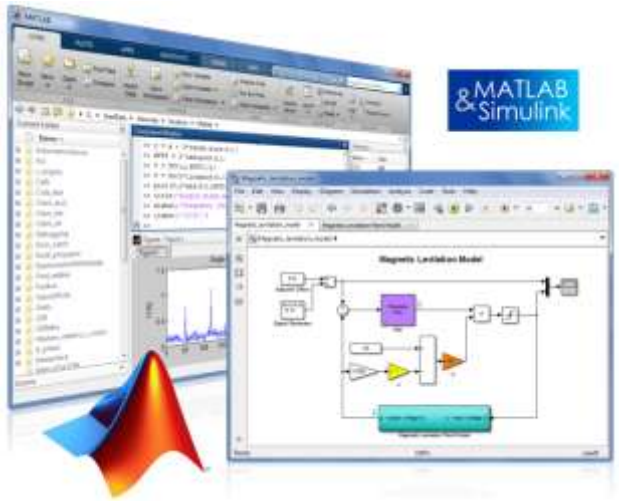
HUMUSOFT®

TCC 2018

Deep Learning v prostředí MATLAB



Jaroslav Jirkovský

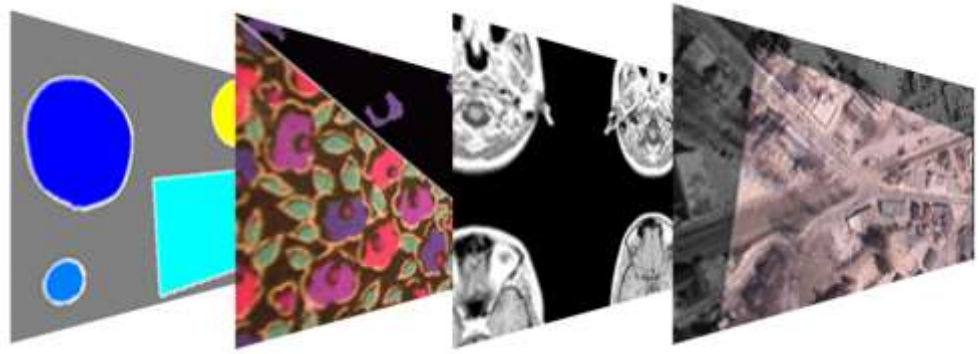
jirkovsky@humusoft.cz

www.humusoft.cz
info@humusoft.cz

www.mathworks.com

Zpracování obrazu a počítačové vidění

- Snímání reálného obrazu
- Zpracování obrazu a videa
  - úprava obrazu, transformace, segmentace
  - práce s barevnými prostory
- Počítačové vidění
  - detekce a sledování objektů
  - detekce obličeje, postav
  - 3-D vision, OCR
- Deep Learning
  - rozpoznávání obrazu a detekce objektů
  - sémantická segmentace



Počítačového vidění: typy úloh a jejich řešení

- **Hledání vzorového objektu**
  - nalezení a porovnání příznaků (BRISK, SURF, KAZE, MSER, corner)

- **Detekce objektů**
  - cascade object detector (Viola-Jones)
  - ACF object detector
  - R-CNN, Fast R-CNN, Faster R-CNN

- **Klasifikace objektů (snímků)**
  - bag-of-visual words
  - CNN

deep learning

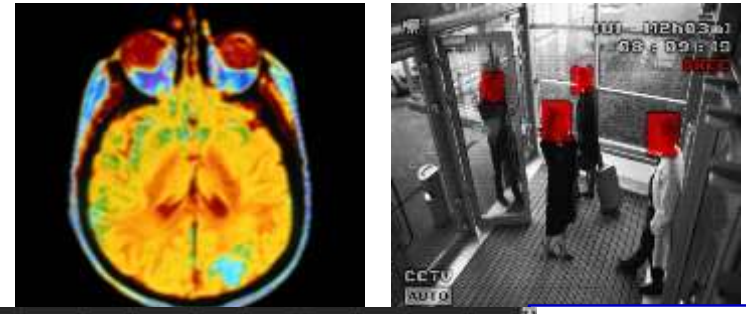
- **Sledování objektů**
  - sledování bodů (KLT)
  - sledování oblasti na základě histogramu

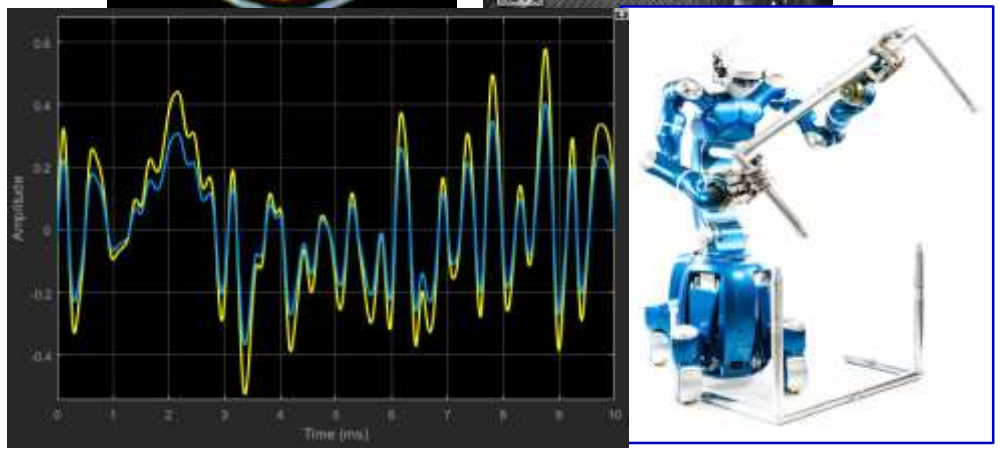
- **Odhad a predikce pohybu**
- **Detekce popředí, …**

Deep Learning is Ubiquitous

Computer Vision

- Pedestrian and traffic sign detection
- Landmark identification
- Scene recognition
- Medical diagnosis and discovery





