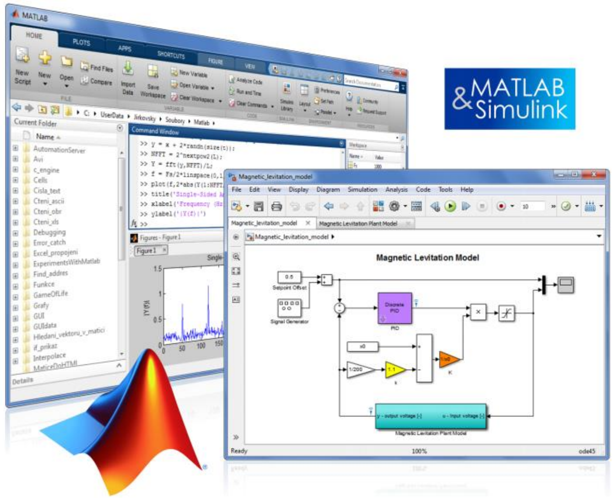
20.11.2019 Praha

TCP 2019

Deep learning v prostředí MATLAB



```
>> y = x + 2*randn(size(t));
>> NFFT = 2^nextpow2(L);
>> Y = fft(y,NFFT)/L;
>> f = Fs/2*linspace(0,1
>> plot(f,2*abs(Y(1:NFFT
>> title('Single-Sided A
>> xlabel('Frequency (Hz
>> ylabel('|Y(f)|')
>>
```

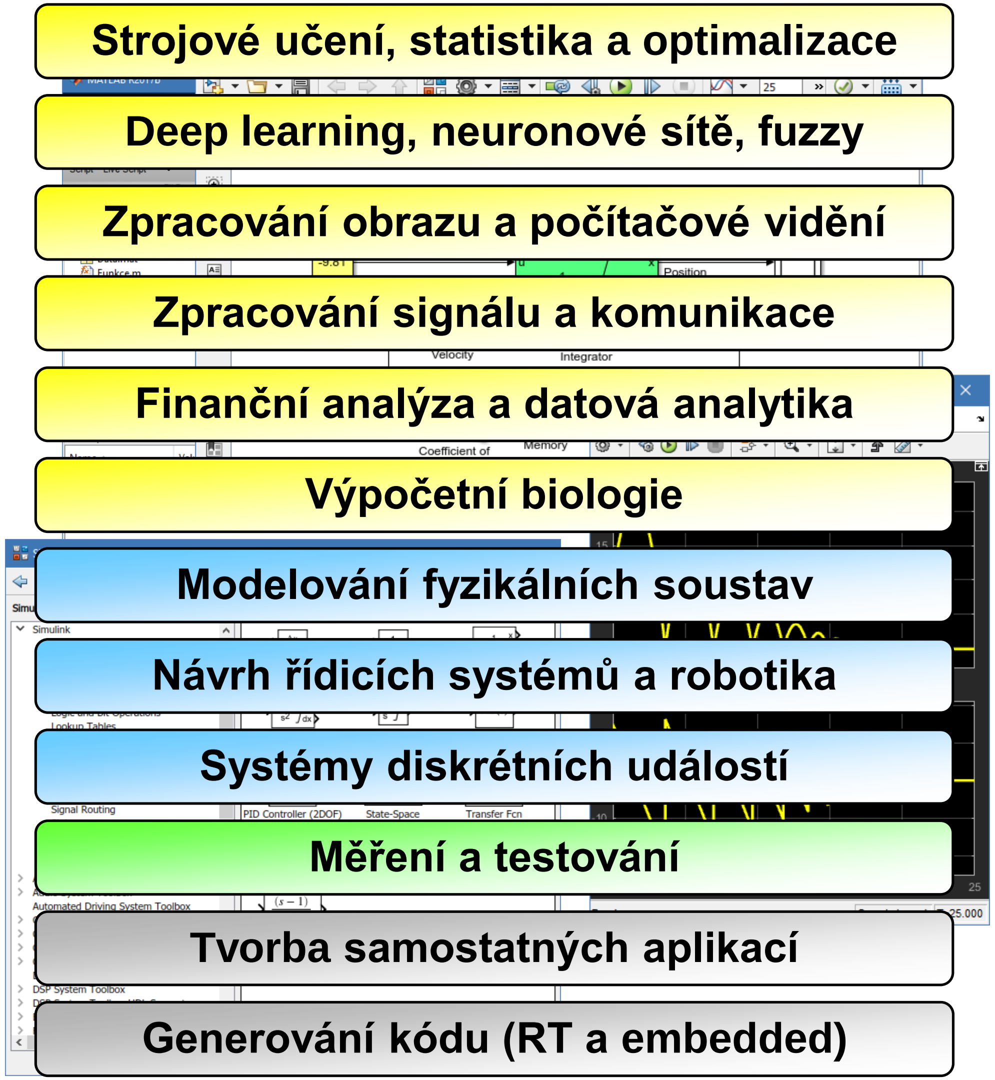
Jaroslav Jirkovský
jirkovsky@humusoft.cz

www.humusoft.cz
info@humusoft.cz

www.mathworks.com

Co je MATLAB a Simulink

- MATLAB
  - inženýrský nástroj a interaktivní prostředí pro vědecké a technické výpočty
  - grafické a výpočetní nástroje
  - grafické aplikace (GUI, APPS)
  - otevřený systém
- Simulink
  - nadstavba MATLABu
  - modelování, simulace a analýza dynamických systémů
  - prostředí blokových schémat
  - platforma pro Model Based Design
- Aplikační knihovny



Strojové učení, statistika a optimalizace

Deep learning, neuronové sítě, fuzzy

Zpracování obrazu a počítačové vidění

Zpracování signálu a komunikace

Finanční analýza a datová analytika

Výpočetní biologie

Modelování fyzikálních soustav

Návrh řídicích systémů a robotika

Systémy diskrétních událostí

Měření a testování

Tvorba samostatných aplikací

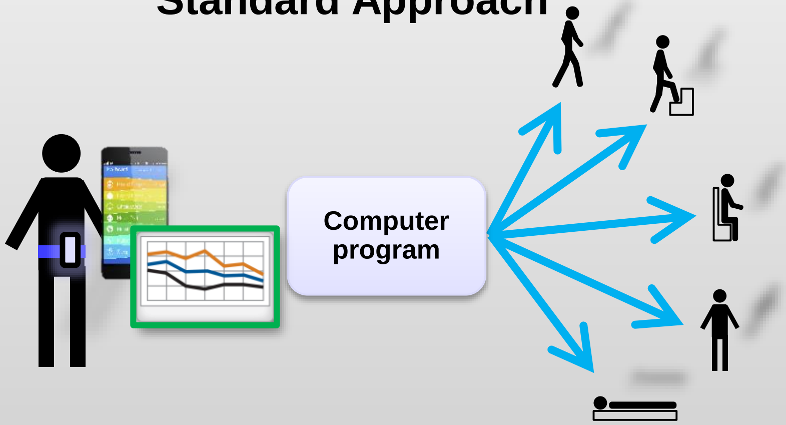
Generování kódu (RT a embedded)

What is Machine Learning ?

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

Task: Human Activity Detection

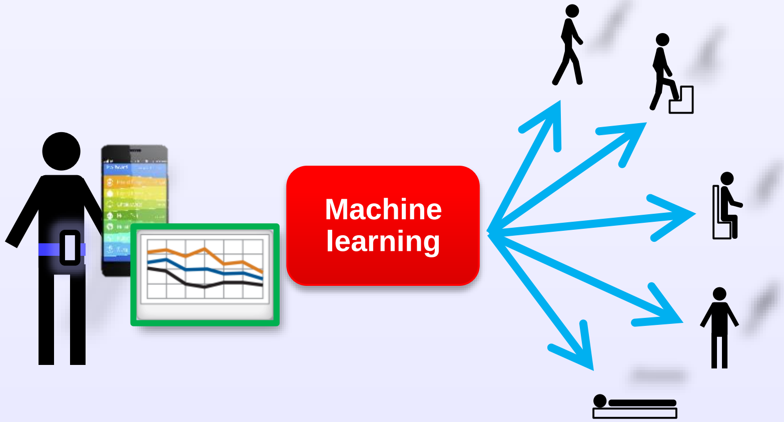
Standard Approach



Computer program

Hand Written Program	Formula or Equation
If X_acc > 0.5 then “SITTING” If Y_acc < 4 and Z_acc > 5 then “STANDING” …	$Y_{activity}$ $= \beta_1 X_{acc} + \beta_2 Y_{acc}$ $+ \beta_3 Z_{acc} +$ …

Machine Learning Approach



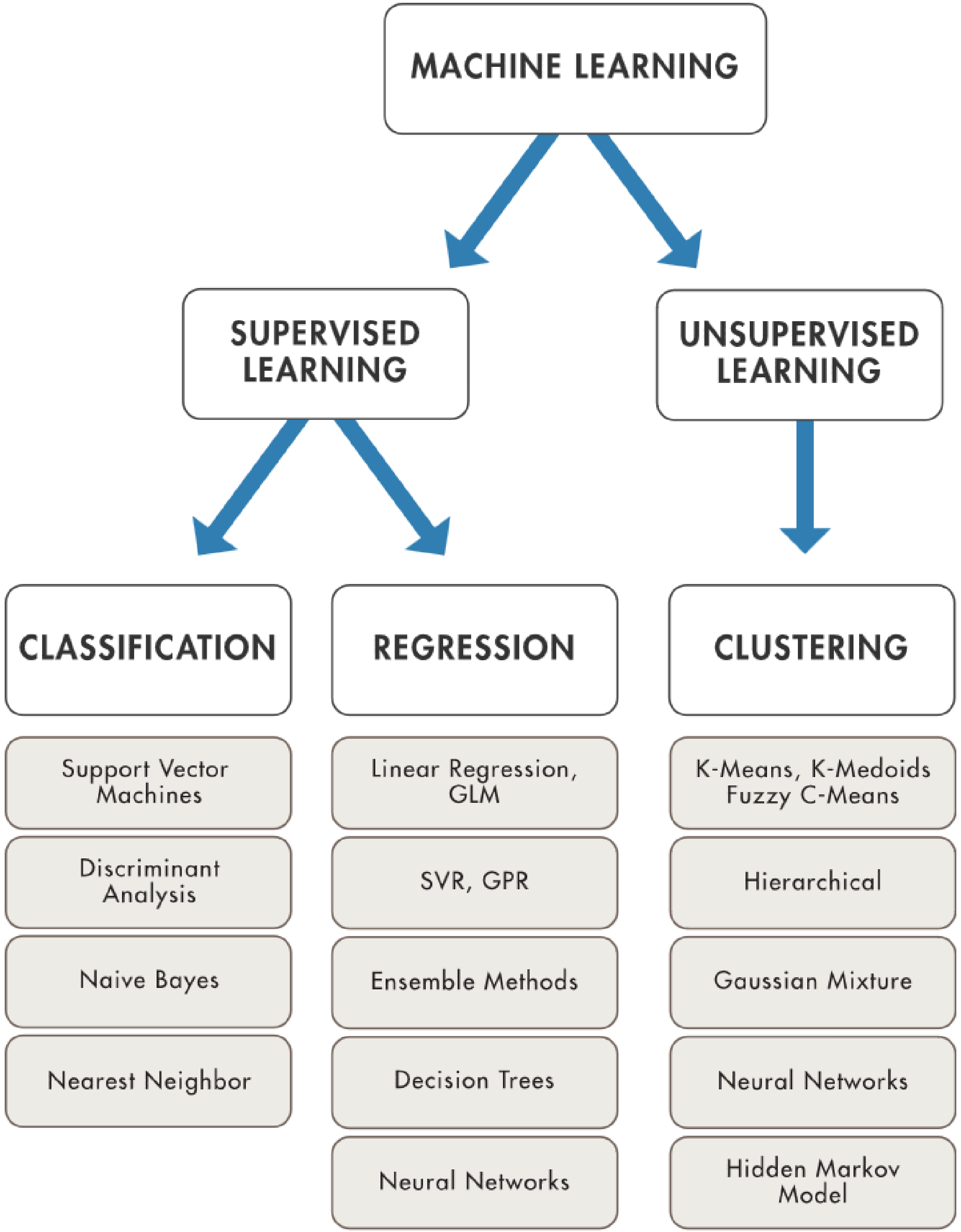
Machine learning

$$\textit{model}: \text{Inputs} \rightarrow \text{Outputs}$$

$$\textit{model} = < \begin{matrix}\textit{Machine}\\ \textit{Learning}\\ \textit{Algorithm}\end{matrix} > (\textit{sensor\_data}, \textit{activity})$$

Machine Learning

- **Different Types of Learning:**



MACHINE LEARNING

SUPERVISED LEARNING

UNSUPERVISED LEARNING

CLASSIFICATION

REGRESSION

CLUSTERING

Support Vector Machines

Discriminant Analysis

Naive Bayes

Nearest Neighbor

Linear Regression, GLM

SVR, GPR

Ensemble Methods

Decision Trees

Neural Networks

K-Means, K-Medoids Fuzzy C-Means

Hierarchical

Gaussian Mixture

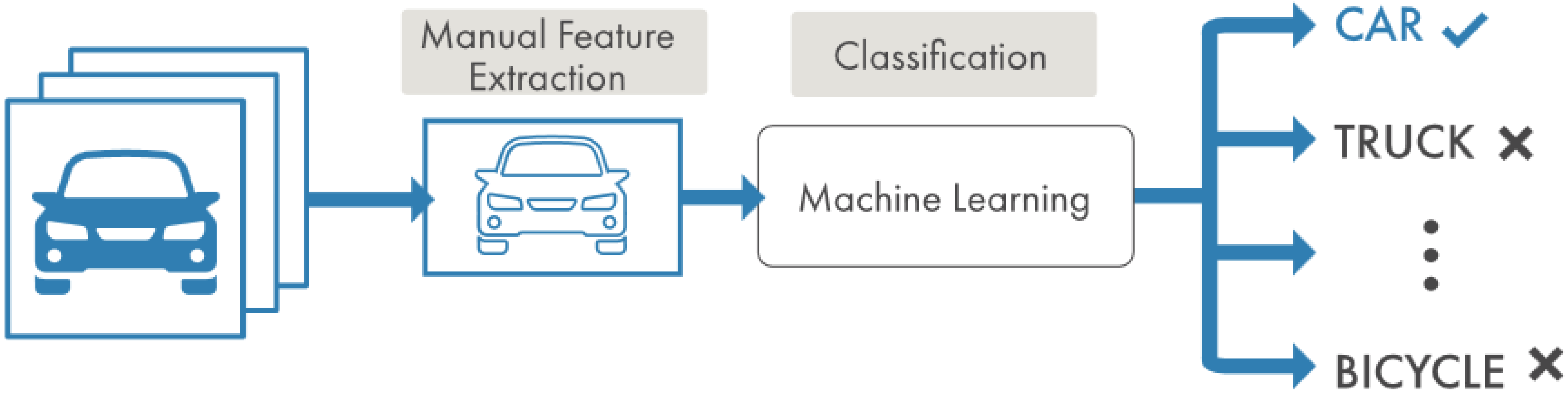
Neural Networks

Hidden Markov Model

What is Machine Learning ?

