



Oprogramowanie
Naukowo-Techniczne
sp. z o.o.

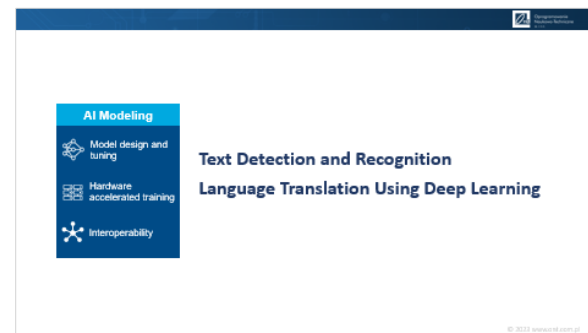
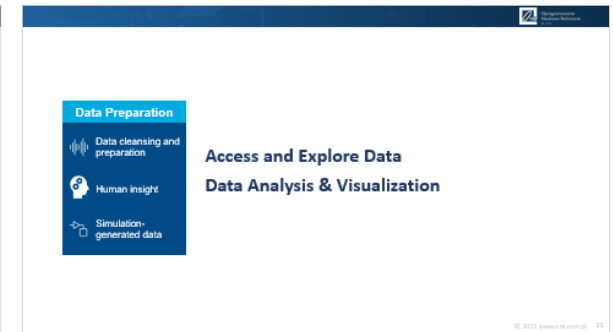
Kraków 19.10.2023 r.

Image analysis and artificial intelligence with MATLAB

Jaromir Przybyło

Agenda

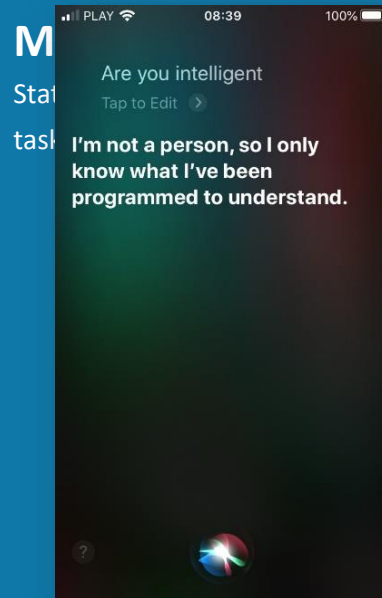
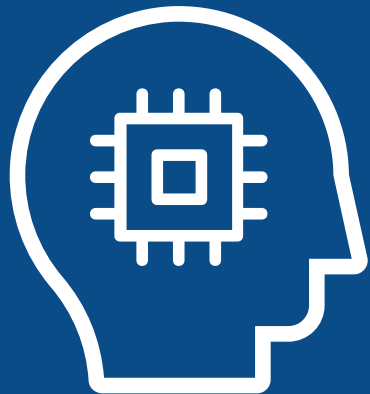
- Introduction
- Data preparation
 - Access and Explore Data
 - Data Analysis & Visualization
- AI Modeling
 - Text Detection and Recognition
 - Language Translation Using Deep Learning
- Application Deployment
 - Integration with complex systems
 - App Building
- Deployment
 - MATLAB WebApp Server



Deep learning is a key technology driving the AI megatrend

ARTIFICIAL INTELLIGENCE

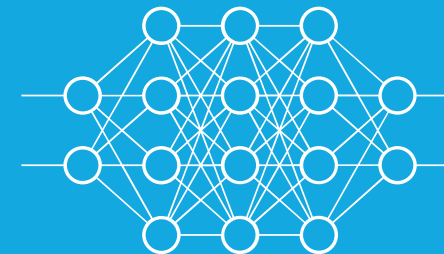
Any technique that enables machines to mimic human intelligence



Machines to “learn”
by programming

DEEP LEARNING

Neural networks with many layers that learn representations and tasks “directly” from data



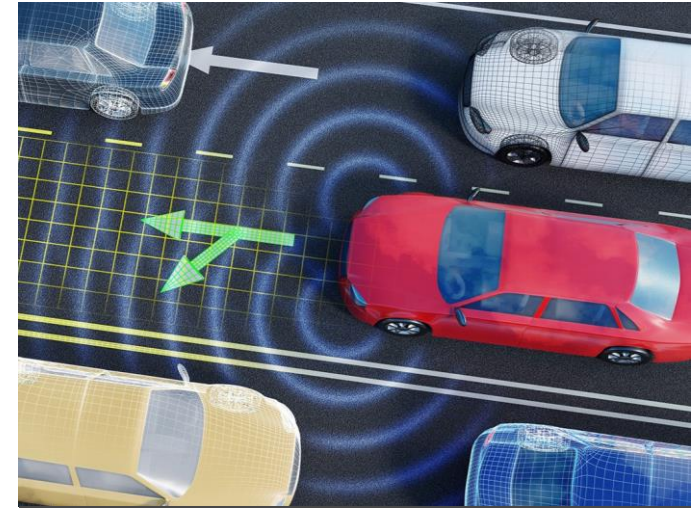
Deep learning is part of our everyday lives



Speech Recognition



Face Detection



Automated Driving

Applications of deep learning for images and video

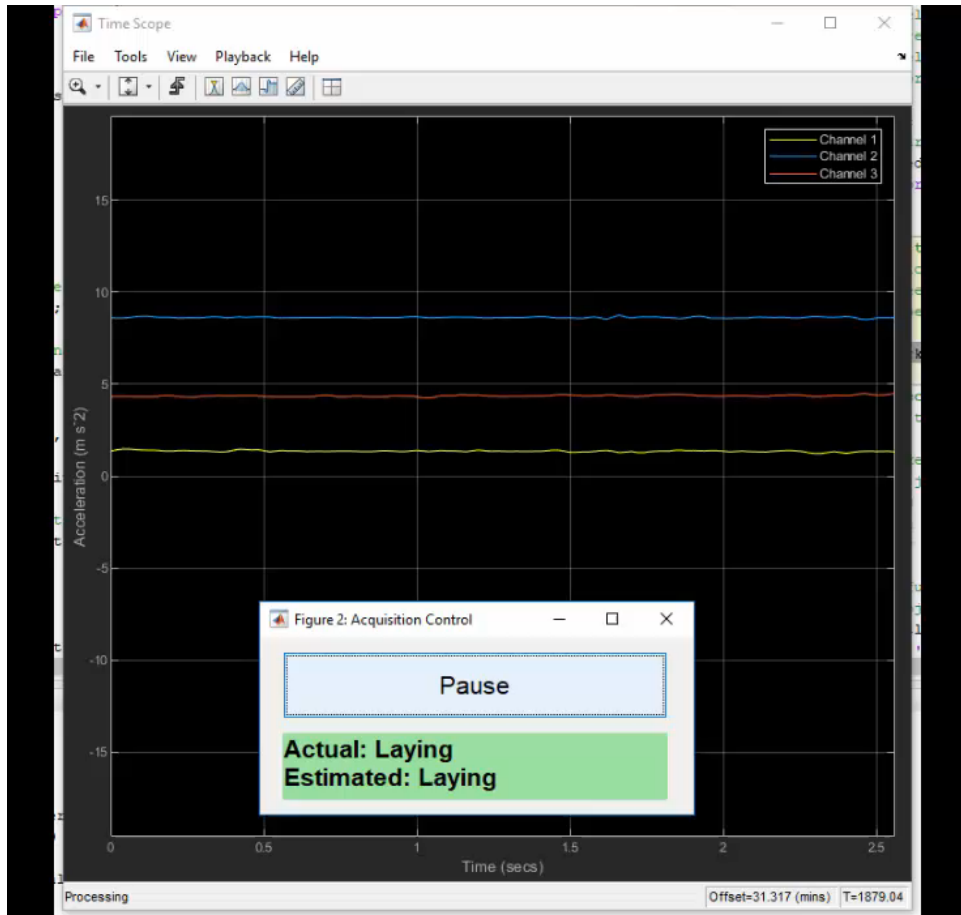


YOLO v2 (You Only Look Once)

