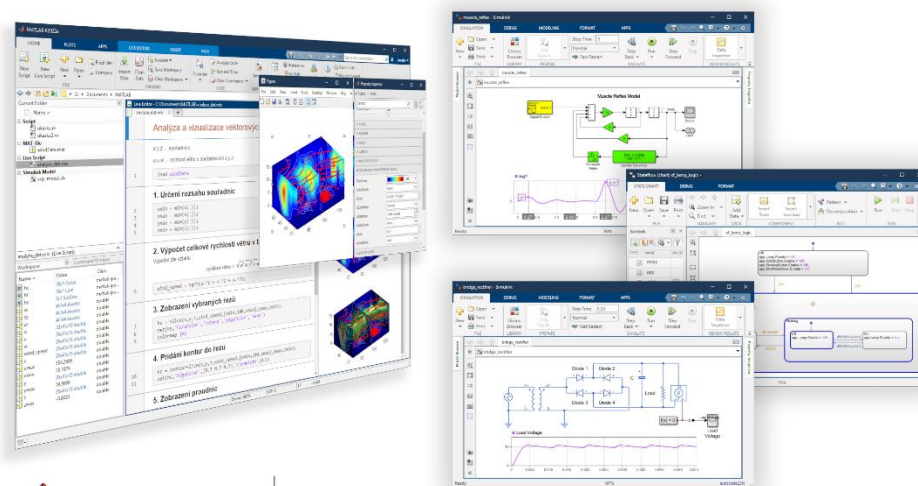


# AI a Model-Based Design

## Vývoj technických systémů využívajících AI

### v prostředí MATLAB a Simulink



Jaroslav Jirkovský  
jirkovsky@humusoft.cz

[www.humusoft.cz](http://www.humusoft.cz)

[info@humusoft.cz](mailto:info@humusoft.cz)

[www.mathworks.com](http://www.mathworks.com)

# Co je MATLAB a Simulink

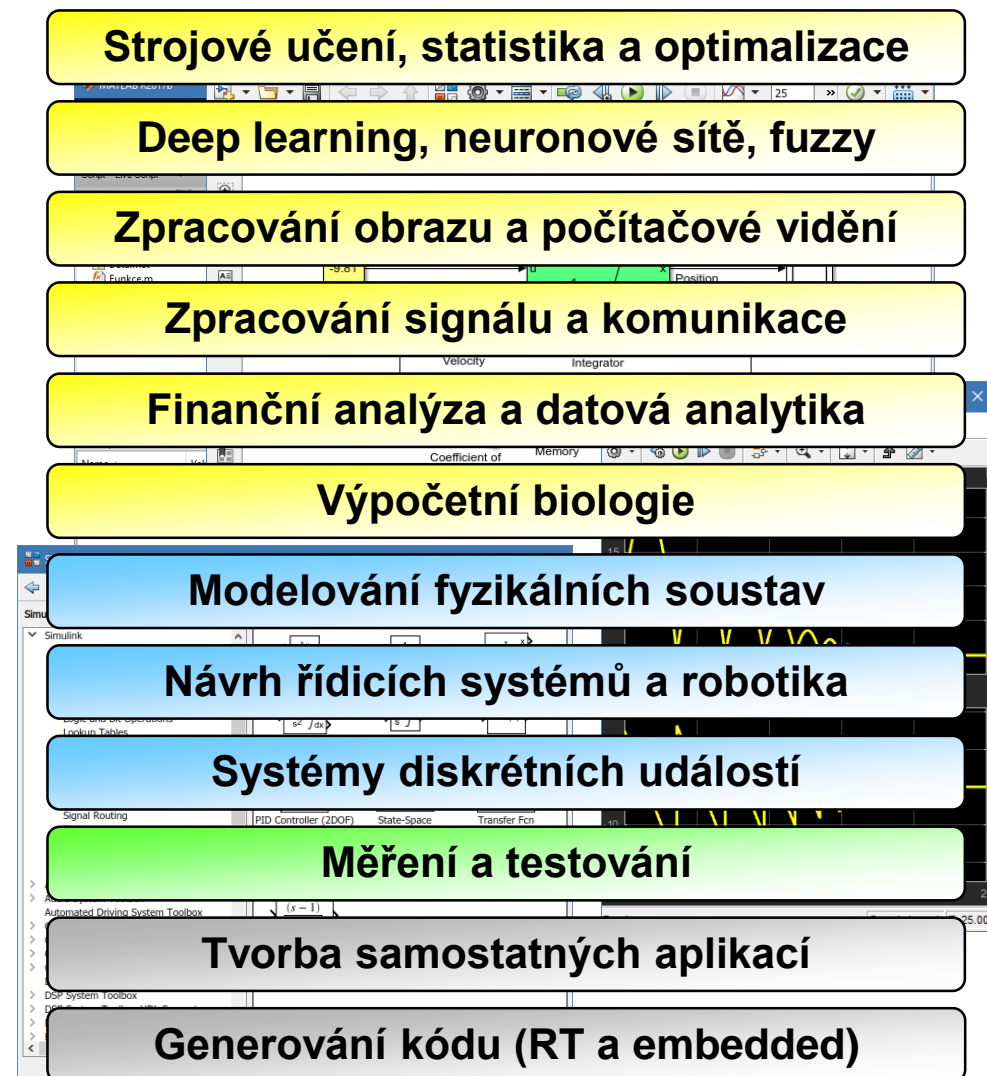
- MATLAB

- inženýrský nástroj a interaktivní prostředí pro vědecké a technické výpočty
- grafické a výpočetní nástroje
- grafické aplikace (APPS)
- otevřený systém

- Simulink

- nadstavba MATLABu
- modelování, simulace a analýza dynamických systémů
- prostředí blokových schémat
- platforma pro Model Based Design

