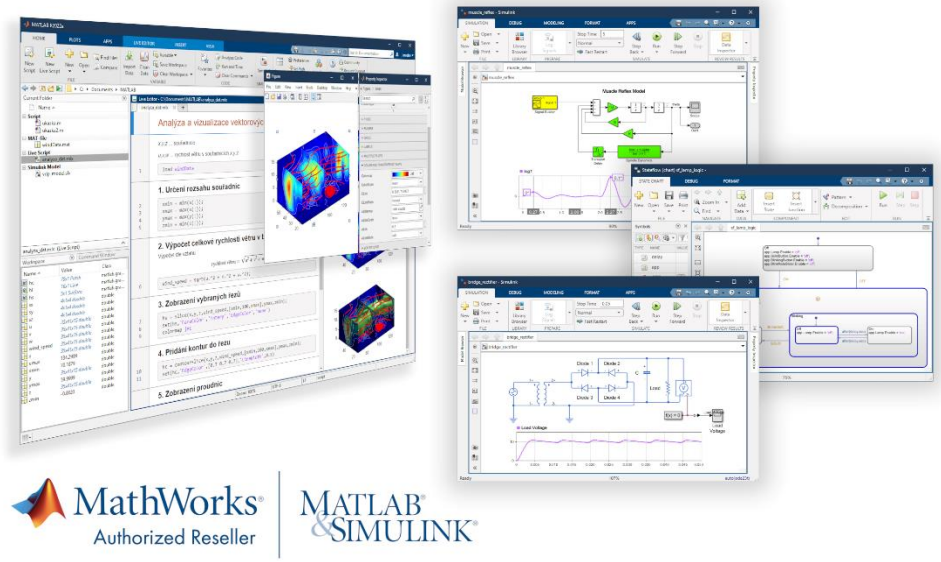
HUMUSOFT®

Master Class: MATLAB a Simulink pro vývoj systémů využívajících AI



Jaroslav Jirkovský

jirkovsky@humusoft.cz

www.humusoft.cz
info@humusoft.cz

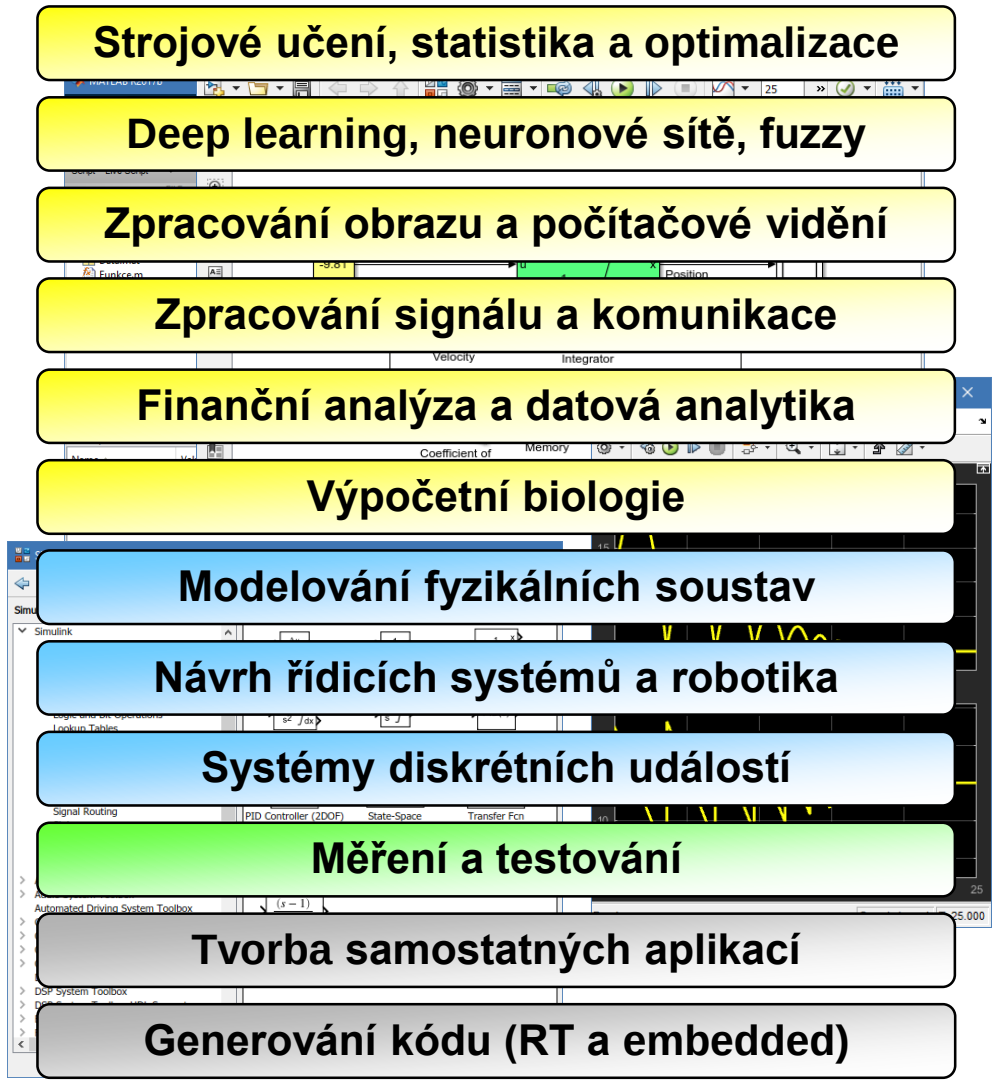
www.mathworks.com

Obsah

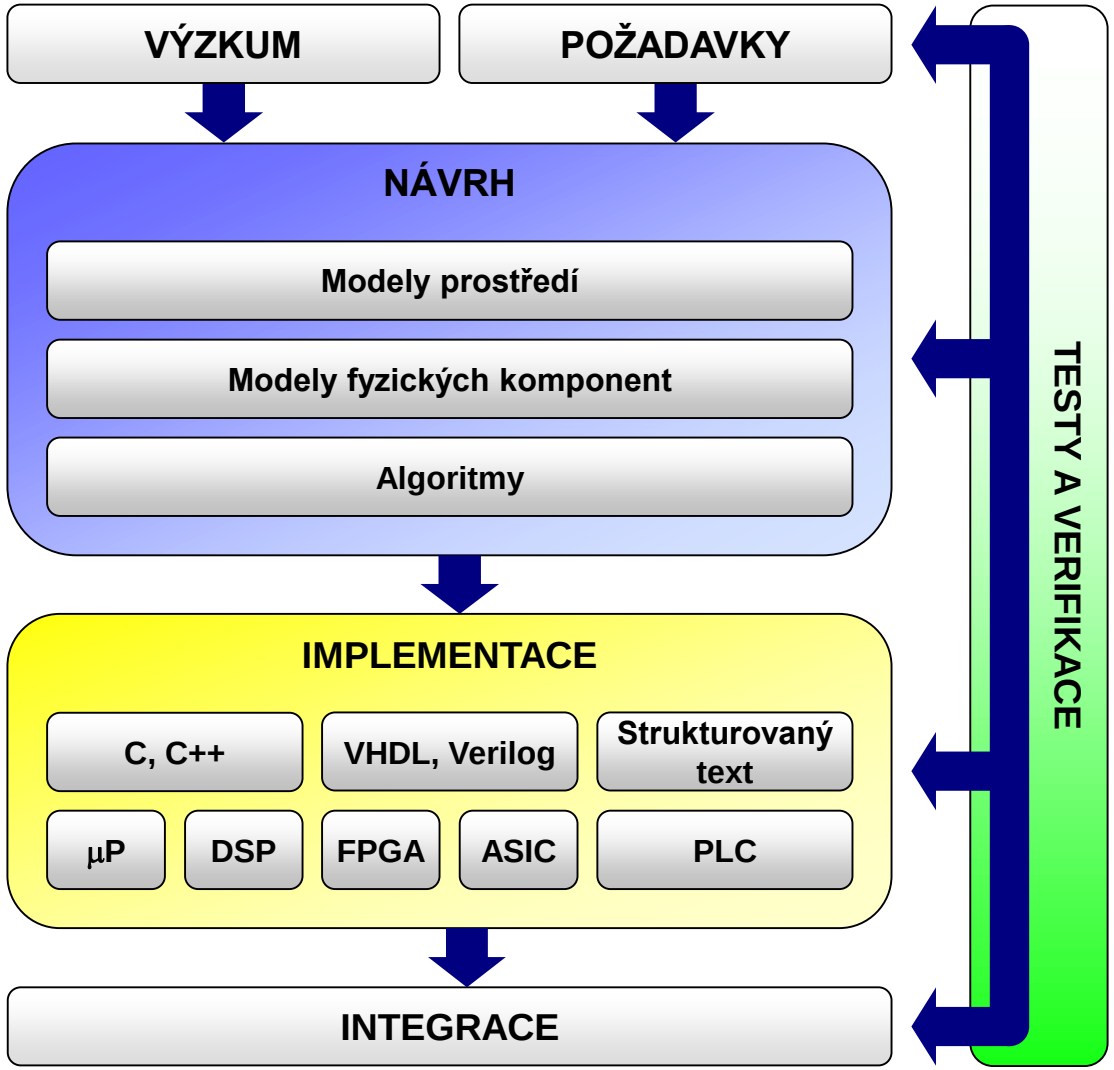
- Metoda Model-Based Design
  - co je metoda Model-Based Design
- Machine learning a deep learning
  - jak vytvořit a naučit model založený na umělé inteligenci
  - integrace modelu s ostatními algoritmy
- Nasazení algoritmů na embedded platformy
  - přizpůsobení modelu umělé inteligence požadavkům embedded platforem
  - automatické generování kódu pro embedded hardware

Co je MATLAB a Simulink

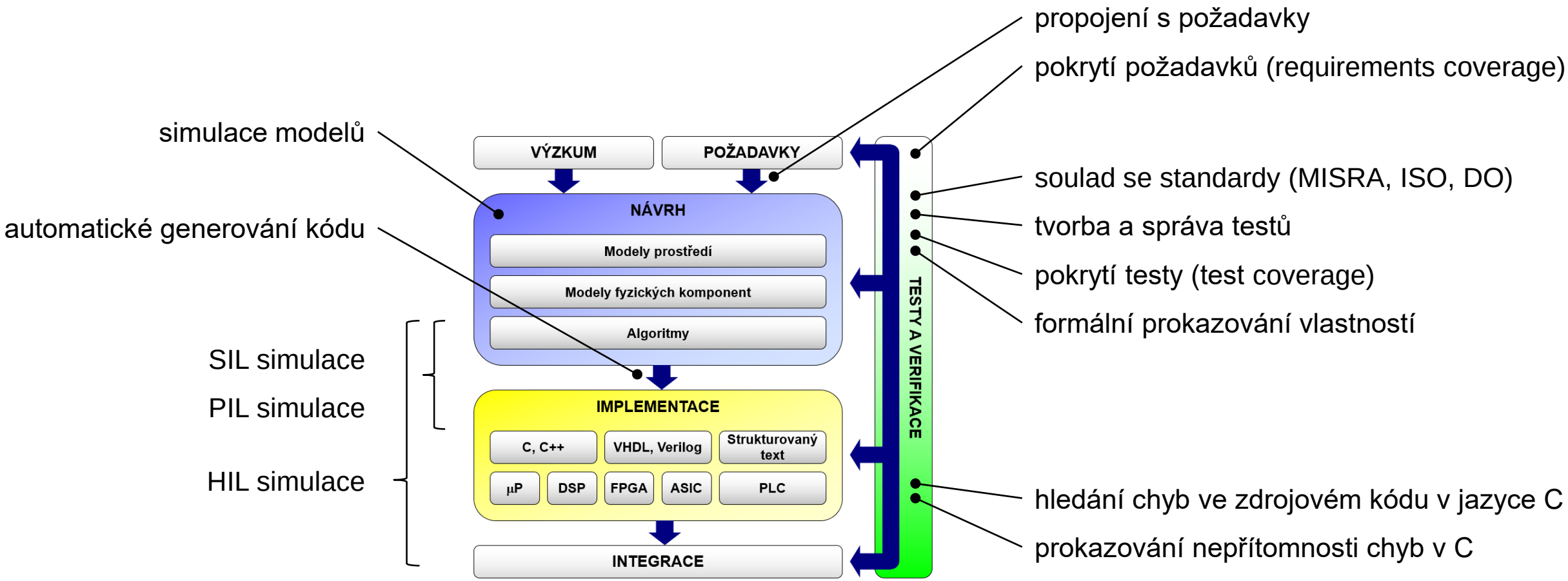
- MATLAB
  - inženýrský nástroj a interaktivní prostředí pro vědecké a technické výpočty
  - grafické a výpočetní nástroje
  - grafické aplikace (APPS)
  - otevřený systém
- Simulink
  - nadstavba MATLABu
  - modelování, simulace a analýza dynamických systémů
  - prostředí blokových schémat
  - platforma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



