

MATLAB EXPO 2018

Introduction to Machine Learning
and Deep Learning

Valerie Leung



Machine learning in action

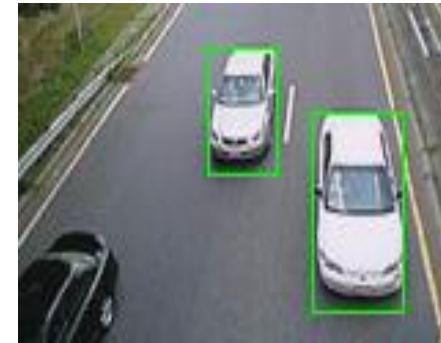
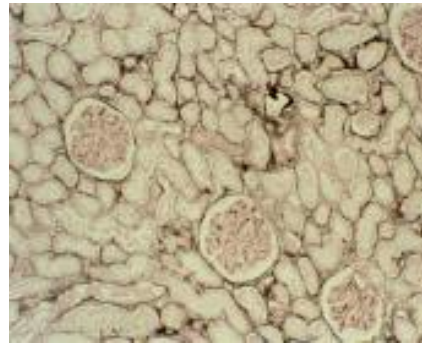
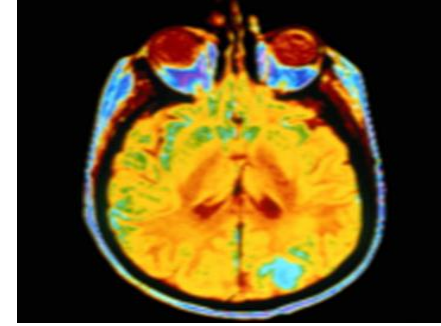


CamVid Dataset

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

Machine learning is everywhere

- Image recognition
- Speech recognition
- Stock prediction
- Medical diagnosis
- Predictive maintenance
- Language translation
- and more...



Agenda

- Machine Learning
 - What it is
 - Example : object classification
- Deep Learning
 - What it is and why it is popular
 - Object classification revisited
- Tackling time series with deep learning

What is machine learning?

Machine learning uses **data** and produces a **program** to perform a **task**

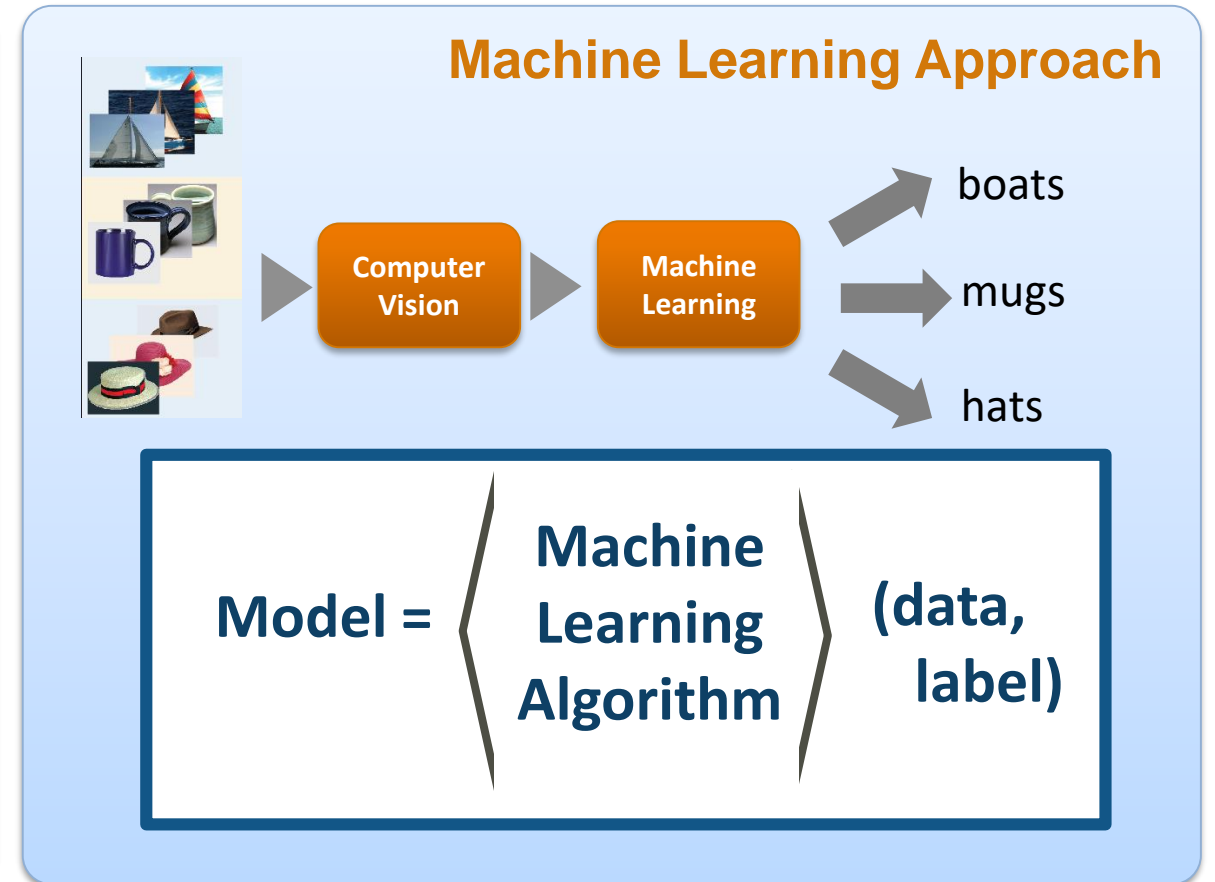
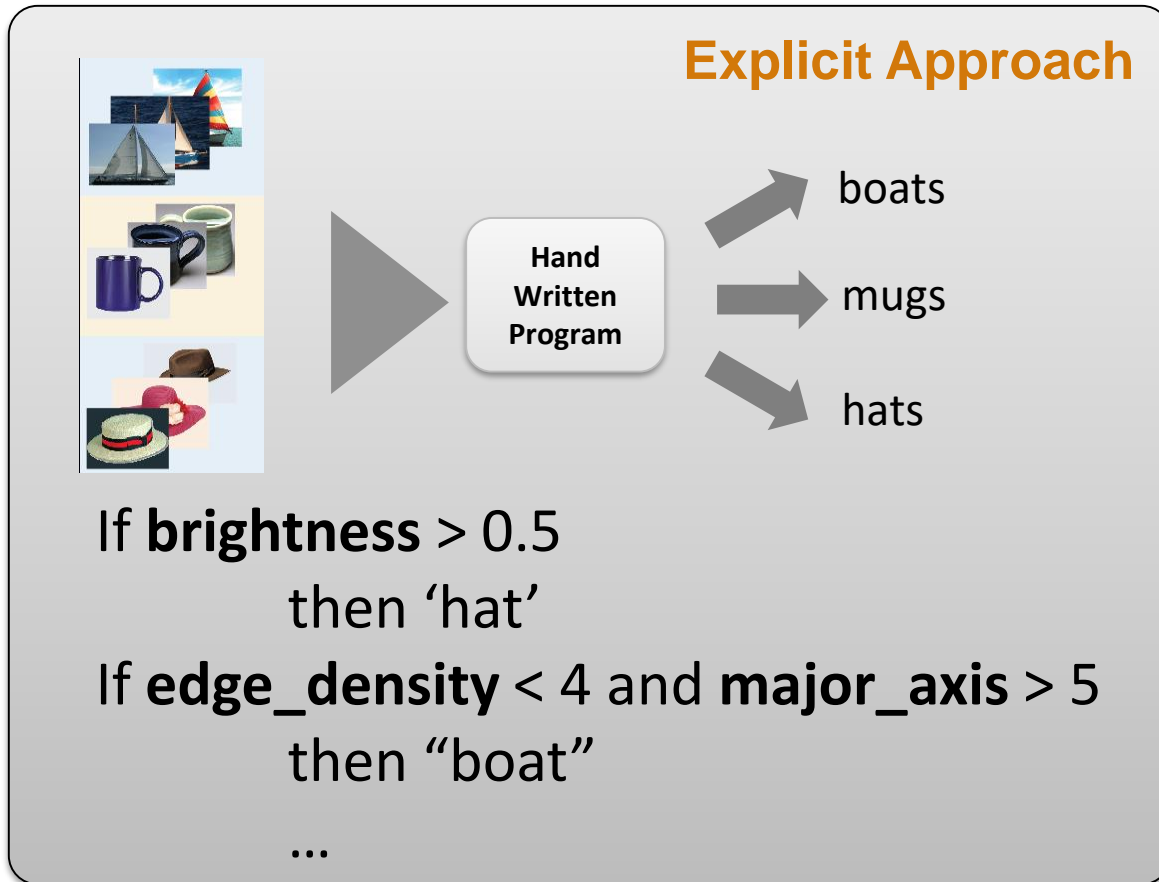
Task: Image Category Recognition