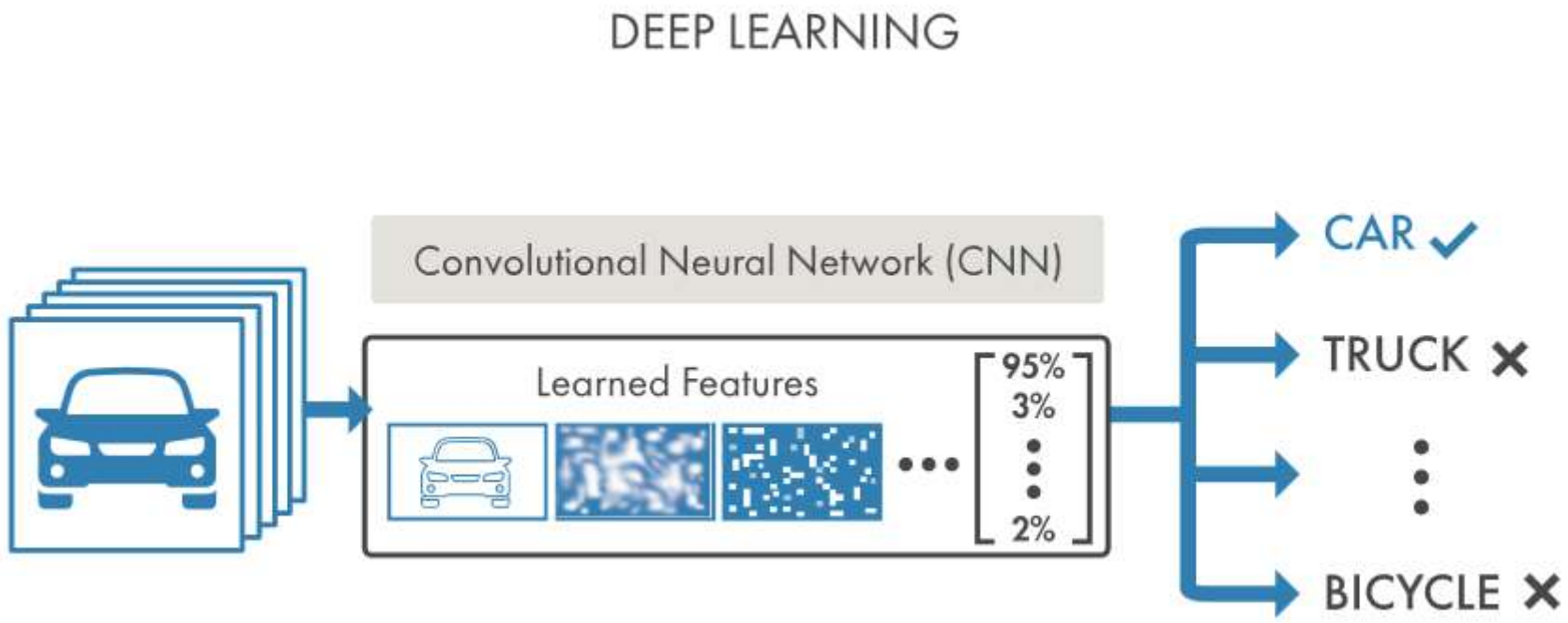
Signal and Time Series Processing

Text Analytics

…

What is Deep Learning ?

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

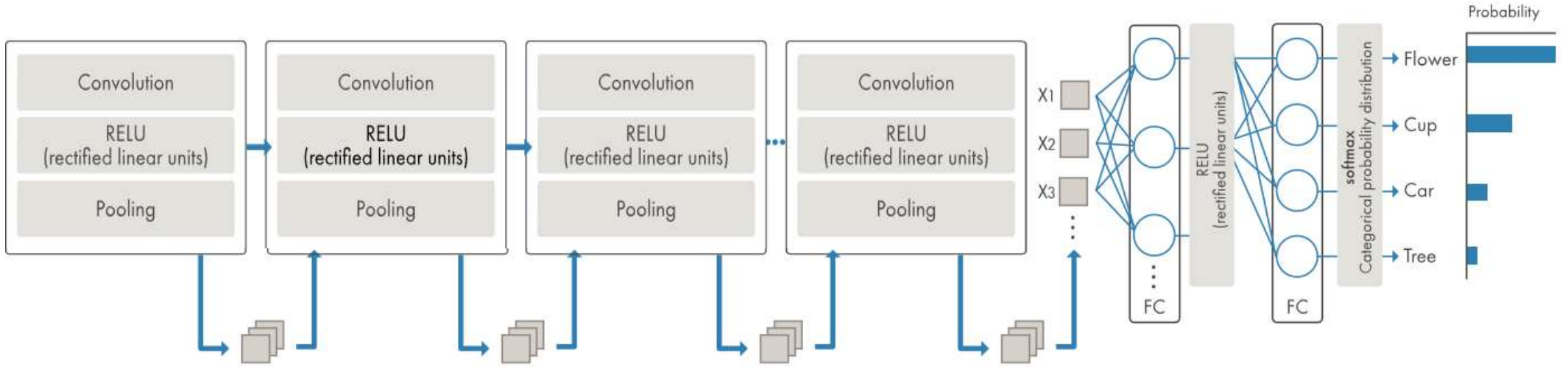


Why is Deep Learning so Popular ?

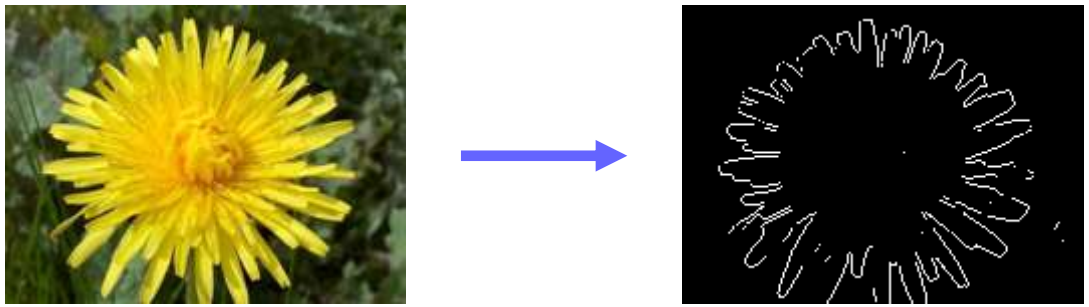
- **Results:**
  - **95% + accuracy**
    - on ImageNet 1000 class challenge
- **Computing Power:**
  - **GPU's**
  - **advances to processor technologies**
  - ⇨ **possible to train networks on massive sets of data.**
- **Data:**
  - **availability of storage**
  - **access to large sets of labeled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

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

Convolutional Neural Networks



What do filters do?



