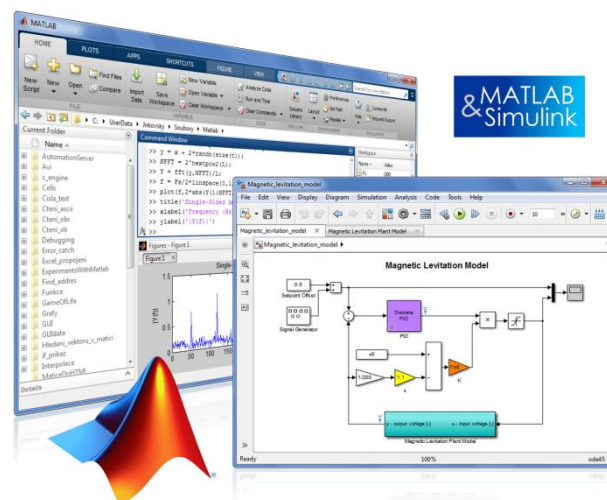


5.9.2019 Brno

TCC 2019

Deep Learning v prostředí MATLAB



Jaroslav Jirkovský
jirkovsky@humusoft.cz

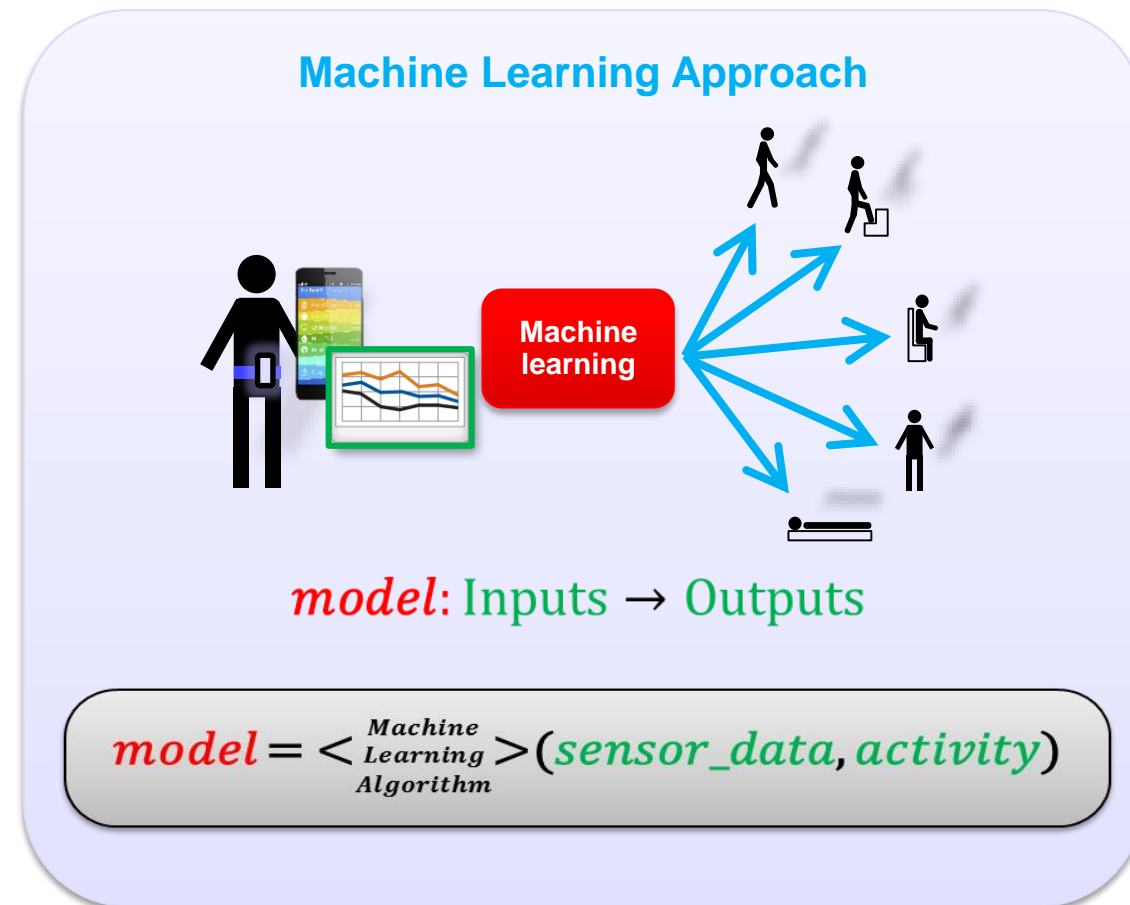
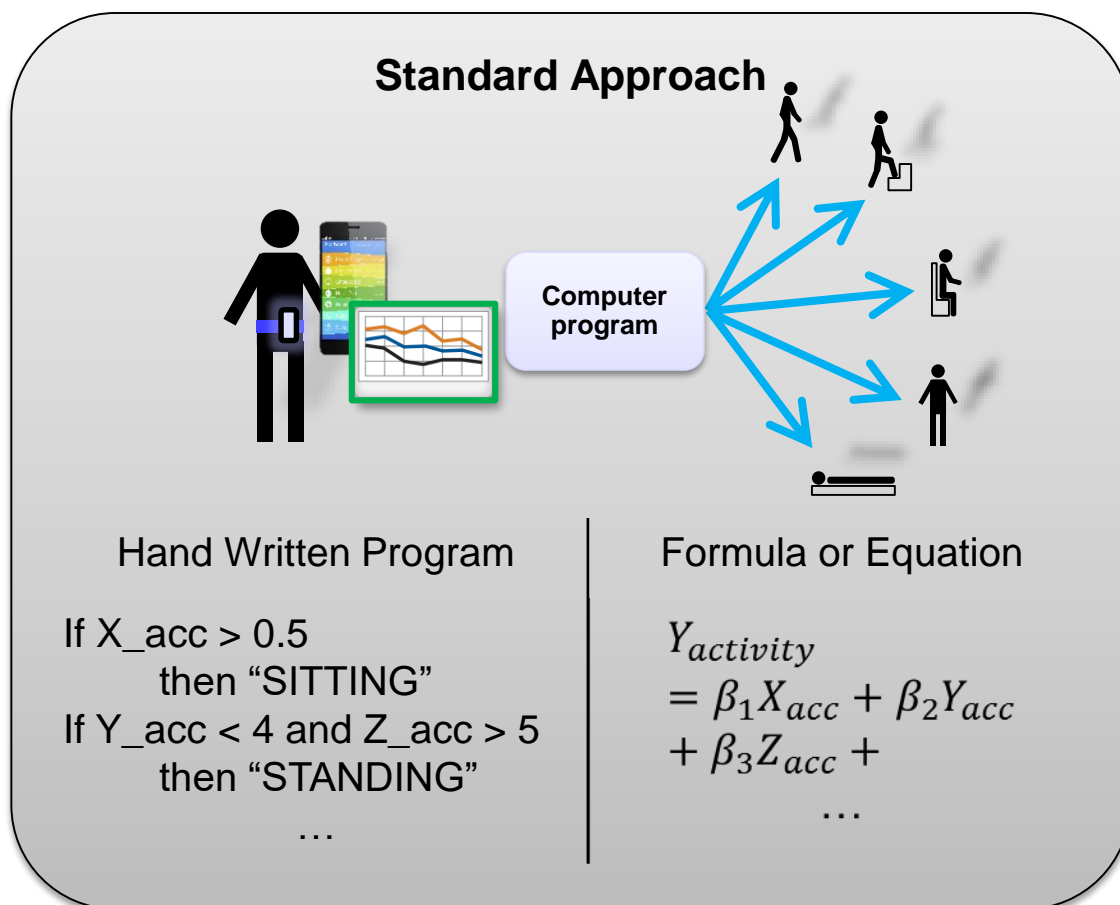
www.humusoft.cz
info@humusoft.cz

www.mathworks.com

What is Machine Learning ?

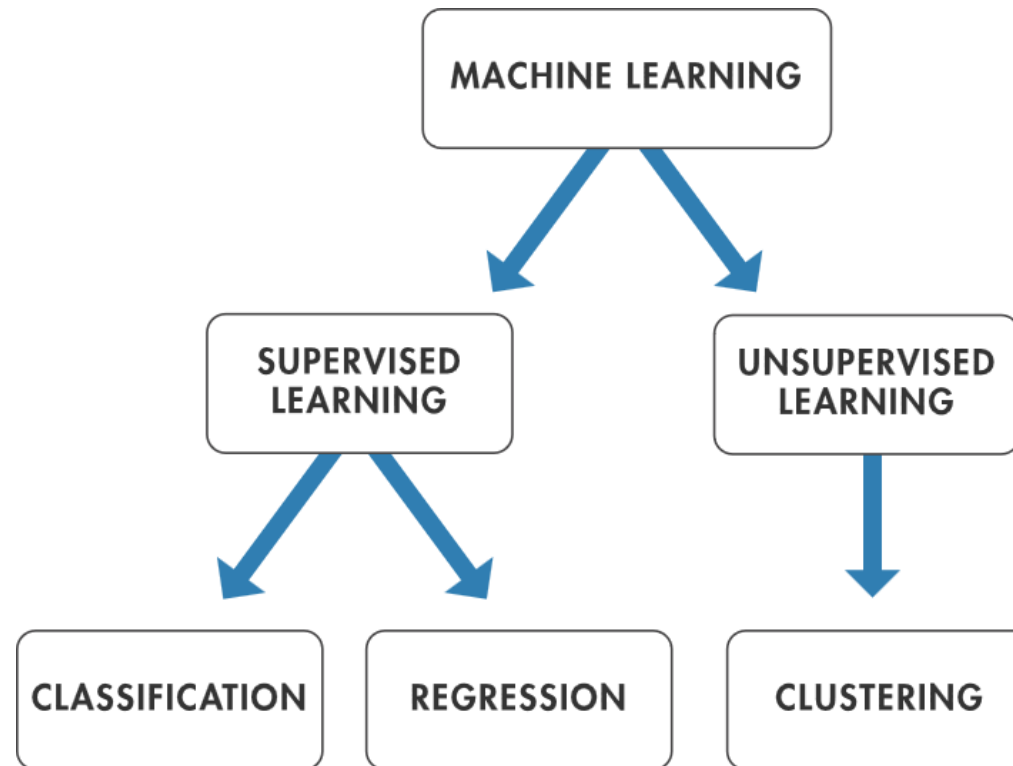
Machine learning uses **data** and produces a **program** to perform a **task**