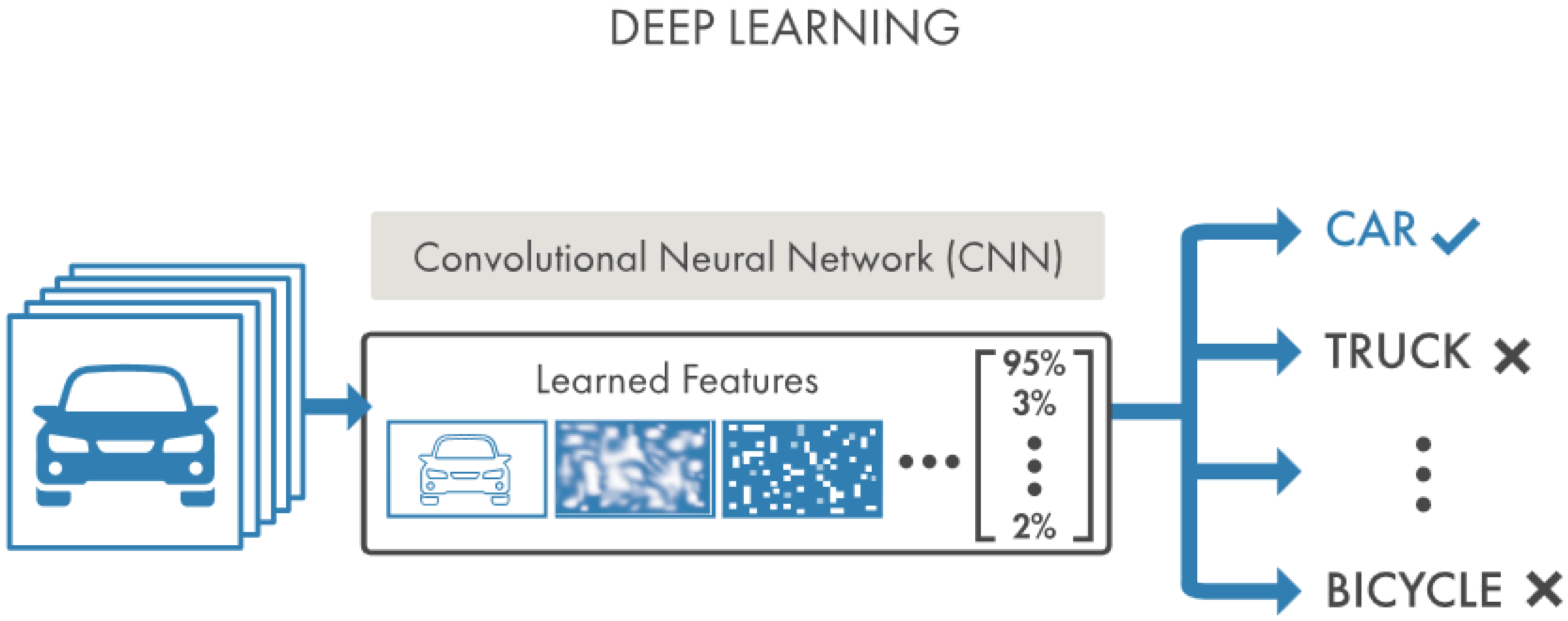
Machine learning uses data and produces a program to perform a task

MACHINE LEARNING



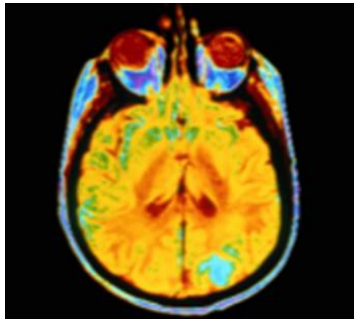
What is Deep Learning ?

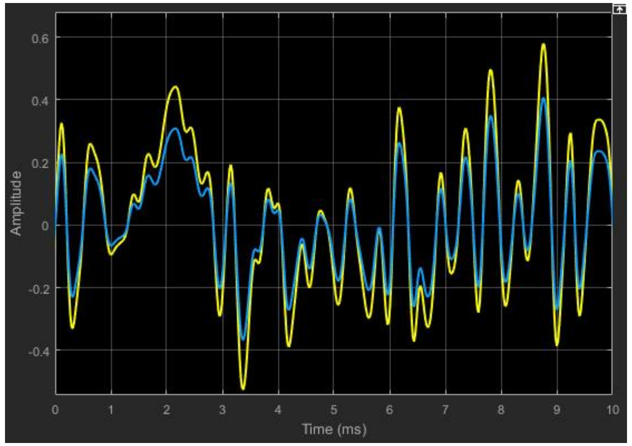
Deep learning performs end-end learning by learning features, representations and tasks directly from images, text and sound



Deep Learning is Ubiquitous

- Computer Vision
- Signal Processing
- Robotics & Controls
- …





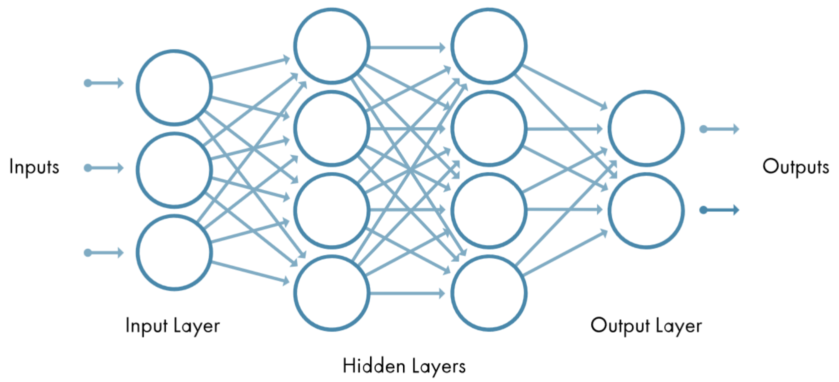
Why is Deep Learning so Popular ?

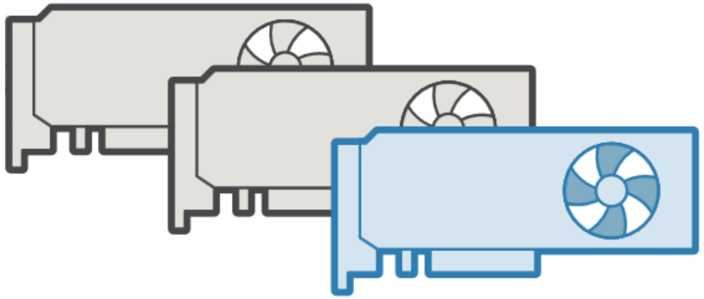
- **Results:**
  - – **95% + accuracy**
    - on ImageNet 1000 class challenge
- **Computing Power:**
  - – **GPU's**
  - – **advances to processor technologies**
  - ⇨ **possible to train networks on massive sets of data**
- **Data:**
  - – **availability of storage**
  - – **access to large sets of labeled data**

Year	Error Rate
Pre-2012 (traditional computer vision and machine learning techniques)	> 25%
2012 (Deep Learning)	~ 15%
2015 (Deep Learning)	<5 %

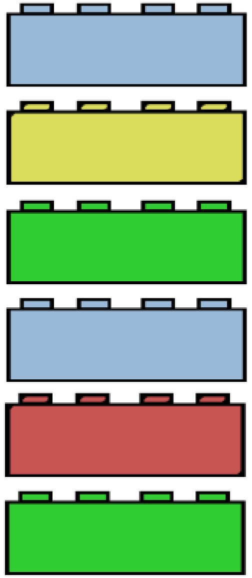
MATLAB for Deep Learning

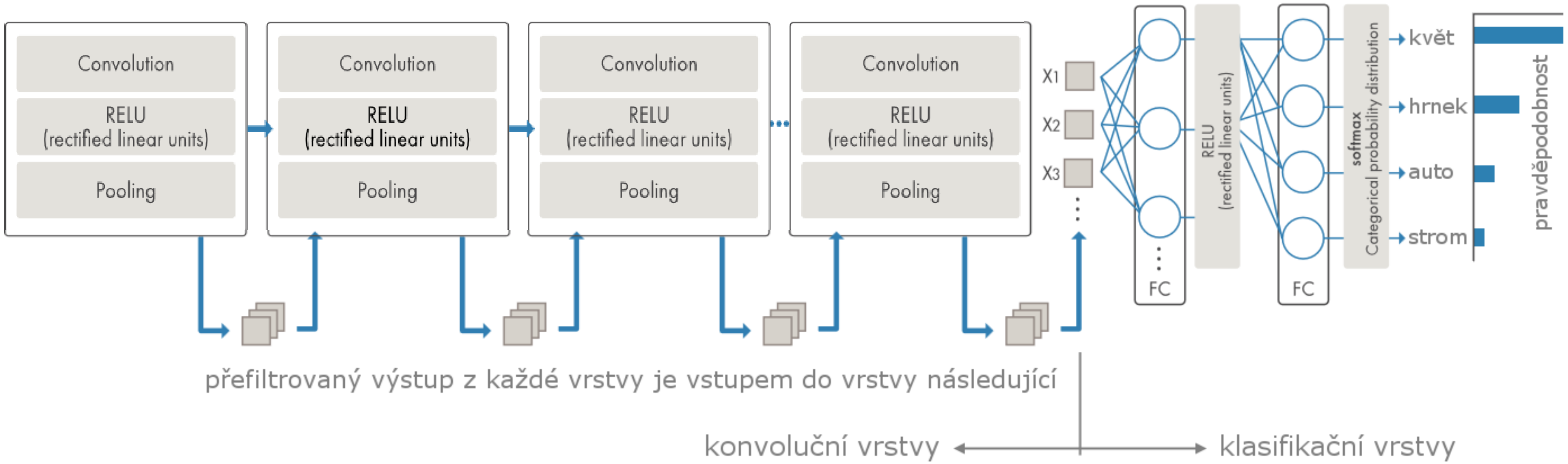
- Network Architectures and Algorithms
- Training and Visualization
- Access the Latest Pretrained Models
- Scaling and Acceleration
- Handling Large Sets of Images
- Object Detection
- Semantic Segmentation
- Ground-Truth Labeling
- Embedded Deployment



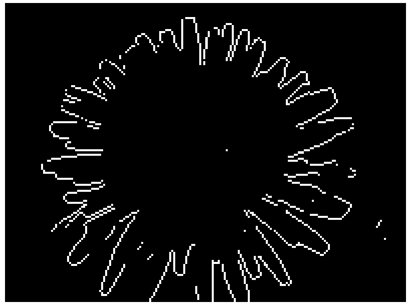


Convolutional Neural Networks (CNN)





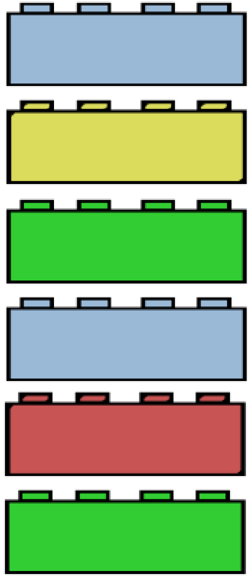
What do filters do?

