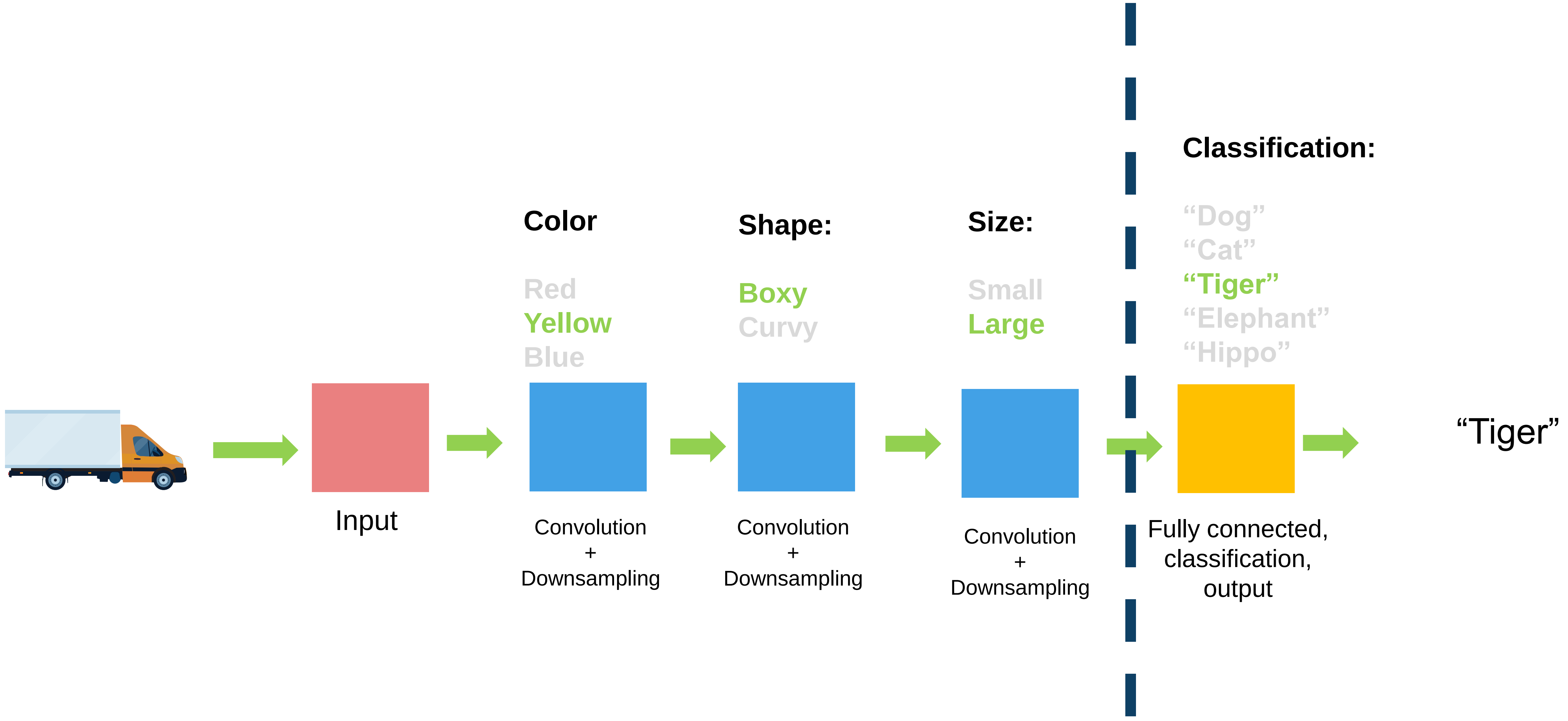
Transfer Learning



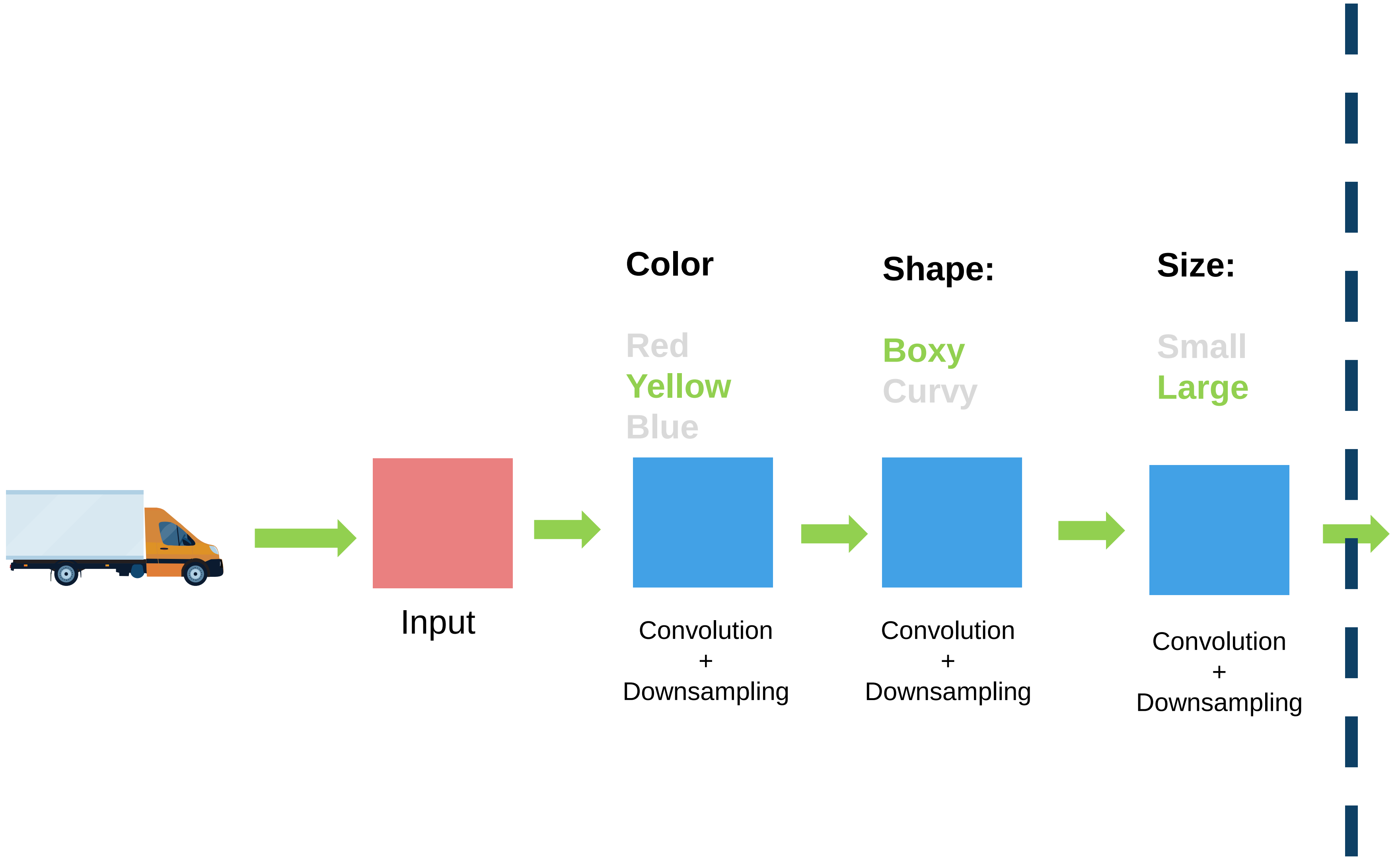
Transfer Learning



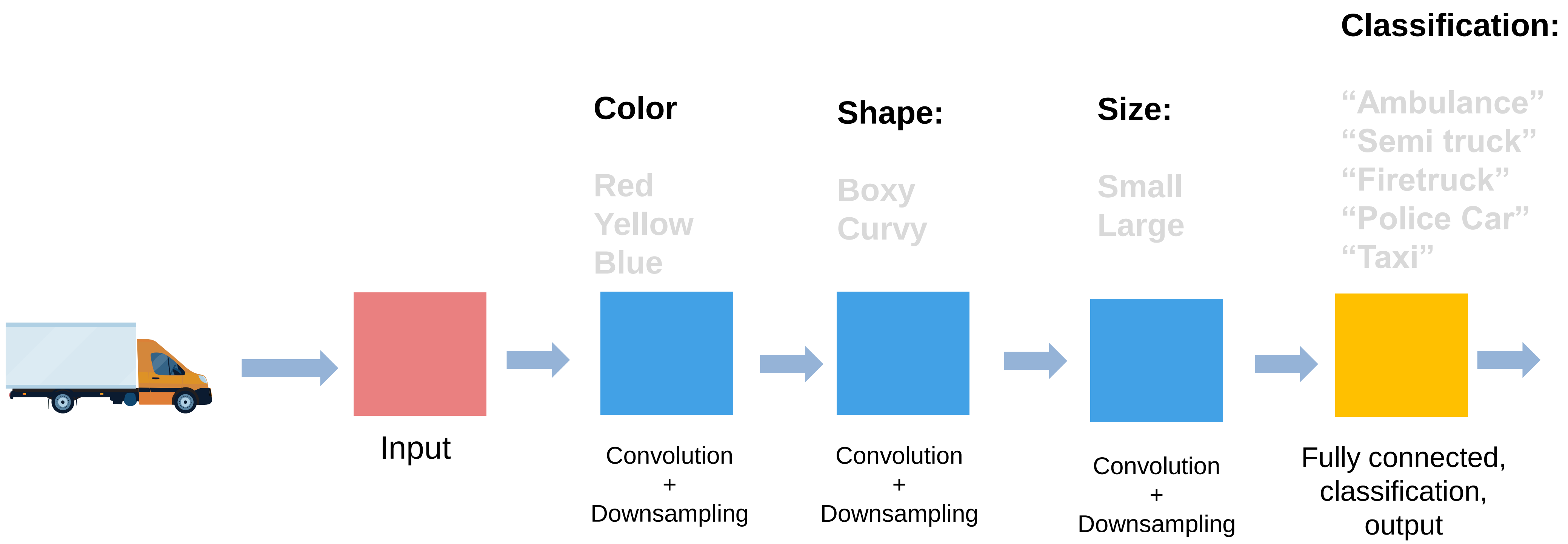
Transfer Learning



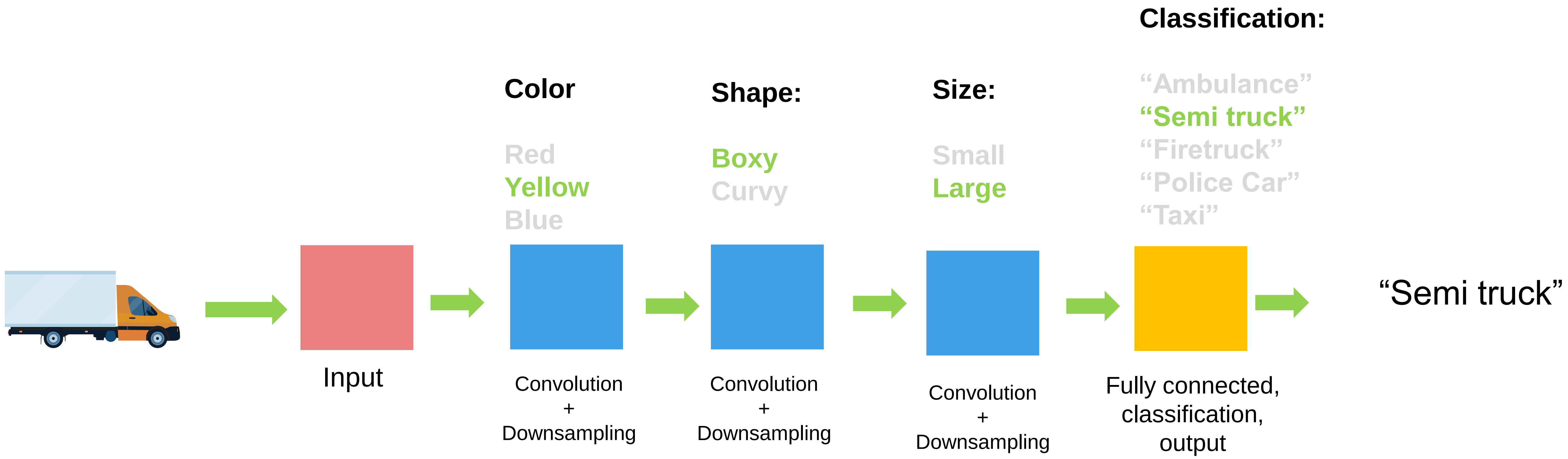
Transfer Learning



Transfer Learning



Transfer Learning