Task: Human Activity Detection



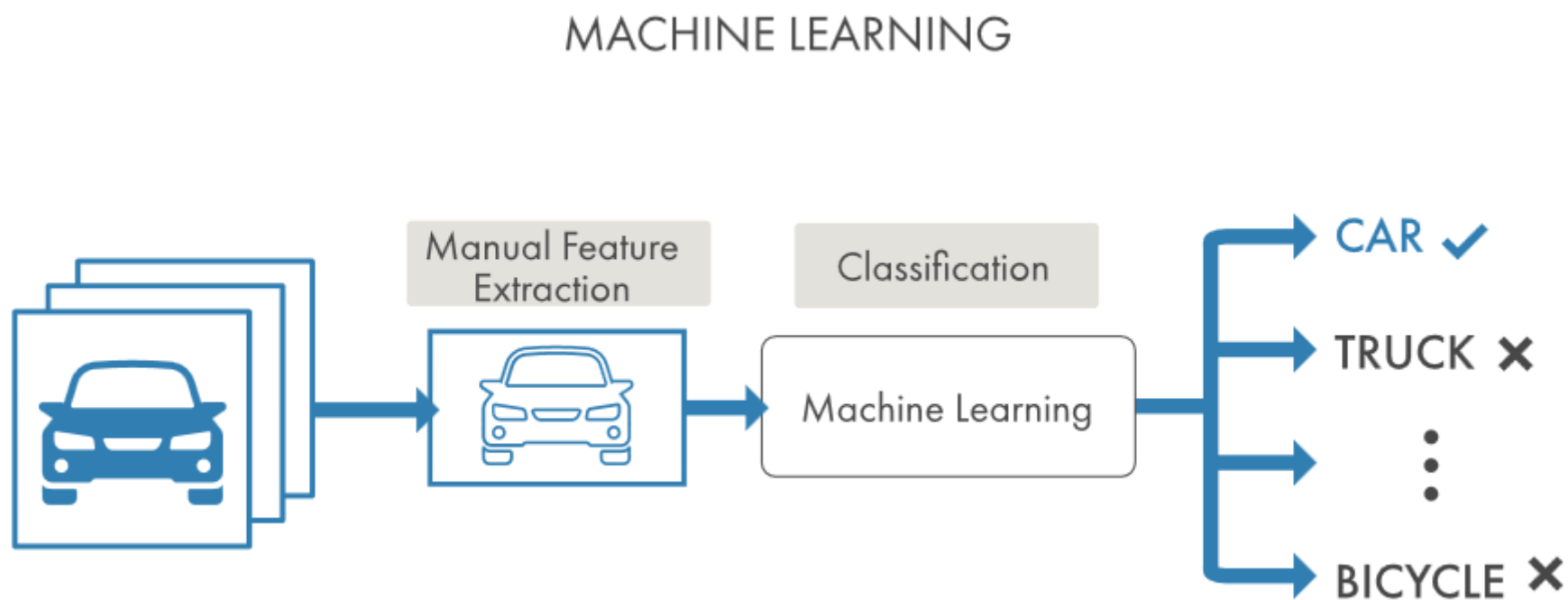
Machine Learning

Different Types of Learning:



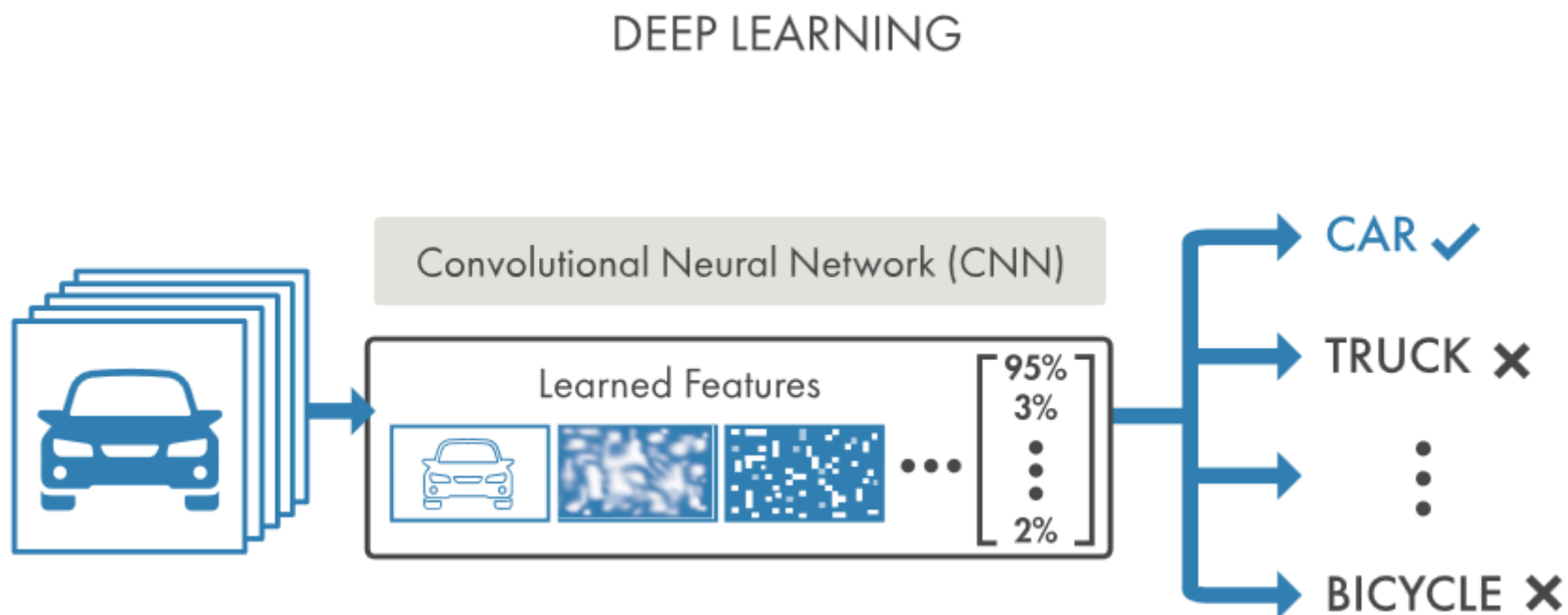
Machine Learning

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



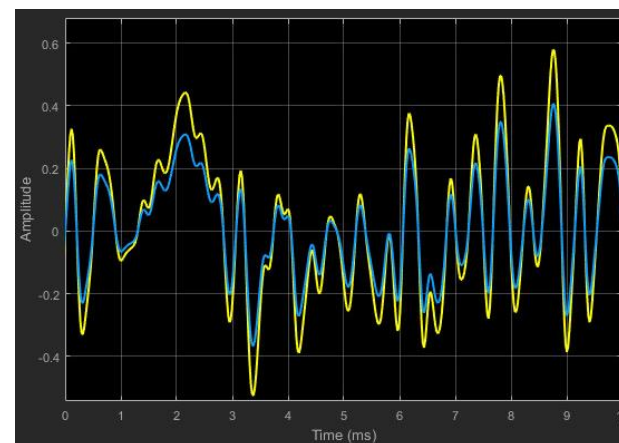
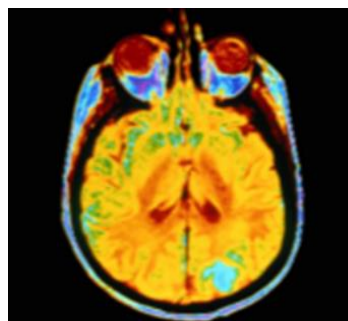
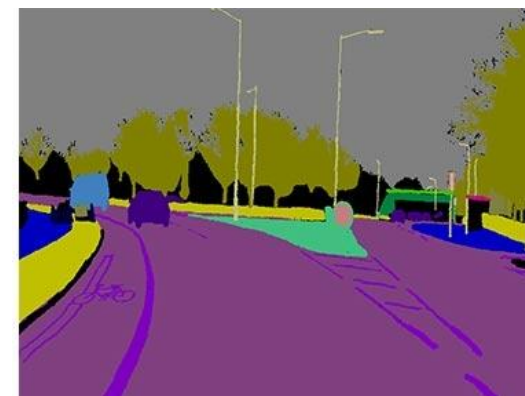
What is Deep Learning ?