CNN in MATLAB

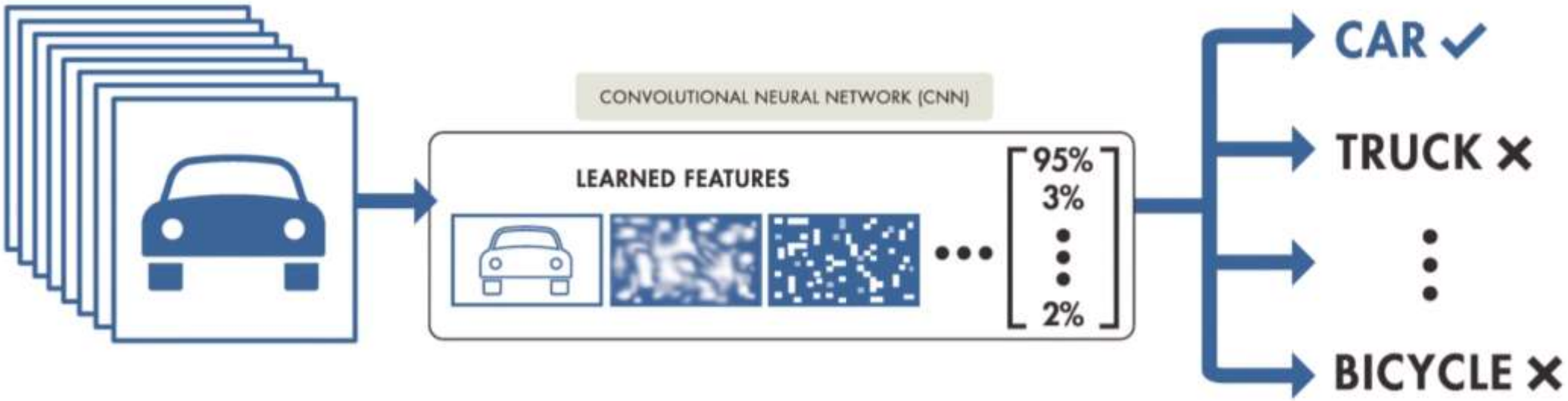
```
layers = [imageInputLayer(image_size)
          convolution2dLayer(filter_size,num_filters)
          reluLayer()
          maxPooling2dLayer(window_size,'Stride',step)
          fullyConnectedLayer(num_classes)
          softmaxLayer()
          classificationLayer()];
options = trainingOptions('sgdm');
convnet = trainNetwork(trainingData,layers,options);
results = classify(convnet,newData);
```

CNN in MATLAB

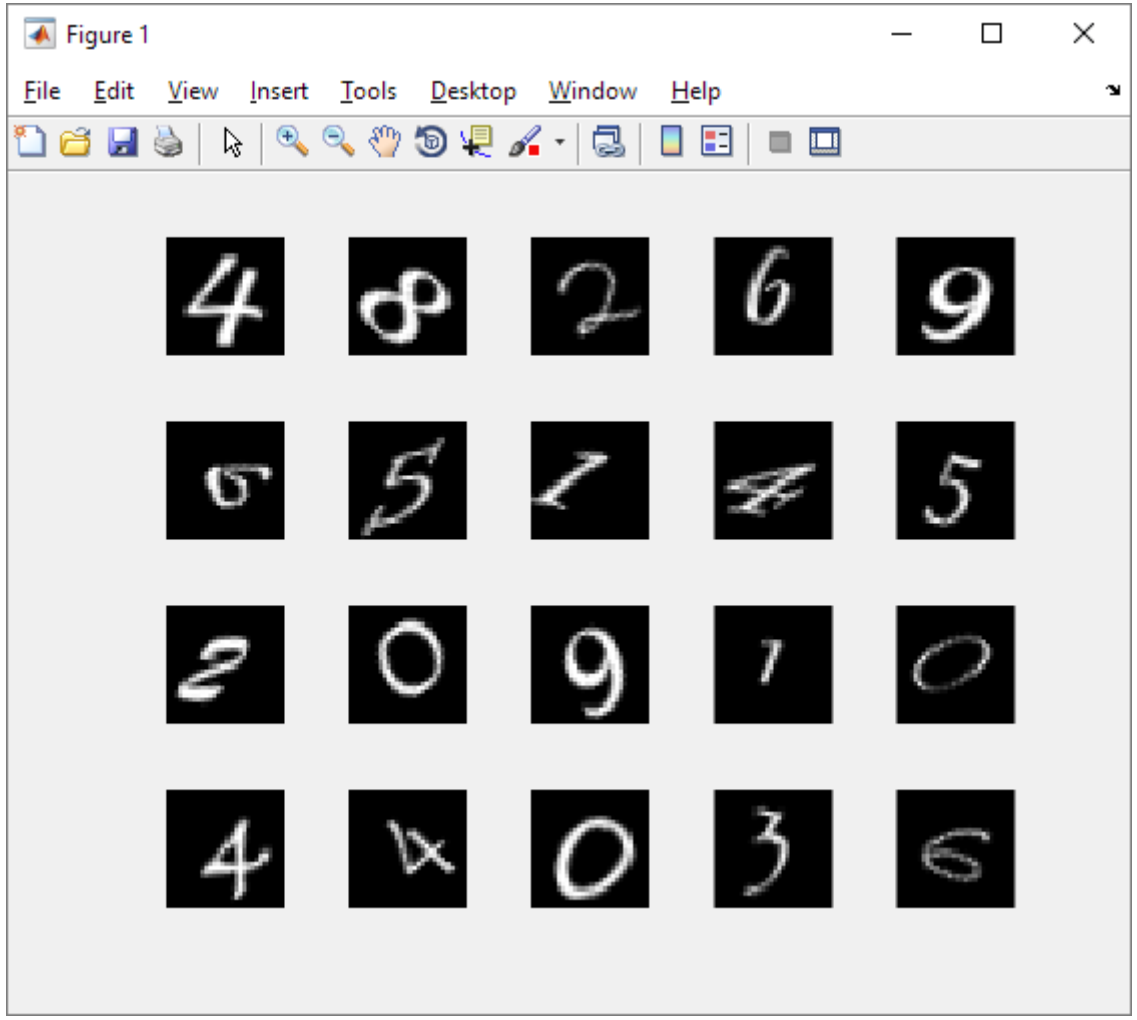
```
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(5,20)
          reluLayer()
          maxPooling2dLayer(2,'Stride',2)
          fullyConnectedLayer(10)
          softmaxLayer()
          classificationLayer()];
options = trainingOptions('sgdm');
convnet = trainNetwork(trainingData,layers,options);
results = classify(convnet,newData);
```