Exercise 4 – Transfer Learning

Purpose:

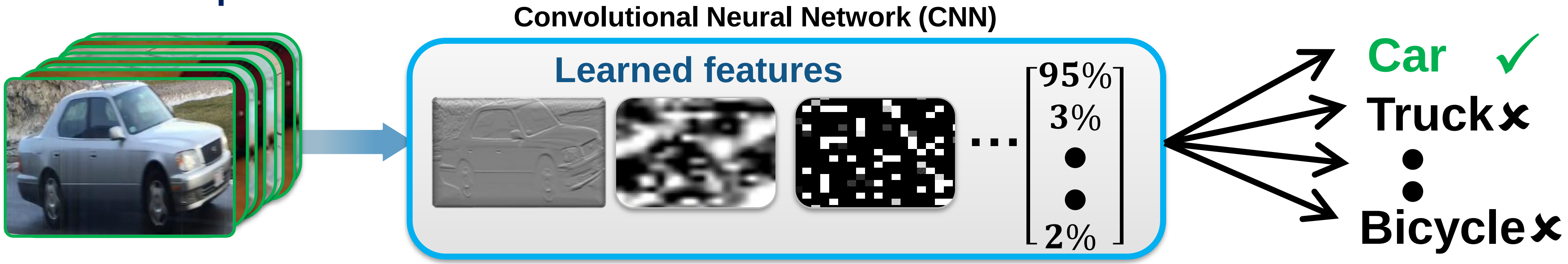
- Use transfer learning to leverage a pretrained model to classify 5 types of food
- Visualize activations within a network

To Do:

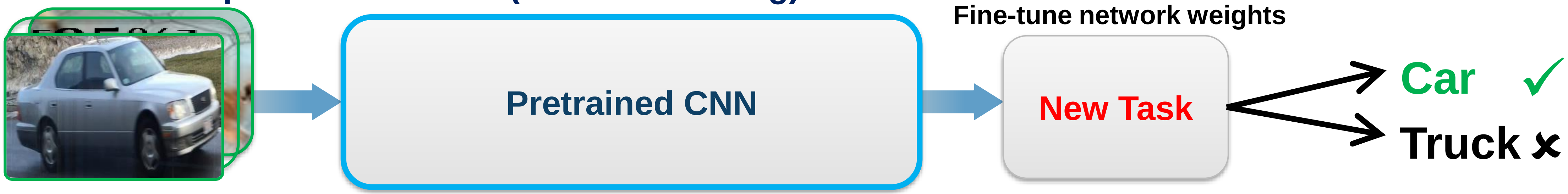
1. Open work_pretrainednetworks.mlx.