Vývoj metodou Model-Based Design



Vývoj metodou Model-Based Design



Systém

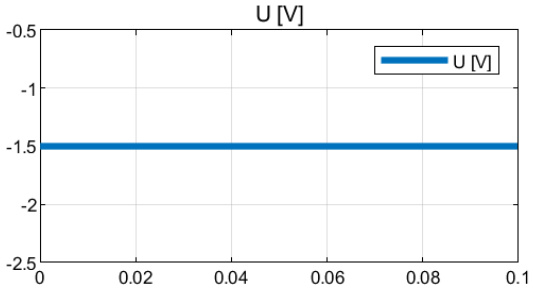
SYSTÉM

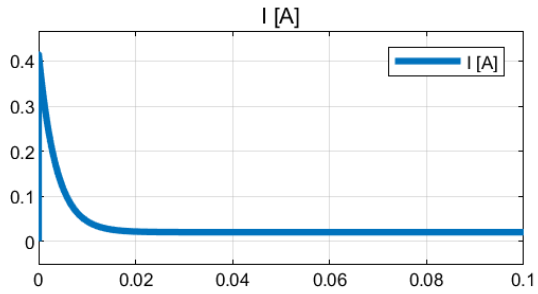
vstup

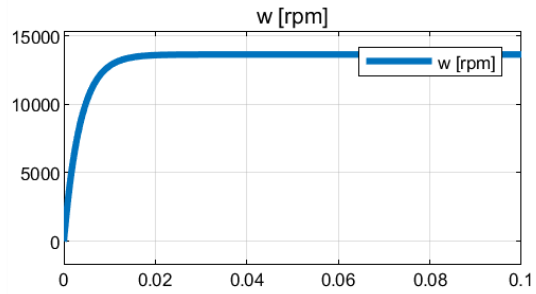
napětí U

výstup

proud I
otáčky ω







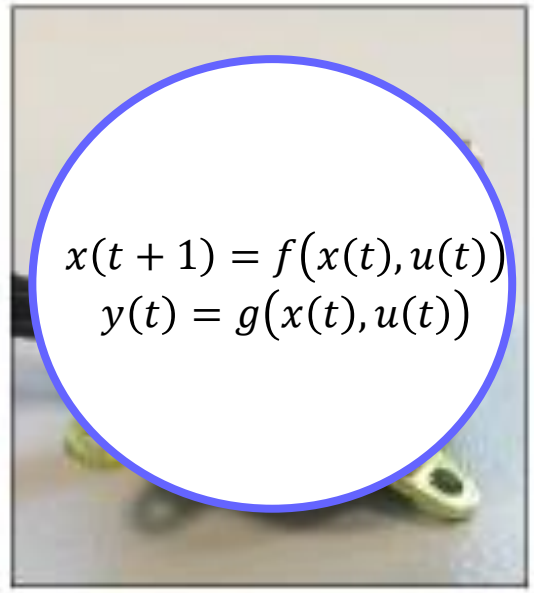
Matematický model systému

MODEL

u

vstup

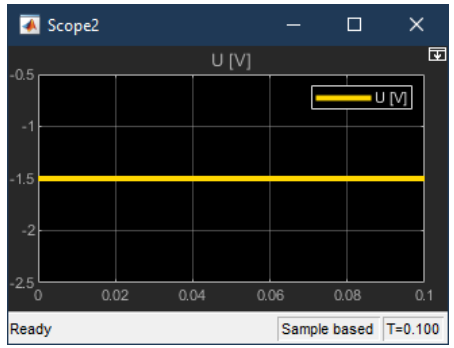
napětí U

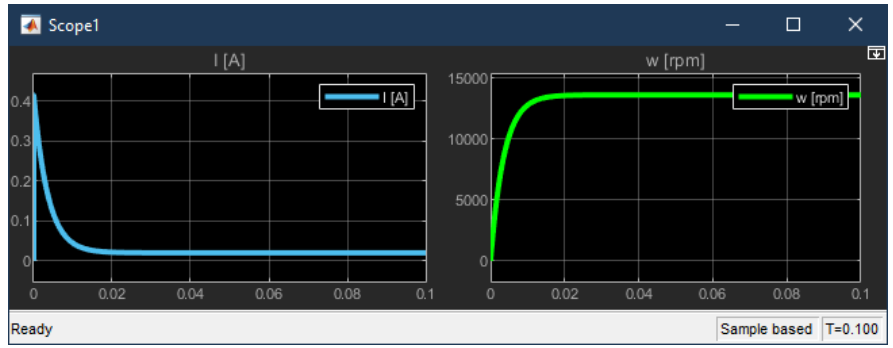


y

výstup

proud I
otáčky ω





Přístupy k modelování

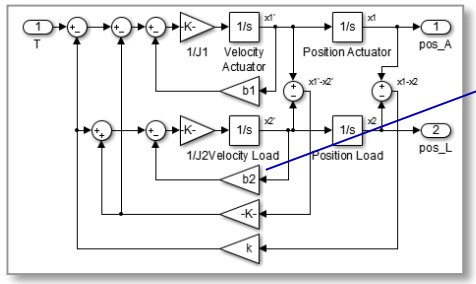
- Pro různé situace jsou vhodné různé přístupy

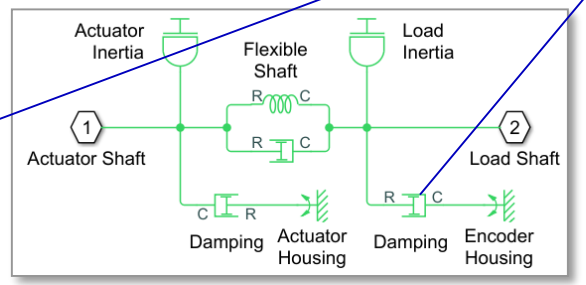
Fyzikální vztahy **Naměřená data**

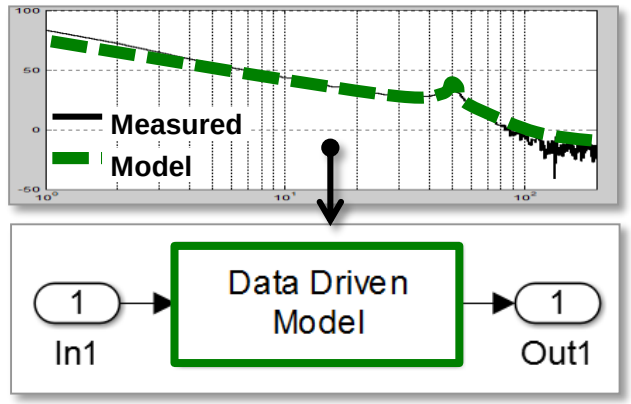
„white-box“ model „grey-box“ model „black-box“ model

Modelování rovnic **Fyzikální sítě** **Ladění neznámých parametrů** **Identifikace soustav**

$$J_1 x_1'' = -b_1 x_1' - k(x_1 - x_2) - b_{12}(x_1' - x_2') + T$$
$$J_2 x_2'' = -b_2 x_2' + k(x_1 - x_2) - b_{12}(x_1' - x_2')$$

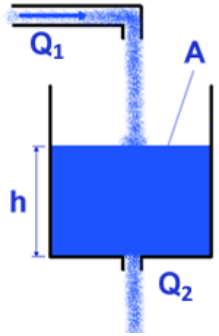


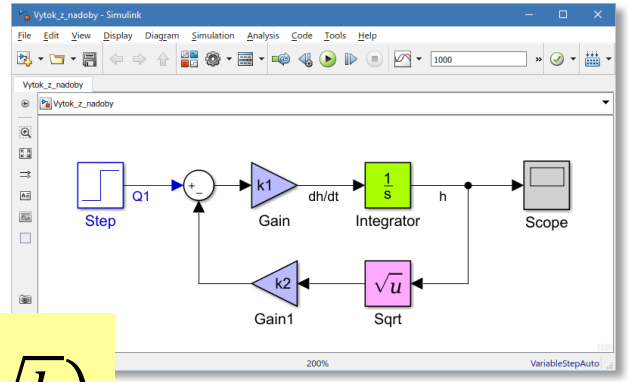




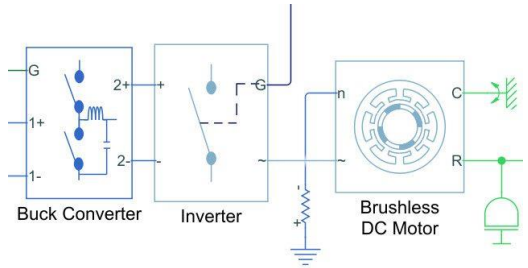
Modelování soustav

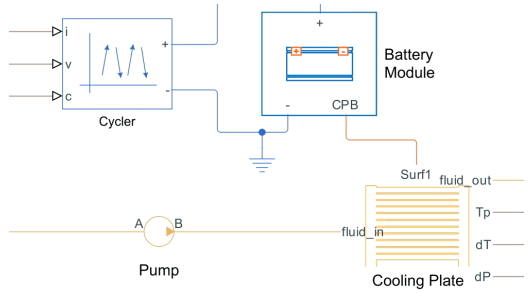
- Popis matematickými rovnicemi
  - matematicko-fyzikální analýza
  - identifikace soustavy z naměřených dat
- Fyzikální modelování
  - elektromechanika
  - baterie
  - tekutinové systémy
  - 3-D mechanika
  - převodové systémy
- Aplikačně zaměřené
  - automobily, letadla, drony, roboty

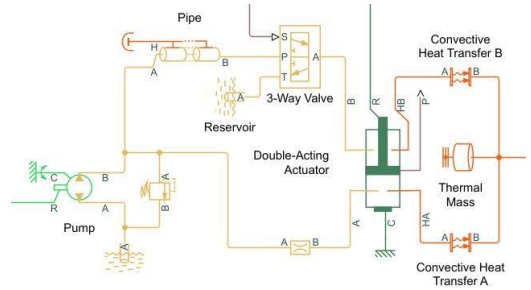


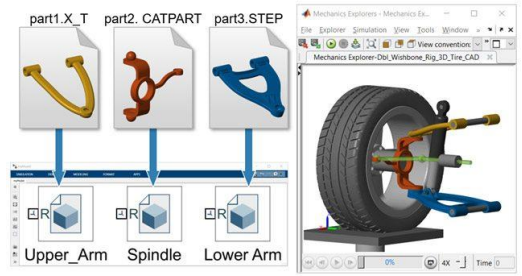


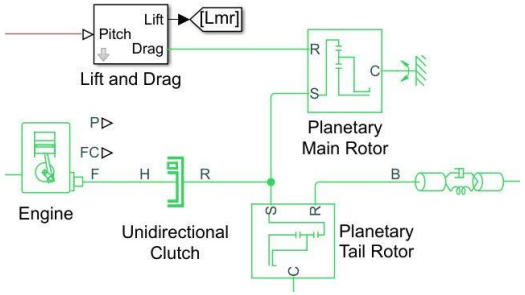
$$\frac{dh}{dt} = k_1\left(Q_1 - k_2\sqrt{h}\right)$$