2 Approaches for Deep Learning

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

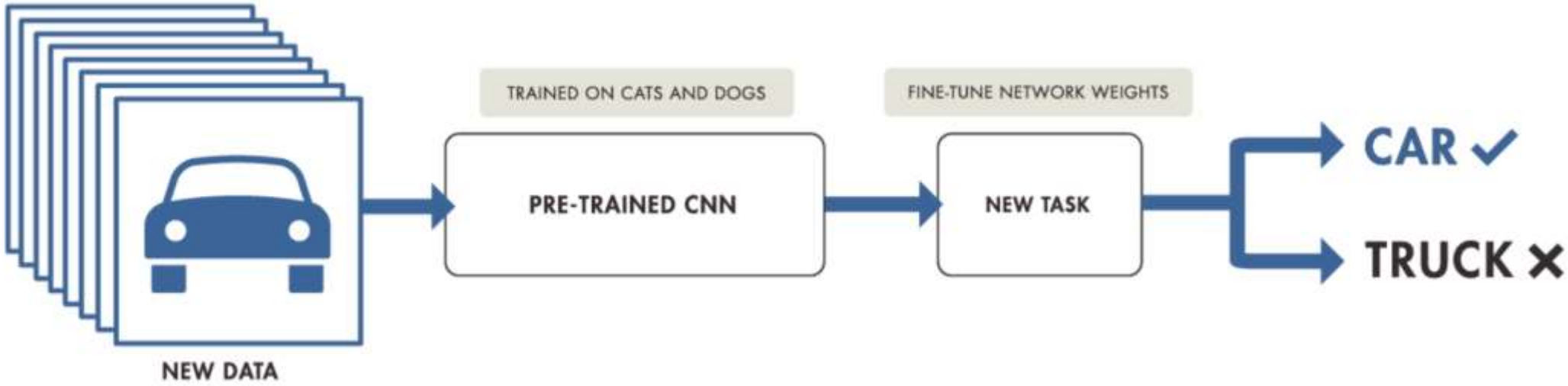


Demo : Train a Deep Neural Network from Scratch

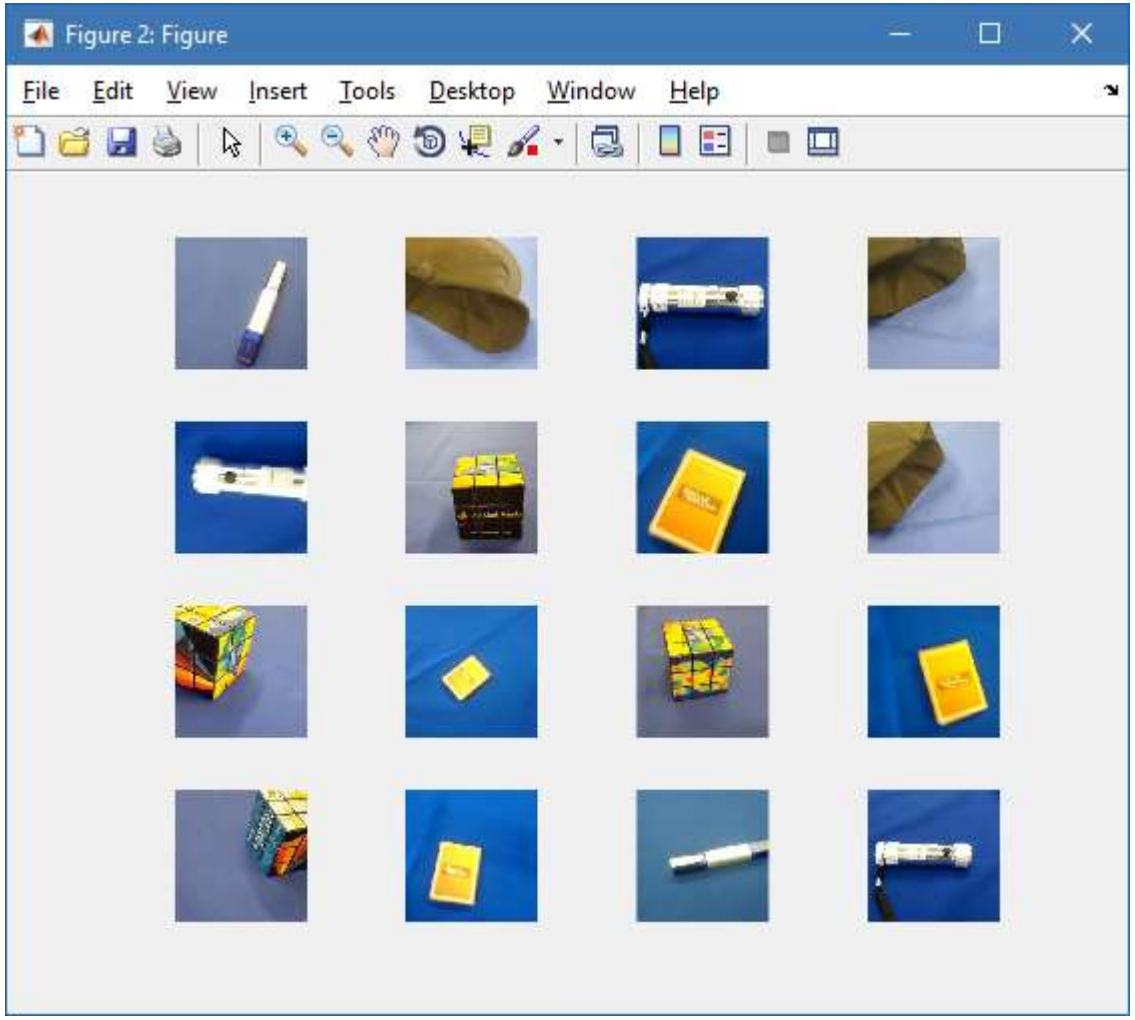


2 Approaches for Deep Learning

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



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

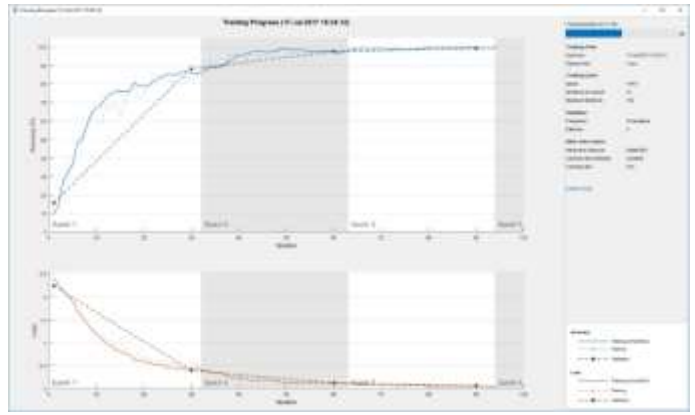


Available pre-trained CNNs

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

- Import models from Caffe (including Caffe Model Zoo)
- Import models from TensorFlow-Keras

