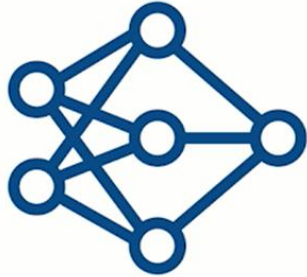
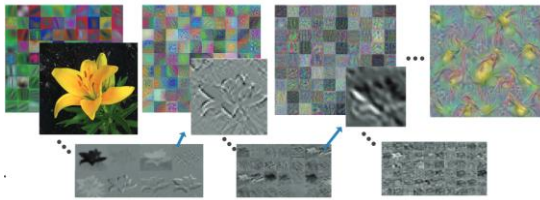


Demystifying Deep Learning

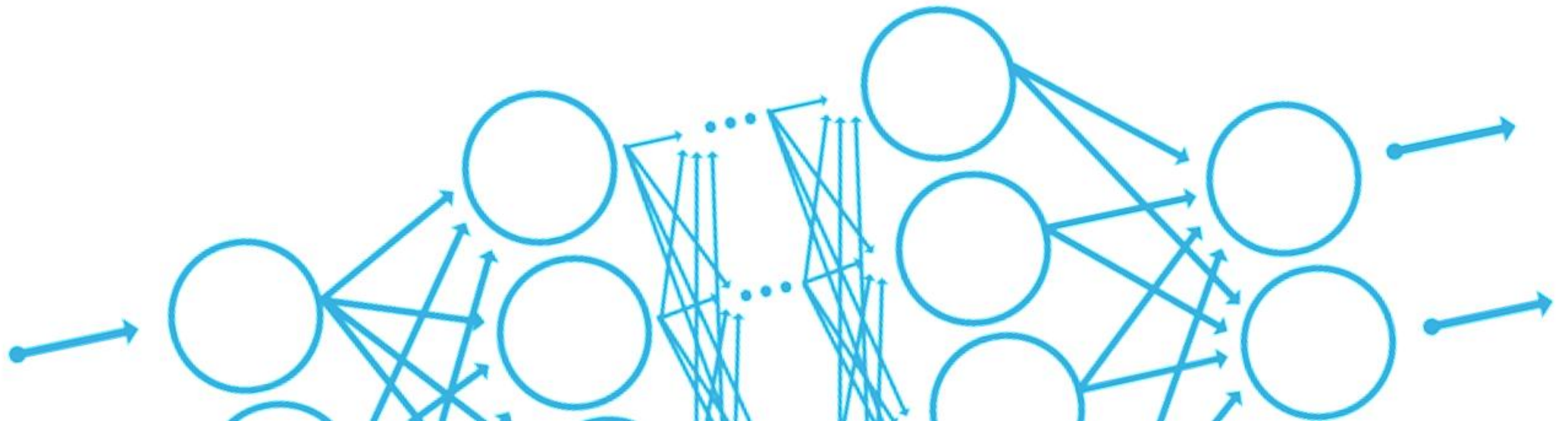
A Practical Approach in MATLAB



Timothy Jones

Applications Engineer

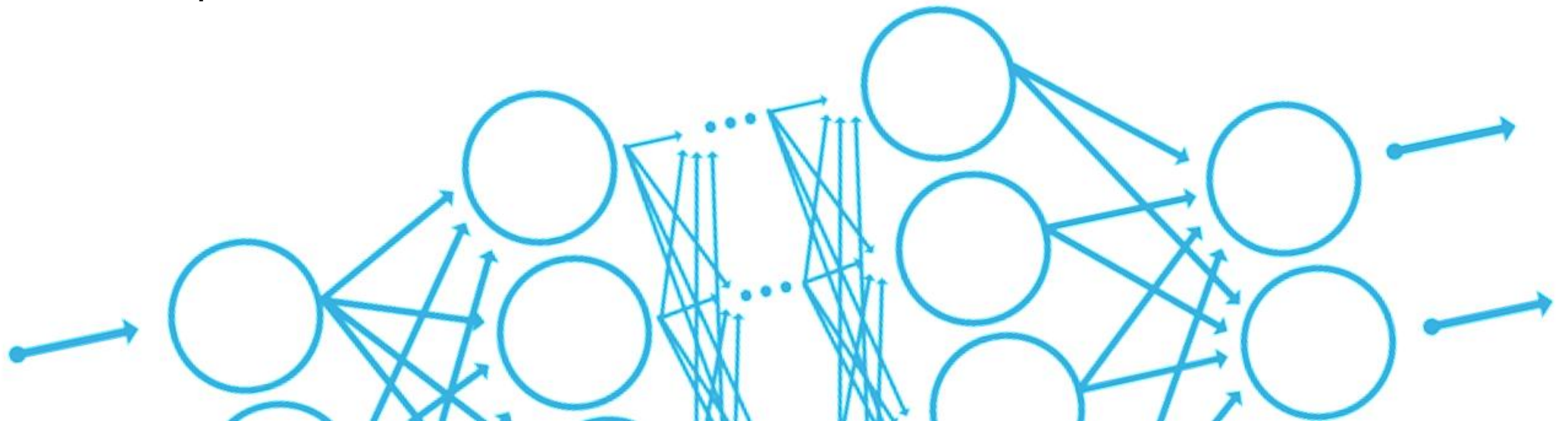
What is Deep Learning?



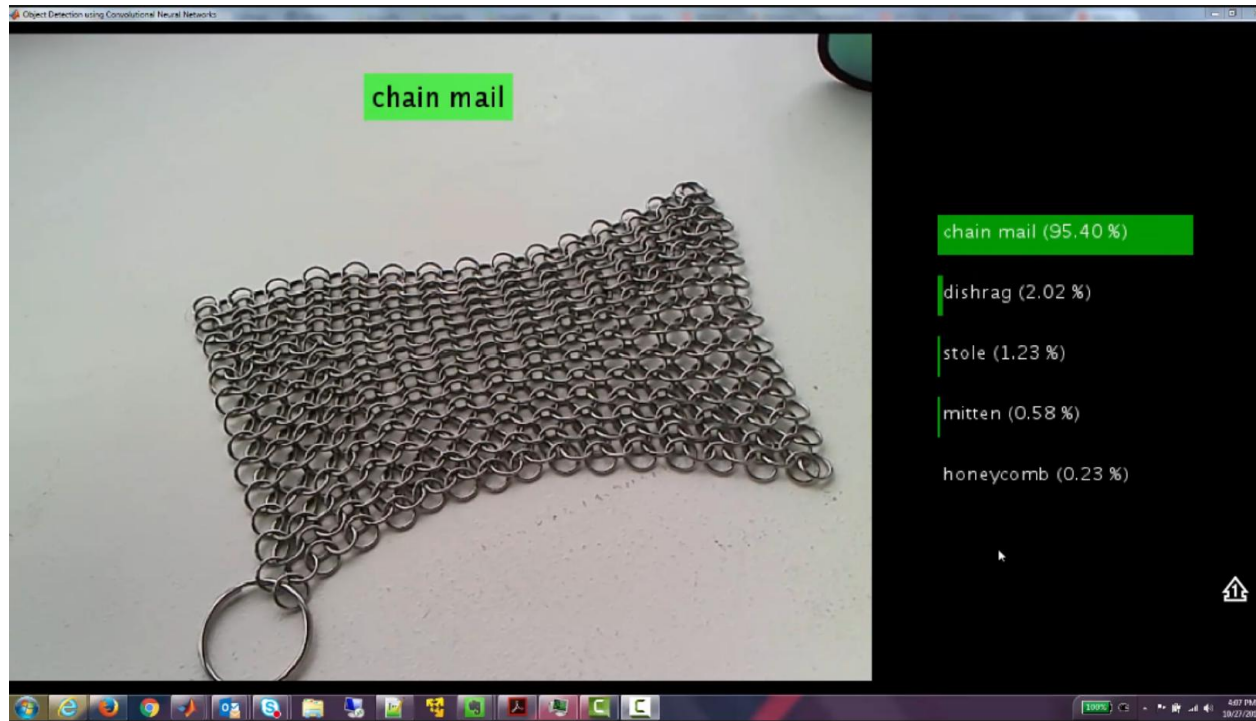
Deep learning is a type of machine learning in which a model learns to perform classification tasks directly from images, text, or sound.

Deep learning is usually implemented using a **neural network architecture**.

The term “deep” refers to the **number of layers** in the network—the more layers, the deeper the network.

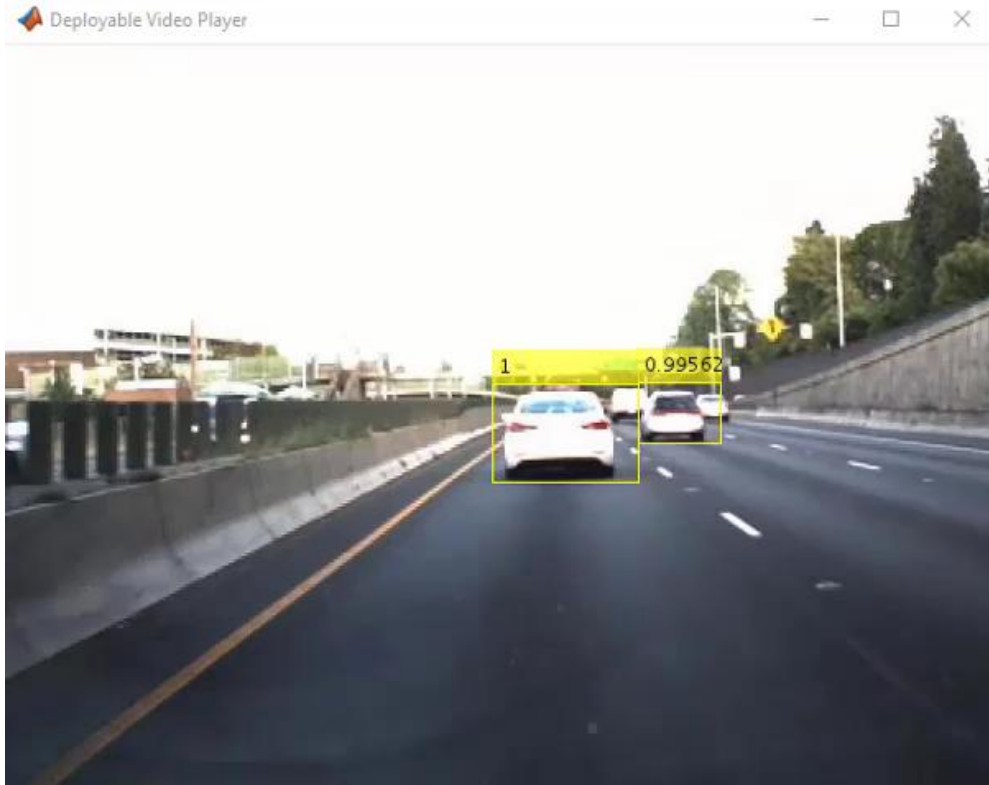


Example 1: Object recognition using deep learning



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

Example 2: Detection and localization using deep learning

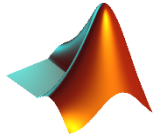


Regions with Convolutional Neural Network Features (R-CNN)



Semantic Segmentation using SegNet

Agenda



Why deep learning?

MNIST: The "Hello, World!" of computer vision

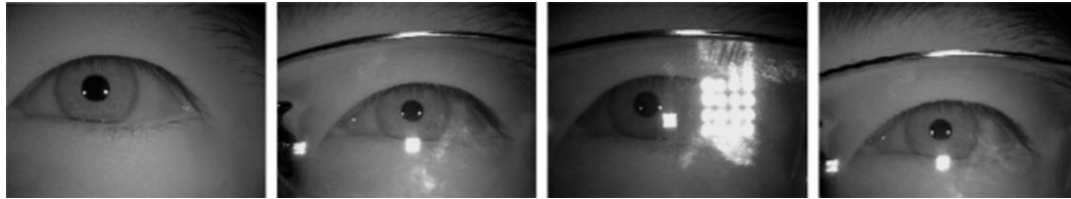
Transfer learning with CNNs

Semantic segmentation

Ground truth labeling of datasets

Everything else in deep learning...

Diverse Applications of Deep Learning



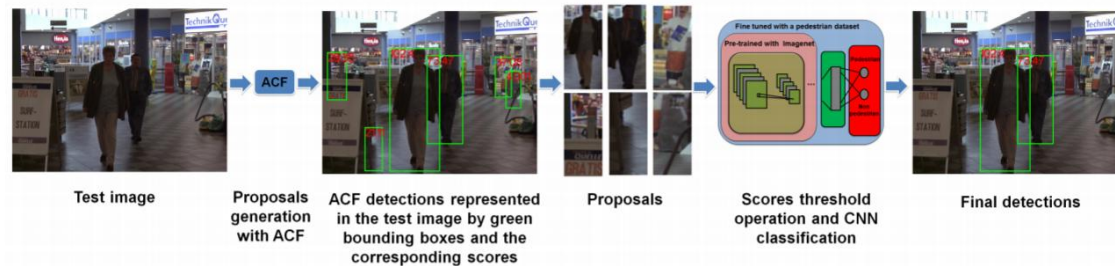
Iris Recognition – 99.4% accuracy¹

MatConvnet



Rain Detection and Removal³

MatCaffe

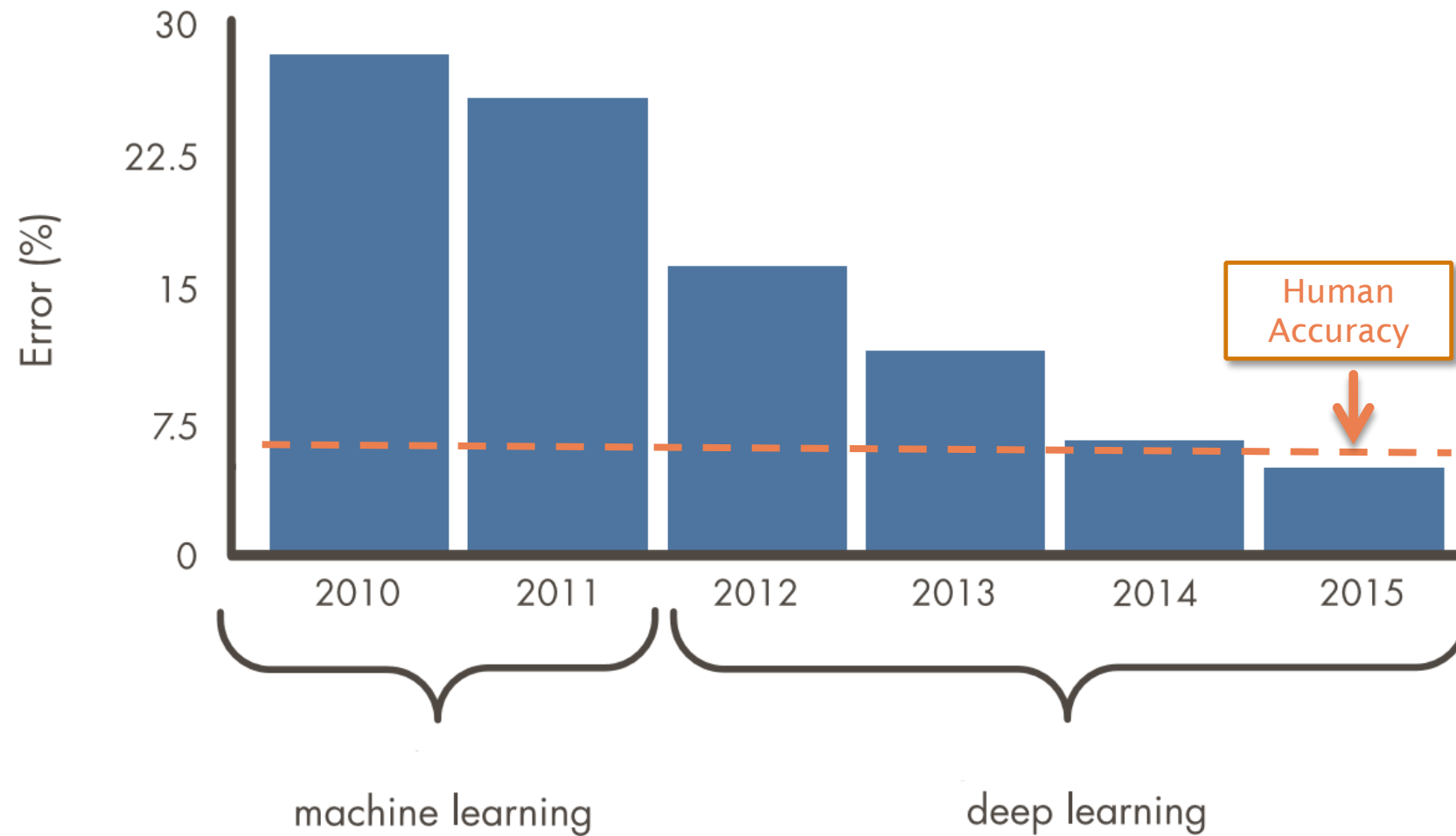


Human Aware Navigation for Robots²

MatConvnet

1. 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
2. "A Real-Time Pedestrian Detector using Deep Learning for Human-Aware Navigation" David Ribeiro, Andre Mateus, Jacinto C. Nascimento, and Pedro Miraldo
3. Deep Joint Rain Detection and Removal from a Single Image" Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan

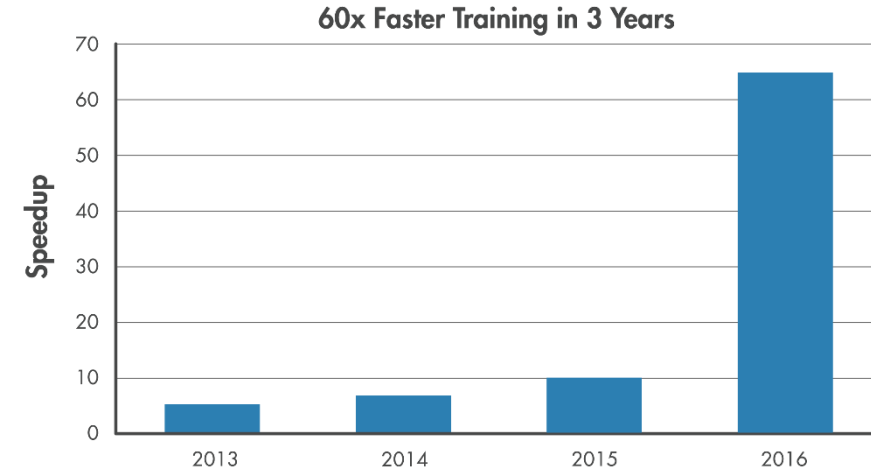
Why is Deep Learning So Popular Now?