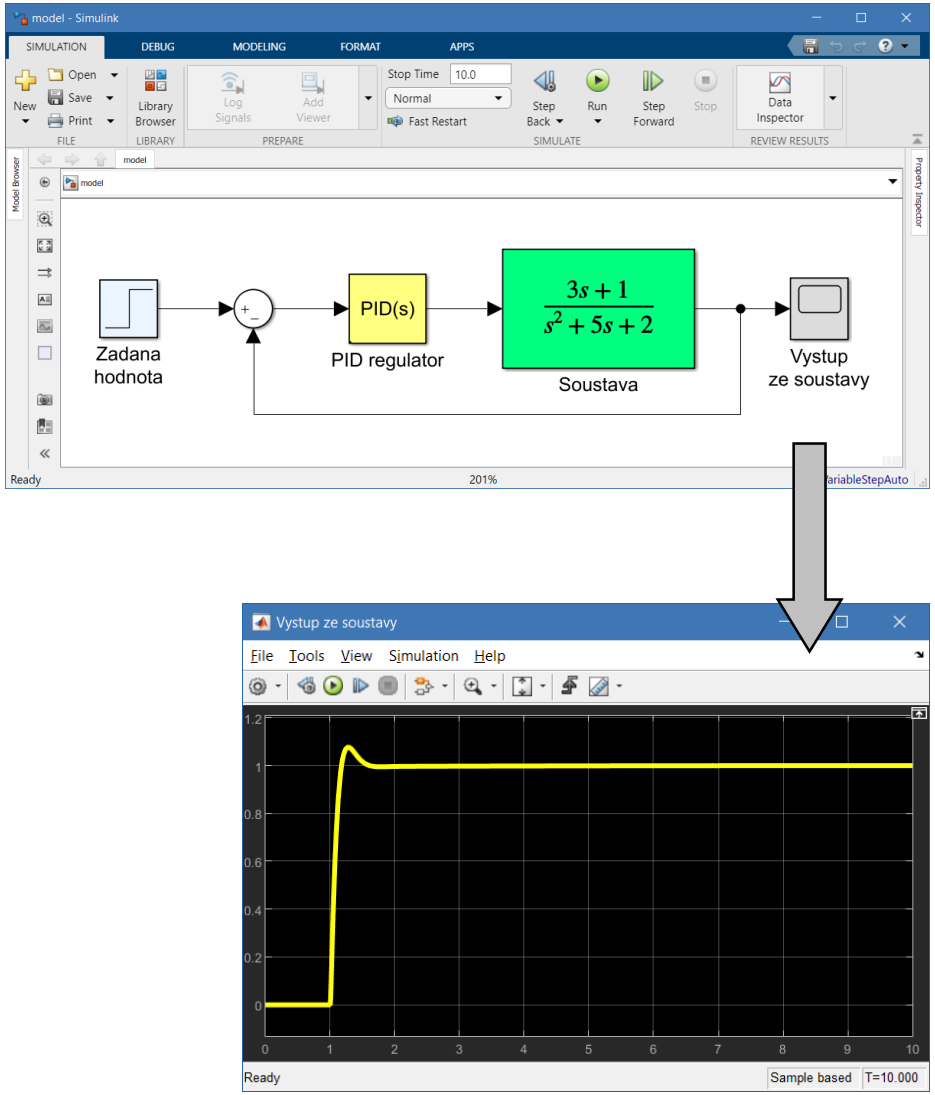


Modelování algoritmů

- 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

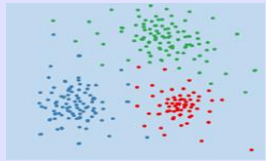
UMĚLÁ INTELIGENCE (AI)

Libovolná technika, která umožňuje strojům napodobit lidskou inteligenci

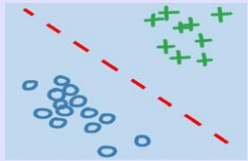
MACHINE LEARNING

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

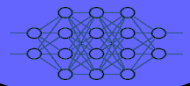
UNSUPERVISED LEARNING
(neoznačená data)



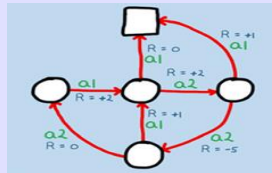
SUPERVISED LEARNING
(označená data)



DEEP LEARNING
(Neuronové sítě s mnoha vrstvami)

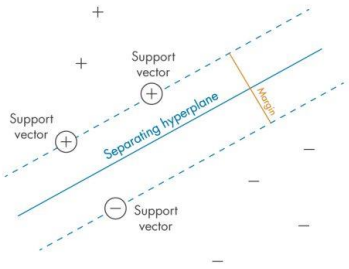


REINFORCEMENT LEARNING
(data z interakce)