Training and Visualization



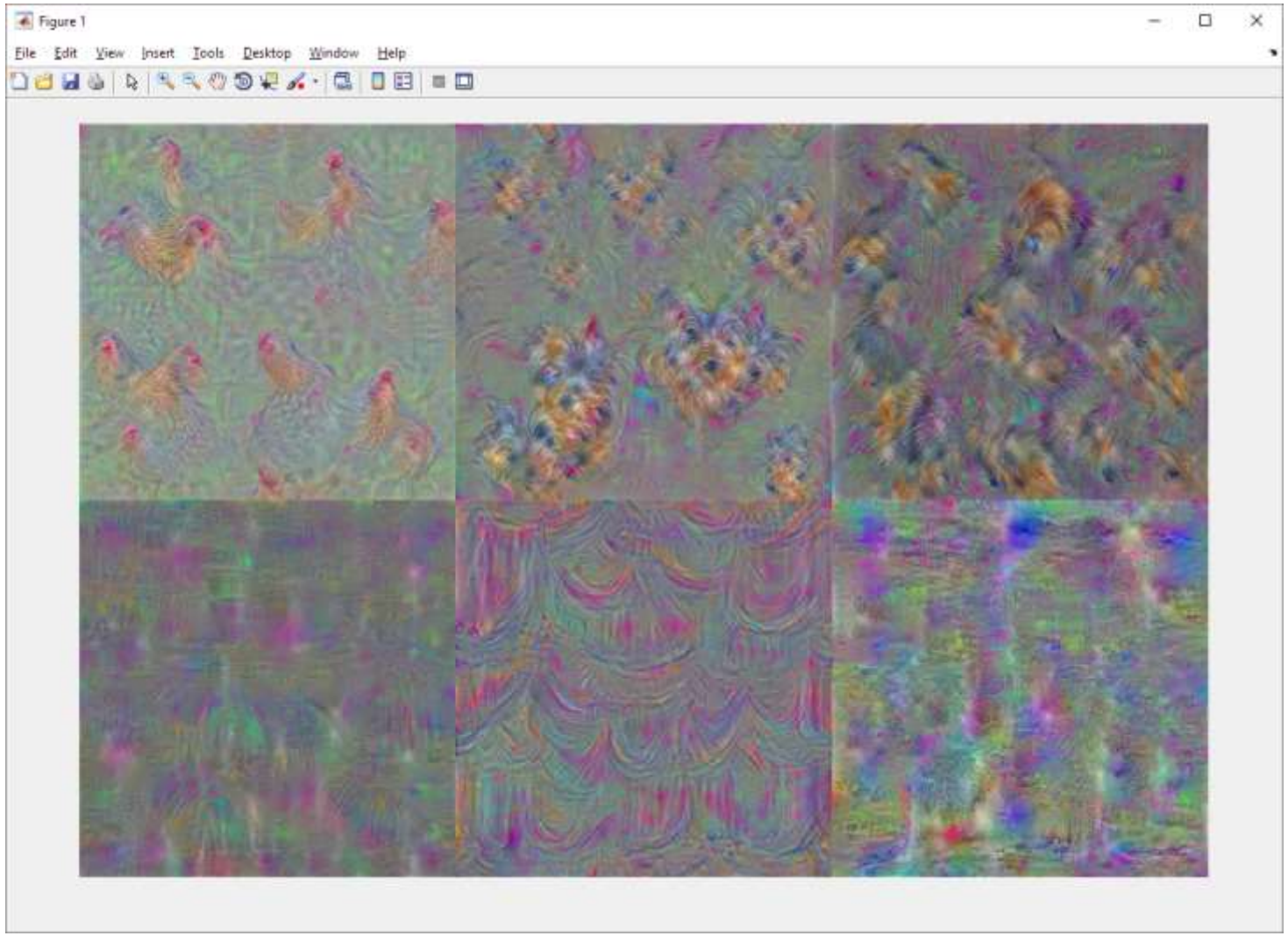
- **Monitor training progress**
  - plots for accuracy, loss, validation metrics, and more
- **Automatically validate network performance**
  - stop training when the validation metrics stop improving
- **Perform hyperparameter tuning using Bayesian optimization**

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

Verification using Deep Dream Images

- Visualize what the learned features look like
- Generate images that strongly activate a particular channel of the network layers
- function `deepDreamImage`

Demo : Deep Dream Images Using AlexNet



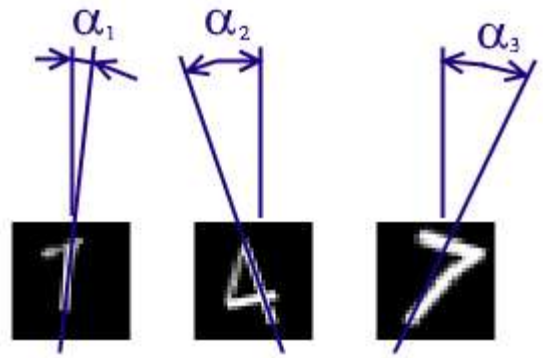
Handling Large Sets of Images

- 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`, `augmentedImageSource`

Deep Learning Models for Regression

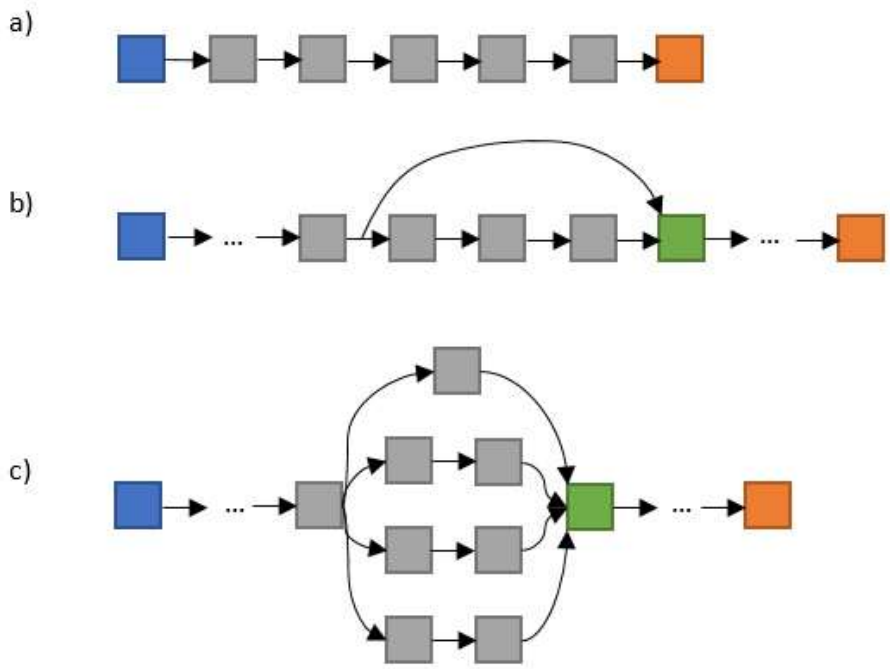
- To predict continuous data such as angles and distances in images
- Include a regression layer at the end of the network

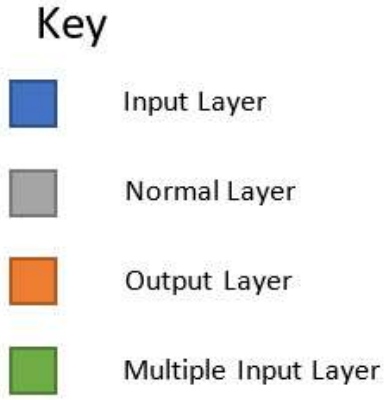
```
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(12,25)
          reluLayer()
          fullyConnectedLayer(1)
          regressionLayer()];
options = trainingOptions('sgdm');
convnet = trainNetwork(trainImages,trainAngles,layers,options);
results = predict(convnet,newImages);
```



