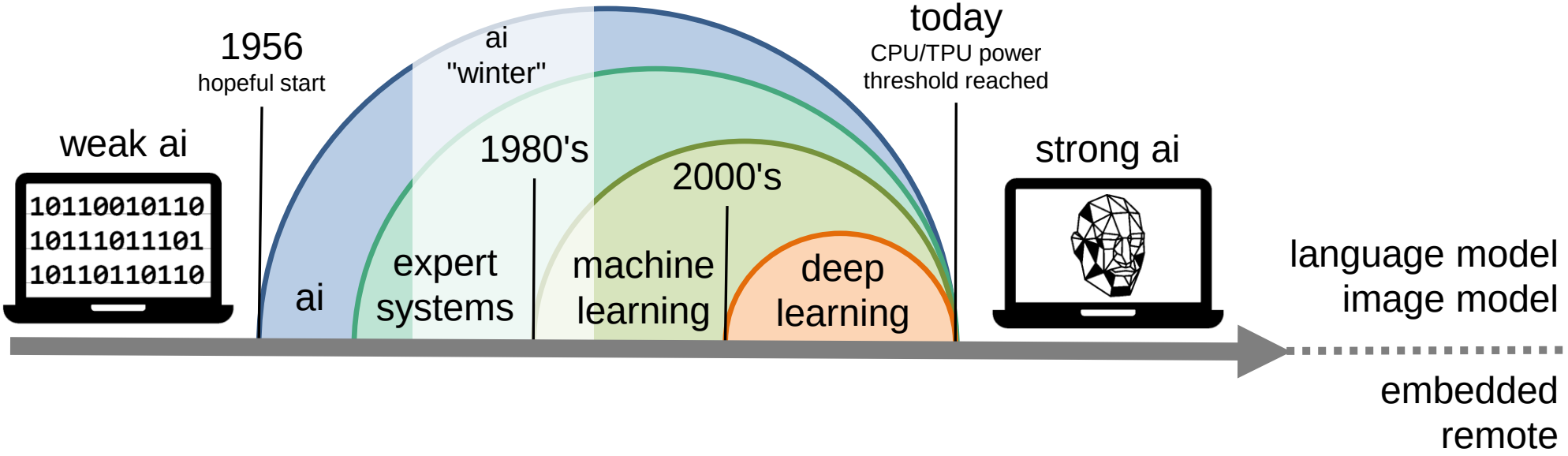
Novel Approach for Teaching AI in Entry Level Education

Christian Bildhauer-Buggle, Samuel Kübler, Thomas Schiepp

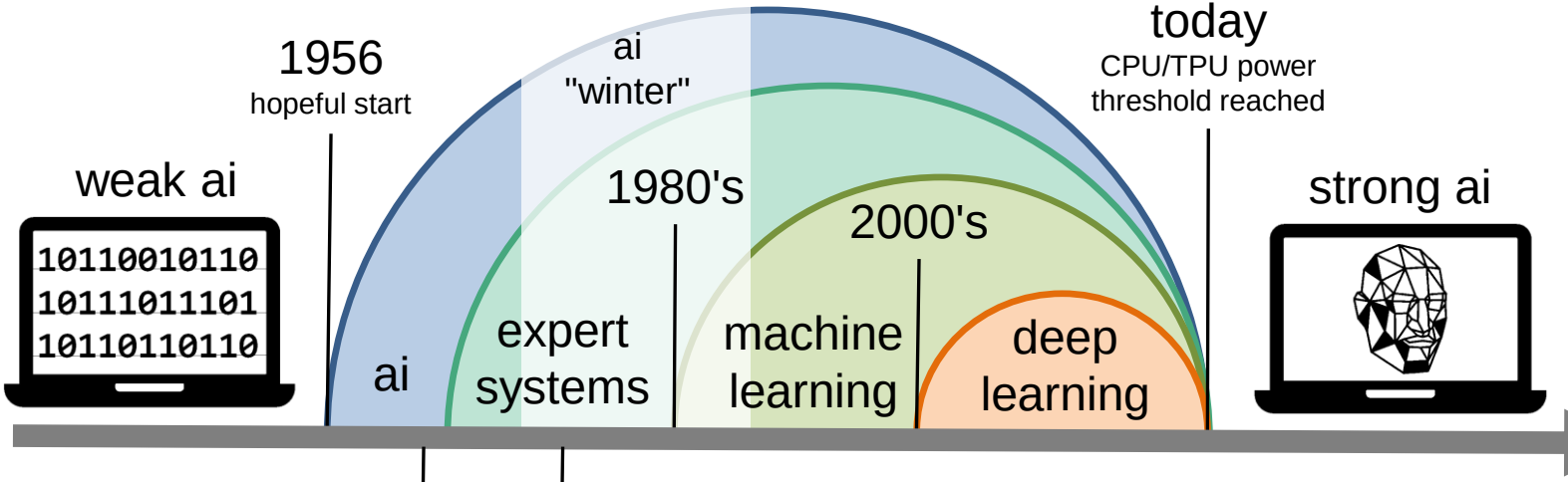
Furtwangen Hochschule University, Faculty of Medical and Mechanical Engineering,
Campus Schwenningen, Jakob-Kienzle-Straße 17, 78054 VS-Schwenningen, Germany