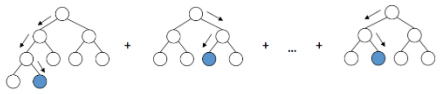
AI modely v prostředí MATLAB

Machine learning



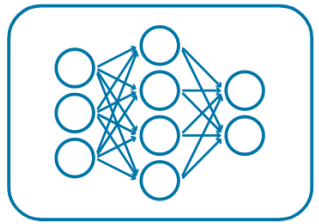
SVM

Clustering

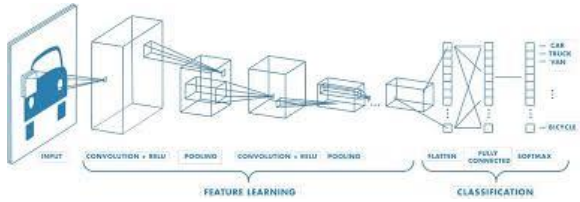


Decision trees

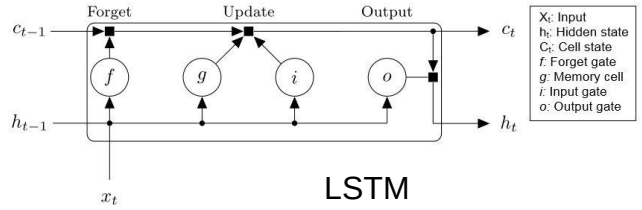
Deep learning



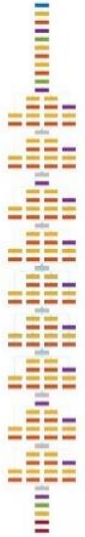
FC



CNN



LSTM

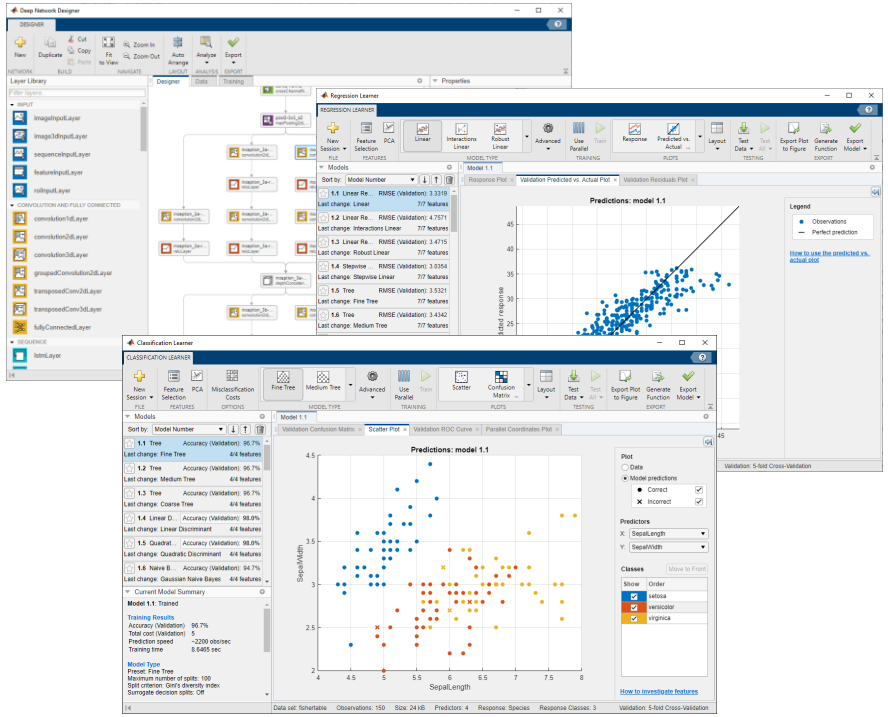


3 cesty k vytvoření AI modelu v prostředí MATLAB

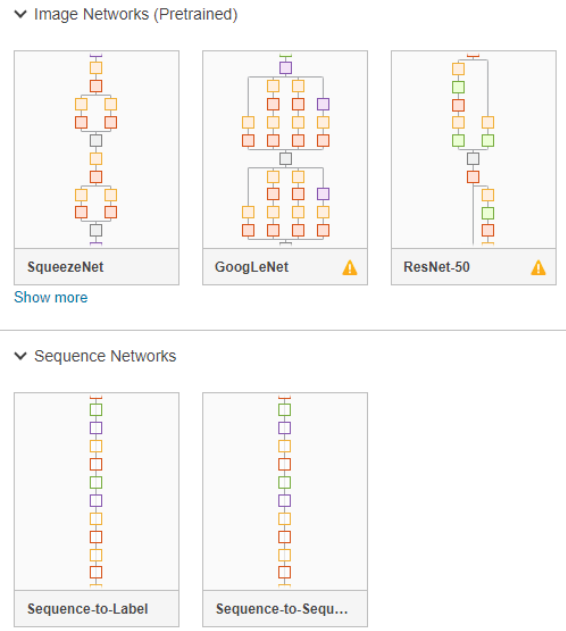
```
inputSize = 12;
numHiddenUnits = 100;
numClasses = 9;

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

`fitcauto / fitrauto`

Funkce, skripty



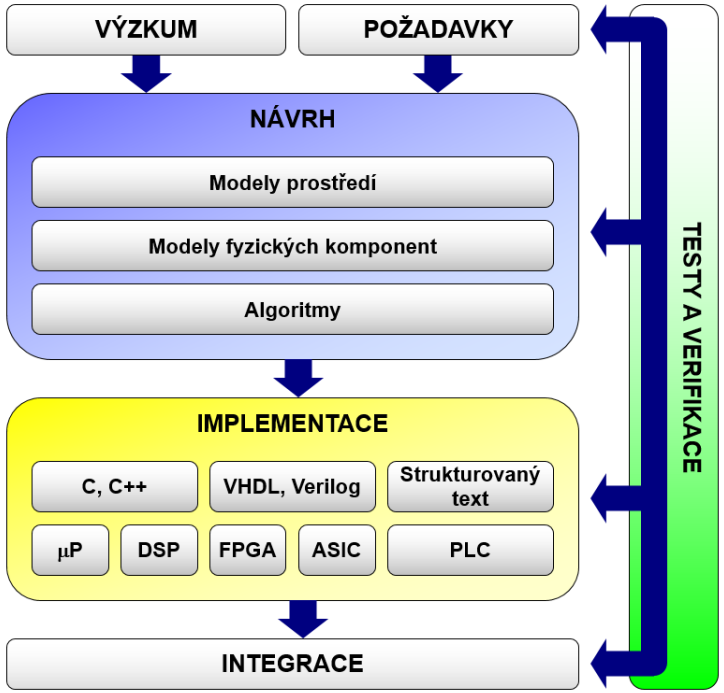
Interaktivní návrh prostřednictvím grafických aplikací



Využít předdefinované sítě a před-učené sítě

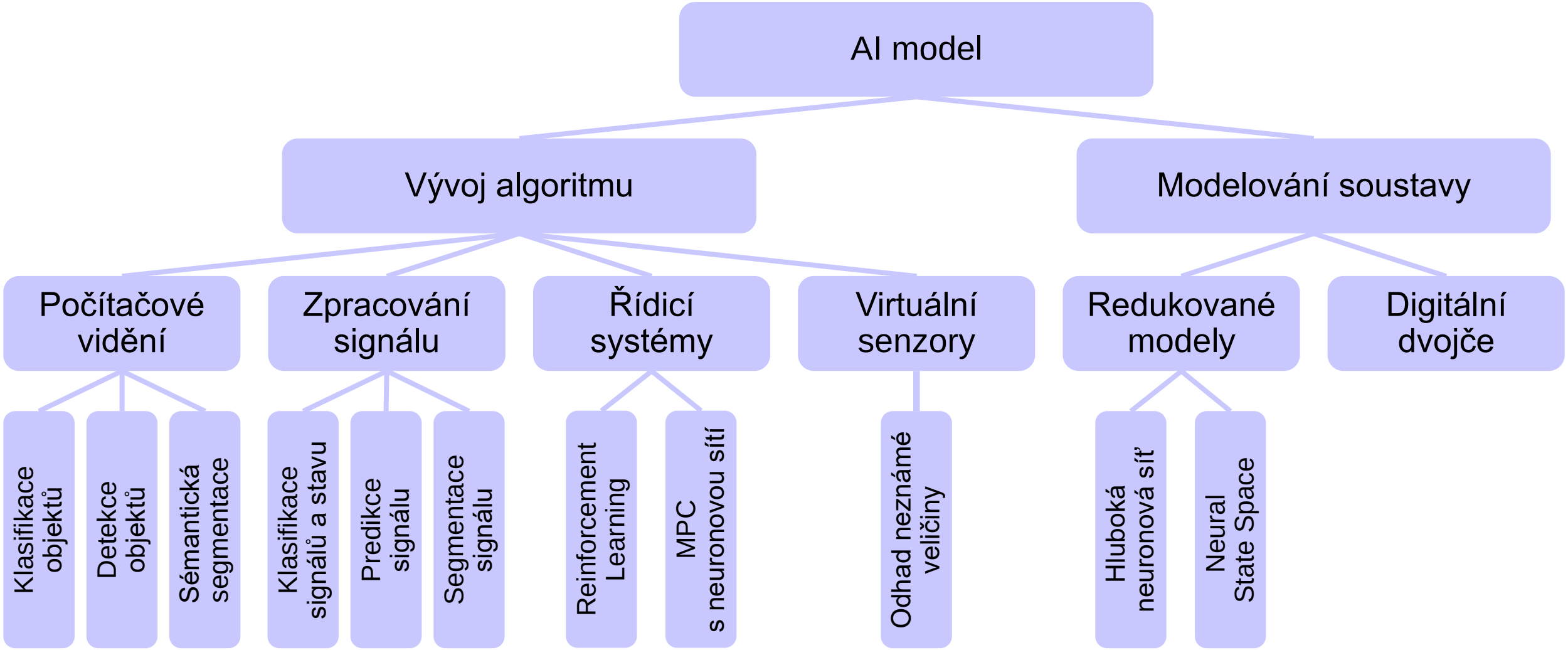
AI a Model-Based Design

+



https://www.mathworks.com/solutions/ai-model-based-design.html

Modelování AI – Typ algoritmu



Postup návrhu systému využívajícího AI

Příprava dat


Čištění a příprava dat


Lidský vhled


Data generovaná simulacemi

Modelování AI


Návrh a ladění AI modelu


Hardwarová akcelerace učení


Interoperabilita

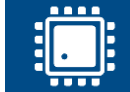
Návrh systému


Integrace AI s dalšími algoritmy


Simulace systému