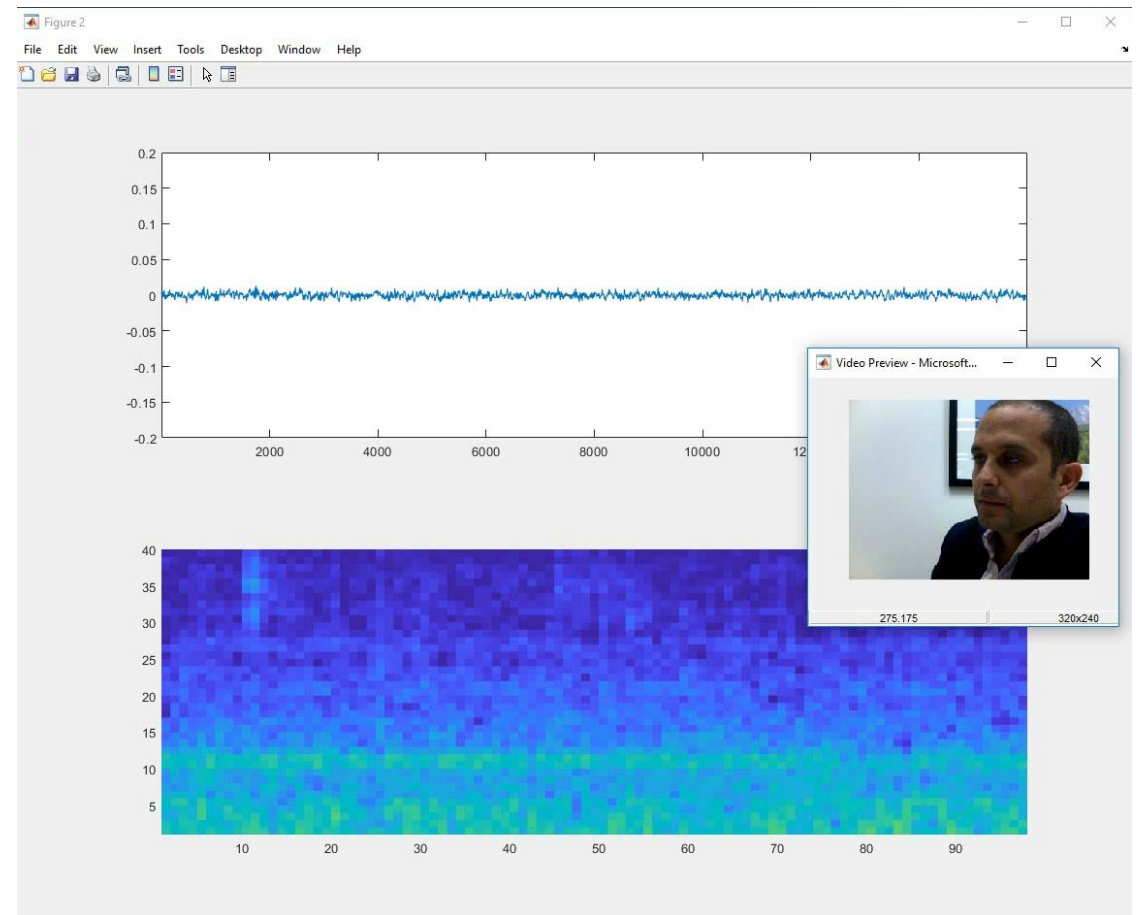


Semantic Segmentation using SegNet

Applications of deep learning for signal processing

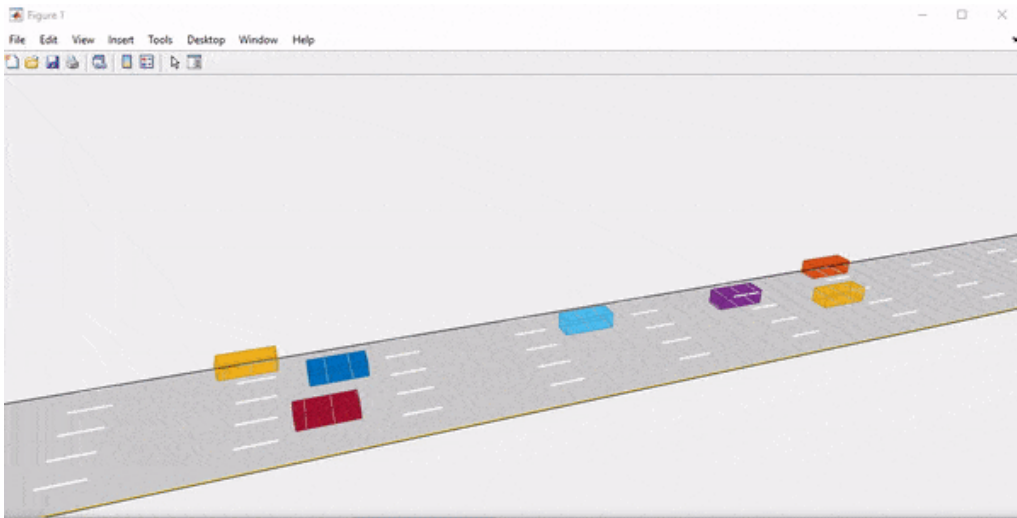


Signal Classification using LSTMs

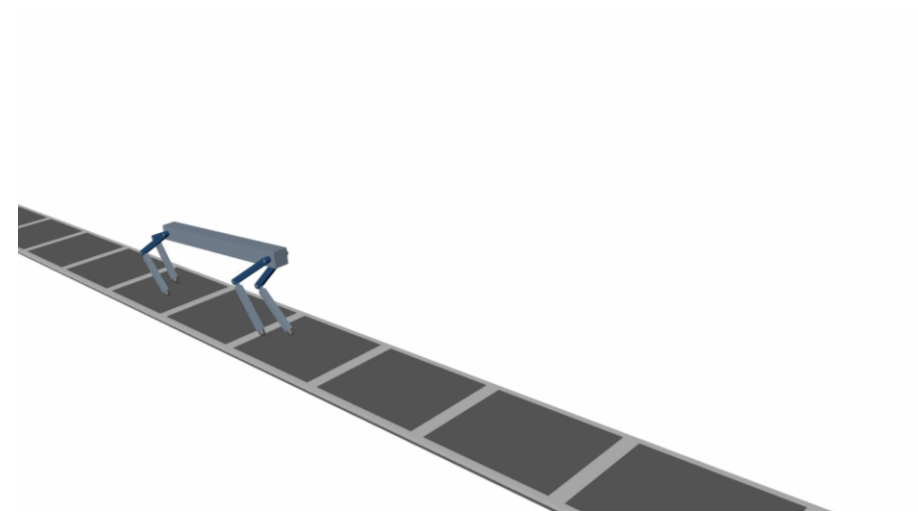


Speech Recognition using CNNs

Applications of reinforcement learning



Teach a car to navigate traffic



Train a Quadruped robot to walk

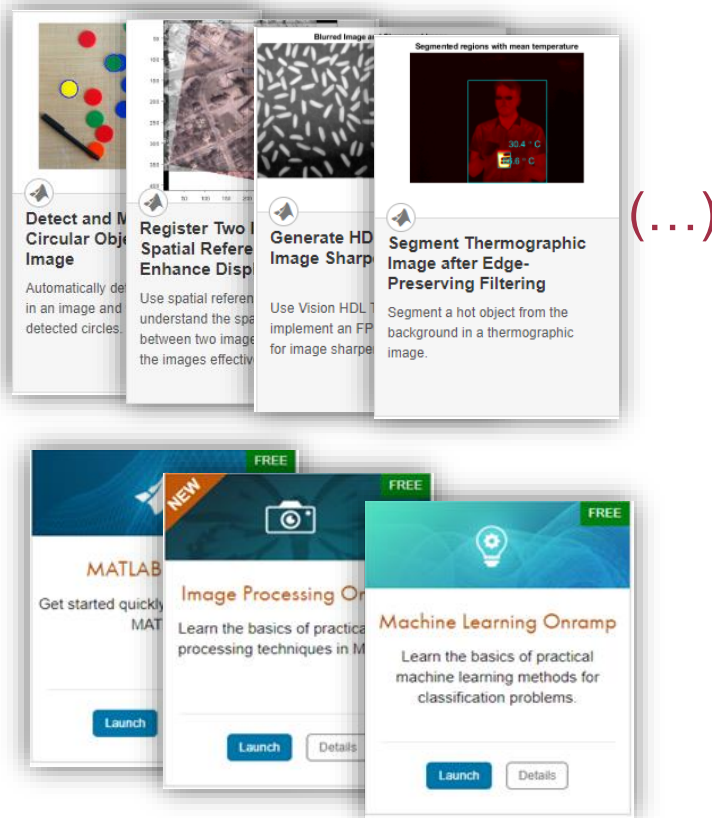
Example: Creating a personal translator app

