TeraSoft
www.terasoft.com.tw

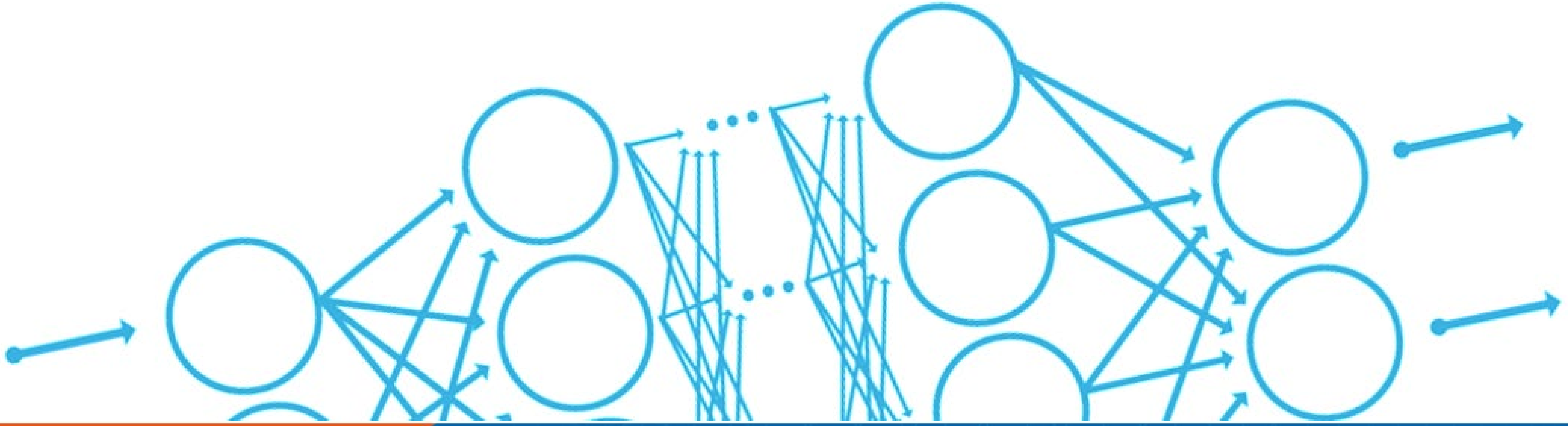
Deep Learning with MATLAB

Fred Liu

Application Engineer

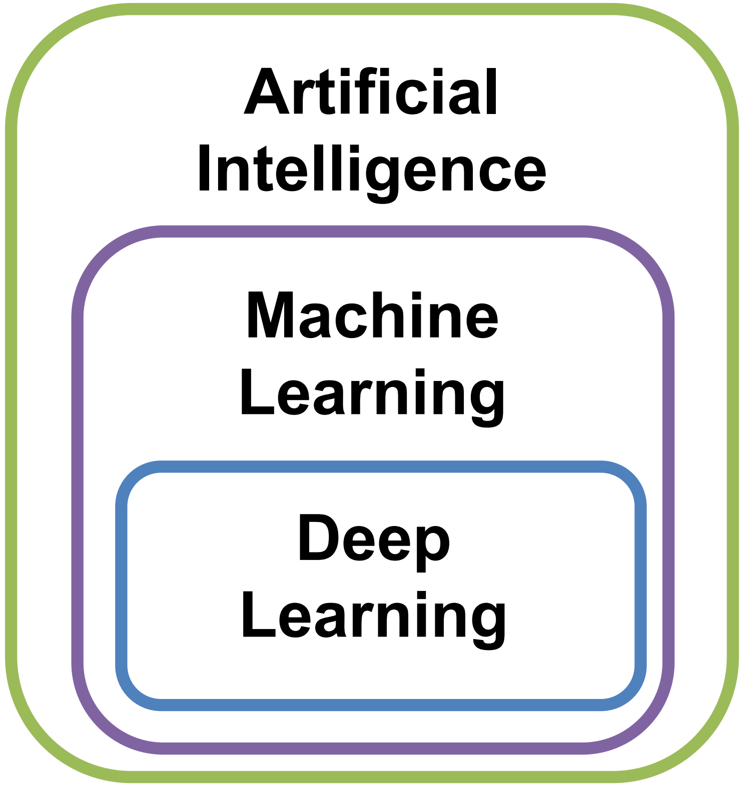
MathWorks®
Accelerating the pace of engineering and science

What is Deep Learning?



Deep Learning

- Subset of machine learning that performs **automatic feature extraction**
  - Learns features and tasks directly from data

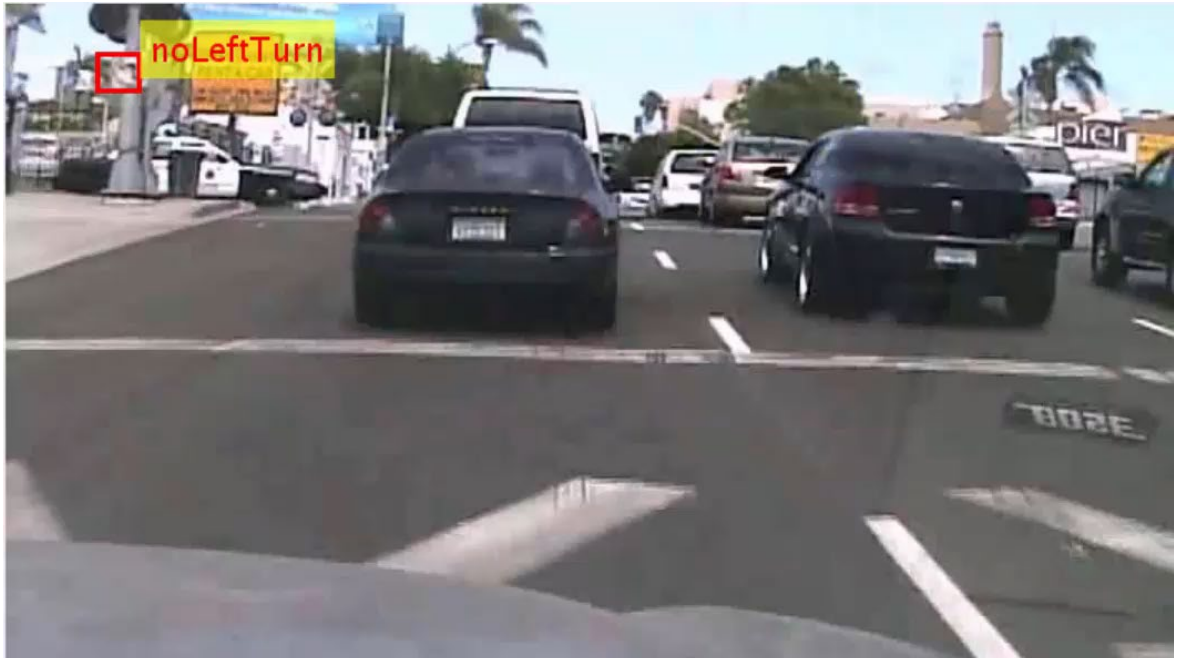


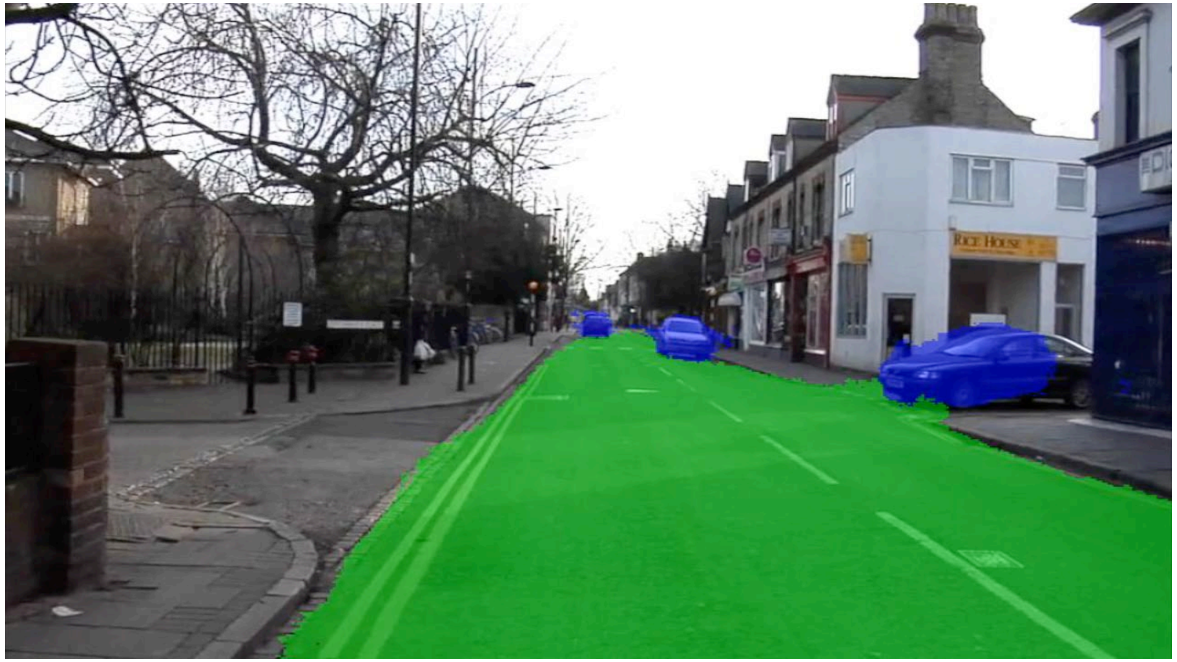
Artificial Intelligence

Machine Learning

Deep Learning

Example 1: Detection and localization using deep learning

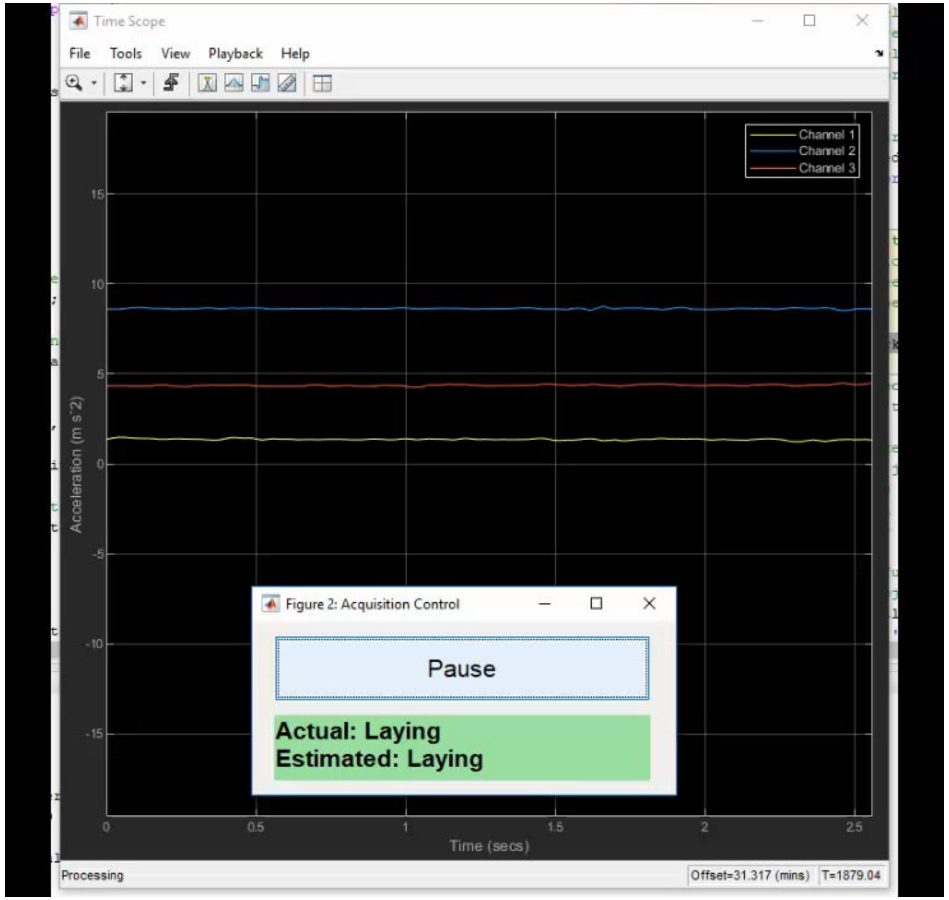




YOLO v2 (You Only Look Once)