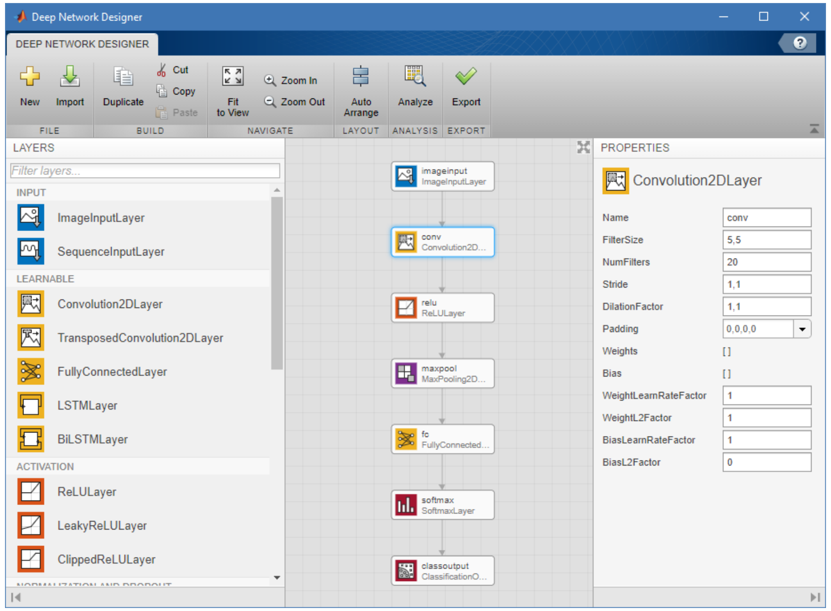


Great for classification:

- Convolution Layer
- ReLU Layer
- Max Pooling Layer

CNN in MATLAB





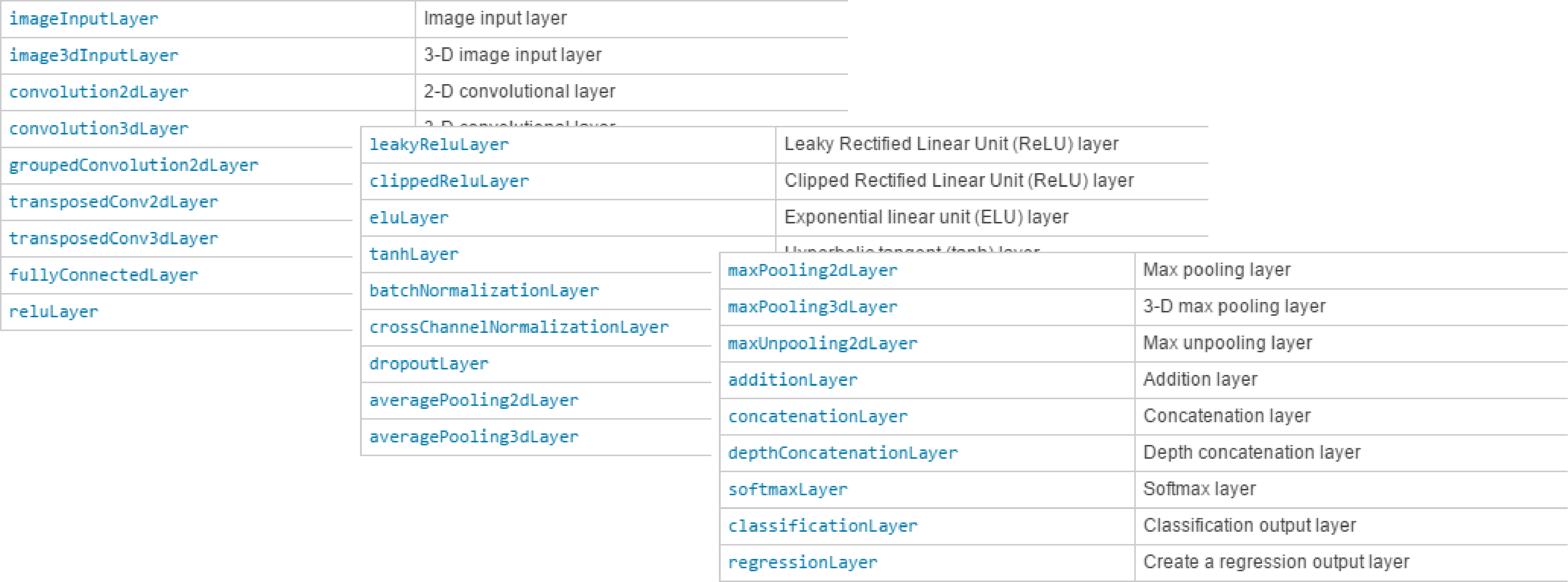
Convolution2DLayer

Name	conv
FilterSize	5,5
NumFilters	20
Stride	1,1
DilationFactor	1,1
Padding	0,0,0,0
Weights	[]
Bias	[]
WeightLearnRateFactor	1
WeightL2Factor	1
BiasLearnRateFactor	1
BiasL2Factor	0

```
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(5,20)
          reluLayer()
          maxPooling2dLayer(2,'Stride',2)
          fullyConnectedLayer(10)
          softmaxLayer()
          classificationLayer()];
```

```
options = trainingOptions('sgdm');
convnet = trainNetwork(trainingData,layers,options);
results = classify(convnet,newData);
```

>30 Layers



imageInputLayer	Image input layer
image3dInputLayer	3-D image input layer
convolution2dLayer	2-D convolutional layer
convolution3dLayer	3-D convolutional layer
groupedConvolution2dLayer	
transposedConv2dLayer	
transposedConv3dLayer	
fullyConnectedLayer	
reluLayer	

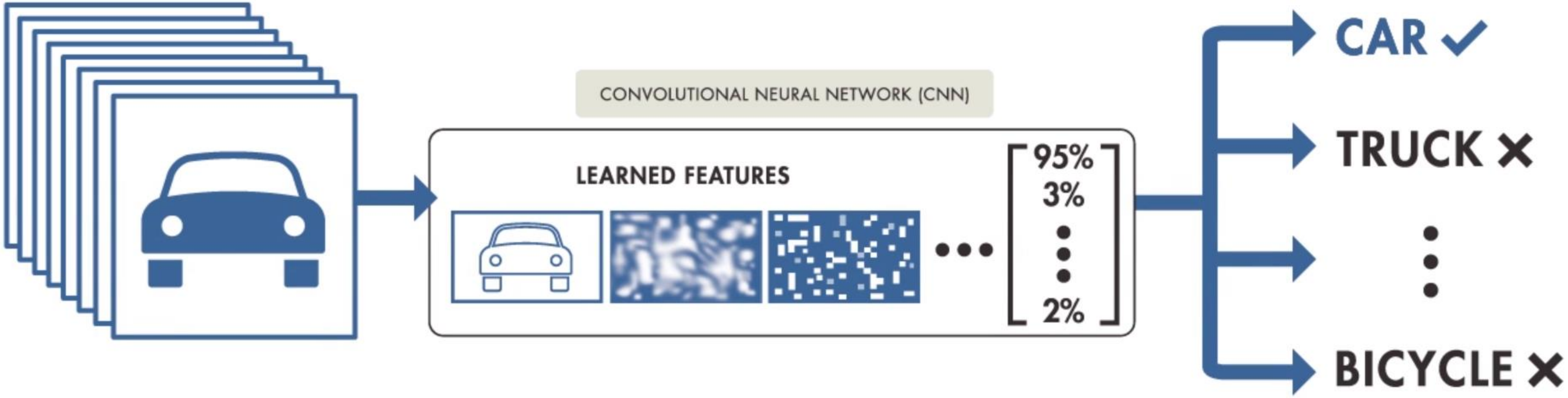
leakyReluLayer	Leaky Rectified Linear Unit (ReLU) layer
clippedReluLayer	Clipped Rectified Linear Unit (ReLU) layer
eluLayer	Exponential linear unit (ELU) layer
tanhLayer	Hyperbolic tangent (tanh) layer
batchNormalizationLayer	
crossChannelNormalizationLayer	
dropoutLayer	
averagePooling2dLayer	
averagePooling3dLayer	

maxPooling2dLayer	Max pooling layer
maxPooling3dLayer	3-D max pooling layer
maxUnpooling2dLayer	Max unpooling layer
additionLayer	Addition layer
concatenationLayer	Concatenation layer
depthConcatenationLayer	Depth concatenation layer
softmaxLayer	Softmax layer
classificationLayer	Classification output layer
regressionLayer	Create a regression output layer

- **Author custom layers in MATLAB using the Custom Layer API**
  - **including automatic differentiation**

2 Approaches for Deep Learning

- **Approach 1: Train a Deep Neural Network from Scratch**

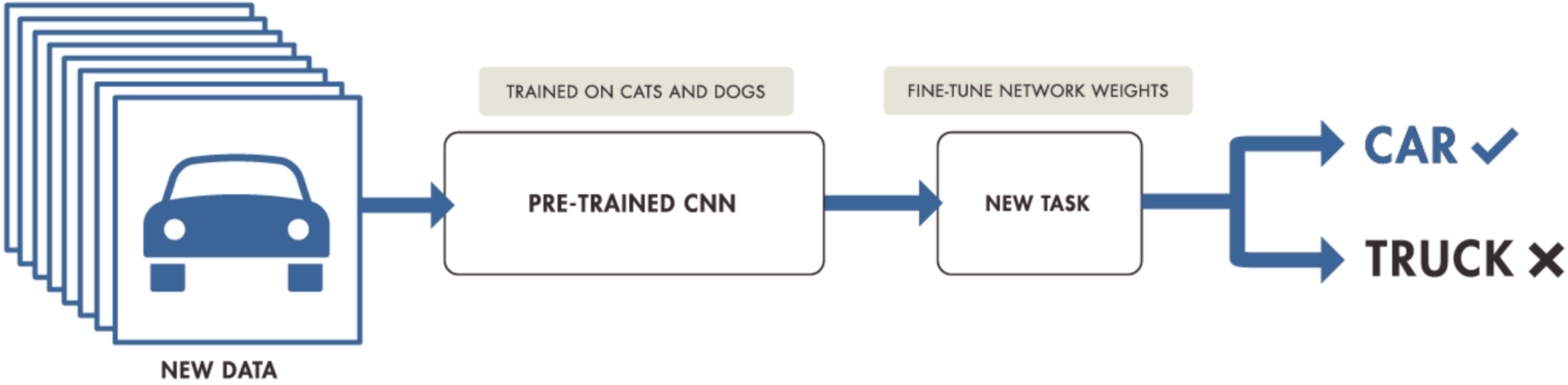


Recommended only when:

Training data	1000s to millions of labeled images
Computation	Compute intensive
Training Time	Days to Weeks for real problems
Model accuracy	High (can overfit to small datasets)

2 Approaches for Deep Learning

- **Approach 2: Fine-tune a pre-trained model (transfer learning)**

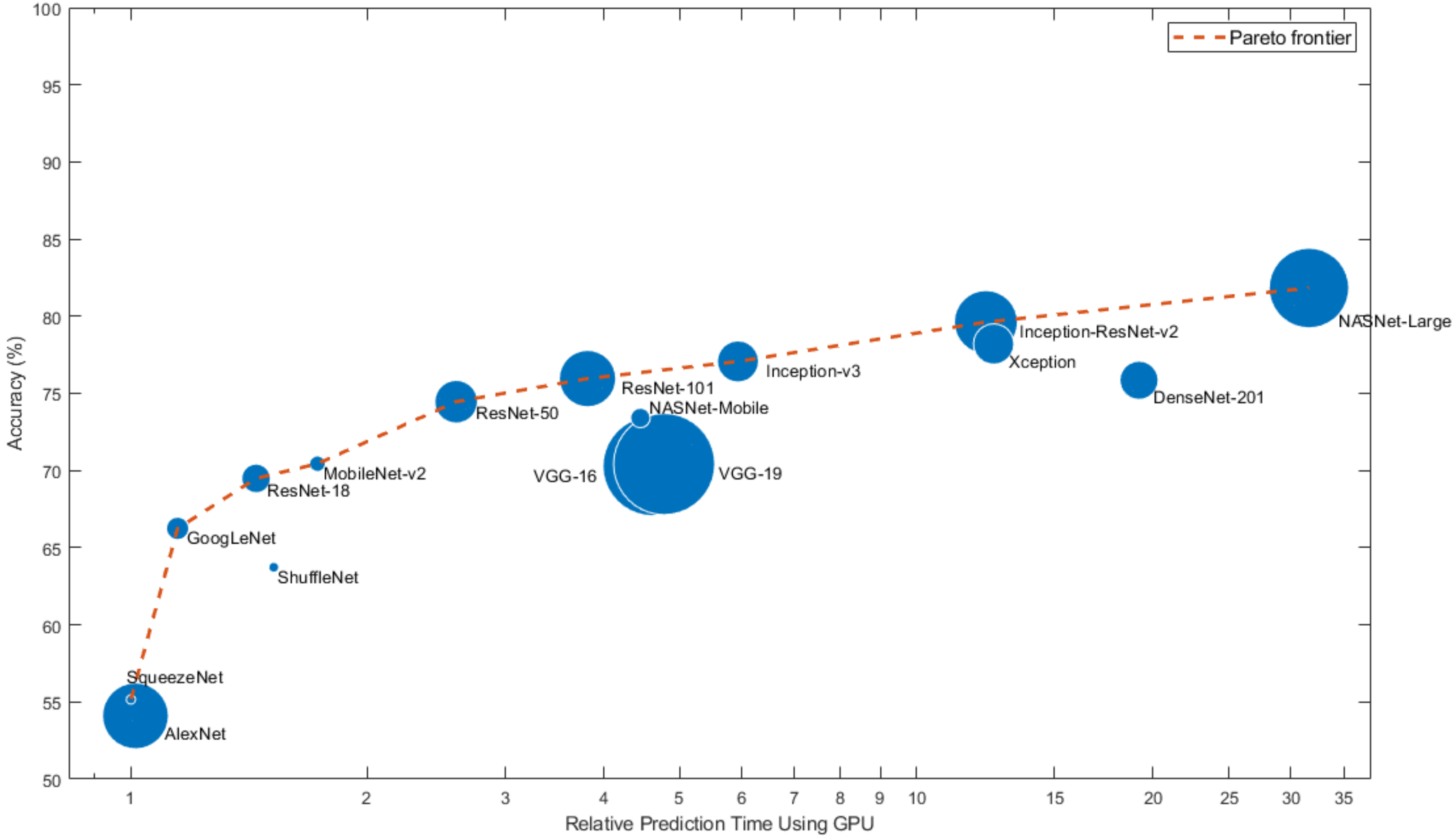


Recommended when:

Training data	100s to 1000s of labeled images (small)
Computation	Moderate computation
Training Time	Seconds to minutes
Model accuracy	Good, depends on the pre-trained CNN model

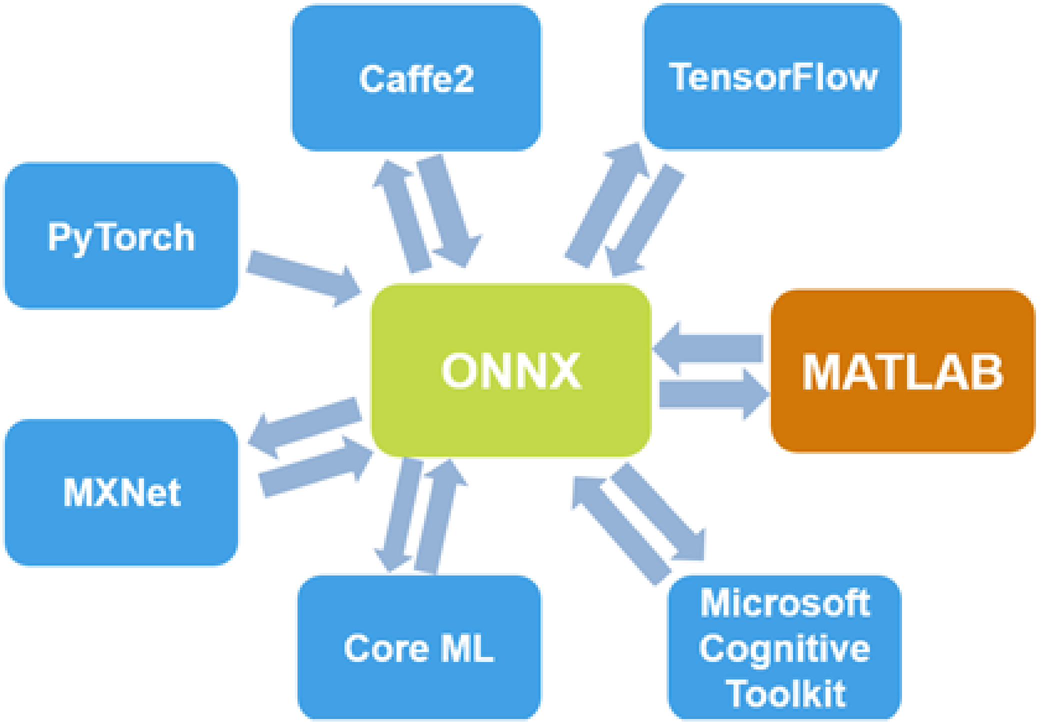
Transfer Learning using Pre-Trained Networks

- **Pre-Trained Networks**
  - AlexNet
  - VGG-16 and VGG-19
  - GoogLeNet
  - ResNet-50 and ResNet-101
  - Inception-v3
  - Inception-ResNet-v2
  - SqueezeNet
  - and more …



Transfer Learning using Pre-Trained Networks

- **Pre-Trained Networks**
  - AlexNet
  - VGG-16 and VGG-19
  - GoogLeNet
  - ResNet-50 and ResNet-101
  - Inception-v3
  - Inception-ResNet-v2
  - SqueezeNet
  - and more …

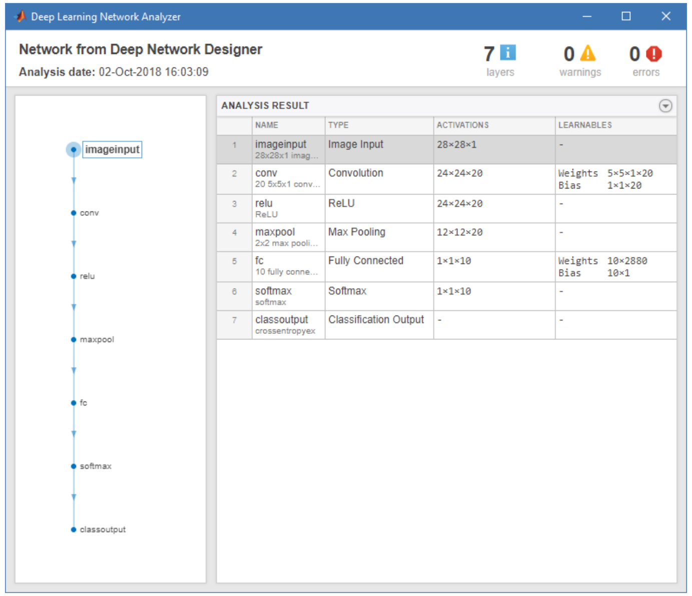


- **ONNX Model Converter**

Training, Validation and Visualization

- **Network Analyzer (`analyzeNetwork`)**
  - find problems in network architectures before training

- **Monitor training progress**
  - plots for accuracy, loss, validation metrics, and more
- **Automatically validate network performance**
  - stop training when the validation metrics stop improving
- **Perform hyperparameter tuning**
  - using Bayesian optimization



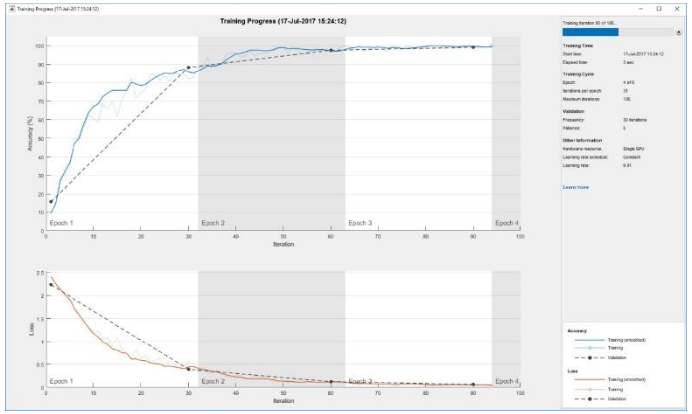
Network from Deep Network Designer

Analysis date: 02-Oct-2018 16:03:09

7 layers 0 warnings 0 errors

ANALYSIS RESULT

	NAME	TYPE	ACTIVATIONS	LEARNABLES
1	imageinput 28x28x1 imag...	Image Input	28×28×1	-
2	conv 20 5x5x1 conv...	Convolution	24×24×20	Weights 5×5×1×20 Bias 1×1×20
3	relu ReLU	ReLU	24×24×20	-
4	maxpool 2x2 max pooli...	Max Pooling	12×12×20	-
5	fc 10 fully conne...	Fully Connected	1×1×10	Weights 10×2880 Bias 10×1
6	softmax softmax	Softmax	1×1×10	-
7	classoutput crossentropyex	Classification Output	-	-



Training iteration 95 of 186...

Training Time
Start time: 17-Jul-2017 15:24:12
Elapsed time: 3 sec

Training Cycle
Epoch: 4 of 6
Iterations per epoch: 31
Maximum iterations: 186

Validation
Frequency: 30 iterations
Patience: 5

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01

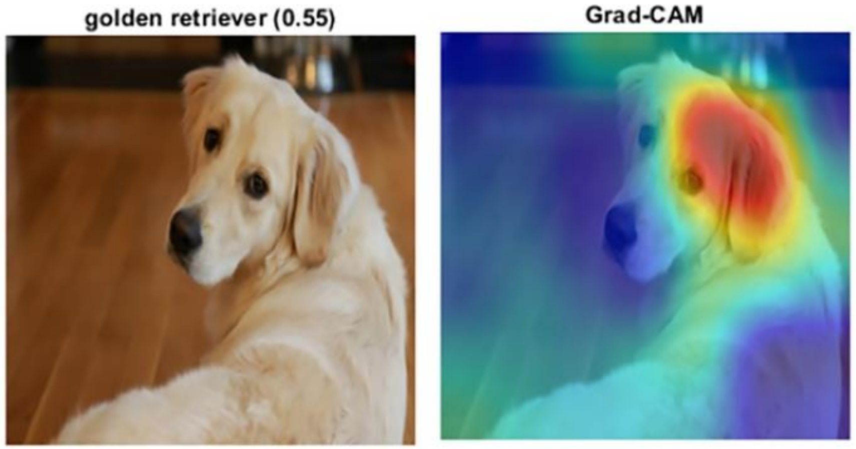
Learn more

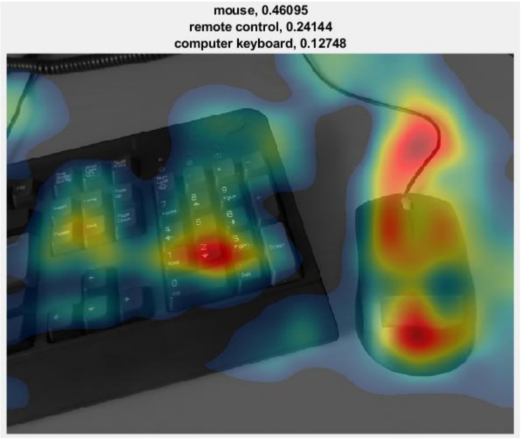
Debugging and Visualization

- Visualize activations and filters from intermediate layers

- CAM (Class Activation Mapping)
- Grad-CAM
- Occlusion sensitivity maps

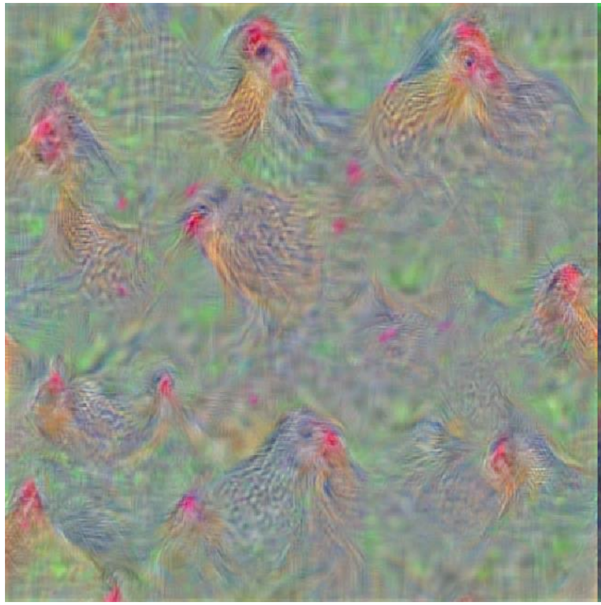
- Deep Dream visualization



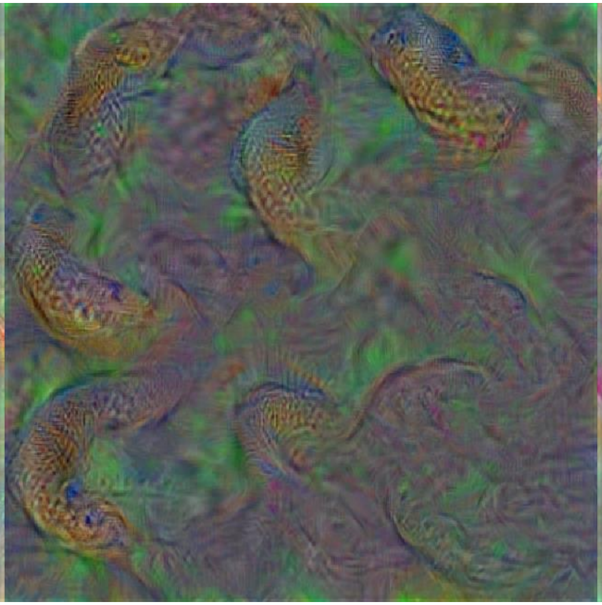


Deep Dream Images Using AlexNet

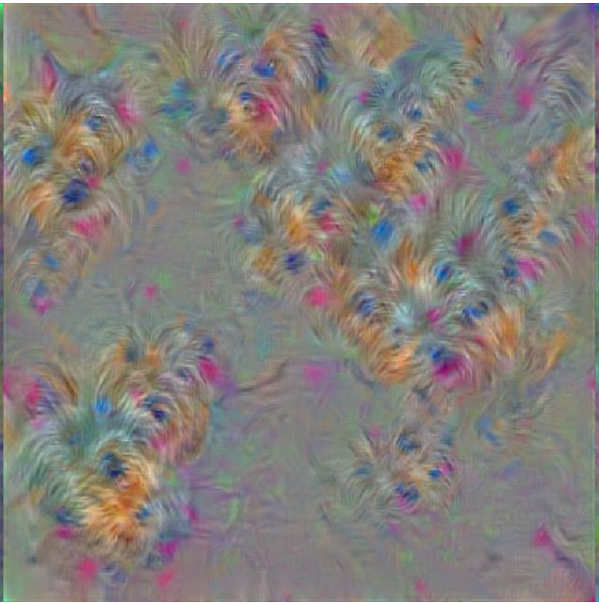
Hen



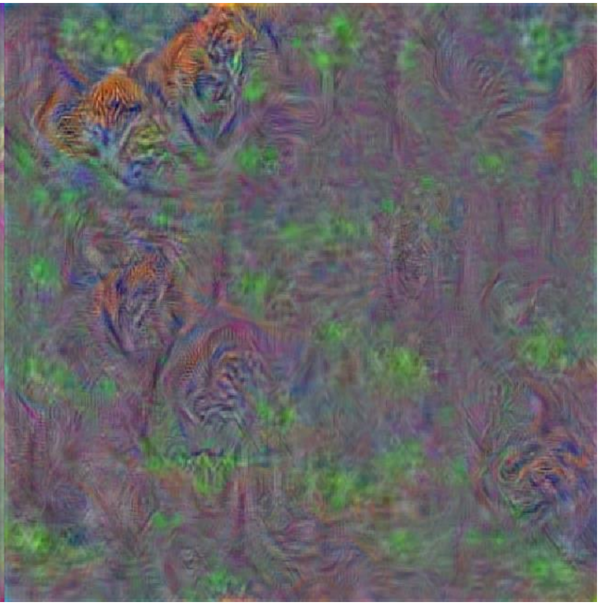
Indian cobra



Yorkshire terrier



Tiger



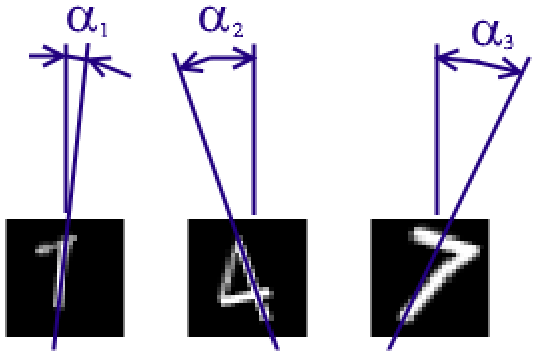
Handling Large Sets of Images

- Use `imageDataStore`
  - easily read and process large sets of images
- Access data stored in
  - local files
  - networked storage
  - databases
  - big data file systems
- Efficiently resize and augment image data
  - increase the size of training datasets
  - `imageDataAugmenter`, `augmentedImageDatastore`