- **Goal:** Create an application that allows automatic recognition of text from images and its translation
- **Approach:**
 - Image processing and analysis
 - OCR
 - Deep Learning (translation)

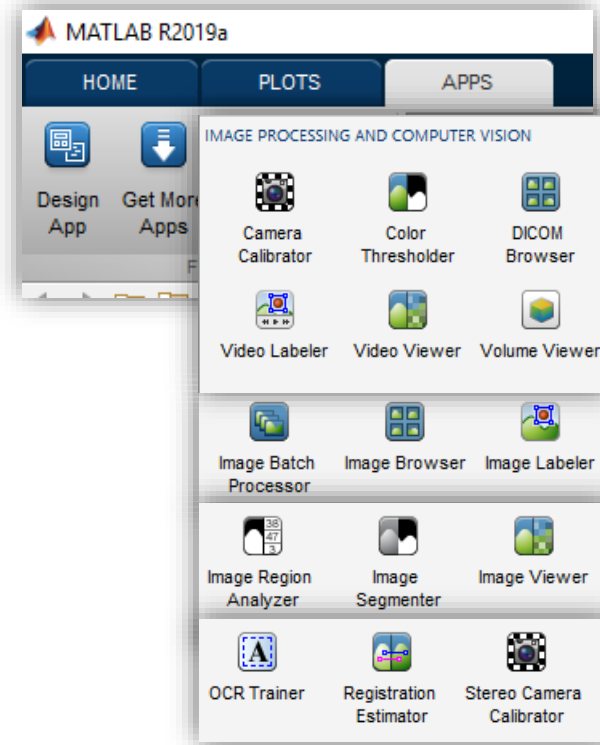


What does MATLAB offer?

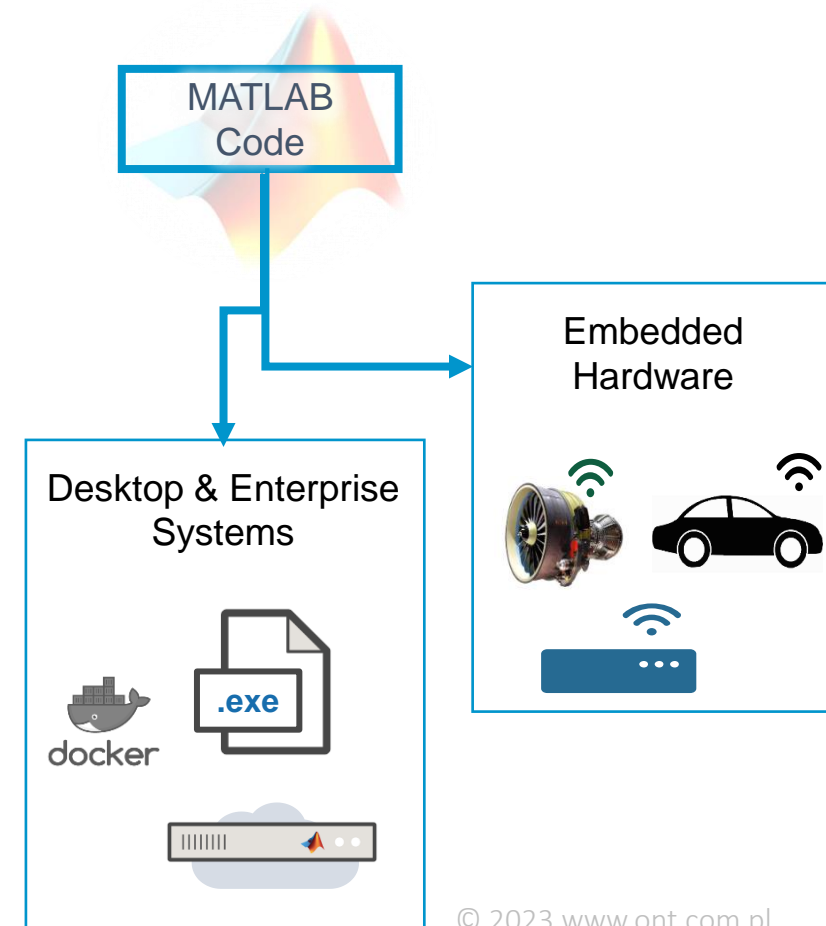
Ease of Use and Thorough Documentation



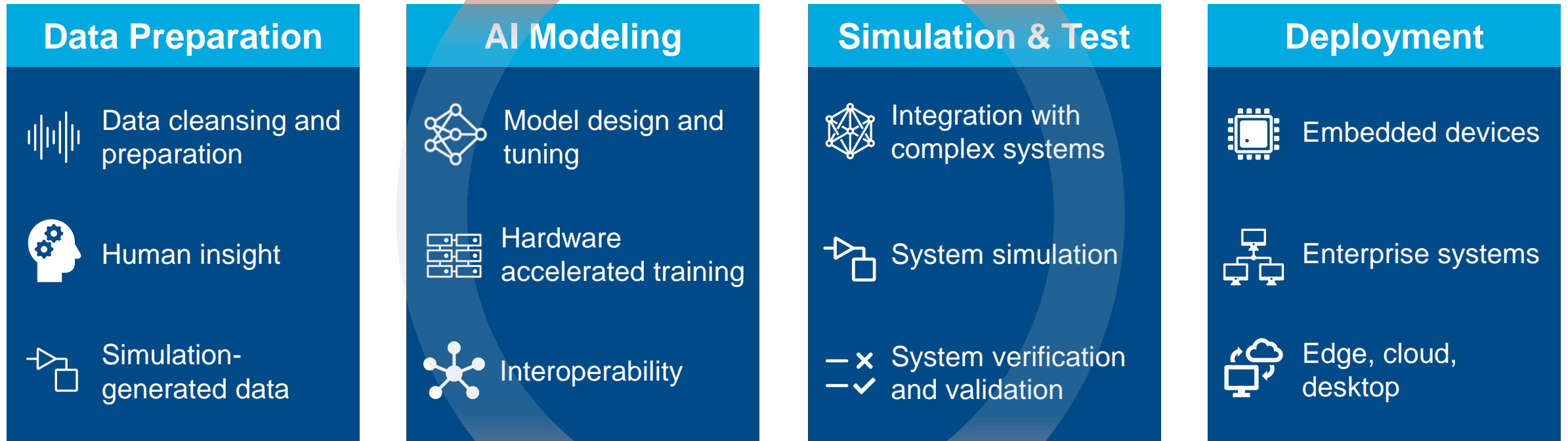
Rapid Prototyping and Algorithm Development