Techniques Covered so Far

1. Train a Deep Neural Network from Scratch



2. Fine-tune a pretrained model (transfer learning)



Deep Learning and Machine Learning Combined

3. Extract features with a pretrained CNN model

Pretrained CNN

Features

Machine Learning Model

SVM,
Decision
Tree etc.

Car ✓

Truck ✗

Click HERE to learn more about Machine Learning with MATLAB

Deep Learning Workflow – Prepare Data

PREPARE DATA

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

TRAIN MODEL

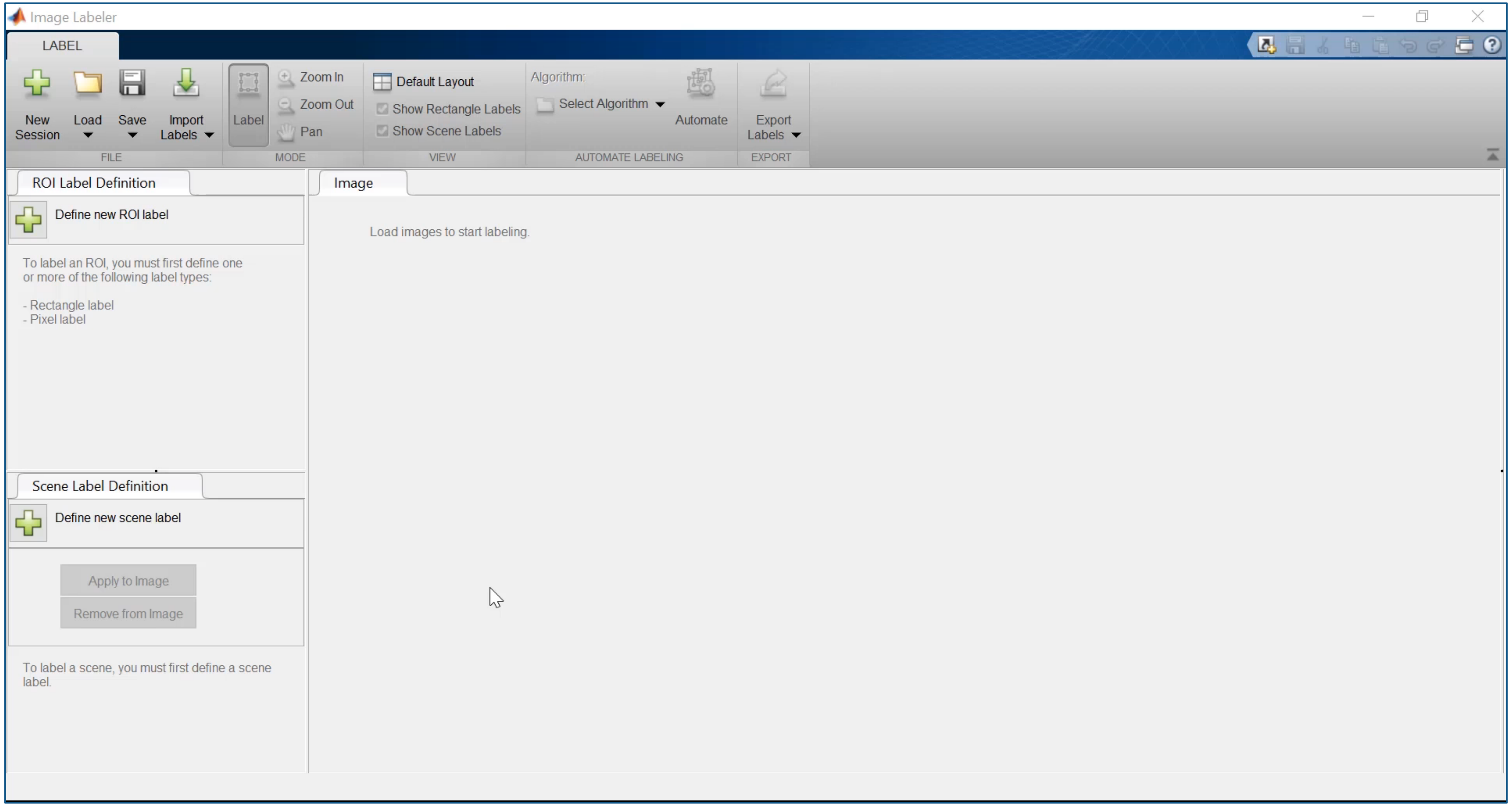
- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

DEPLOY SYSTEM

How do I label my data?