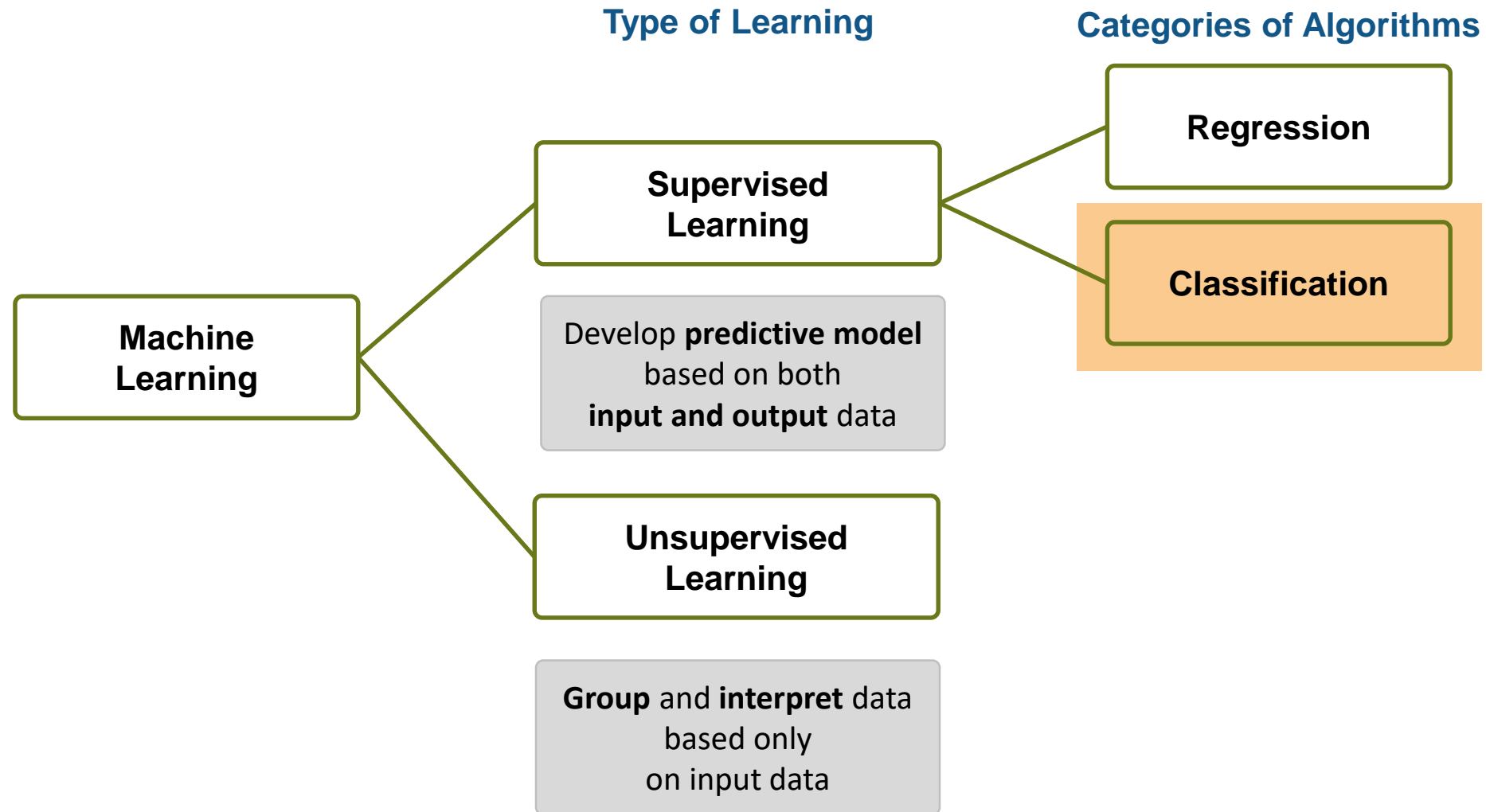
What is machine learning?