Application Deployment and Code Generation



AI-driven system design



Data Preparation



Data cleansing and
preparation



Human insight



Simulation-
generated data

Access and Explore Data


Data Analysis & Visualization



Access and Explore Data

Access

Files

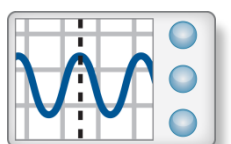


Software

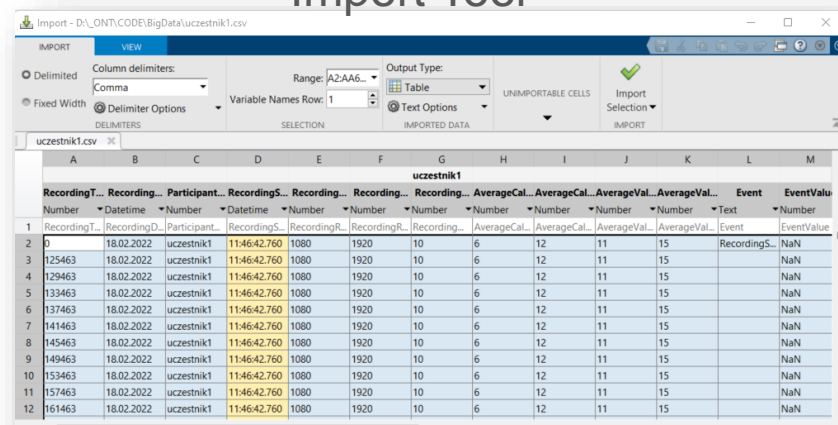


Code & Applications

Hardware

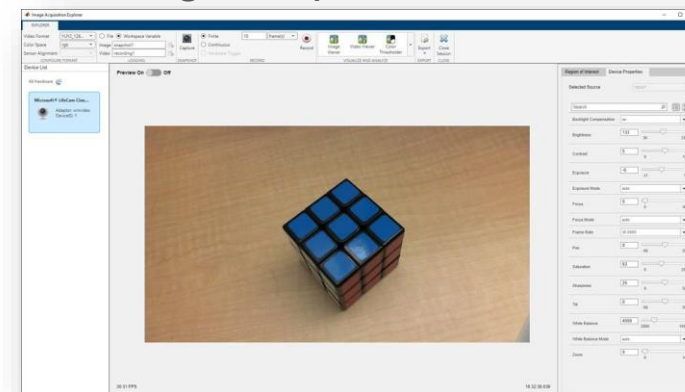


Import Tool



<https://www.mathworks.com/help/matlab/ref/importtool.html>

Image Acquisition Toolbox



<https://www.mathworks.com/products/image-acquisition.html>

Access data in many formats from many locations

- Type
- Structure
- Location





Data Analysis & Visualization

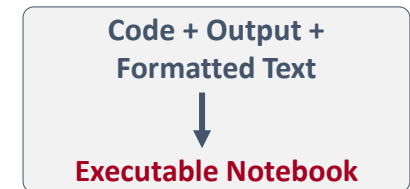
- Create scripts that combine code, output, and formatted text in an executable notebook

The screenshot displays the MATLAB Live Editor window titled "Live Editor - C:\MATLAB\SunriseSunset.mlx". The interface is divided into several sections:

- Header:** Includes tabs for "LIVE EDITOR", "INSERT", and "VIEW", along with a toolbar for file operations (New, Open, Save, Find Files, Compare, Go To, Print), text formatting (Normal, Bold, Italic, Underline, Monospace), code execution (Run Section, Run and Advance, Run and End, Run, Step, Stop), and section management (Section Break).
- Content Area:**
 - Title:** "Estimating Sunrise and Sunset"
 - Images:** Two small images showing a sunset over a horizon.
 - Text:** "We can calculate sunrise and sunset times from the following equations."
 - Equations:**
$$\text{sunrise} = 12 - \frac{\cos^{-1}(-\tan \phi \tan \delta)}{15^\circ} - \frac{SC}{60}$$
$$\text{sunset} = 12 + \frac{\cos^{-1}(-\tan \phi \tan \delta)}{15^\circ} - \frac{SC}{60}$$
 - Code:**

```
long = -71 ;
lat = 42 ;
timeZone = 'Eastern (UTC-5)';
sc = solarT('Central (UTC-6)', :timeZone);
delta = asin(360*(days - 81)/365);
sunrise = 12 - acosd(-tand(lat)*tand(delta))/15 - sc/60;
sunset = 12 + acosd(-tand(lat)*tand(delta))/15 - sc/60;

plot(days, sunrise, days, sunset, 'LineWidth', 4)
title('Sunrise and Sunset')
xlabel('Day of Year')
```
- Figure:** A plot titled "Sunrise and Sunset" showing "Time of Day" (Y-axis, 0 to 20) versus "Day of Year" (X-axis, 0 to 350). The plot features two lines: a blue line for "Sunrise" and an orange line for "Sunset". The area between these two lines is shaded orange, representing the duration of daylight. The sunrise line starts at approximately 7:30 and ends at 6:30, while the sunset line starts at 6:30 and ends at 7:30.



<https://www.mathworks.com/products/matlab/live-editor.html>

AI Modeling



Model design and
tuning



Hardware
accelerated training



Interoperability

Text Detection and Recognition

Language Translation Using Deep Learning

Leverage Application Specific Reference Examples

- Text Detection and Recognition



Automatically Detect and Recognize Text Using Pretrained CRAFT Network...

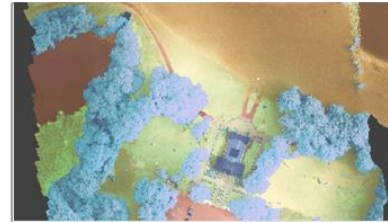
Perform text recognition by using a deep learning based text detector and OCR. In the example, you use a pretrained CRAFT (character region

[Open Live Script](#)