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

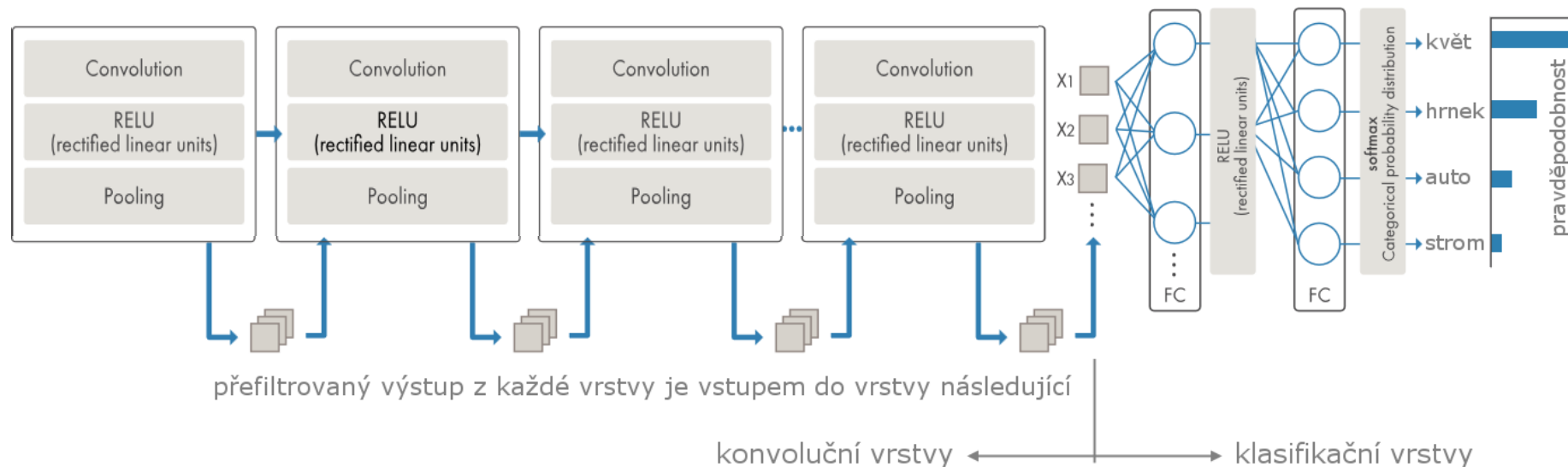
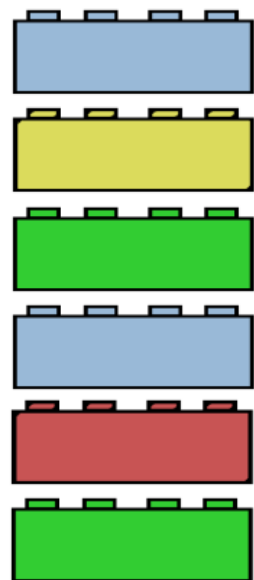


Deep Learning is Ubiquitous

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



Convolutional Neural Networks (CNN)



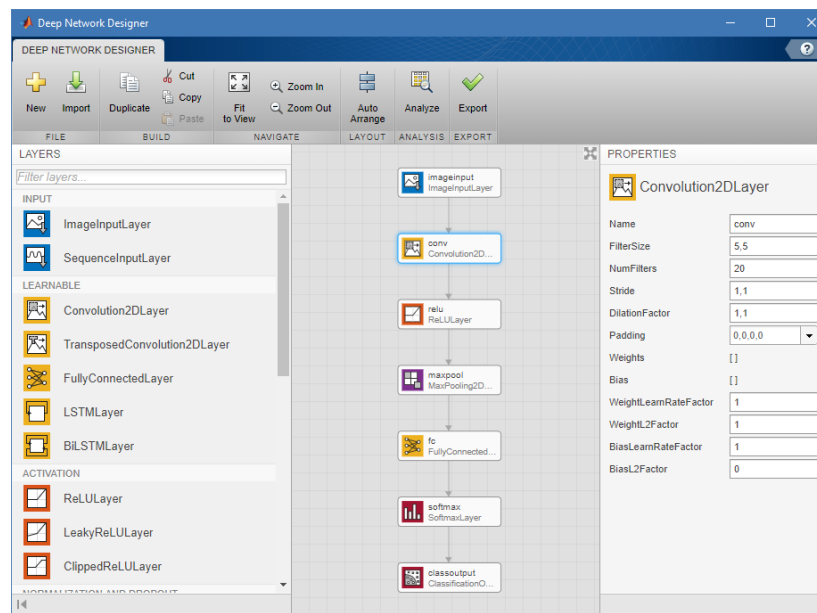
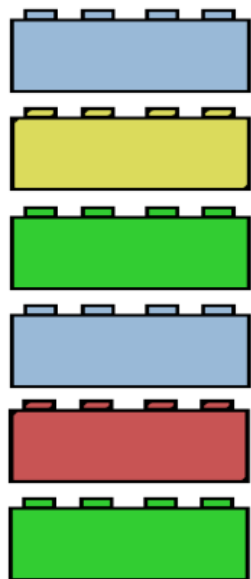
What do filters do?



Great for classification:

- Convolution Layer
- ReLU Layer
- Max Pooling Layer

CNN in MATLAB



```
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);
```

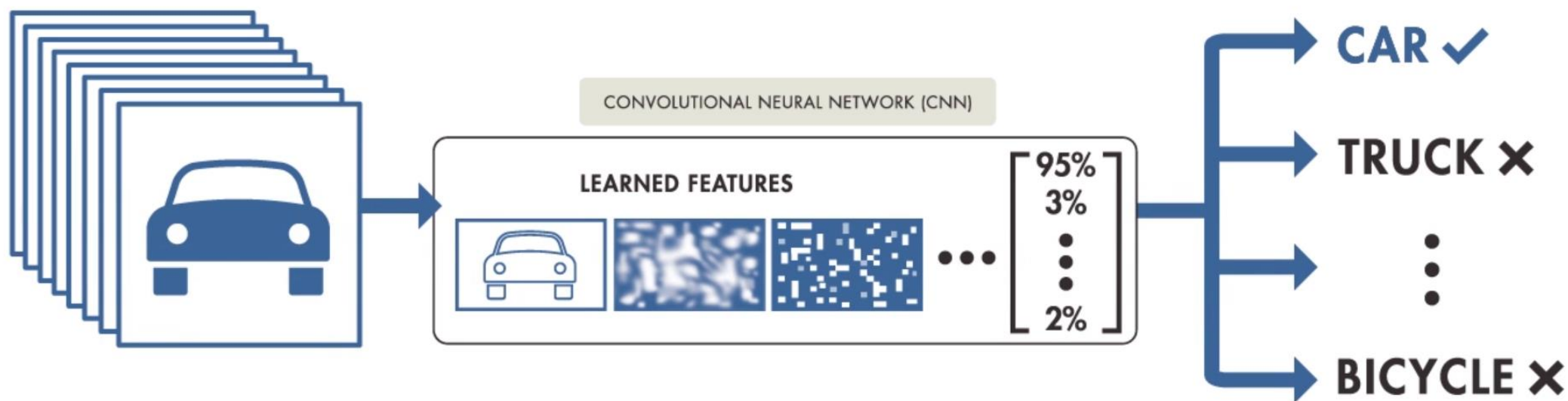

>30 Layers

<code>imageInputLayer</code>	Image input layer		
<code>image3dInputLayer</code>	3-D image input layer		
<code>convolution2dLayer</code>	2-D convolutional layer		
<code>convolution3dLayer</code>	3-D convolutional layer		
<code>groupedConvolution2dLayer</code>	<code>leakyReluLayer</code>	Leaky Rectified Linear Unit (ReLU) layer	
<code>transposedConv2dLayer</code>	<code>clippedReluLayer</code>	Clipped Rectified Linear Unit (ReLU) layer	
<code>transposedConv3dLayer</code>	<code>eluLayer</code>	Exponential linear unit (ELU) layer	
<code>fullyConnectedLayer</code>	<code>tanhLayer</code>	Hyperbolic tangent (tanh) layer	
<code>reluLayer</code>	<code>batchNormalizationLayer</code>	<code>maxPooling2dLayer</code>	Max pooling layer
	<code>crossChannelNormalizationLayer</code>	<code>maxPooling3dLayer</code>	3-D max pooling layer
	<code>dropoutLayer</code>	<code>maxUnpooling2dLayer</code>	Max unpooling layer
	<code>averagePooling2dLayer</code>	<code>additionLayer</code>	Addition layer
	<code>averagePooling3dLayer</code>	<code>concatenationLayer</code>	Concatenation layer
		<code>depthConcatenationLayer</code>	Depth concatenation layer
		<code>softmaxLayer</code>	Softmax layer
		<code>classificationLayer</code>	Classification output layer
		<code>regressionLayer</code>	Create a regression output layer

- Author custom layers in MATLAB using the Custom Layer API

2 Approaches for Deep Learning

- Approach 1: Train a Deep Neural Network from Scratch

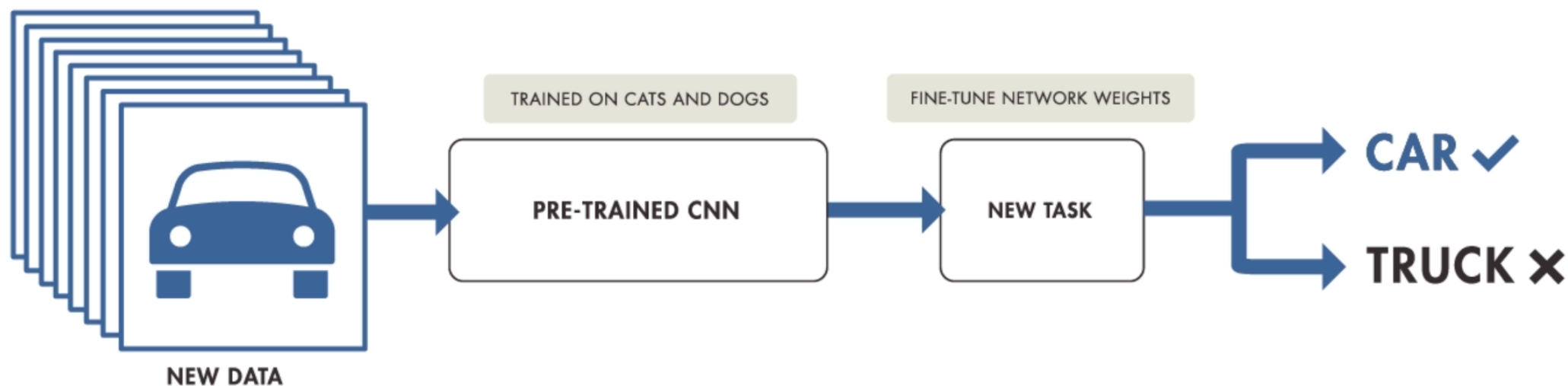


Recommended only when:

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)



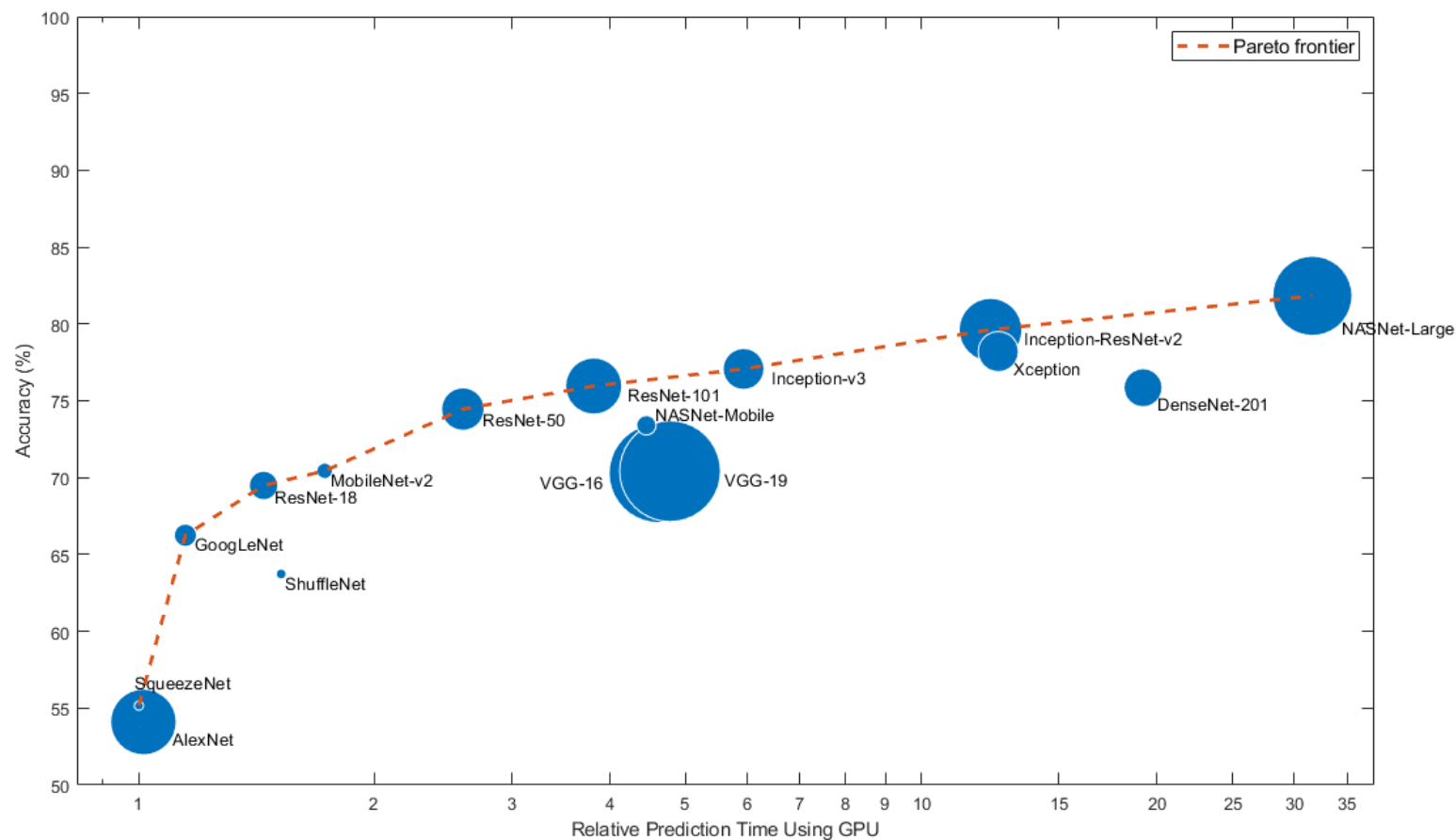
Recommended when:

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**

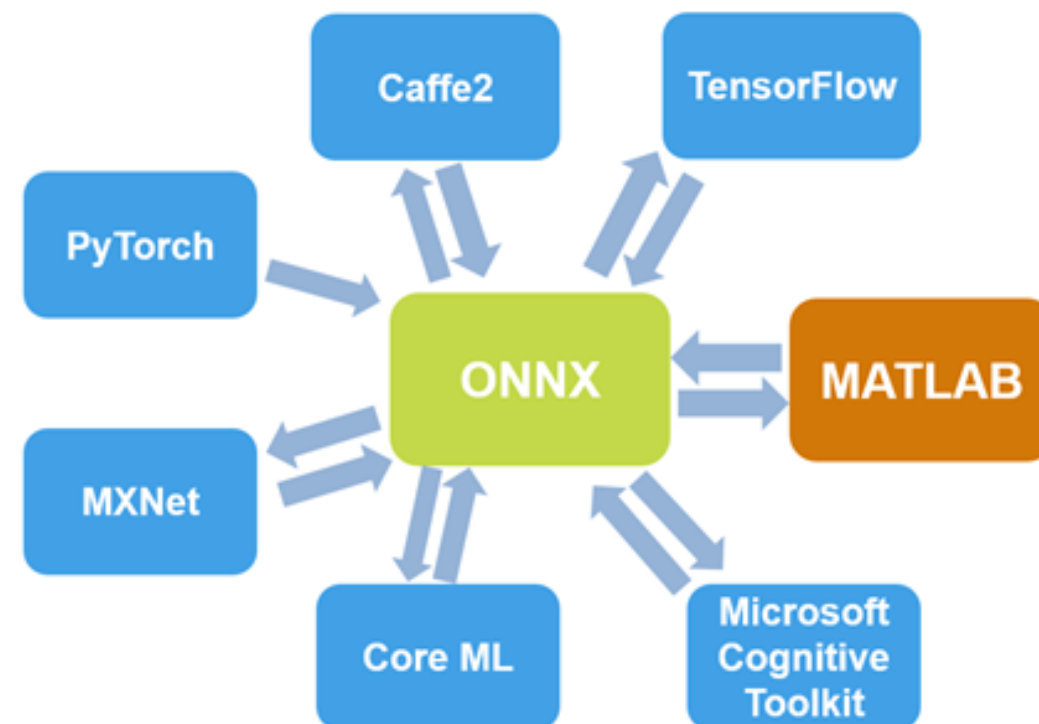
- AlexNet
- VGG-16 and VGG-19
- GoogLeNet
- ResNet-50 and ResNet-101
- Inception-v3
- Inception-ResNet-v2
- SqueezeNet
- and more ...



Transfer Learning using Pre-Trained Networks

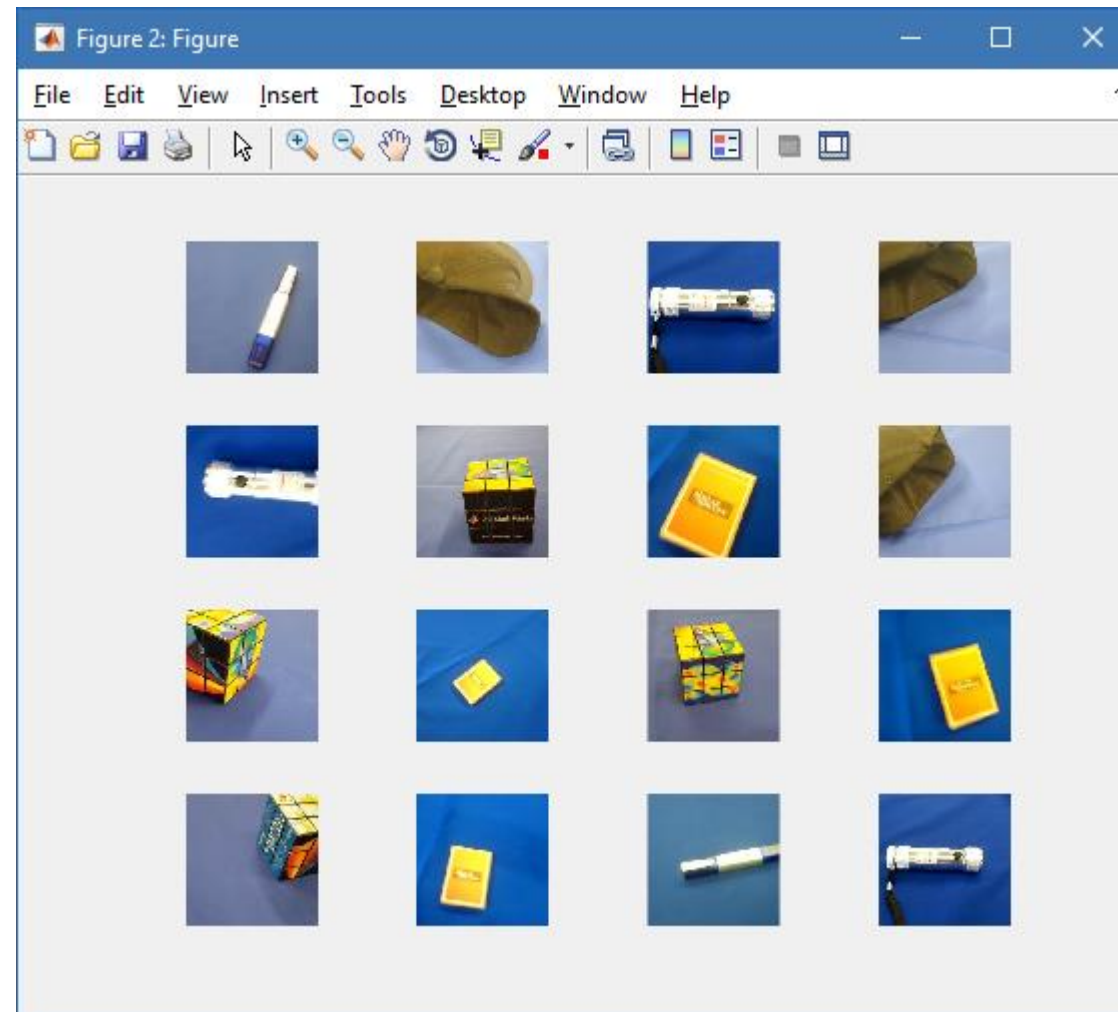
- **Pre-Trained Networks**

- AlexNet
- VGG-16 and VGG-19
- GoogLeNet
- ResNet-50 and ResNet-101
- Inception-v3
- Inception-ResNet-v2
- SqueezeNet
- and more ...