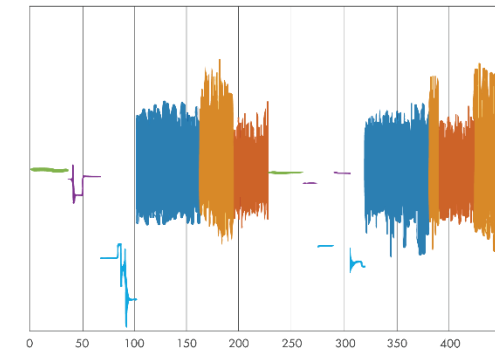
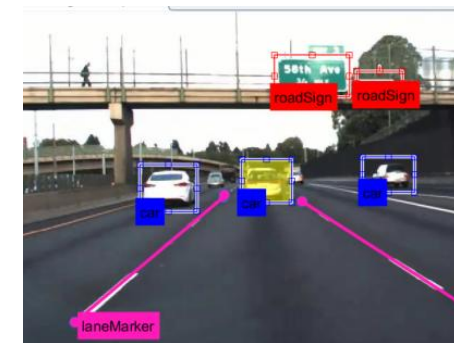
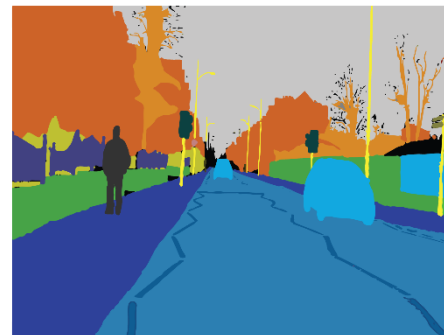
Source: ILSVRC Top-5 Error on ImageNet

Deep Learning Enablers

- Increased GPU acceleration
- World-class models
- Labeled public datasets



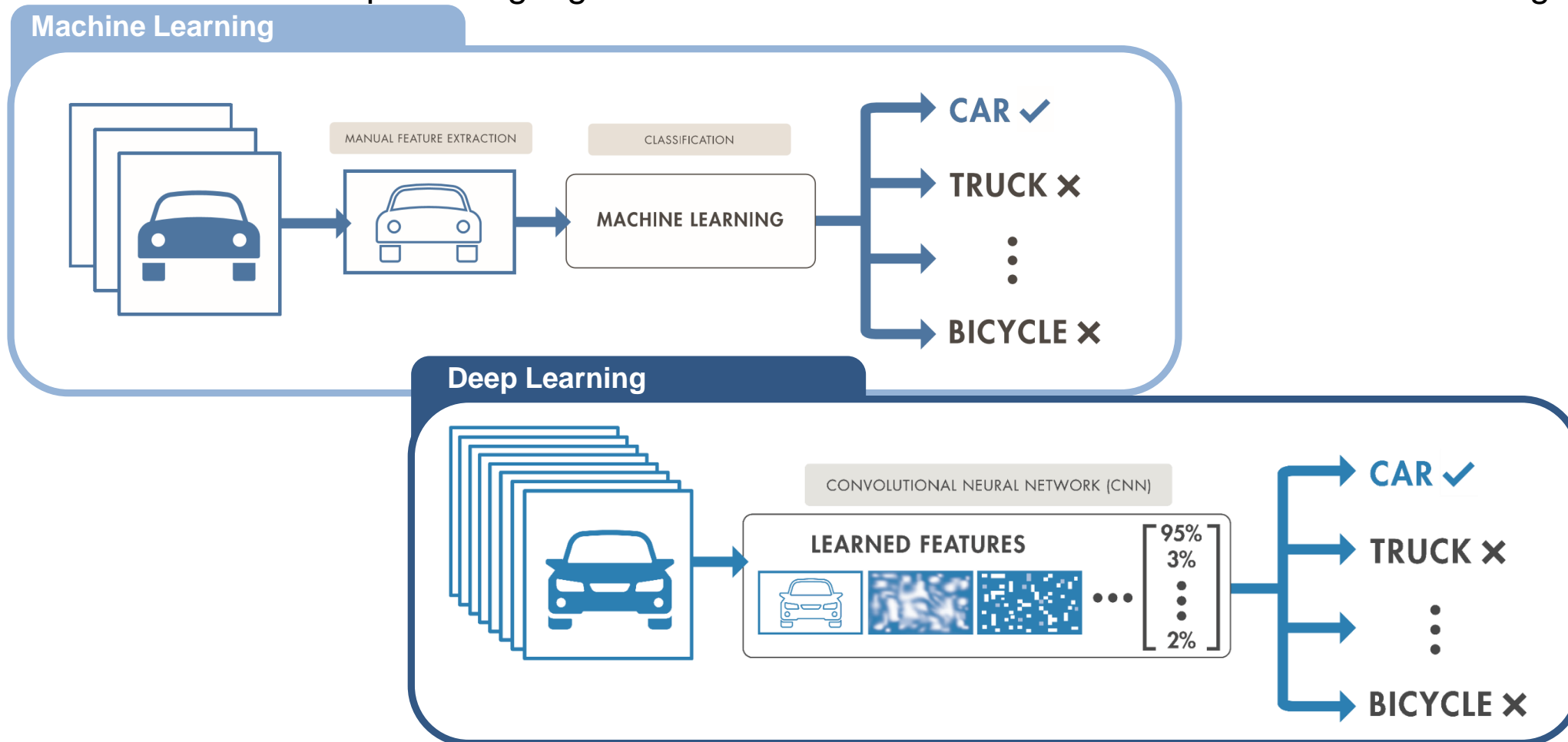
<p>AlexNet PRETRAINED MODEL</p>	<p>VGG-16 PRETRAINED MODEL</p>	<p>ResNet PRETRAINED MODEL</p>
<p>Caffe MODELS</p>	<p>GoogLeNet PRETRAINED MODEL</p>	<p>TensorFlow/Keras MODELS</p>



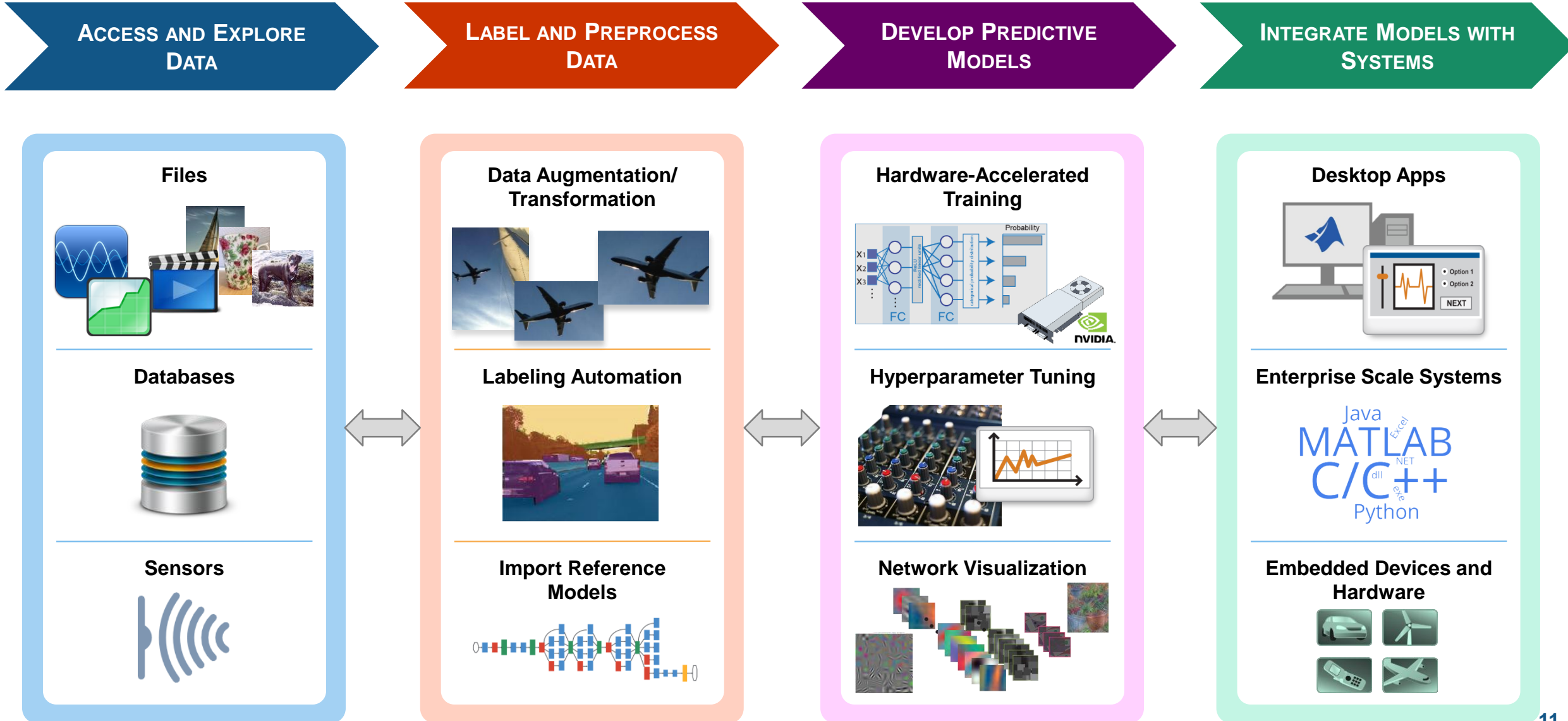
Machine Learning vs Deep Learning

Deep learning performs **end-to-end learning** by learning **features, representations and tasks** directly from **images, text and sound**

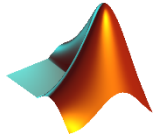
Deep learning algorithms also **scale with data** – traditional machine learning **saturates**



Deep Learning Workflow



Agenda



Why deep learning?

MNIST: The “Hello, World!” of computer vision

Transfer learning with CNNs

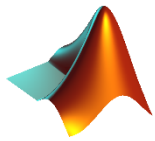
Semantic segmentation

Ground truth labeling of datasets

Everything else in deep learning...

Agenda

Why deep learning?



MNIST: The “Hello, World!” of computer vision

Transfer learning with CNNs

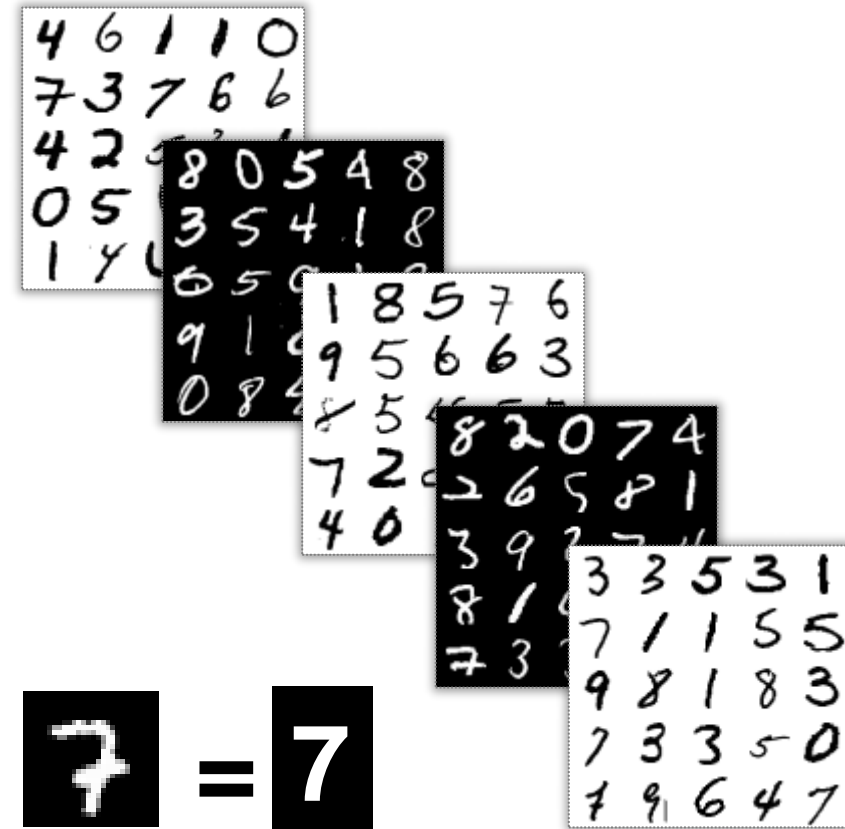
Semantic segmentation

Ground truth labeling of datasets

Everything else in deep learning...

MNIST: The “Hello, World!” of computer vision

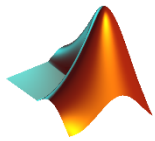
What?	A set of handwritten digits from 0-9
Why?	An easy task for machine learning beginners
How many?	60,000 training images 10,000 test images
Best results?	99.79% accuracy



Sources: <http://yann.lecun.com/exdb/mnist/>
<https://rodrigob.github.io/are-we-there-yet/build/classification-datasets-results>

Agenda

Why deep learning?



MNIST: The “Hello, World!” of computer vision

Transfer learning with CNNs

Semantic segmentation

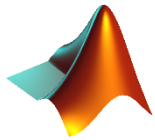
Ground truth labeling of datasets

Everything else in deep learning...

Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision



Transfer learning with CNNs

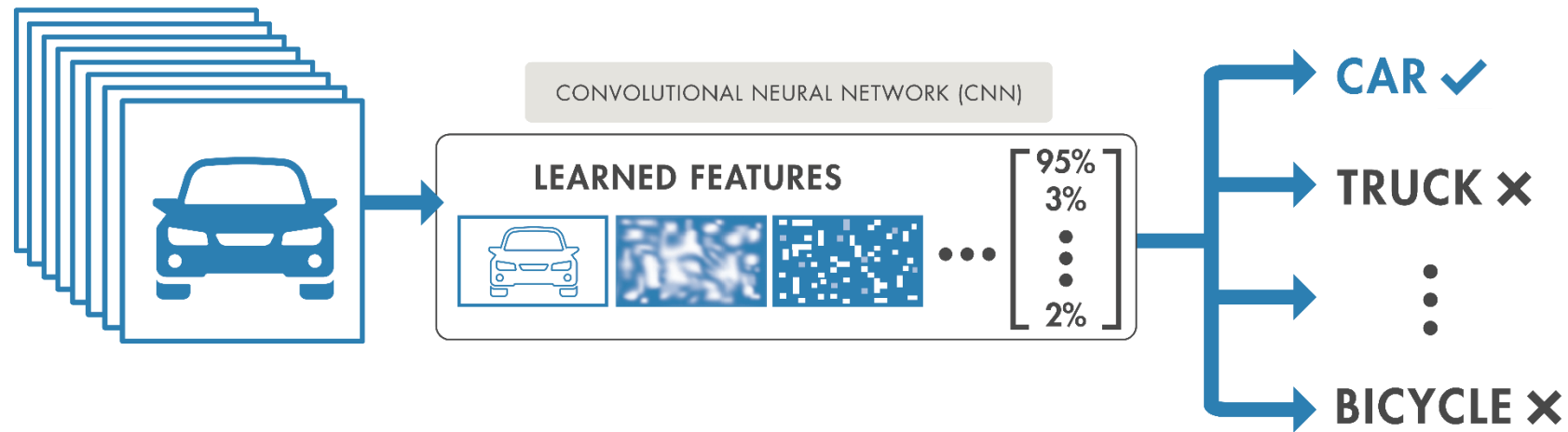
Semantic segmentation

Ground truth labeling of datasets

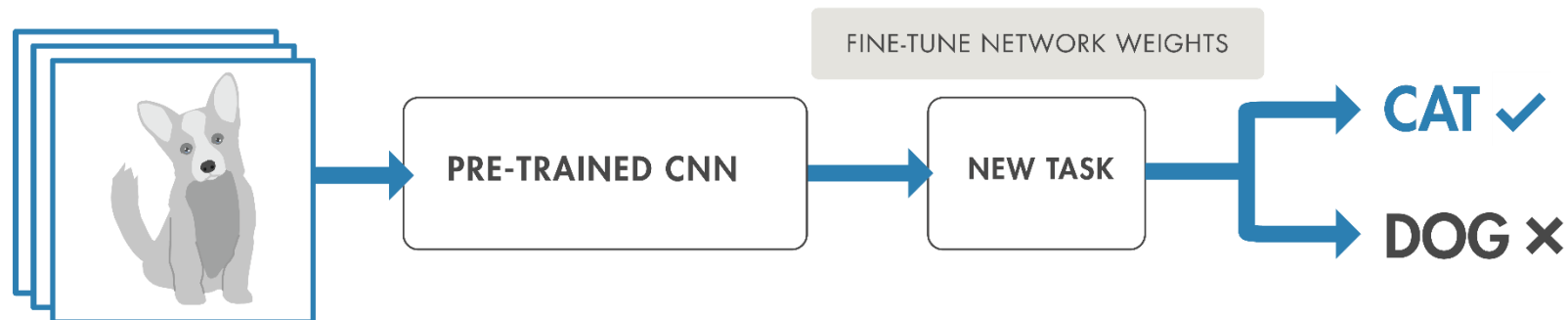
Everything else in deep learning...