Image Labeler
+ Video labeler

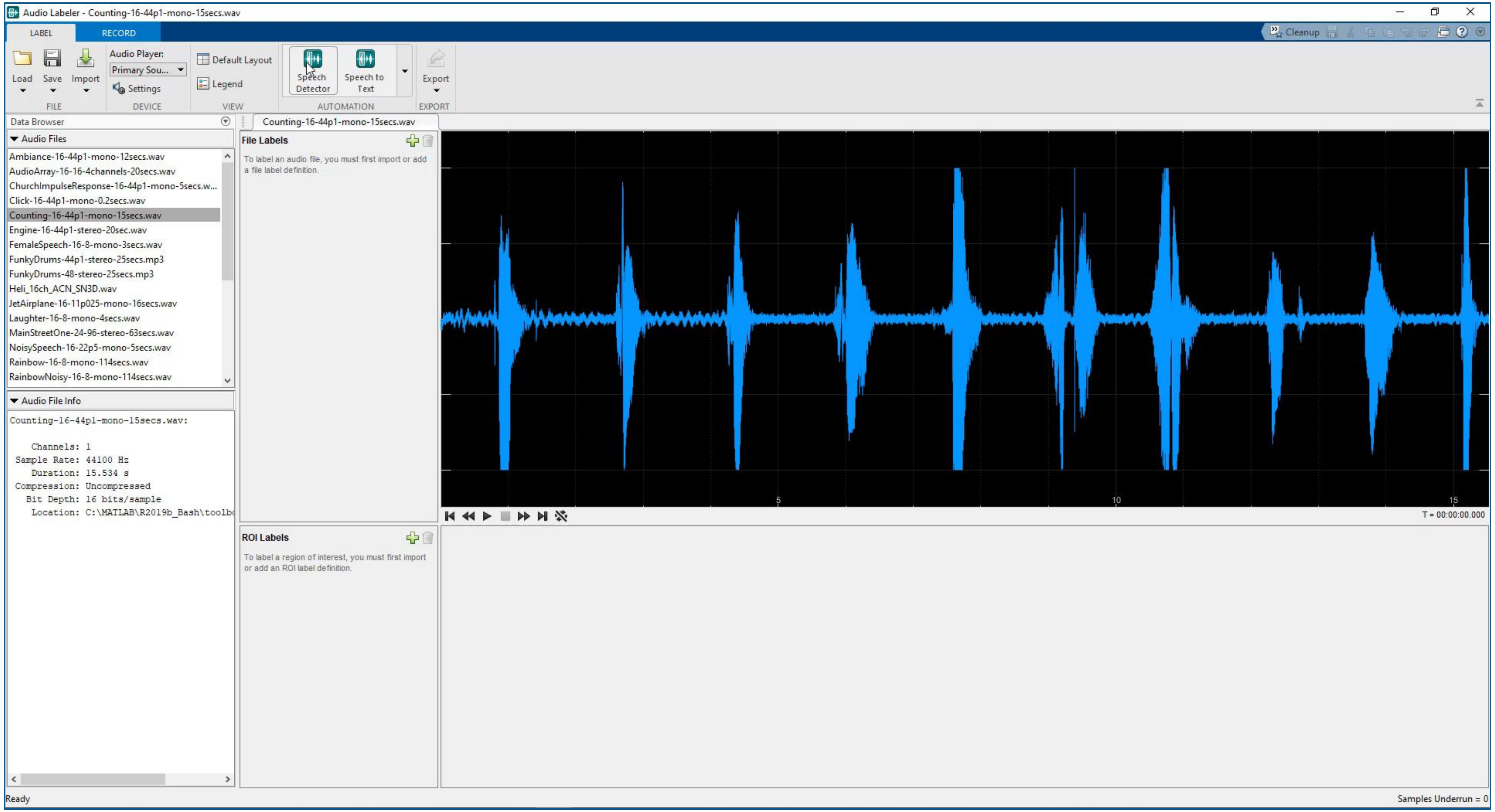
Signal Labeler
+ Audio Labeler



How do I label my data?

Image Labeler
+ Video labeler

Signal Labeler
+ Audio Labeler



Deep Learning Workflow – Deploy System

PREPARE DATA

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

TRAIN MODEL

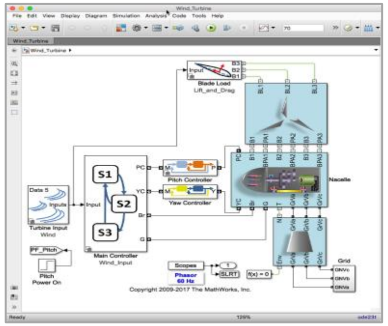
- Model design and tuning
- Hardware-accelerated training
- Model exchange across frameworks

DEPLOY SYSTEM

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

Deployment and Scaling for A.I.