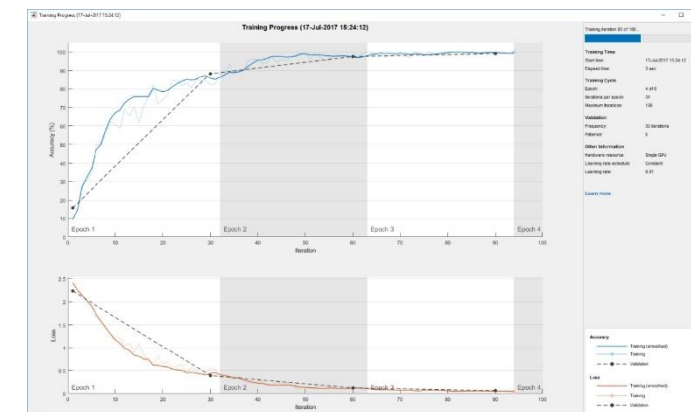
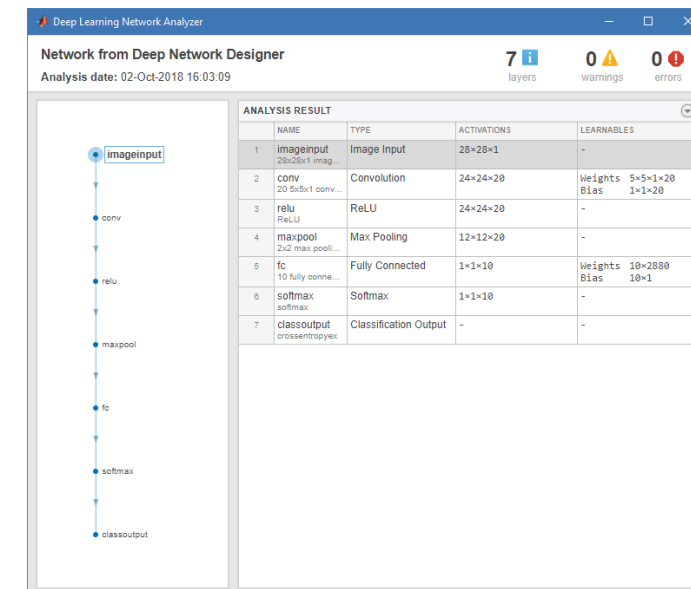
- **ONNX Model Converter**

Example: Fine-tune a pre-trained model (transfer learning)



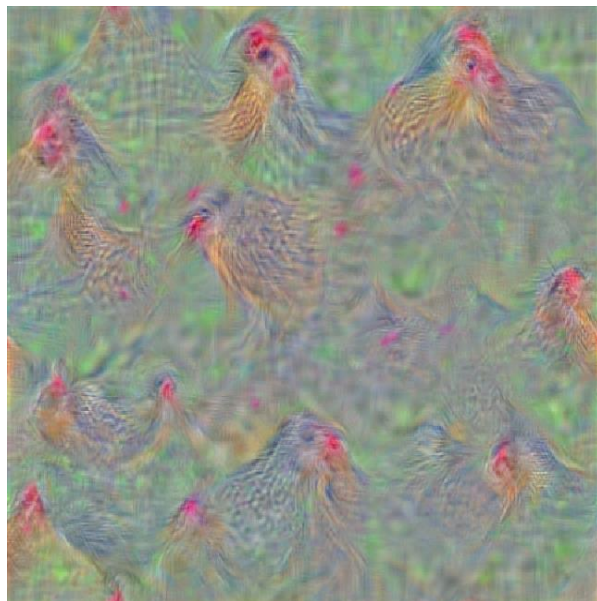
Training, Validation and Visualization

- **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
- **Visualize activations and filters from intermediate layers, CAM**
- **Deep Dream visualization**

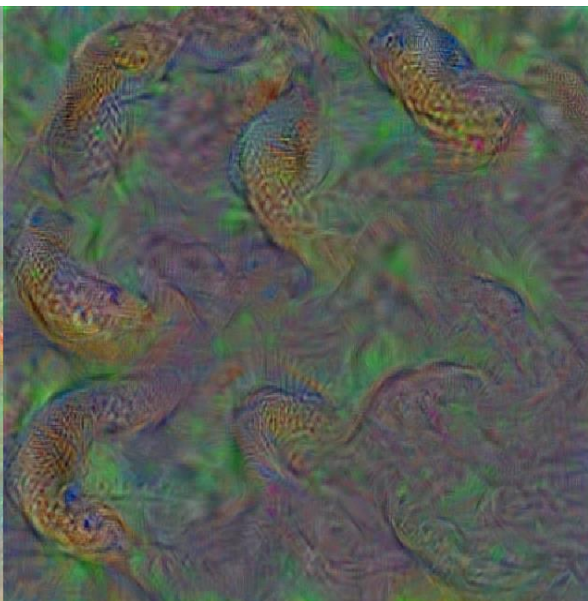


Deep Dream Images Using AlexNet

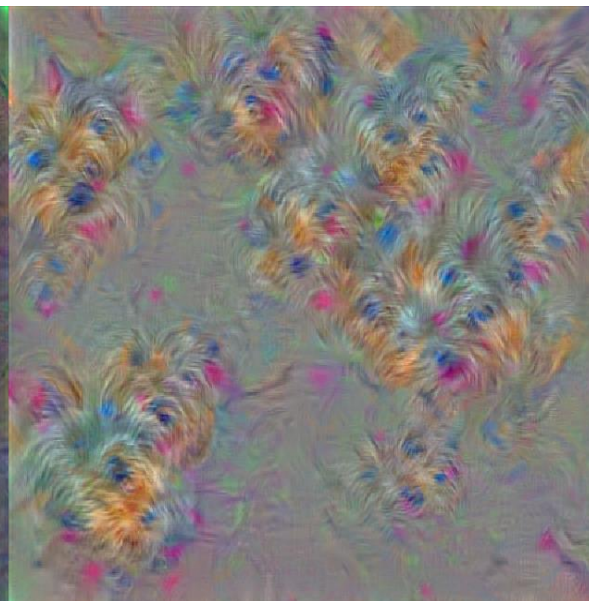
Hen



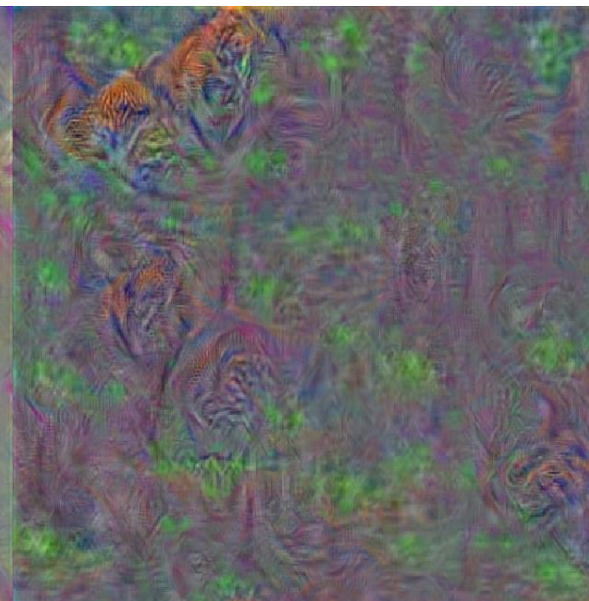
Indian cobra



Yorkshire terrier

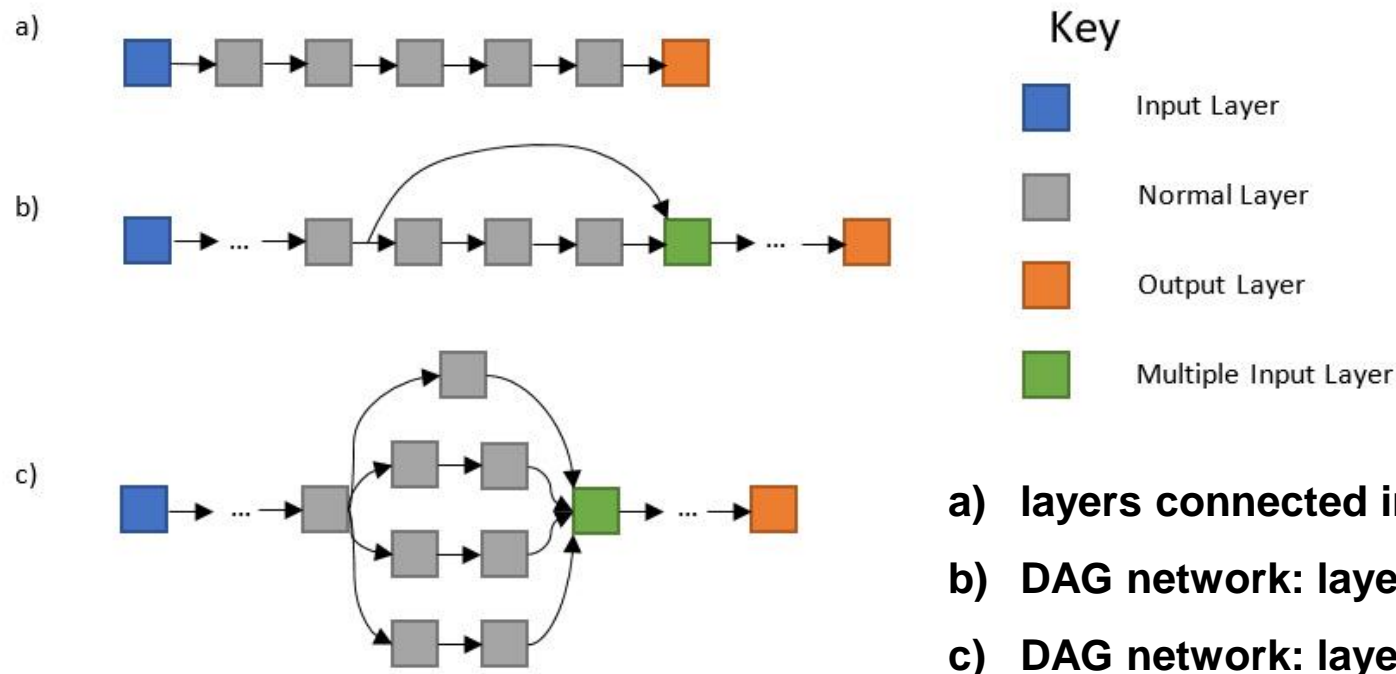


Tiger



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)