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)**

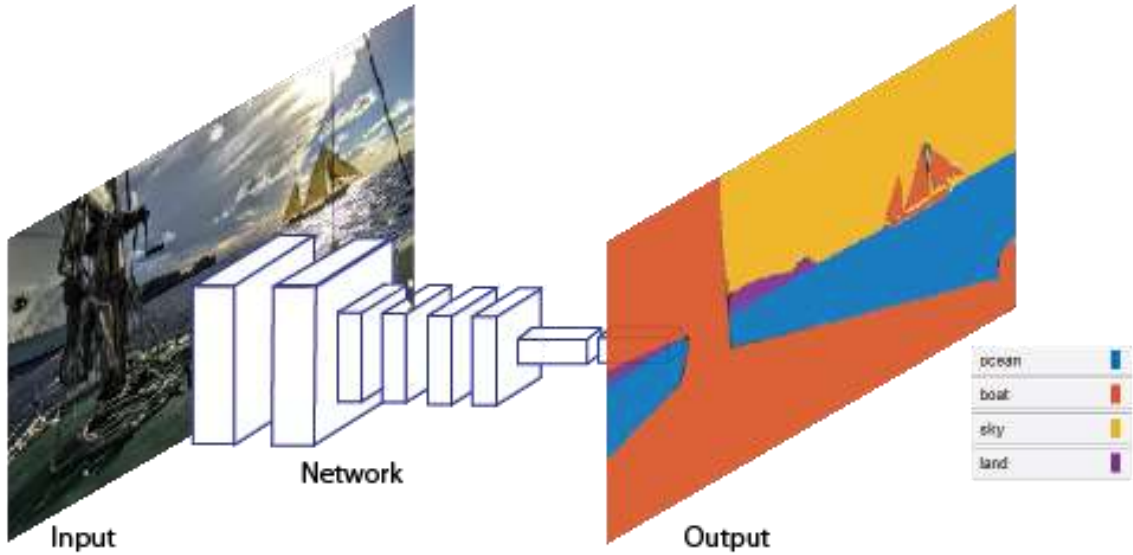
Image Classification vs. Object Detection

- **Image Classification**
  - classify whole image using set of distinct categories
- **Object Detection**
  - recognizing and locating the (small) object in a scene
  - multiple objects in one image

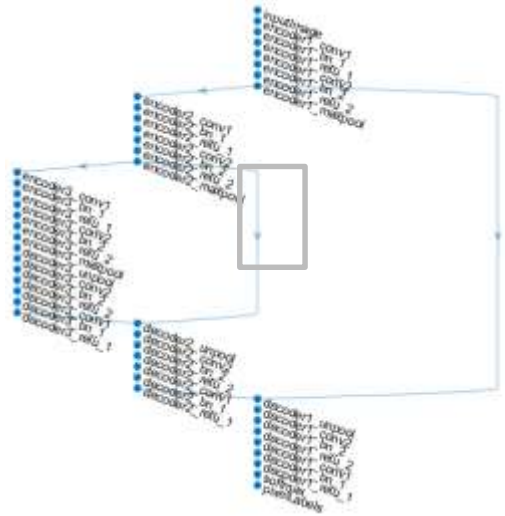
Detector	Function
R-CNN deep learning detector	`trainRCNNObjectDetector`
Fast R-CNN deep learning detector	`trainFastRCNNObjectDetector`
Faster R-CNN deep learning detector	`trainFasterRCNNObjectDetector`

Semantic Segmentation

- **Classify individual pixels**
- **Functions:**
  - **perform semantic segmentation**
    - `semanticseg`
  - **special layers:**
    - `pixelClassificationLayer`, `crop2dLayer`
  - **complete networks:**
    - `segnetLayers`, `fcnLayers`