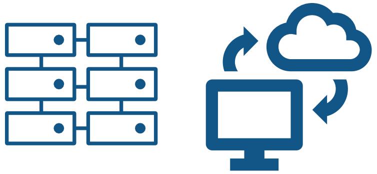
MATLAB



Embedded Systems

CPU GPU FPGA

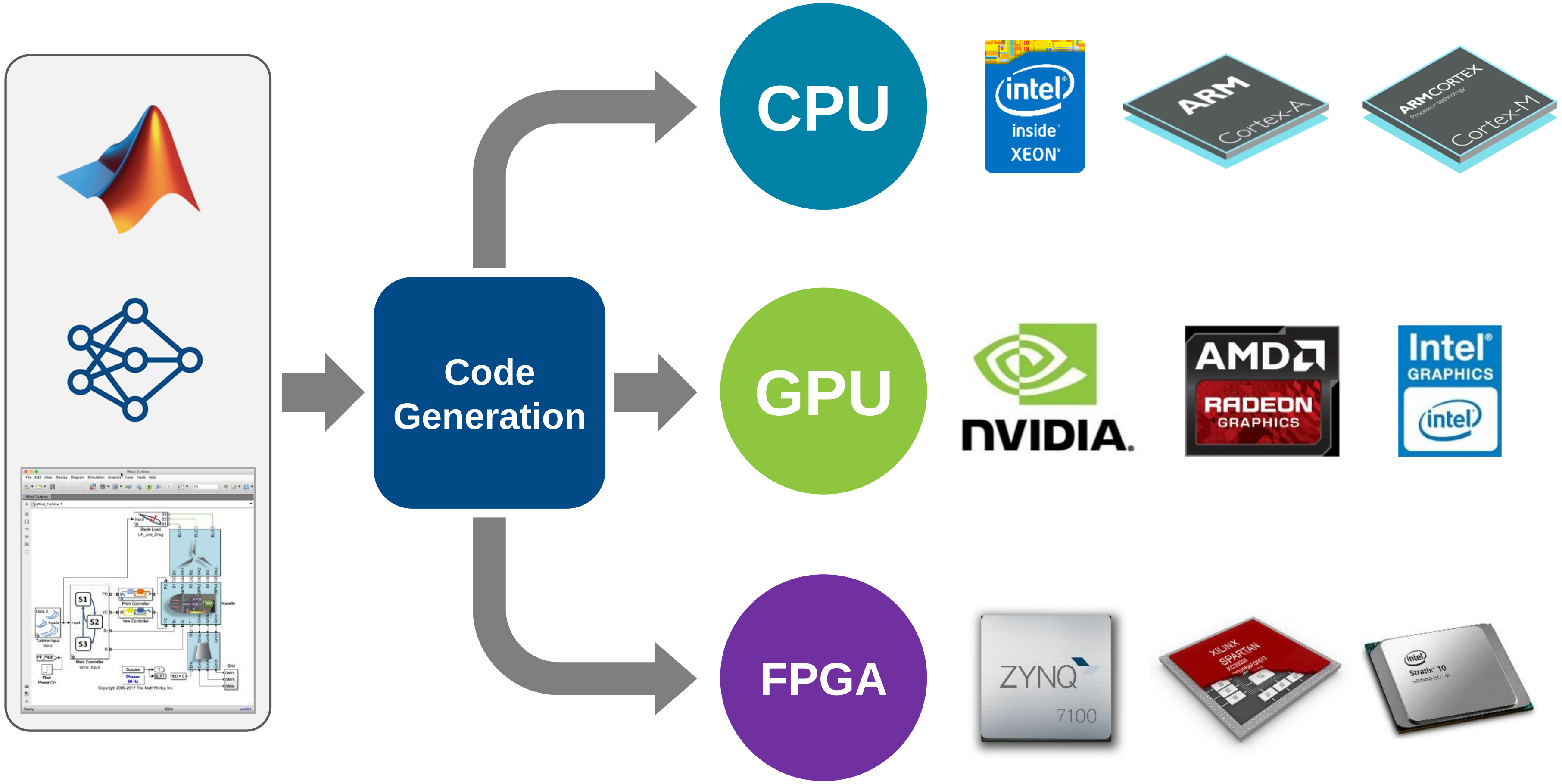
Enterprise Systems



Embedded Deployment – Automatic Code Generation

MATLAB Code → **Auto-generated Code (C/C++/CUDA)** → **Deployment Target**

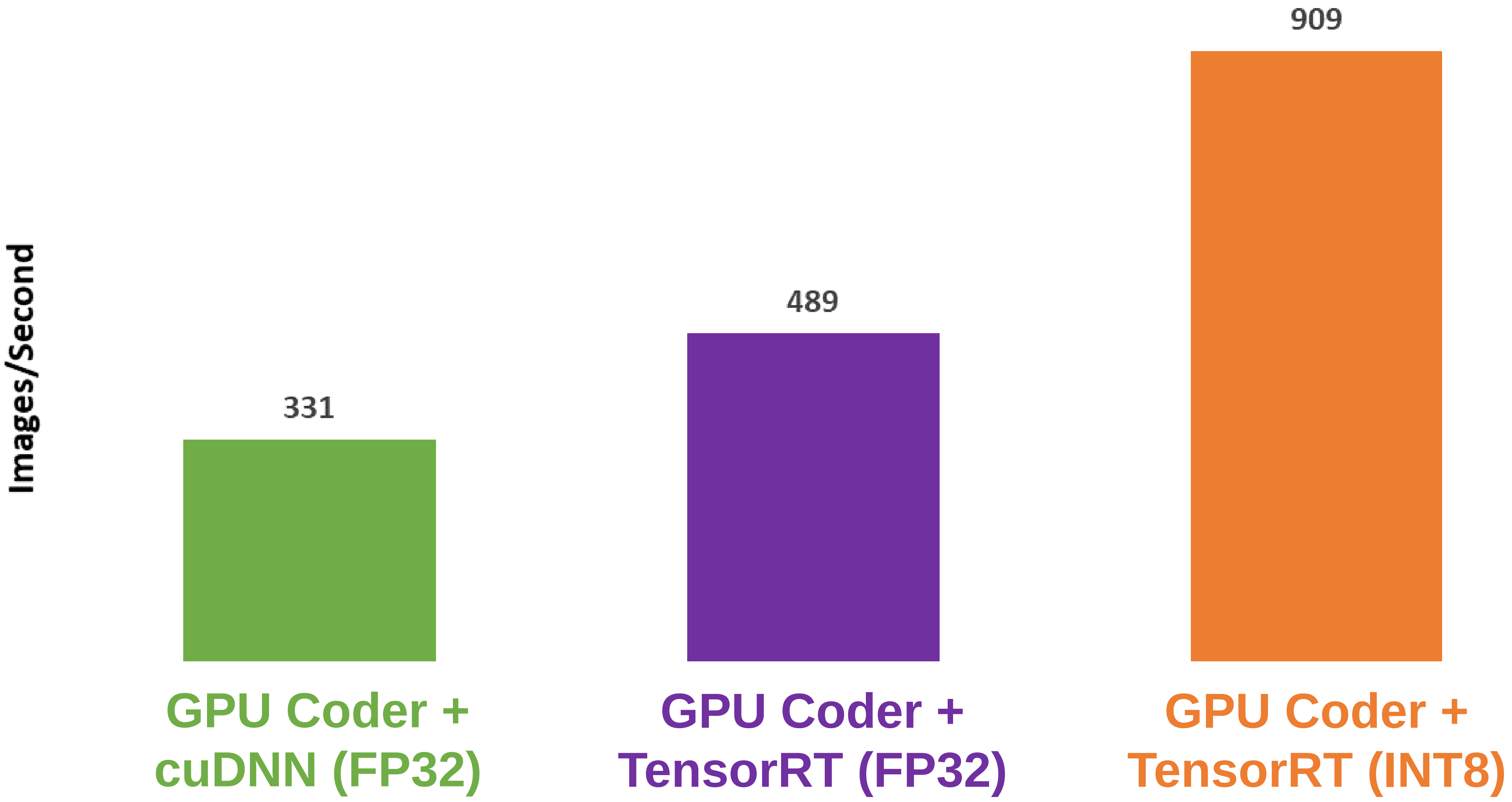
Deploying Models for Inference



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

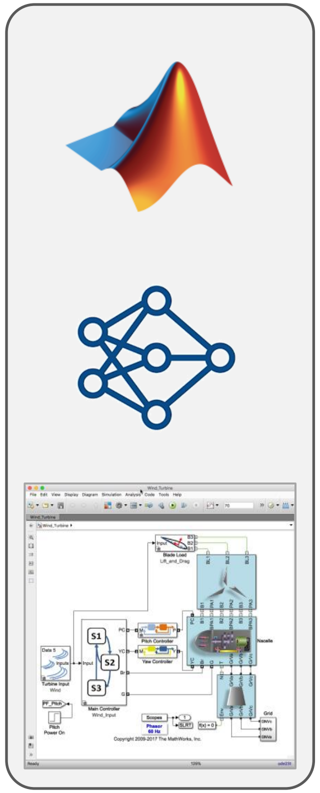
GPU Coder Inference Performance with ResNet-50 on Titan V