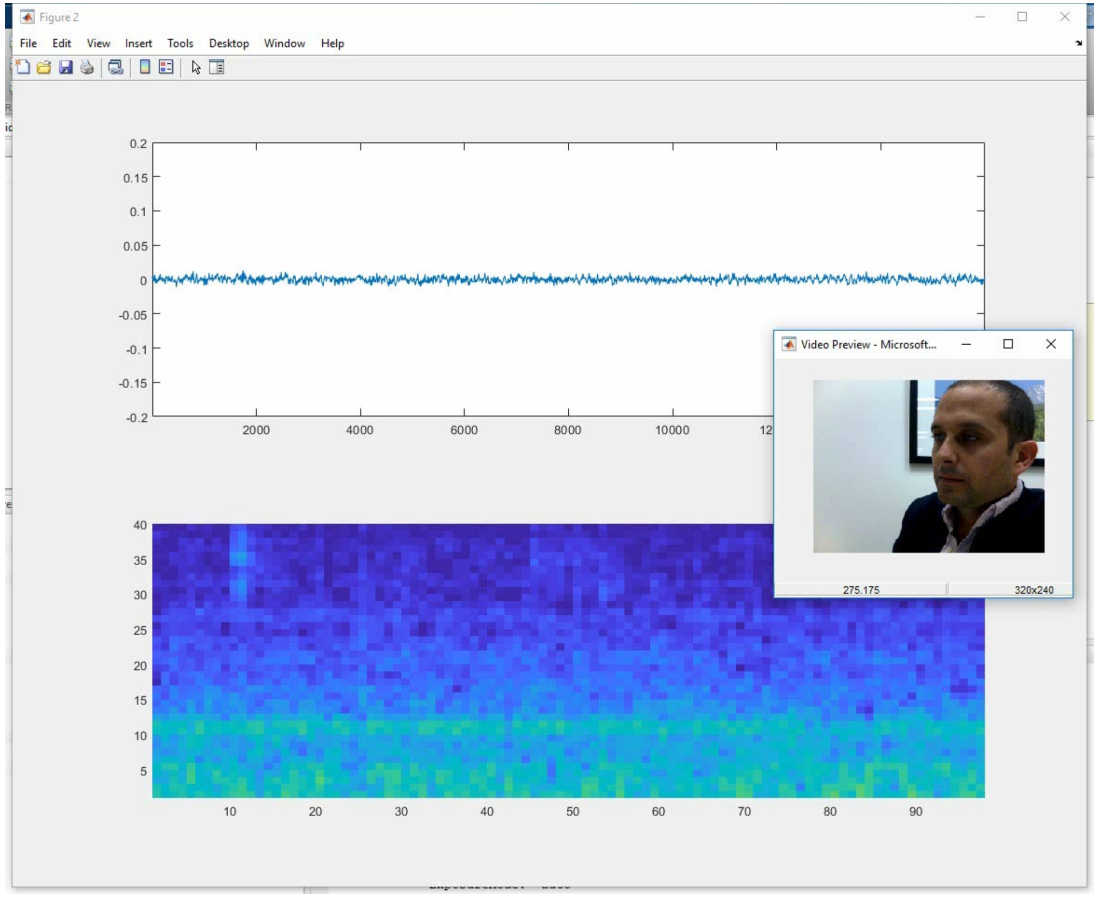
Semantic Segmentation using SegNet

Example 2: Analyzing signal data using deep learning



Actual: Laying
Estimated: Laying

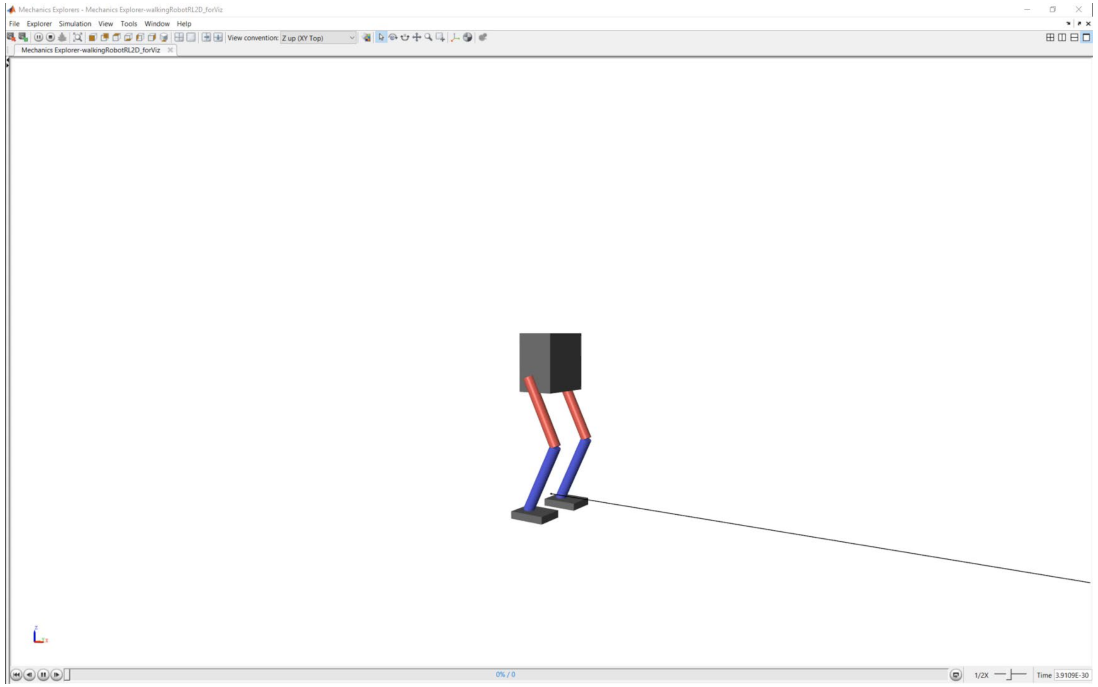
Processing | Offset=31.317 (mins) | T=1879.04



Signal Classification using LSTMs

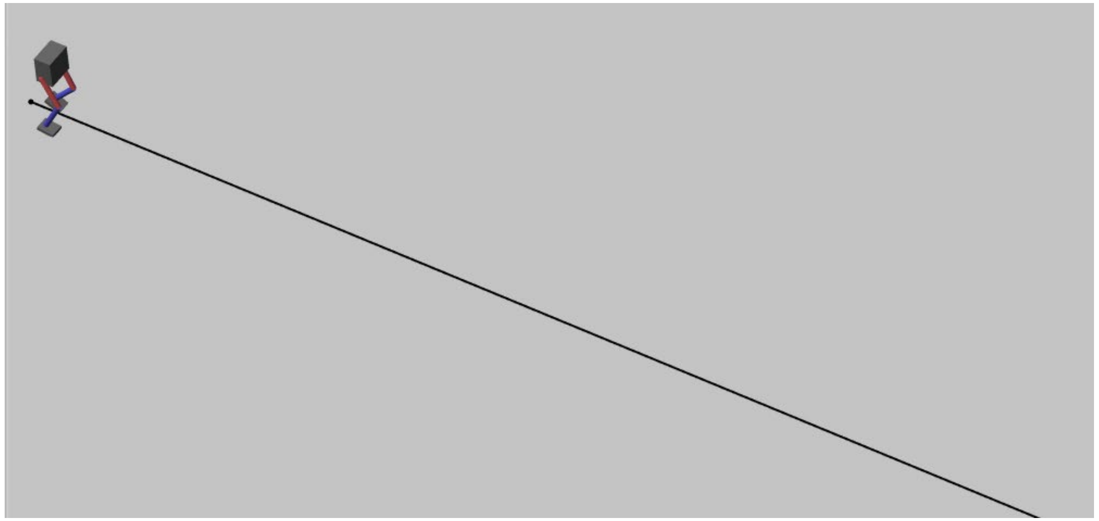
Speech Recognition using CNNs

Example 3: Robot walking with deep reinforcement learning


Mechanics Explorers - Mechanics Explorer-walkingRobotRL2D_forViz
File
Explorer
Simulation
View
Tools
Window
Help
Mechanics Explorer-walkingRobotRL2D_forViz
0% / 0
1/2X

View convention: Z up (XY Top)

Time 3.9109E-30



Simulation During Training

Trained Agent (DQN, DDPG, etc.)

Agenda

- Machine Learning Overview

- How to use Pretrained model

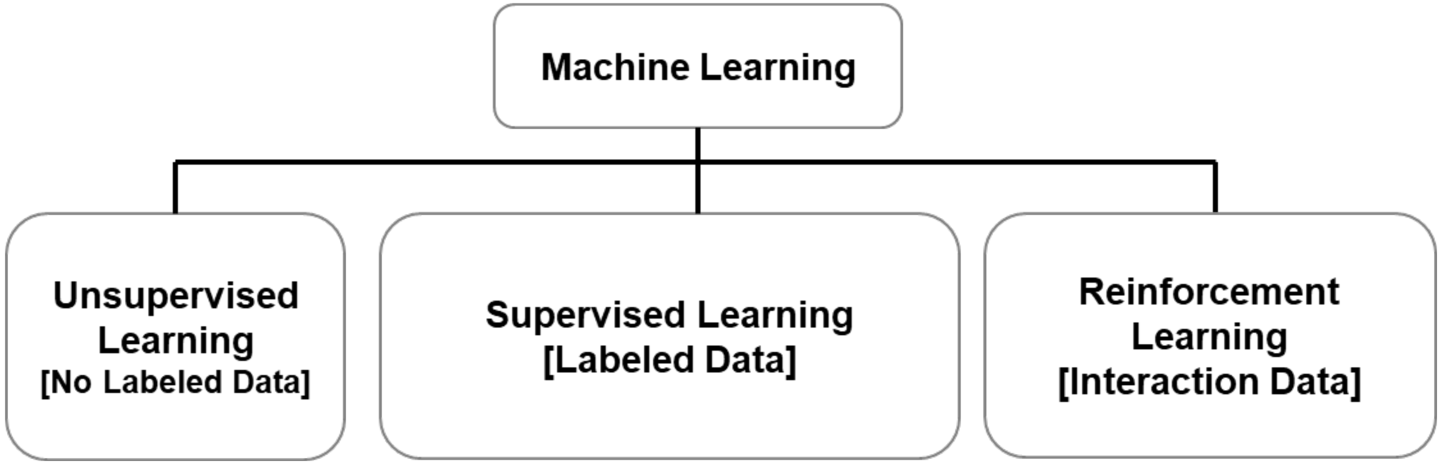
- Create Deep Learning Model(MNIST)

- Try to do Transfer Learning

Agenda

- Machine Learning Overview
- How to use Pretrained model
- Create Deep Learning Model(MNIST)
- Try to do Transfer Learning

Reinforcement Learning vs Machine Learning vs Deep Learning



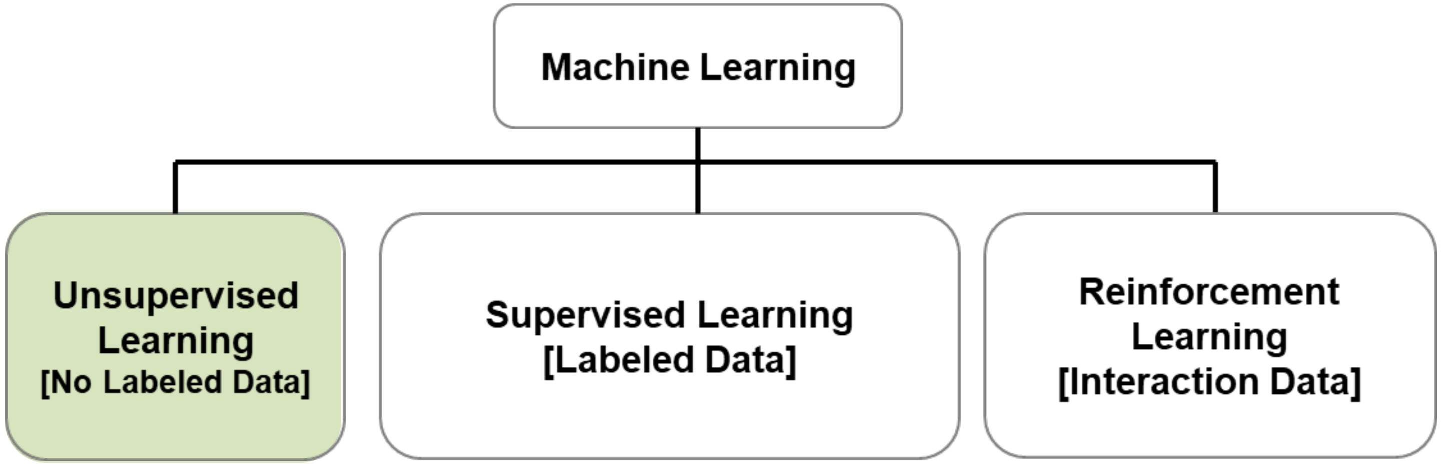
Machine Learning

Unsupervised Learning
[No Labeled Data]

Supervised Learning
[Labeled Data]

Reinforcement Learning
[Interaction Data]

Reinforcement Learning vs Machine Learning vs Deep Learning

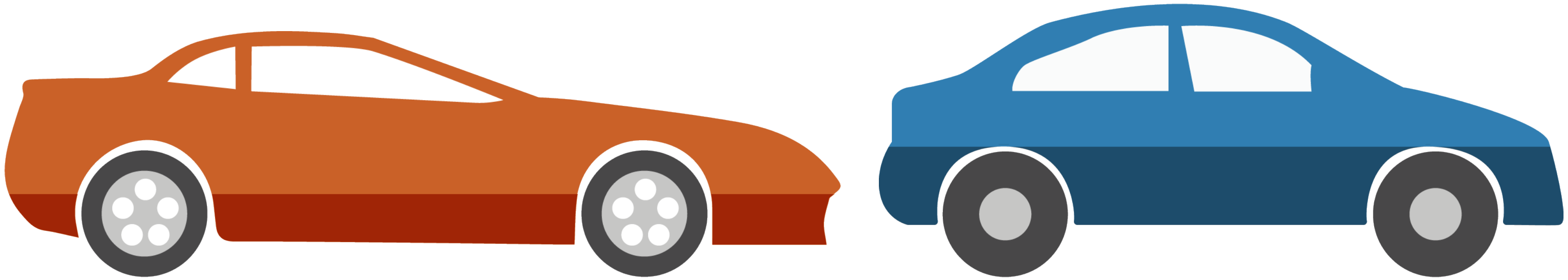


Machine Learning

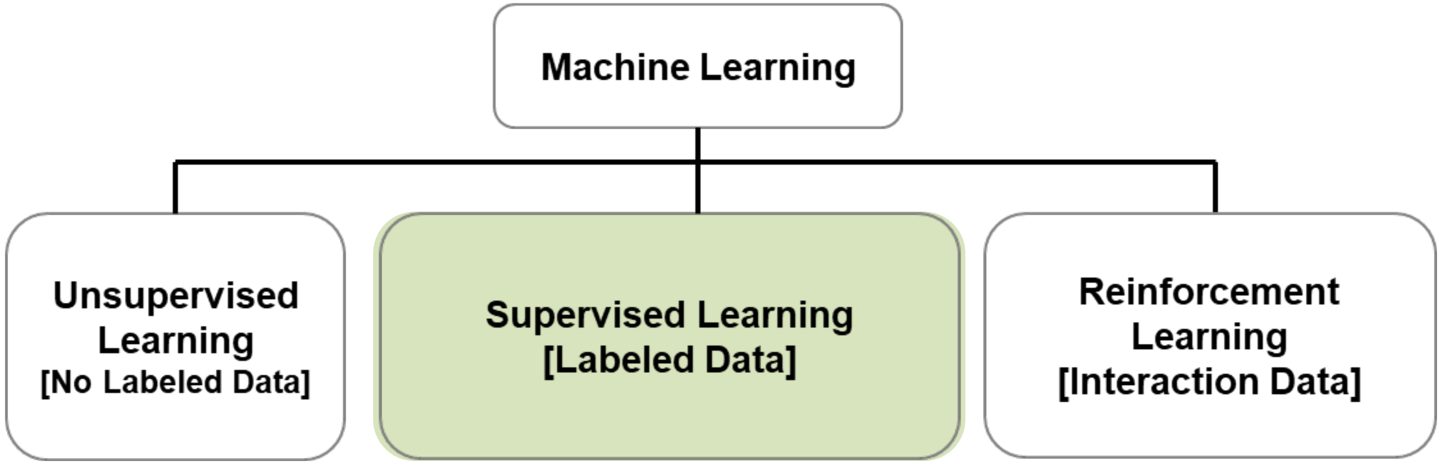
Unsupervised Learning
[No Labeled Data]

Supervised Learning
[Labeled Data]

Reinforcement Learning
[Interaction Data]