Machine learning uses **data** and produces a **program** to perform a **task**

Task: Image Category Recognition

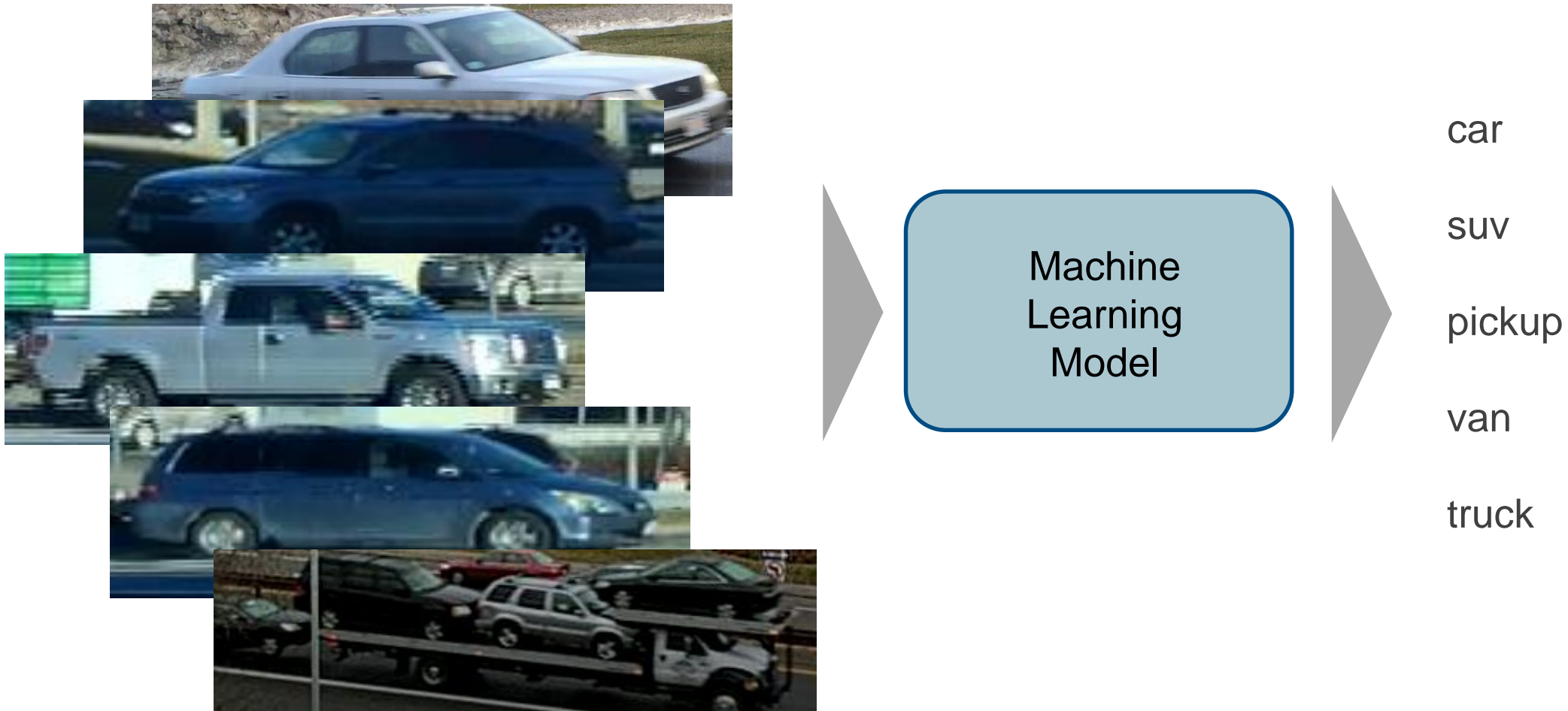


Machine Learning: problem specific overview

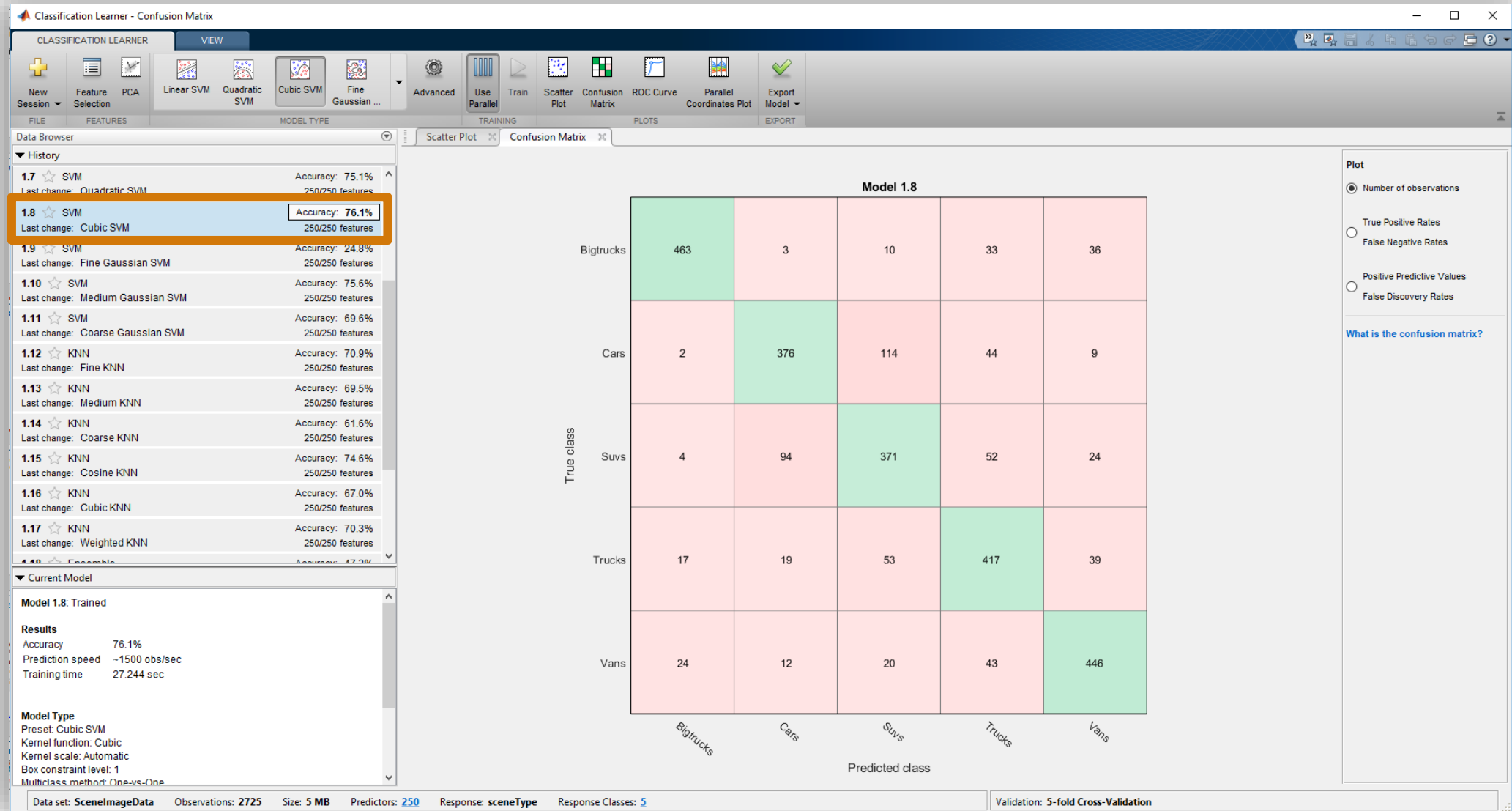


Example: Object Classification with Machine Learning

Task: Distinguish between 5 categories of vehicles



Using the Classification Learner App to determine the best model



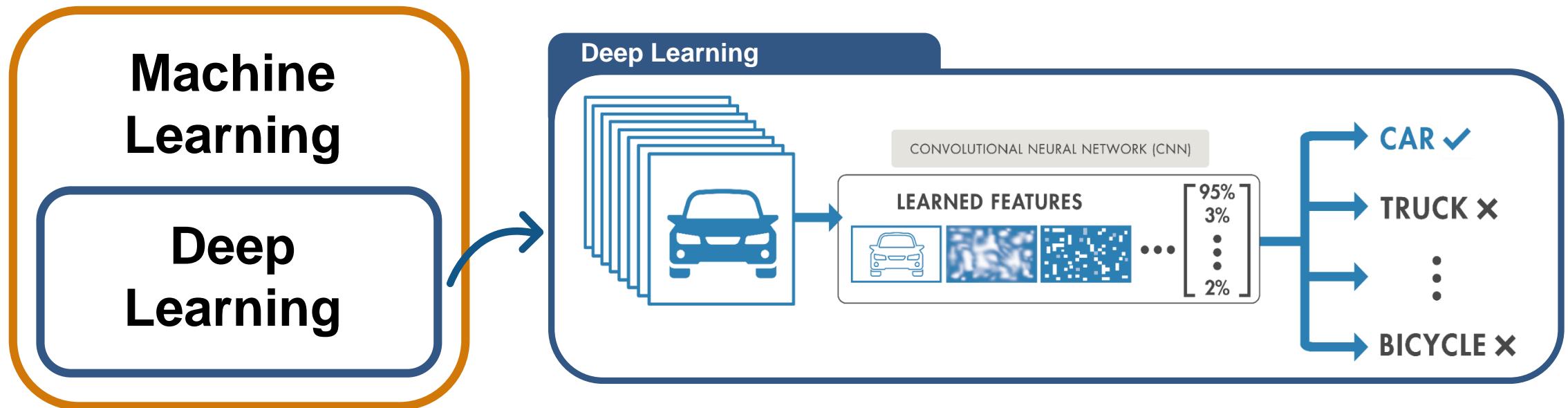
Agenda

- Machine Learning
 - What it is
 - Example : object classification
- Deep Learning
 - What it is and why it is popular
 - Object classification revisited
- Tackling time series with deep learning

Deep Learning for Classification

What is Deep Learning ?

- Subset of machine learning with **automatic feature extraction**
 - Learns features and tasks directly from data
 - More data = better model



Deep Learning for Classification

Applications: autonomous driving, image classification, etc.