Deep Learning Models for Regression

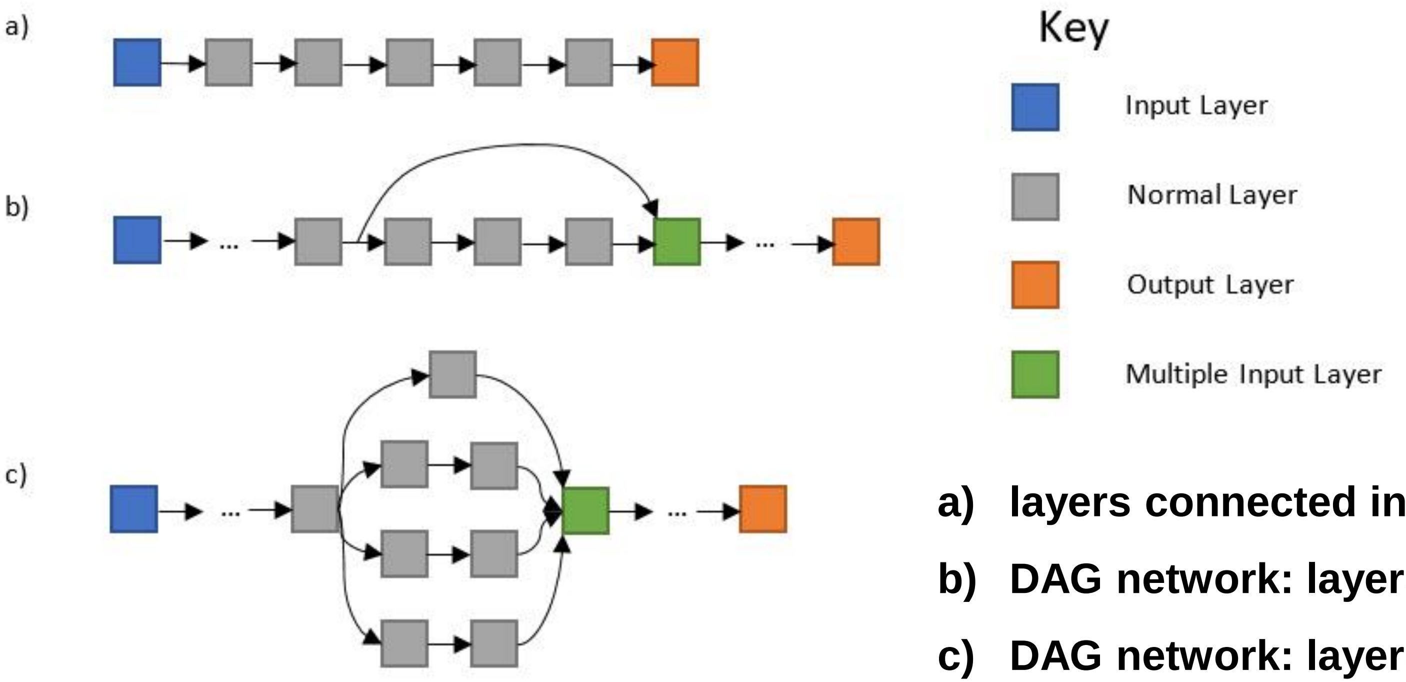
- To predict continuous data such as angles and distances in images
- Include a regression layer at the end of the network

```
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(12,25)
          reluLayer()
          fullyConnectedLayer(1)
          regressionLayer()];
options = trainingOptions('sgdm');
convnet = trainNetwork(trainImages,trainAngles,layers,options);
results = predict(convnet,newImages);
```



Directed Acyclic Graphs (DAG) Networks

- **Represent complex architectures**
  - `layerGraph`, `plot`, `addLayers`, `removeLayers`, `connectLayers`, `disconnectLayers`
- **Addition layer, Depth concatenation layer**



a) **layers connected in series**

b) **DAG network: layers are skipped (ResNet)**

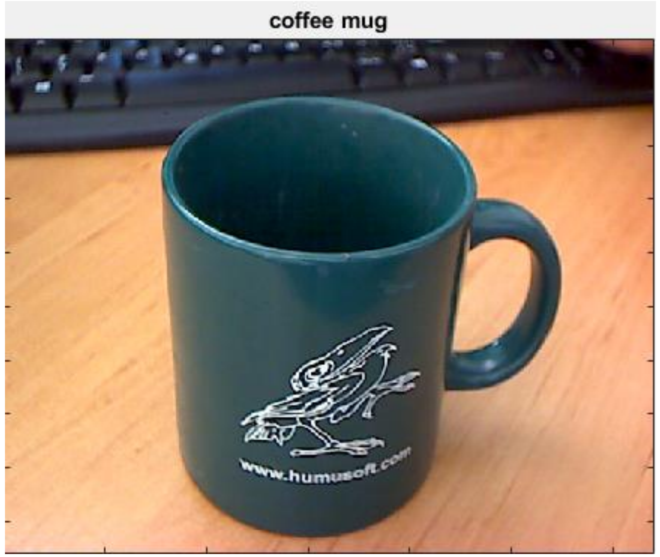
c) **DAG network: layers are connected in parallel (GoogLeNet)**

Customizations

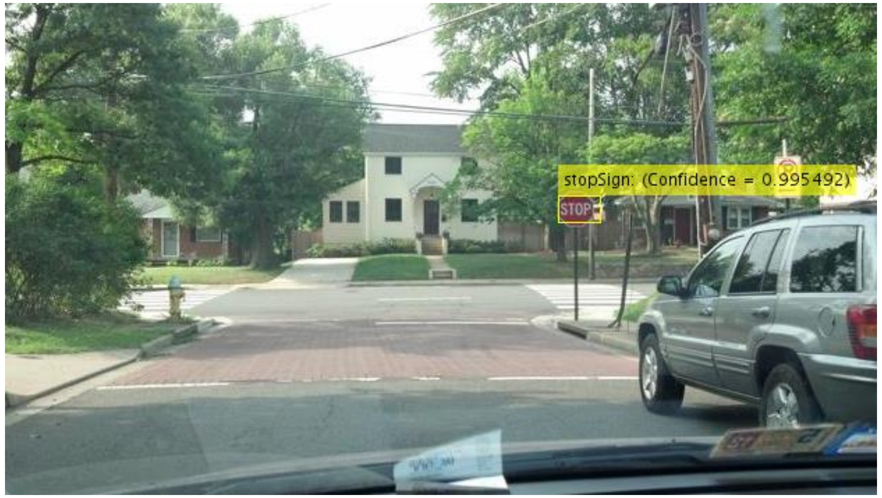
- **Define and train complex networks using**
  - custom training loops
  - automatic differentiation
  - shared weights
  - custom loss functions
- **Custom layers support**
  - define new layers, including layers with multiple inputs and outputs
- **Multi-Input, Multi-Output Networks**
  - create and train networks with multiple inputs and multiple outputs
- **Build advanced network architectures**
  - GANs, Siamese networks, attention networks, ...

Image Classification vs. Object Detection

- **Image Classification**
  - classify whole image using set of distinct categories
  - object recognition
  - scene recognition



- **Object Detection**
  - recognizing and locating the (small) object in a scene
  - multiple objects in one image

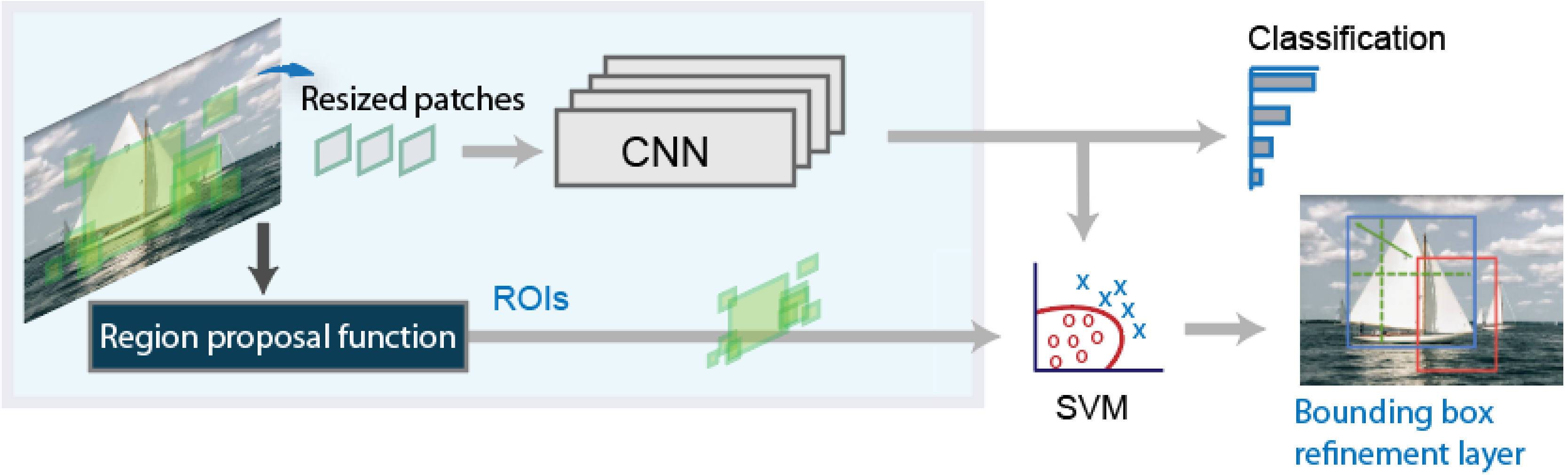


Standard Image Classification and Object Detection Algorithms in MATLAB

- Object detection using extracted features
  - edges, corners, SURF, MSER, HOG, LBP, …
- Template matching
- Bag of features
- Image segmentation and blob analysis
- Viola-Jones algorithm, ACF

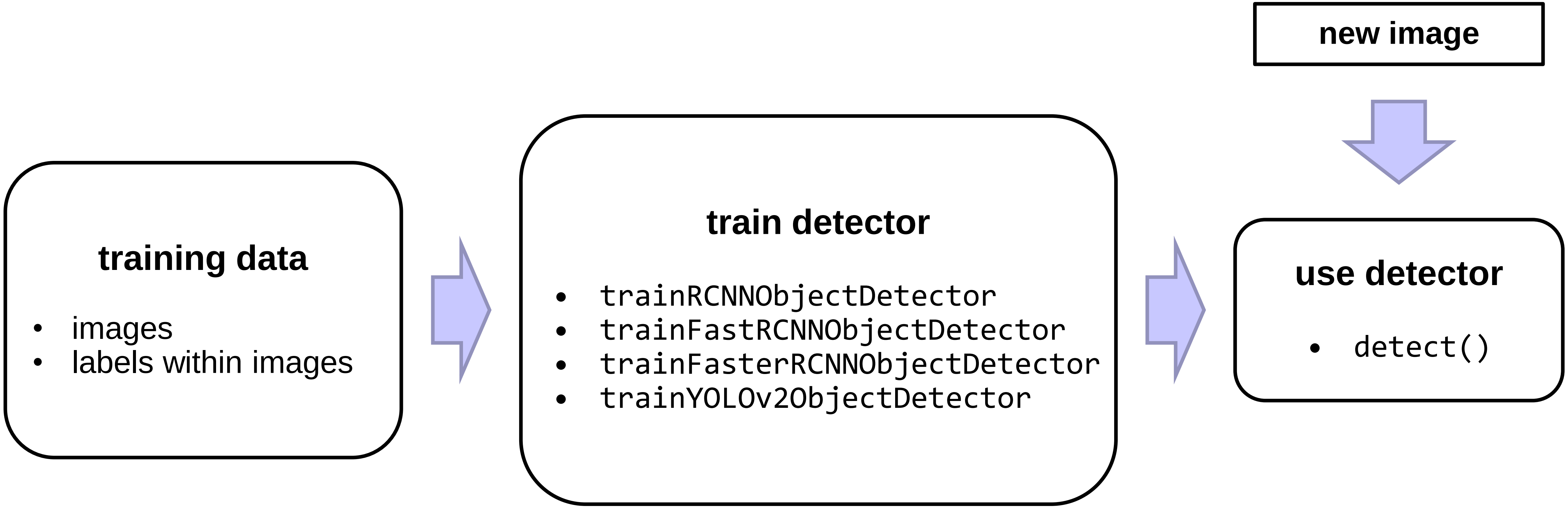
Object Detection using Deep Learning

- **Family of R-CNN object detectors (Regions with Convolutional Neural Networks)**
  - R-CNN, Fast R-CNN, Faster R-CNN
  - uses region proposal to detect objects within images