Batch 1



Intel® Xeon® CPU 3.6 GHz - Titan V - NVIDIA libraries: CUDA10.0/1 - cuDNN 7.5.0

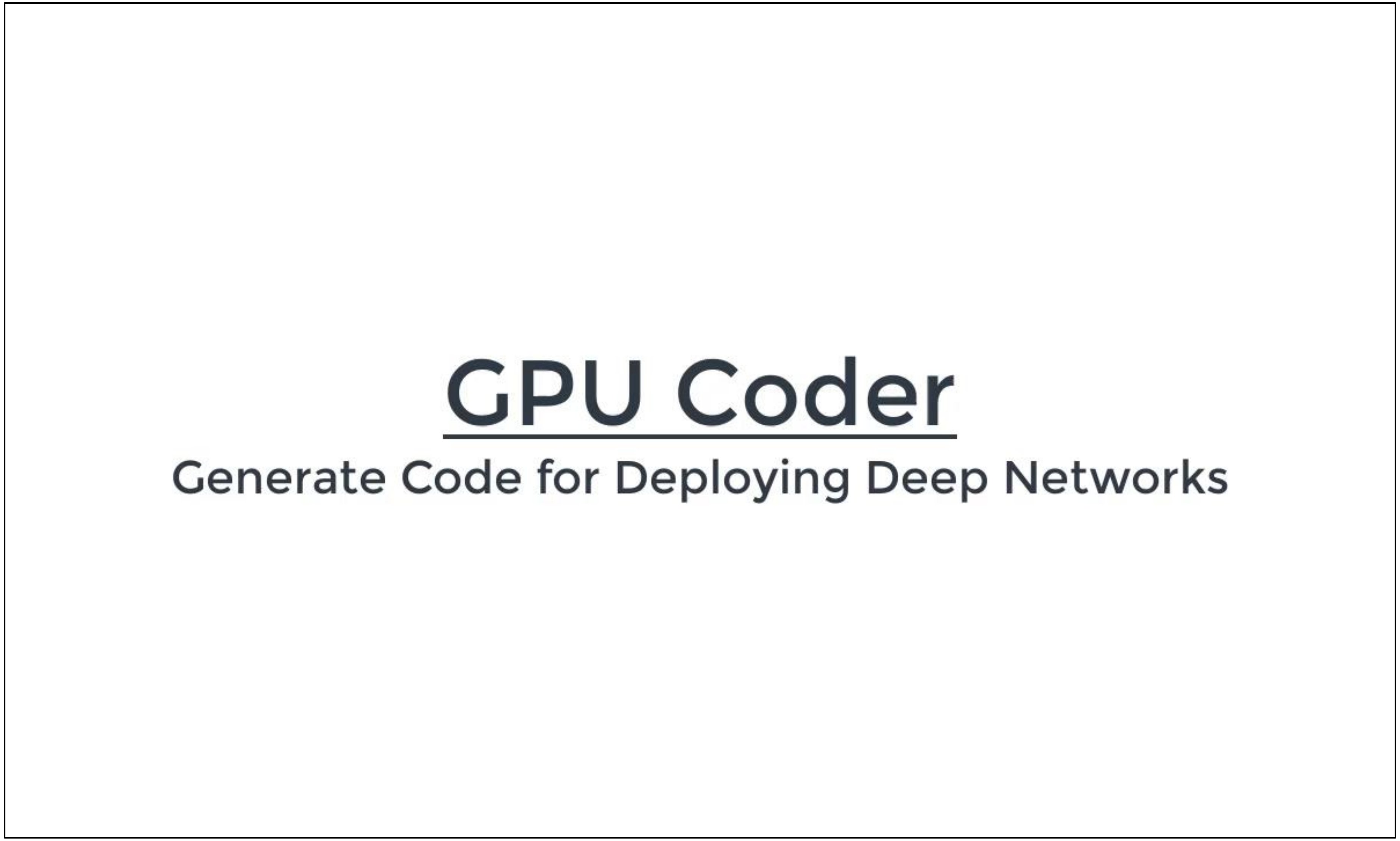
Deploy to Enterprise IT Infrastructure



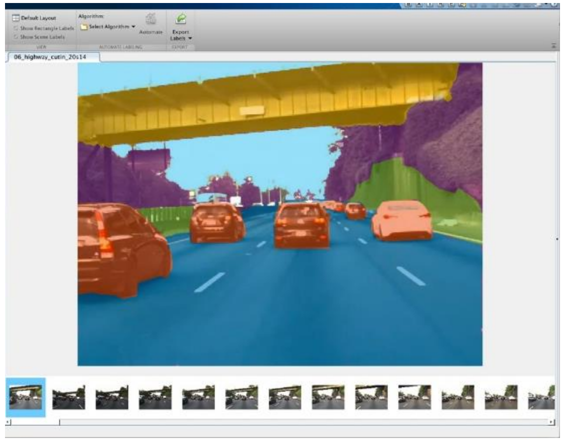
MATLAB Production Server

- Databases
- Streaming
- Cloud Storage
- Dashboards
- Containers
- Custom Tools
- Cloud & Datacenter Infrastructure

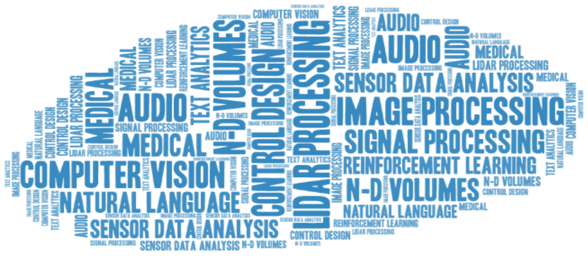
Generate GPU Code for Deep Networks



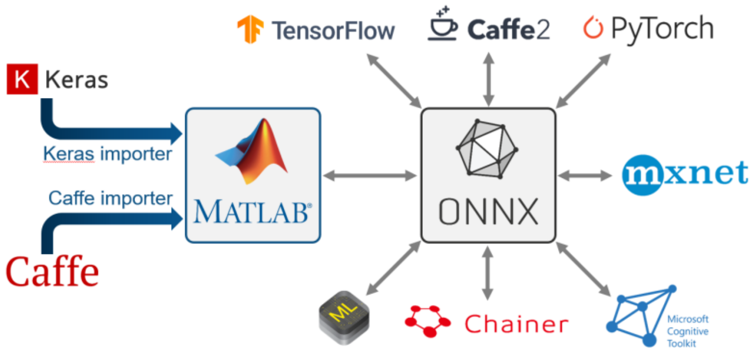
Why Use MATLAB?



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



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



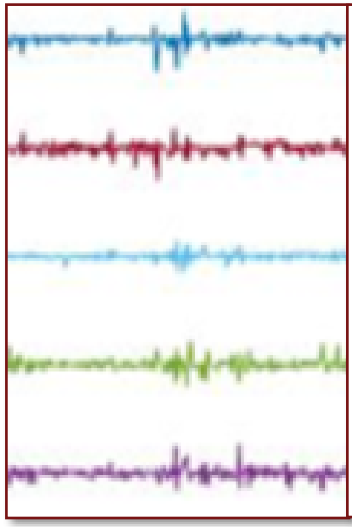
MATLAB **interoperates and enhances** Open Source frameworks

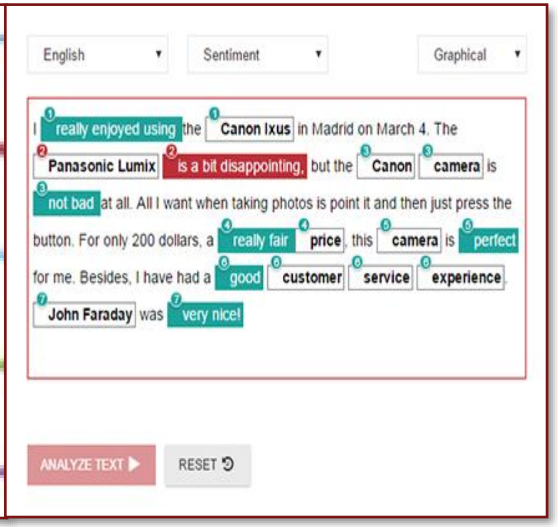
Selecting a Network Architecture

Image Data

CNN

Signal or Text Data



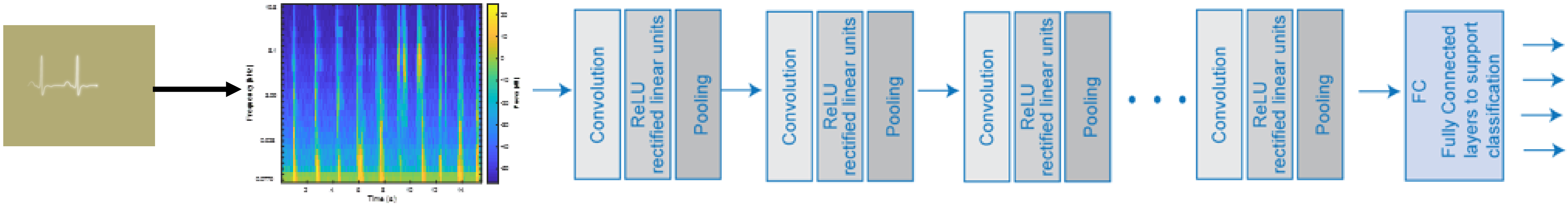


LSTM or CNN

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

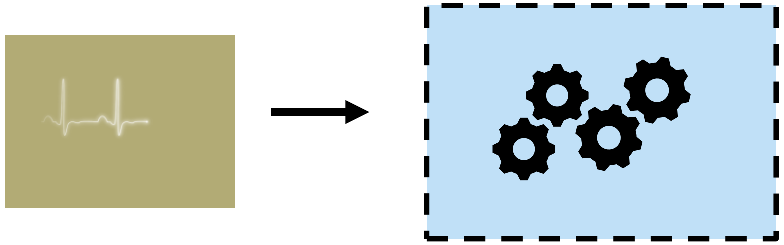
Signal Processing Architectures

Convolutional Neural Networks (**CNN**)