Detection of cars and road in autonomous driving systems

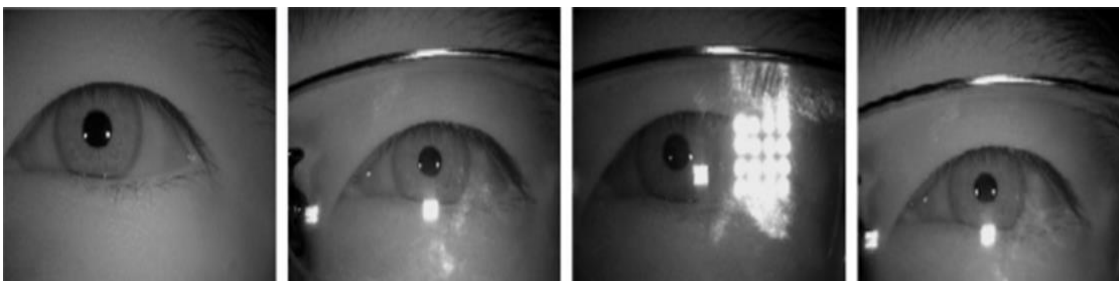


Single Image Age Estimation^{3,4}



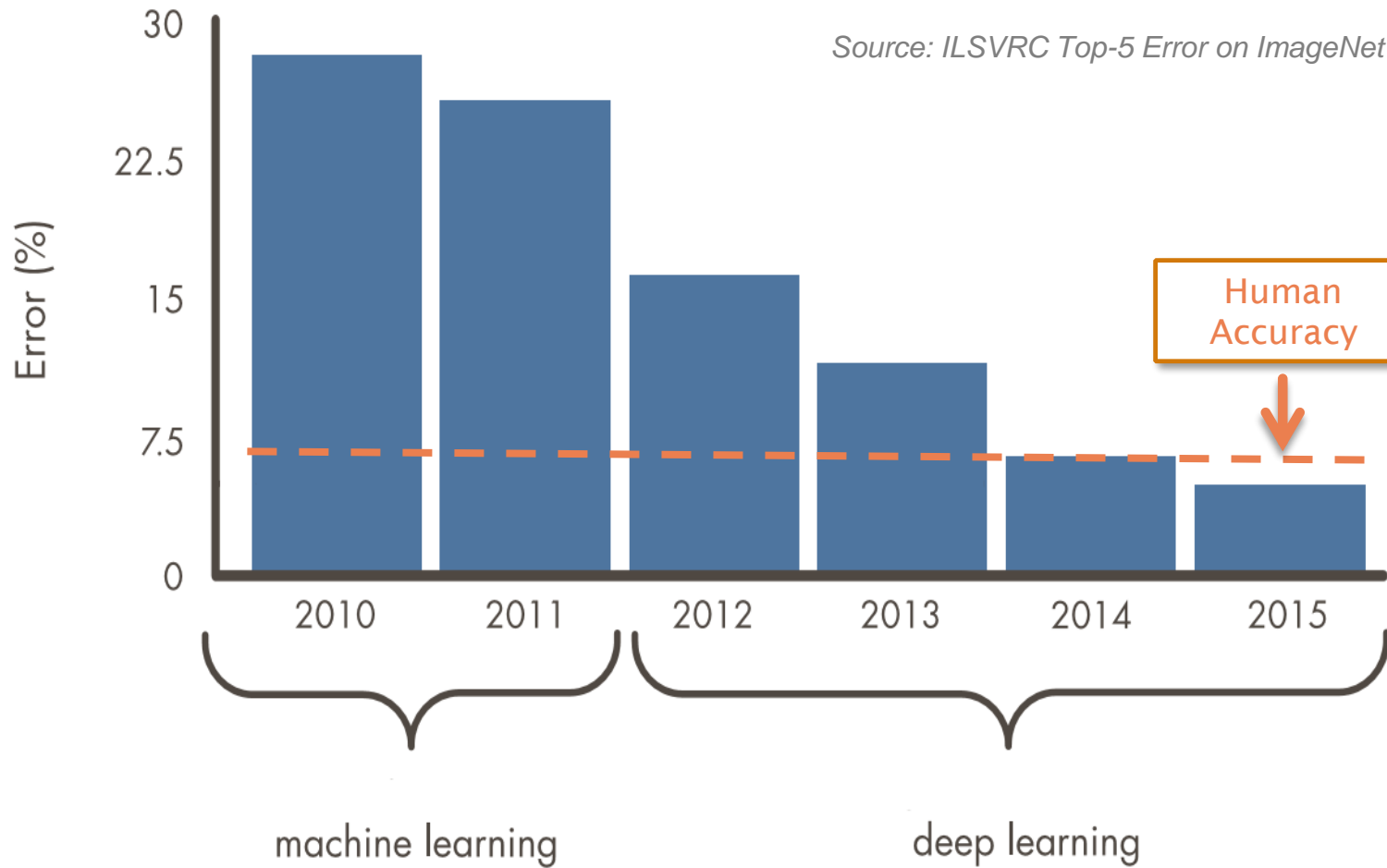
Rain Detection and Removal¹

Iris Recognition – 99.4% accuracy²

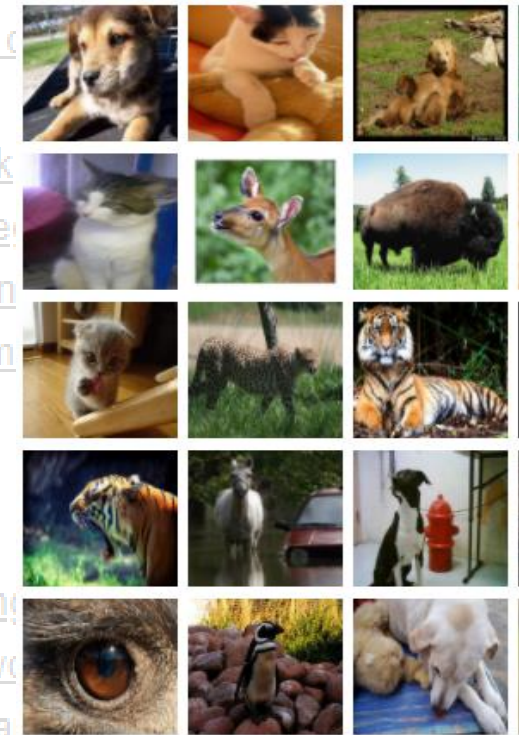


1. "Deep Joint Rain Detection and Removal from a Single Image" Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan
2. Source: An experimental study of deep convolutional features for iris recognition Signal Processing in Medicine and Biology Symposium (SPMB), 2016 IEEE Shervin Minaee ; Amirali Abdolrashidiy ; Yao Wang; An experimental study of deep convolutional features for iris recognition
3. "DEX: Deep Expectation of apparent age from a single image", Rasmus Rothe, Radu Timofte and Luc Van Gool, Looking at People Workshop, ICCV 2015
4. Image source : https://en.wikipedia.org/wiki/Emmanuel_Macron

Deep learning models can surpass human accuracy



IMAGENET¹
Database for visual
object recognition
research

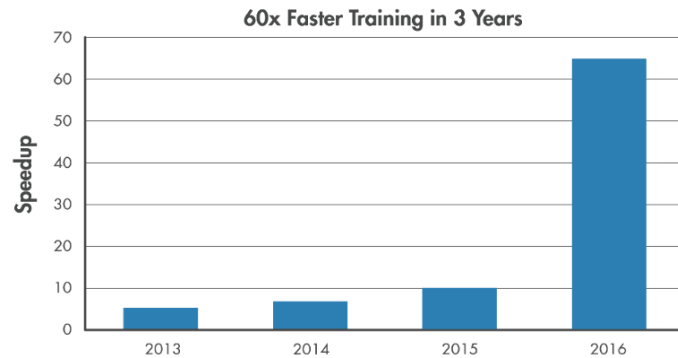


1. <http://www.image-net.org/>

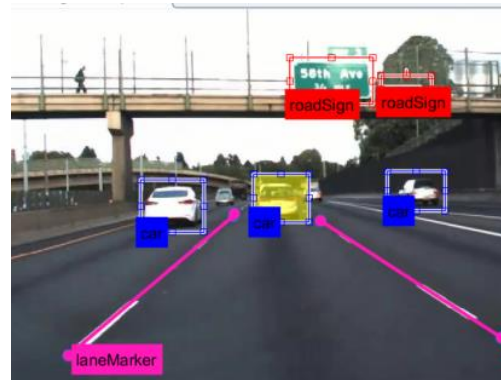
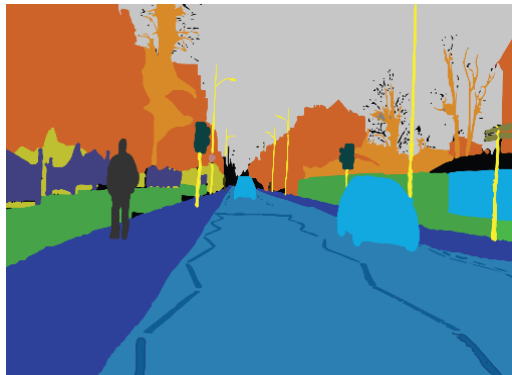
Deep Learning for Classification