Image Classification vs. Object Detection

- **Image Classification**

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



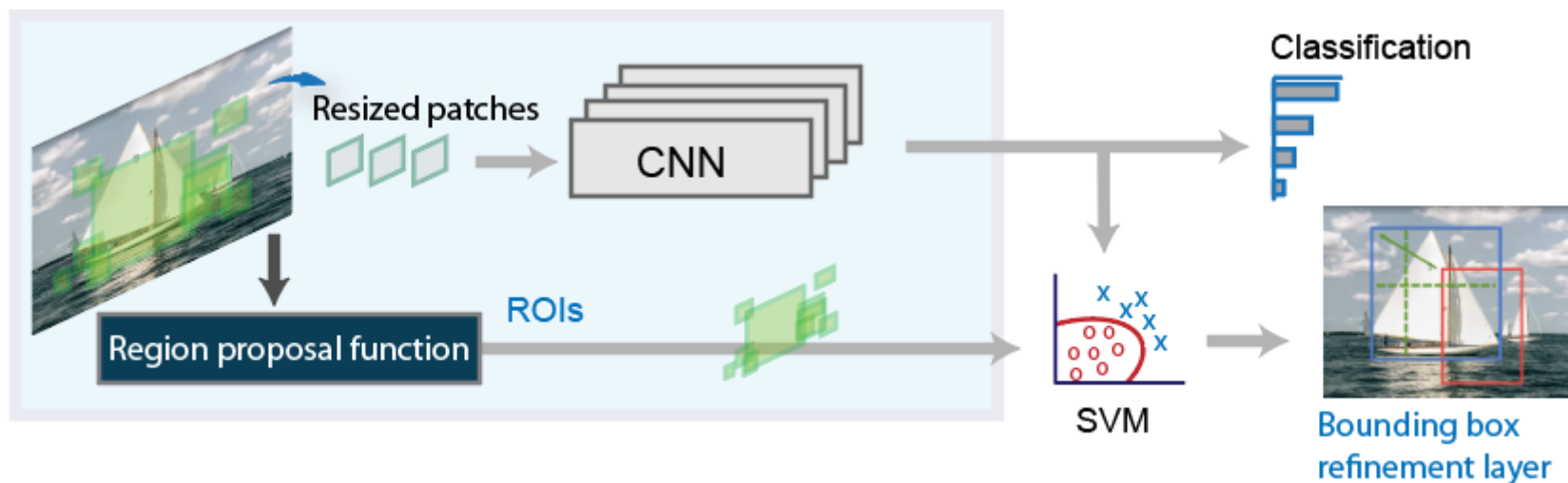
- **Object Detection**

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



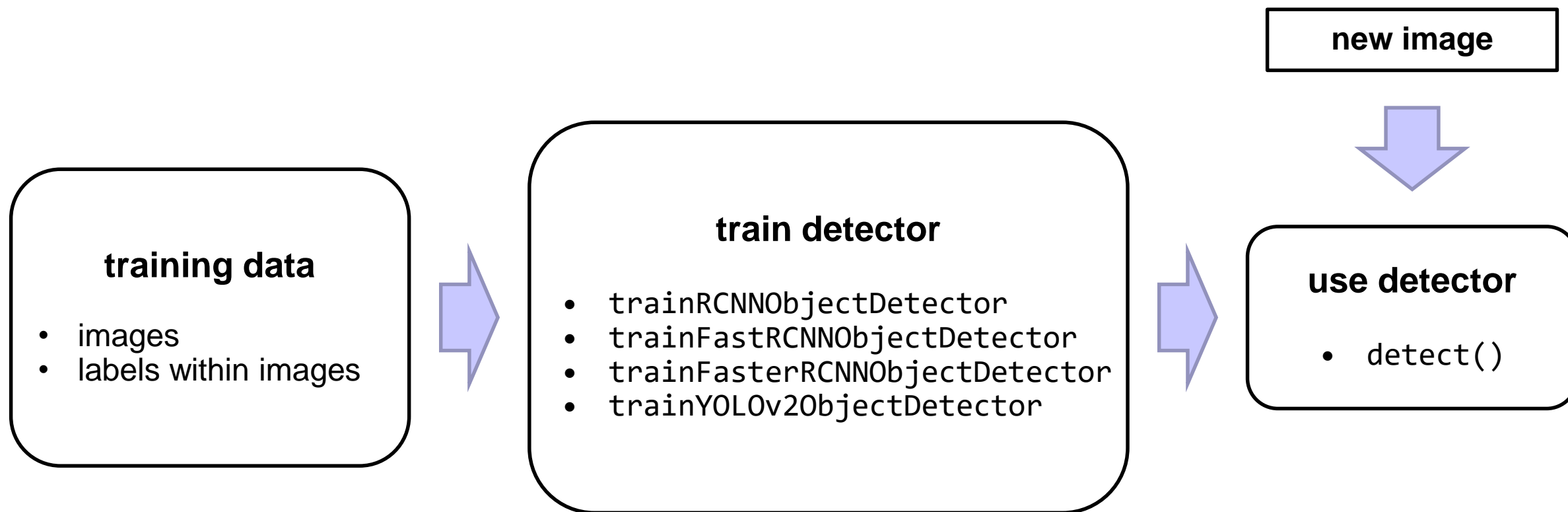
Object Detection using Deep Learning

- **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



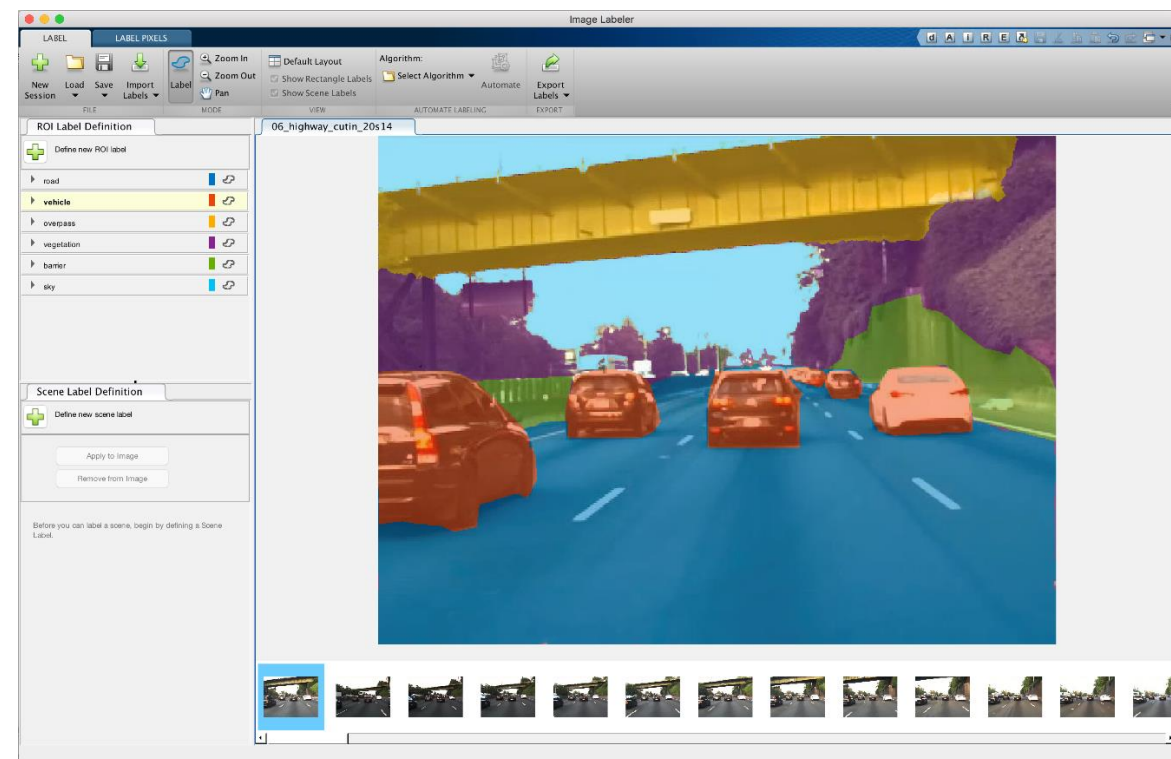
- Fast and Faster R-CNN improve detection performance for large number of regions
- **YOLO v2 deep learning object detector (you-only-look-once)**