Verifikace a validace systému

Nasazení


Embedded zařízení


Enterprise systémy

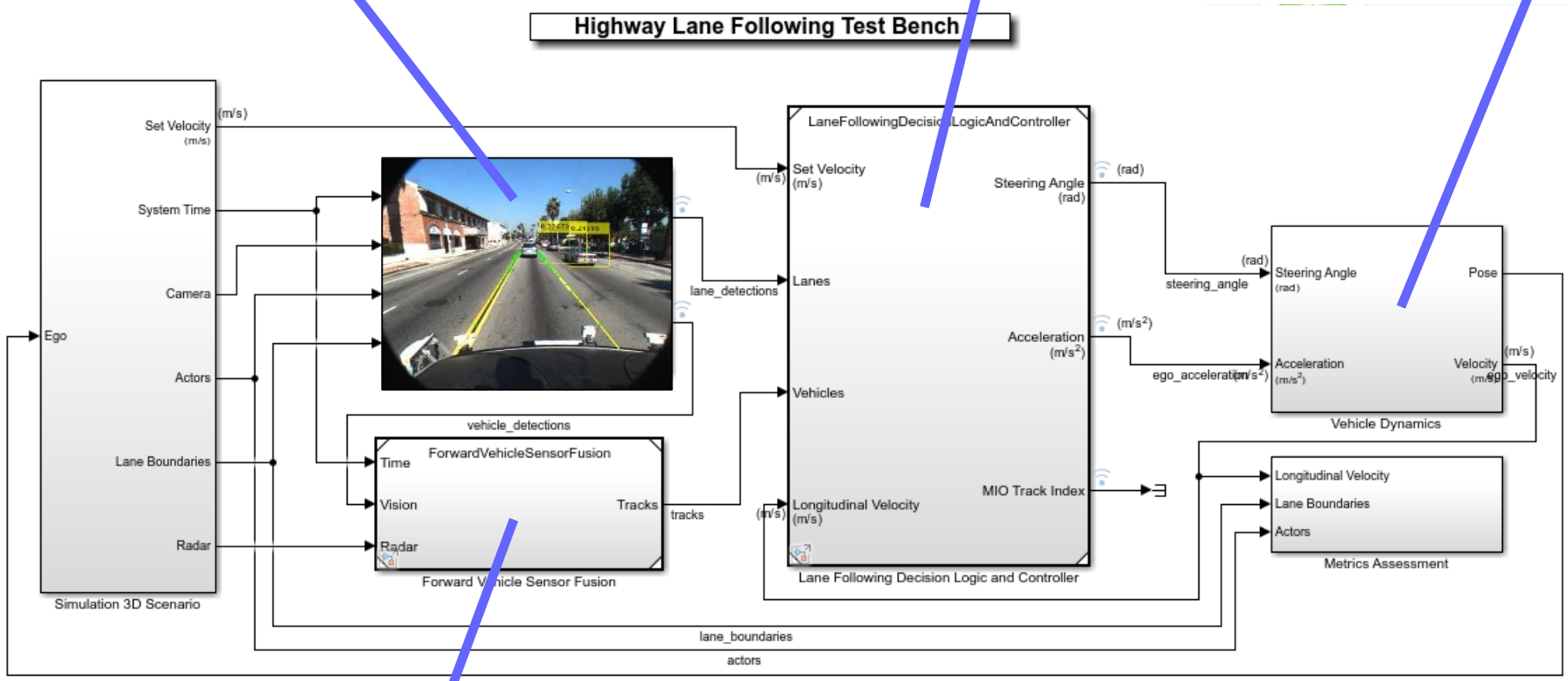

Koncová zařízení, cloud, desktop

Typický příklad: Asistenční systém automobilu

AI algoritmus
detekce jízdních pruhů a vozidel

Řídicí systém

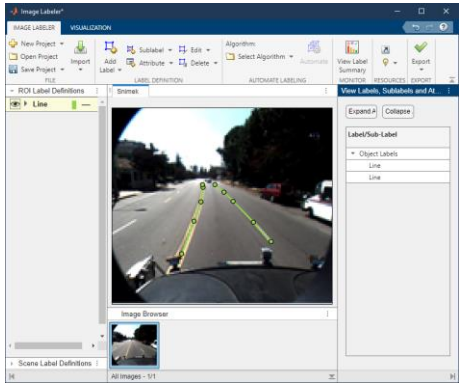
Model dynamiky vozidla
pro testování chování systému



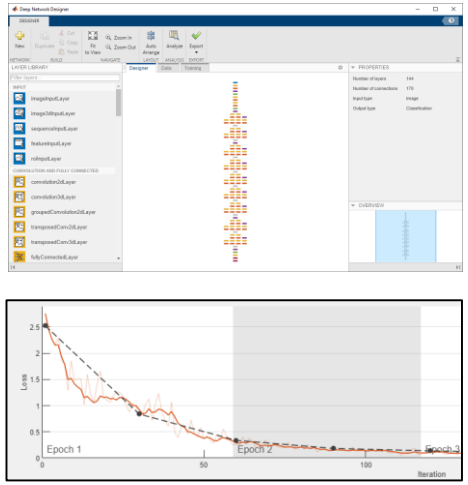
Senzorická fúze

Typický příklad: Asistenční systém automobilu

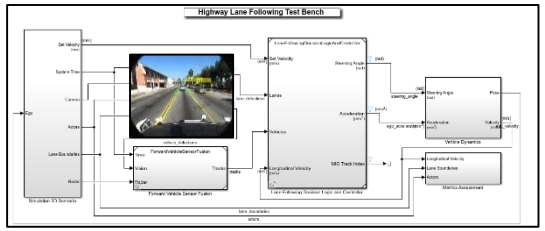
Příprava dat → Modelování AI → Návrh systému → Nasazení



- 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

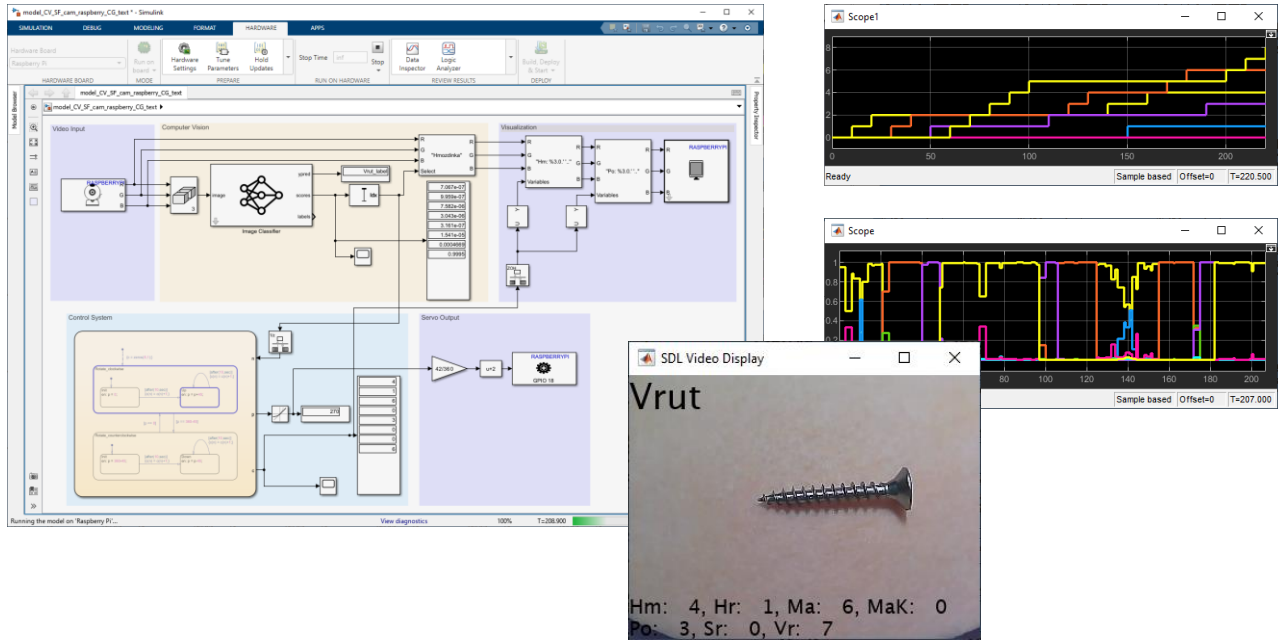


- propojení AI modelů s ostatními algoritmy
- simulace a testování modelu celého software

- generování kódu pro cílovou embedded platformu

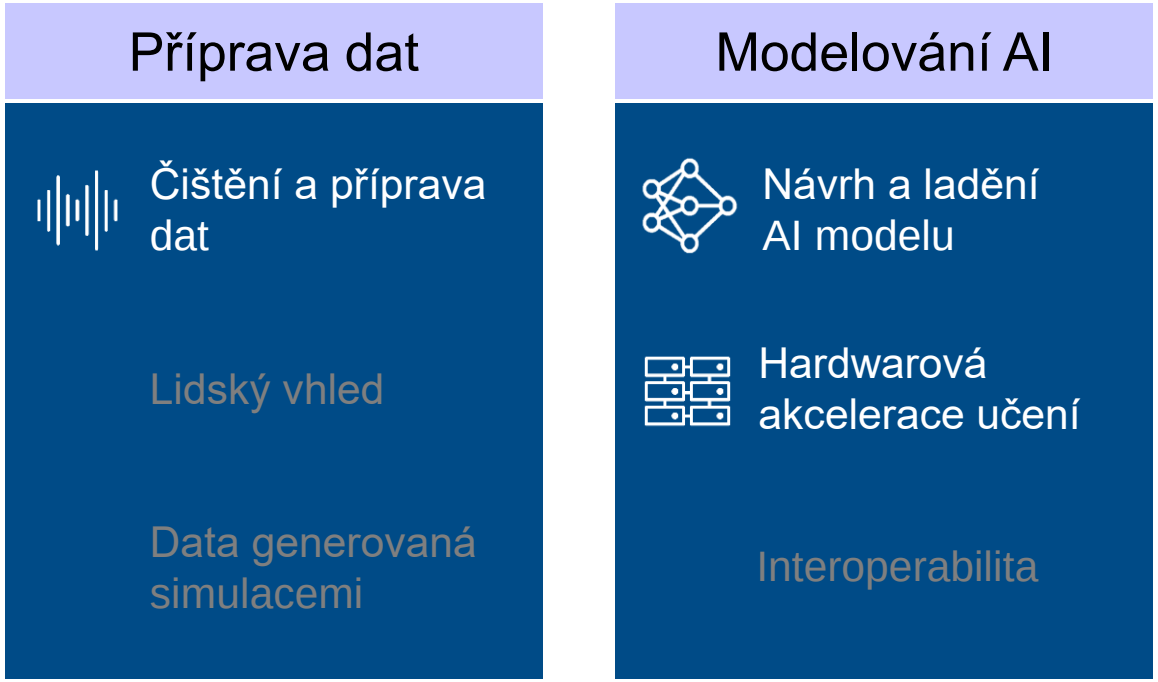
Ukázka: Systém pro klasifikaci a počítaní objektů

- 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

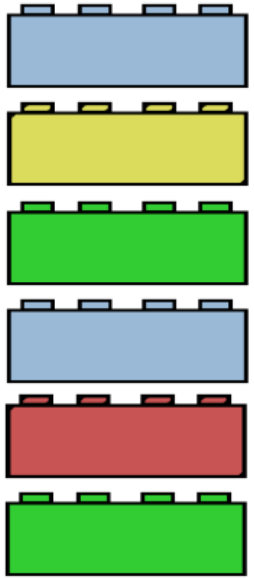


Ukázka: Systém pro klasifikaci a počítaní objektů

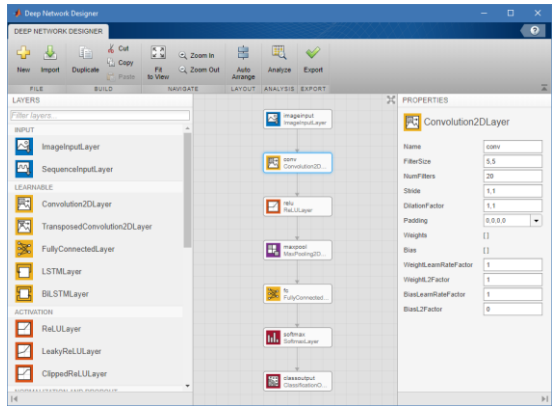
- 1. vytvoření AI modelu pro klasifikaci objektů z obrazových dat



Vytvoření hluboké neuronové sítě v prostředí MATLAB



Deep Network Designer



učení
v rámci
aplikace