Deep Learning Enablers

Increased GPU acceleration



Labeled public datasets



World-class models to be leveraged

AlexNet
PRETRAINED MODEL

VGG-16
PRETRAINED MODEL

ResNet
PRETRAINED MODEL

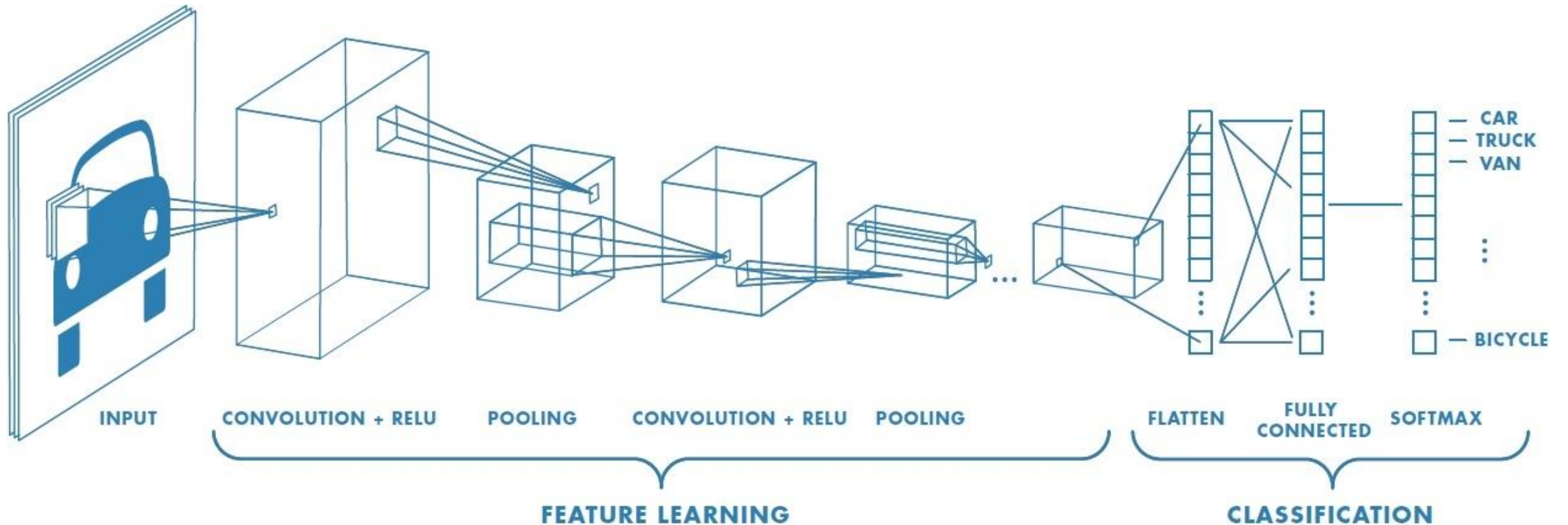
Caffe
MODELS

GoogLeNet
PRETRAINED MODEL

TensorFlow/Keras
MODELS

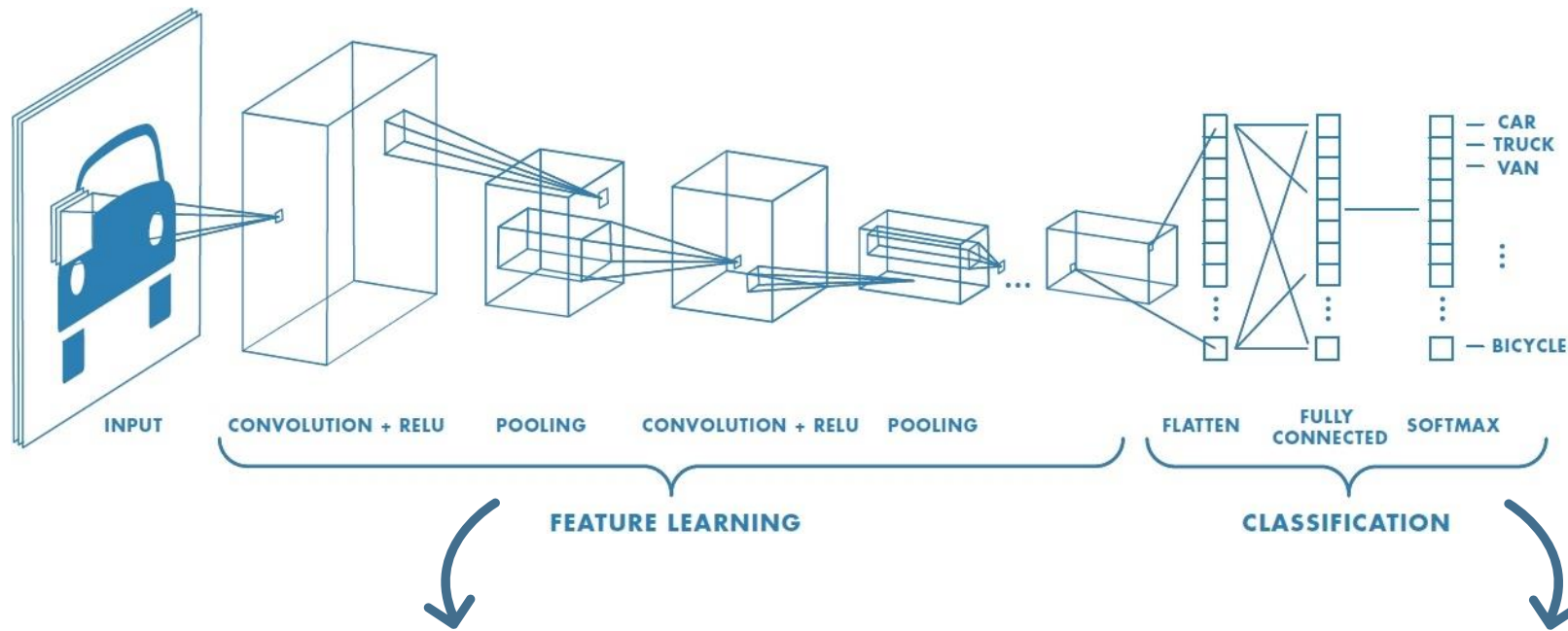
Deep Learning for Classification

What does a Convolutional Neural Network (CNN) architecture look like?



Deep Learning for Classification

What does a Convolutional Neural Network (CNN) architecture look like?



Convolution puts the input images through a set of convolutional filters, each of which activates certain features from the images.

Rectified linear unit (ReLU) allows for faster and more effective training by mapping negative values to zero and maintaining positive values.

Pooling simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network needs to learn about.

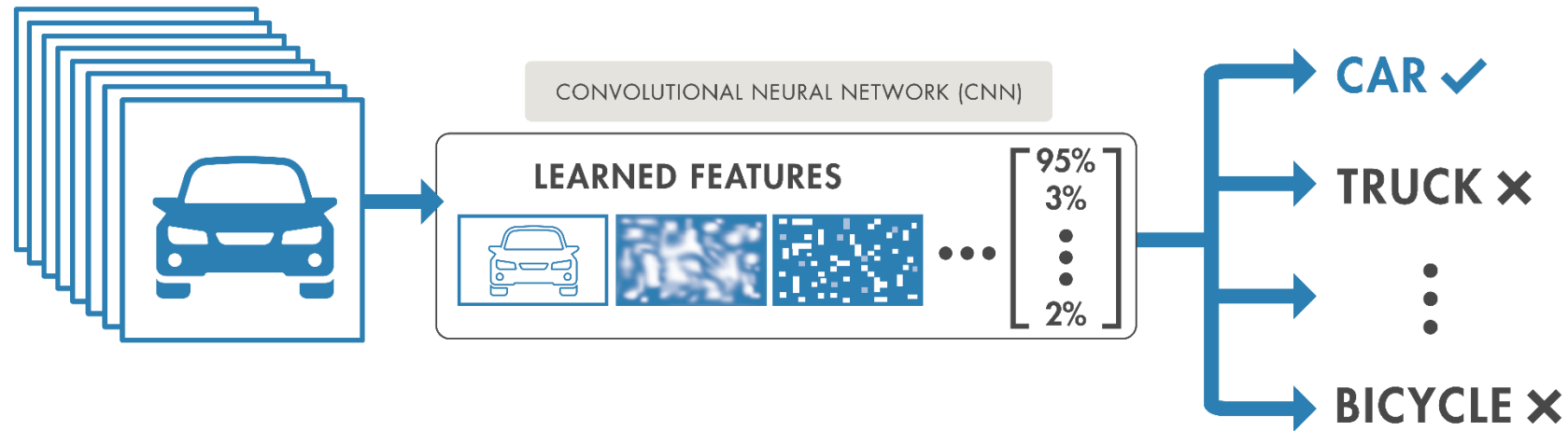
The next-to-last layer is a **fully connected layer (FC)** that outputs a vector of K dimensions where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified.

The final layer of the CNN architecture uses a **softmax** function to provide the classification output.