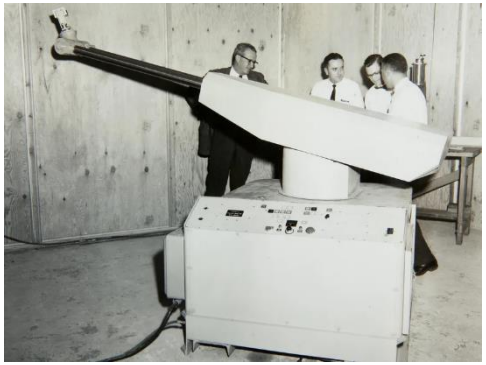
15.3.2024

motivation: past, present, future



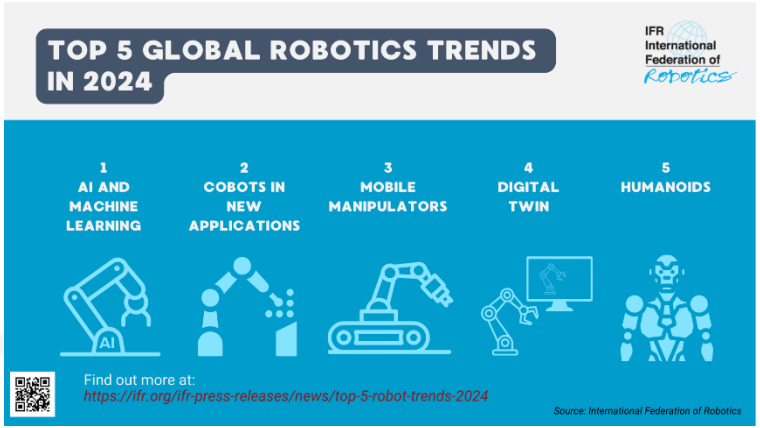
motivation: past, present, future





1961 1971

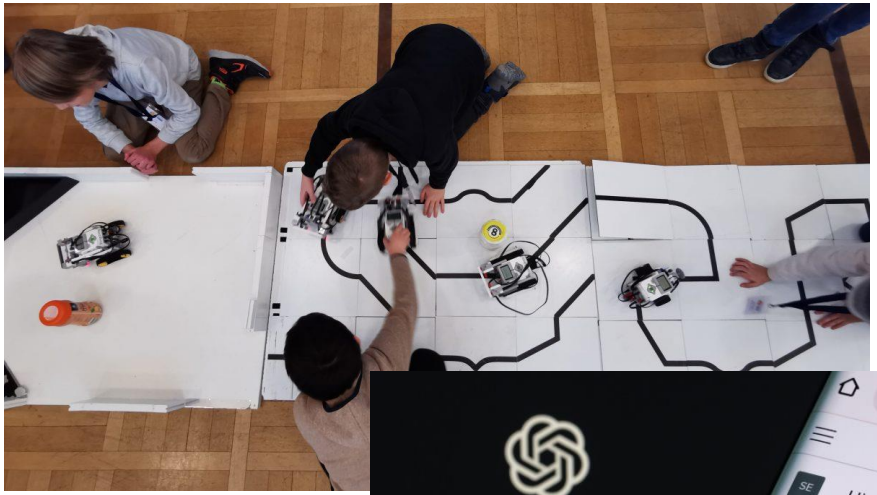
today 3,9 million robots

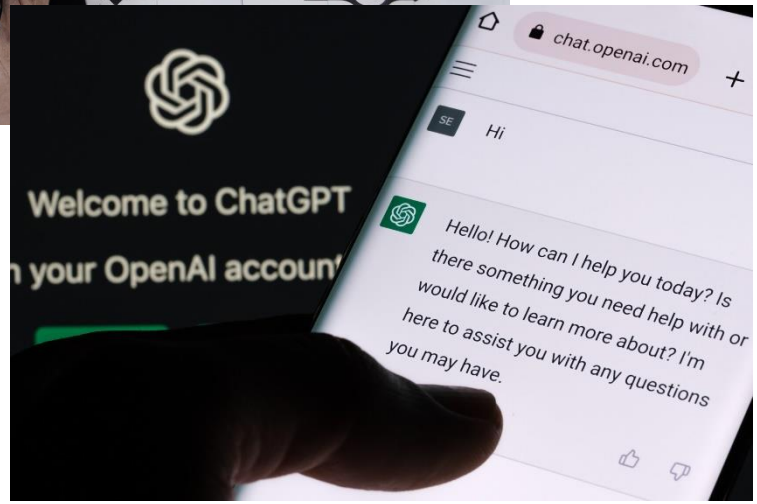


motivation: education

typical robot exercises are in mobile robots (RoboCup):
inexpensive components, yet all technologies:

sensors, drive mechanics, mechatronics, electronics, programming, …





however this is not ai !
embedded software running an **algorithm**

ChatGPT is ai, but no real-world interaction

first level exercise: program the robot

HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

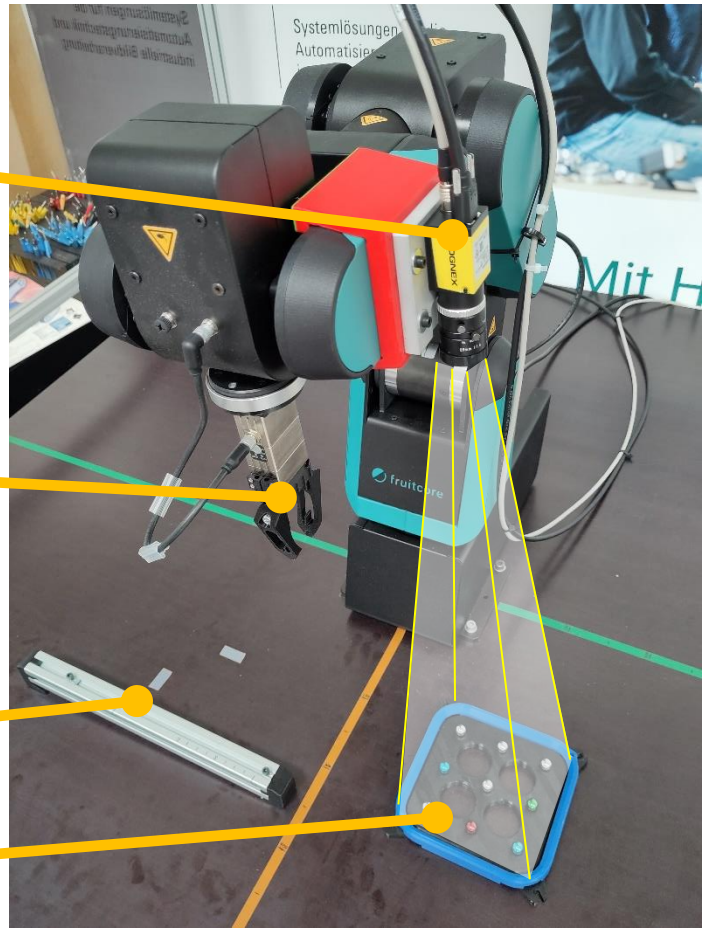
creating a real-live application:

pick-up, dip and drop-off

first level exercise: program the robot

creating a real-live application:

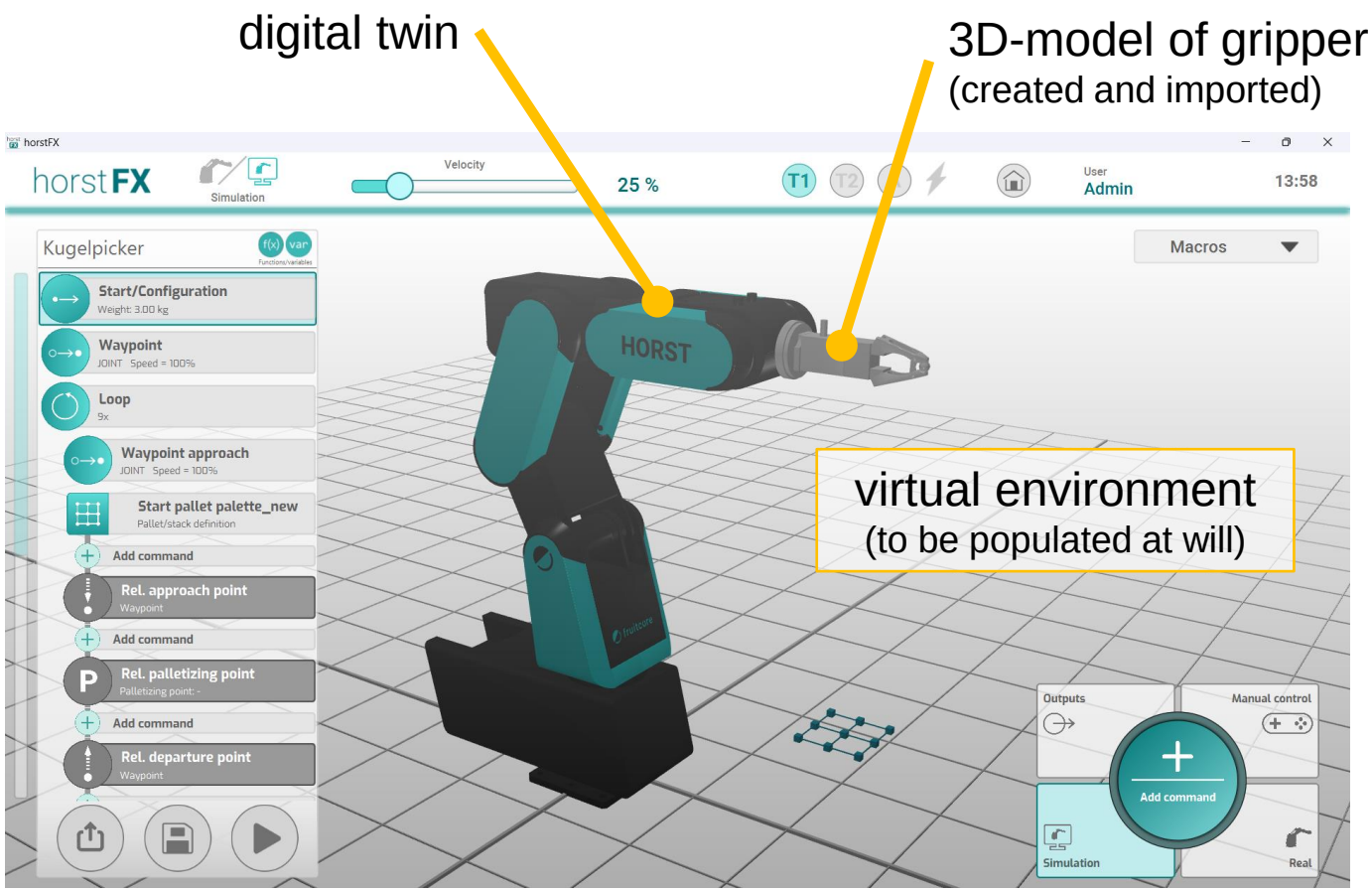
pick-up, dip and drop-off



first level exercise: program the robot

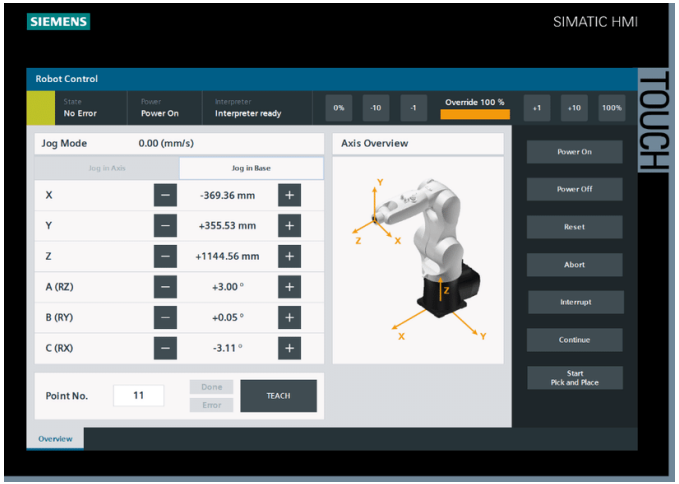
SW can run on
- normal desktop
- panel-PC
- true robot control unit