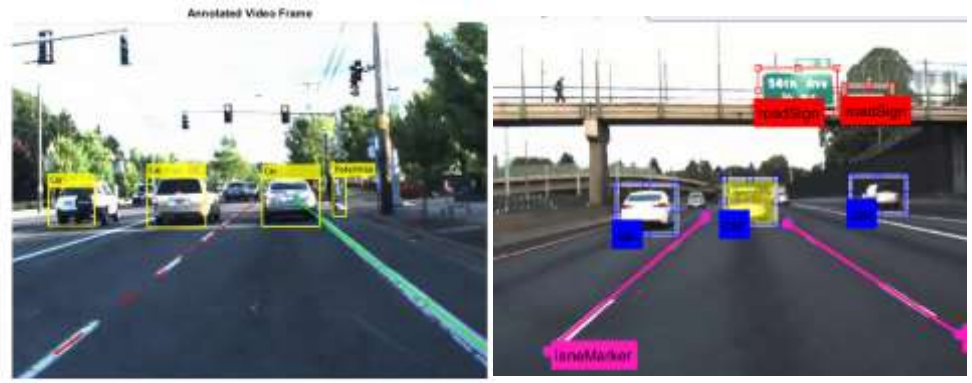
SegNet Convolutional Neural Network

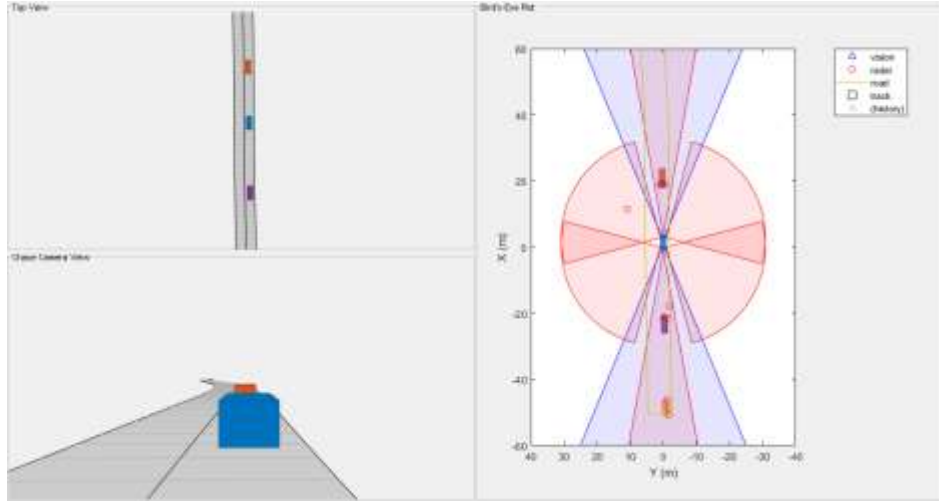


Semantic Segmentation

Automated Driving System 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



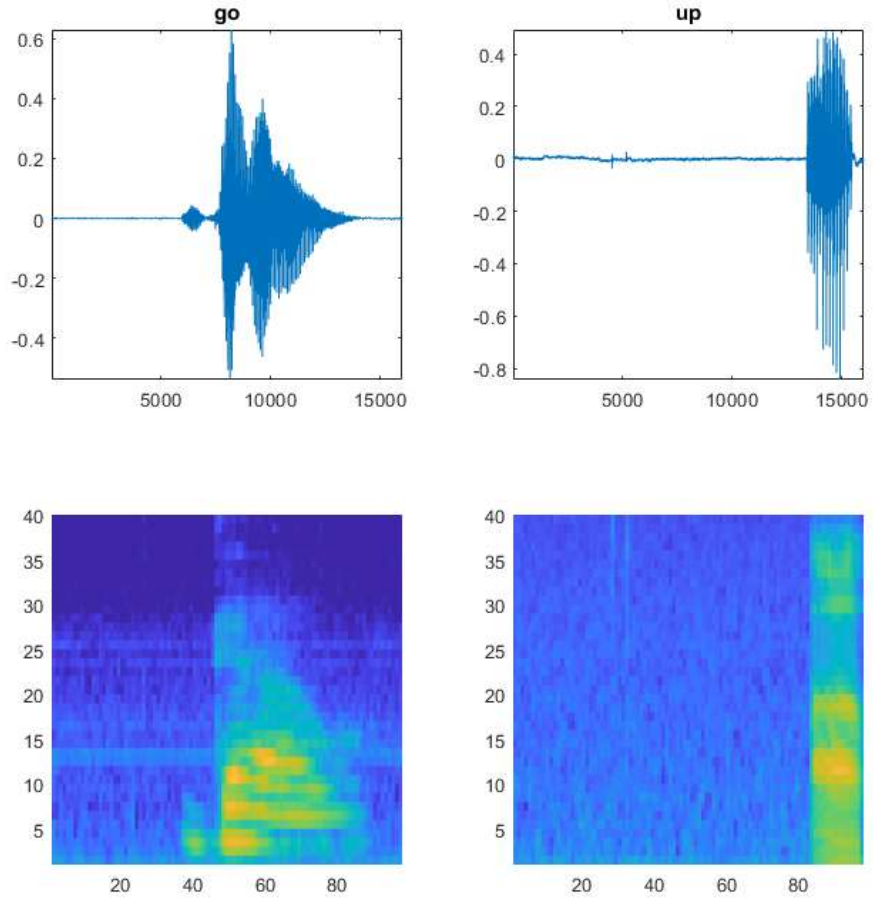


Deep Learning with Time Series and Sequence Data

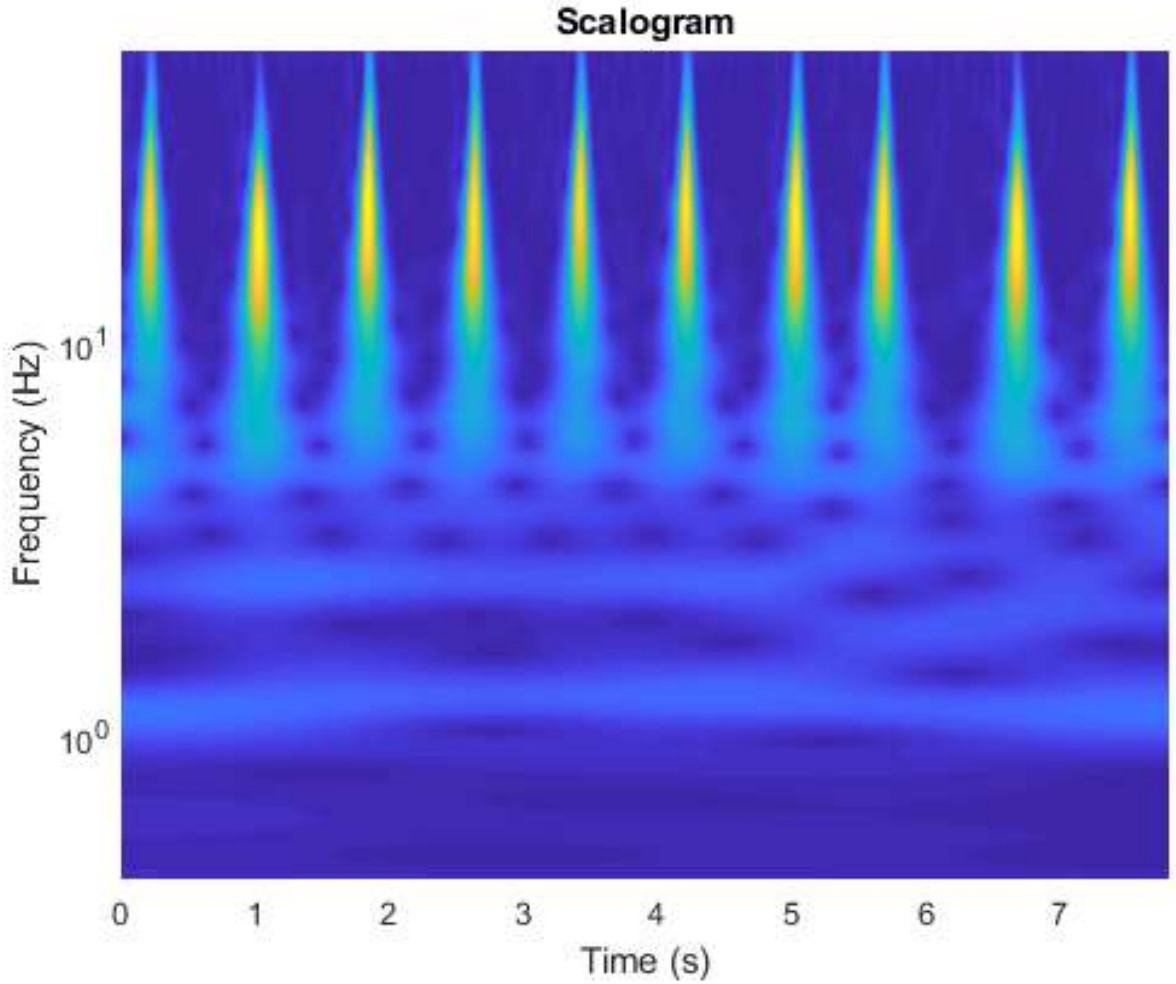
- **Create time-frequency representation of the signal data**
  - *Signal Analyzer* app
  - spectrogram
    - `spectrogram`, `pspectrum`
  - scalogram (continuous wavelet transform)
    - `cwt`