- Aplikační knihovny



# MATLAB a Simulink v průmyslu



Letecký



Automobilový



Biotech



Farmaceutický



Telekomunikace



Oceány a Země



Elektronika



Energetika



Automatizace



Medicína



Těžební



Neurovědy



Aplikovaná fyzika



Železnice a doprava

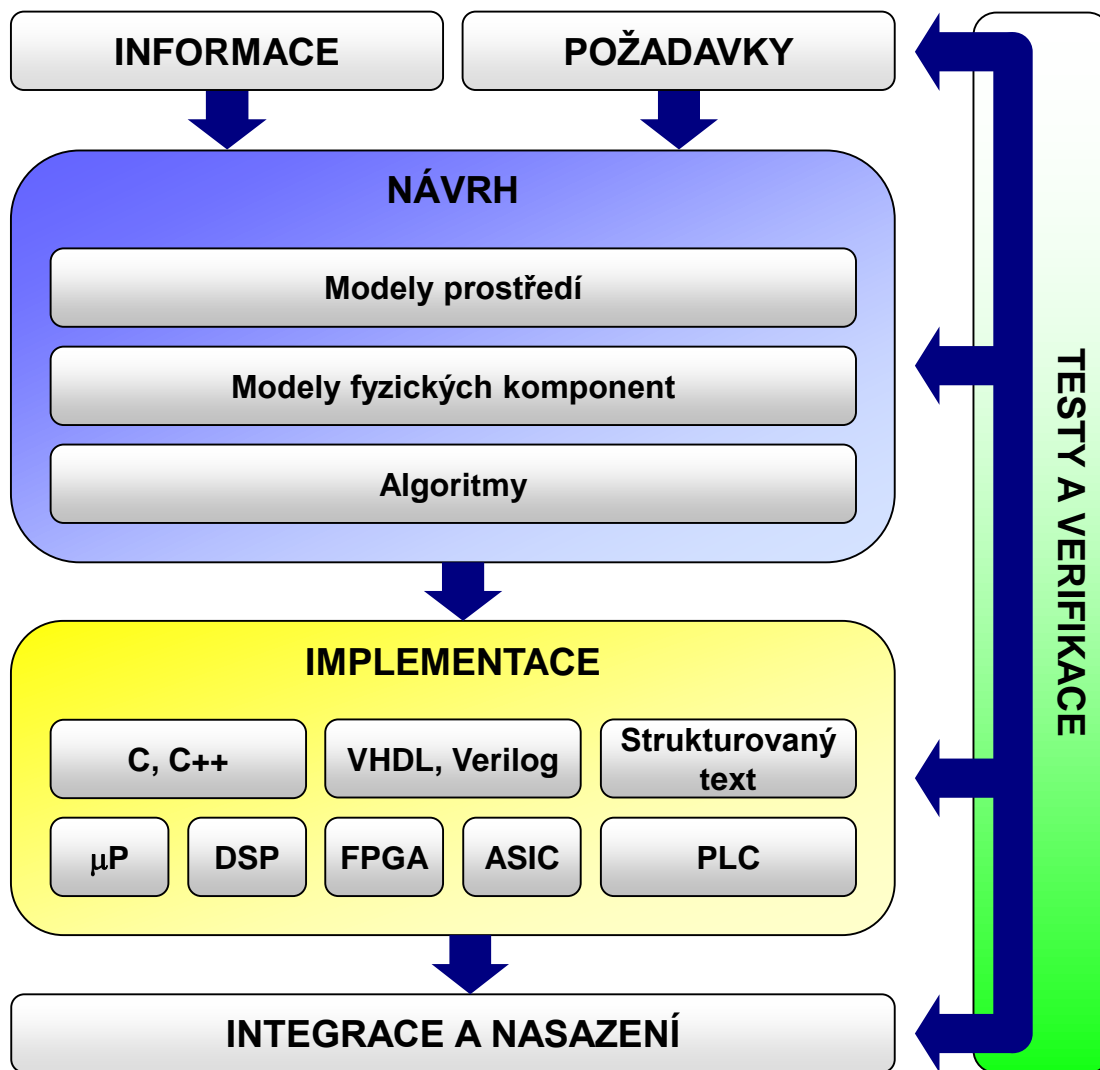


Finanční

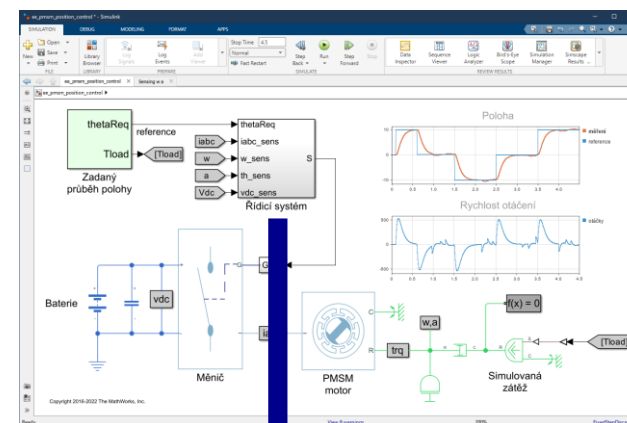


Informatika

# Vývoj metodou Model-Based Design



## Modelování, simulace a testování



Automatické generování kódu

```

Code
Control_System.c
6 * Model version : 6.4
7 * Simulink Coder version : 9.9 (R2023a) 19-Nov-2022
8 * C/C++ source code generated on : Tue Apr 25 11:16:44 2023
9
10 * Target selection: ert-tlc
11 * Embedded hardware selection: Texas Instruments->c2000
12 * Code generation objectives: unspecified
13 * Validation result: not run
14
15
16 #include "Control_System.h"
17 #include <math.h>
18 #include "rt_udefints.h"
19 #include "rtwtypes.h"
20 #include <string.h>
21
22 /* Block signals (default storage) */
23 #blockio_control_system Control_System_B;
24
25 /* Block states (default storage) */
26 #blockstates_control_system Control_System_S;
27
28 /* External inputs (root import signals with default storage) */
29 ExternalInputs_control_System Control_System_U;
30
31 /* External outputs (root outputs fed by signals with default storage) */
32 ExternalOutputs_control_System Control_System_Y;
33
34 /* Real-time model */
35 static #if_HOML_control_System Control_System_M;
36 #if_HOML_control_System *const control_System_M = &Control_System_M;
37
38 /* Model step function */
39 void control_System_step(void)
40 {
41     real32_T B1AS;
42     real32_T uB;
43 }
44
45 _ATAPL282134_1801Podklady/control_System_ert_rtw/control_System.c Ln 6 Col 13
  
```

Nasazení

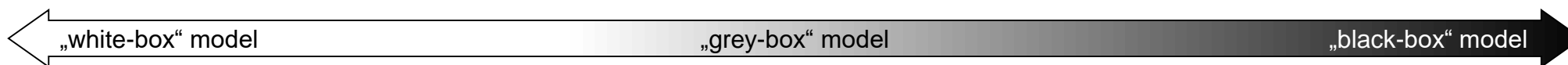


# Přístupy k modelování

- Pro různé situace jsou vhodné různé přístupy

## Fyzikální vztahy

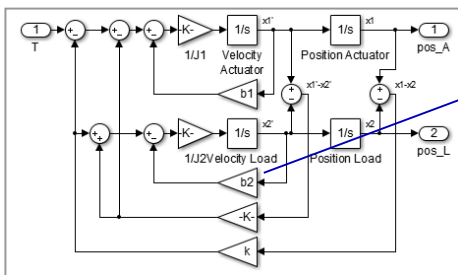
## Naměřená data



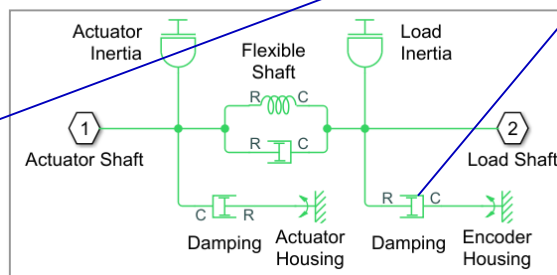
### Modelování rovnic

$$J_1 x_1'' = -b_1 x_1' - k(x_1 - x_2) - b_{12}(x_1' - x_2') + T$$

$$J_2 x_2'' = -b_2 x_2' + k(x_1 - x_2) - b_{12}(x_1' - x_2')$$

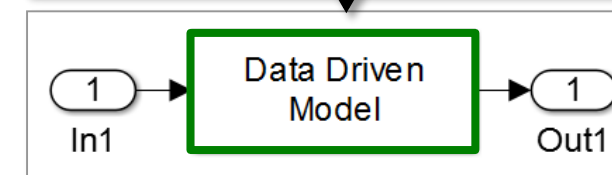
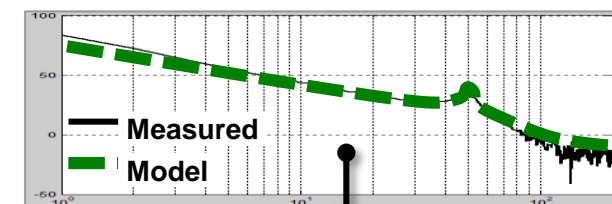