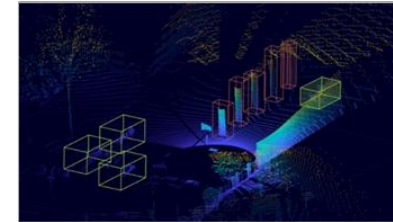
Predictive Maintenance

Anomaly Detection and Condition Monitoring



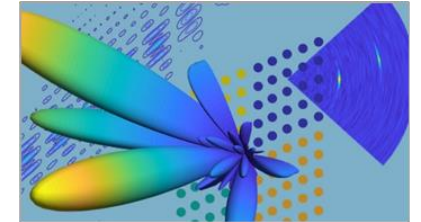
Geospatial Analysis

Hyperspectral Image Classification



Lidar

3-D Point Cloud Object Detection



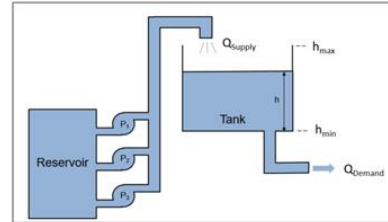
Radar

Waveform Classification



Wireless Comms

Data Synthesis for 5G Channel Estimation



Controls Systems

PID Tuning & System Scheduling



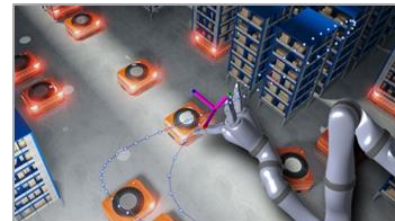
Computational Finance

Trading & Risk Management



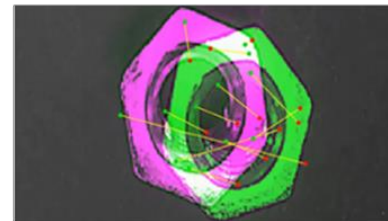
Automated Driving

Pedestrian & Vehicle Detection



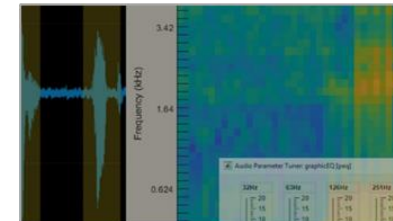
Robotics

Path Planning & Process Optimization



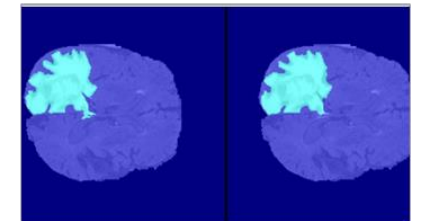
Visual Inspection

Defect Detection



Audio

Speech Recognition

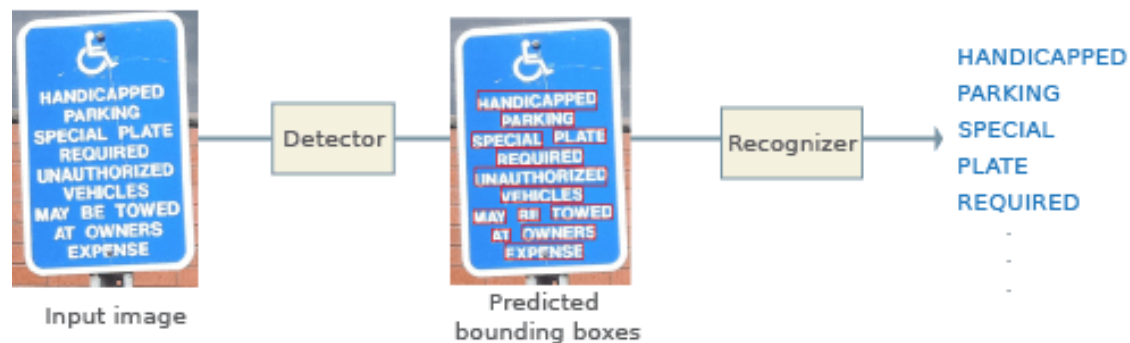


Medical Imaging

Tumor Detection



Text Detection and Recognition



- Use local image features or deep learning to locate or segment text within an image
- Process the text regions to return the predicted text using machine learning or deep learning models



Recognize Text Using Optical Character Recognition (OCR)

Recognize text in images using optical character recognition.



Recognize Seven-Segment Digits Using OCR

Use OCR to recognize seven-segmented digits in text detected by CRAFT and region properties.



Automatically Detect and Recognize Text Using MSER and OCR




Automatically detect and recognize text in images using MSER and OCR.



Automatically Detect and Recognize Text Using Pretrained CRAFT Network...

Perform text recognition by using a deep learning based text detector and OCR. In the example, you use a pretrained CRAFT (character region

OCR Evaluation

| | | |
|--|--|--|
| STEAM | STEAM | STEAM |
| STEAL | TEAM | STREAM |
|  Substitution |  Deletion |  Insertion |

- Character Error Rate (CER)

$$CER = \frac{S + D + I}{N}$$

- Word Error Rate (WER)

$$WER = \frac{S_w + D_w + I_w}{N_w}$$

- MATLAB helper functions:
 - intersect, setdiff, unique...
 - histcounts
- Text Analytics Toolbox

Text Detection and Recognition example – summary

- Detect texts in images by using CRAFT deep learning model
- Recognize text using optical character recognition
- Improve OCR quality through image preprocessing



- Useful links:
 - <https://www.mathworks.com/help/vision/ref/detecttextcraft.html>
 - <https://www.mathworks.com/help/vision/ref/ocr.html>
 - <https://www.mathworks.com/help/images/ref/imagesegmenter-app.html>



Deep Learning - Application Specific Reference Examples

Documentation

Using MATLAB



Applications



AI, Data Science, and Statistics

Products for AI, Data Science, and Statistics

Deep Learning Toolbox

Design, train, and analyze deep learning networks

Statistics and Machine Learning Toolbox

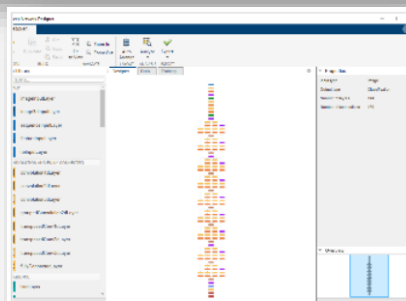
Analyze and model data using statistics and machine learning

Curve Fitting Toolbox

Fit curves and surfaces to data using regression, interpolation, and smoothing

Text Analytics Toolbox

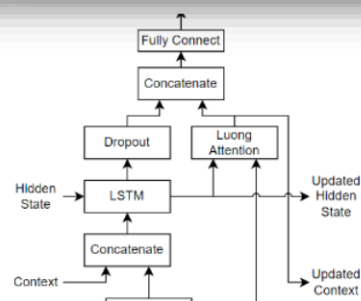
Analyze and model text data



Get Started with Deep Network Designer

Use Deep Network Designer to adapt a pretrained GoogLeNet network to classify a new collection of images. This process is called transfer

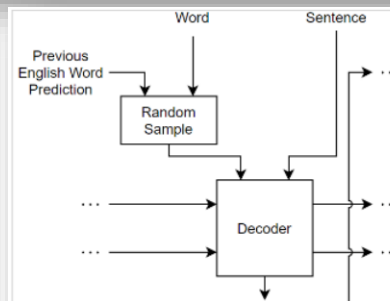
[Open Live Script](#)



Sequence-to-Sequence Translation Using Attention

Convert decimal strings to Roman numerals using a recurrent sequence-to-sequence encoder-decoder model with attention.

[Open Live Script](#)

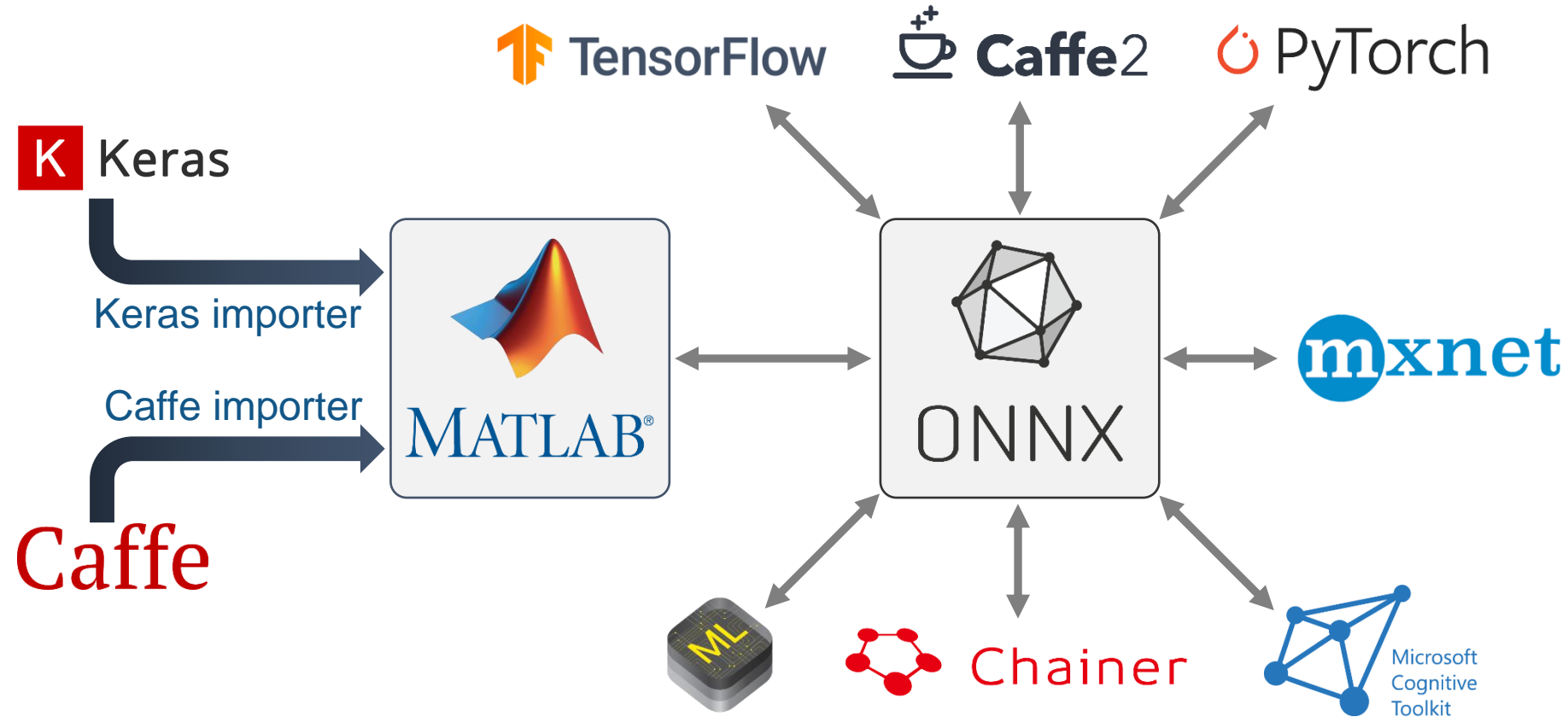


Language Translation Using Deep Learning

Train a German to English language translator using a recurrent sequence-to-sequence encoder-decoder model with attention.