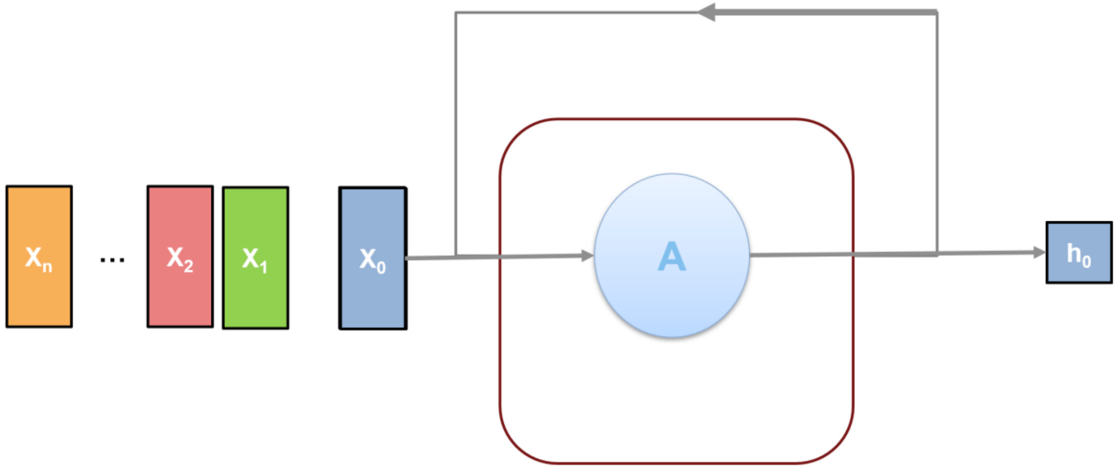


Time-Frequency Transformation

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



Feature Engineering

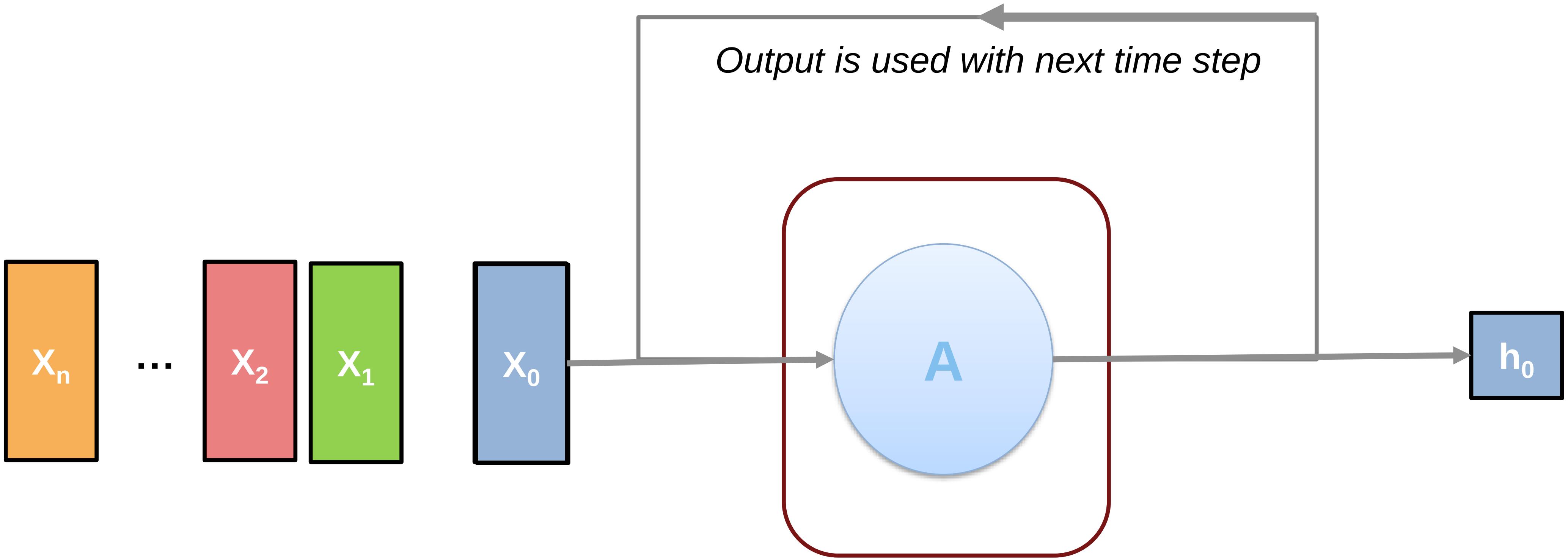


I was born in France…

… I speak ____________ ?

Recurrent Neural Networks

Take into account previous data when making new predictions



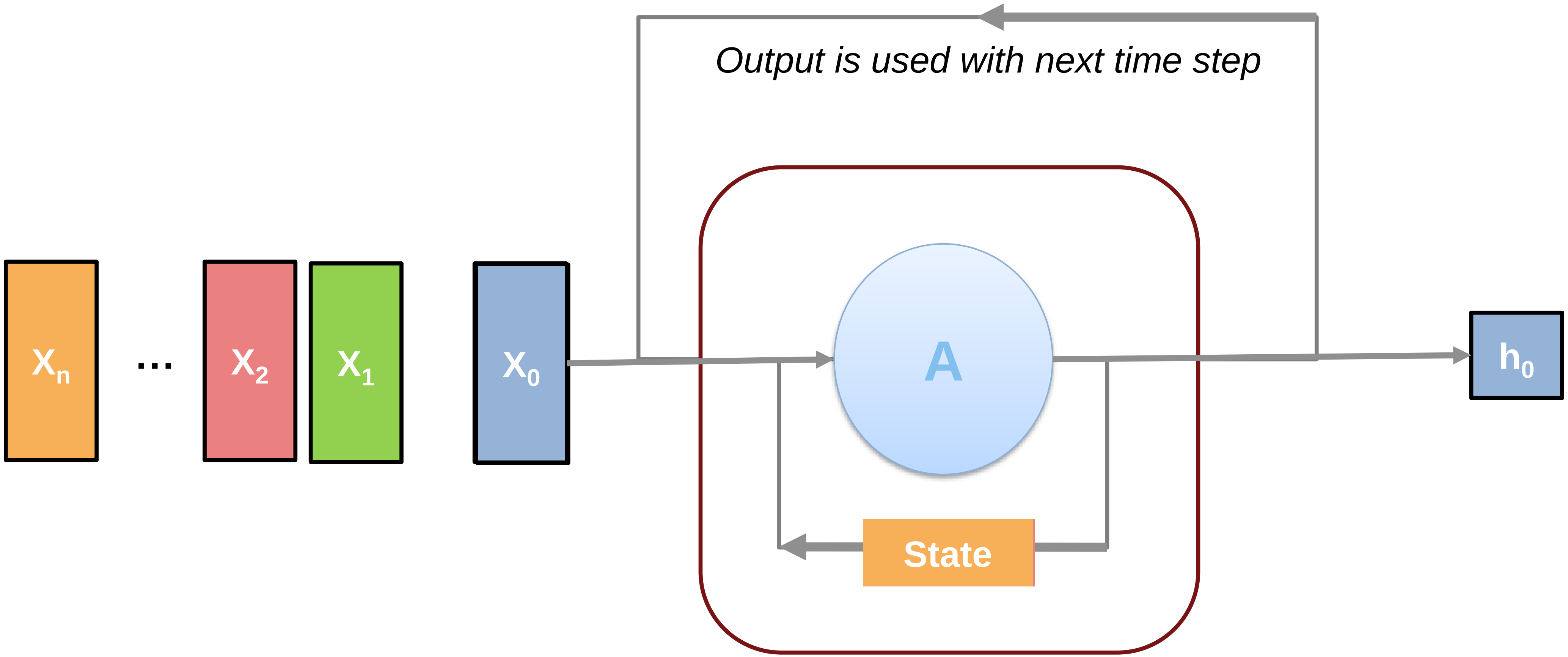
I was born in France…

[2000 words]

… I speak ____________ ?

