National Competence Center

Work Package

Artificial Intelligence and Machine Learning

František Hakl, Petra Vidnerová

Institute of Computer Science

Sep 2019

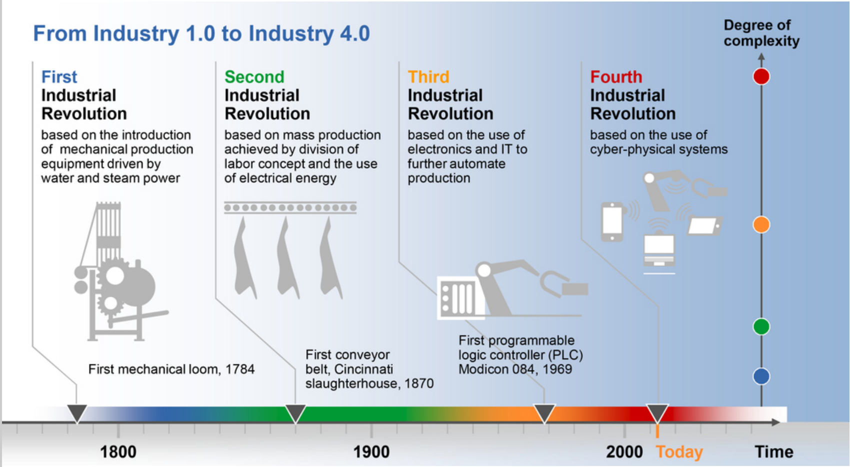
Project: **National Competence Center-Cybernetics and Artificial Intelligence**
Project identification code: TN01000024
Programme: National Centres of Competence 1
Donor: Technology Agency of the Czech Republic

- project ID: TN01000024
- participants:
  University - CTU (Prague), TUO (Ostrava), BUT (Brno), UVB (Plzeň)
  Academy of Sciences - ICS, IITA, IT
  16 industry partners (Skoda auto, Siemens, ...)
- duration: 2019-2020 (...2027)
- budget: 80 MCZK p.a.

Work package **WP1**: Robotics and Cybernetics for Industry/Society 4.0
Sub-project name: Artificial Intelligence and Machine Learning

- project ID: TN01000024/03
- participants: CTU Prague, ICS AC CR, Siemens
- duration: 2019-2020 (...2027)
- budget: WP1 7250 kCZK p.a., ICS 2358 kCZK p.a.
- František Hakl, David Coufal,
  Petra Vidnerová, Věra kůrková

Industry (r)evolution



From Industry 1.0 to Industry 4.0

First Industrial Revolution

based on the introduction of mechanical production equipment driven by water and steam power

First mechanical loom, 1784

Second Industrial Revolution

based on mass production achieved by division of labor concept and the use of electrical energy

First conveyor belt, Cincinnati slaughterhouse, 1870

Third Industrial Revolution

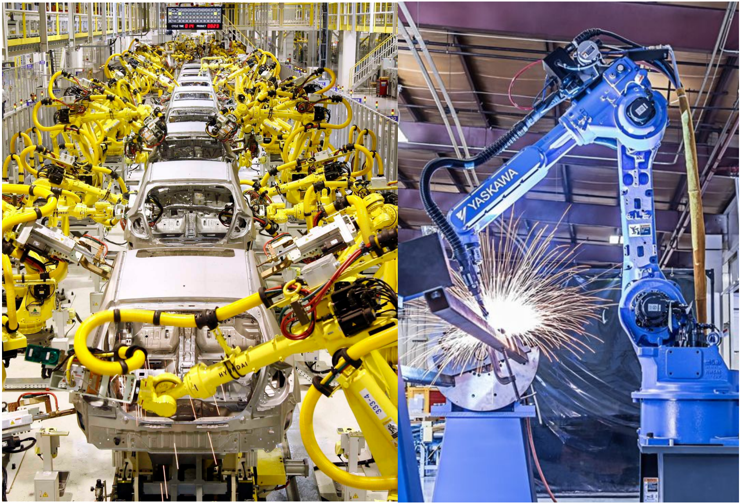
based on the use of electronics and IT to further automate production