Object Detection Training Workflow

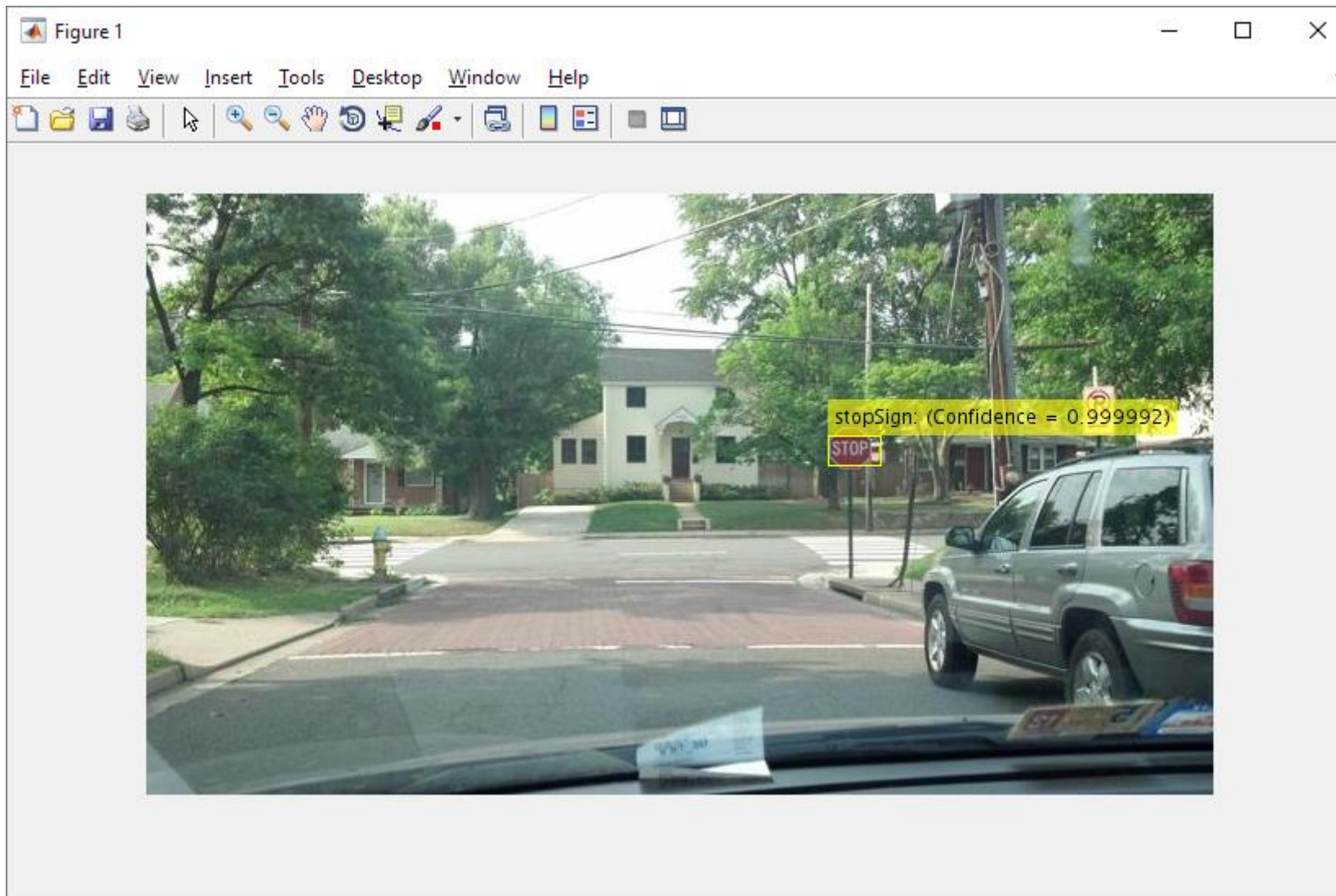


Ground-Truth Labeling

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

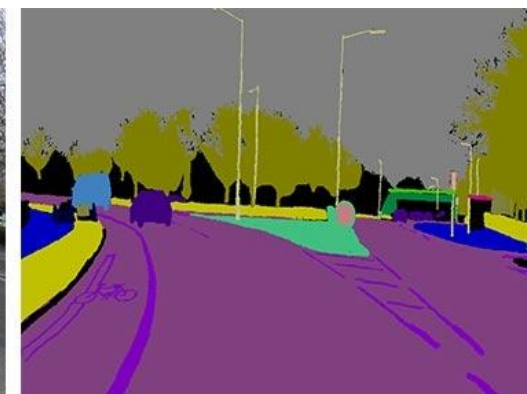
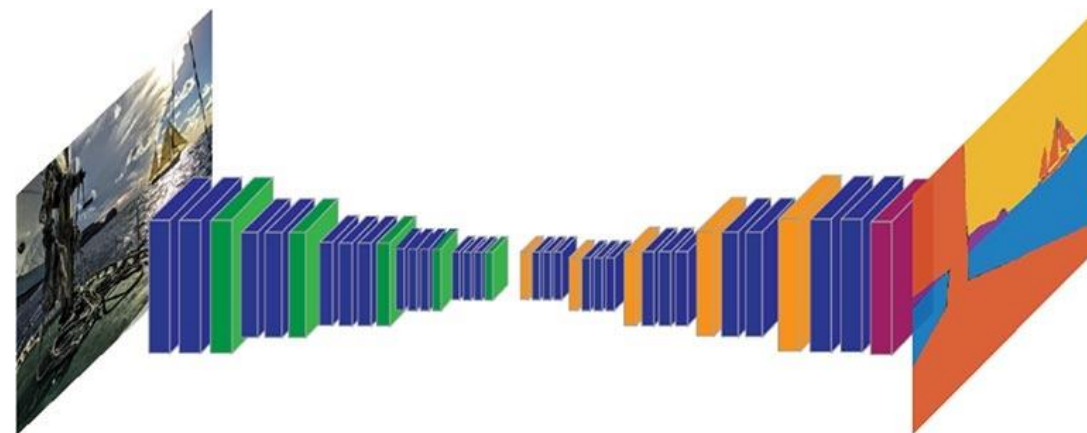


Example: Object Detection using Deep Learning

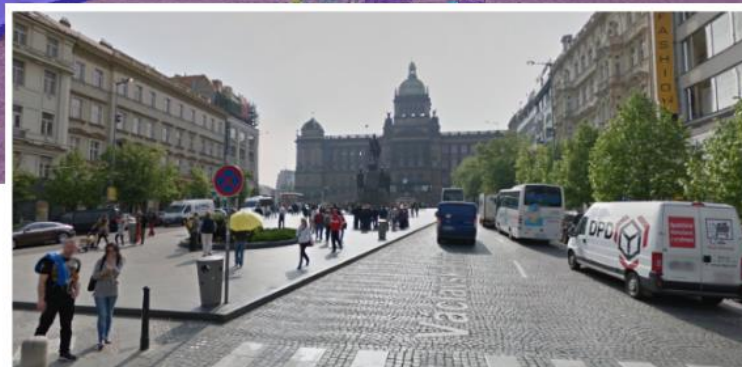
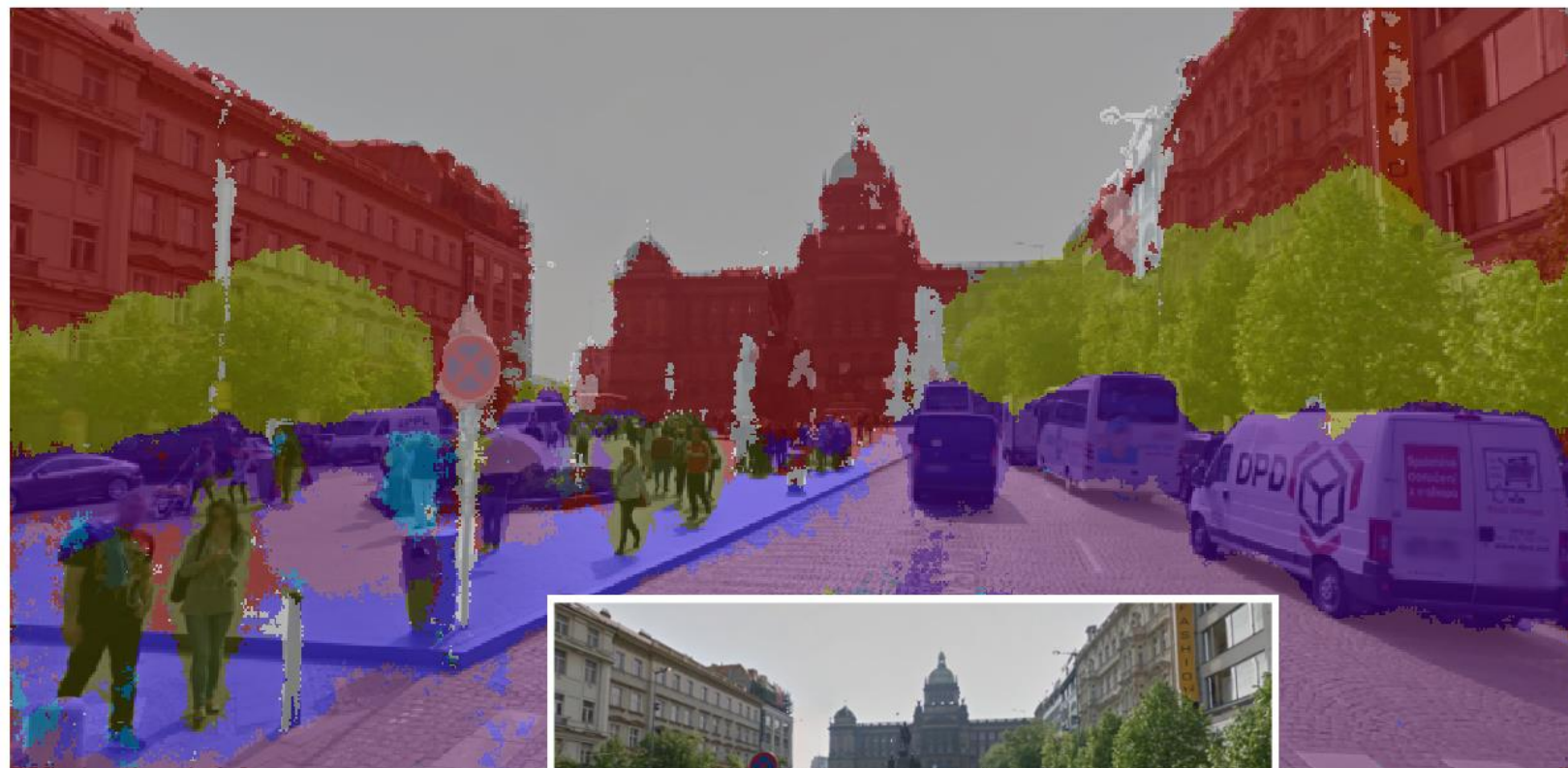