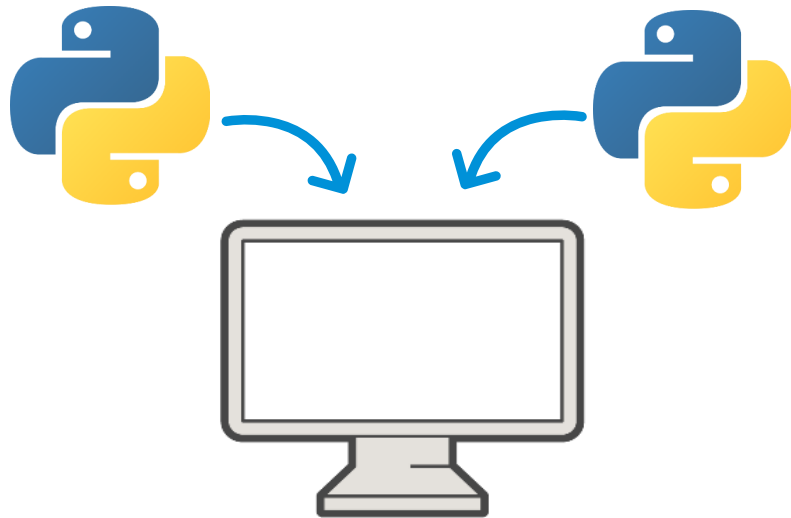
[Open Live Script](#)

Model Interoperability



<https://www.mathworks.com/solutions/deep-learning/models.html>

Language Translation Using Deep Learning



Call Python from MATLAB

- Transformers library:
 - <https://huggingface.co/>



**The AI community
building the future.**

Build, train and deploy state of the art models powered by
the reference open source in machine learning.

- T5-based models
 - <https://github.com/sdadas/polish-nlp-resources>
 - MT5 and Flan-T5 models fine-tuned for Polish-English and English-Polish translation. The models were trained on 70 million sentence pairs from OPUS

Call Python from MATLAB

- Set-up your Python environment

```
pyenv
```

```
ans =
```

```
PythonEnvironment with properties:
```

```
Version: "3.9"
```

```
Executable: "C:\Users\ydebray\AppData\Local\WPY64-39100\python-3.9.10.amd64\python.exe"
```

```
Library: "C:\Users\ydebray\AppData\Local\WPY64-39100\python-3.9.10.amd64\python39.dll"
```

```
Home: "C:\Users\ydebray\AppData\Local\WPY64-39100\python-3.9.10.amd64"
```

```
Status: NotLoaded
```

```
ExecutionMode: OutOfProcess
```

```
pyenv('Version', 'C:\Users\uname\WinPython-64bit-3..2.1\python-3..2.amd64\python.exe')
```

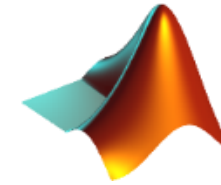
https://www.mathworks.com/help/matlab/matlab_external/install-supported-python-implementation.html

Syntax differences when calling Python from MATLAB



Python

```
>>> import math
>>> math.sqrt(42)
```



MATLAB

```
>> py.math.sqrt(42)
```

```
>>> print('hello', 'world', sep=', ')
```



```
>> py.print('hello', 'world', ...
            sep = ', ')
```

Before
22a

```
>> py.print('hello', 'world', ...
            pyargs('sep', ', '))
```




Language Translation Using Deep Learning - example

```
class Translate:  
    def __init__(self, model_path):  
        self.tokenizer = T5Tokenizer.from_pretrained(model_path)  
        self.model = T5ForConditionalGeneration.from_pretrained(model_path, device_map=  
        self.model.to("cuda")
```

```
pyLibT = py.importlib.import_module('translateLib')
```

```
pyLibT =
```

```
Python module with properties:
```

```
Translate: [1x1 py.type]  
T5Tokenizer: [1x1 py.type]  
T5ForConditionalGeneration: [1x1 py.type]
```

```
disp(oryginaltext3)
```

```
"Designed for the way you think and the work you do"  
" MATLAB® combines a desktop environment tuned for iterative analysis and design."  
" It includes the Live Editor for creating scripts that combine code, output, and  
" Professionally Built MATLAB toolboxes are professionally developed, rigorously  
" With Interactive Apps MATLAB apps let you see how different algorithms work w  
" Iterate until you've got the results you want, then automatically generate a  
" And the Ability to Scale Scale your analyses to run on clusters, GPUs, and cl  
" There's no need to rewrite your code or learn big data programming and out-of  
""
```

```
disp(textpl3)
```

```
"Zaprojektowany dla Twojego sposobu myślenia i pracy, którą wykonujesz"  
"MATLAB® łączy środowisko pulpitu dostosowane do analizy iteracyjnej i procesów  
"Obejmuje on Live Editor do tworzenia skryptów, które łączą kod, wyjście i sfor  
"Profesjonalnie budowane skrzynki narzędzi MATLAB są profesjonalnie opracowane,
```

Cheat Sheets for Using MATLAB with Python

<https://www.mathworks.com/campaigns/offers/matlab-python-cheat-sheets.html>

- ✓ Using MATLAB and Python Together
- ✓ MATLAB for Python Users
- ✓ MATLAB Basic Functions Reference
- ✓ Importing and Exporting Data
- ✓ Preprocessing Time Series Data

Python Type to MATLAB Scalar Type Mapping

When you pass Python® data as input arguments to MATLAB® functions, the MATLAB Engine for Python converts the data into equivalent MATLAB data types.


| Python Input Argument Type — Scalar Values Only | Resulting MATLAB Data Type |
|---|---|
| float | double |
| complex | Complex double |
| int | int64 |
| long (Python 2.7 only) | int64 |
| float(nan) | NaN |
| float(inf) | Inf |
| bool | logical |
| str | char |
| unicode (Python 2.7 only) | char |
| dict | Structure if all keys are strings not supported otherwise |

Python Container to MATLAB Array Type Mapping

| Python Input Argument Type — Container | Resulting MATLAB Data Type |
|--|----------------------------|
| matlab numeric array object (see MATLAB Arrays as Python Variables) | Numeric array |
| bytearray | uint8 array |
| bytes (Python 3.x) | uint8 array |
| bytes (Python 2.7) | char array |
| list | Cell array |
| set | Cell array |
| tuple | Cell array |



Using MATLAB® and Python® Together

The  icon provides links to relevant sections of the MATLAB documentation to learn more.