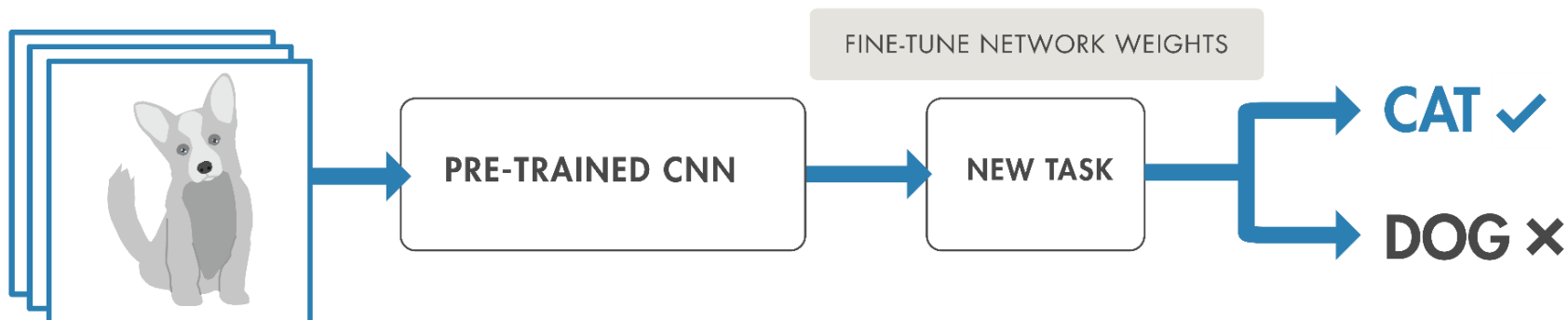
Deep Learning for Classification

Two approaches

1. Train a Deep Neural Network from Scratch



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



Deep Learning for Classification

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

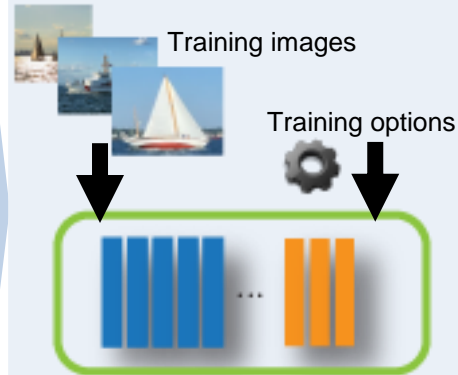
Replace final layers

New layers to learn features specific to your data



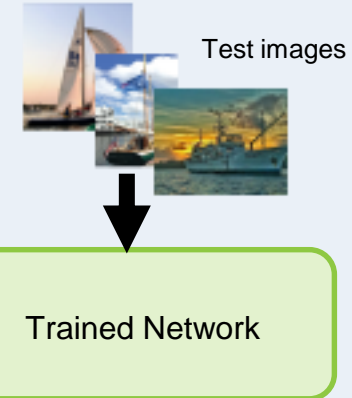
Fewer classes
Learn faster

Train network

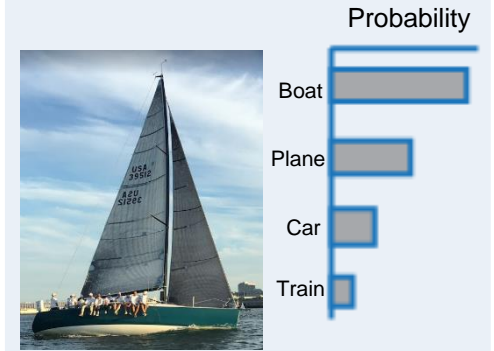


100s images
10s classes

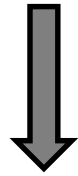
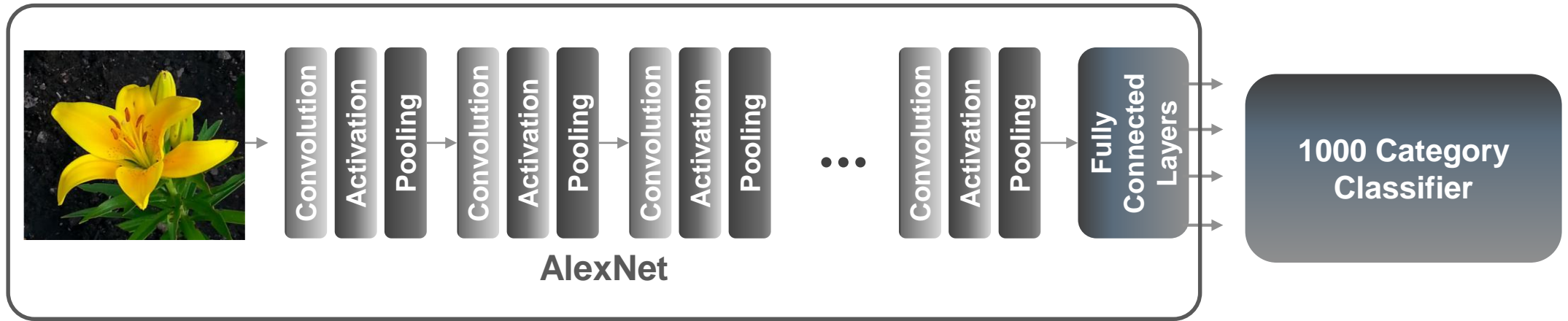
Predict and assess network accuracy



Deploy results



Example: Object Classification revisited with Transfer Learning



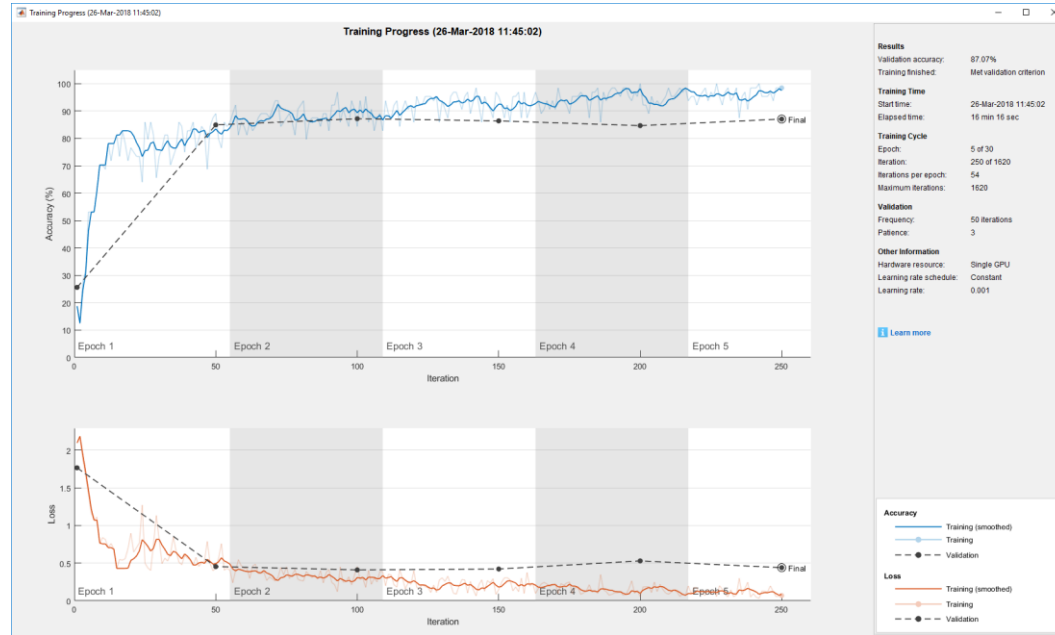
New Data



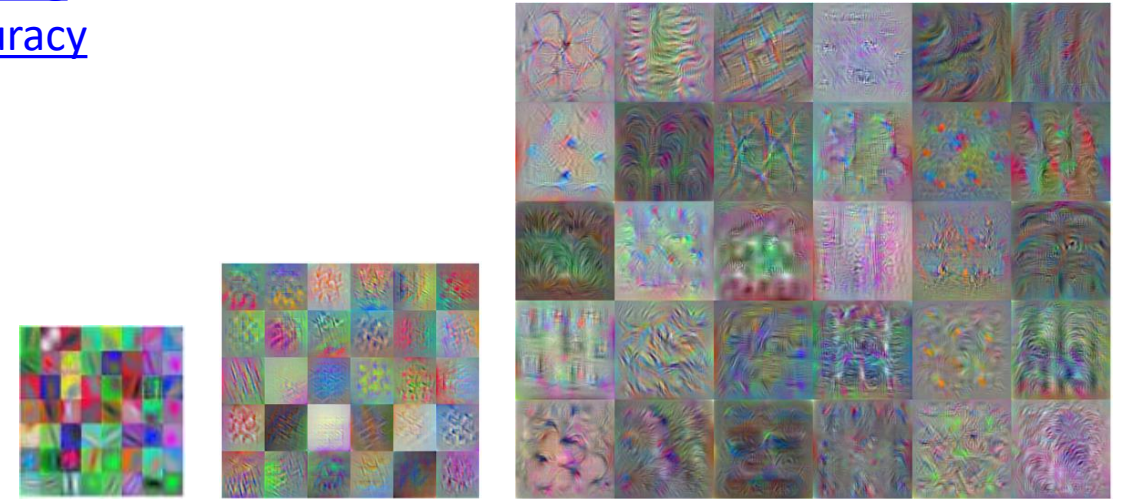
- car →
- suv →
- pickup →
- van →
- truck →



How do I visualize and debug deep neural networks?