Two Approaches for Deep Learning

1. Train a Deep Neural Network from Scratch



2. Fine-tune a pre-trained model (transfer learning)

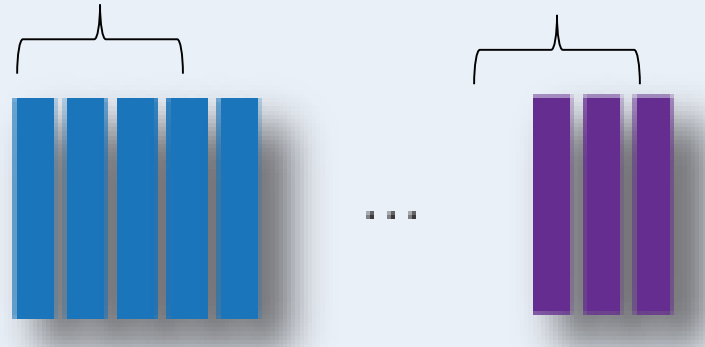


Transfer Learning Workflow

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

Load pretrained network

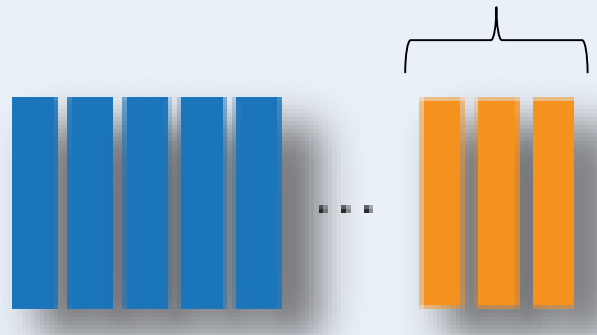
Early layers that learned low-level features (edges, blobs, colors) Last layers that learned task specific features



1 million images
1000s classes

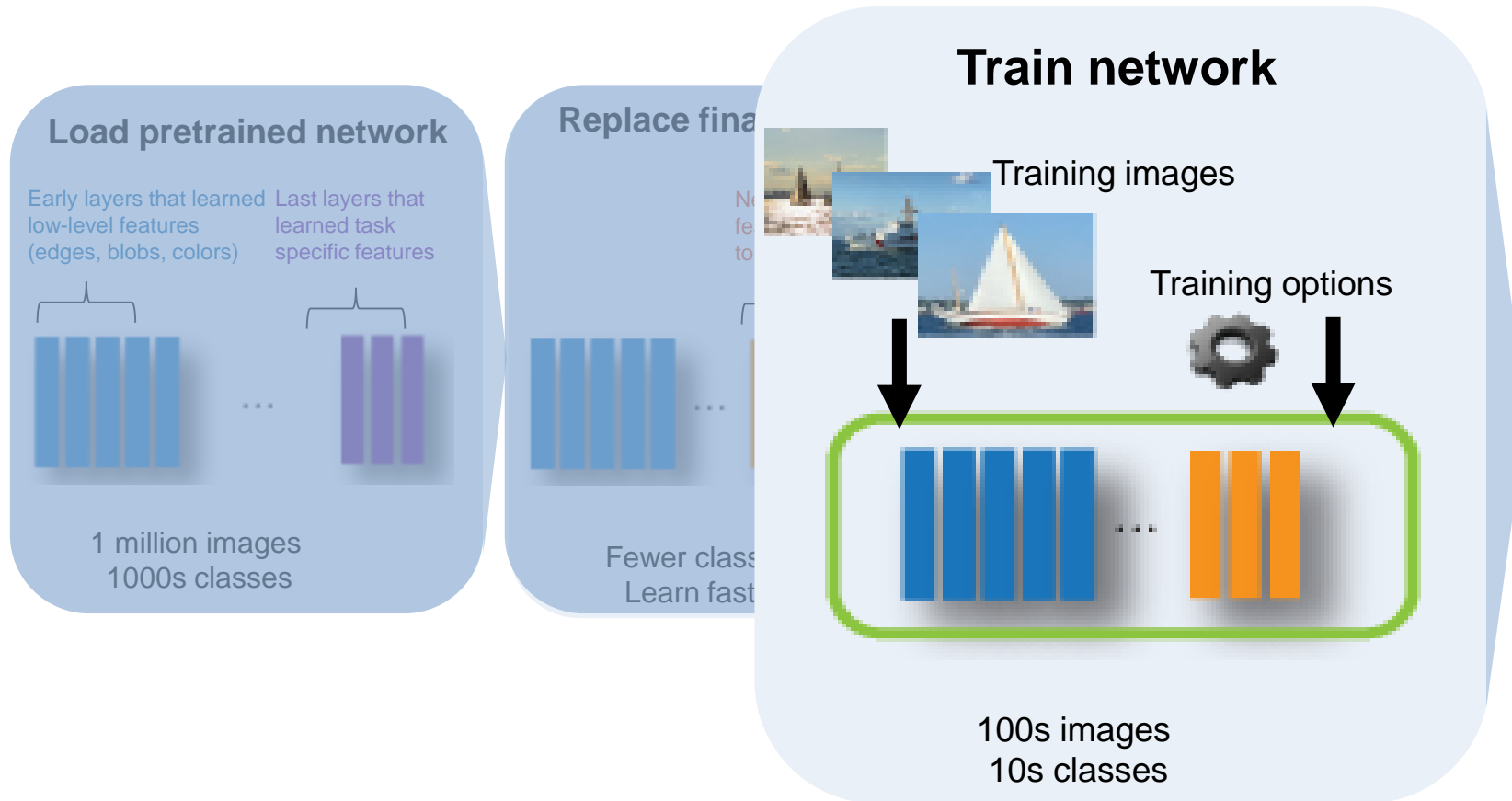
Replace final layers

New layers learn features specific to your data

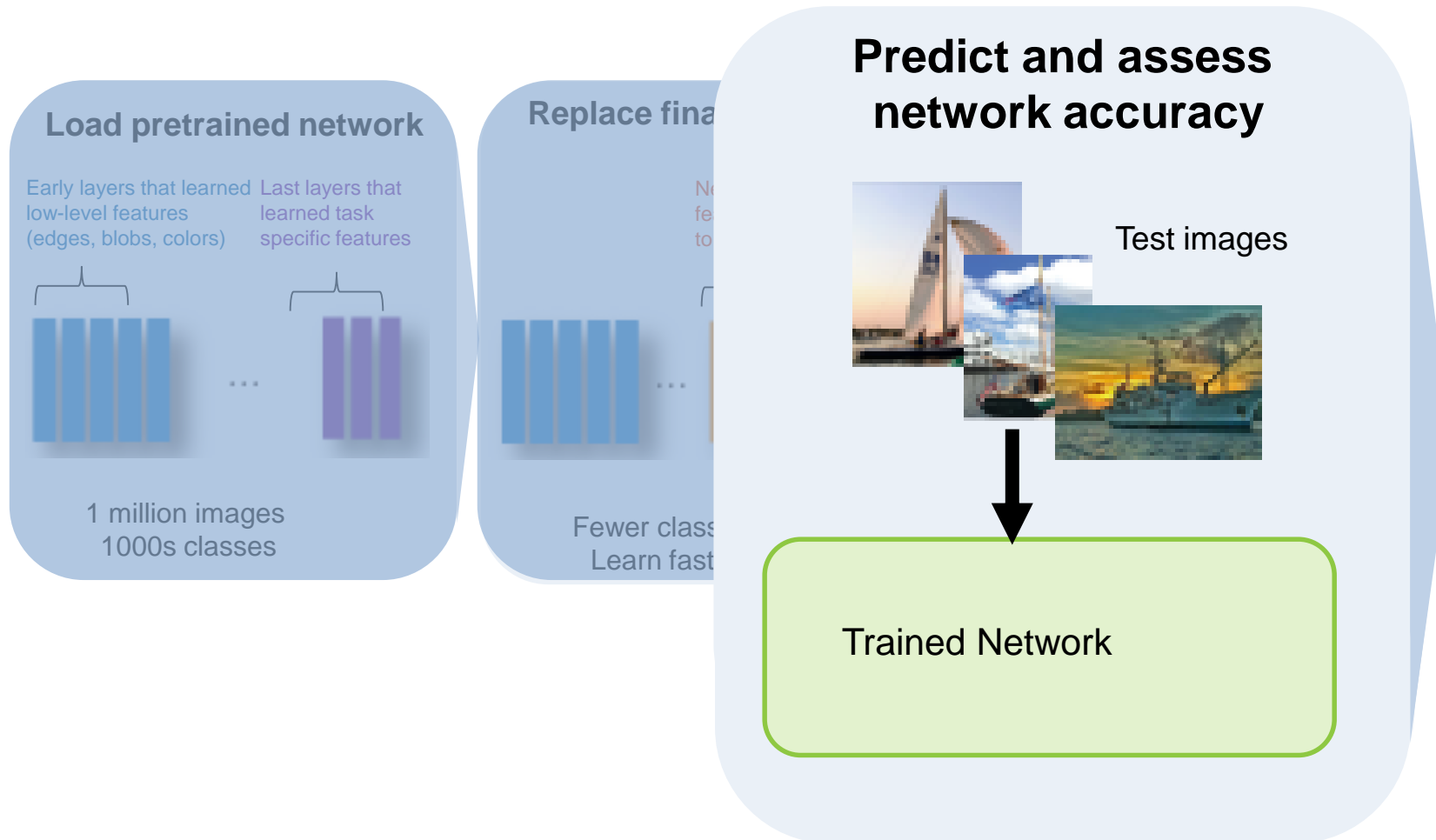


Fewer classes
Learn faster

Transfer Learning Workflow



Transfer Learning Workflow



Transfer Learning Workflow

Load pretrained network

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



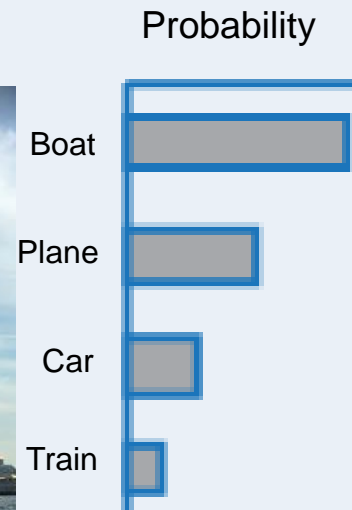
1 million images
1000s classes

Replace final

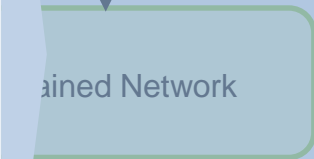
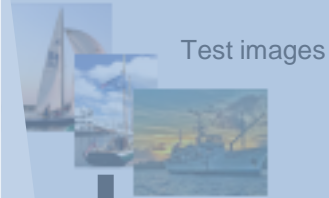


Fewer classes
Learn fast

Deploy results



Predict and assess network accuracy



Transfer Learning Workflow

Load pretrained network

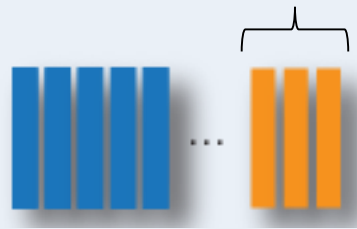
Early layers that learned low-level features (edges, blobs, colors) Last layers that learned task specific features



1 million images
1000s classes

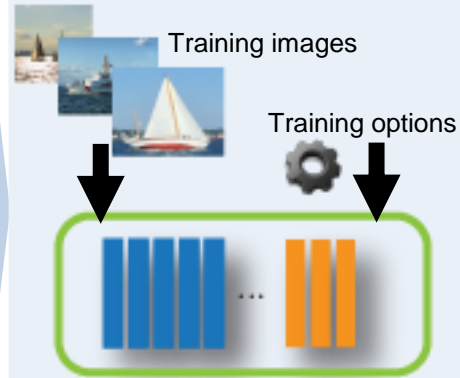
Replace final layers

New layers to learn features specific to your data



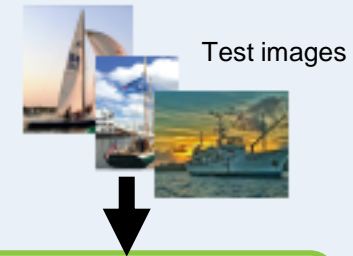
Fewer classes
Learn faster

Train network



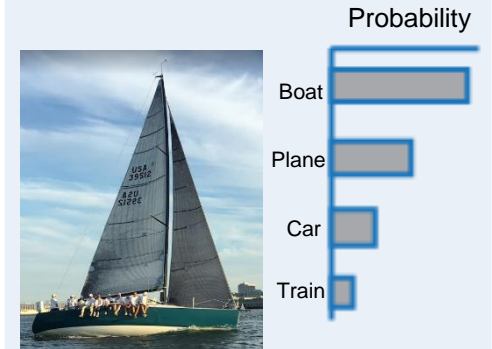
100s images
10s classes

Predict and assess network accuracy



Trained Network

Deploy results



Example: Food classifier using deep transfer learning



Caesar salad →
French fries →
Burgers →
Pizza →
Sushi →

**5 Category
Classifier**

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

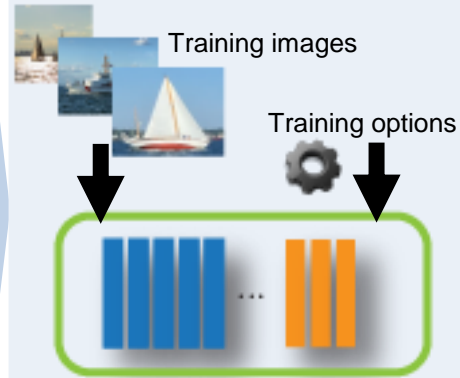
Replace final layers

New layers to learn features specific to your data



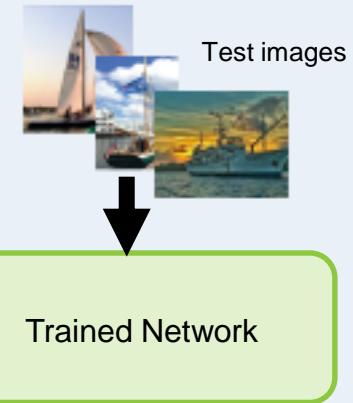
Fewer classes
Learn faster

Train network

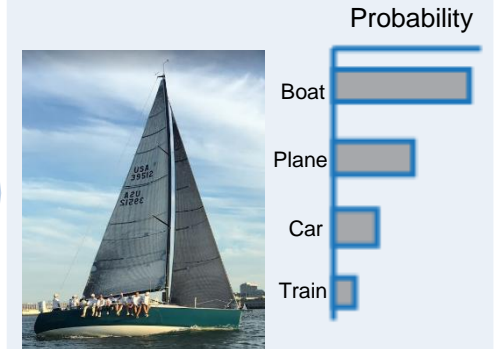


100s images
10s classes

Predict and assess network accuracy

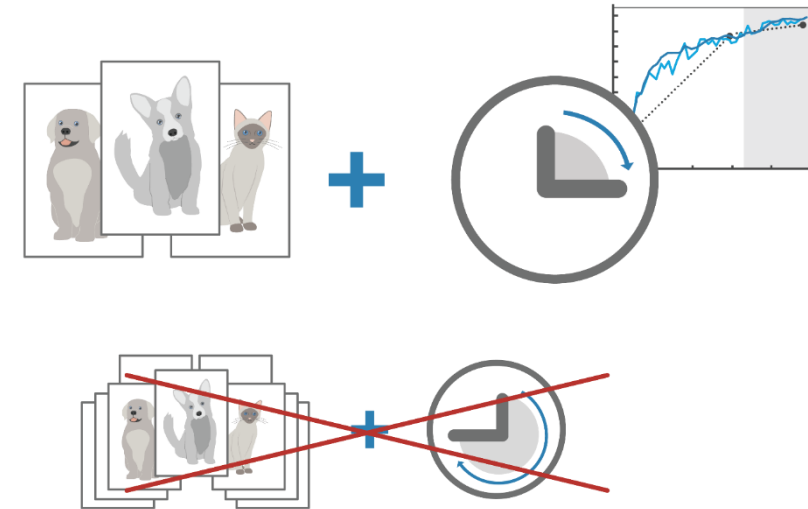


Deploy results



Why Perform Transfer Learning

- Requires less data and training time
- Reference models (like AlexNet, VGG-16, VGG-19) are great feature extractors
- Leverage best network types from top researchers