Call Python in MATLAB

Access settings and status of Python interpreter:
`>> pe = pyenv`


Specify version to use:
`>> pe = pyenv('Version', '3.7')`

Call Python modules and functions:
`py.module_name.function_name`
`>> py.math.sqrt(42)`

Pass keyword arguments
 Use `pyargs` to pass keyword arguments
`>> foo(5, bar=42)`
`>> py.foo(5, pyargs('bar', 42))`

Reload modules
 Reload the module after making updates:
`>> py.importlib.reload(module)`

Call MATLAB in Python

Install MATLAB Engine API for Python 
 Run `setup.py` from an OS command window
`$ cd [matlabroot]/extern/engines/python`
`$ python setup.py install`


Call MATLAB functions
 Import the module and start the engine
`>>> import matlab.engine`
`>>> eng = matlab.engine.start_matlab()`

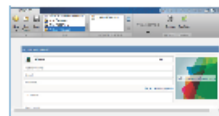
Call functions through the engine
`>>> x = eng.sqrt(42.0)`

Capture multiple outputs
`>>> x = eng.gcd(42.0, 8.0, nargout=3)`

Stop the engine
`>>> eng.exit()`

Create Python Package


Package MATLAB functions 
 Use the Library Compiler App to create a Python package for MATLAB functions



Invoke MATLAB functions from the Python package
`>>> import PackageName`
`>>> pkg = PackageName.initialize()`
`>>> result = pkg.foo()`

Close package
`>>> pkg.terminate()`

Data Type Conversions

Data types will be automatically  converted where possible.

| MATLAB | Python |
|---------------------------------------|---------------|
| double, single | float |
| complex single | complex |
| complex double | complex |
| (u)int8, (u)int16, (u)int32, (u)int64 | int |
| NaN | float(nan) |
| Inf | float(inf) |
| string, char | str |
| Logical | bool |
| Structure | dict |
| Vectors | array.array() |
| Cell array | list, tuple |

Some MATLAB data types need to be converted.


| MATLAB | Conversion Function |
|-------------|---------------------|
| categorical | char |
| string | char |
| table | table2struct |
| timetable | timetable2struct |
| datetime | char |


Note: The default numeric type is integer in Python and double in MATLAB when typing `42`

To create a float in Python:
`>>> x = 42.0`
`>>> x = float(42)`

Create integer from MATLAB:
`>>> x = int32(42)`

Data Science Libraries


Apache Parquet 



Use Apache Parquet to efficiently transfer tabular data

From MATLAB:
`>> tbl = parquetread(fname)`
`>> parquetwrite(tbl, fname)`

From Python:
`>>> df = pandas.read_parquet(fname)`
`>>> pandas.DataFrame.to_parquet(df)`

Deep Learning 
 Access frameworks in MATLAB with importers for TensorFlow-Keras, ONNX, etc.
`>> net = importKerasNetwork(model)`

Deployment



Embedded devices



Enterprise systems



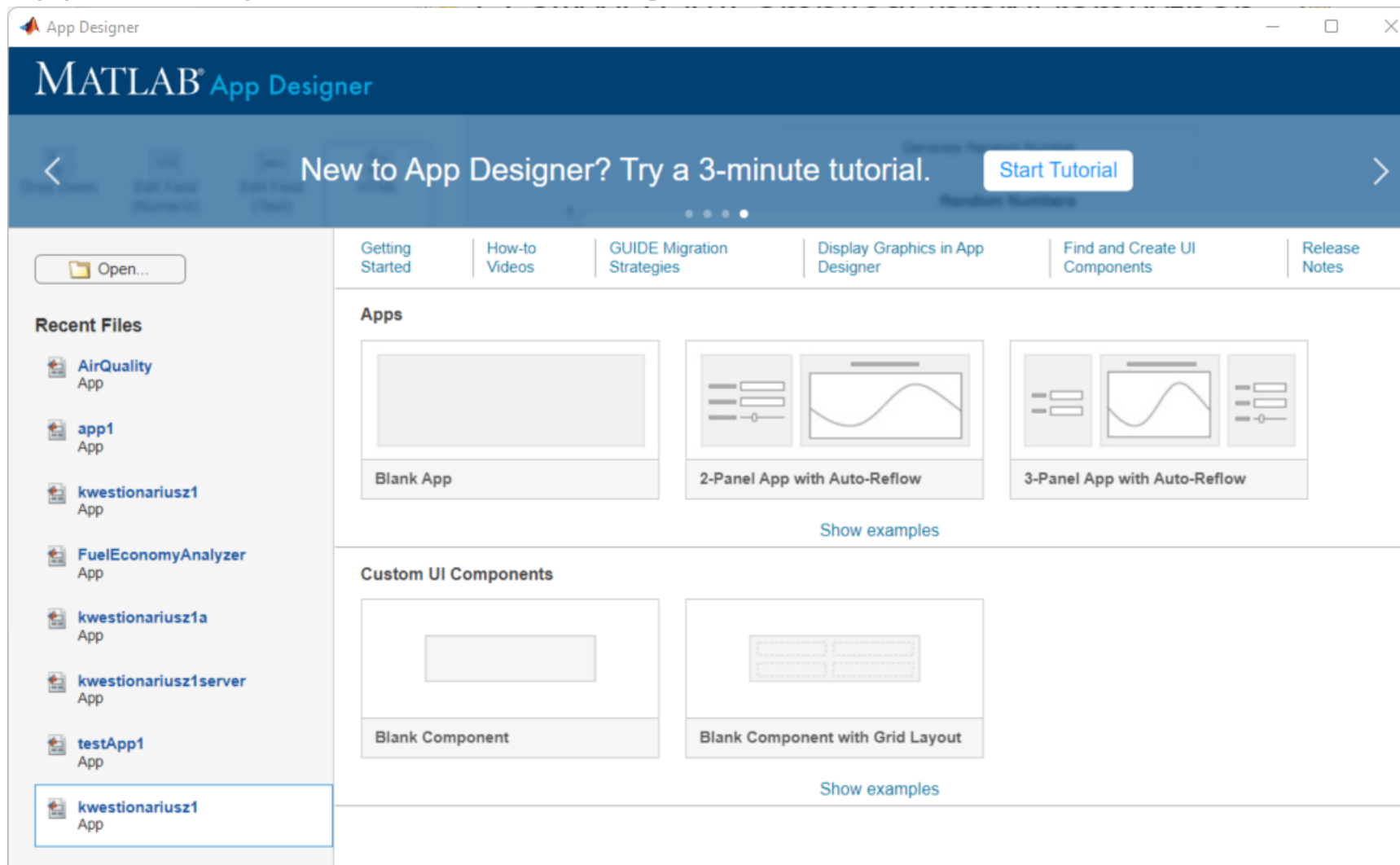
Edge, cloud,
desktop

App Building (app designer) Deployment (MATLAB WebApp Server)