Reinforcement Learning vs Machine Learning vs Deep Learning

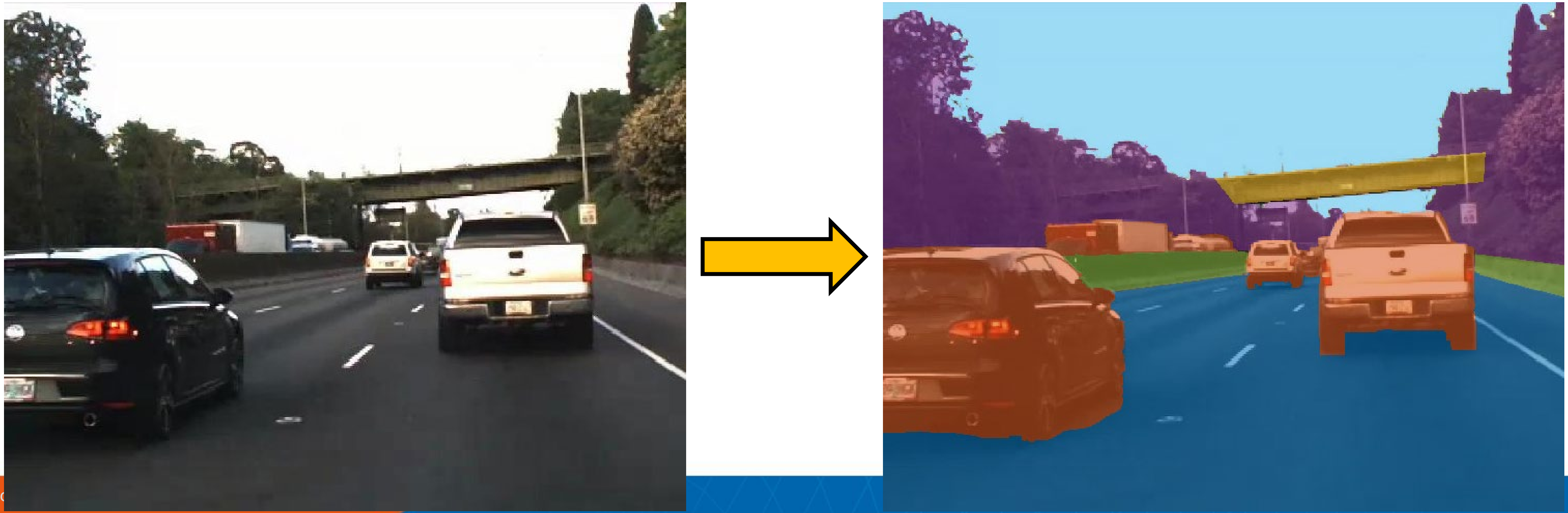


Machine Learning

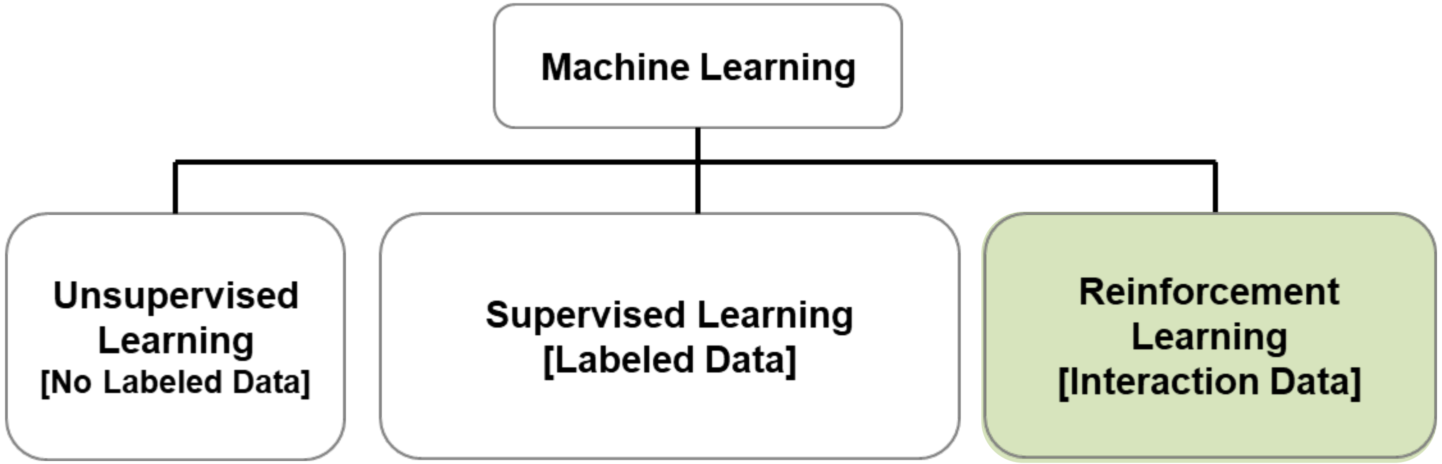
Unsupervised Learning
[No Labeled Data]

Supervised Learning
[Labeled Data]

Reinforcement Learning
[Interaction Data]



Reinforcement Learning vs Machine Learning vs Deep Learning



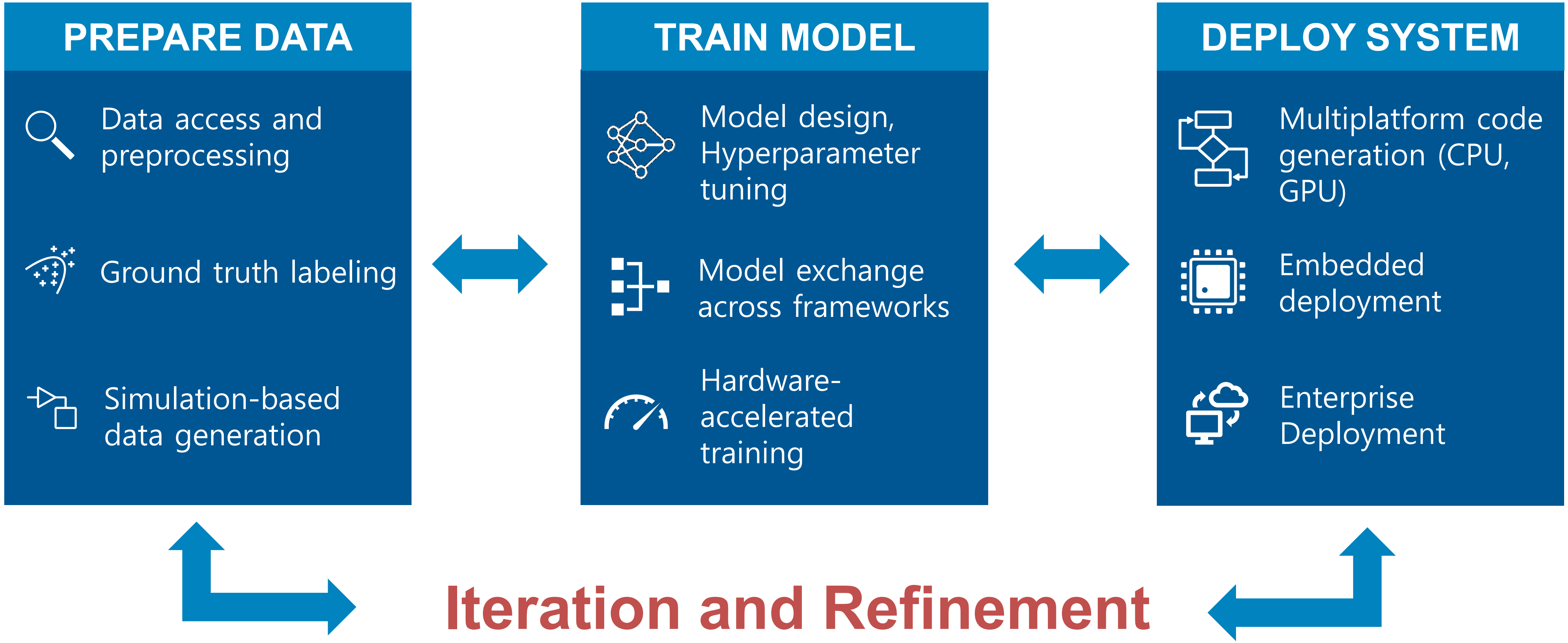
Machine Learning

Unsupervised Learning
[No Labeled Data]

Supervised Learning
[Labeled Data]

Reinforcement Learning
[Interaction Data]

Deep Learning and Reinforcement Learning Workflow



PREPARE DATA

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

TRAIN MODEL

- Model design, Hyperparameter tuning
- Model exchange across frameworks
- Hardware-accelerated training

DEPLOY SYSTEM

- Multiplatform code generation (CPU, GPU)
- Embedded deployment
- Enterprise Deployment

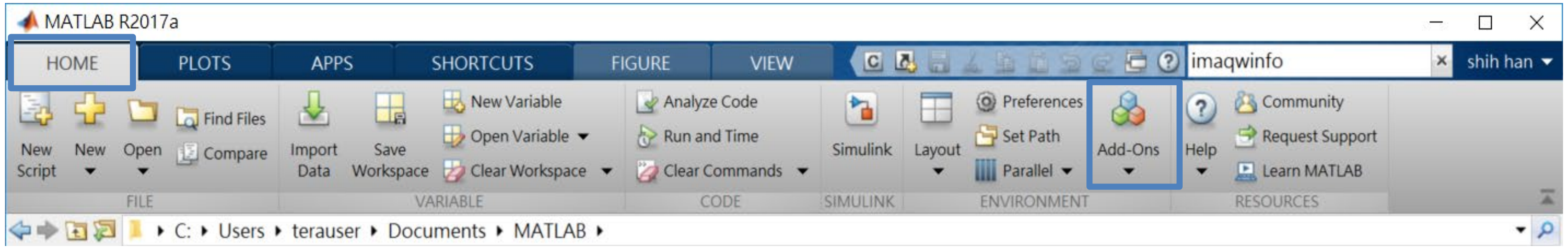
Iteration and Refinement

Agenda

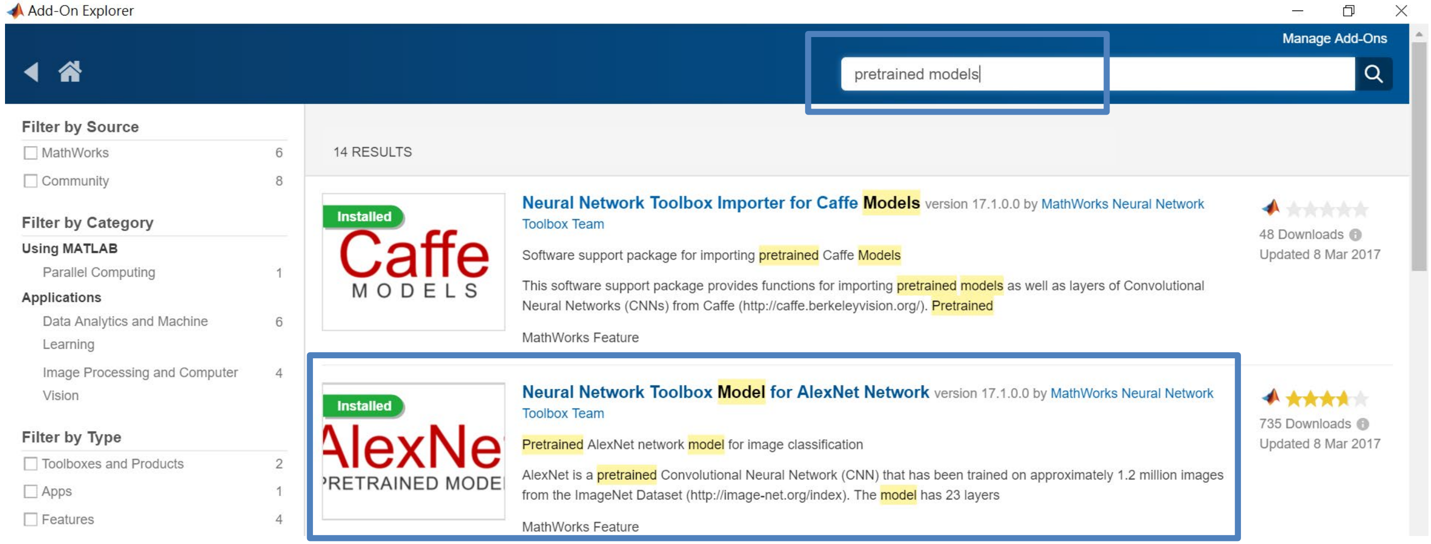
- Machine Learning Overview
- How to use Pretrained model
- Create Deep Learning Model(MNIST)
- Try to do Transfer Learning

Download and Install Pre-trained Networks

1. Go to the **Home** tab and click the **Add-Ons** icon.



2. Search **‘pretrained models’.** Click AlexNet add-on to open its detailed information page.



14 RESULTS

Neural Network Toolbox Importer for Caffe Models version 17.1.0.0 by MathWorks Neural Network Toolbox Team

Software support package for importing pretrained Caffe Models

This software support package provides functions for importing pretrained models as well as layers of Convolutional Neural Networks (CNNs) from Caffe (http://caffe.berkeleyvision.org/). Pretrained

MathWorks Feature

48 Downloads
Updated 8 Mar 2017

Neural Network Toolbox Model for AlexNet Network version 17.1.0.0 by MathWorks Neural Network Toolbox Team

Pretrained AlexNet network model for image classification

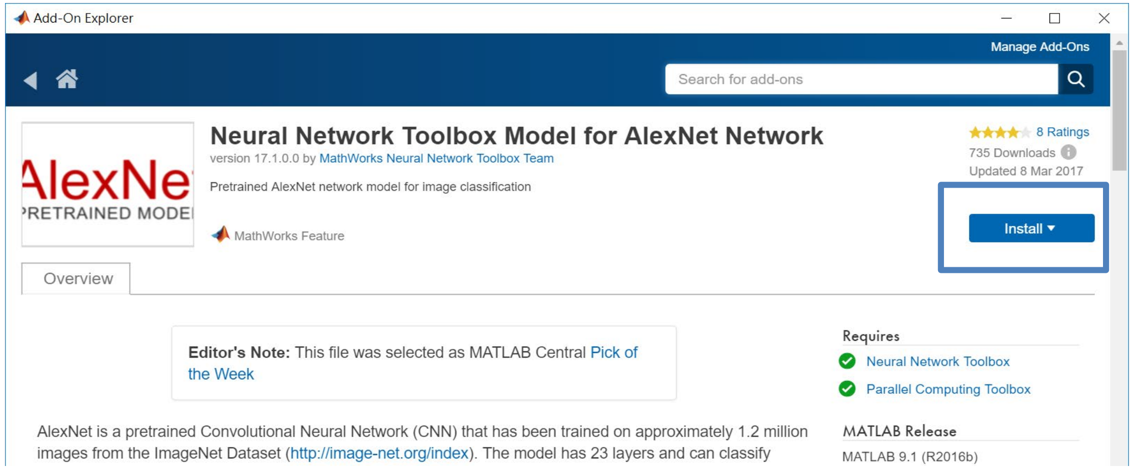
AlexNet is a pretrained Convolutional Neural Network (CNN) that has been trained on approximately 1.2 million images from the ImageNet Dataset (http://image-net.org/index). The model has 23 layers

MathWorks Feature

735 Downloads
Updated 8 Mar 2017

Download and Install Pre-trained Networks

3. Install the add-on.



Neural Network Toolbox Model for AlexNet Network

version 17.1.0.0 by MathWorks Neural Network Toolbox Team

Pretrained AlexNet network model for image classification

MathWorks Feature

★★★★☆ 8 Ratings
735 Downloads
Updated 8 Mar 2017

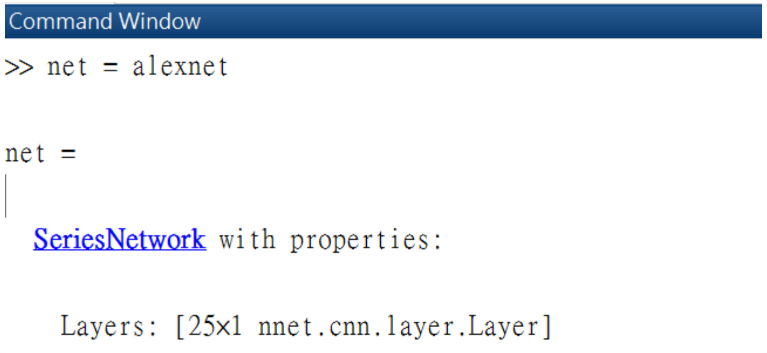
Editor's Note: This file was selected as MATLAB Central Pick of the Week

AlexNet is a pretrained Convolutional Neural Network (CNN) that has been trained on approximately 1.2 million images from the ImageNet Dataset (http://image-net.org/index). The model has 23 layers and can classify

Requires
- ✓ Neural Network Toolbox
- ✓ Parallel Computing Toolbox

MATLAB Release
MATLAB 9.1 (R2016b)