AlexNet

PRETRAINED MODEL

Caffe

MODELS

VGG-16

PRETRAINED MODEL

GoogLeNet

PRETRAINED MODEL

ResNet

PRETRAINED MODEL

TensorFlow/Keras

MODELS

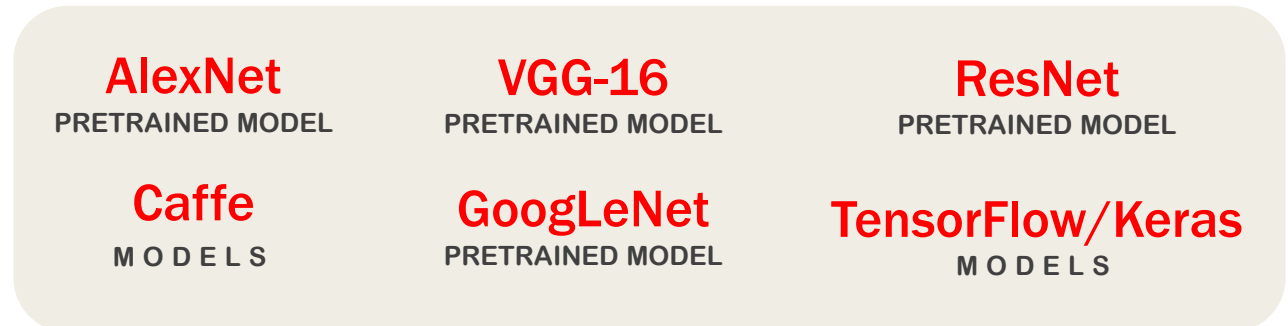
Import the Latest Models for Transfer Learning

Pretrained Models*

- AlexNet
- VGG-16
- VGG-19
- GoogLeNet
- InceptionV3 (coming soon)
- Resnet50 (coming soon)

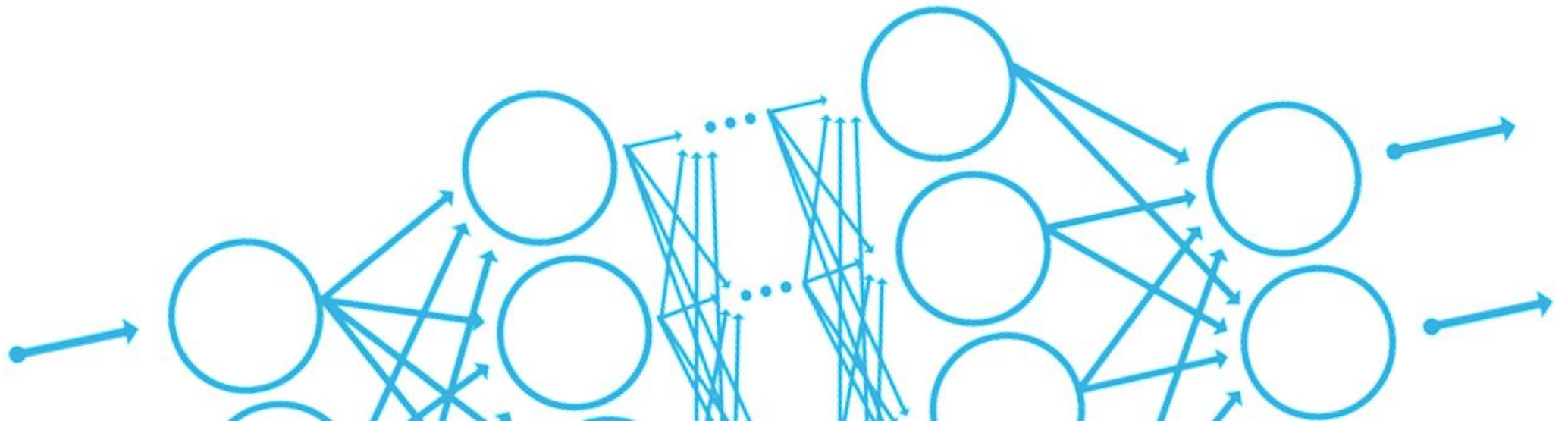
Import Models from Frameworks

- Caffe Model Importer
- TensorFlow/Keras Model Importer (coming soon)



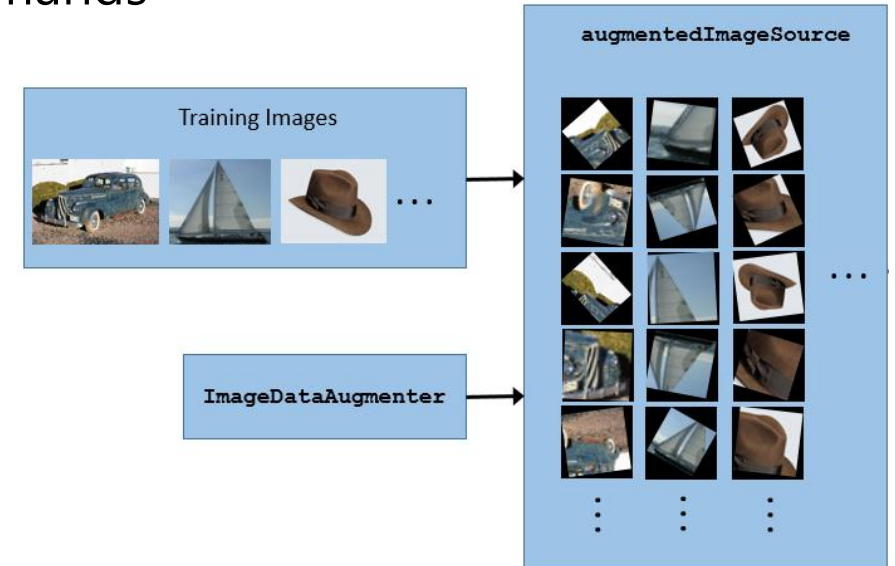
* single line of code to access model

A short recap...



What we covered in this section...

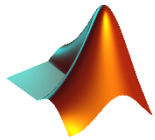
- **Transfer learning**
 - Easily modify existing networks with one-line commands
- **Access large datasets**
 - Using the `imageDatastore`
- **Improve training results**
 - Data augmentation with `imageDataAugmenter`
 - Automatic parameter selection with **Bayesian hyperparameter tuning**



Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision



Transfer learning with CNNs

Semantic segmentation

Ground truth labeling of datasets

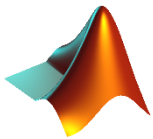
Everything else in deep learning...

Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision

Transfer learning with CNNs

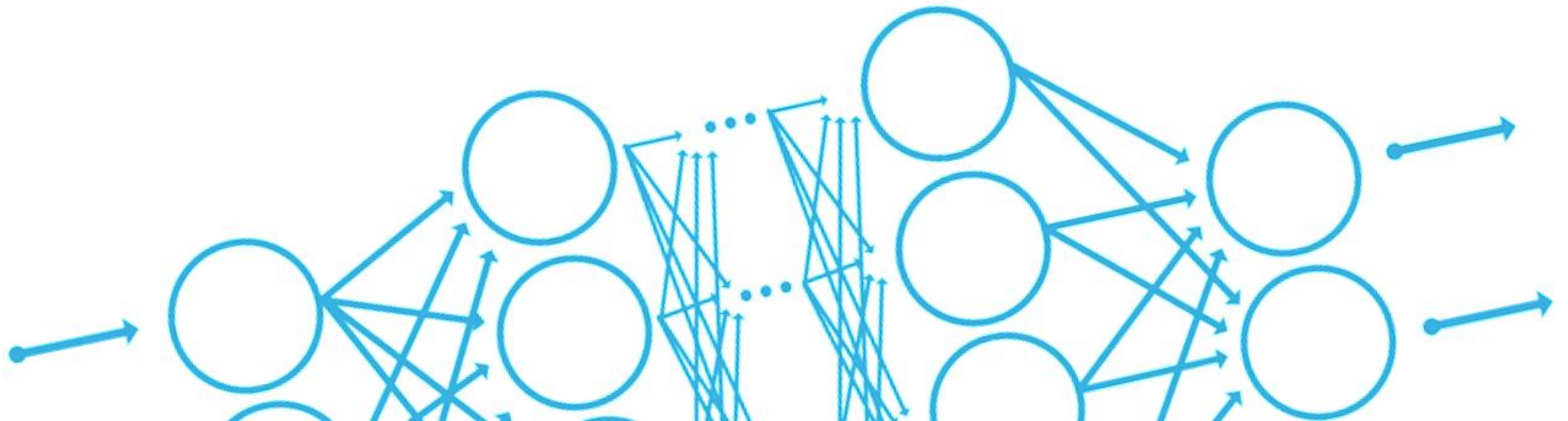


Semantic segmentation

Ground truth labeling of datasets

Everything else in deep learning...

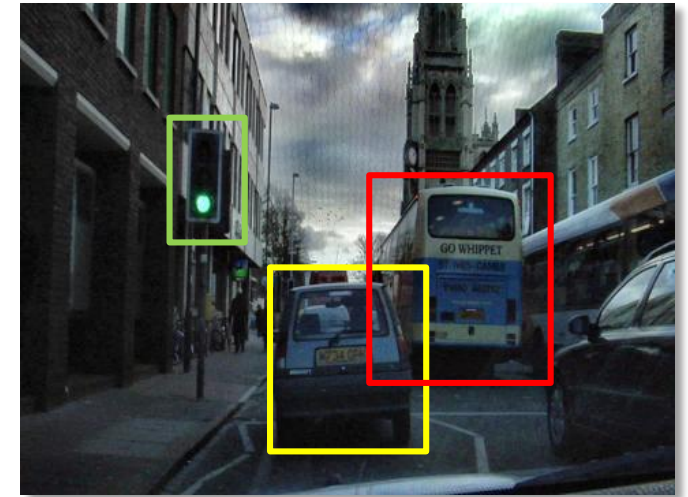
What is semantic segmentation?



Original Image



ROI detection



Pixel classification



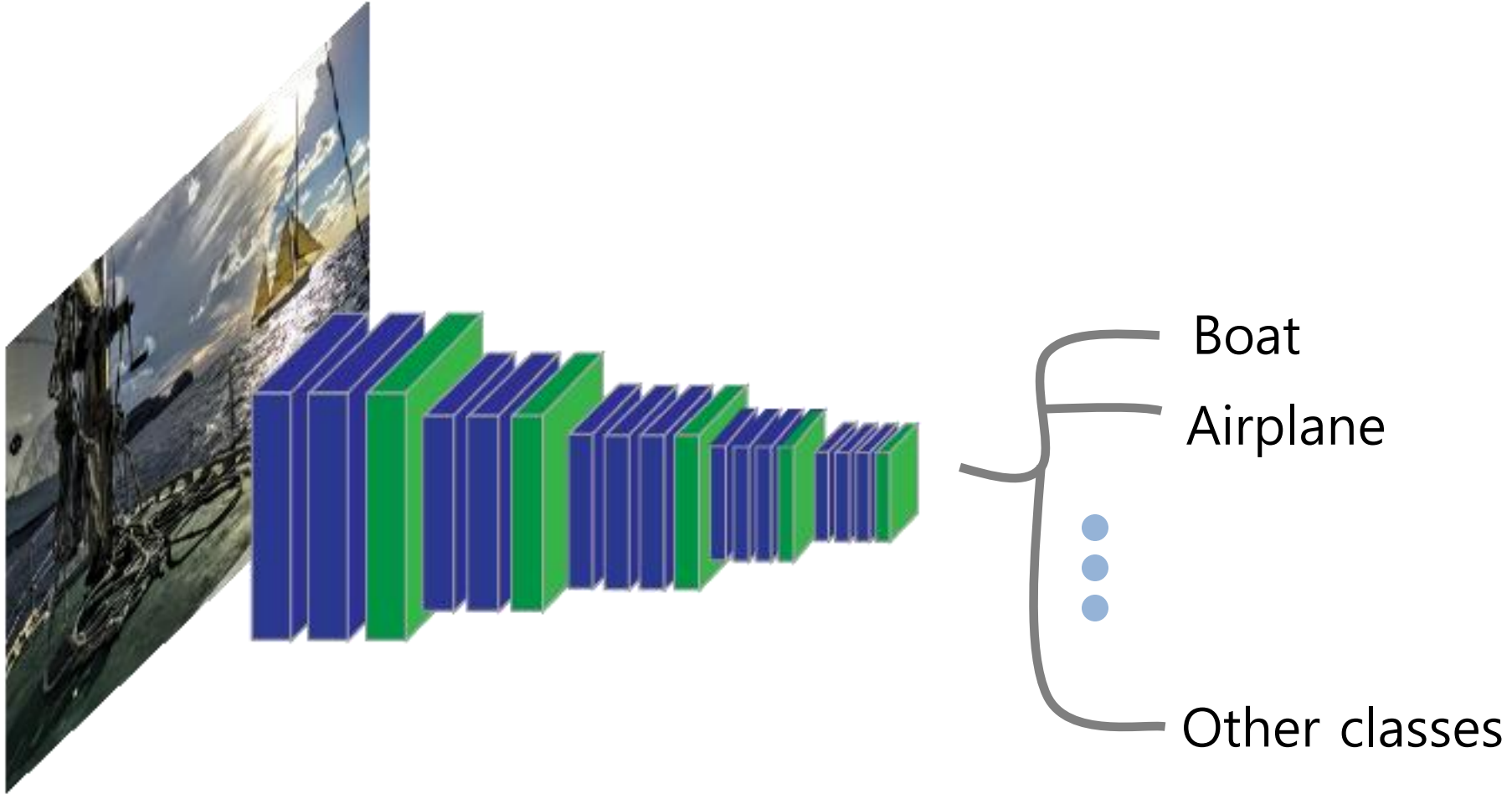
Semantic Segmentation



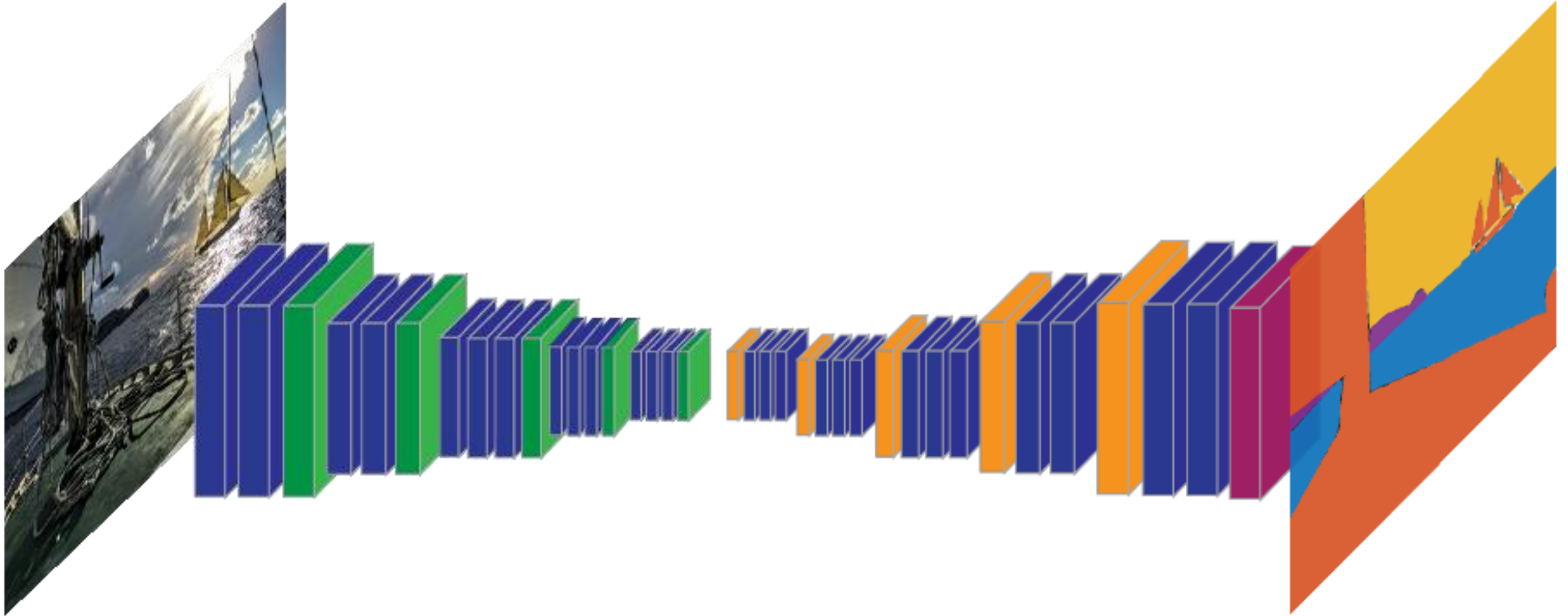
CamVid Dataset

1. Segmentation and Recognition Using Structure from Motion Point Clouds, ECCV 2008
2. Semantic Object Classes in Video: A High-Definition Ground Truth Database ,Pattern Recognition Letters

Semantic Segmentation Network



Semantic Segmentation Network

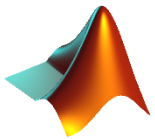


Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision

Transfer learning with CNNs



Semantic segmentation

Ground truth labeling of datasets

Everything else in deep learning...