### Fyzikální síť



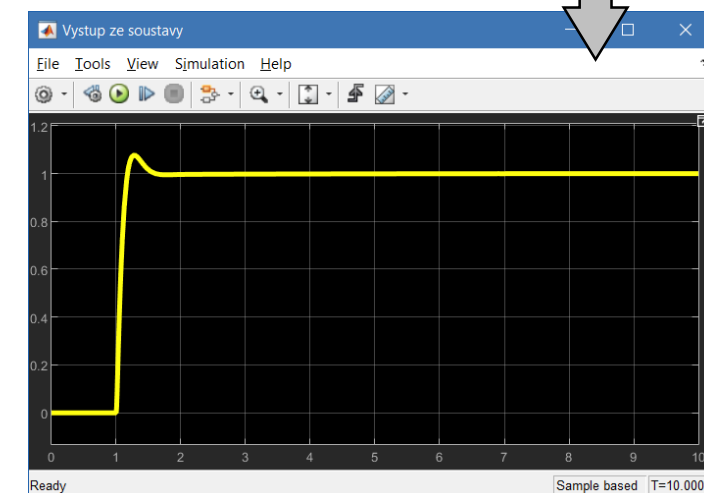
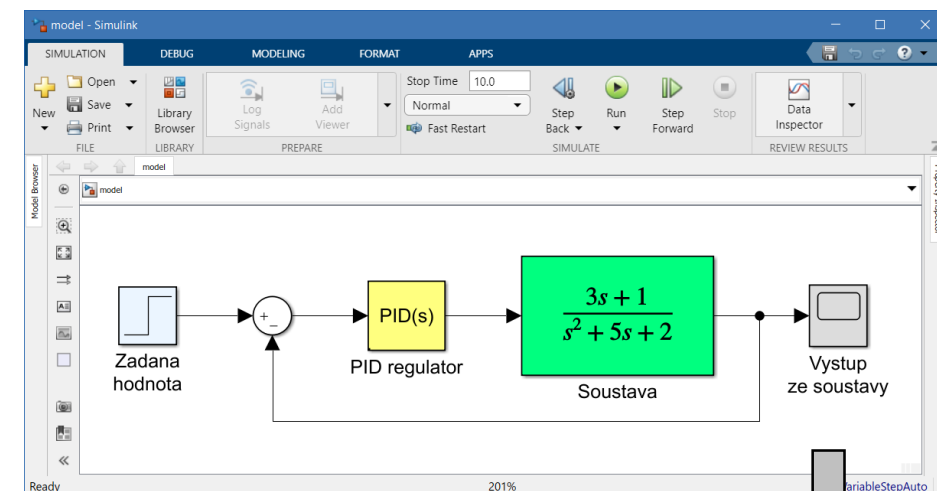
### Ladění neznámých parametrů

### Identifikace soustav



# Modelování algoritmů

- Řídicí systémy
- Zpracování signálu a komunikace
- Zpracování obrazu a počítačové vidění
- a další ...
  
- Společná simulace soustav a algoritmů
  
- Generování kódu pro cílové platformy
  - C/C++, HDL, PLC, CUDA



# AI

## UMĚLÁ INTELIGENCE (AI)

Libovolná technika, která umožňuje strojům napodobit lidskou inteligenci



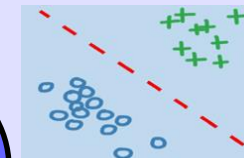
## MACHINE LEARNING

Statistická metoda, která umožní stroji "naučit se" zadanou úlohu na základě dat bez explicitní tvorby programu

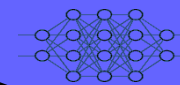
UNSUPERVISED LEARNING  
(neoznačená data)



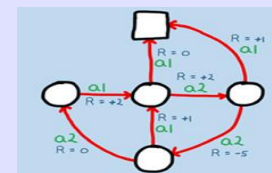
SUPERVISED LEARNING  
(označená data)



DEEP LEARNING  
(Neuronové sítě s mnoha vrstvami)



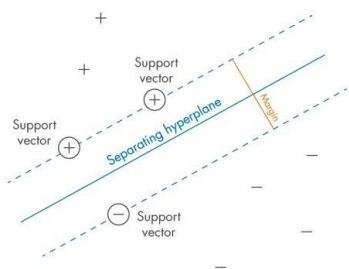
REINFORCEMENT LEARNING  
(data z interakce)



# AI modely v prostředí MATLAB

Machine learning

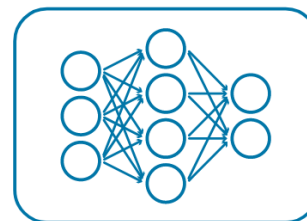
Deep learning



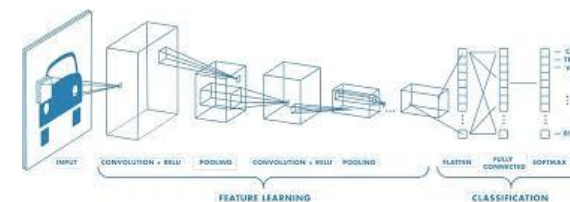
SVM



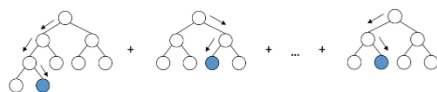
Clustering



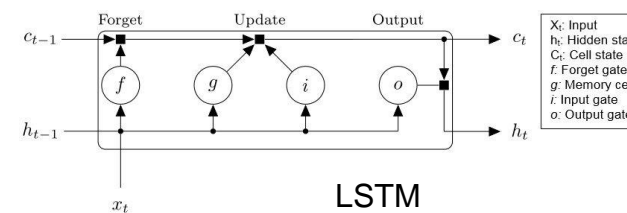
FC



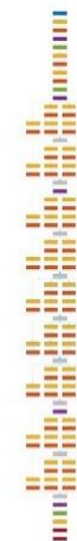
CNN



Decision trees



LSTM





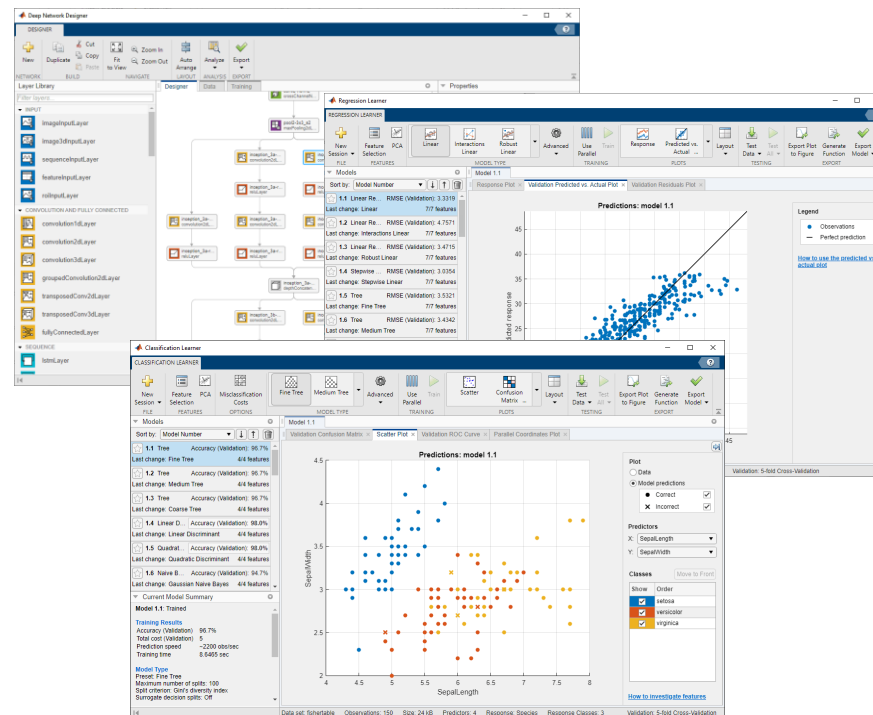
# 3 cesty k vytvoření AI modelu v prostředí MATLAB

```
inputSize = 12;
numHiddenUnits = 100;
numClasses = 9;

layers = [ ...
    sequenceInputLayer(inputSize)
    lstmLayer(numHiddenUnits, 'OutputMode', 'last')
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer]
```