First programmable logic controller (PLC) Modicon 084, 1969

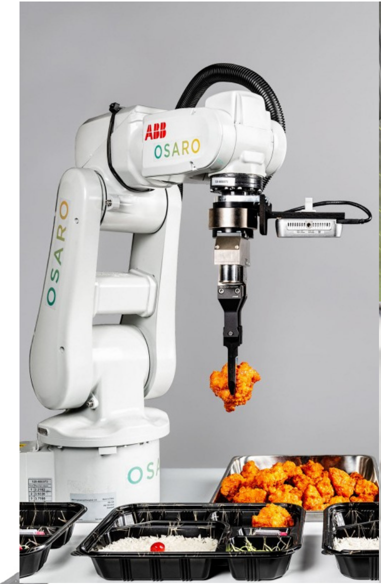
Fourth Industrial Revolution

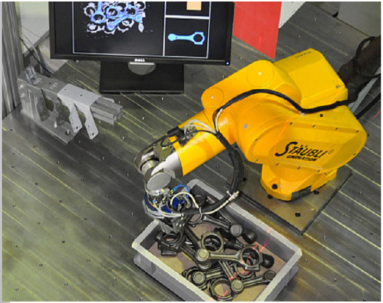
based on the use of cyber-physical systems

"trivial" (predefined motions and actions)



NONtrivial (based on 3D vision and scenery understanding)





Main industry objectives

... the research and application of ML/AI methods in various fields of industrial automation:

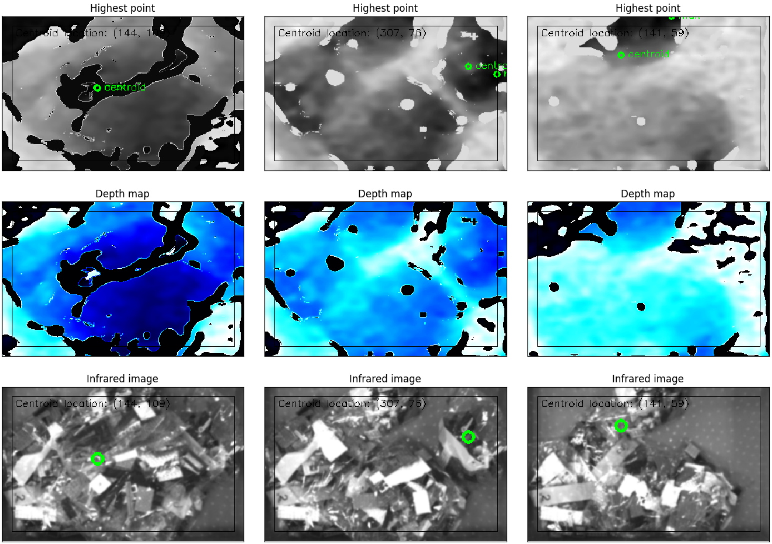
- **Grasping and handling of objects** – so-called bin picking and its generalization of handling packs of various shapes and sizes that can be hardly described in an analytical way. Such packs may contain different kinds of parts, whose position in the pack is random.
- **Force-compliant moving and handling** – a lower-level control to deal with the force feedback to allow pick & place operations for more complex tasks, especially when amorphous objects are to be handled.
- **Mimicking of a human worker** – hand guiding of the robot to be taught individual pieces of its trajectory and places where and how to handle the parts.

Main AI objectives

... general-purpose library of ML/AI methods:

- **Methods selection** – design, developement and testing algorithms and tools based on a suite of techniques, including deep neural networks, in complex systems to improve their functionality and performance.
- **Data processing** – focus on regression and classification methods for large data sets, regularization techniques for such data, fast classifiers for high-dimension data processing, algorithms for efficient machine learning based on kernel methods, meta-learning algorithms, detection and prevention of adversarial patterns.
- **Results** – will be in the form of software implemented on a range of architectures, including those dedicated for machine learning problems such as GPUs or specialized neuroprocessors.

Example of three LEGO scenes seen by the eyes of a robot (234x366 px)



Highest point — Highest point — Highest point

Depth map — Depth map — Depth map

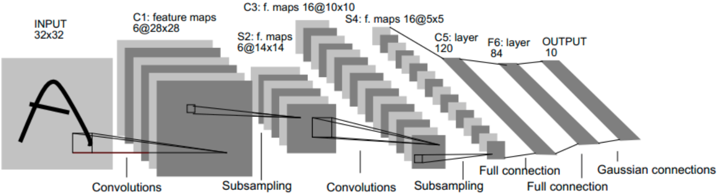
Infrared image — Infrared image — Infrared image

Centroid location: (144, 109)
centroid
Centroid location: (307, 76)
centro
Centroid location: (141, 59)
centroid
Centroid location: (144, 109)
Centroid location: (307, 76)
Centroid location: (141, 59)