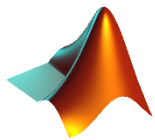
Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision

Transfer learning with CNNs

Semantic segmentation



Ground truth labeling of datasets

Everything else in deep learning...

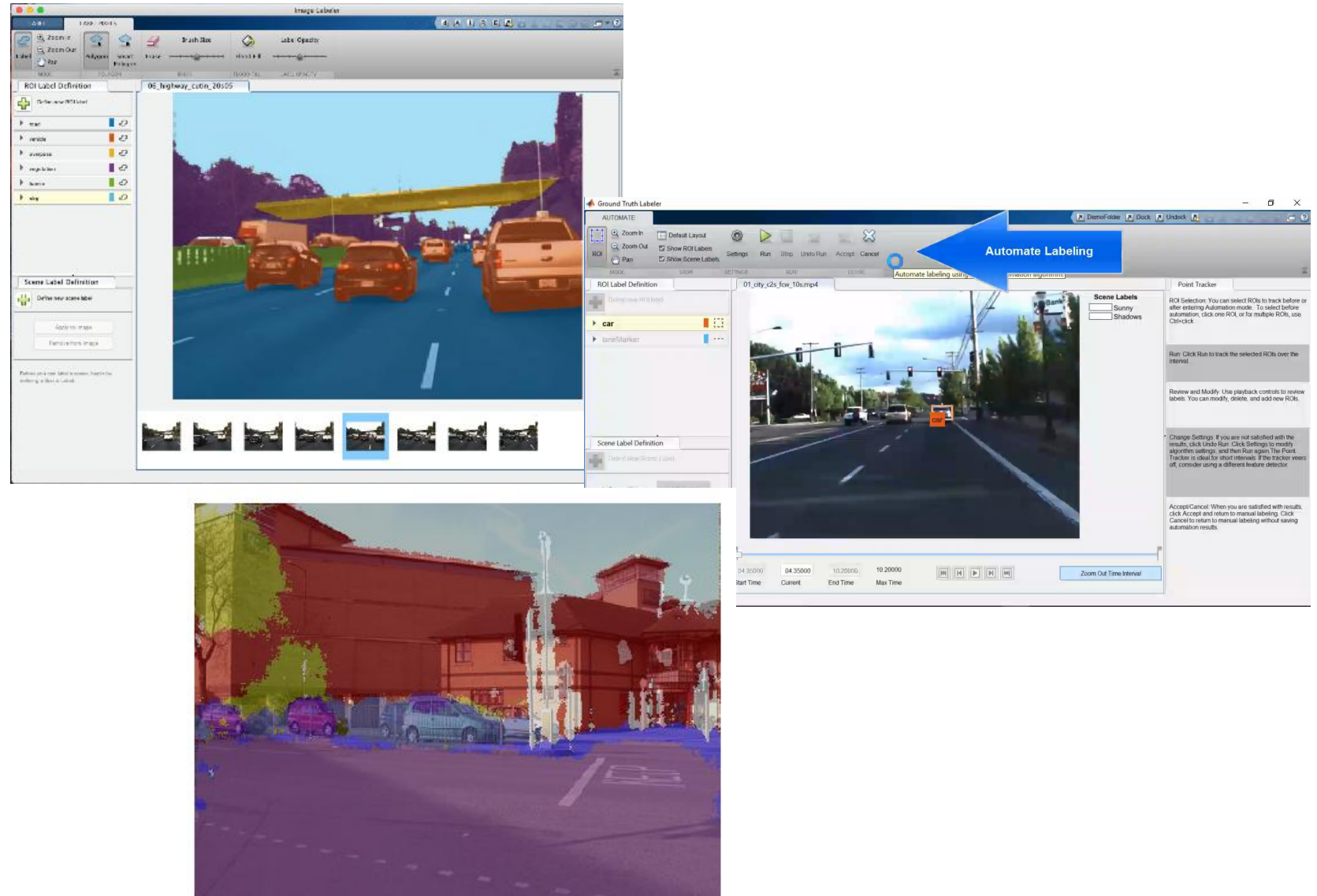
Ground truth Labeling

“How do I *label* my data?”

New App for Ground Truth Labeling

Label pixels and regions for semantic segmentation

Data



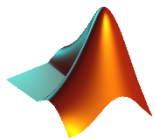
Agenda

Why deep learning?

MNIST: The "Hello, World!" of computer vision

Transfer learning with CNNs

Semantic segmentation



Ground truth labeling of datasets

Everything else in deep learning...

Agenda

Why deep learning?

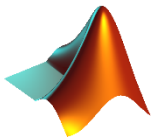
MNIST: The "Hello, World!" of computer vision

Transfer learning with CNNs

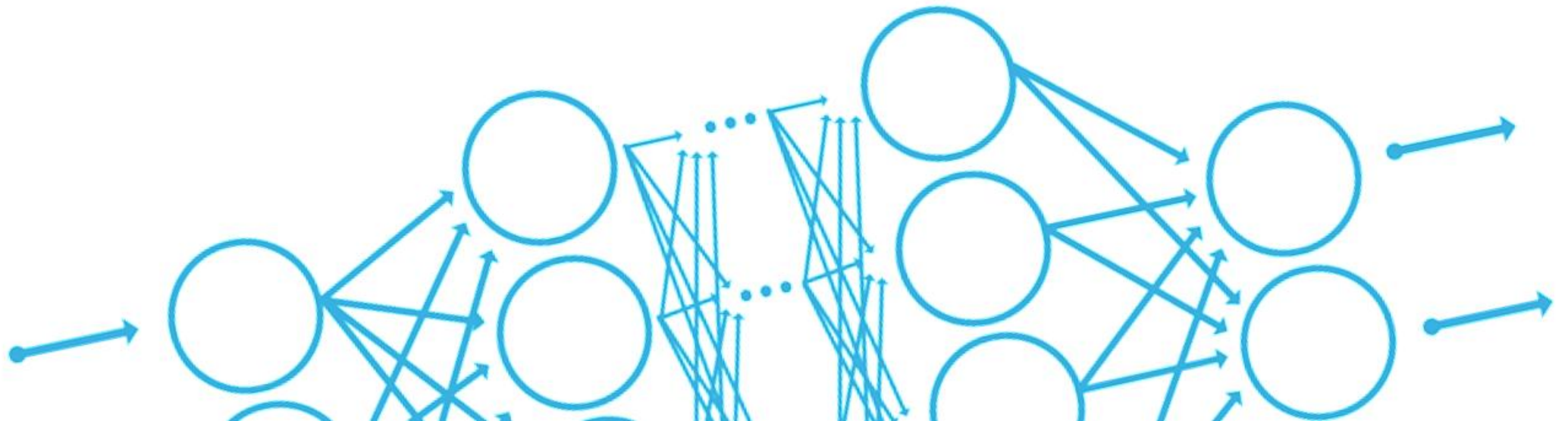
Semantic segmentation

Ground truth labeling of datasets

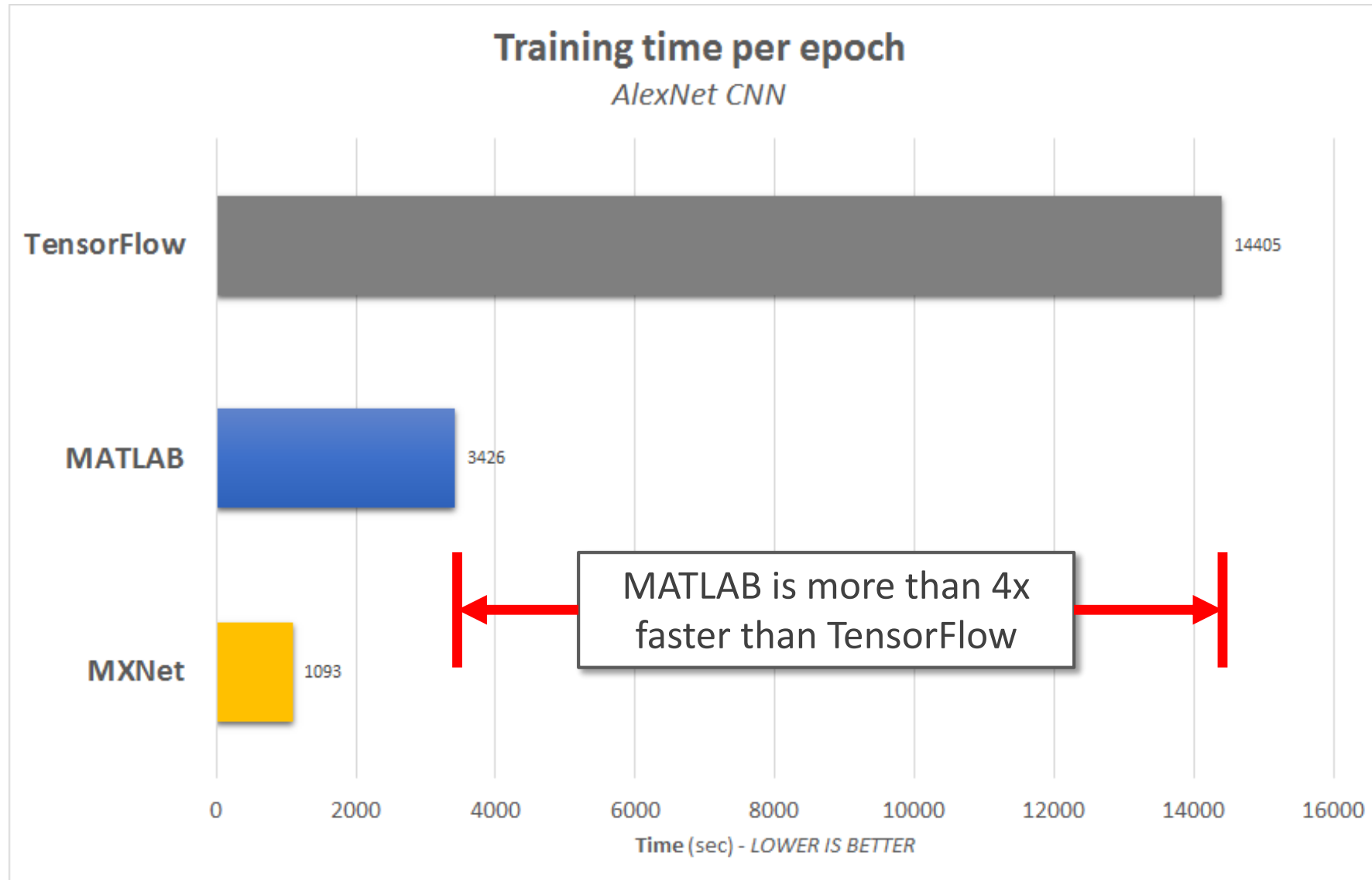
Everything else in deep learning...



Training Performance and Scalability

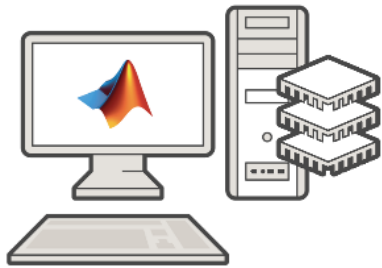


Training in MATLAB is fast

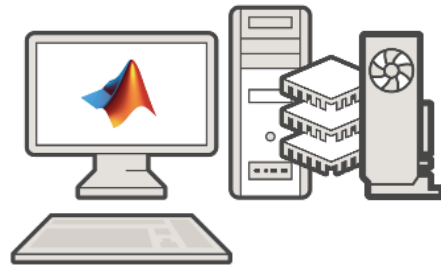


AlexNet CNN architecture trained on the ImageNet dataset, using batch size of 32, on a Windows 10 desktop with single NVIDIA GPU

Deep learning on CPU, GPU, multi-GPU and clusters



Single CPU



Single CPU
Single GPU

HOW TO TARGET?

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

Deep learning on CPU, GPU, multi-GPU and clusters

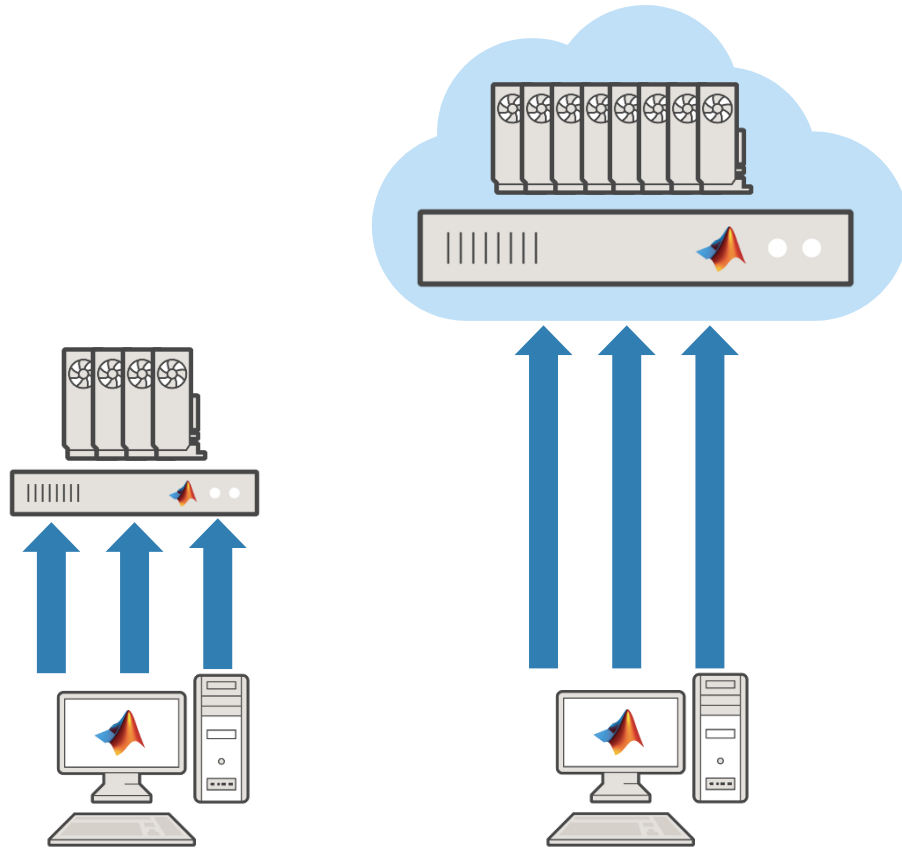


Single CPU
Multiple GPUs

HOW TO TARGET?

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

Deep learning on CPU, GPU, multi-GPU and clusters



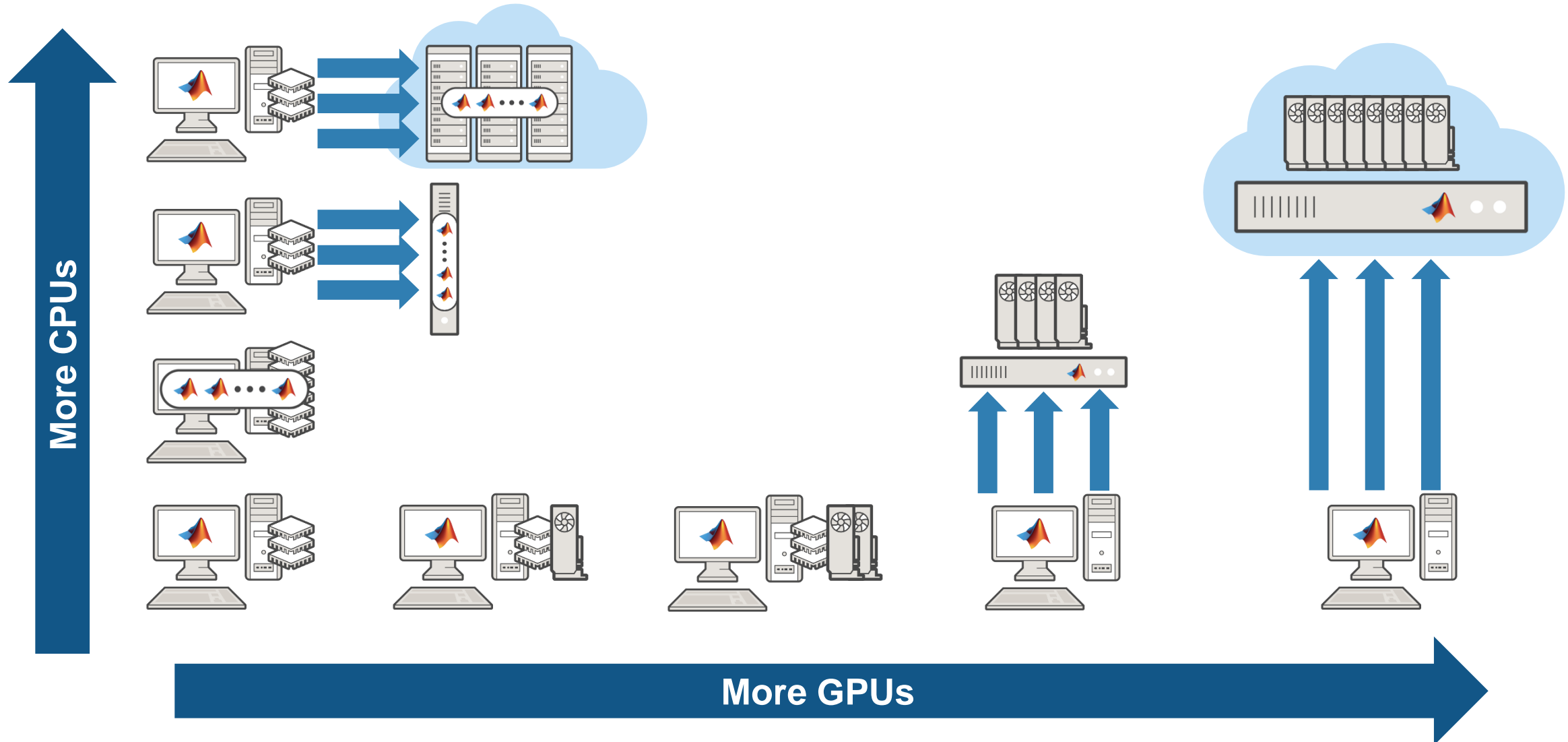
On-prem server with
GPUs

Cloud GPUs
(AWS, Azure, etc.)