  - Fast and Faster R-CNN improve detection performance for large number of regions
- **YOLO v2 deep learning object detector (you-only-look-once)**

Object Detection Training Workflow



new image

training data

- images
- labels within images

train detector

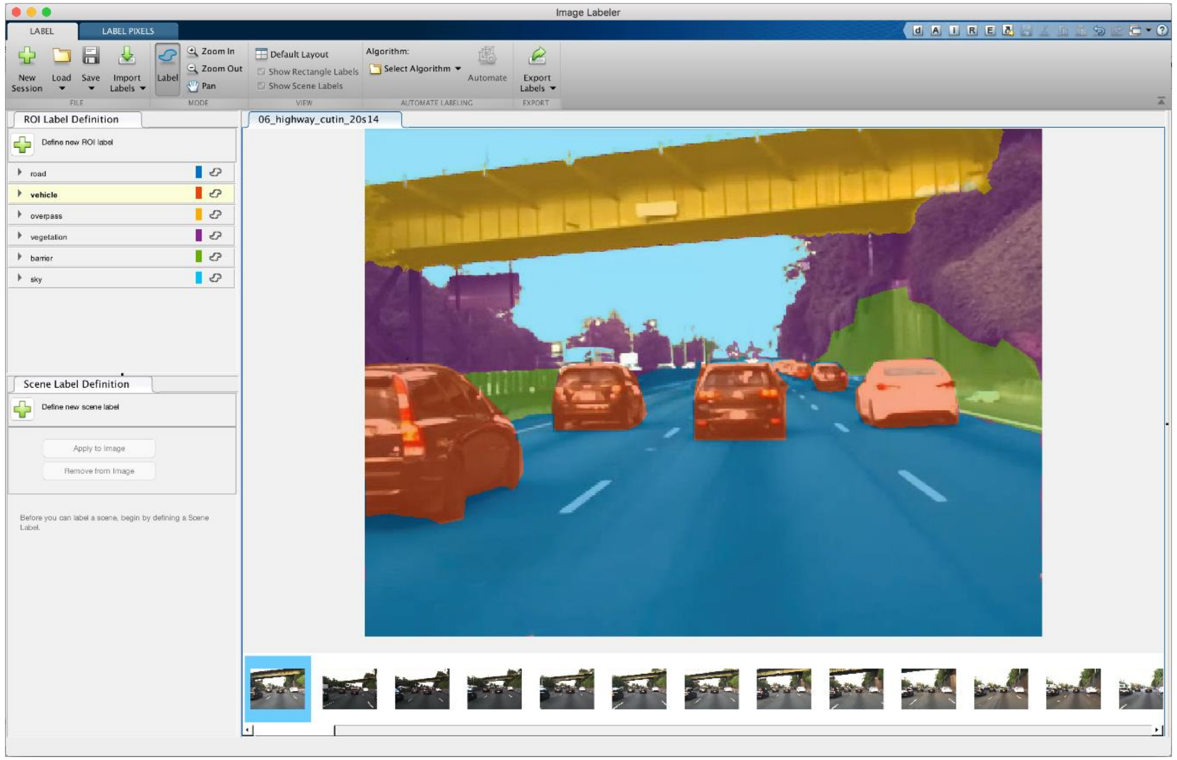
- `trainRCNNObjectDetector`
- `trainFastRCNNObjectDetector`
- `trainFasterRCNNObjectDetector`
- `trainYOLOv2ObjectDetector`

use detector

- `detect()`

Ground-Truth Labeling

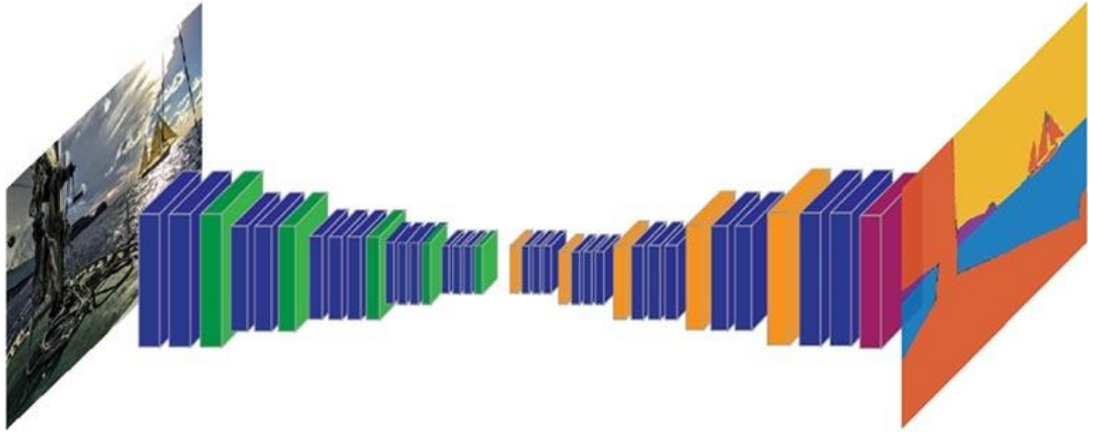
- App to label pixels and regions
  - *ImageLabeler App*
  - for object detection
  - for semantic segmentation
- Automate ground-truth labeling
  - automation API
- Video annotation
  - *VideoLabeler App*

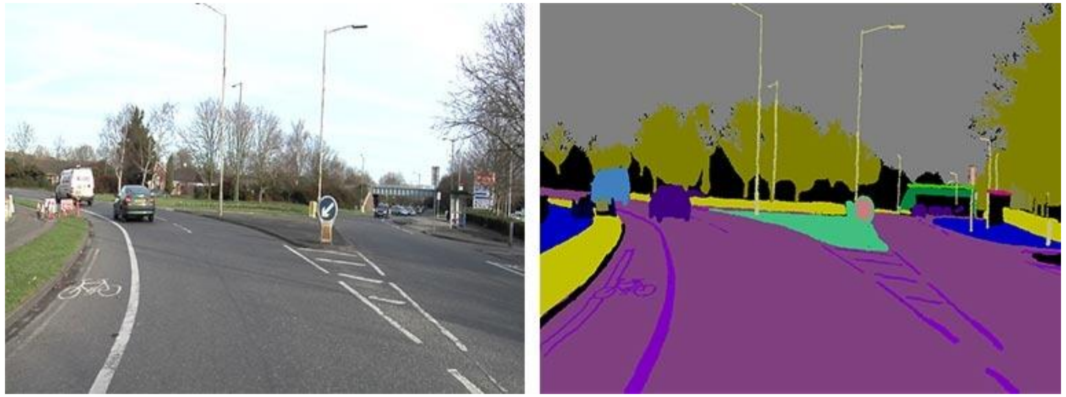


Before you can label a scene, begin by defining a Scene Label.

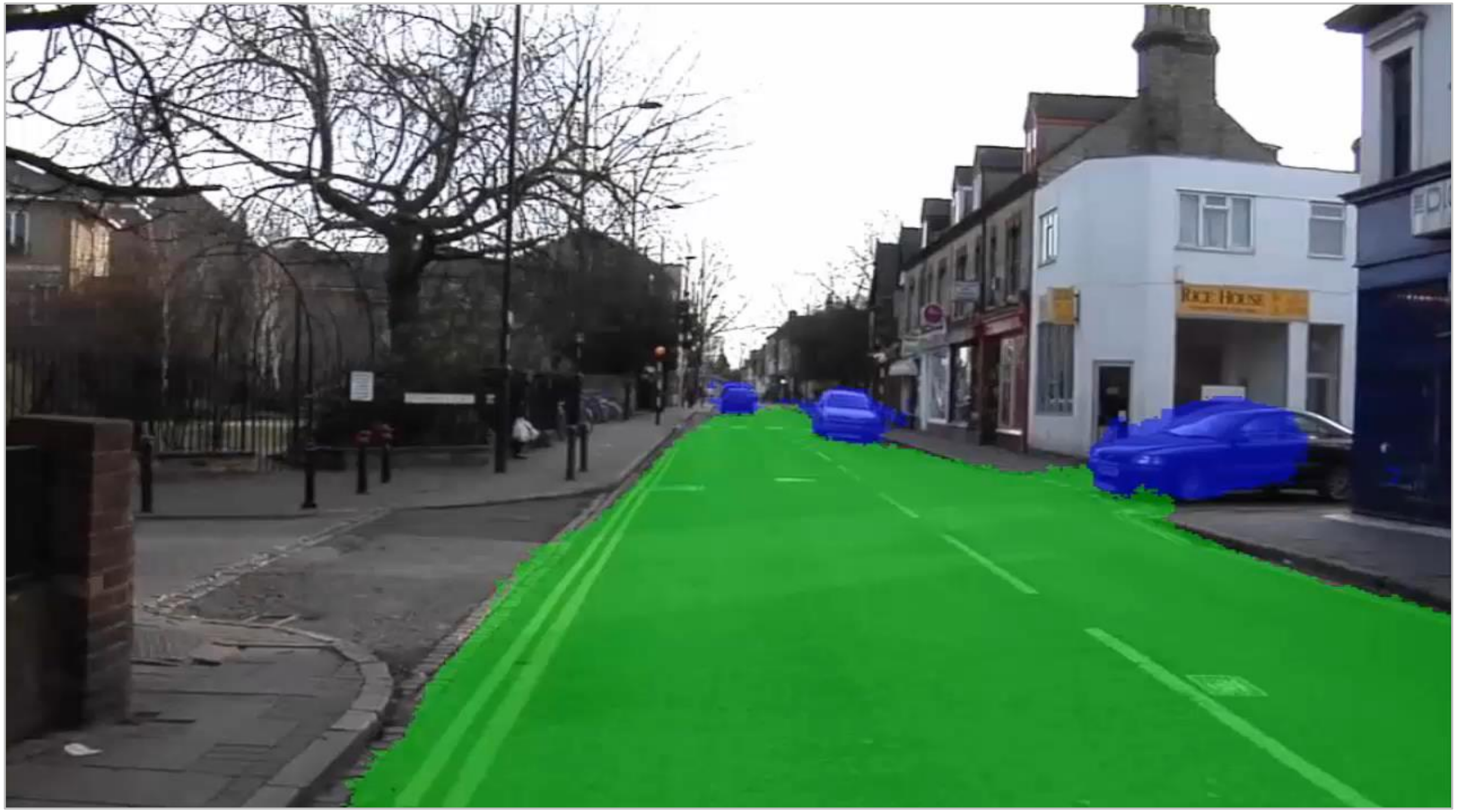
Semantic Segmentation

- **Classify individual pixels**
- **Manage data**
  - `imageDatastore` + `pixelLabelDatastore`
  - `pixelLabelImageDatastore`
- **Perform semantic segmentation**
  - `semanticseg`
- **Special layers**
  - `pixelClassificationLayer`, `crop2dLayer`
- **Complete networks**
  - `segnetLayers`, `fcnLayers`, `unetLayers`