Semantic Segmentation

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



Semantic Segmentation



- Cyklista
- Chodec
- Automobil
- Plot
- Dopravní značka
- Strom
- Chodník
- Silnice
- Sloupek
- Budova
- Obloha

Semantic Segmentation



Deep Learning with Time Series Workflow

1. Create time-frequency representation of the signal data

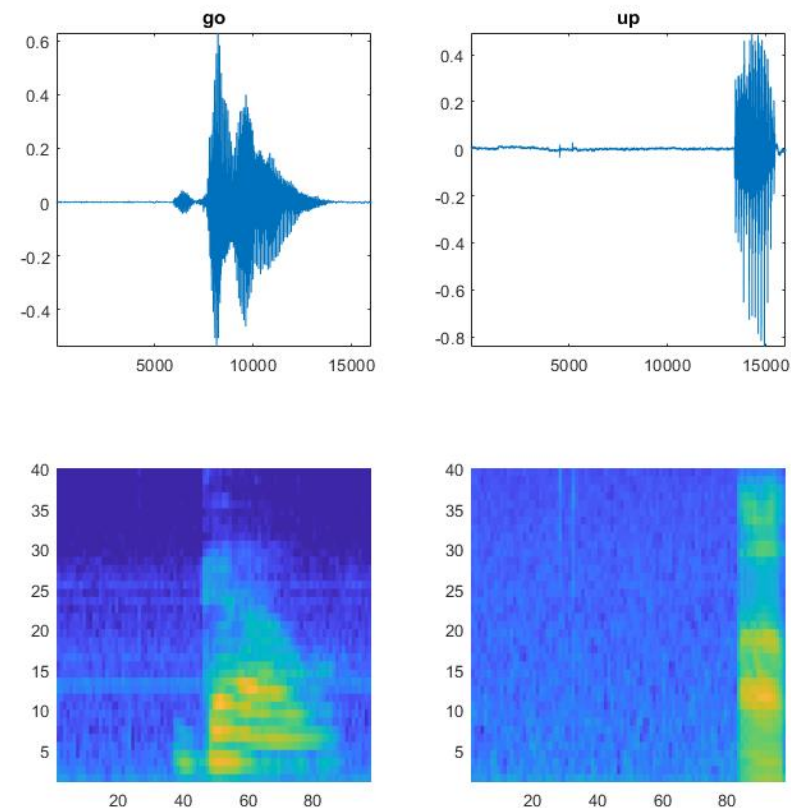
- *Signal Analyzer app*
- spectrogram
- scalogram (continuous wavelet transform)

2. Capture time-frequency images

3. Apply CNN to the images

or

Use **Long Short Term Memory (LSTM) Networks** directly with signal data

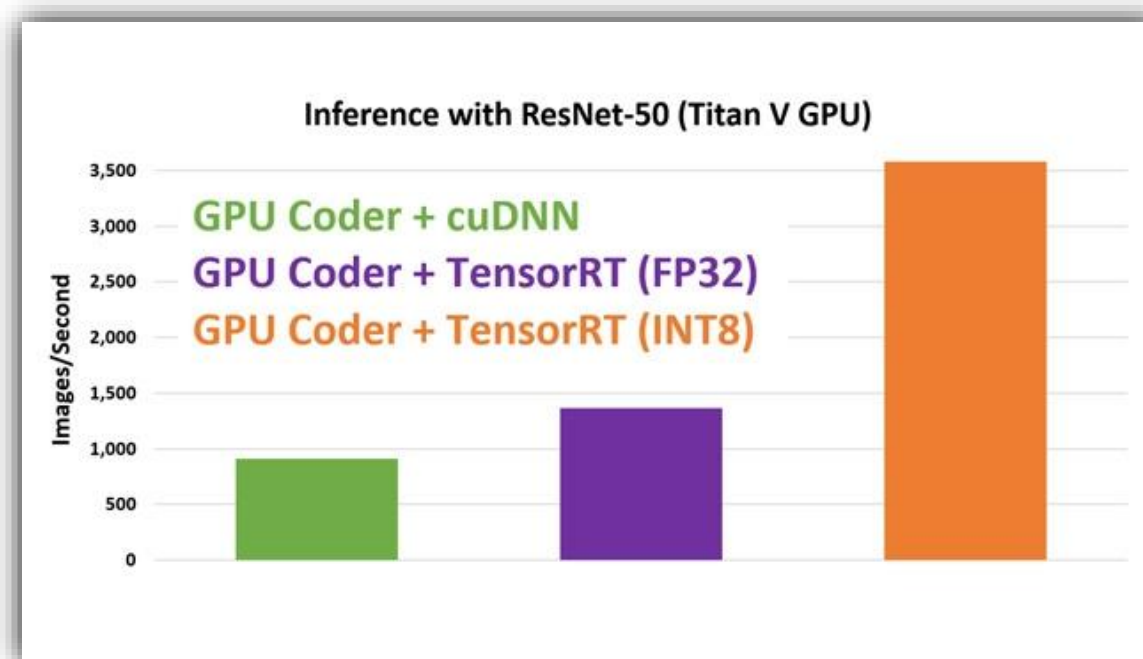


Multi-Platform Deployment

- **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.**



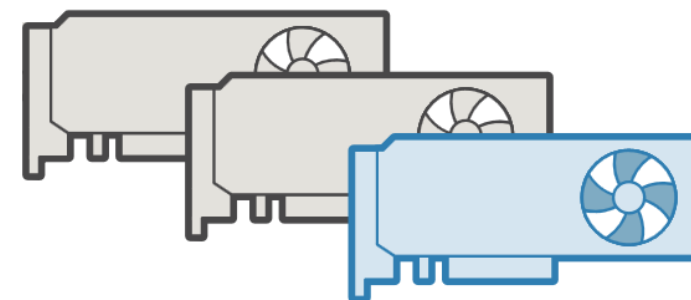
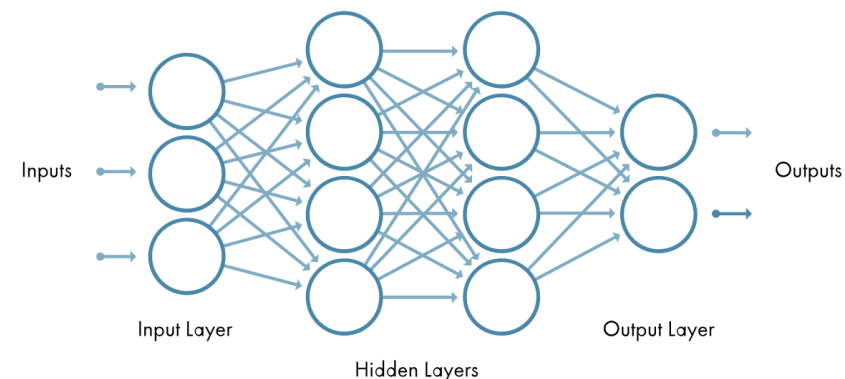
Multi-Platform Deployment



⇒ deployable models with high-performance inference speed.

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**

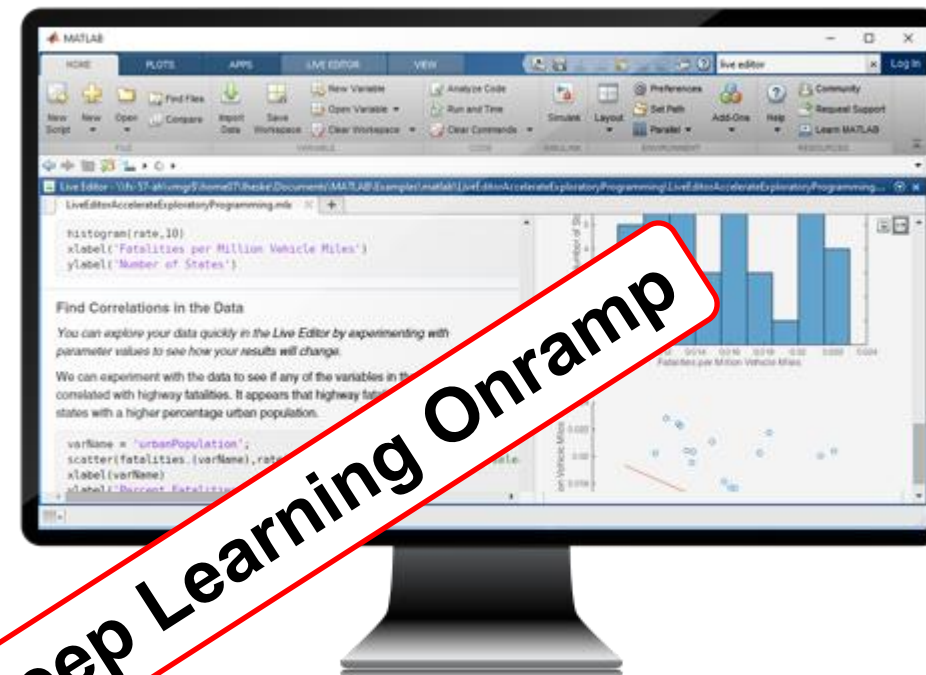


Jak začít s prostředím MATLAB?

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

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

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



+ Deep Learning Onramp

Děkuji za pozornost