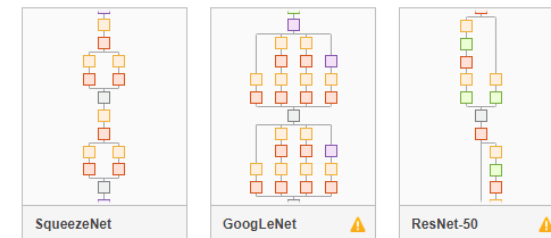
fitcauto / fitrauto

Funkce, skripty



Interaktivní návrh prostřednictvím grafických aplikací

## Image Networks (Pretrained)



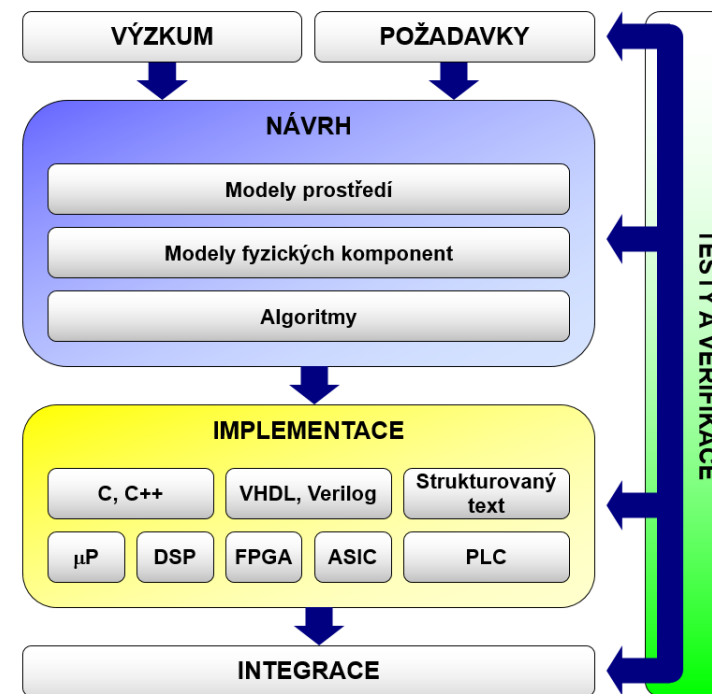
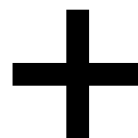
Show more

## Sequence Networks

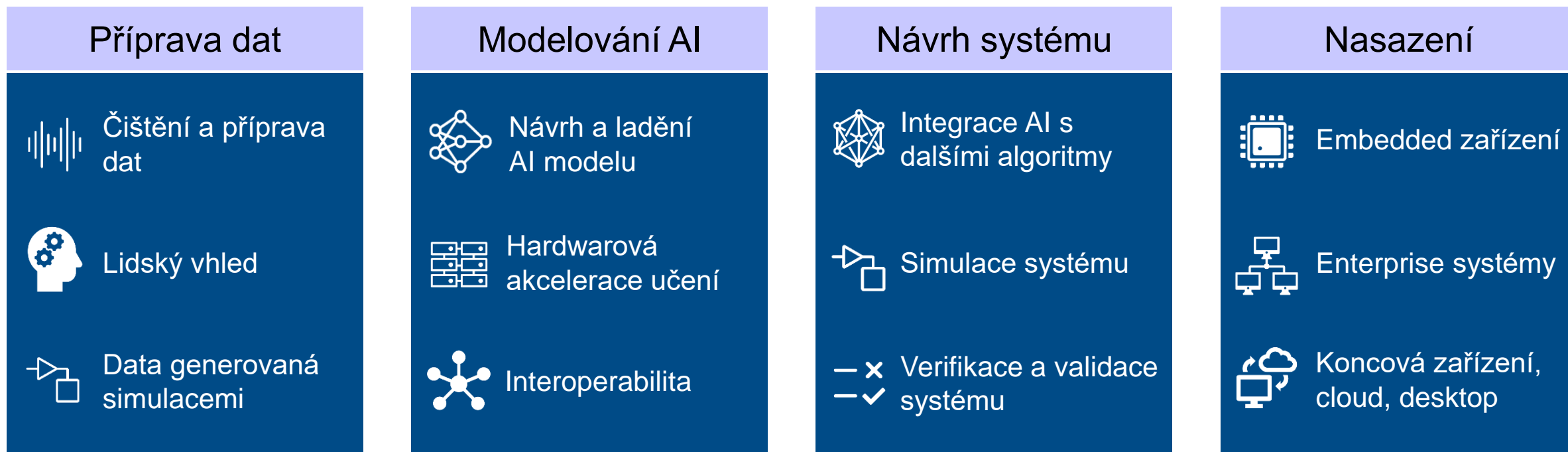


Využit předdefinované sítě a před-učené sítě

# AI a Model-Based Design



# Postup návrhu systému využívajícího AI

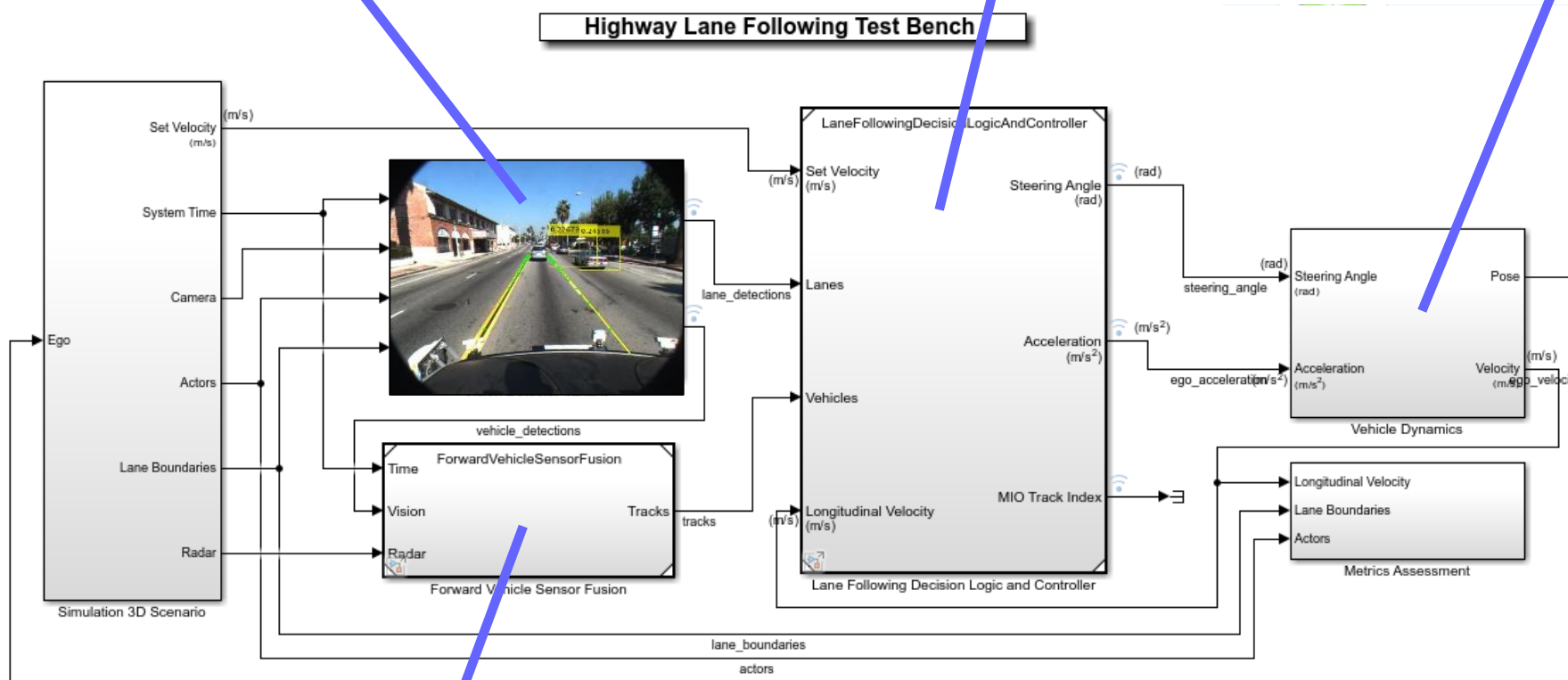


# Typický příklad: Asistenční systém automobilu

AI algoritmus  
detekce jízdnic a vozidel

Řídicí systém

Model dynamiky vozidla  
pro testování chování systému



Senzorická fúze

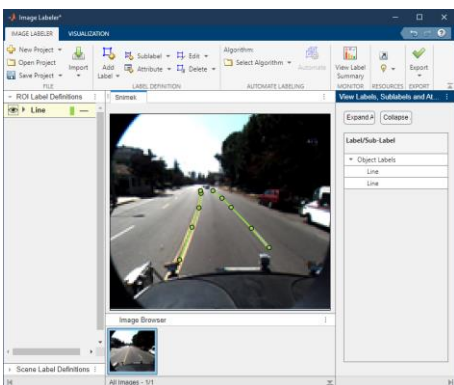
# Typický příklad: Asistenční systém automobilu