4. Import the model by typing **net = alexnet** in command window!



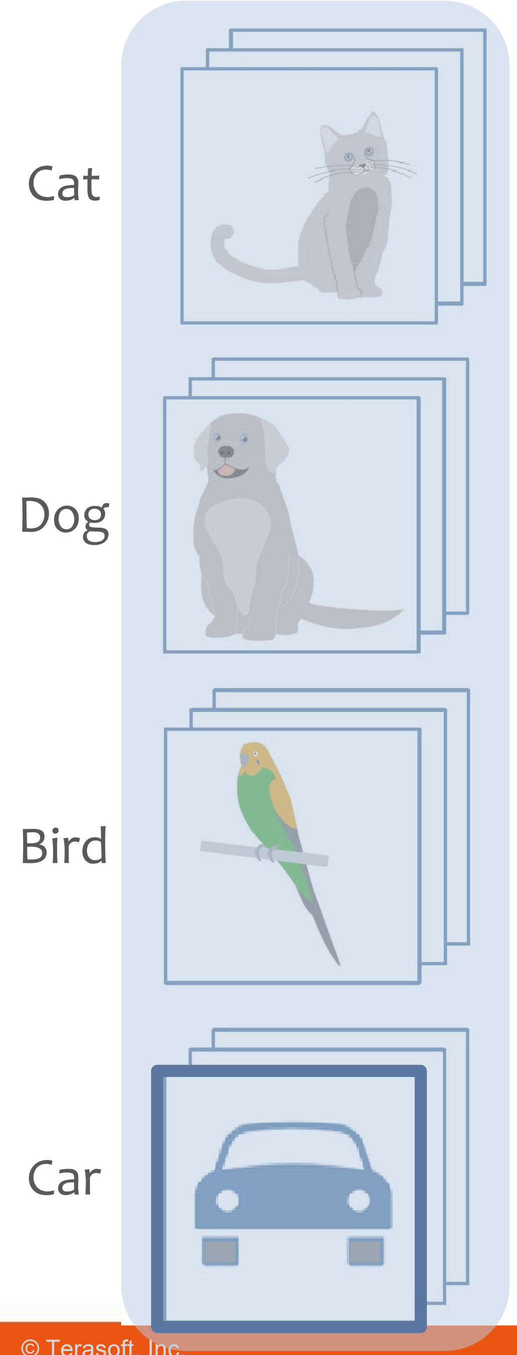
Command Window

```
>> net = alexnet

net =

  SeriesNetwork with properties:

    Layers: [25x1 nnet.cnn.layer.Layer]
```

AlexNet
PRETRAINED MODEL

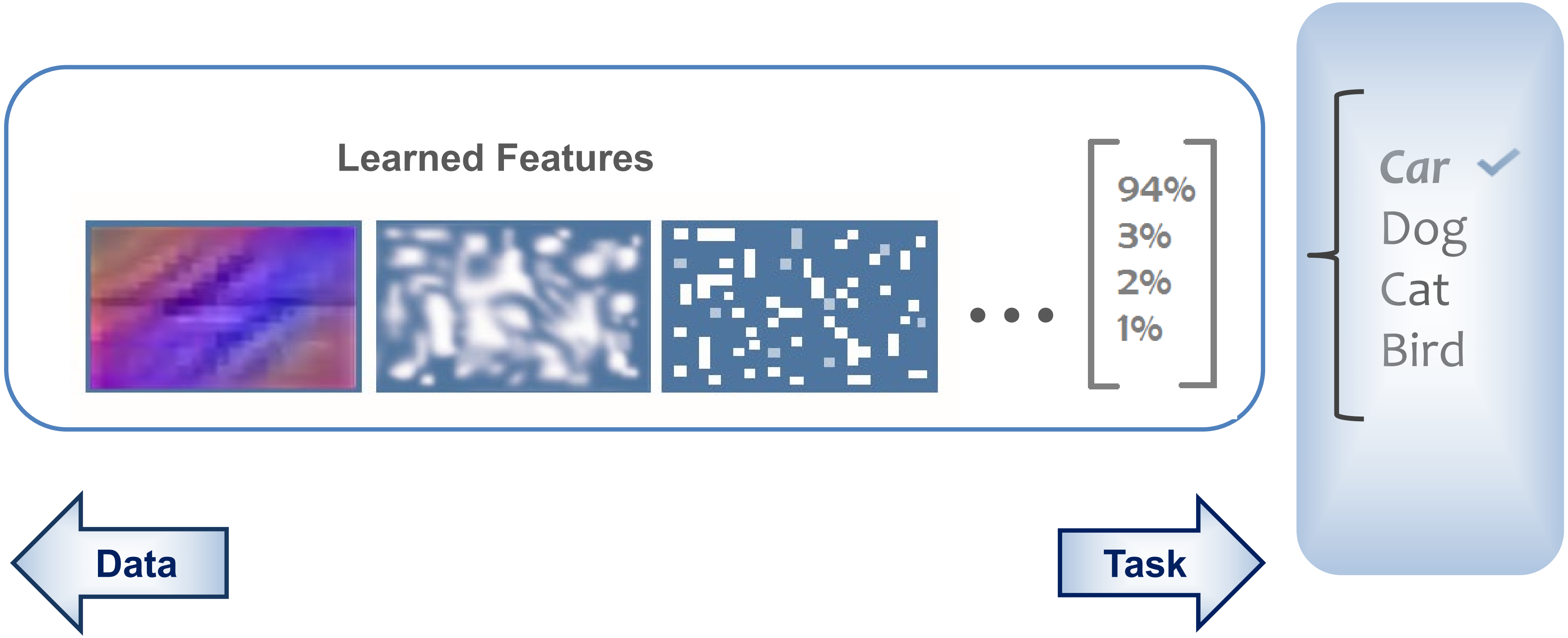
VGG-16
PRETRAINED MODEL

Caffe
MODELS



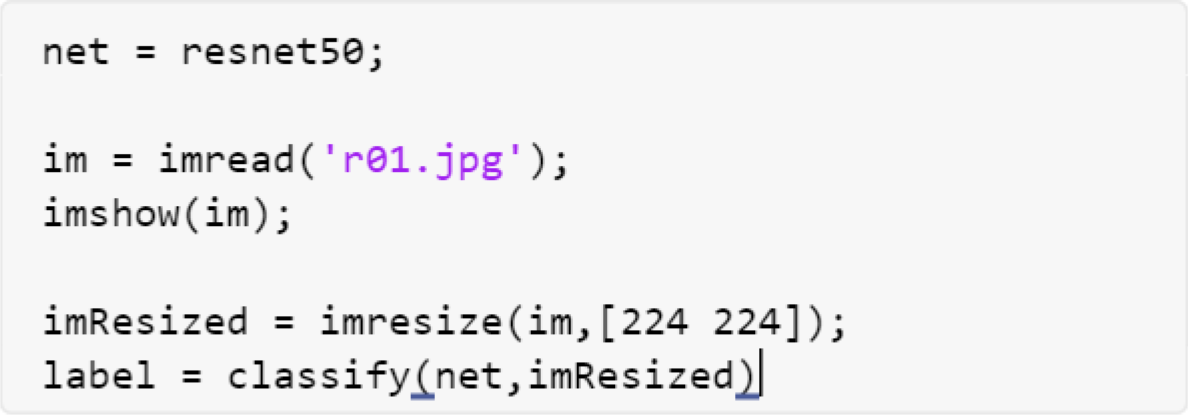
What is Deep Learning ?

Deep learning is a type of **machine learning** that learns tasks *directly* from data

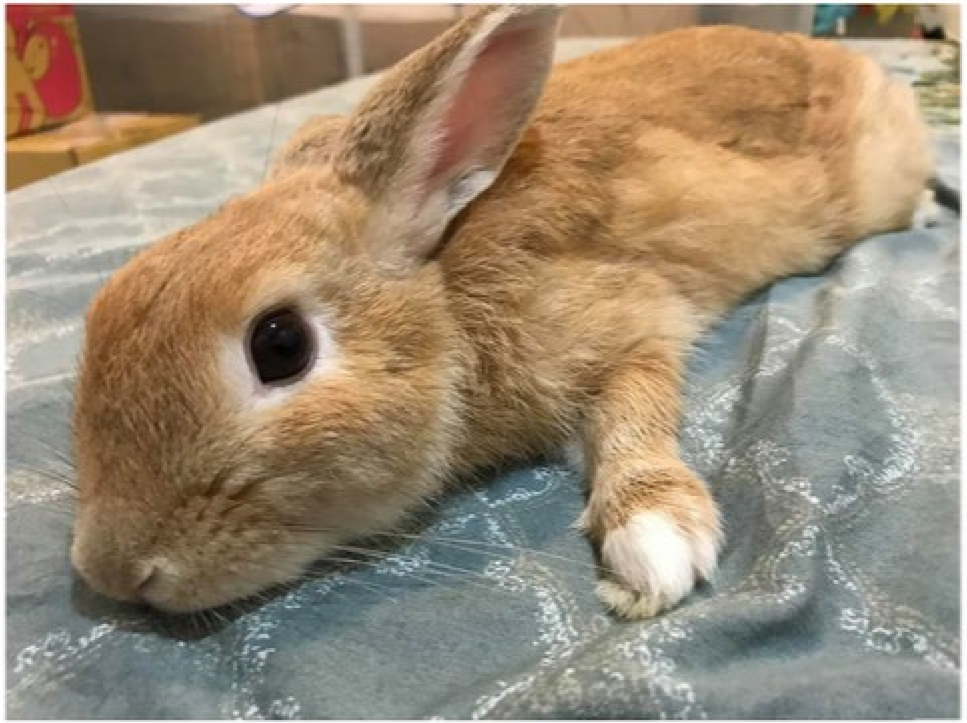


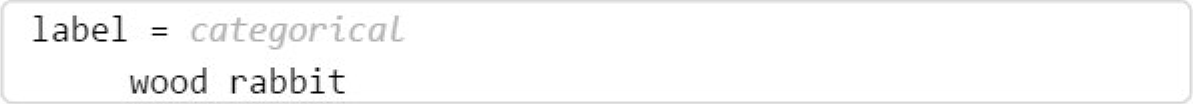
Deep Learning In 5 Lines of Code

Deep Learning In 5 Lines of Code



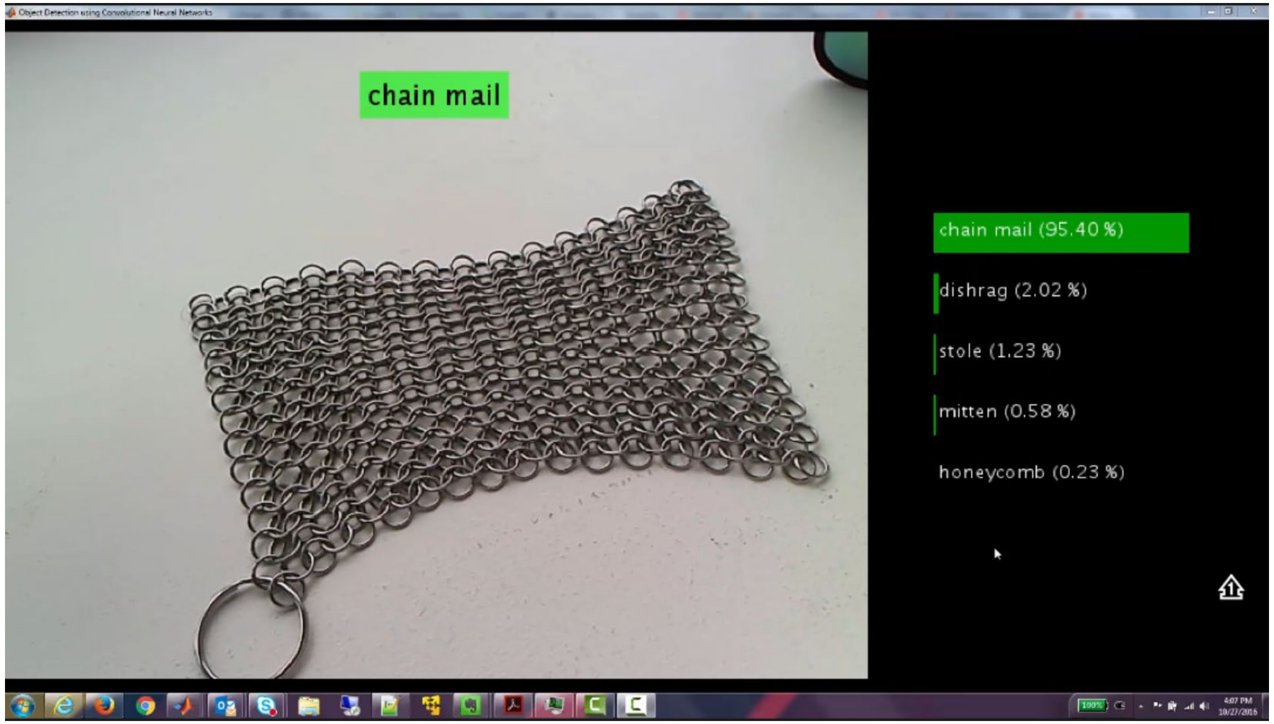
```
net = resnet50;

im = imread('r01.jpg');
imshow(im);

imResized = imresize(im,[224 224]);
label = classify(net,imResized)
```





```
label = categorical
    wood rabbit
```

Object Recognition Using Deep Learning



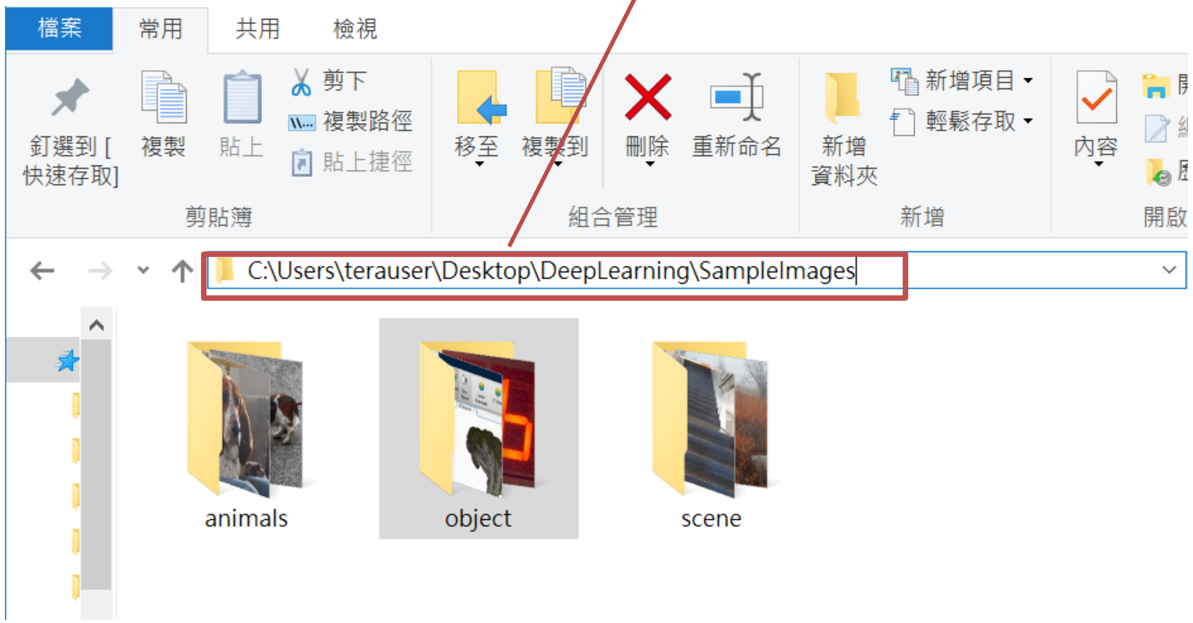
```
chain mail (95.40 %)
dishrag (2.02 %)
stole (1.23 %)
mitten (0.58 %)
honeycomb (0.23 %)
```