```
options = trainingOptions('sgdm');
net = trainNetwork(data,layers,opts);
```

```
results = classify(net,newData);
```

vhodné pro většinu úloh

Připravené funkce

```
layers = [imageInputLayer(inputSize)
          convolution2dLayer(filterSize,numFilters)
          reluLayer()
          maxPooling2dLayer(poolSize)
          fullyConnectedLayer(numClasses)
          softmaxLayer()
          classificationLayer()];
```

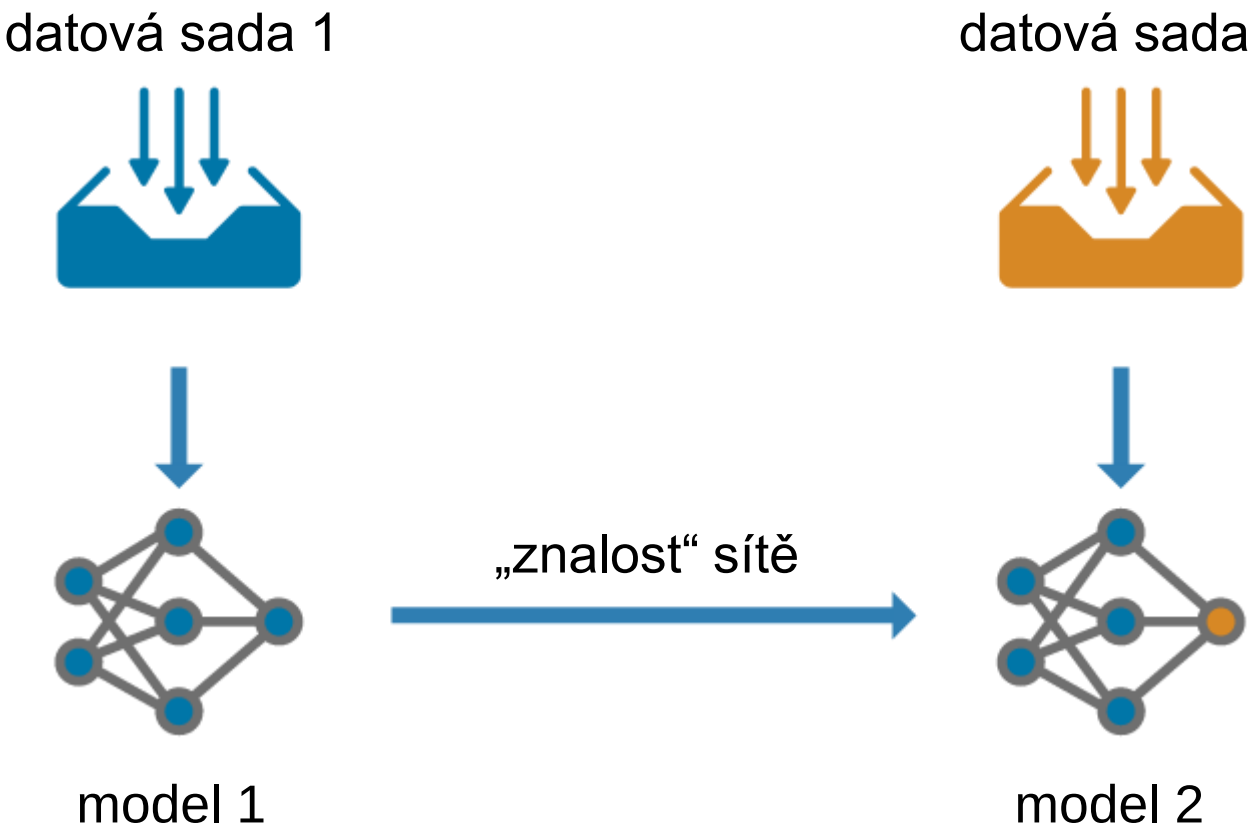
Přizpůsobení na míru

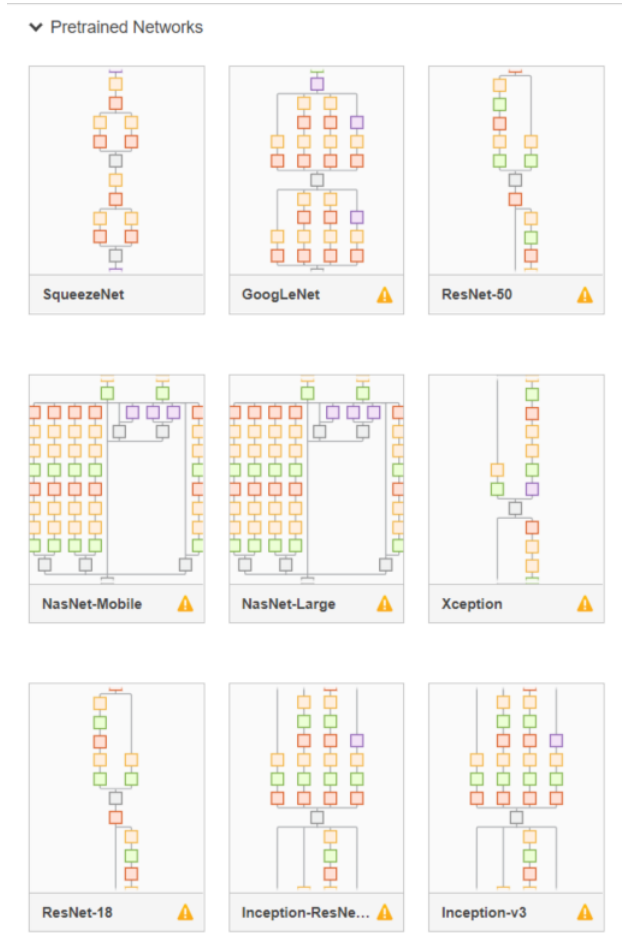
uživatelské smyčky pro učení sítí
automatická diferenciace
sdílené váhy
uživatelské ztrátové funkce
...

GAN, CGAN, siamské sítě, ...

Transfer learning

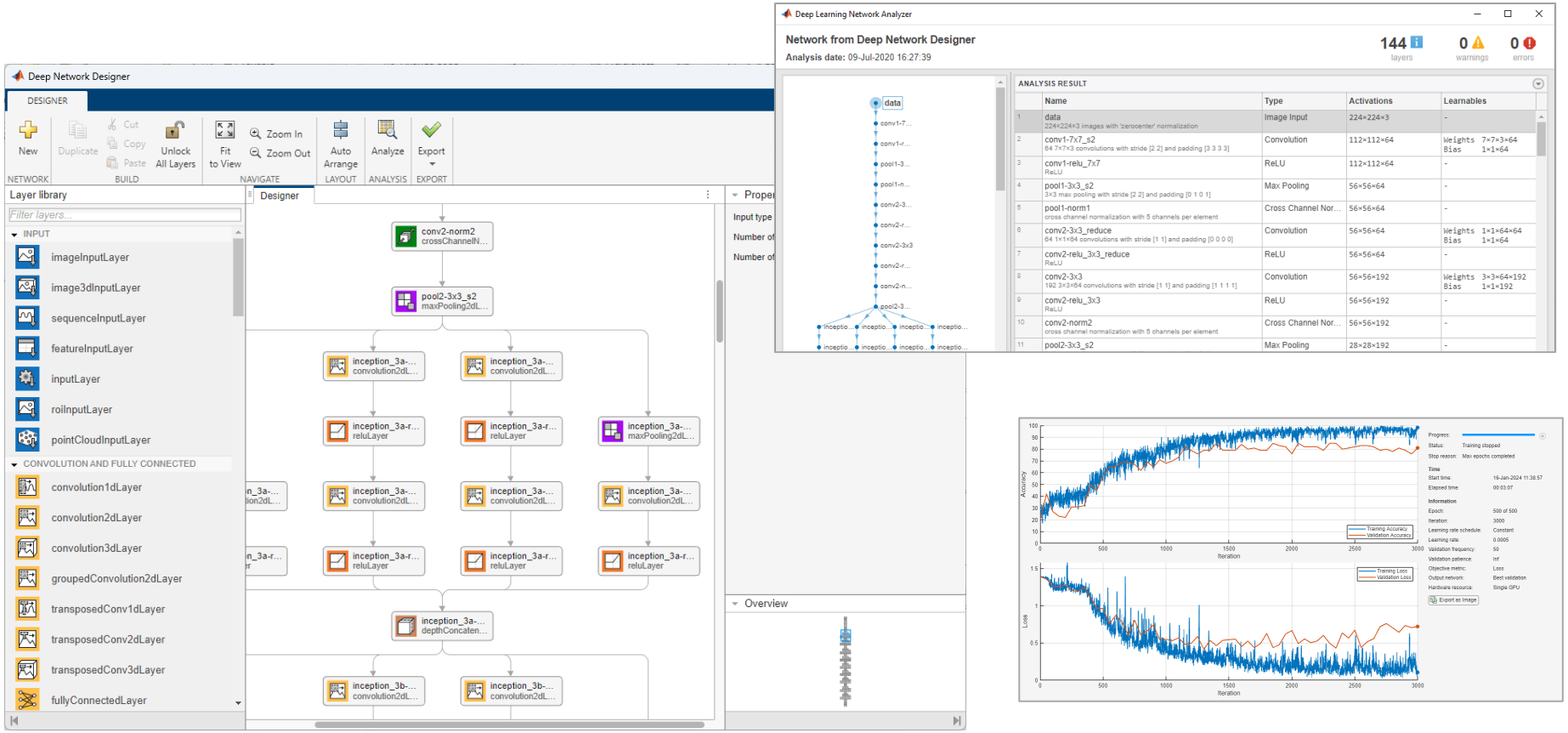
- Využití předdefinovaných a před-učených sítí





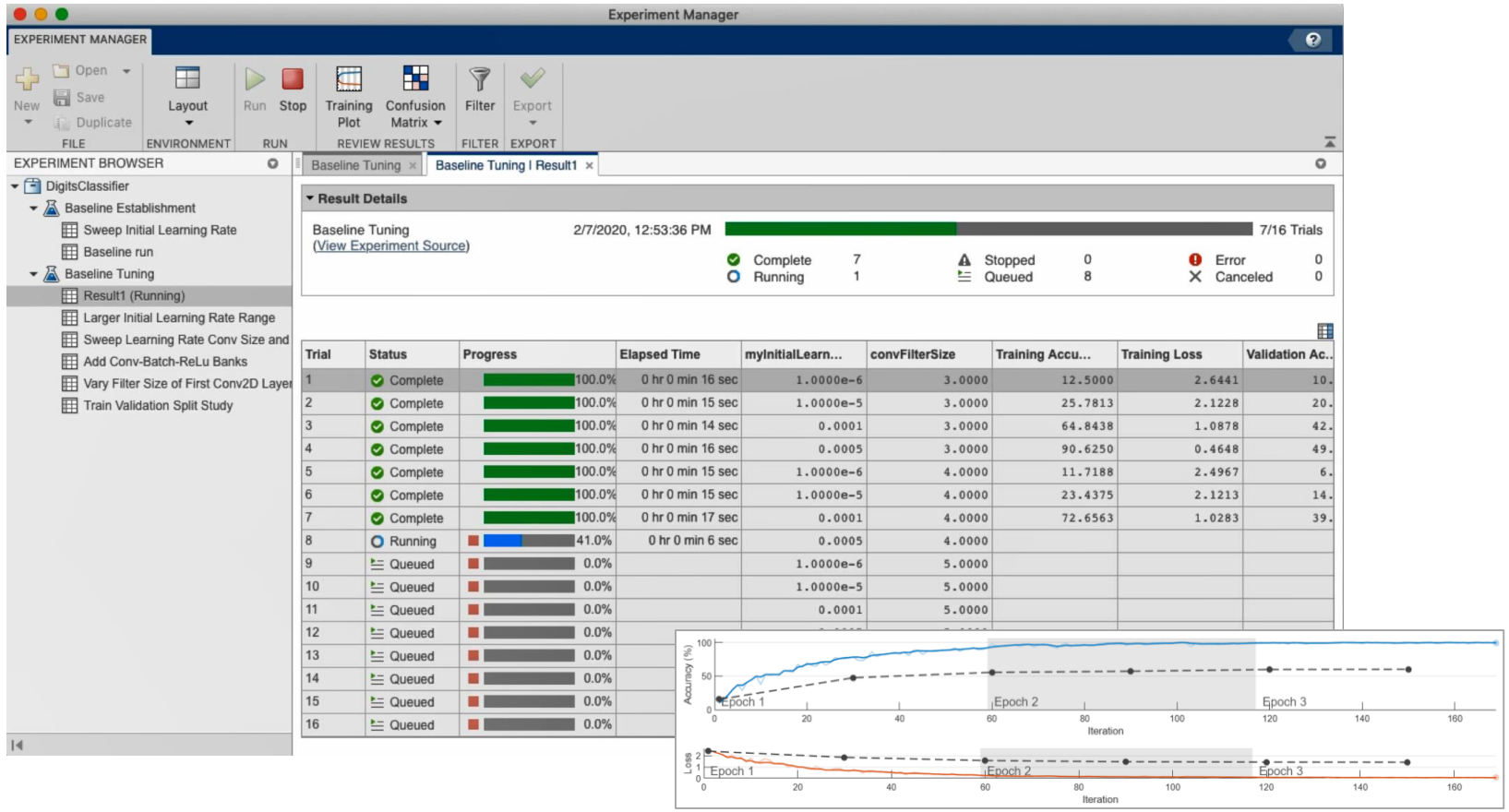
Deep Network Designer

- Grafická tvorba a úprava sítí pro rychlejší návrh



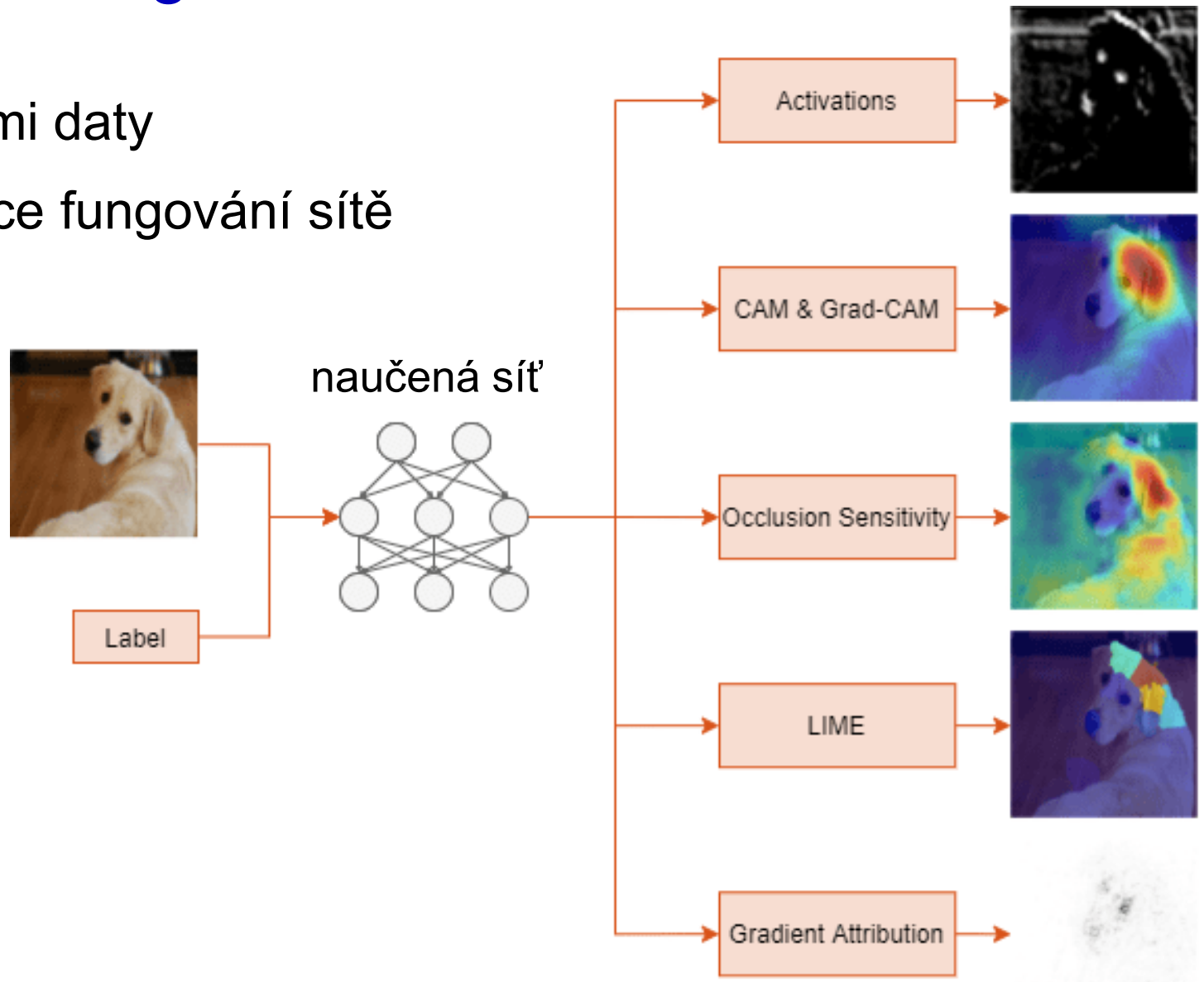
Experiment Manager

- Nalezení optimální sítě pomocí experimentů



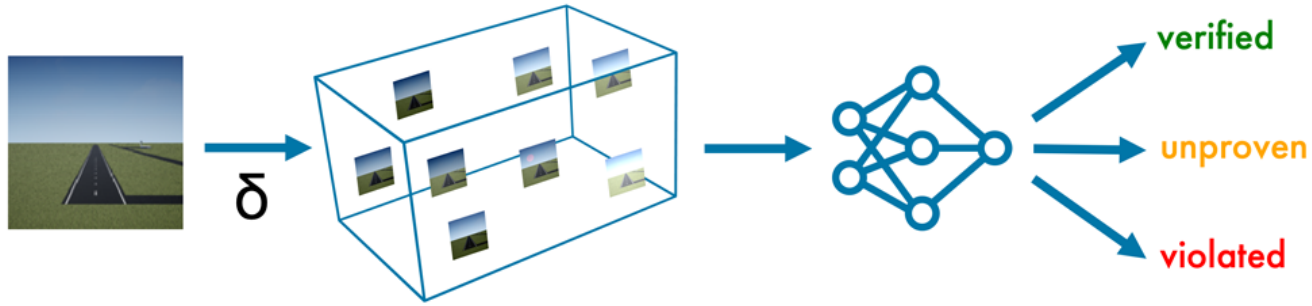
Ověření správného fungování sítě

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



Verifikace AI modelů

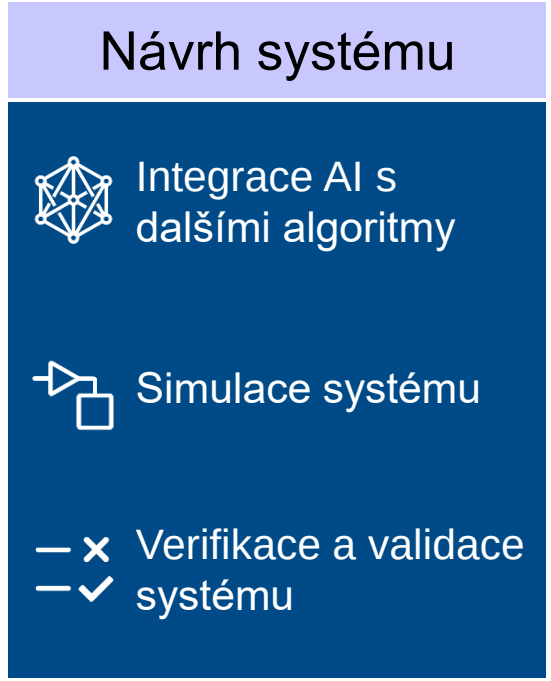
- Deep Learning Toolbox Verification Library
  - cíl: zajistit robustnost a spolehlivost hlubokých neuronových sítí
- Formální metody pro ověření robustnosti