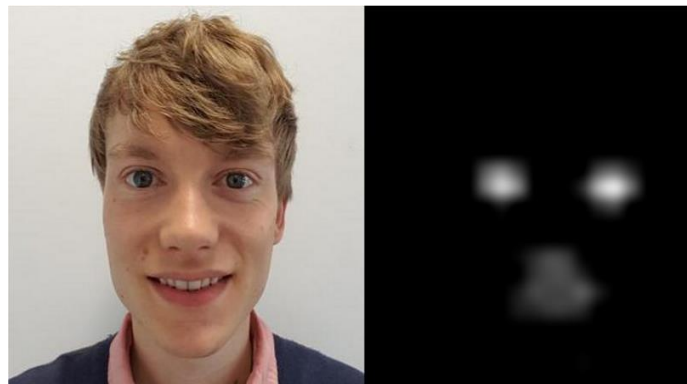


Training Accuracy Plot



Feature Visualization

Network Activations



Deep Dream

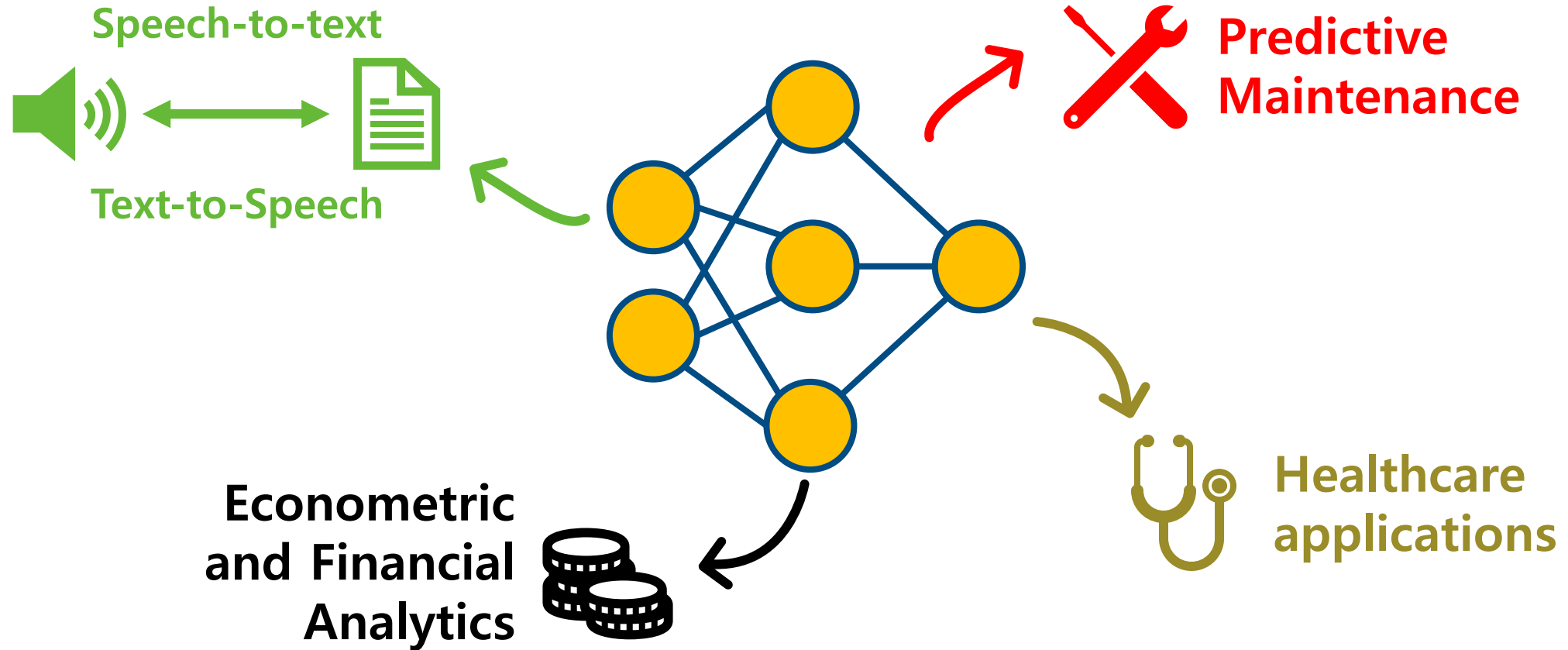
Recap: Deep Learning for Images

- Deep learning: end-to-end training including automatic feature extraction
- Two approaches:
 - Train from scratch - requires lots of labeled training data
 - Transfer learning – leverage existing models and adapt to new task
- Visualize and debug networks with built-in functions

Agenda

- Machine Learning
 - What it is
 - Example : object classification
- Deep Learning
 - What it is and why it is popular
 - Object classification revisited
- Tackling time series with deep learning

Deep Learning for Time Series

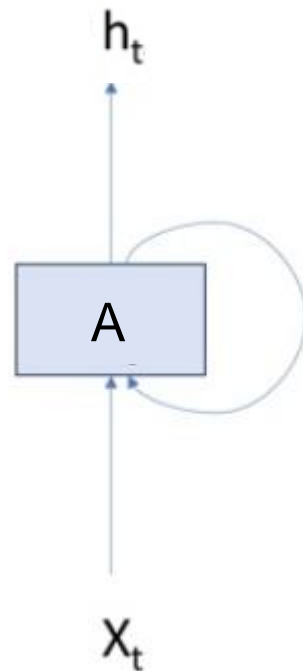


Basic Network: Recurrent Neural Network (RNN)