⇨ **time-frequency images**

- **Apply deep neural network to the images**



Demo : Signal Classification with CNN

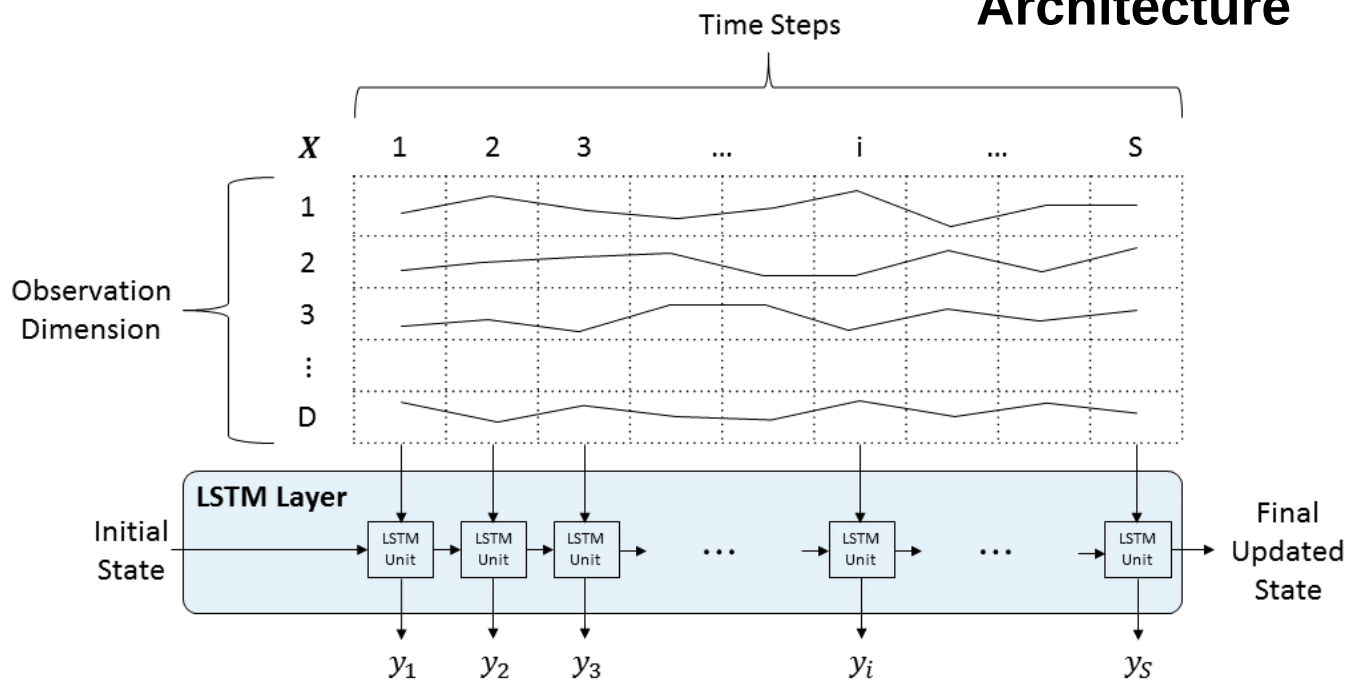


Long Short Term Memory (LSTM) Networks

- **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**
  - `lstmLayer`, `bilstmLayer`

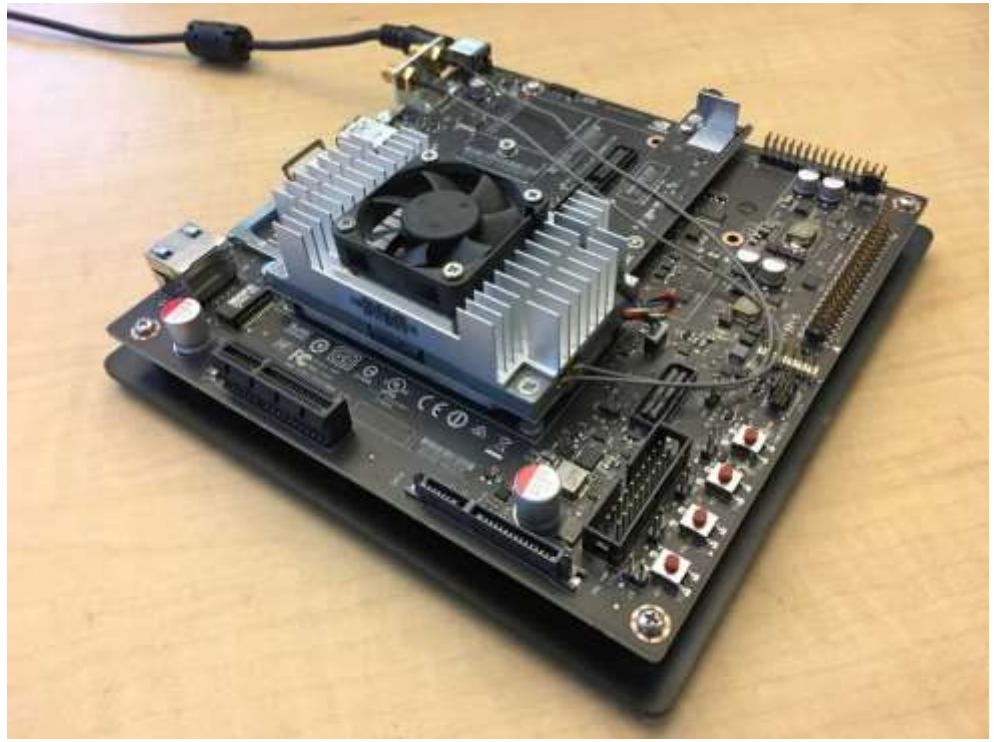
```
layers = [ ...
    sequenceInputLayer(12)
    lstmLayer(100)
    fullyConnectedLayer(9)
    softmaxLayer
    classificationLayer]
```

LSTM Layer Architecture



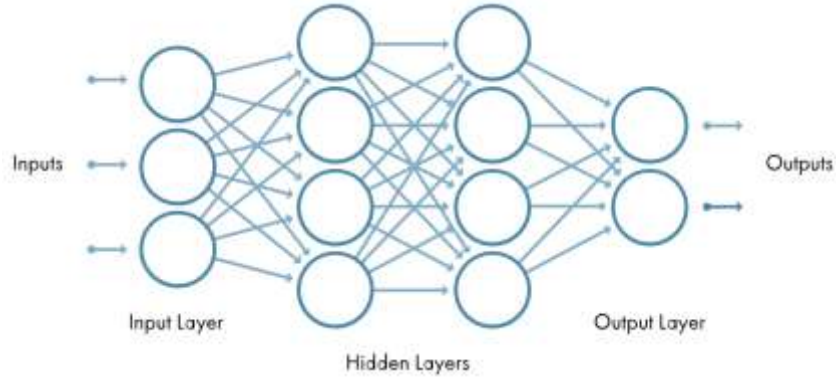
Embedded Deployment - GPU Coder

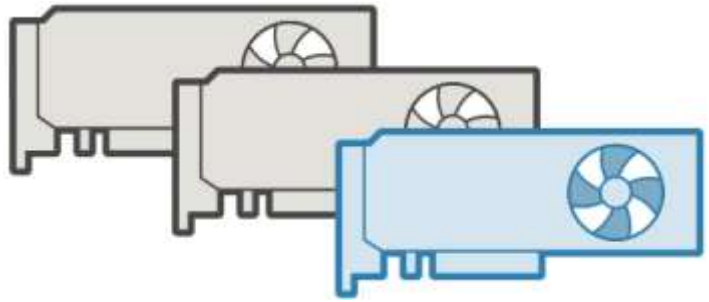
- **Generates optimized CUDA code from MATLAB code**
  - deep learning, embedded vision, and autonomous systems
- **Calls optimized NVIDIA CUDA libraries**
  - cuDNN, cuSolver, and cuBLAS
- **Generate CUDA as:**
  - source code
  - static libraries
  - dynamic libraries
- **Prototyping on GPUs**
  - NVIDIA Tesla® and NVIDIA Tegra®
- **Acceleration using MEX**



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





Děkuji za pozornost