Příprava dat

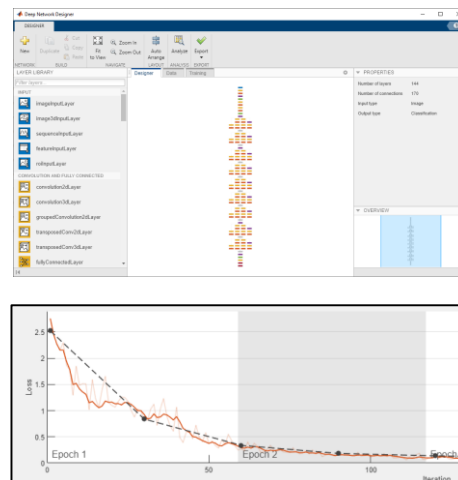
Modelování AI

Návrh systému

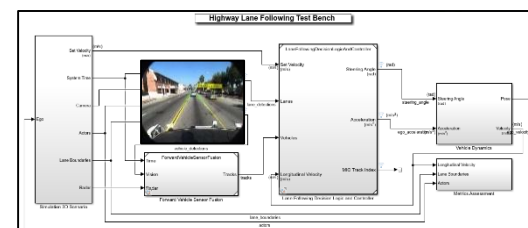
Nasazení



- úprava dat z kamery
- označení jízdních pruhů a vozidel v datech pro učení AI modelů



- výběr a úprava AI modelů (hluboké neuronové sítě)
- učení AI modelů
- ověření AI modelů



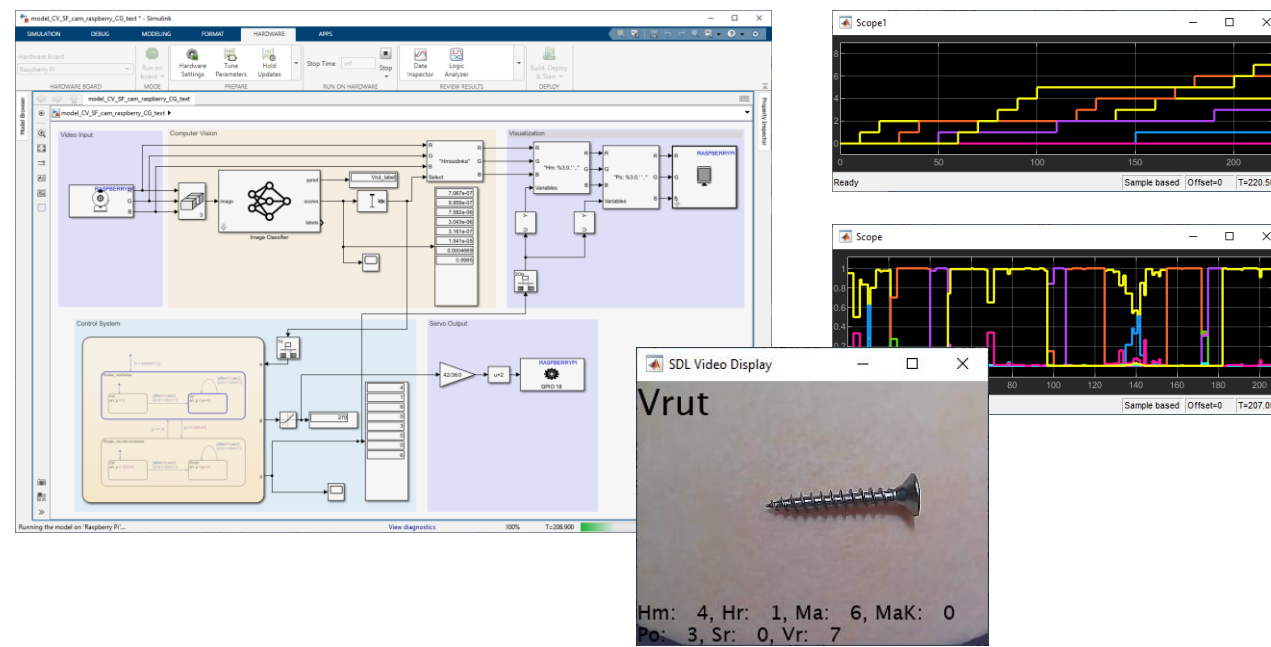
- propojení AI modelů s ostatními algoritmy
- simulace a testování modelu celého software



- generování kódu pro cílovou embedded platformu

# Ukázka: Systém pro klasifikaci a počítání objektů

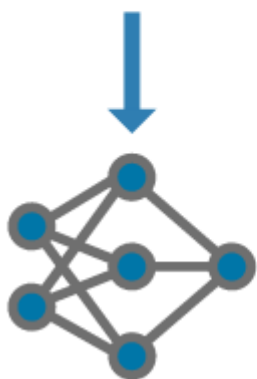
- Klasifikační algoritmus založený na AI
  - blok pro inferenci naučeného modelu z knihovny Deep Learning Toolbox
- Řídicí systém
  - stavový model v jazyce Stateflow
  - řízení otáčení stolu
  - počítání stejných objektů
- Spuštění a nasazení algoritmu
  - v testovacím režimu
  - v režimu samostatné aplikace
- Hardware
  - Raspberry Pi 4, webkamera, servomotor



# Transfer learning

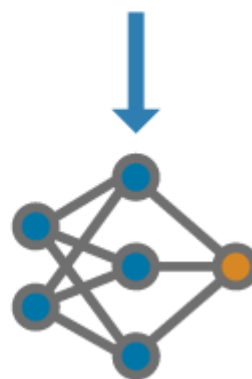
- Využití předdefinovaných a před-učených sítí