Output

Hidden state

Input

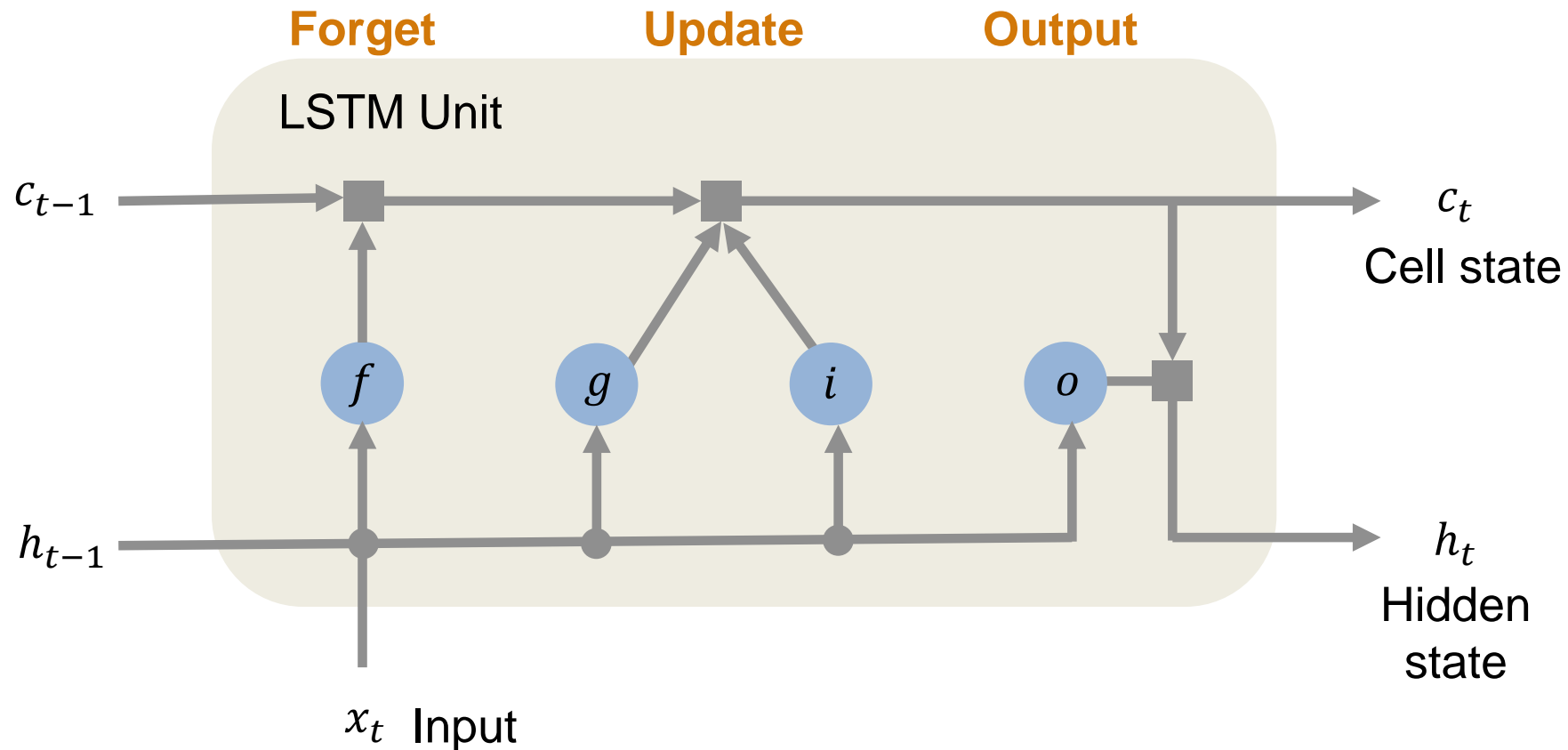


- Problems

- Vanishing gradients: only short-term dependencies are captured – information from earlier time steps decays
- Exploding gradients: error can grow drastically with each time step

Long Short-Term Memory (LSTM)







Selectively retains relevant information and forgets irrelevant information

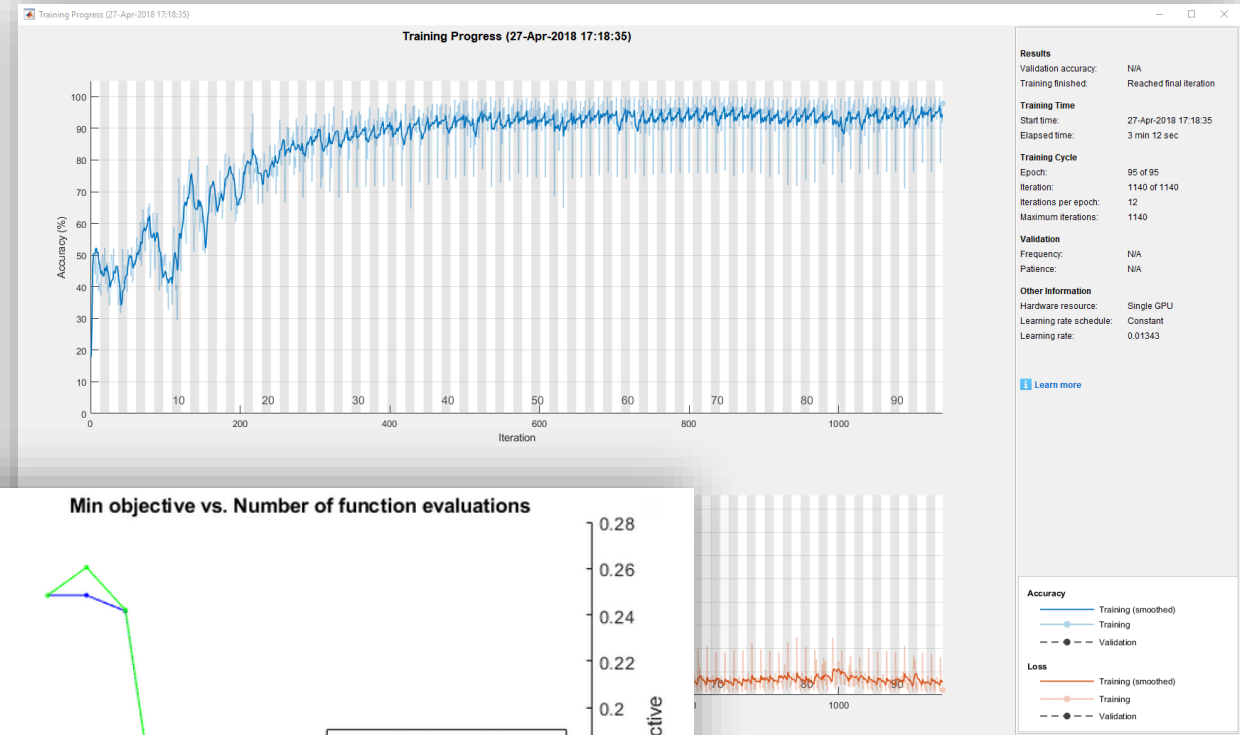


Example: Human Activity Recognition from Mobile Phone Data

Objective: Train a classifier to classify human activity from sensor data

Data:

Predictors	3-axial Accelerometer and Gyroscope data 
Response	Activity:     



Results
 Validation accuracy: N/A
 Training finished: Reached final iteration

Training Time
 Start time: 27-Apr-2018 17:18:35
 Elapsed time: 3 min 12 sec

Training Cycle
 Epoch: 95 of 95
 Iteration: 1140 of 1140
 Iterations per epoch: 12
 Maximum iterations: 1140

Validation
 Frequency: N/A
 Patience: N/A

Other Information
 Hardware resource: Single GPU
 Learning rate schedule: Constant
 Learning rate: 0.01343

[Learn more](#)

Accuracy
 Training (smoothed)
 Training
 Validation

Loss
 Training (smoothed)
 Training
 Validation

How do I know if my deep network is defined correctly?

MathWorks® Products Solutions Academia Support Community Events

File Exchange File Exchange

MATLAB Central ▾ Files Authors Tags Comments My File Exchange Submit About Trial software

Network Analyzer
Visualize and Analyze network

Deep Learning Network Analyzer for Neural Network Toolbox

version 1.0 (15.1 KB) by MathWorks Neural Network Toolbox Team

Visualize and Analyze Deep Learning Networks

★★★★★ 6 Ratings
134 Downloads
Updated 19 Apr 2018

Deep Learning Network Analyzer

rnet 177 layers 0 warnings 0 errors

Analysis date: 11-May-2018 14:42:34

ANALYSIS RESULT

↑	NAME	TYPE	ACTIVATIONS	LEARNABLES
1	input_1 224x224x3 images with 'zerocenter' normalization	Image Input	224x224x3	-
2	conv1 64 7x7x3 convolutions with stride [2 2] and padding [3 3 3 3]	Convolution	112x112x64	Weights 7x7x3x64 Bias 1x1x64
3	bn_conv1 Batch normalization with 64 channels	Batch Normalization	112x112x64	Offset 1x1x64 Scale 1x1x64
4	activation_1_relu ReLU	ReLU	112x112x64	-
5	max_pooling2d_1 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	55x55x64	-
6	res2a_branch2a 64 1x1x64 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	55x55x64	Weights 1x1x64x64 Bias 1x1x64
7	bn2a_branch2a Batch normalization with 64 channels	Batch Normalization	55x55x64	Offset 1x1x64 Scale 1x1x64
8	activation_2_relu ReLU	ReLU	55x55x64	-
9	res2a_branch2b 64 3x3x64 convolutions with stride [1 1] and padding 'same'	Convolution	55x55x64	Weights 3x3x64x64 Bias 1x1x64
10	bn2a_branch2b	Batch Normalization	55x55x64	Offset 1x1x64

Recap – Deep Learning for time series

- LSTMs good for handling temporal dependence
- Define with [lstm_layer](#) and [bi_lstm_layer](#)

Key Takeaways : Why use MATLAB for Deep Learning

- Handle and label large sets of data
 - Handle images with `imageDatastore`
- Accelerate deep learning with GPUs on desktops, clusters, and clouds
- Visualize and debug deep neural networks
- Access and use models from experts