Training (GPU)	Millions of images from 1000 different categories
Prediction	Real-time object recognition using a webcam connected to a laptop

Manage Large Sets of Images – (1)

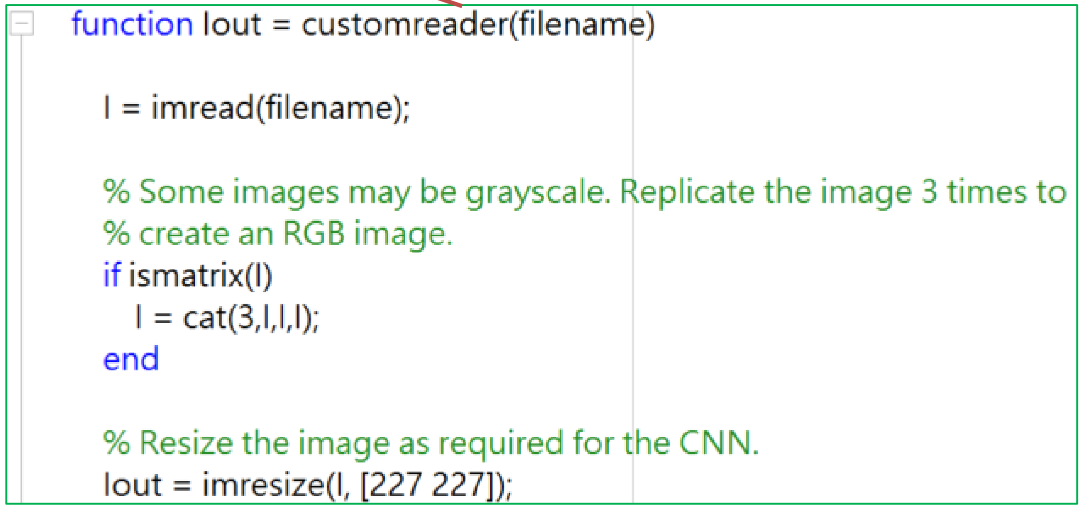
- imds = imageDatastore(ImagesPath, …

```
imds = imageDatastore( ImagesPath, …
                                   'IncludeSubfolders', true,…
                                   'LabelSource','foldernames’,…
                                   'ReadFcn', @customreader);
```



C:\Users\terauser\Desktop\DeepLearning\SampleImages

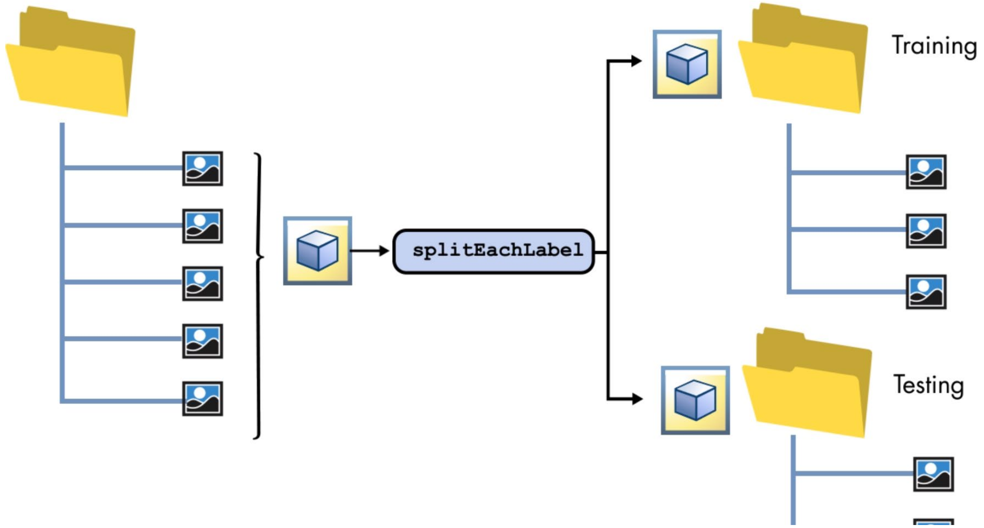
Customized MATLAB function

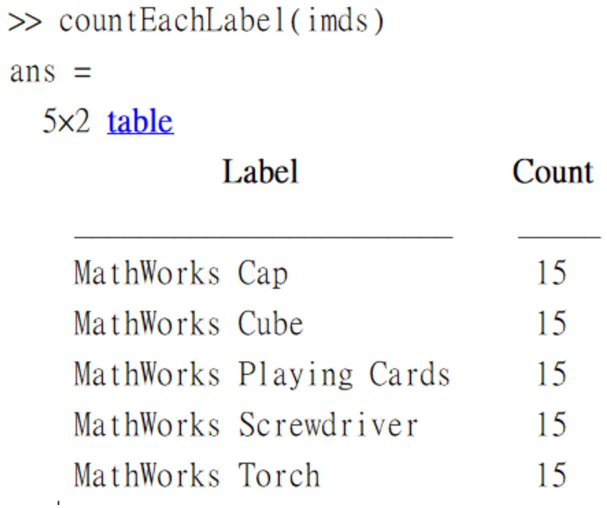


```
function Iout = customreader(filename)

    I = imread(filename);

    % Some images may be grayscale. Replicate the image 3 times to
    % create an RGB image.
    if ismatrix(I)
        I = cat(3,I,I,I);
    end

    % Resize the image as required for the CNN.
    Iout = imresize(I, [227 227]);
```

Manage Large Sets of Images – (2)

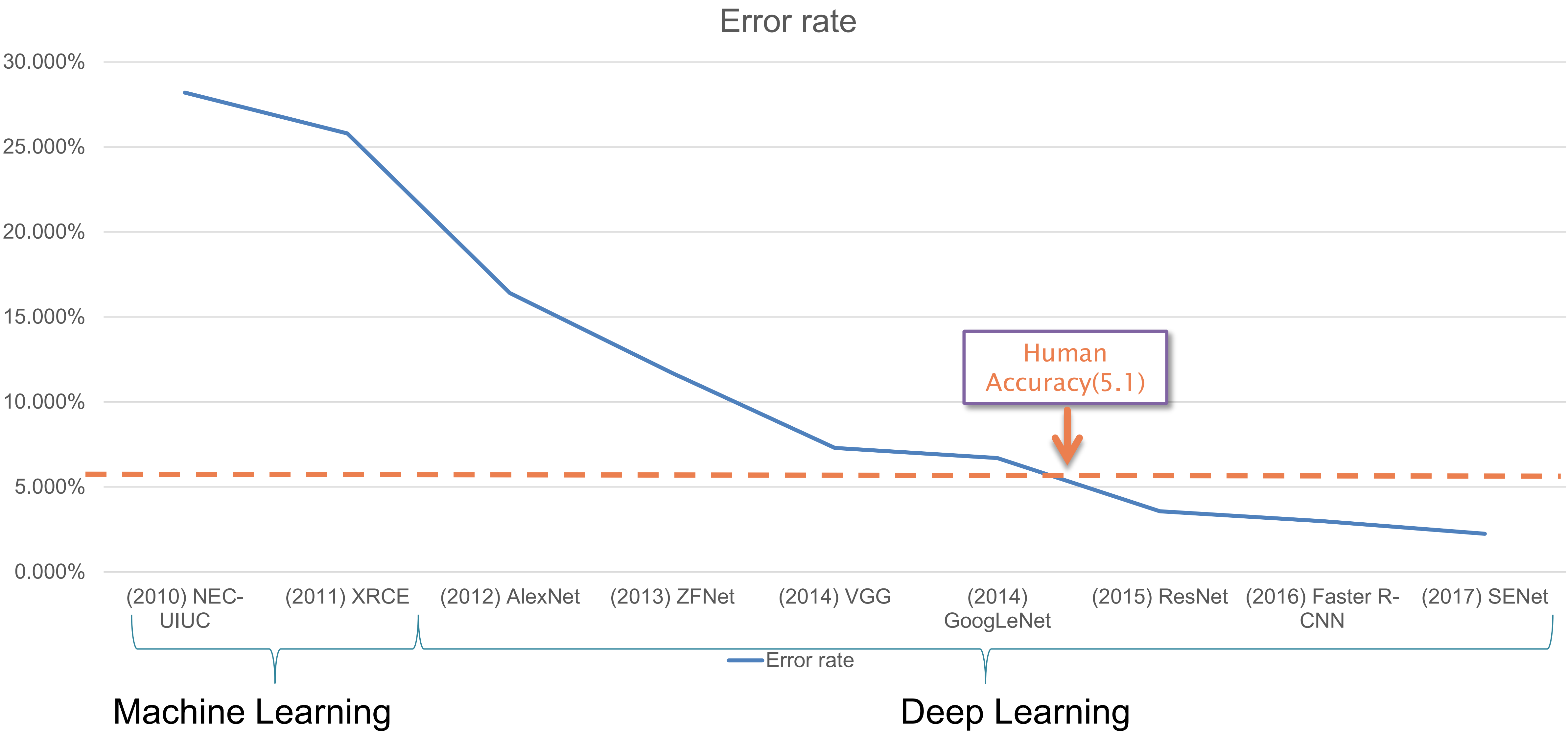
- Methods for ImageDatastore:
  - **splitEachLabel:** Split ImageDatastore labels by proportions
  - **countEachLabel:** Count files in ImageDatastore labels
  - **readimage:** Read specified image from ImageDatastore





```
>> countEachLabel(imds)
ans =
  5×2 table
              Label                  Count

    _________________________        _____

    MathWorks Cap                     15
    MathWorks Cube                    15
    MathWorks Playing Cards           15
    MathWorks Screwdriver             15
    MathWorks Torch                   15
```

Why is Deep Learning So Popular Now?

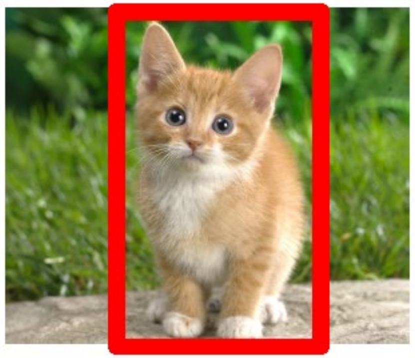


Computer Vision Tasks in Deep Learning

Classification

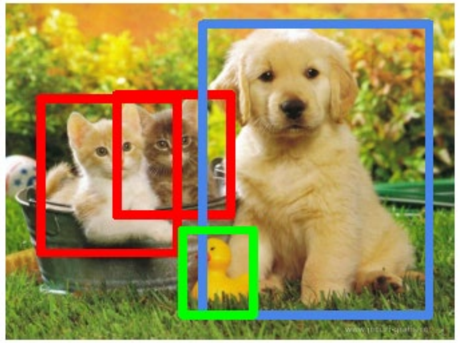
CAT

Classification + Localization



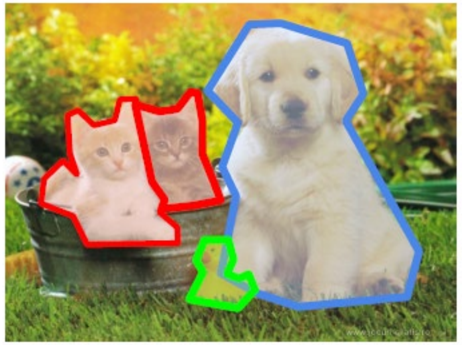
CAT

Object Detection



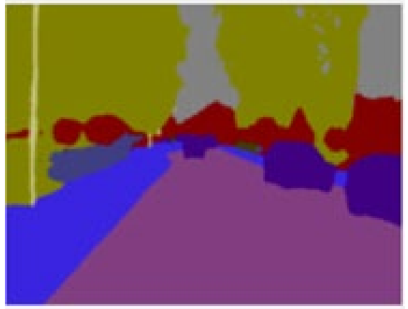
CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