datová sada 1



model 1

datová sada 2

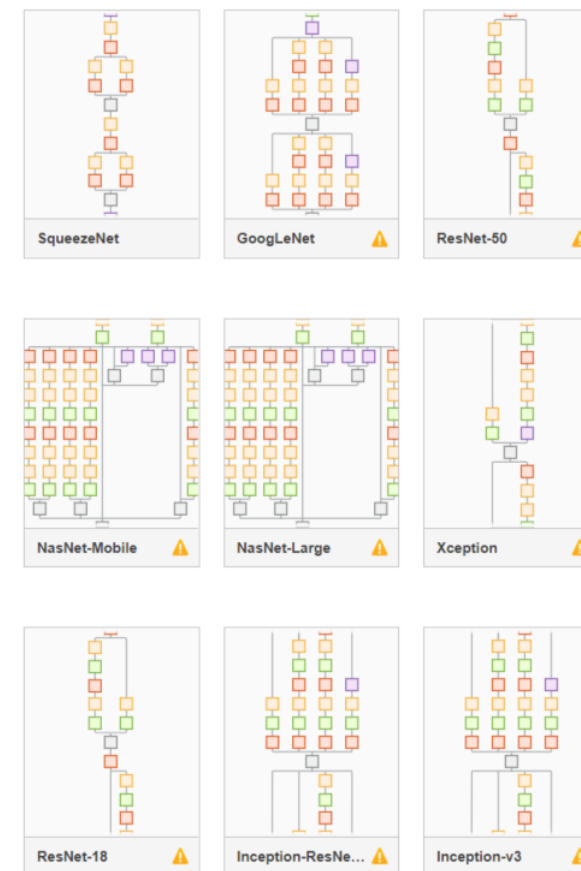


model 2

„znalost“ síť

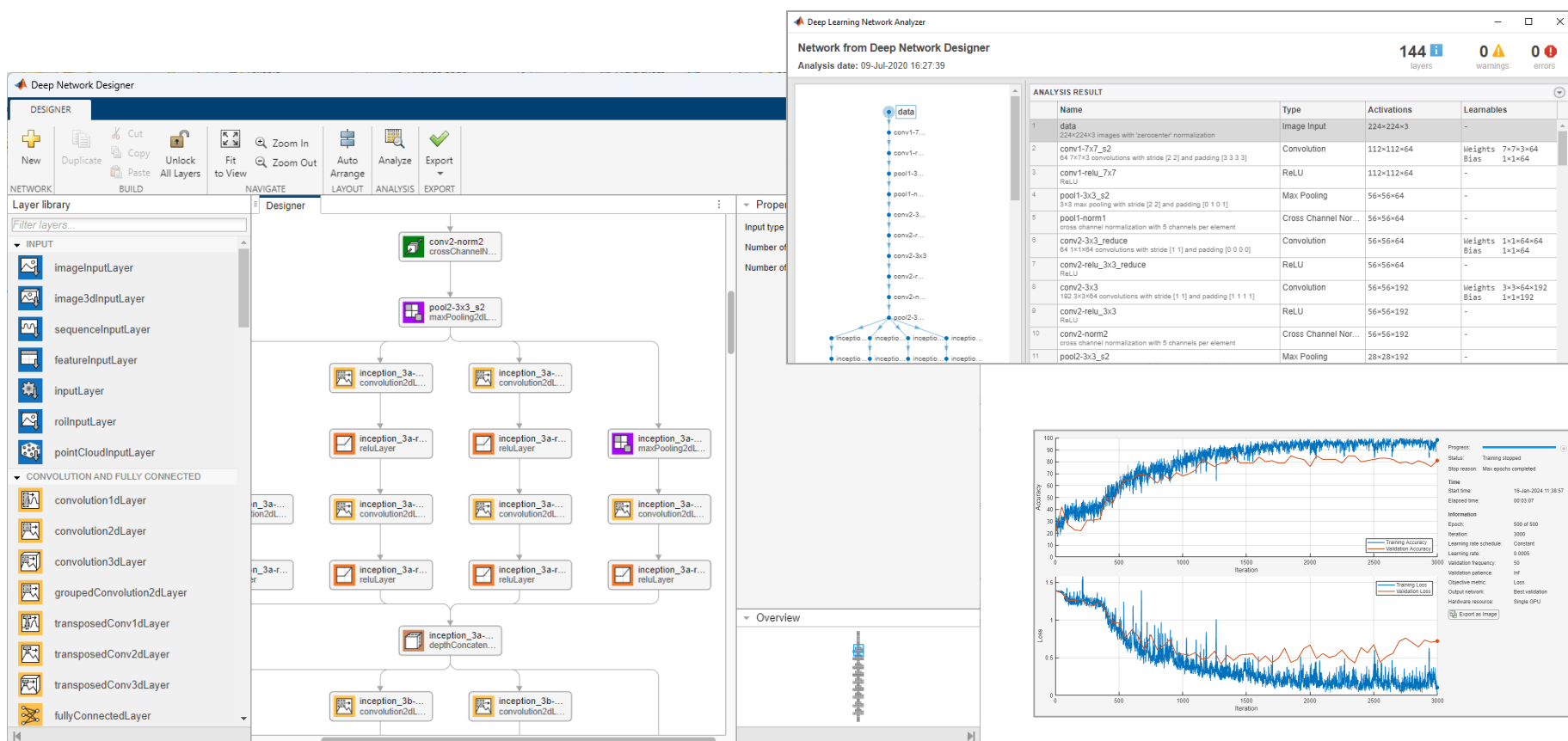


▼ Pretrained Networks



# Deep Network Designer

- Grafická tvorba a úprava sítí pro rychlejší návrh



The screenshot displays the Deep Network Designer interface, which includes a layer library, a central network design canvas, and an analysis window.

**Network Design Canvas:** Shows a hierarchical flowchart of a neural network. The input layer is followed by a convolutional layer with normalization, a pooling layer, and an Inception v3 architecture consisting of multiple parallel branches of convolutional and pooling layers.

**Analysis Window:** Titled "Deep Learning Network Analyzer", it provides a summary of the network's performance. It shows 144 layers, 0 warnings, and 0 errors. The analysis date is 09-Jul-2020 16:27:39.

| Name   | Type                 | Activations | Learnables                         |
|--|----------------------|-------------|------------------------------------|
| 1 data<br>224x224x3 images with 'zerocenter' normalization                           | Image Input          | 224x224x3   | -                                  |
| 2 conv1-7x7_s2<br>64 7x7 convolutions with stride [2 2] and padding [3 3 3 3]        | Convolution          | 112x112x64  | Weights 7x7x3x64<br>Bias 1x1x64    |
| 3 conv1-relu_7x7<br>ReLU   | ReLU                 | 112x112x64  | -                                  |
| 4 pool1-3x3_s2<br>3x3 max pooling with stride [2 2] and padding [0 1 0 1]            | Max Pooling          | 56x56x64    | -                                  |
| 5 pool1-norm1<br>cross channel normalization with 5 channels per element             | Cross Channel Nor... | 56x56x64    | -                                  |
| 6 conv2-3x3_reduce<br>64 1x1x64 convolutions with stride [1 1] and padding [0 0 0 0] | Convolution          | 56x56x64    | Weights 1x3x64x64<br>Bias 1x1x64   |
| 7 conv2-relu_3x3_reduce<br>ReLU  | ReLU                 | 56x56x64    | -                                  |
| 8 conv2-3x3<br>192 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]       | Convolution          | 56x56x192   | Weights 3x3x64x192<br>Bias 1x1x192 |
| 9 conv2-relu_3x3<br>ReLU   | ReLU                 | 56x56x192   | -                                  |
| 10 conv2-norm2<br>cross channel normalization with 5 channels per element            | Cross Channel Nor... | 56x56x192   | -                                  |
| 11 pool2-3x3_s2  | Max Pooling          | 28x28x192   | -                                  |

**Training Progress Graphs:** Two line graphs are shown. The top graph plots Accuracy (0-100) against Iteration (0-3000), showing training accuracy (blue) and validation accuracy (orange) both rising and stabilizing around 80%. The bottom graph plots Loss (0-1.5) against Iteration (0-3000), showing training loss (blue) and validation loss (orange) both decreasing and stabilizing around 0.5.



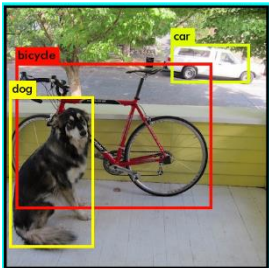
# Začlenění deep learning modelů do Simulinku

- Připravené bloky pro deep learning

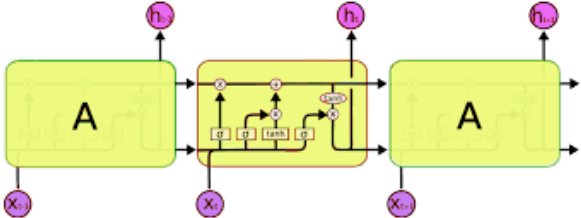


leopard

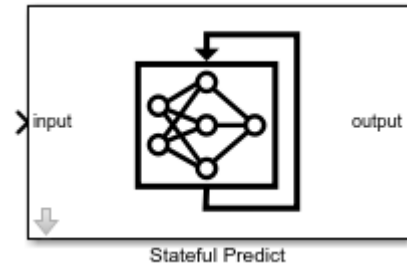
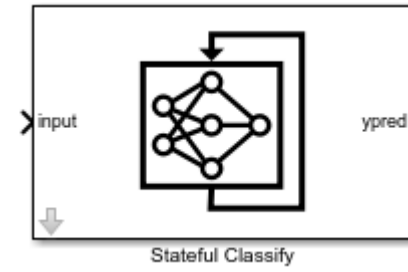
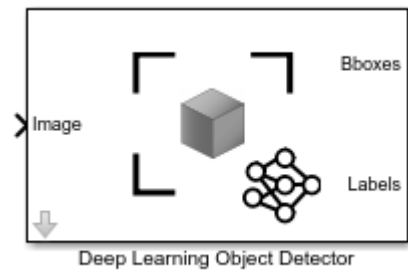
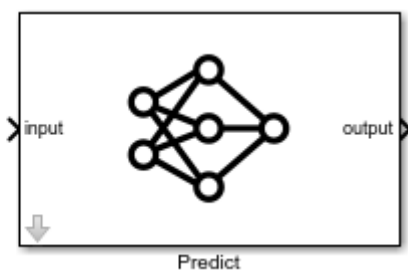
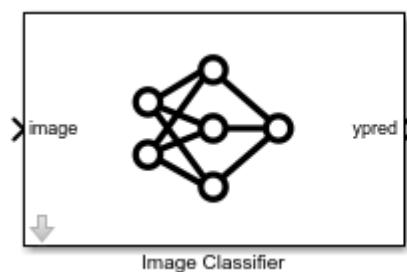
**Klasifikace obrazu,  
sémantická segmentace**