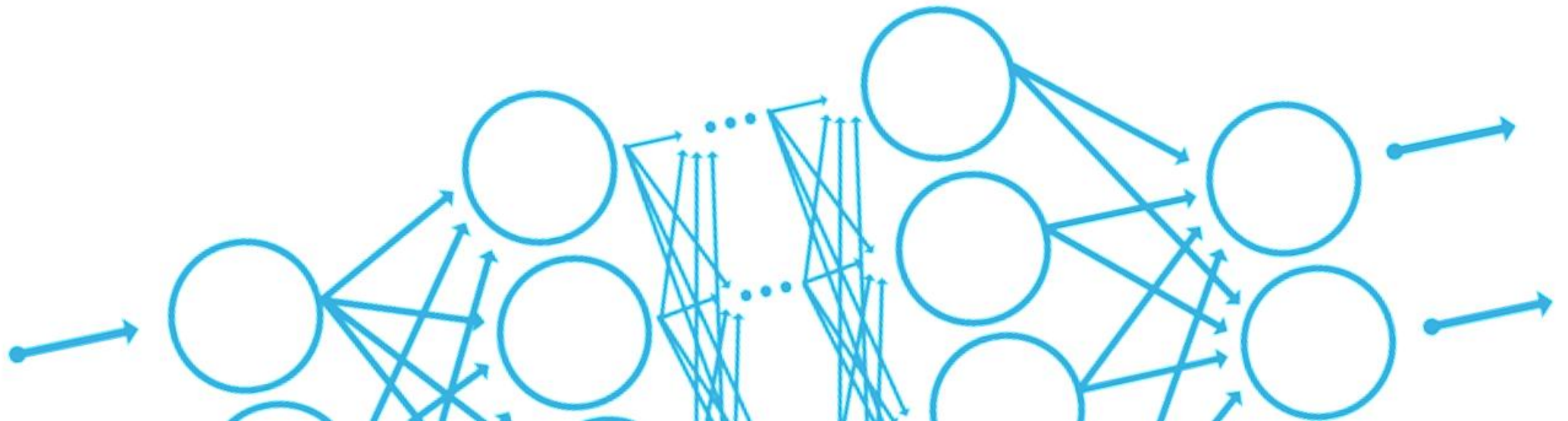
HOW TO TARGET?

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

Deep learning on CPU, GPU, multi-GPU and clusters

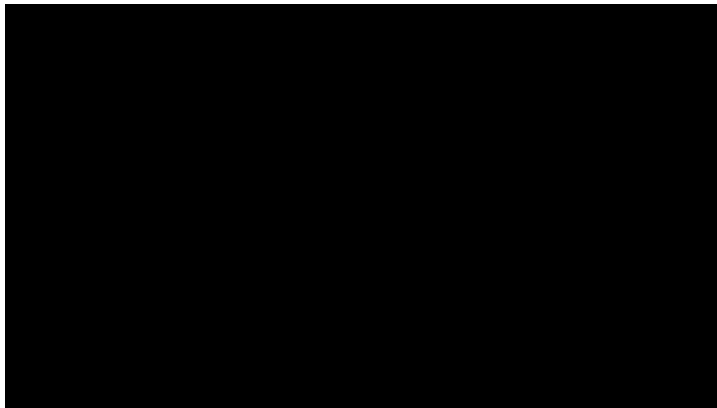


Inference Performance and Deployment

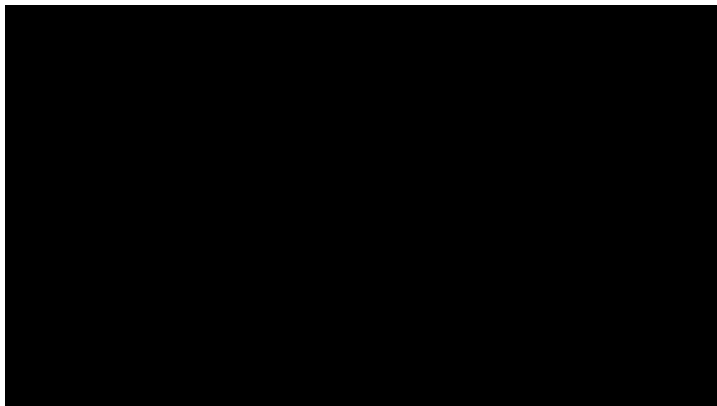


Running a Trained Model

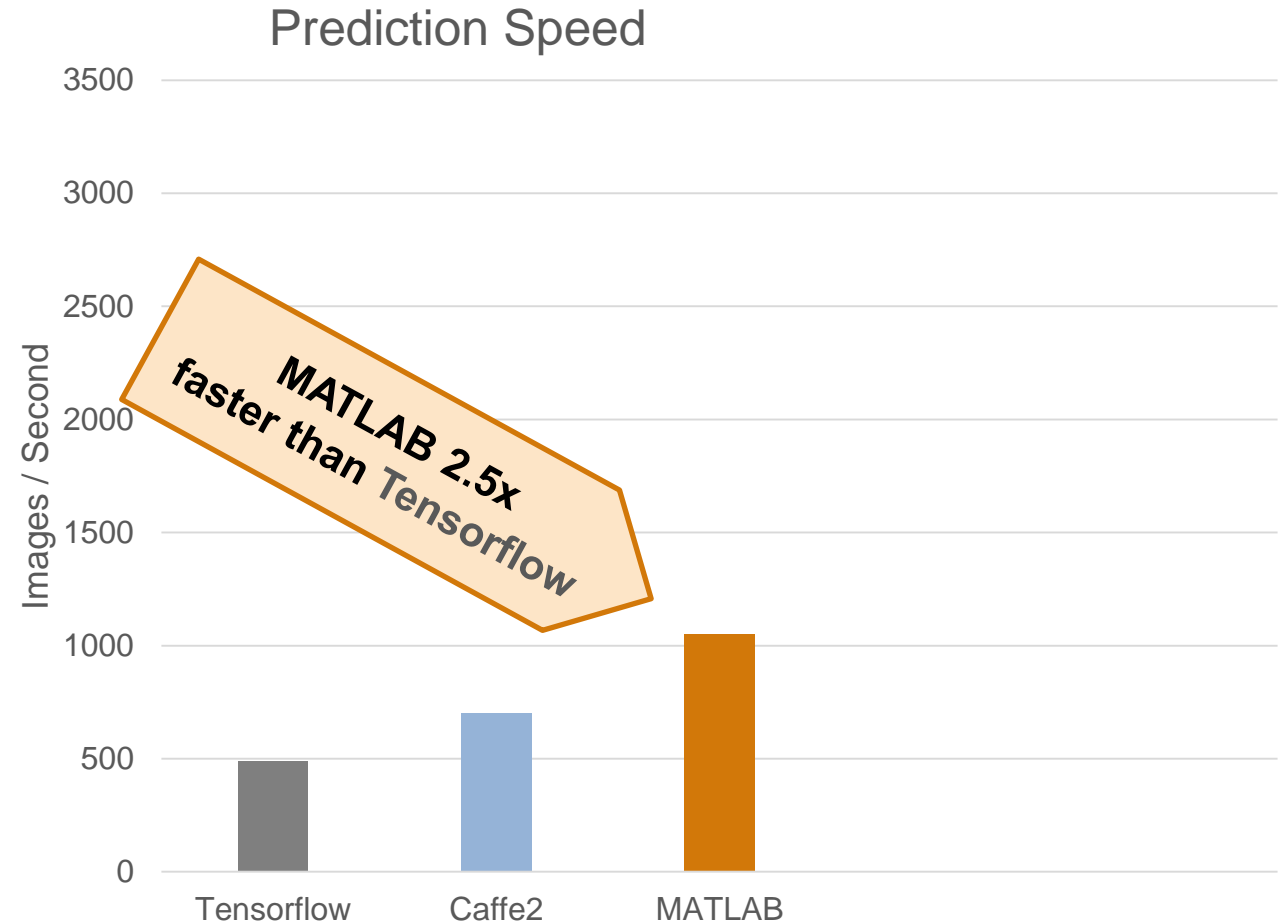
“How do I make my model run faster?”



30 fps



10 fps



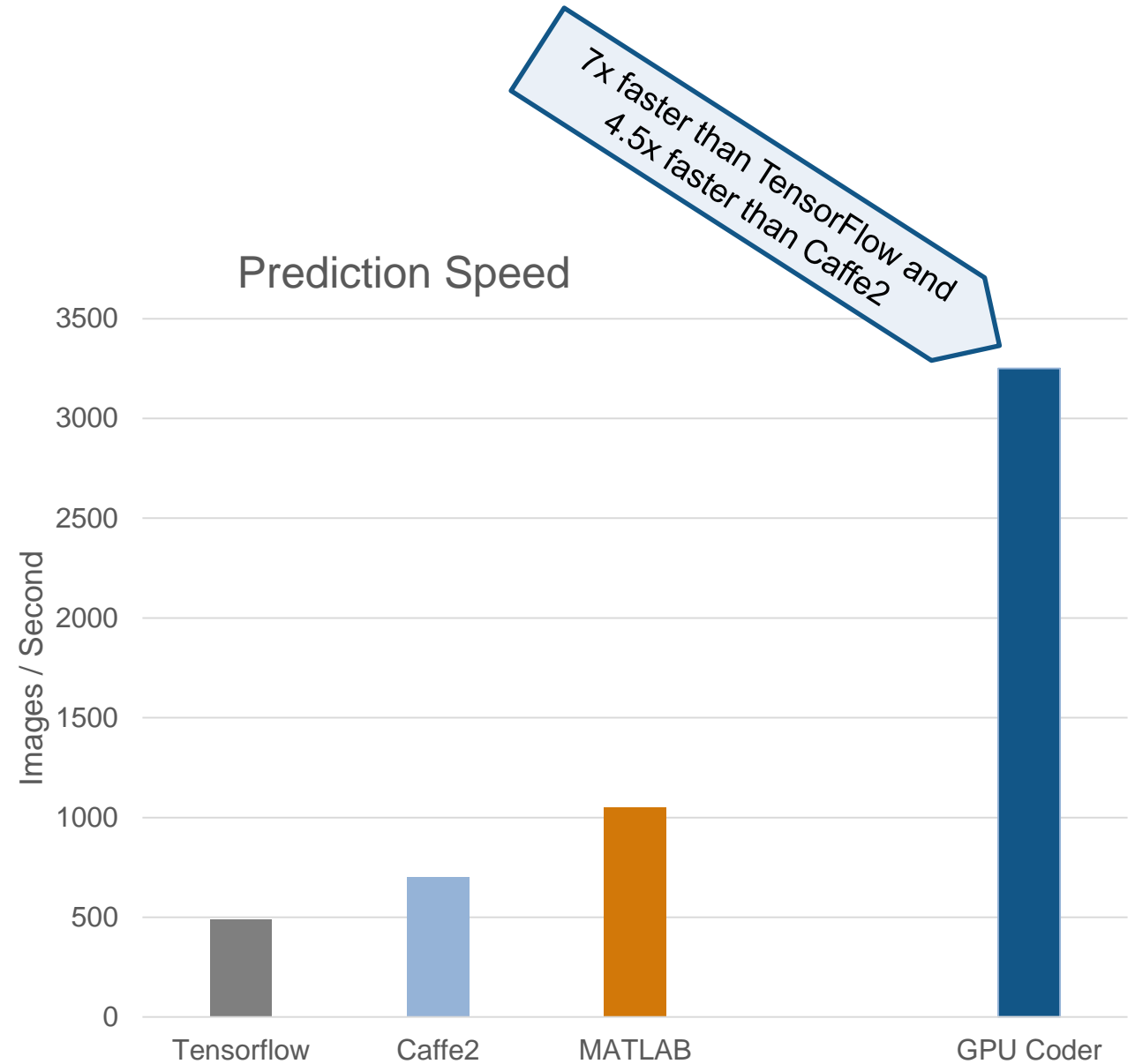
* Internal benchmarks were performed for inference performance of AlexNet using a TitanXP GPU and Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz. Software versions used were MATLAB(R2017b), TensorFlow(1.2.0), and Caffe2(0.8.1). The GPU accelerated versions of each software were used for benchmarks. All tests were run on Windows 10.

Running a Trained Model

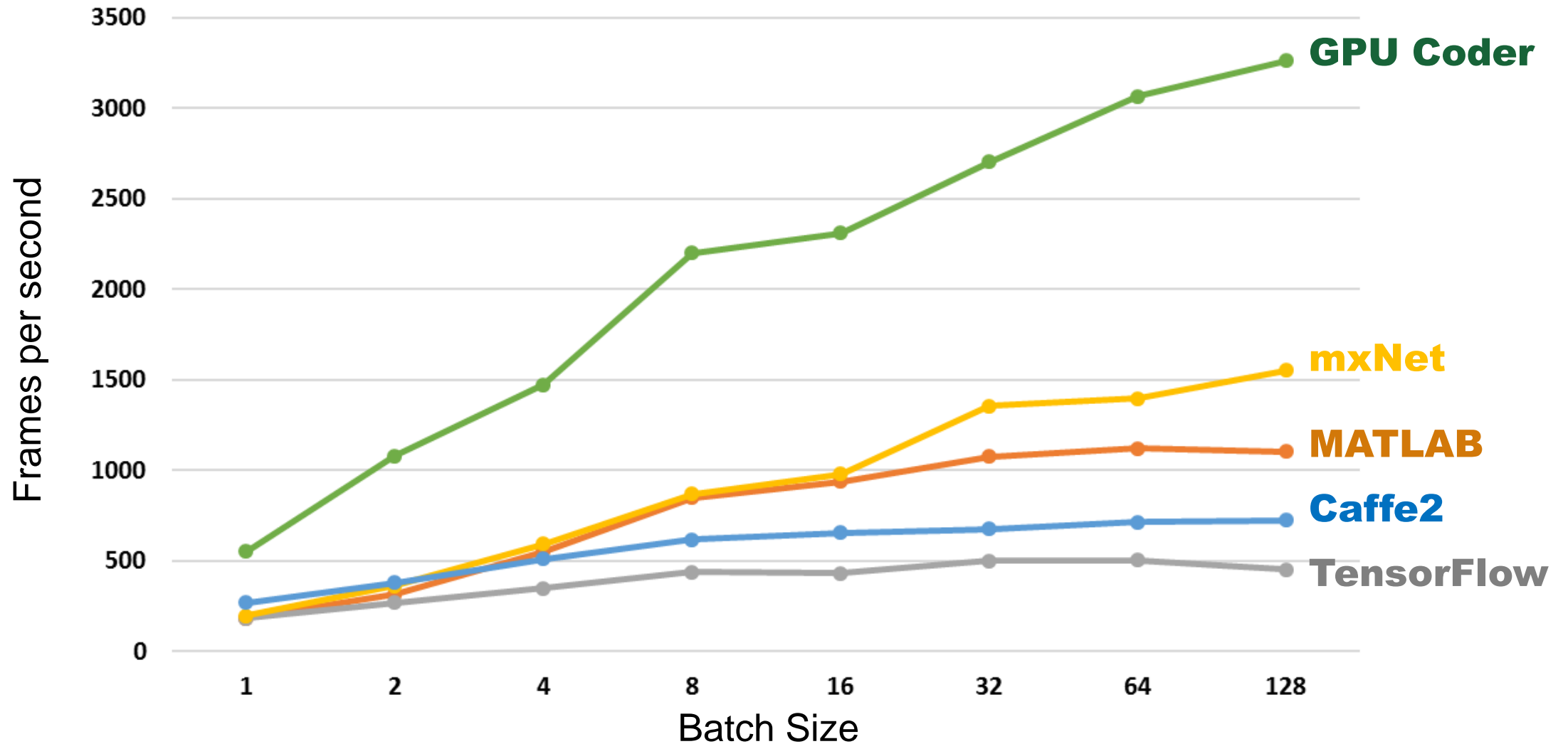
R2017b

Introducing:

**GPU Coder-
Convert to
NVIDIA CUDA
code**

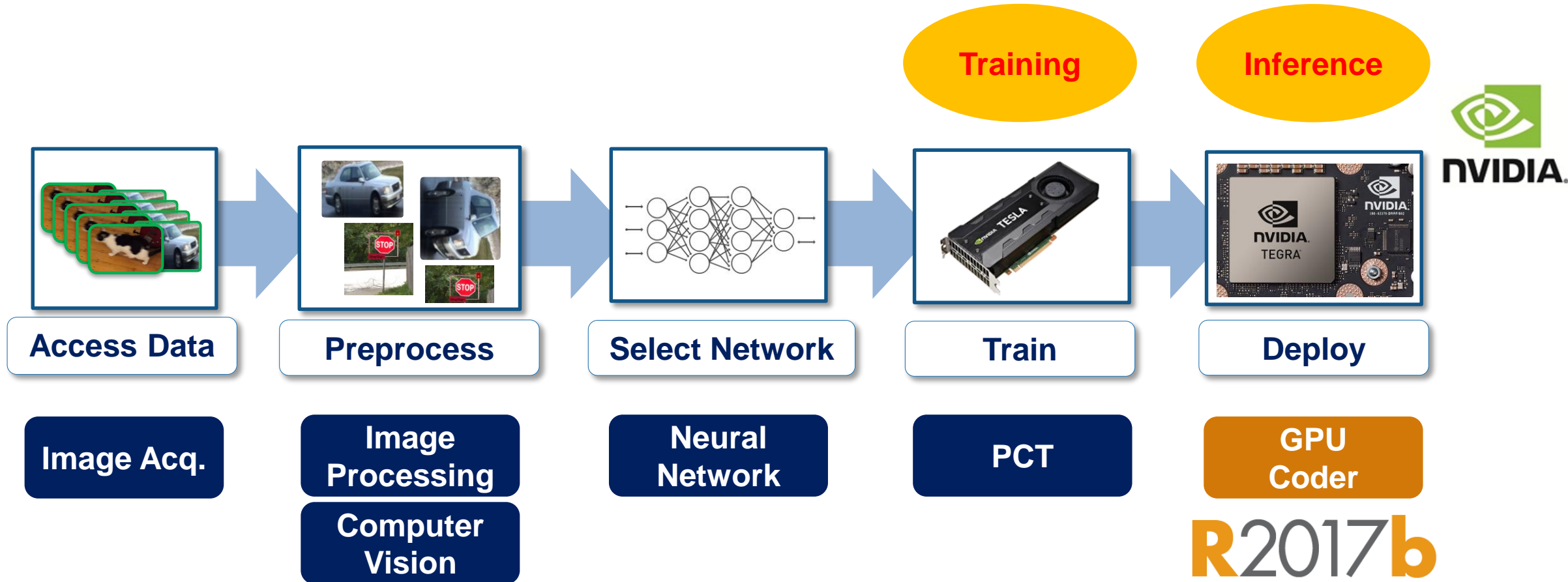


AlexNet Inference on NVIDIA Titan XP GPU



Testing platform	CPU	Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz
	GPU	Pascal TitanXP

GPU Coder Fills a Gap in Our Deep Learning Solution

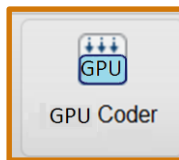
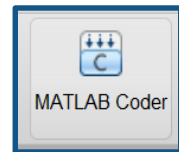
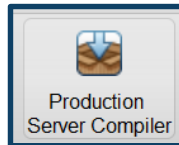
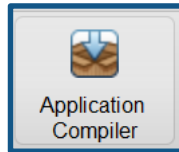


Overview of deep learning deployment options

“How do I *deploy* my model?”

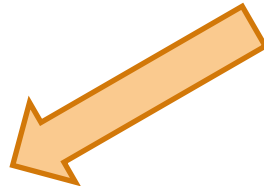
Introducing:
GPU Coder-
 Convert to
NVIDIA CUDA
 code

Deploy / Share



- Create Desktop Apps
- Run Enterprise Solution
- Generate C and C++ Code

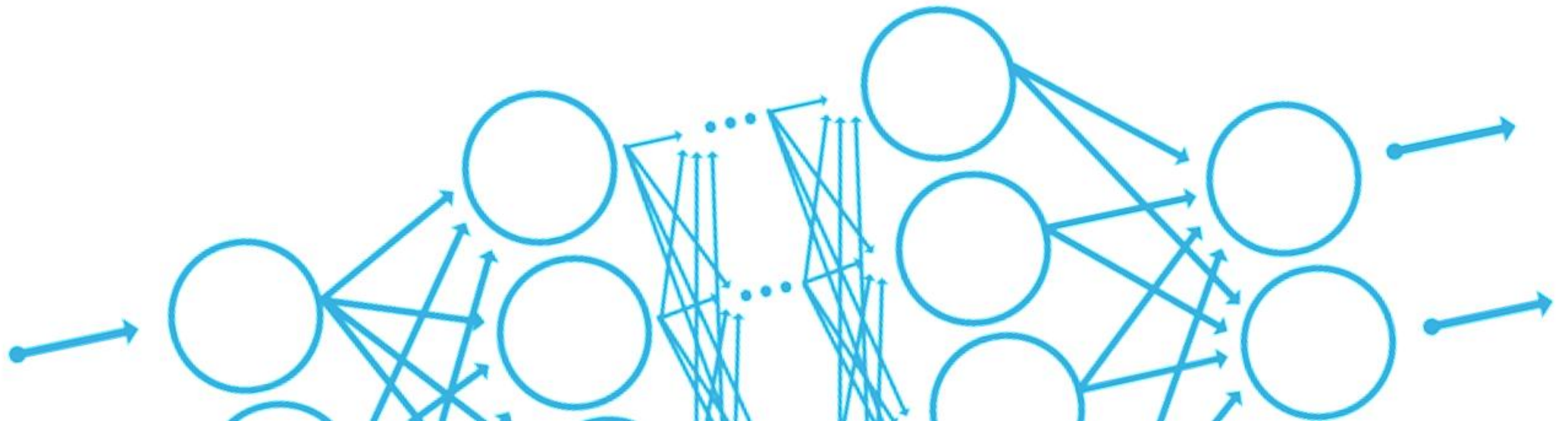
- Target GPUs



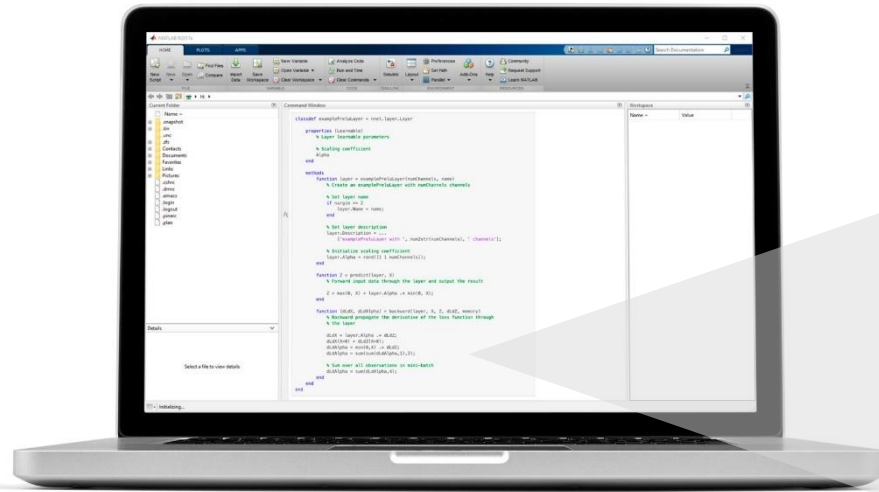
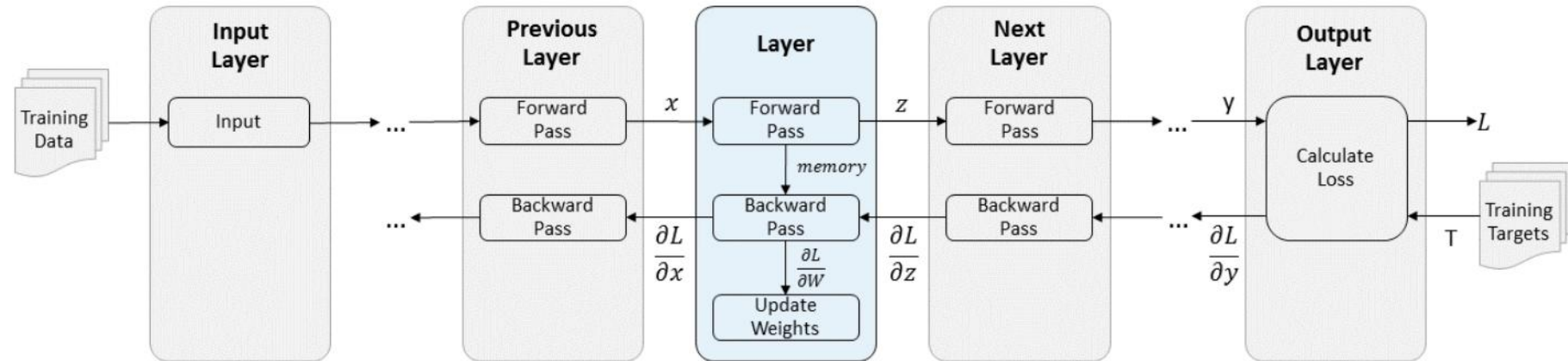
GPU Coder



Other Features



Define new deep neural network layers **R2017b**



```
function [dLdX, dLdAlpha] = backward(layer, X, Z, dLdZ, memory)
    % Backward propagate the derivative of the loss function through
    % the layer
```

```
dLdX = layer.Alpha .* dLdZ;
dLdX(X>0) = dLdZ(X>0);
dLdAlpha = min(0,X) .* dLdZ;
dLdAlpha = sum(sum(dLdAlpha,1),2);
```

Object Detection Frameworks

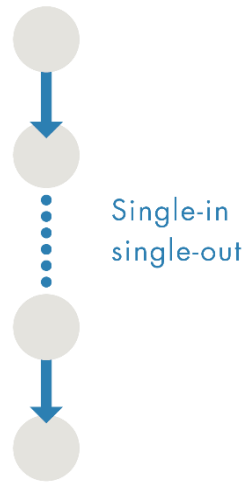
R2017a

- Single line of code to train a detector
- Includes:
 - R-CNN
 - Fast R-CNN
 - Faster R-CNN

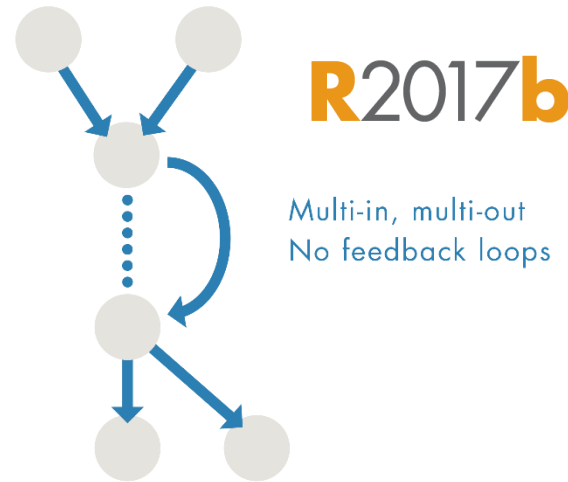


Other deep neural networks

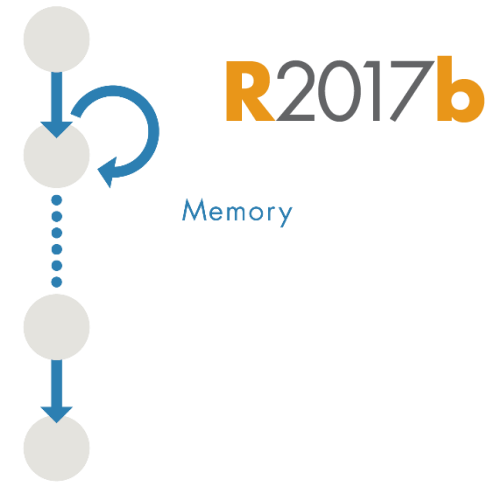
SeriesNetwork



DAGNetwork



Recurrent Network



Networks: MNIST
Alexnet
VGG
Lane detection
Pedestrian detection

Networks: R-CNN (fast, faster)
GoogLeNet
ResNet
SegNet
FCN
DeconvNet

} *Object detection*
} *Semantic segmentation*

Networks: LSTM

Deep learning features overview

R2017b

- Classification
- Regression *
- Semantic segmentation
- Object detection *
- Scalability *
 - Multiple GPUs
 - Cluster or cloud
- Custom network layers *
- Import models *
 - Caffe
 - Keras/TensorFlow
- Data augmentation *
- Hyperparameter tuning *
 - Bayesian optimization
- Python ↔ MATLAB interface *
- LSTM networks *
 - Time series, signals, audio
- Custom labeling *
 - API for ground-truth labeling automation
 - Superpixels
- Data validation *
 - Training and testing

MATLAB products for deep learning

Required products

- Neural Network Toolbox
- Parallel Computing Toolbox
- Image Processing Toolbox
- Computer Vision System Toolbox

Recommended products

- Statistics and Machine Learning Toolbox
- MATLAB Coder
- GPU Coder **R2017b**
- Automated Driving System Toolbox **R2017a**



MathWorks® can help you do Deep Learning

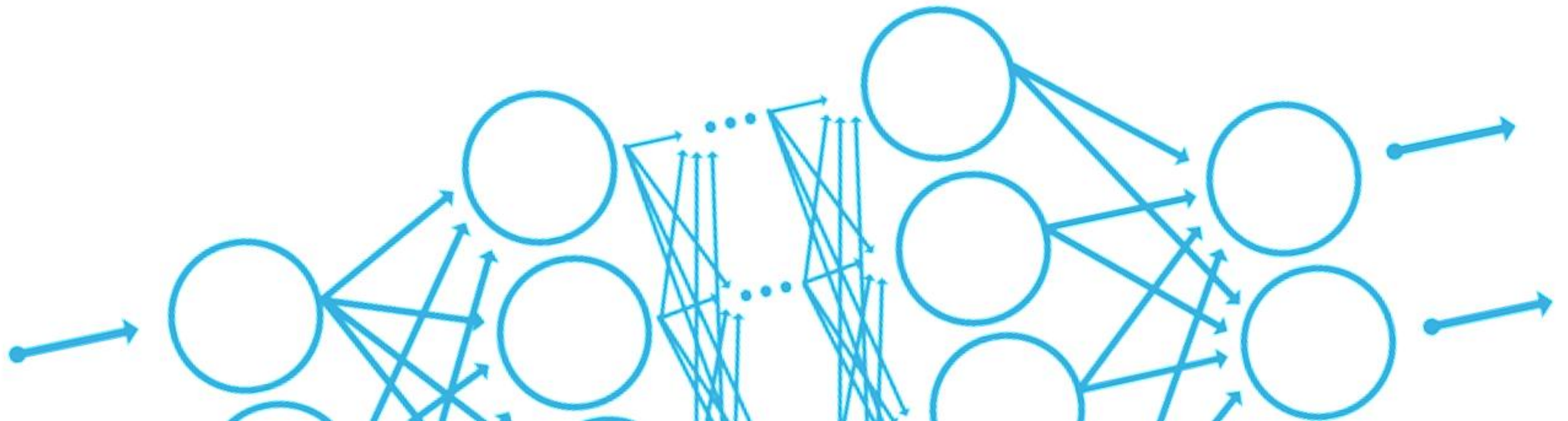
Free resources

- **Guided evaluations with a MathWorks deep learning engineer**
- Proof-of-concept projects
- **Deep learning hands-on workshop**
- Seminars and technical deep dives
- Documentation and user community

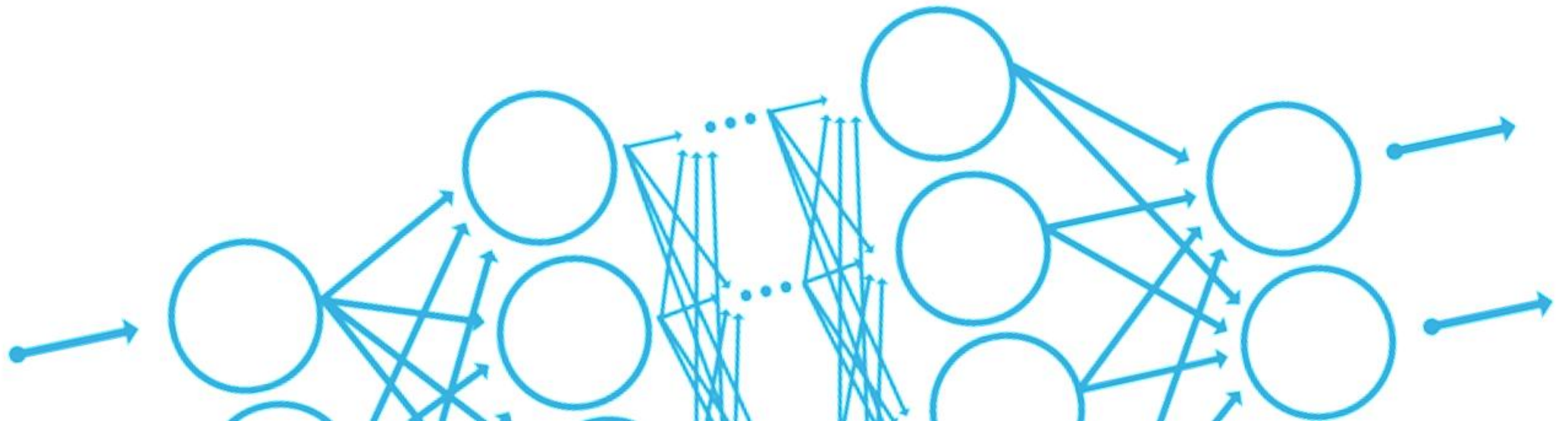
Other options

- Consulting services
- Training courses
- Technical support
- Advanced customer support
- Installation, enterprise, and cloud deployment
- [Deep Learning: Top 7 Ways to Get Started with MATLAB](#)

Thank you!