Long Short-Term Memory Network

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

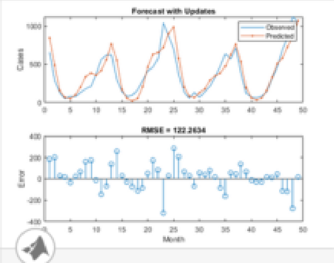


Examples in MATLAB Documentation

Sequence Classification Using Deep Learning

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

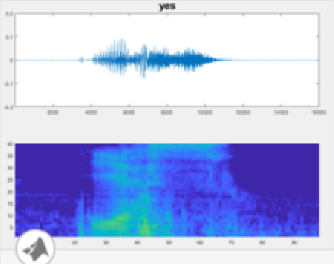
Open Live Script



Time Series Forecasting Using Deep Learning

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

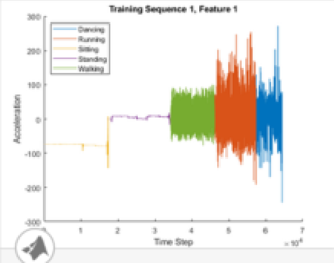
Open Live Script



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

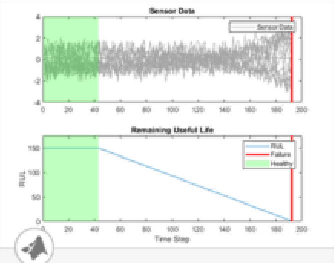
Open Script



Sequence-to-Sequence Classification Using Deep Learning

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

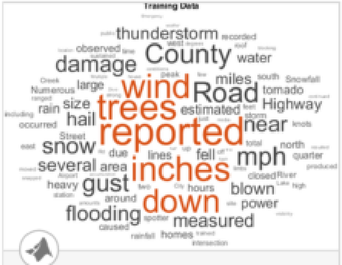
Open Live Script



Sequence-to-Sequence Regression Using Deep Learning

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

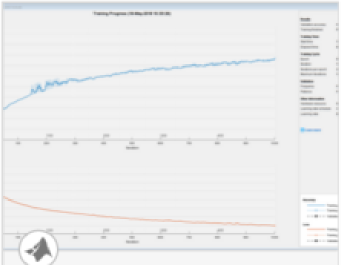
Open Live Script



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

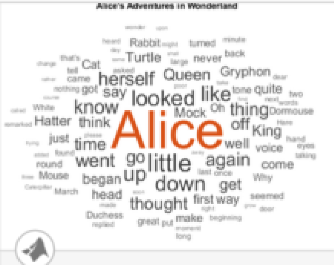
Train a deep learning long short-term memory (LSTM) network to generate text.

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

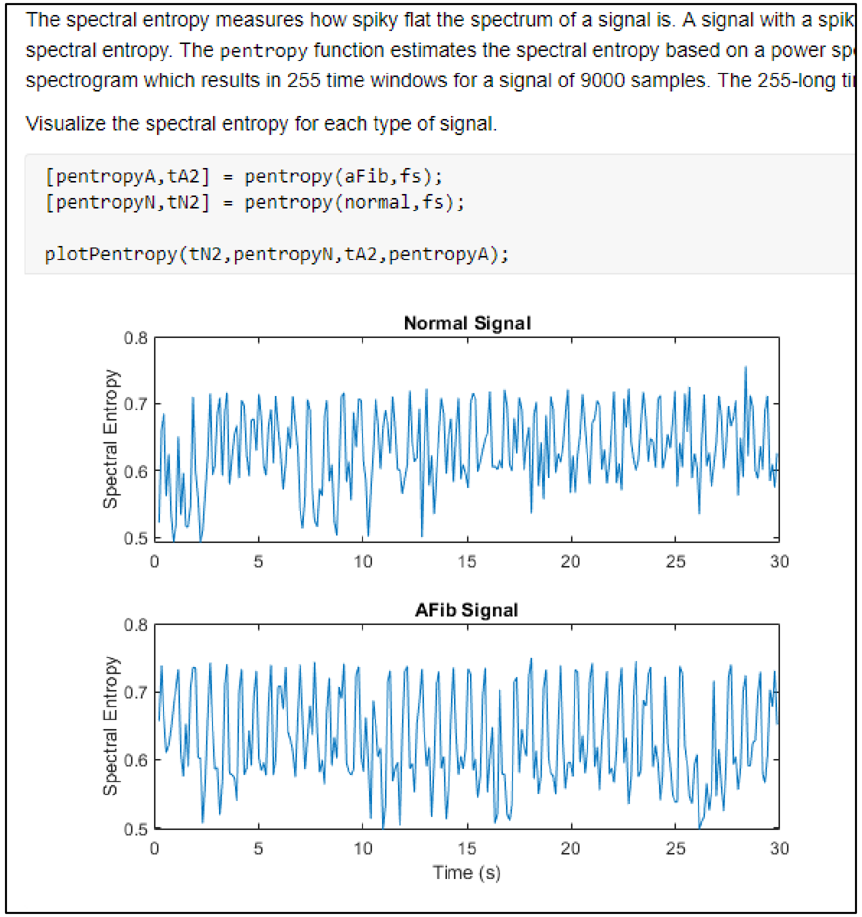
Exercise – ECG Signal Classification

Purpose:

- Use LSTM to classify ECG signal as normal heartbeat or Atrial Fibrillation
- 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);
```

MathWorks Engineering Support

Training

Consulting

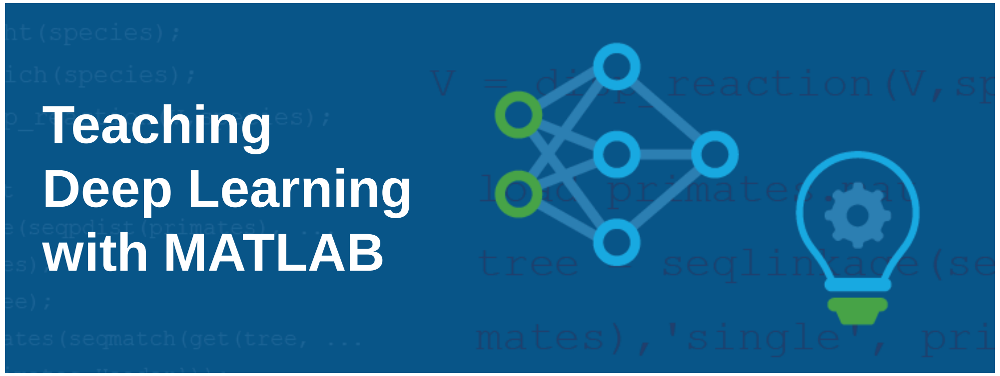
Onsite Workshops and Seminars

Guided Evaluations

Technical Support

Further Learning and Teaching

- Deep Learning Onramp
  - 2 hr online tutorial
- Deep Learning Workshop
  - 3 hr hands on session
  - Contact us to schedule
- Deep Learning Training
  - 16 hr in depth course
  - Online or Instructor Lead
- Teaching Deep Learning with MATLAB
  - Curriculum support



Thank you!