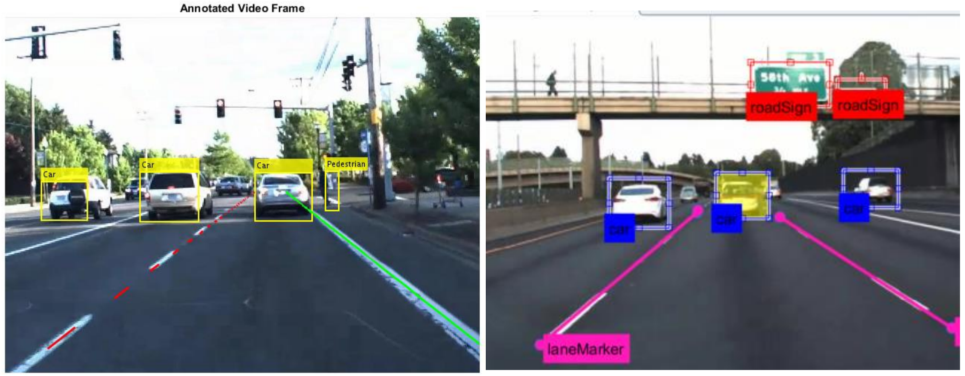


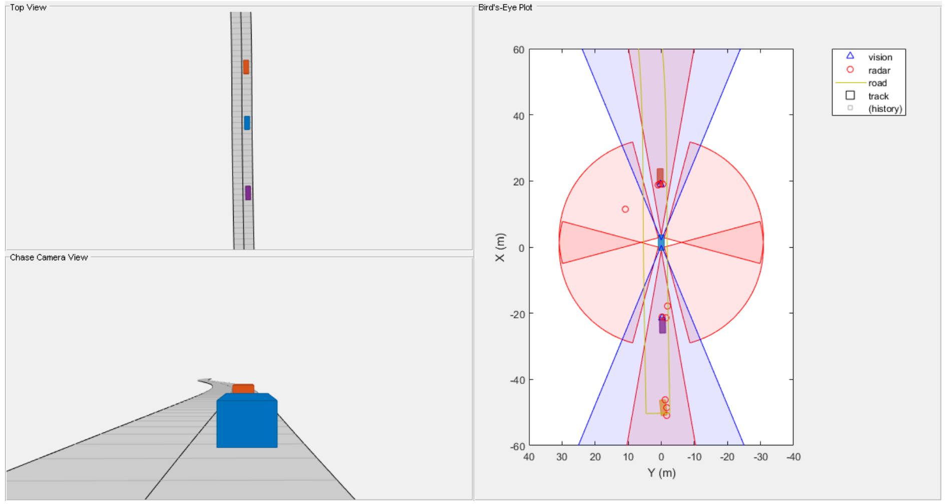
Semantic Segmentation



Automated Driving Toolbox

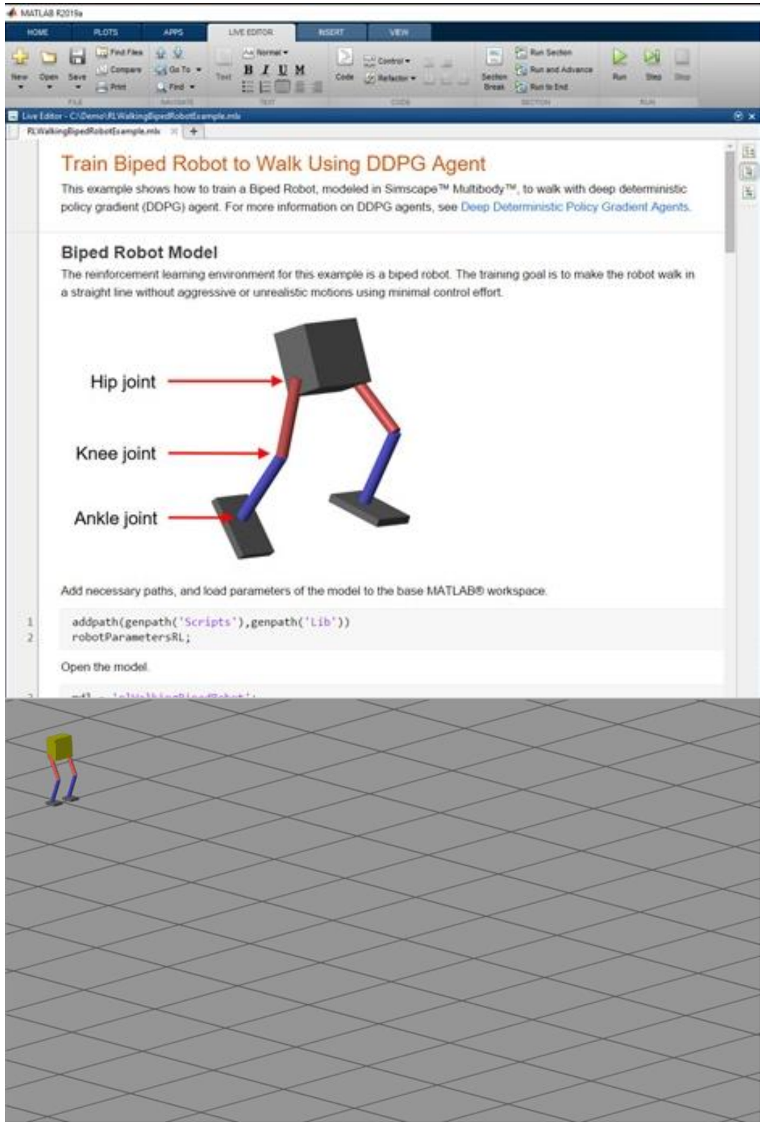
- Design, simulate, and test ADAS and autonomous driving systems
- Object detection
  - – lane marker detection, vehicle detection, …
- Multisensor fusion
  - – vision, radar, ultrasound
- Visualization
  - – annotation, bird’s-eye-view, point cloud
- Scenario Generation
  - – synthetic sensor data for driving scenarios
- Ground-truth labeling
  - – annotating recorded sensor data





Reinforcement Learning Toolbox

- Design and train policies using reinforcement learning
- Use these policies to implement
  - controllers
  - decision-making algorithms
- For complex systems, such as
  - robots
  - autonomous systems, …
- Implement the policies
  - deep neural networks, polynomials, look-up tables
- Environment modeling
  - leverage MATLAB and Simulink models for training


MATLAB R2019a
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
Live Editor - C:\Demo\RLWalkingBipedRobotExample.mlx
RLWalkingBipedRobotExample.mlx
Hip joint
Knee joint
Ankle joint

Train Biped Robot to Walk Using DDPG Agent

This example shows how to train a Biped Robot, modeled in Simscape™ Multibody™, to walk with deep deterministic policy gradient (DDPG) agent. For more information on DDPG agents, see Deep Deterministic Policy Gradient Agents.

Biped Robot Model

The reinforcement learning environment for this example is a biped robot. The training goal is to make the robot walk in a straight line without aggressive or unrealistic motions using minimal control effort.

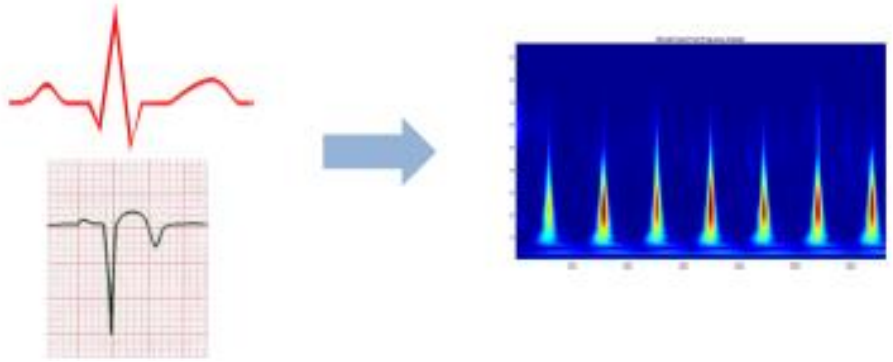
Add necessary paths, and load parameters of the model to the base MATLAB® workspace.

```
addpath(genpath('Scripts'),genpath('Lib'))
robotParametersRL;
```

Open the model.

Challenges with Signals

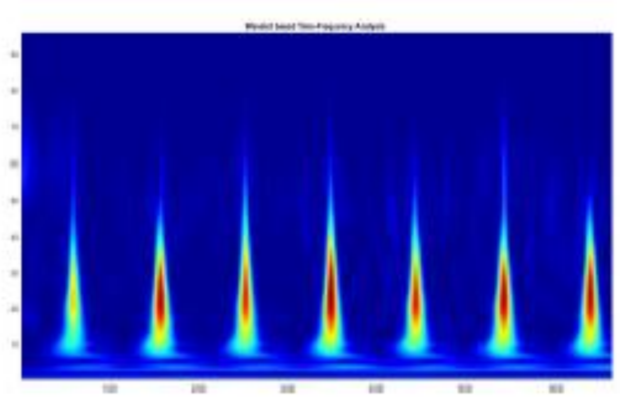
- **Enhancing the subtle information present in signals**
  - signals belonging to different classes can have similar properties
- **Represent signal features occurring at different scales**
  - good time-frequency localization
- **Need for independent representation of signals**
  - signal features within same class can have different amplitudes or polarities etc.



Using Time-Frequency Representations

- A time-frequency representation captures how spectral content of signal evolves over time
  - can be saved as an image

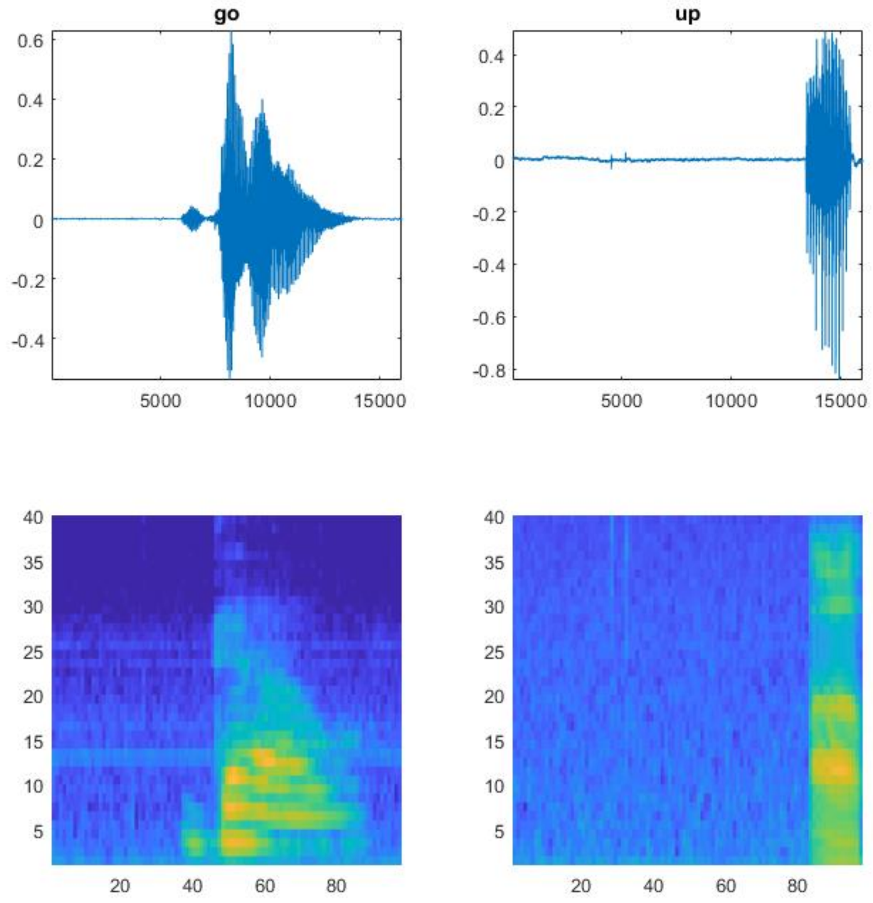
- Many time-frequency representations are available
  - spectrogram
  - scalogram (continuous wavelet transform)
  - constant Q transform etc.



- Generate time-frequency representations of signals with two lines of MATLAB code

Deep Learning with Time Series Workflow

1. **Create time-frequency representation of the signal data**
   - ***Signal Analyzer* app**
   - **spectrogram**
     - `spectrogram`, `pspectrum`
   - **scalogram (continuous wavelet transform)**
     - `cwt`
2. **Capture time-frequency images**
3. **Apply deep neural network to the images**

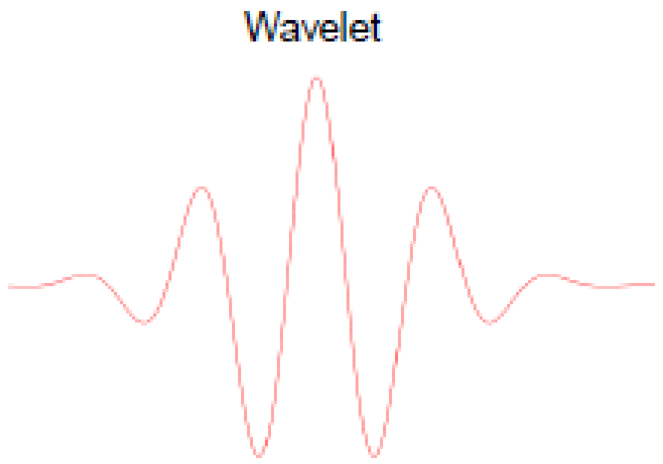


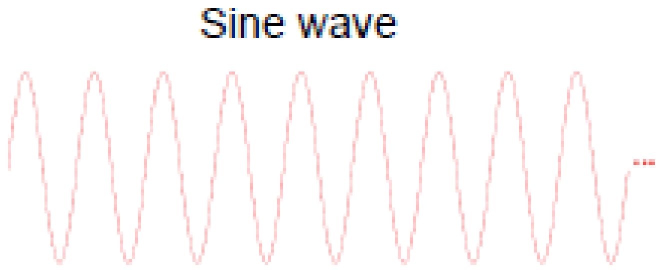
What is a Wavelet

- **A wavelet is a rapidly decaying wave like oscillation with zero mean**

- **Wavelets are best suited to localize frequency content in real world signals**

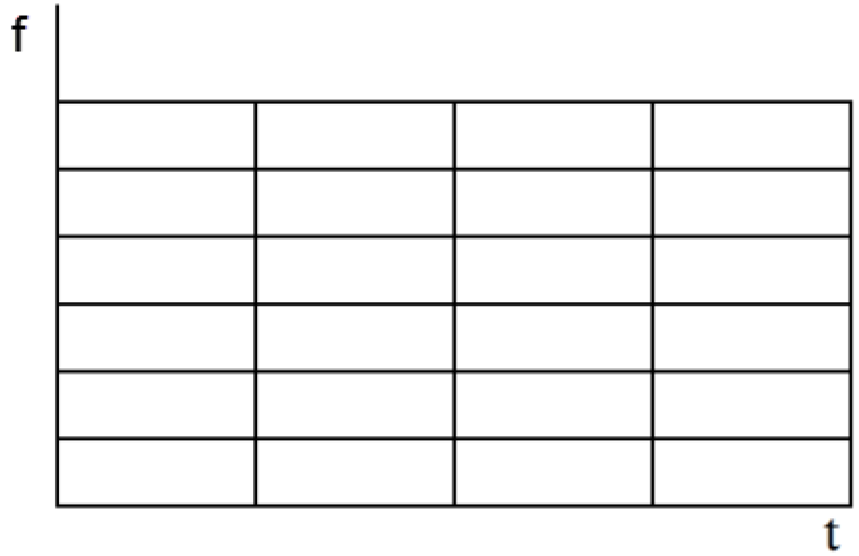
- **MATLAB makes it easy by providing default wavelets**