  - proti tzv. adversarial examples



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

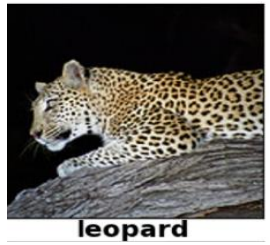
Ukázka: Systém pro klasifikaci a počítaní objektů

- 2. začlenění AI modelu do algoritmu pro ovládání stolu a počítání objektů

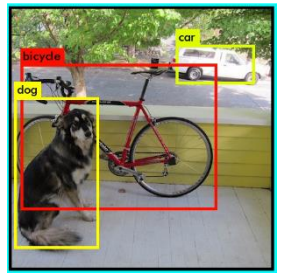


Jak zahrnout deep learning do Simulinku

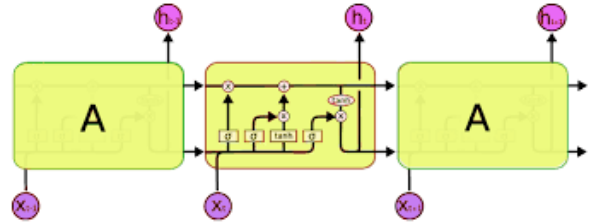
- Připravené bloky pro deep learning



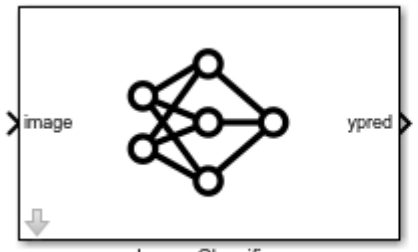
Klasifikace obrazu, sémantická segmentace

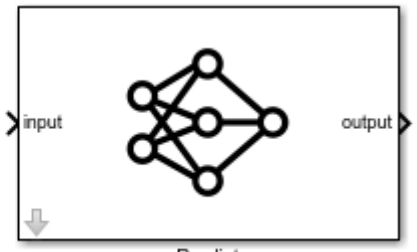


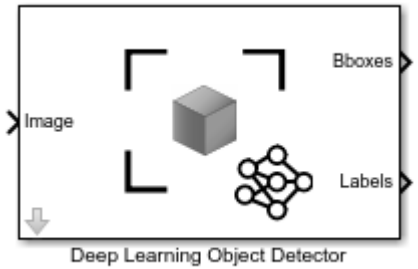
Detektory objektů

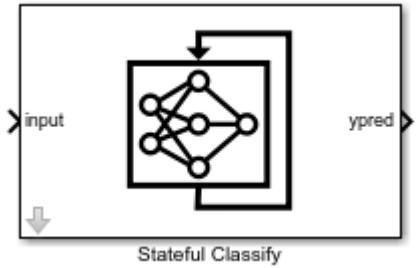


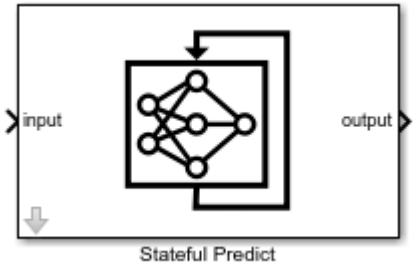
Sekvenční sítě pro zvuková, textová a signálová data





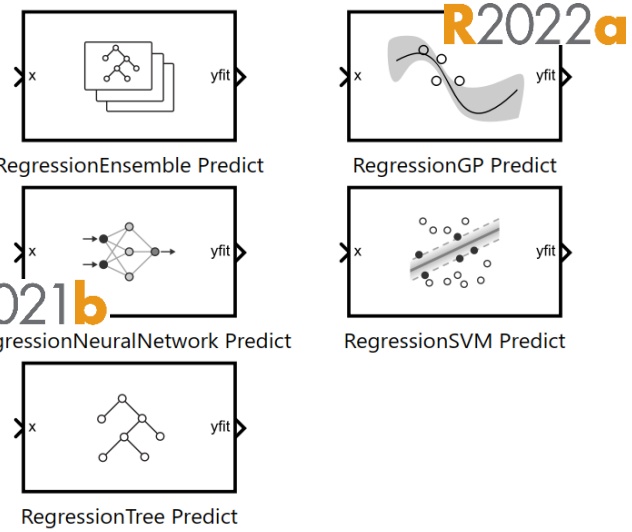


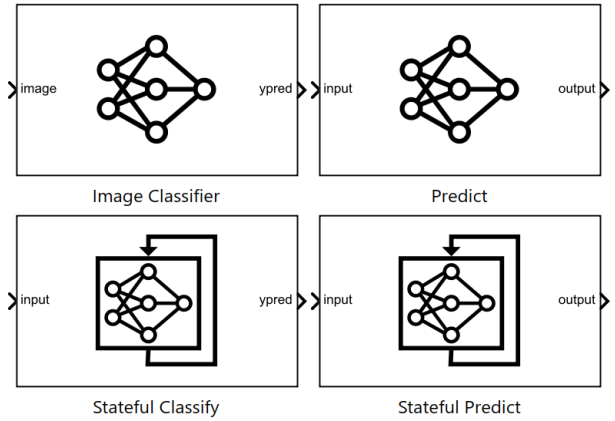




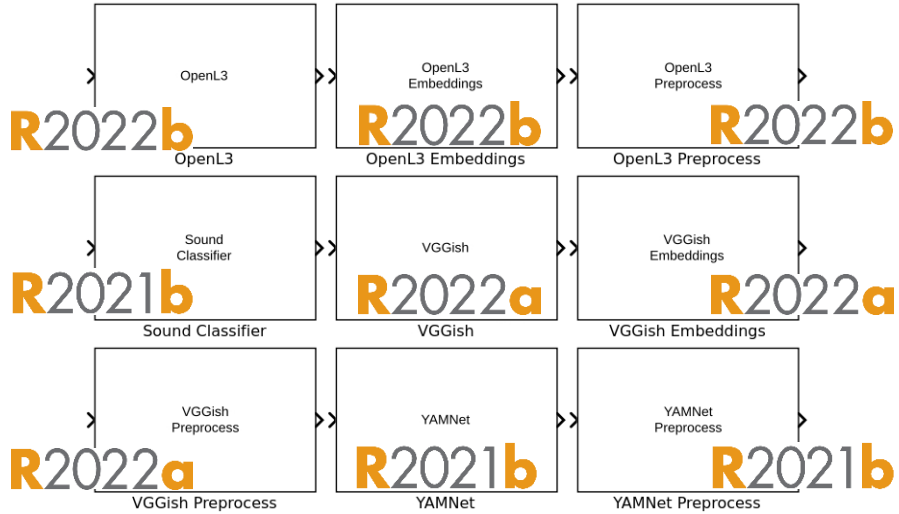
- Blok MATLAB Function

Bloky pro AI v Simulinku stále přibývají

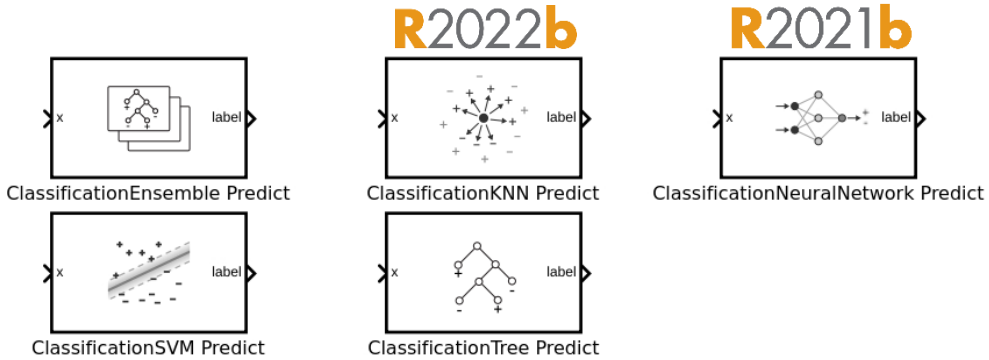




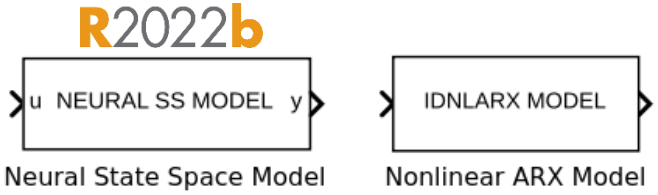
Deep Learning Toolbox



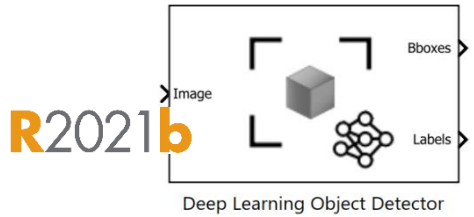
Audio Toolbox



Statistics and Machine Learning Toolbox



System Identification Toolbox

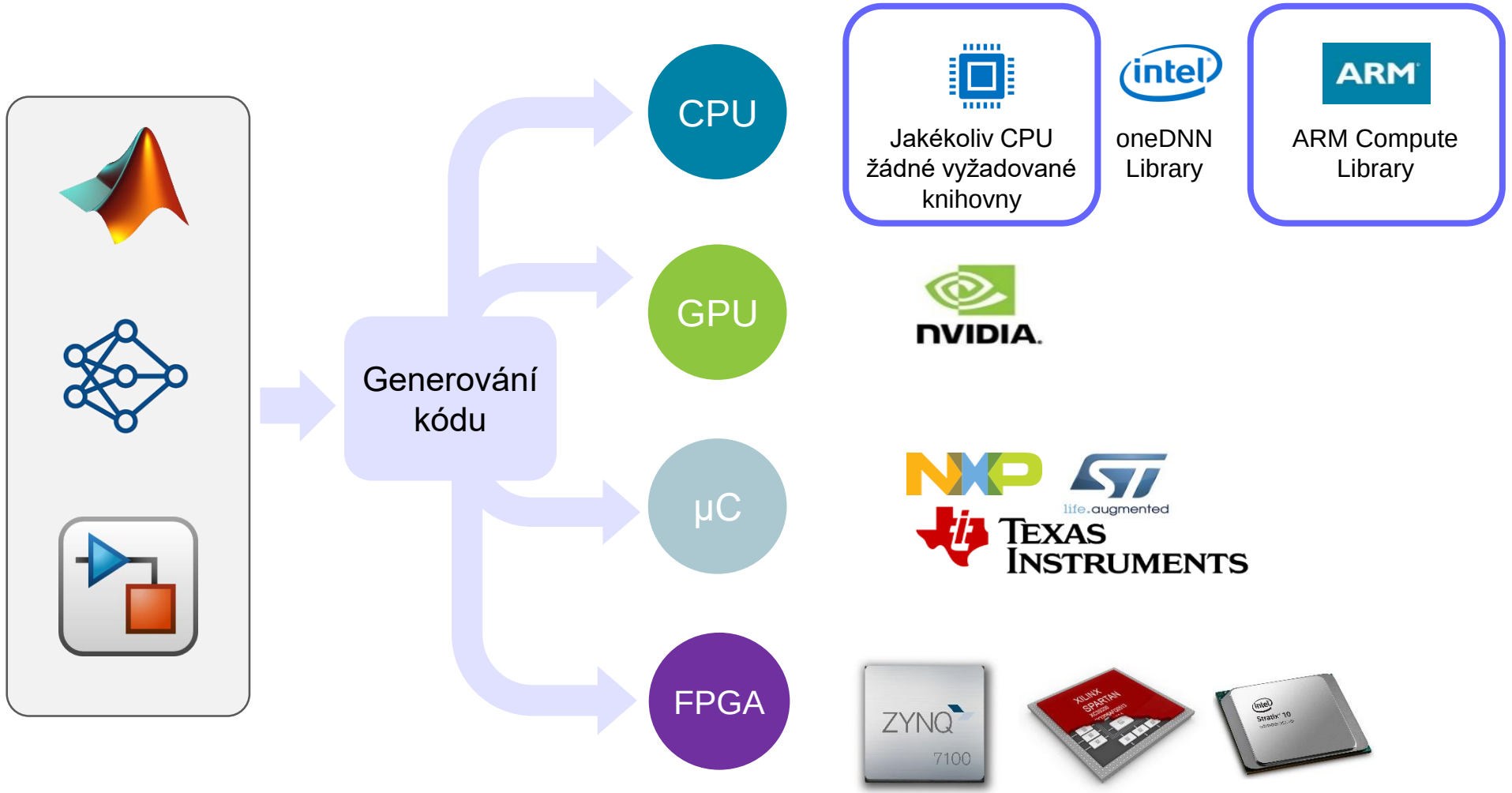


Computer Vision Toolbox

Ukázka: Systém pro klasifikaci a počítaní objektů