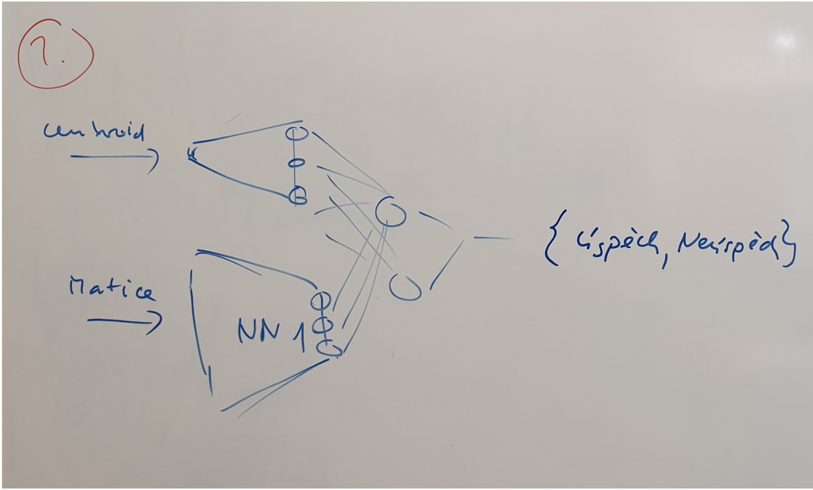
Task definition

Goal: detect difficult situations
classification task
$f : (centroid(vector), image(matrix)) \to \{0, 1\}$

Convolutional network - classifier for images:

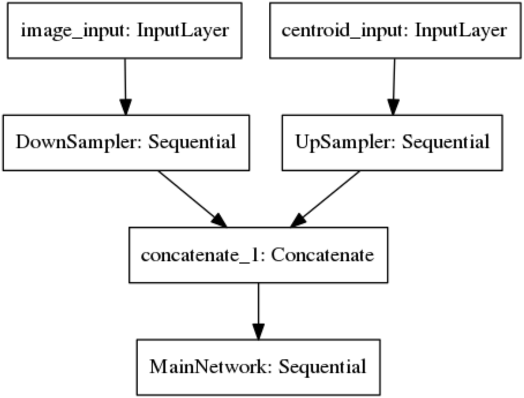


Network proposal



Implementation

Python, Keras, tensorflow
runs on GPUs (experiments on the cluster *haklnv* in our institute)



image_input: InputLayer

centroid_input: InputLayer

DownSampler: Sequential

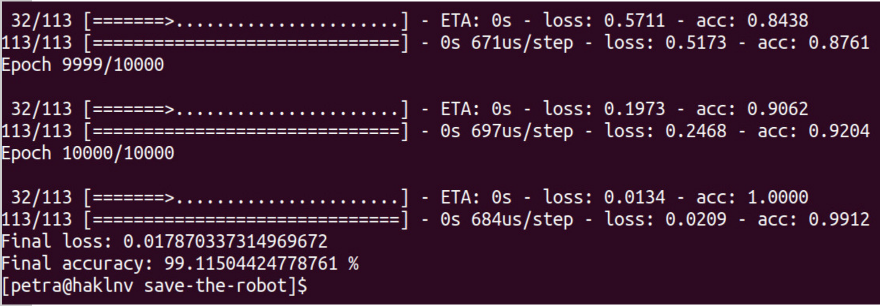
UpSampler: Sequential

concatenate_1: Concatenate

MainNetwork: Sequential

Preliminary results

Perfect fit on trainset - it is possible to achieve 100% classification accuracy.
Low crossvalidation error - best results ∼ 65% classification accuracy !!!
Network overfits - need for bigger data sets.



```
 32/113 [=======>......................] - ETA: 0s - loss: 0.5711 - acc: 0.8438
113/113 [==============================] - 0s 671us/step - loss: 0.5173 - acc: 0.8761
Epoch 9999/10000

 32/113 [=======>......................] - ETA: 0s - loss: 0.1973 - acc: 0.9062
113/113 [==============================] - 0s 697us/step - loss: 0.2468 - acc: 0.9204
Epoch 10000/10000

 32/113 [=======>......................] - ETA: 0s - loss: 0.0134 - acc: 1.0000
113/113 [==============================] - 0s 684us/step - loss: 0.0209 - acc: 0.9912
Final loss: 0.017870337314969672
Final accuracy: 99.11504424778761 %
[petra@haklnv save-the-robot]$
```

Conclusion

Existing outputs:

- Python scripts for data preprocessing
- code for networks for LEGO classification task

Work in progress:

- adopt code for large data sets (does not fit into memory)
- optimize architecture of DownSampler, UpSampler, MainNetwork modules
- adapt the code for new tasks (situation in factory changed)

Main difficulties met:

- data acquisition needs manual control
- difficult to generate enough data samples

Thank you! Questions?