Wavelet

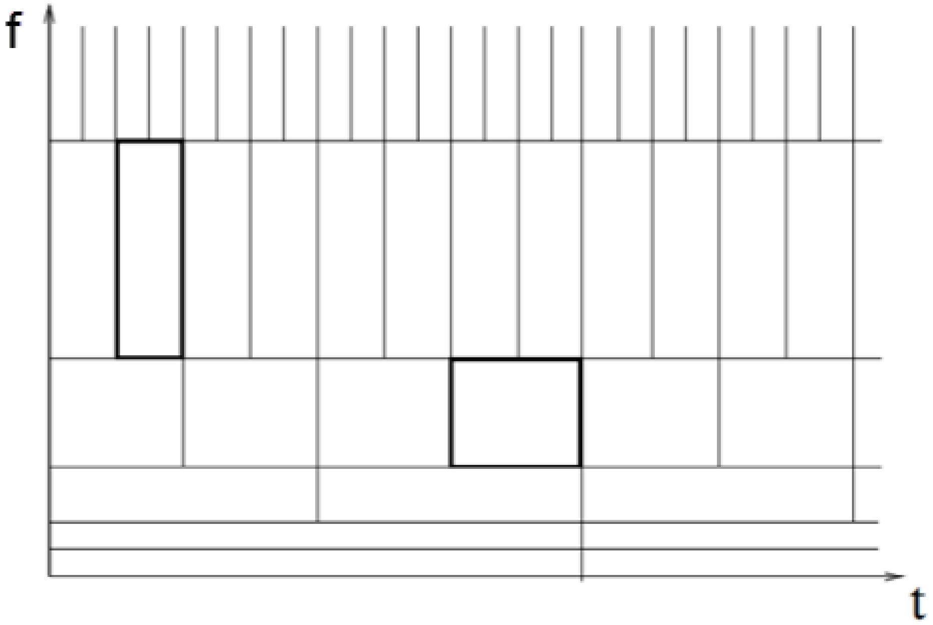

Sine wave

Time-Frequency Analysis – Comparison

- **Short Time Fourier Transform**
- **Continuous Wavelet Transform**



Fixed window size limits the resolution

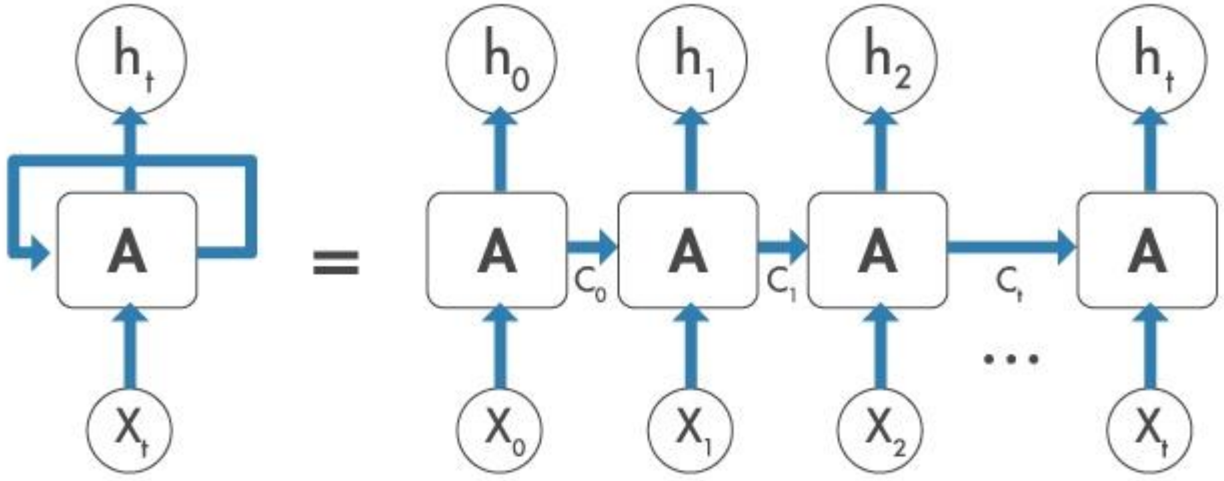


Wavelets – well localized time and frequency

Variable sized windows capture features at different scales simultaneously

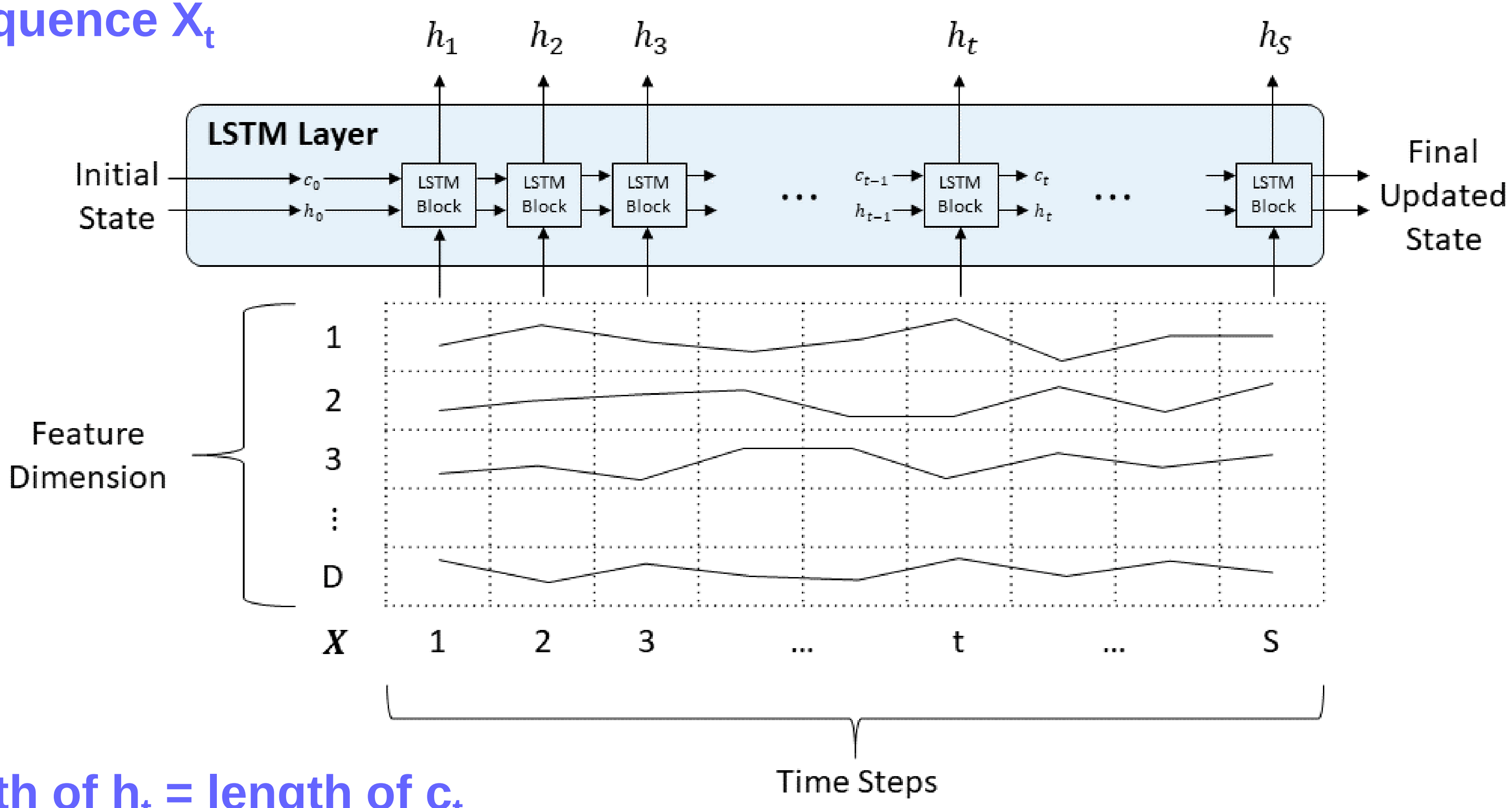
Long Short Term Memory (LSTM) Networks

- **LSTM layer is recurrent neural network (RNN) layer**
  - **learn long-term dependencies between the time steps of sequence data**
- **Prediction and classification on time-series, text, and signal data**



LSTM Layer

- At time step *t*, the block takes:
  - current state of the network (c_{t-1},h_{t-1})
  - next time step of the sequence X_t
- Then computes:
  - the output h_t
  - updated cell state c_t



- LSTM layer parameter
  - numHiddenUnits = length of h_t = length of c_t

LSTM in MATLAB

```
layers = [sequenceInputLayer(num_channels)
                                              'OutputMode','sequence'
          lstmLayer(num_HiddenUnits,      <                             )
                                              'OutputMode','last'
          fullyConnectedLayer(num_classes)
          softmaxLayer()
          classificationLayer()];
options = trainingOptions('adam');
lstmnet = trainNetwork(trainingSequences,Y,layers,options);
results = classify(lstmnet,newSequences);
```

Multi-Platform Deployment

- Deploy deep learning models anywhere
  - CUDA
  - C code
  - enterprise systems
  - or the cloud
- Generate code that leverages optimized libraries
  - Intel® (MKL-DNN)
  - NVIDIA (TensorRT, cuDNN)
  - ARM® (ARM Compute Library)

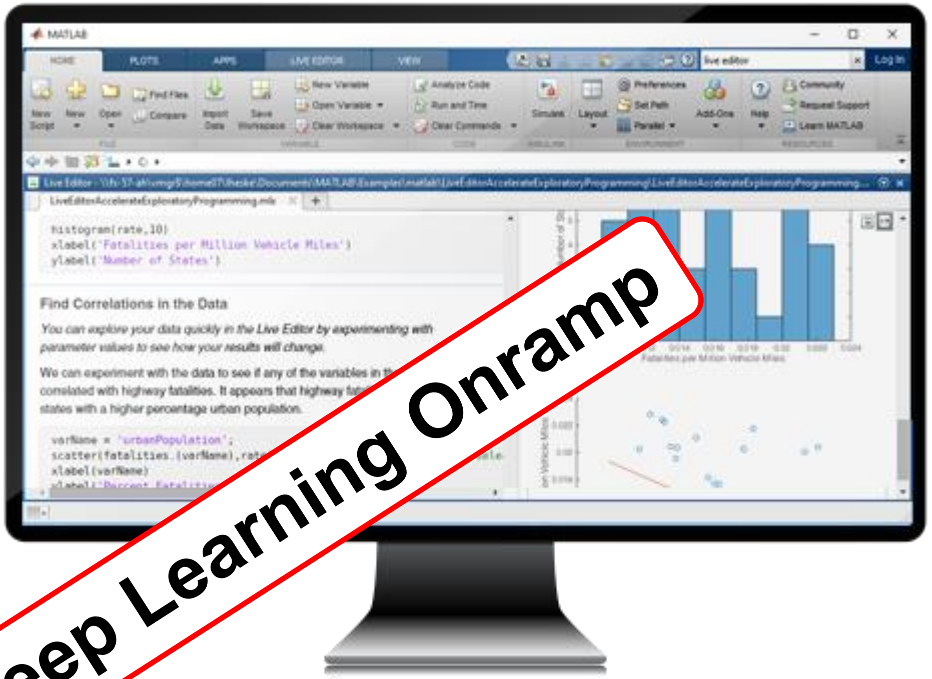
⇨ deployable models with high-performance inference speed.

Latest Features

- **What's New in MATLAB for Deep Learning?**
  - https://www.mathworks.com/solutions/deep-learning/features.html

Jak začít s prostředím MATLAB?

- **Zkušební verze:**
  - plnohodnotná verze MATLAB
  - časově omezena na 30 dní
  - možnost libovolných nadstaveb
  - v případě zájmu využijte kontaktní formulář



```
histogram(rate,10)
xlabel('Fatalities per Million Vehicle Miles')
ylabel('Number of States')
```

Find Correlations in the Data

You can explore your data quickly in the Live Editor by experimenting with parameter values to see how your results will change.

We can experiment with the data to see if any of the variables in th… correlated with highway fatalities. It appears that highway fat… states with a higher percentage urban population.

```
varName = 'urbanPopulation';
scatter(fatalities.(varName),rate
xlabel(varName)
```

+ Deep Learning Onramp

http://www.humusoft.cz/matlab/trial/

- **MATLAB Onramp:**
  - on-line kurz zdarma
  - časová náročnost: 2 hodiny
  - přihlášení: https://matlabacademy.mathworks.com/

Zdroje informací

- Internetové stránky
  - www.humusoft.cz
  - www.mathworks.com
- MATLAB Central
  - mezinárodní komunita příznivců a uživatelů systému MATLAB/Simulink
  - www.mathworks.com/matlabcentral/
- Informační kanály
  - Facebook veřejná skupina MATLAB a Simulink (SK CZ)
  - www.facebook.com/groups/matlab4students/

Zdroje informací

- **Www semináře (webinars)**
  - on-line semináře zdarma (AJ, ČJ, SJ), k dispozici videa z těch, které již proběhly
  - www.humusoft.cz/wwwseminare
- **Workshopy**
  - praktické seznámení s nástroji MATLAB & Simulink a COMSOL Multiphysics
  - www.humusoft.cz/workshop/
- **Školení**
  - MATLAB, Simulink, dSPACE, COMSOL Multiphysics
  - www.humusoft.cz/skoleni

Děkuji za pozornost