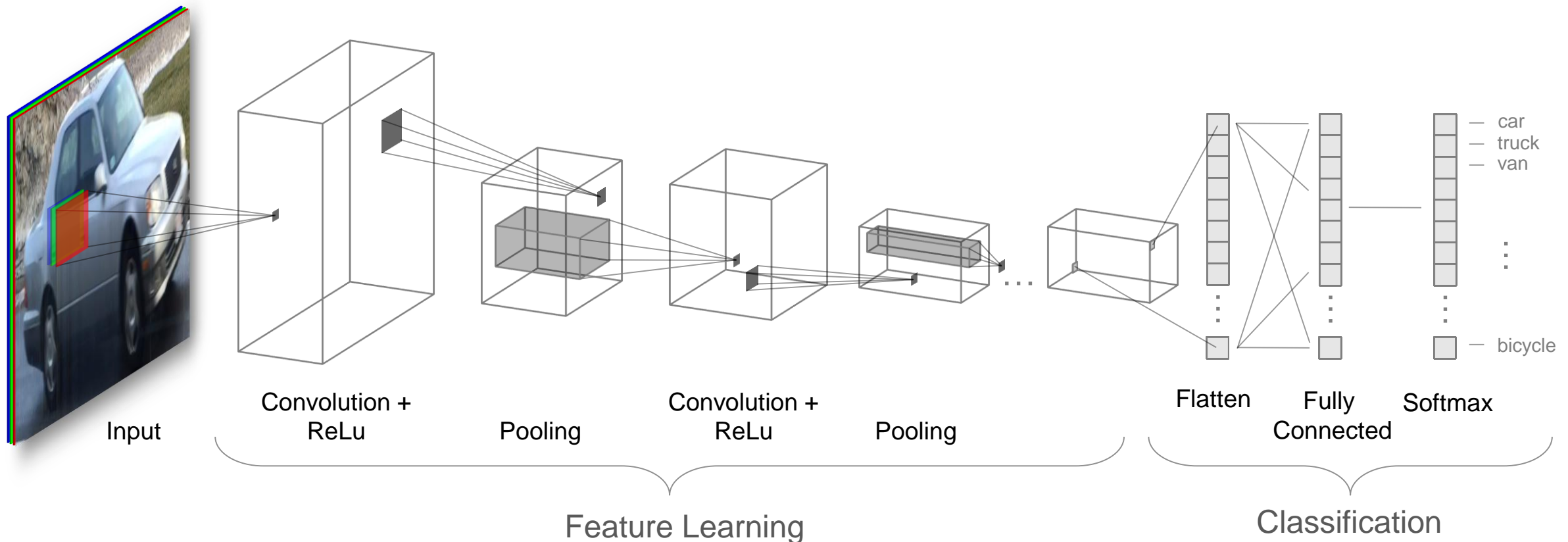


APPENDIX: Extra Slides

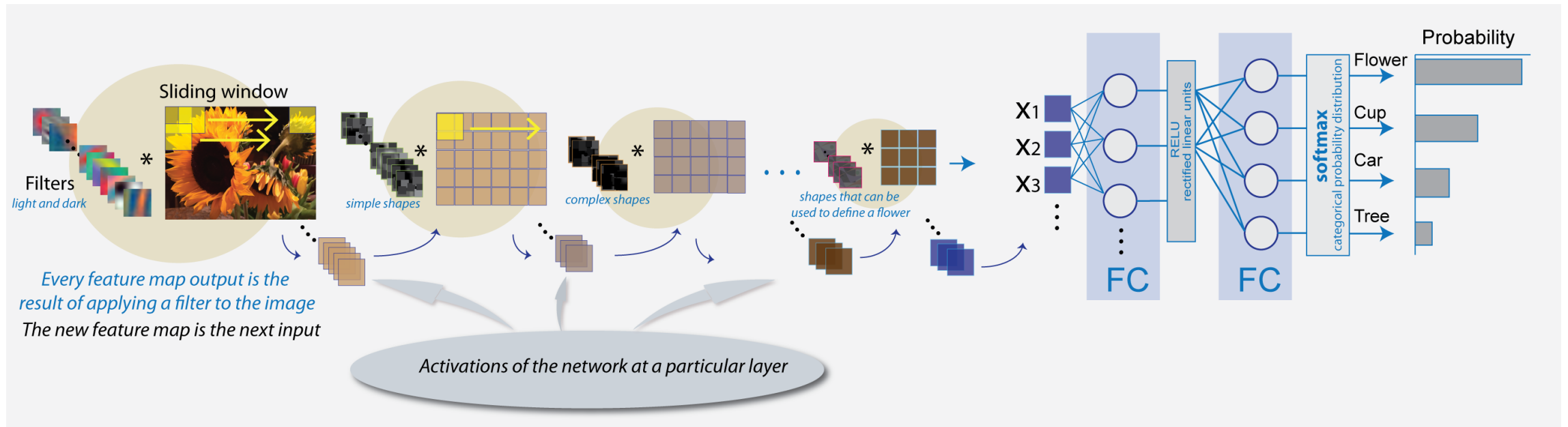
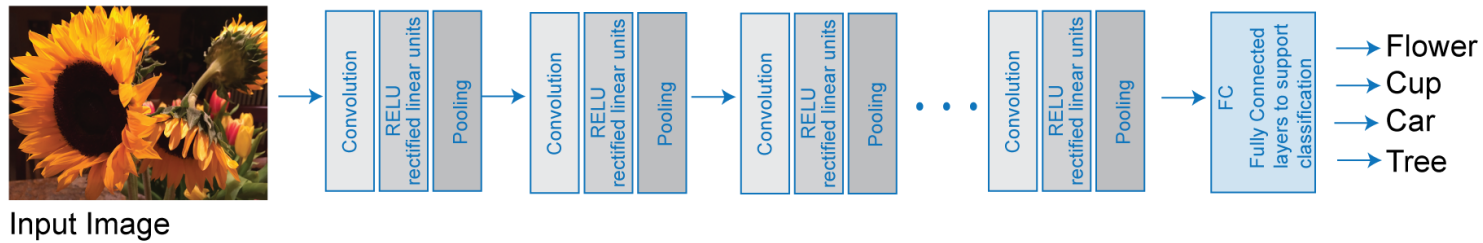


Convolutional Neural Networks

- Train “deep” neural networks on structured data (e.g. images, signals, text)
- Implements Feature Learning: Eliminates need for “hand crafted” features
- Trained using GPUs for performance

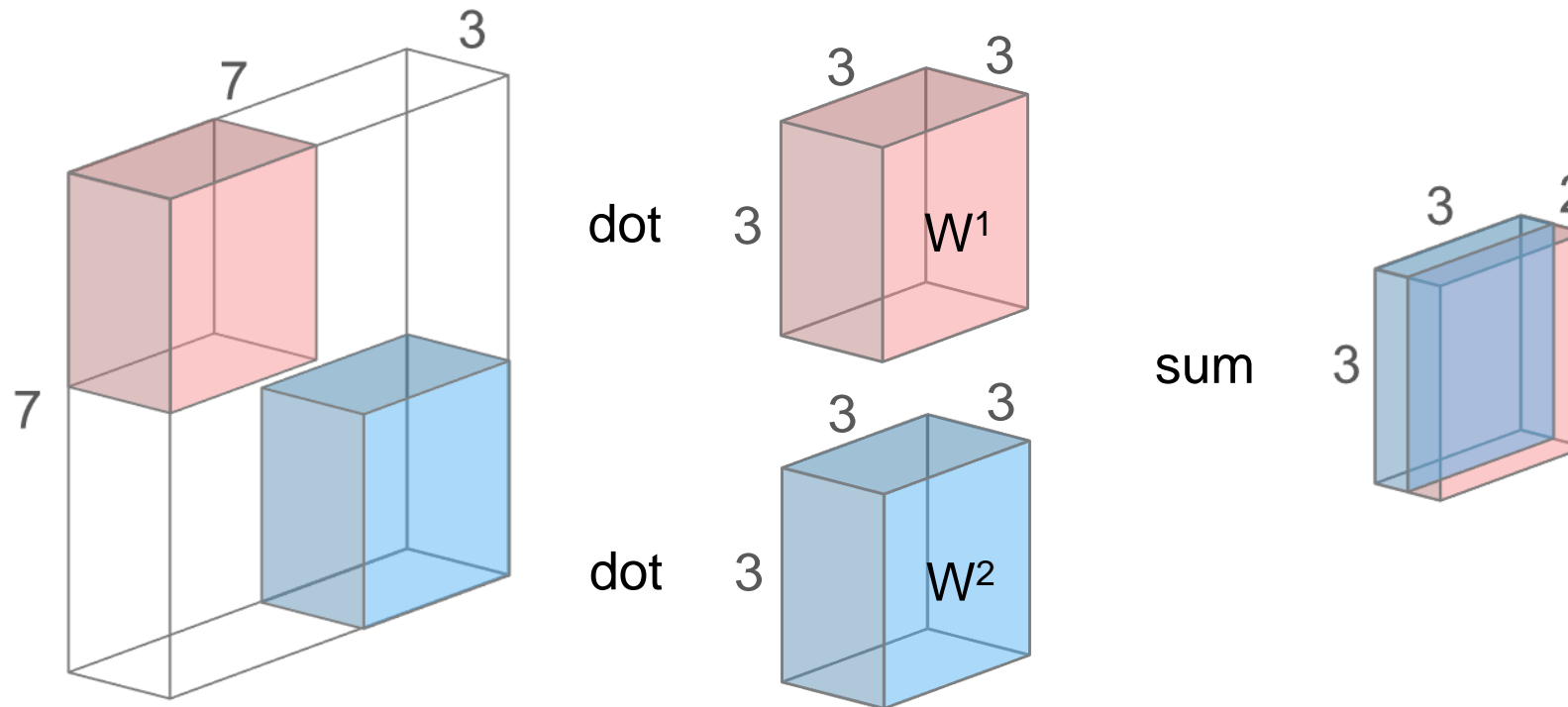


Convolutional Neural Networks



Convolution Layer

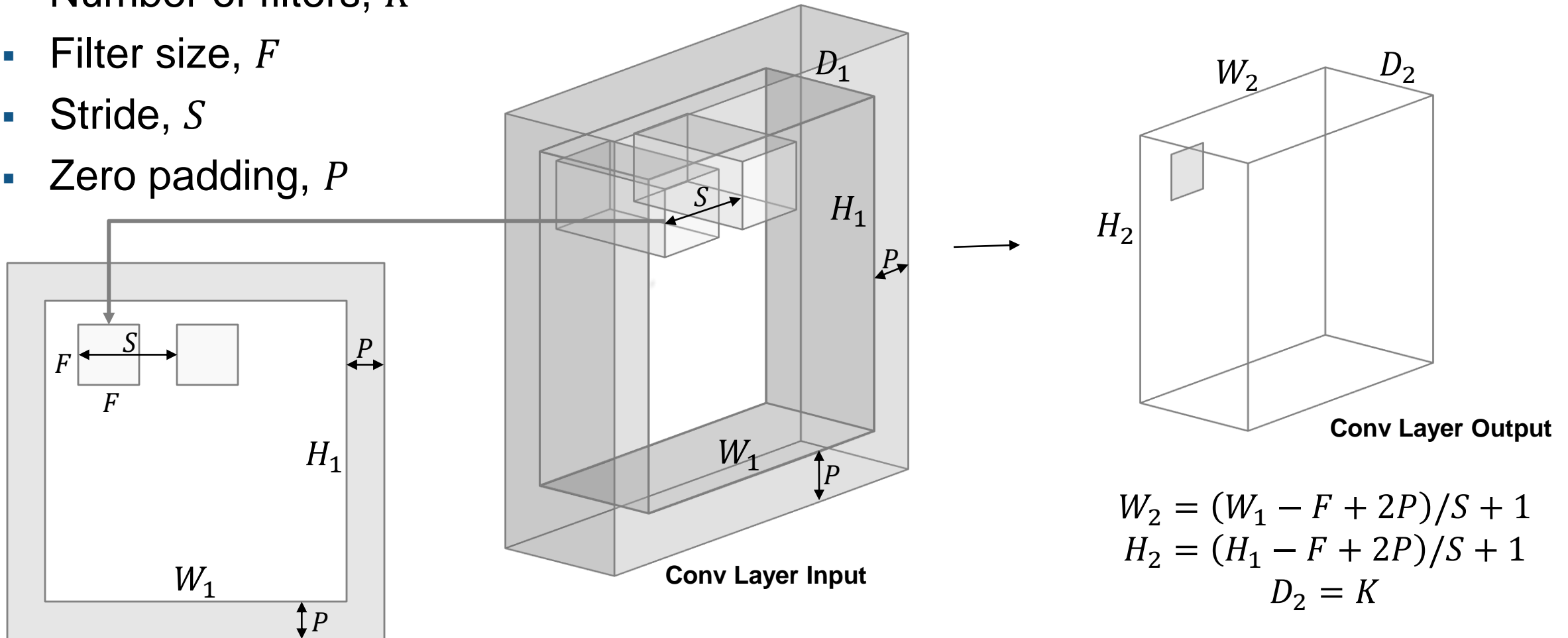
- Core building block of a CNN
- Convolve the filters sliding them across the input, computing the dot product



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

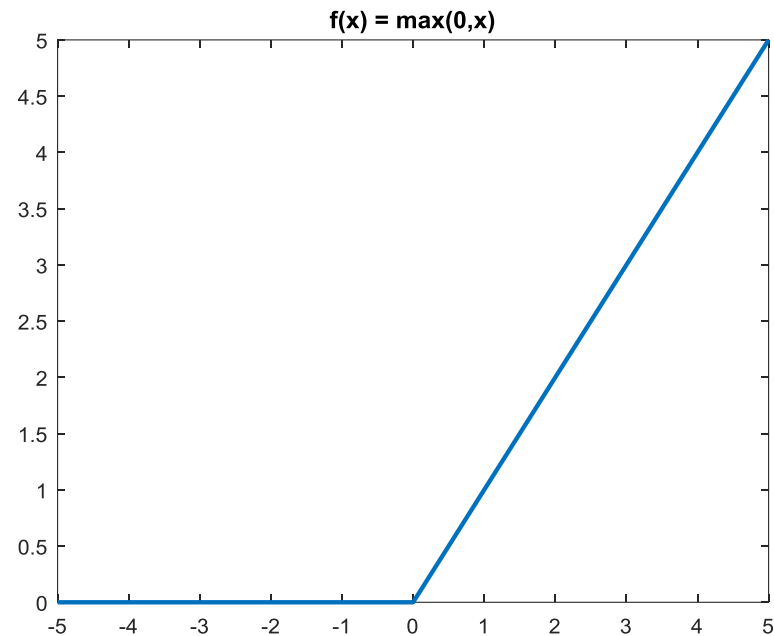
Convolution Layer – Choosing Hyperparameters

- Number of filters, K
- Filter size, F
- Stride, S
- Zero padding, P



Rectified Linear Unit (ReLU) Layer

- Frequently used in combination with Convolution layers
- Do not add complexity to the network
- Most popular choice: $f(x) = \max(0, x)$, activation is thresholded at 0



Benefits of Batch Normalization

- Batch normalization reduces the problem of **internal covariate shift**
 - *Internal covariate shift*: Neural networks can be slow to train because of low learning rates and careful parameter optimization – because each layer's inputs change in distribution during training
- It enables higher learning rates
- It regularizes the model
- Source: *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* (<https://arxiv.org/pdf/1502.03167v3.pdf>)

Pooling Layer

- Perform a **downsampling** operation across the spatial dimensions
- Goal: progressively decrease the size of the layers
- Max pooling and average pooling methods
- Popular choice: Max pooling with 2x2 filters, Stride = 2