Semantic Segmentation



Single object

Multiple object

No object , just pixel

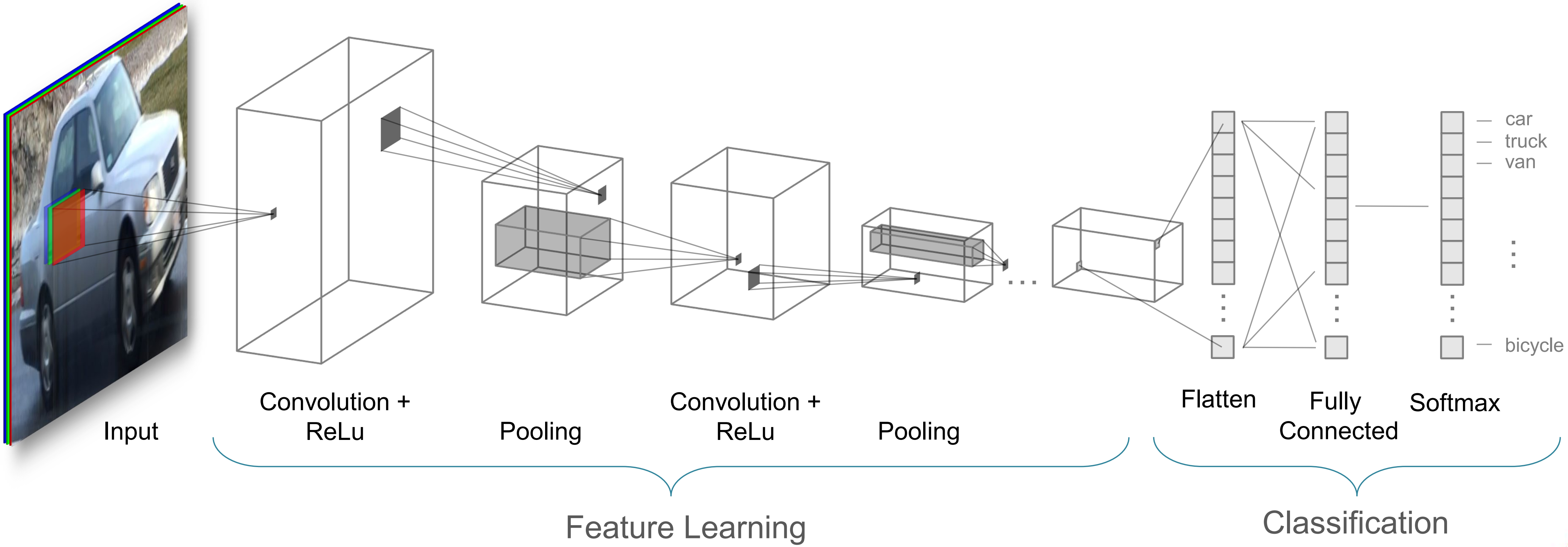
Src: http://blog.csdn.net/u011974639

Agenda

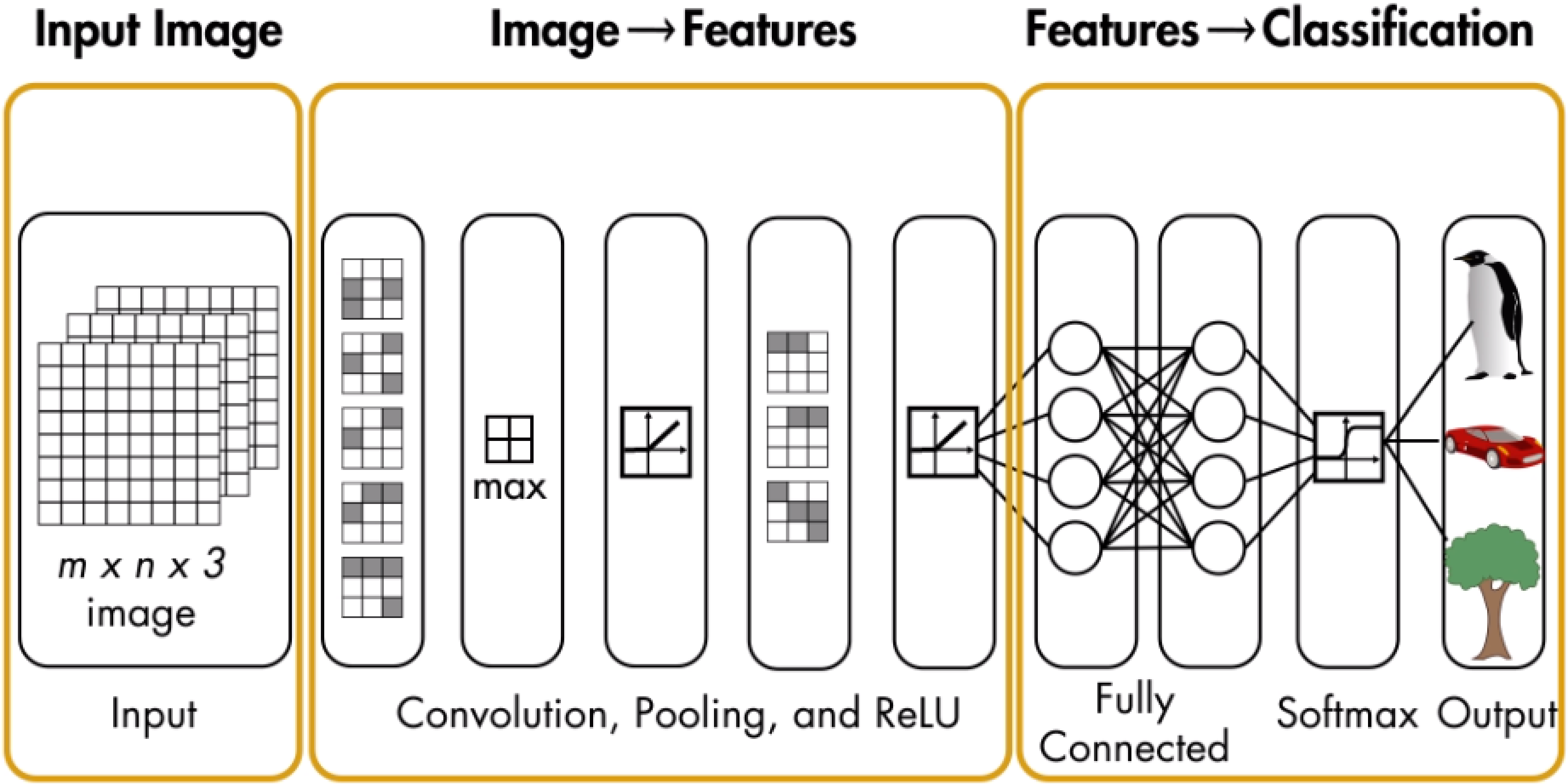
- Machine Learning Overview
- How to use Pretrained model
- Create Deep Learning Model(MNIST)
- Try to do Transfer Learning

Deep Learning:
Convolutional Neural Networks (CNN/ConvNet)

- Multi-layered neural network

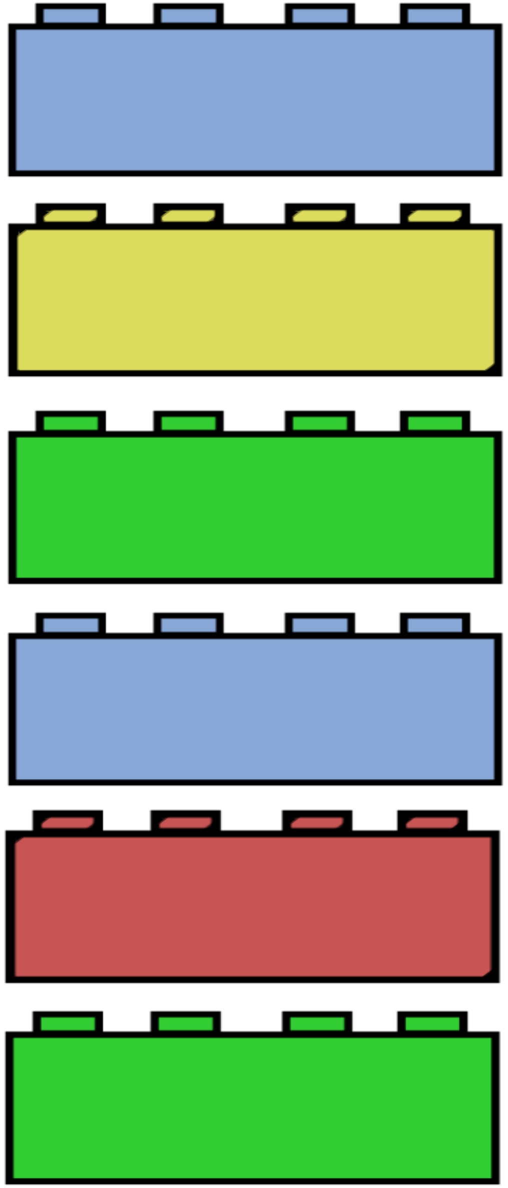


Convolutional Neural Networks (CNN/ConvNet)



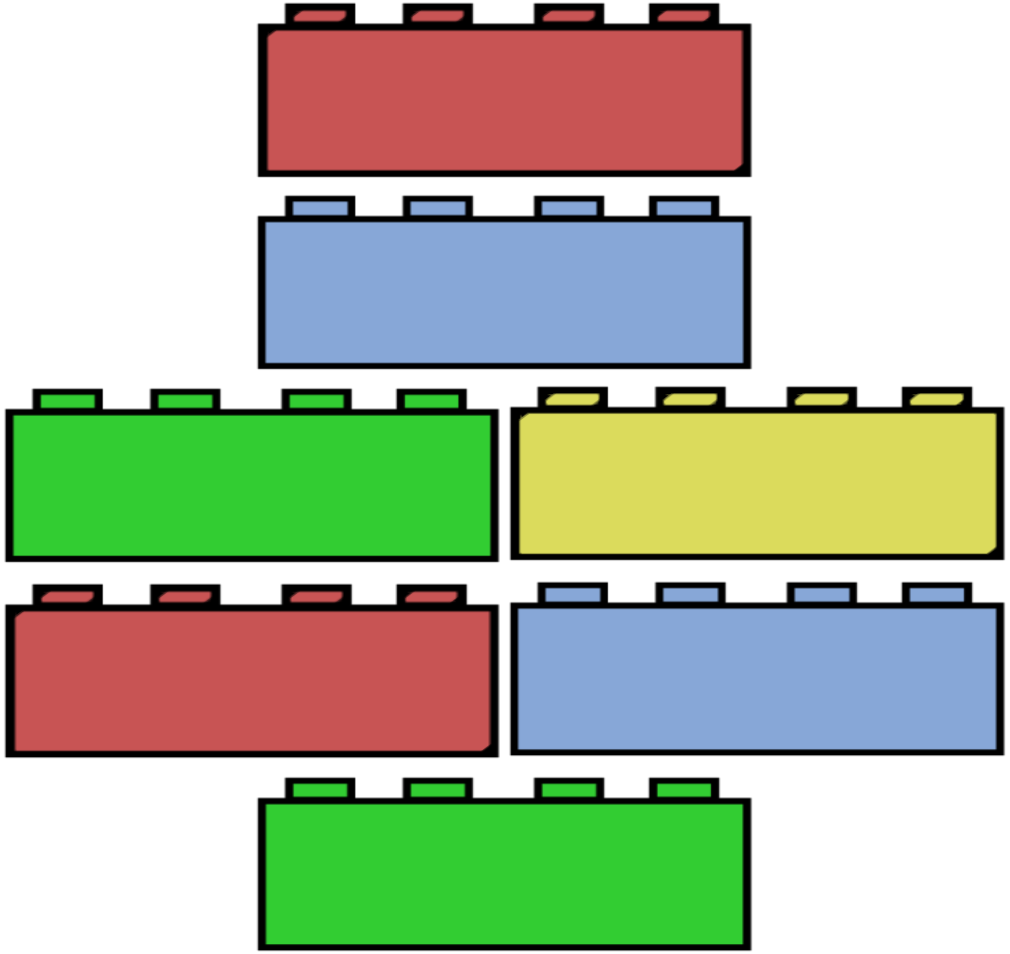
Thinking about Layers

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



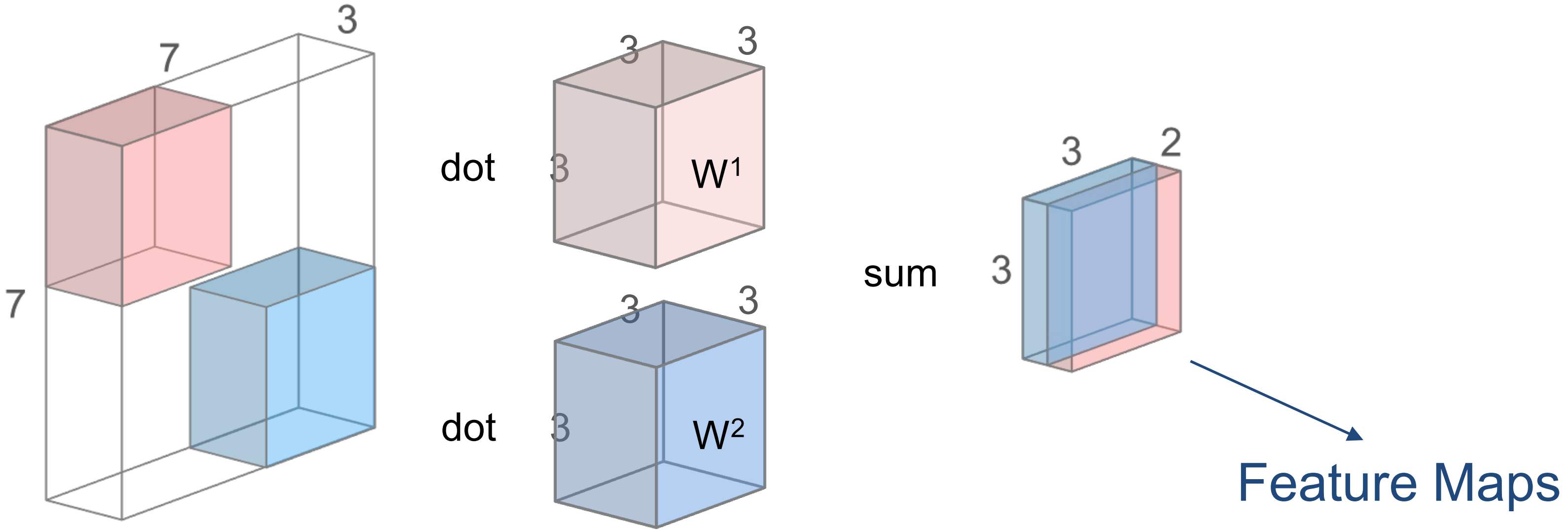
Thinking about Layers

- Layers are like blocks
  - Stack 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



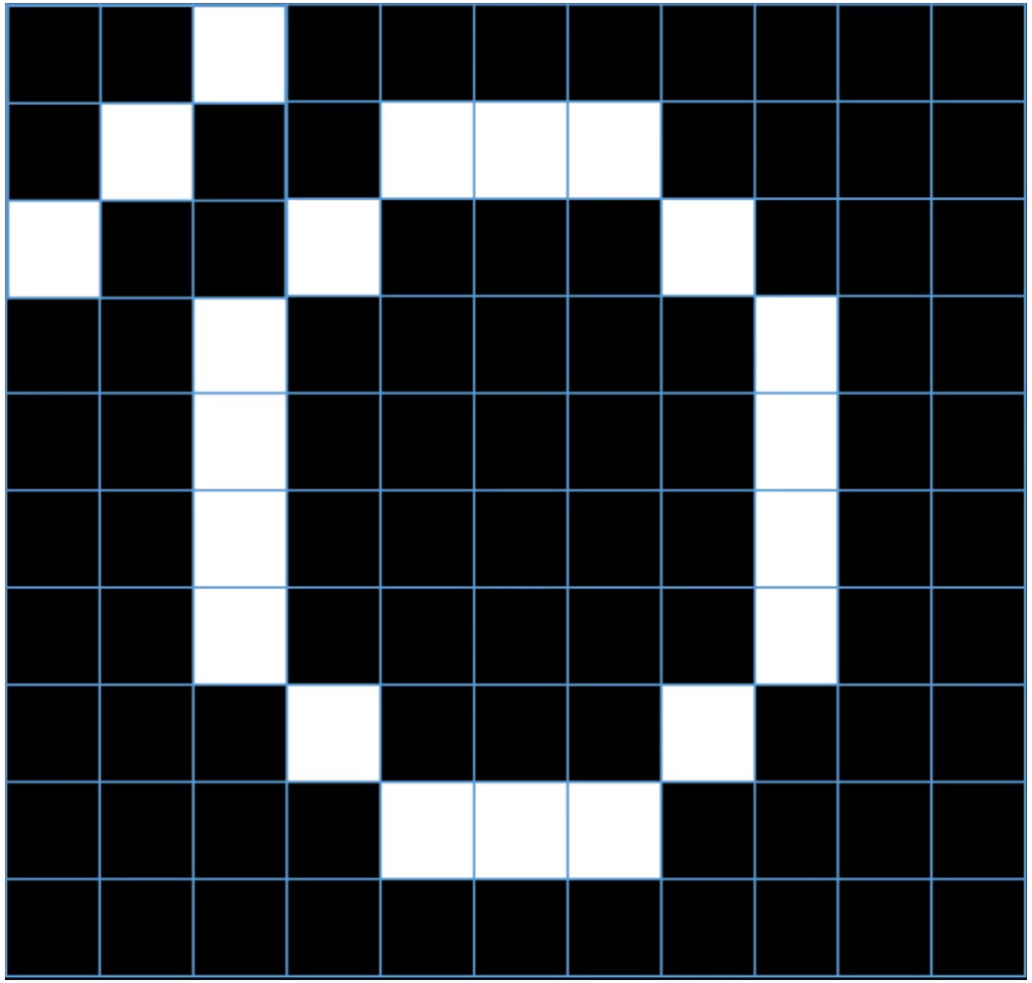
Convolution Layer

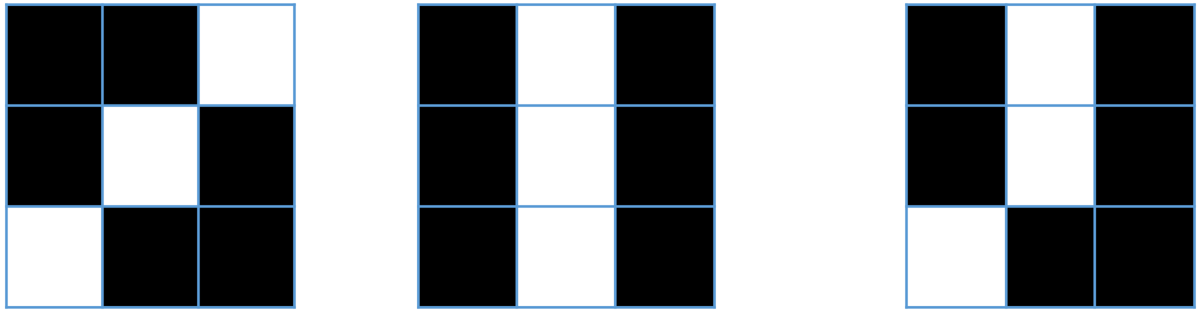
- Core building block of a CNN
- Convolve the filters sliding them across the input, computing the dot product