- 3. nasazení algoritmu na cílovou platformu: automatické generování kódu

Nasazení na cílovou platformu

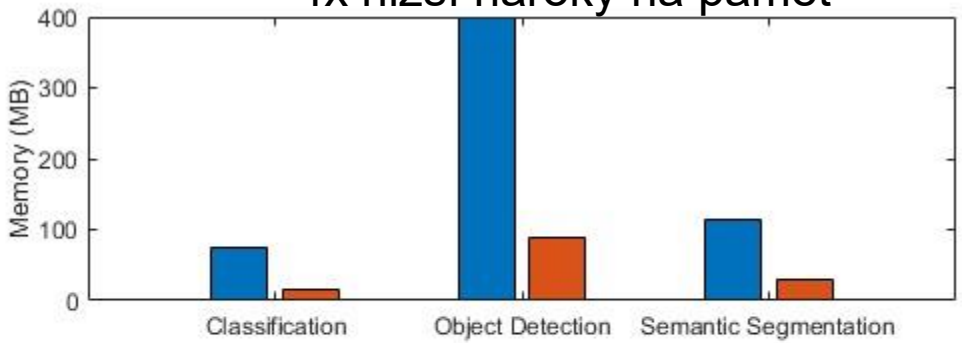


Redukce modelu

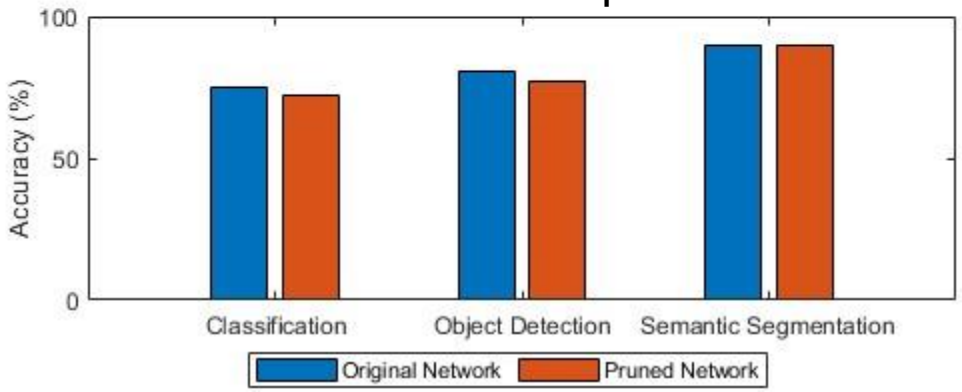
- Snížení paměťových a výpočetních nároků nasazeného modelu
- **Quantizatice**
  - převod z floating point do fixed point aritmetiky
- **Pruning**
  - odstranění nedůležitých částí sítě

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

Pruning
~4x nižší nároky na paměť

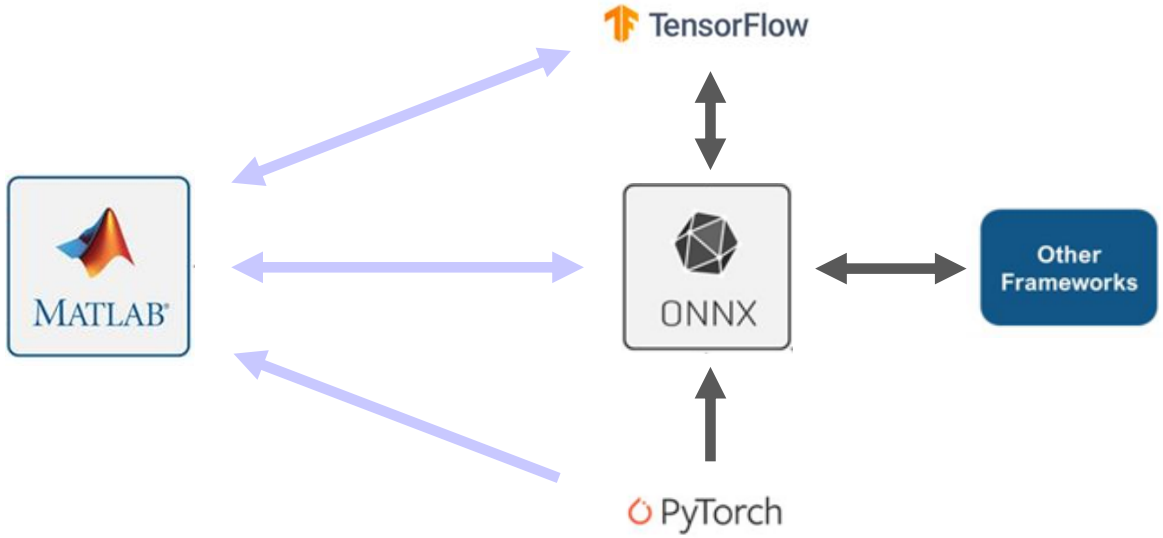


< 5% snížení přesnosti



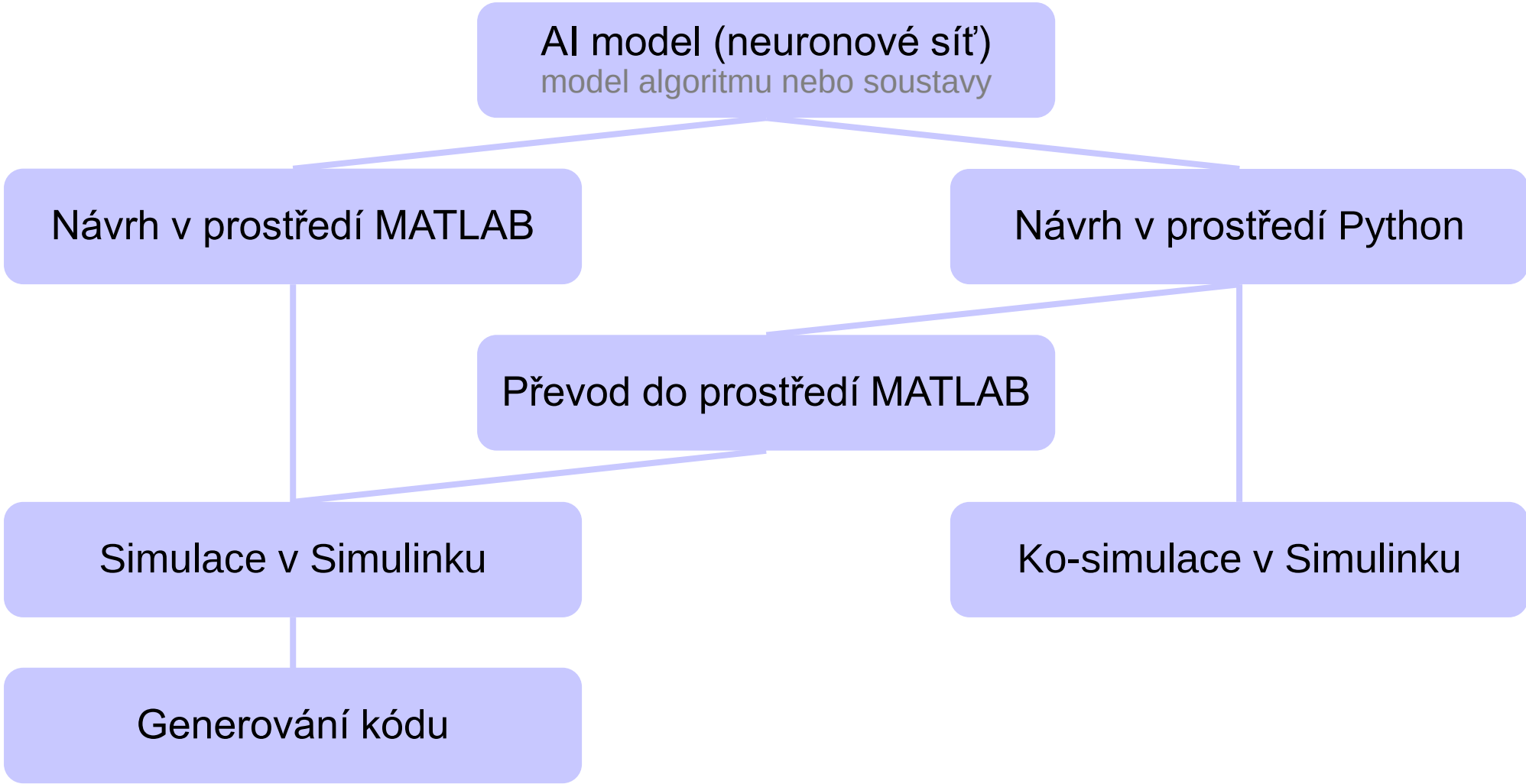
MATLAB a spolupráce s dalšími nástroji

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



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

Modelování AI – Postup tvorby modelu



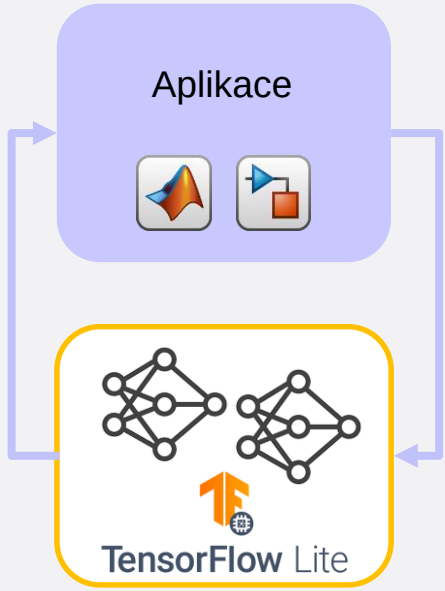
Podpora pro TensorFlow Lite

- Co je TFLite
  - open source nástroj pro deep learning
  - spouštění sítí na koncovém zařízení
- Simulace a nasazení
  - využití předučených modelů z TFLite

TFLite Support Package	R2022a
Support for Windows	R2022b
Support for TFLite 2.8.0	R2023a

Simulace

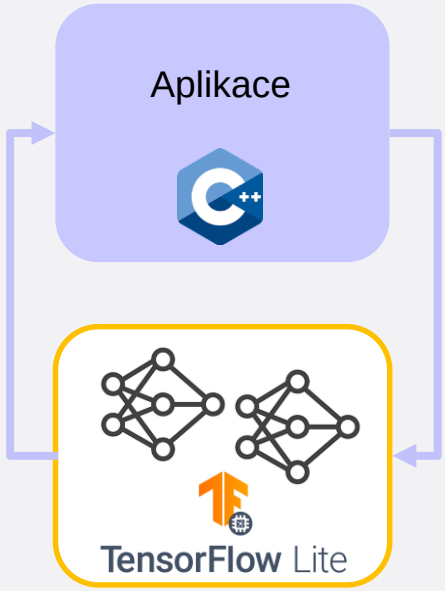
Aplikace spuštěná v MATLAB/Simulink



DL síť spuštěná v TFLite

Generování kódu

Aplikace ve formě generovaného C++ kódu



DL síť spuštěná v TFLite

Podporována podmnožina předučených modelů z TensorFlow Hub

Otázky