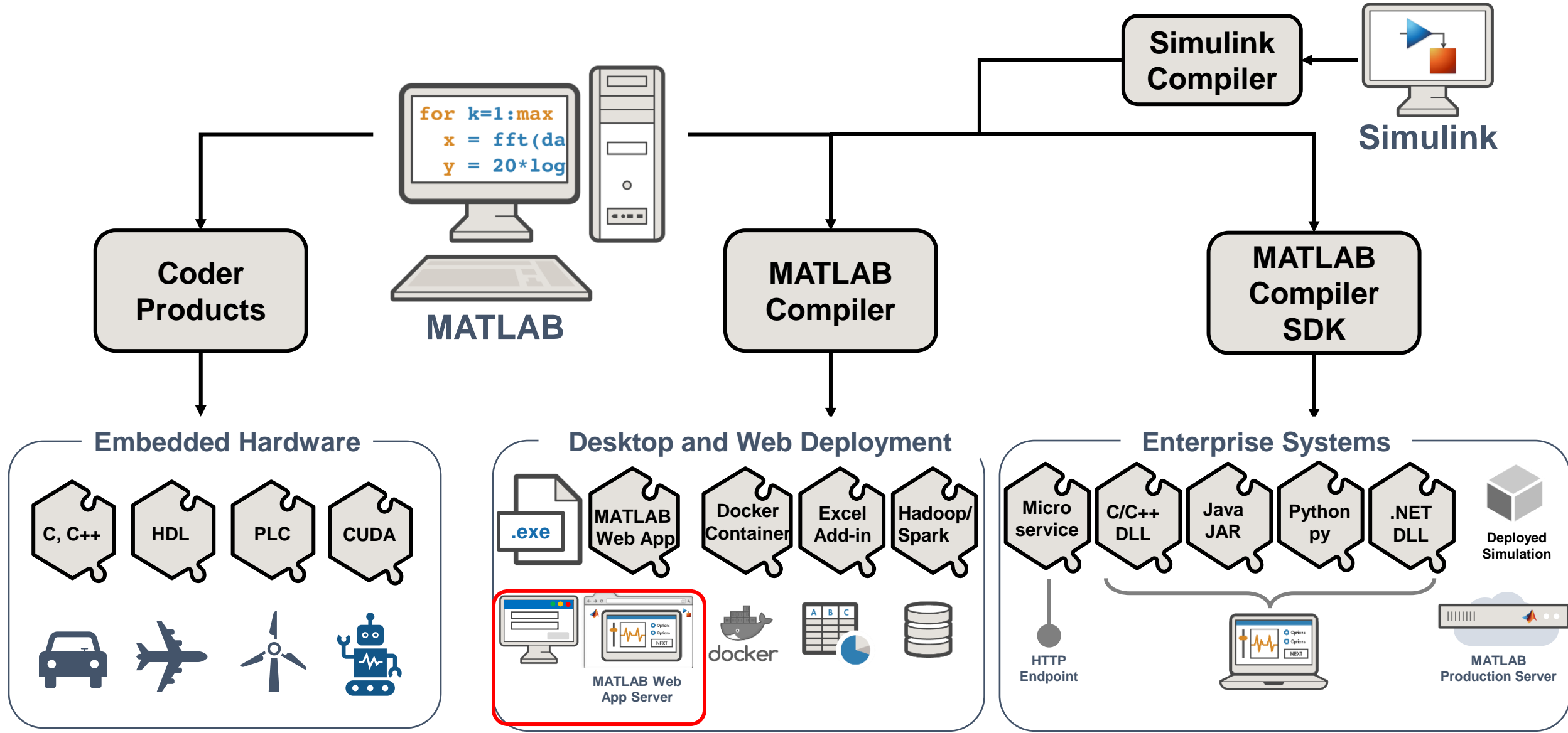


MATLAB App Designer

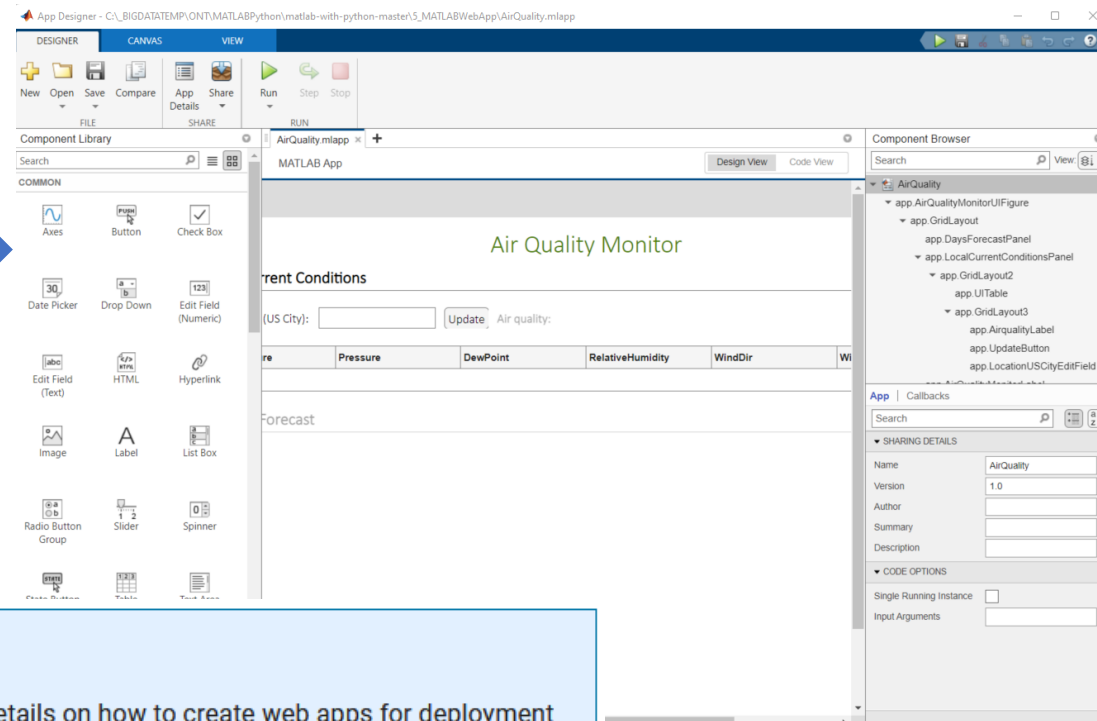
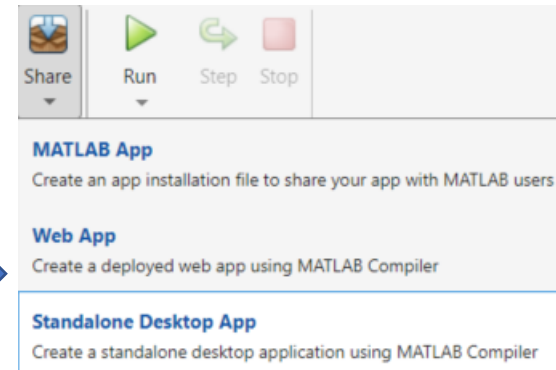
- App development for Non-Programmers



Deployment Overview



Share MATLAB App in the Web



Note

MATLAB apps deployed to the server are created using MATLAB Compiler. For details on how to create web apps for deployment to the server, see [Web Apps \(MATLAB Compiler\)](#).



Oprogramowanie
Naukowo-Techniczne
sp. z o.o.

www.ont.com.pl



matlab.pl



oprogramowanie-
naukowo-techniczne



ONT MATLAB

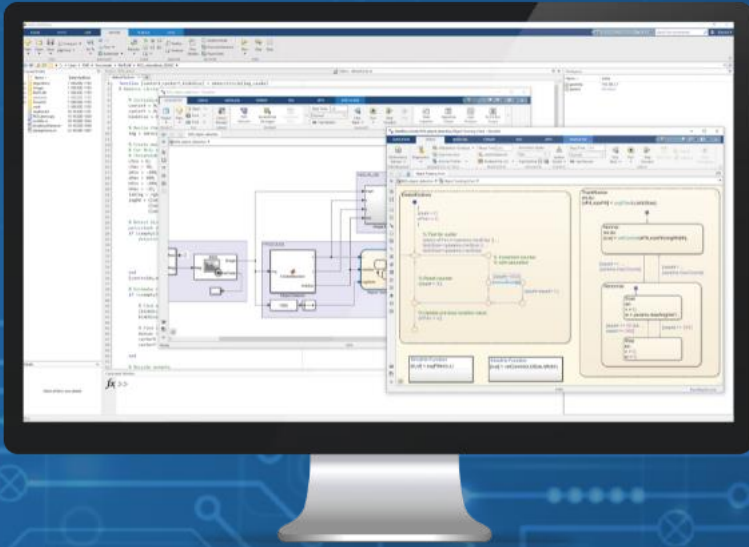


Jaromir Przybyło

Application Engineer, ONT

jaromir.przybylo@ont.com.pl

APPLICATIONS



- ▶ Robotics and Automation
- ▶ Computational Finance
- ▶ Autonomous Vehicles
- ▶ Electronics
- ▶ Artificial Intelligence
- ▶ Biomedical Engineering
- ▶ Systems Engineering and certification
- ▶ Power Electronics and Systems
- ▶ Communications and Radar Systems

Let's stay in touch

Oprogramowanie Naukowo-Techniczne sp. z o.o.
MATLAB and Simulink authorised reseller for Poland
ul. Pod Fortem 19, 31-302 Kraków, Poland | www.ont.com.pl