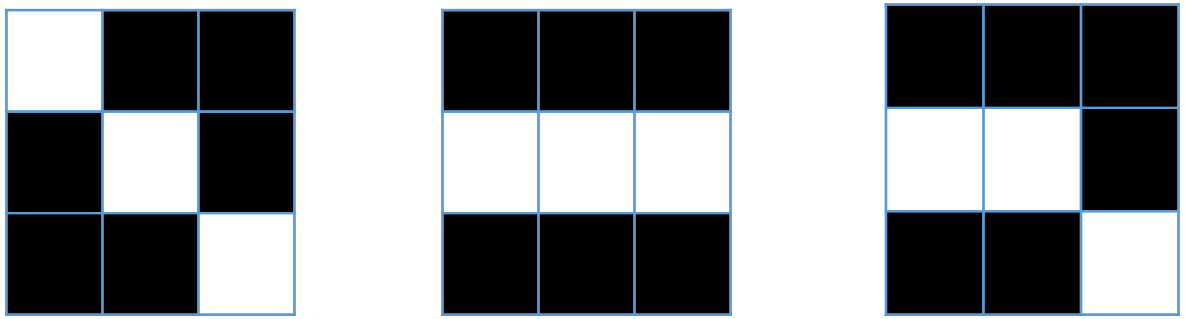
- Intuition: learn filters that activate when they “see” some specific feature

Convolution Layers Search for Patterns





These patterns would be common in the number 0

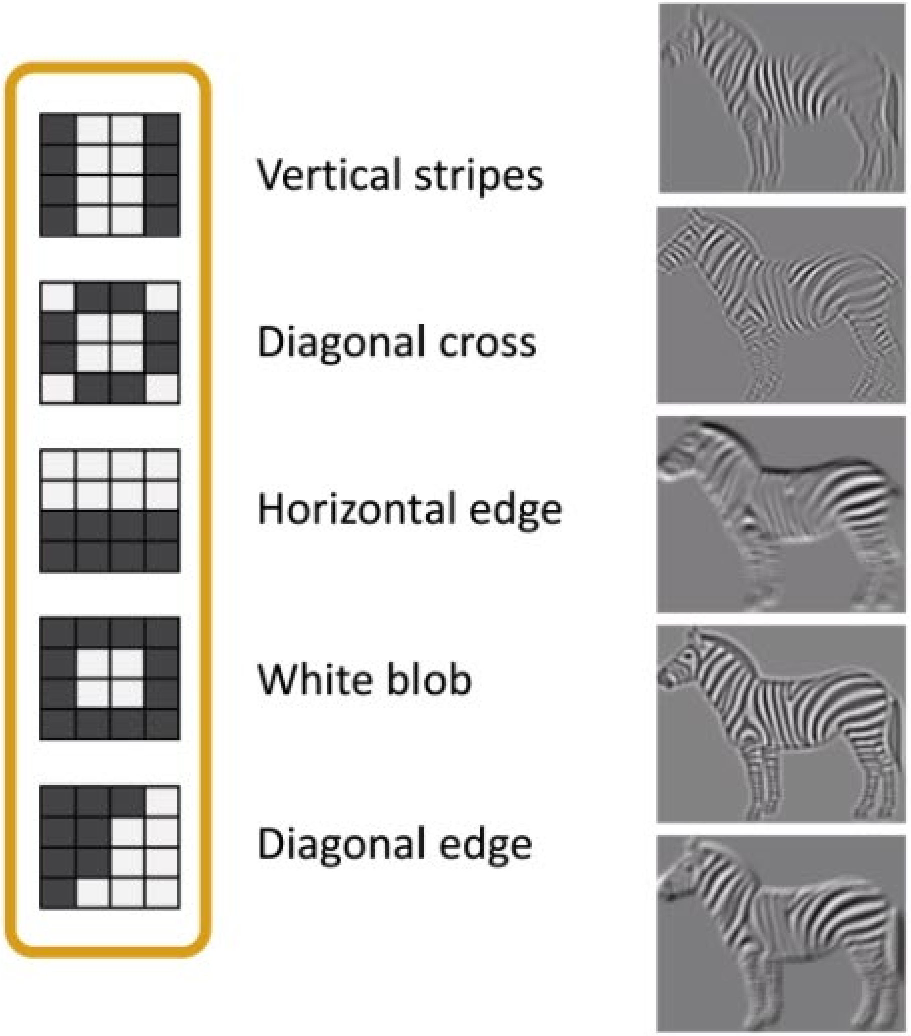


Convolution Layer – (2)

Input image

Filters

Feature Maps



Vertical stripes

Diagonal cross

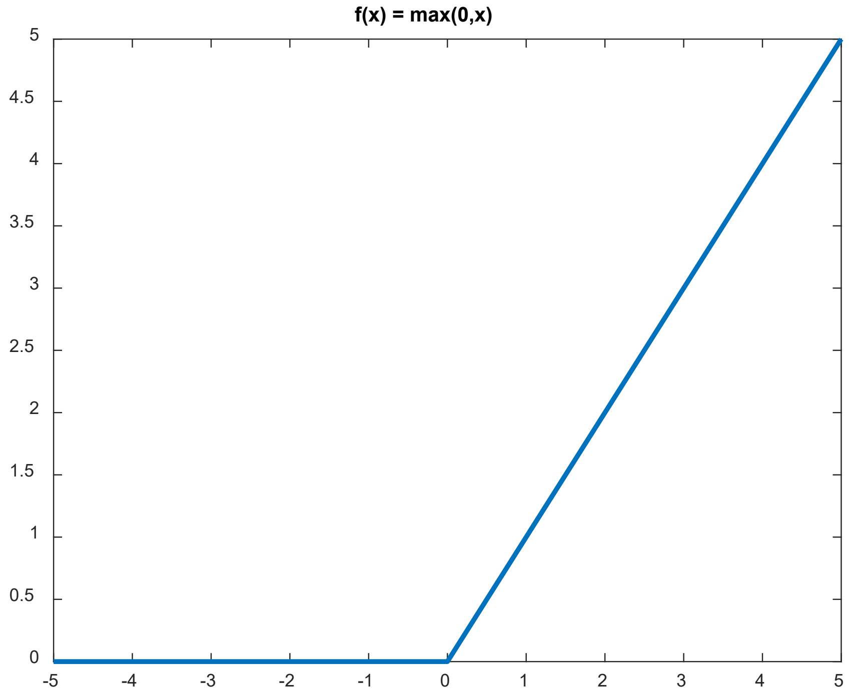
Horizontal edge

White blob

Diagonal edge

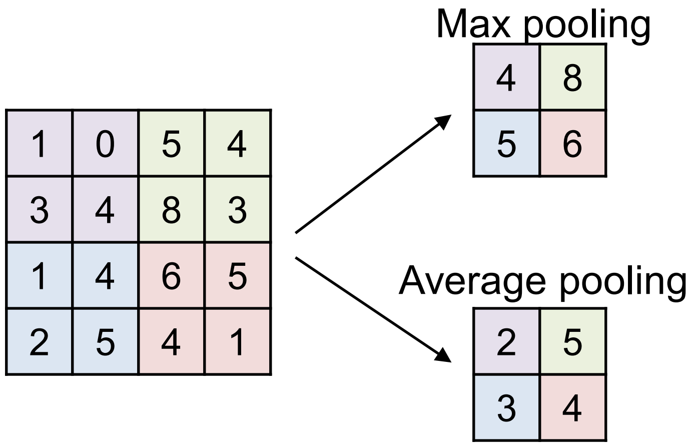
Rectified Linear Unit (ReLU) Layer

- Frequently used in combination with Convolution layers
- Allows for faster and more effective training
- $\boldsymbol{f(x) = max(0, x)}$, activation is thresholded at 0



Pooling Layer

- Perform a downsampling operation across the spatial dimensions
- Goal: reducing the number of parameters
- Max pooling and average pooling methods



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

Max pooling

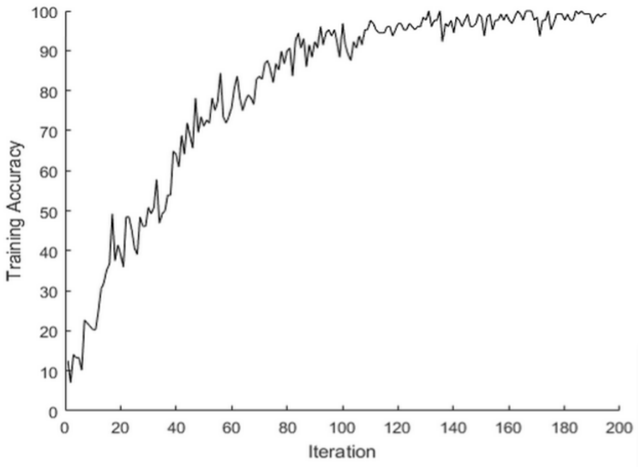
4	8
5	6

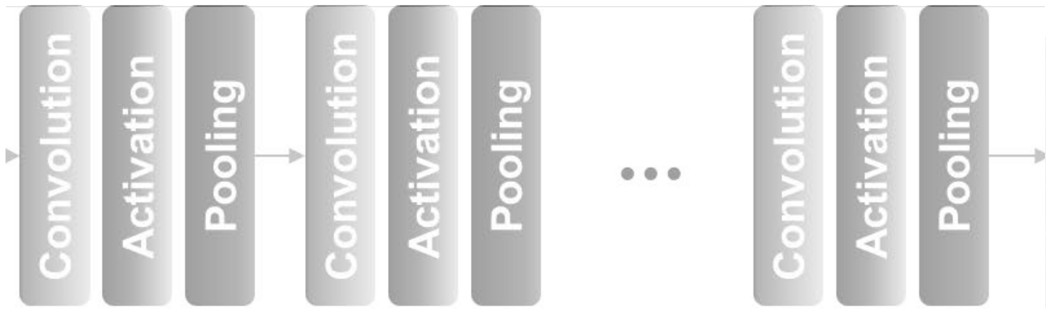
Average pooling

2	5
3	4

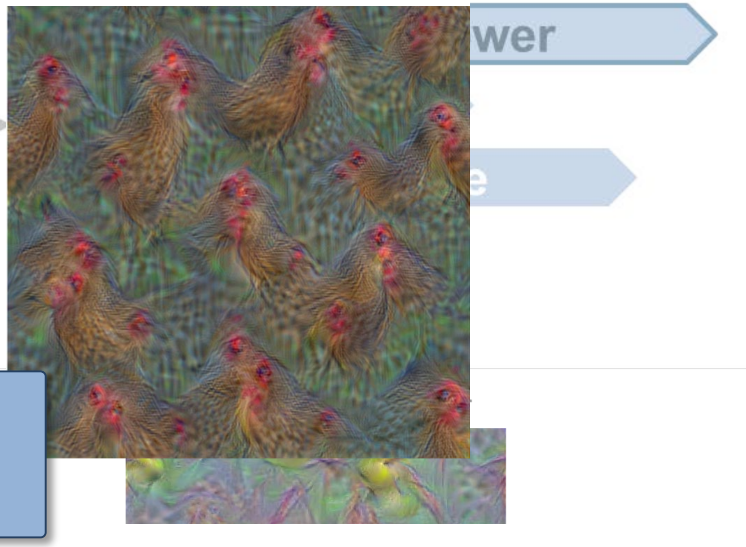
Visualizing and Debugging Intermediate Results

Training Accuracy Visualization



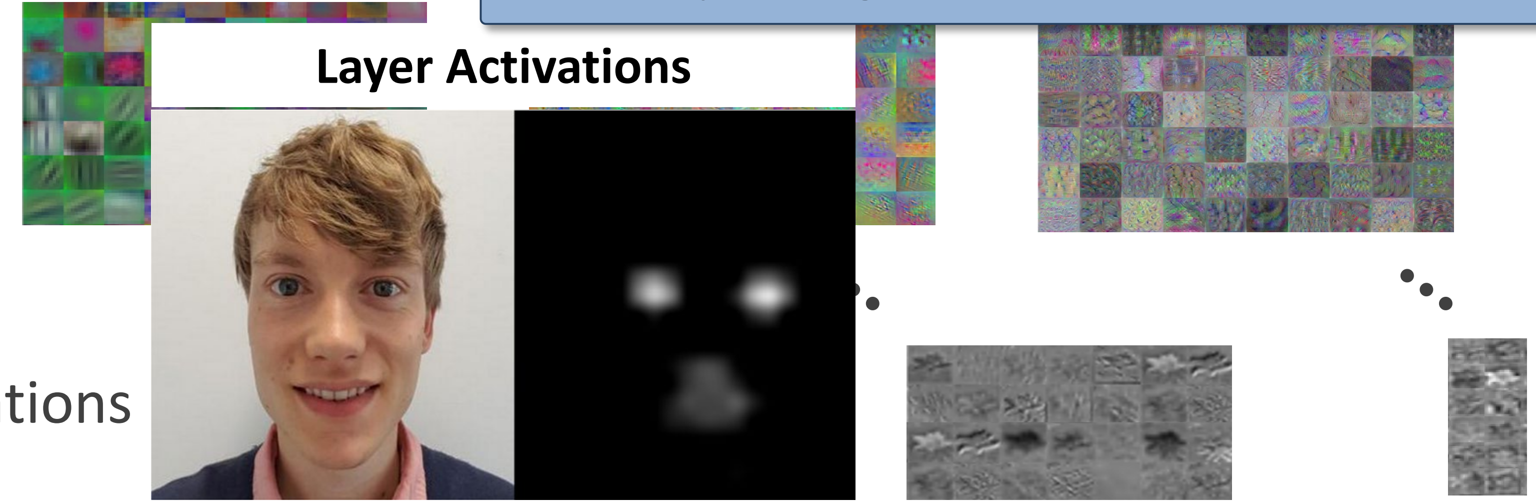


Deep Dream



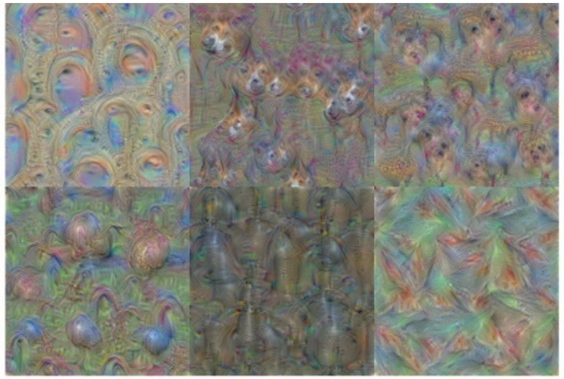
- Many options for visualizations and debugging
- Examples to get started