**Detektory objektů**

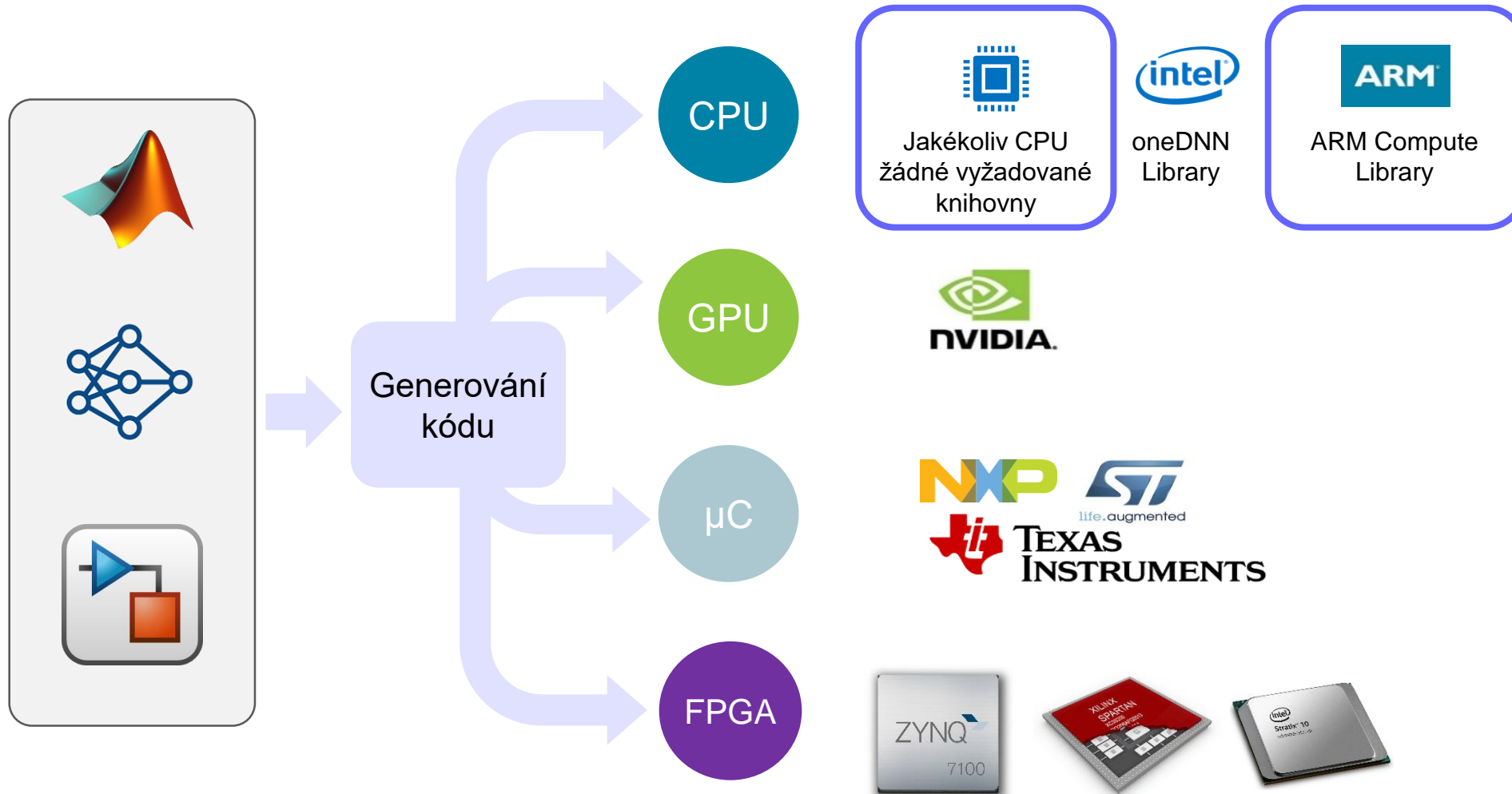


**Sekvenční síť pro zvuková,  
textová a signálová data**



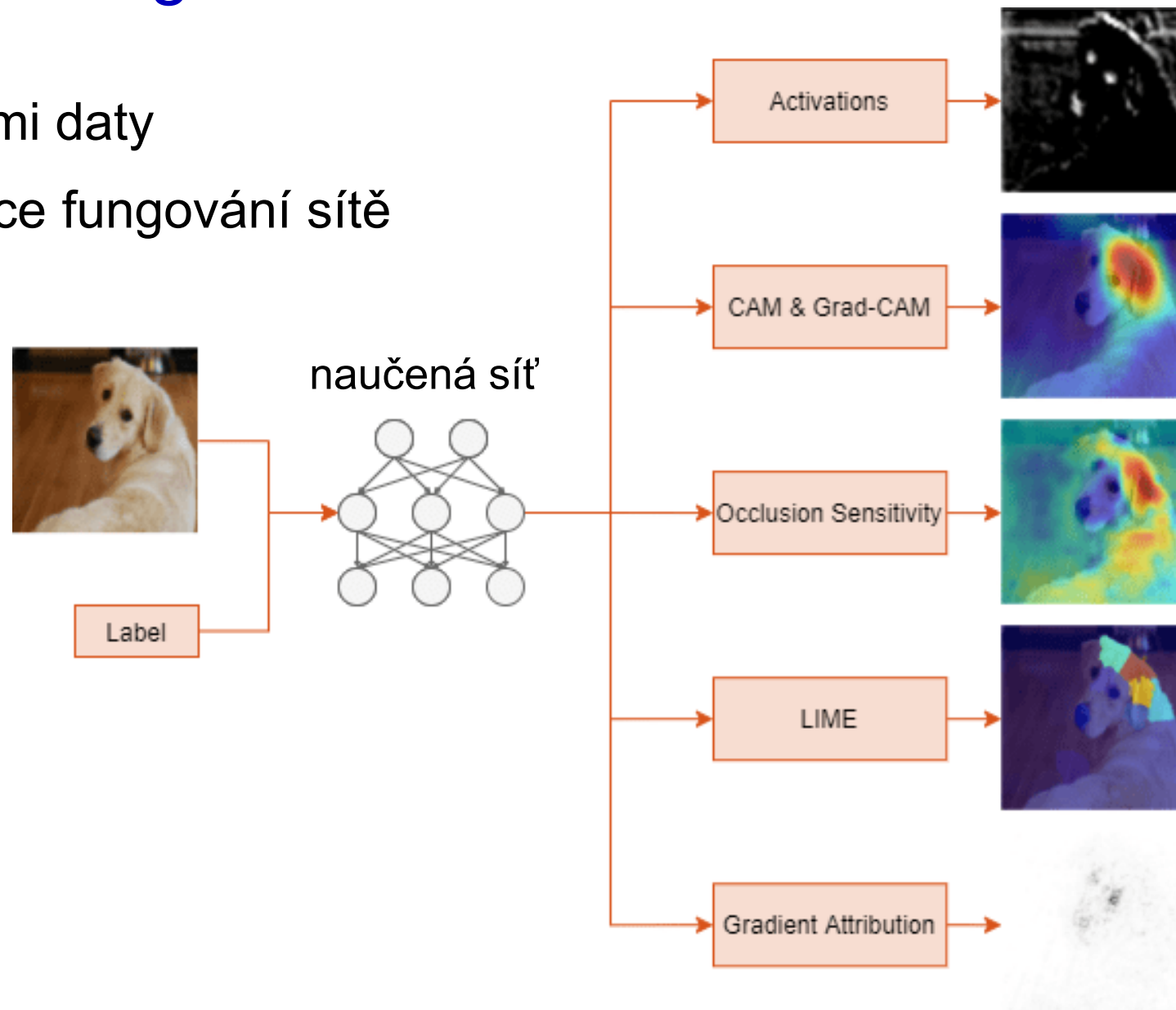
- Blok MATLAB Function

# Nasazení na cílovou platformu



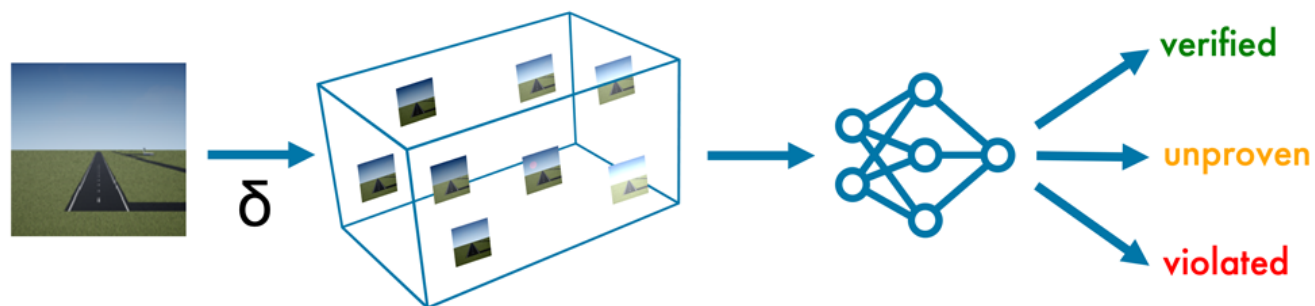
# Ověření správného fungování sítě

- Testování sítě s novými daty
- Vysvětlení a vizualizace fungování sítě



# Verifikace AI modelů

- Deep Learning Toolbox Verification Library
  - cíl: zajistit robustnost a spolehlivost hlubokých neuronových sítí
- Formální metody pro ověření robustnosti
  - proti tzv. adversarial examples

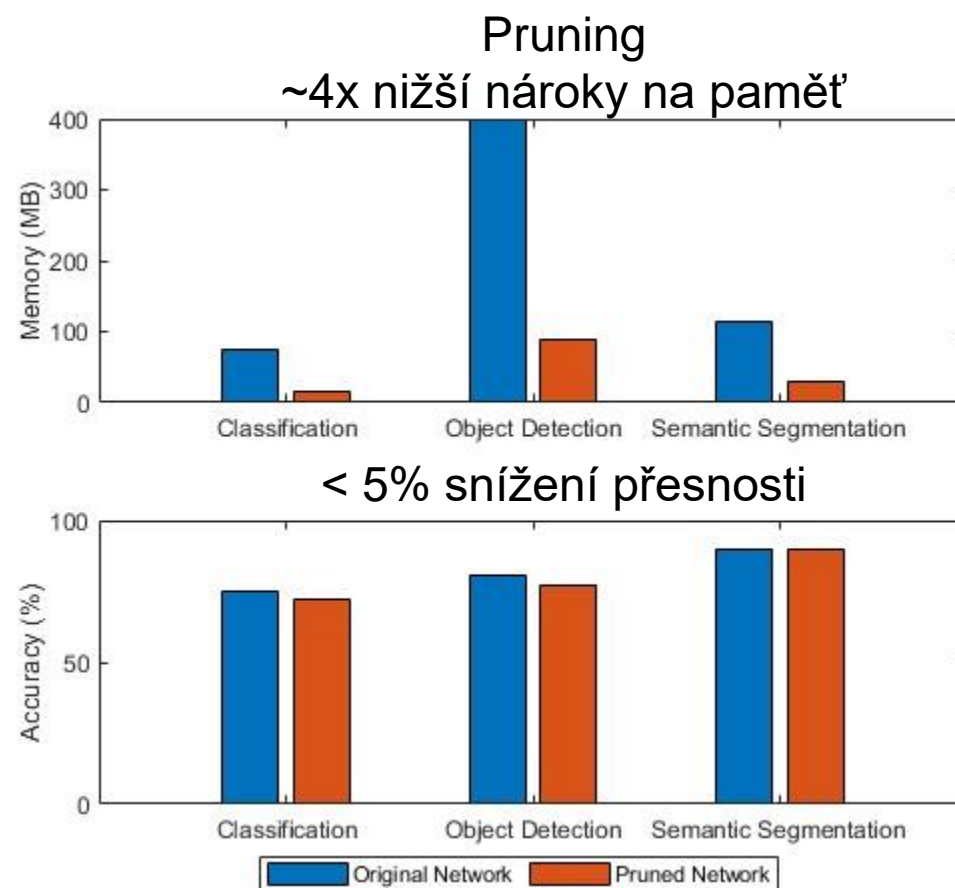


- Odhad citlivosti sítě na drobné změny vstupu
- Rozdělení dat na in-distribution a out-of-distribution
  - možnost sledování za běhu nasazené sítě

# Redukce modelu

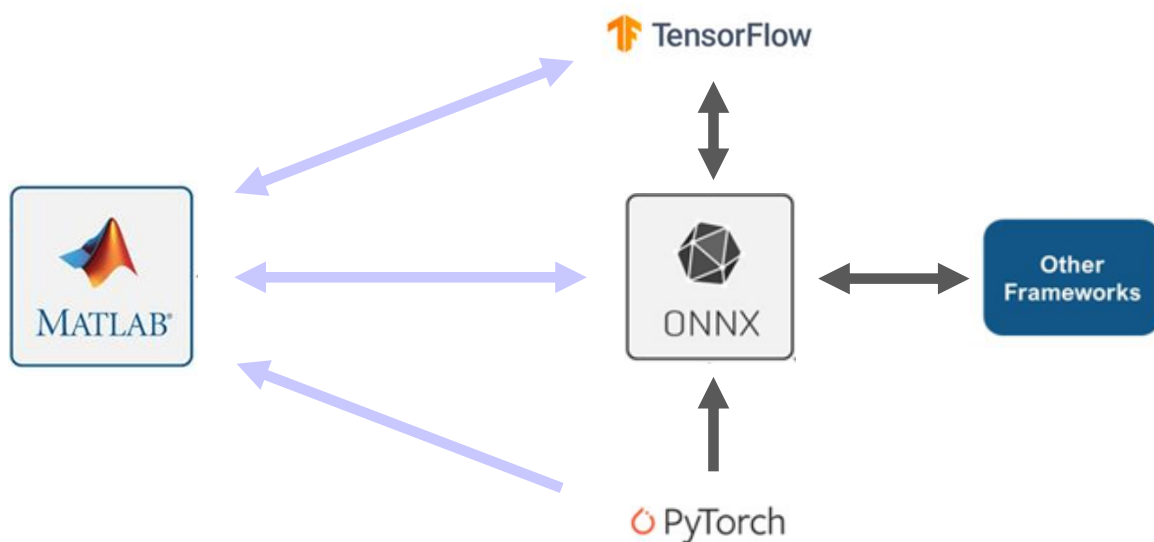
- Snížení paměťových a výpočetních nároků nasazeného modelu
- **Quantizace**
  - převod z floating point do fixed point aritmetiky
- **Pruning**
  - odstranění nedůležitých částí sítě

|  |        |
|--|--------|
| Deep Network Quantization App                  | R2020a |
| Taylor Pruning                                 | R2022a |
| LSTM Pruning                                   | R2022b |
| yolov3 and yolov4 object detector quantization | R2023a |



# MATLAB a spolupráce s dalšími nástroji

- Import modelů z prostředí TensorFlow, PyTorch
- Import modelů v otevřeném formátu ONNX



|   |               |
|---|---------------|
| TensorFlow-Keras Import   | <b>R2017b</b> |
| ONNX Converter (Import & Export)                                  | <b>R2018a</b> |
| TensorFlow Converter (Import)                                     | <b>R2021a</b> |
| TensorFlow Converter (Export)                                     | <b>R2022b</b> |
| PyTorch Converter (Import)  | <b>R2022b</b> |
| More layers, functions, operators, models, and versions supported | <b>R2023a</b> |

Děkuji za pozornost.