Filters



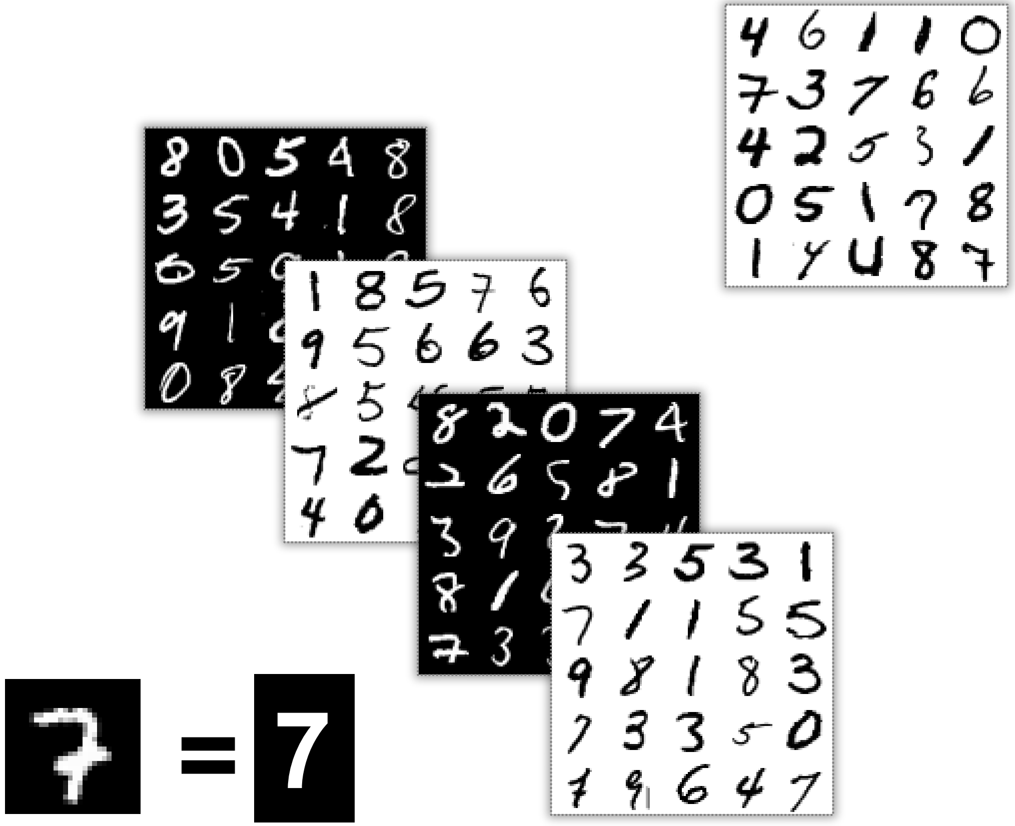
Activations

Feature Visualization



MNIST: The "Hello, World!" of Computer Vision – (1)

What?	A set of handwritten digits from 0-9
Why?	An easy task for machine learning beginners
How many?	60,000 training images 10,000 test images
Best results?	99.79% accuracy



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

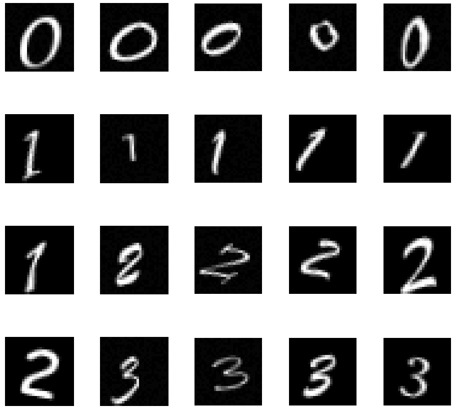
MNIST: The “Hello, World!” of Computer Vision – (2)

```
imds= imageDatastore(location);
```

```
layers = [ ...
    imageInputLayer([28 28 1], 'Normalization', 'none');
    convolution2dLayer(5, 20);
    reluLayer();
    maxPooling2dLayer(2, 'Stride', 2);
    fullyConnectedLayer(10);
    softmaxLayer();
    classificationLayer()];
```

```
opts = trainingOptions('sgdm',…
                   'MaxEpochs', 5,…
                   'InitialLearnRate',0.01,...
                   'MiniBatchSize',128,...
                   'ExecutionEnvironment', ‘multi-gpu’,…
                   'Plots','training-progress');

net = trainNetwork(imds, layers, opts);
```

```
label = classify(net, img)
```



Handle training images.

Create the layers for CNN.

Set training options and train.

Classify image.

Agenda

- Machine Learning Overview
- How to use Pretrained model
- Create Deep Learning Model(MNIST)
- Try to do Transfer Learning

Transfer Learning with Pretrained Models

Inception-v3

MobileNet-v2

VGG-16

Inception-ResNet-v2

ResNet-18/50/101

GoogLeNet

DenseNet-201

NASNet

SqueezeNet

AlexNet

Places365-GoogLeNet

Xception

Import & Export Models Between Frameworks

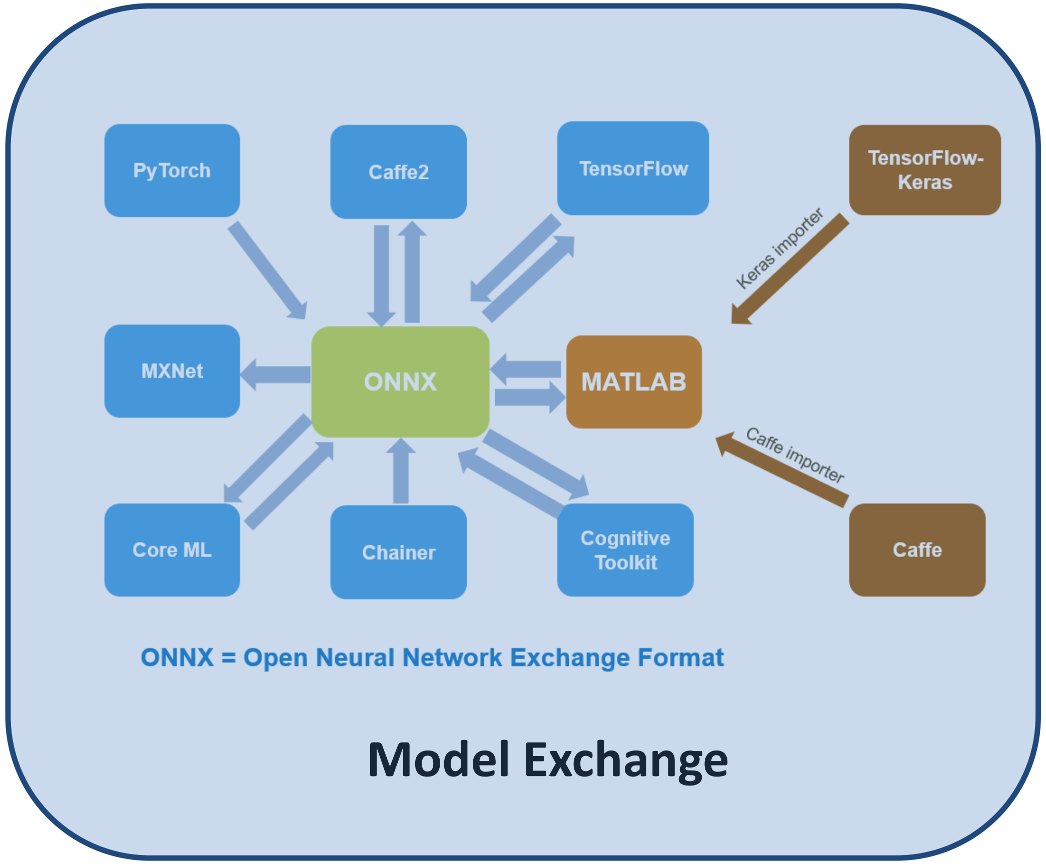
Keras-Tensorflow Importer

Caffe Model Importer

ONNX Model Converter

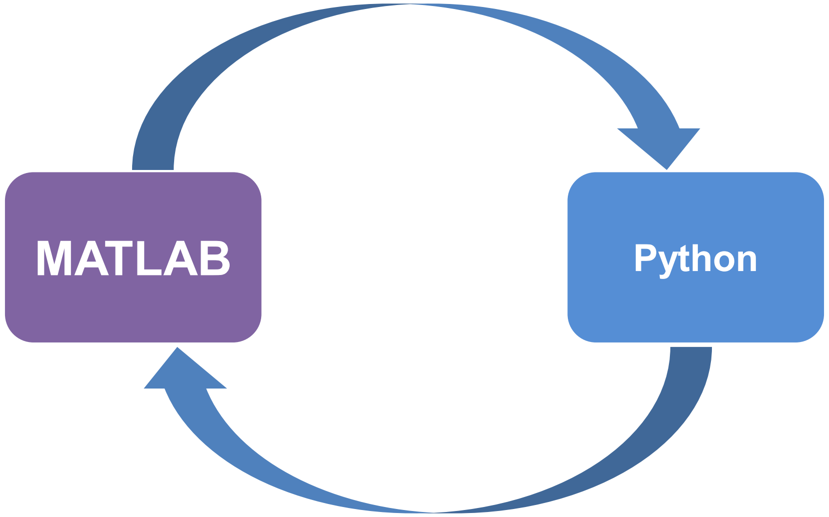
More comprehensive list here: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

Two Ways to Work with TensorFlow and PyTorch



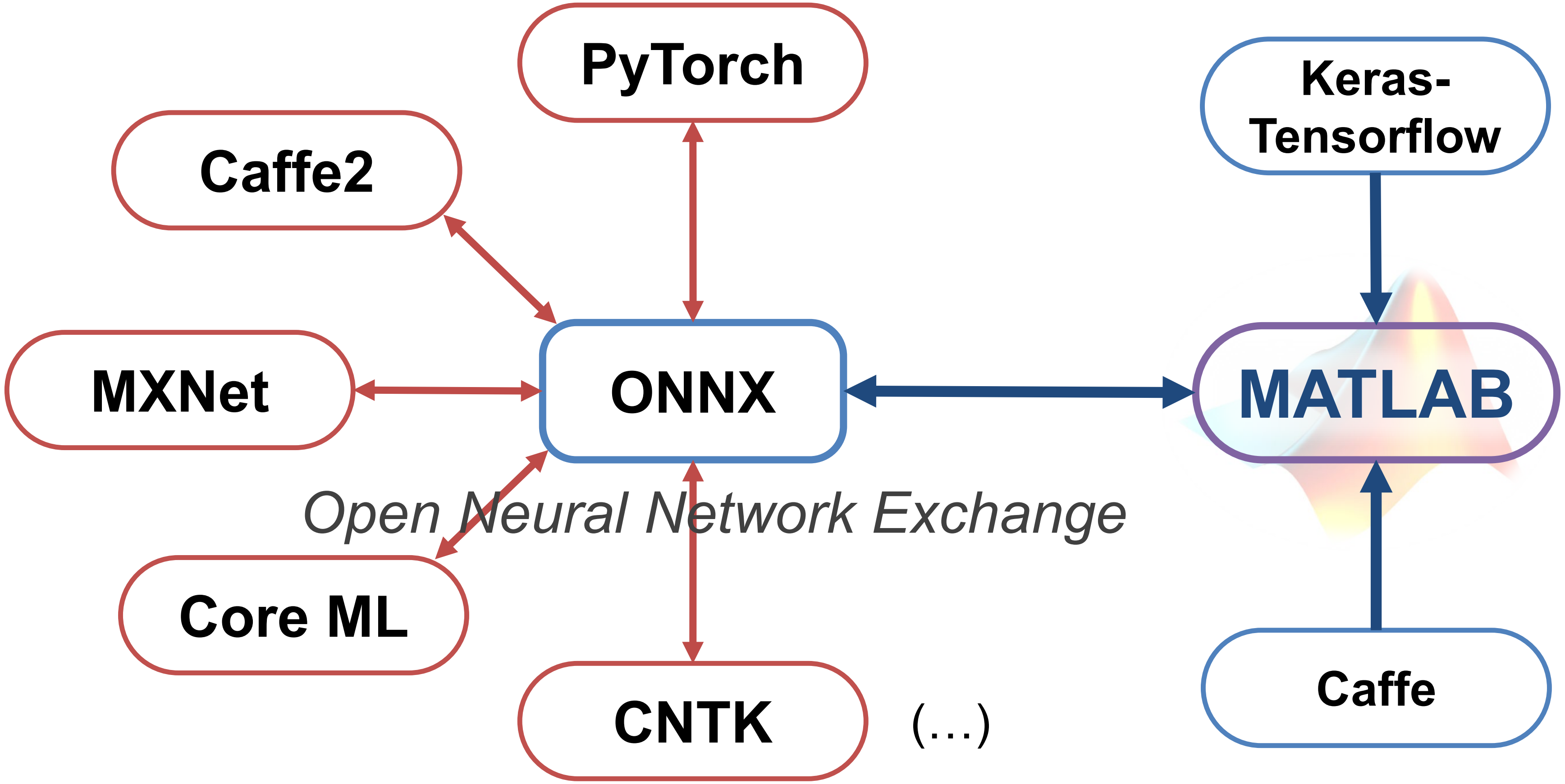
ONNX = Open Neural Network Exchange Format

Model Exchange

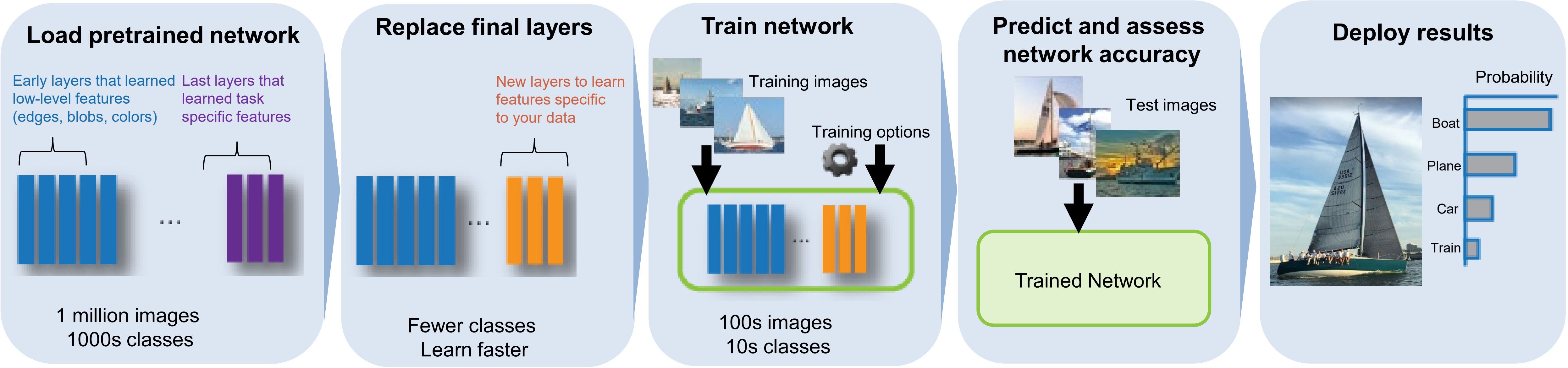


Co-execution (Python and C++)

Model Exchange with MATLAB



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

Deploy results

Thank you!