identical interface allows for **offline exercises**, then transfer of RP's



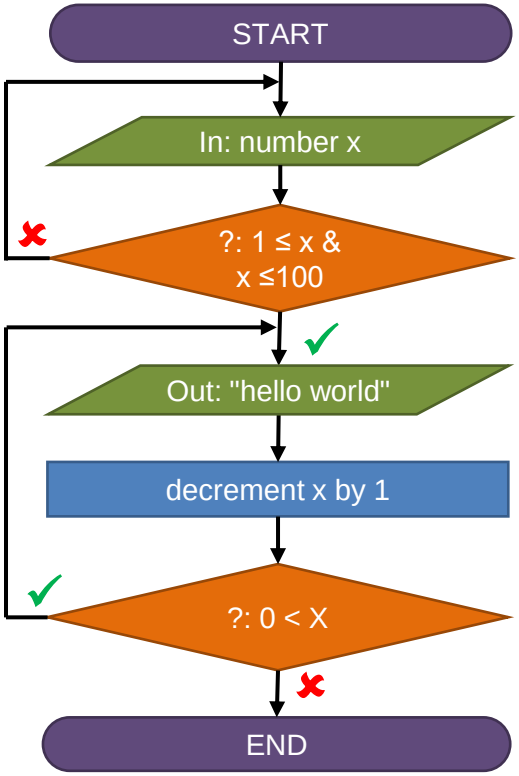
first level exercise: program the robot

all robot manufacturers optimize their interface for ease-of-use…..

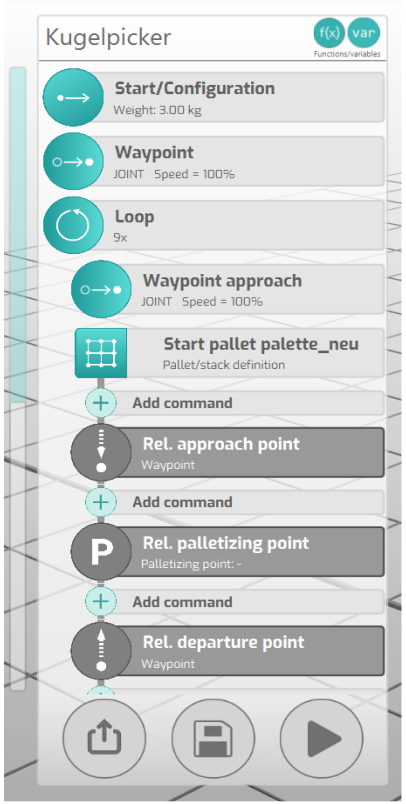


first level exercise: program the robot

Robot Program (RP): a list of sequential commands (~ assembler or BASIC)
loop and *if-then-else* logic by jumps in RP-list



inspiration



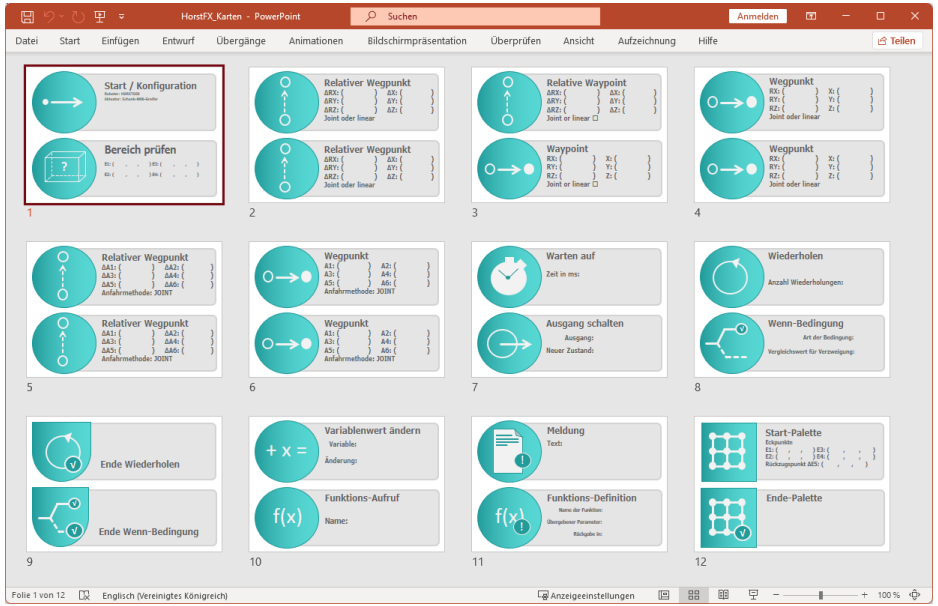
inspiration

step1:
create RP with cards

first level exercise: program the robot

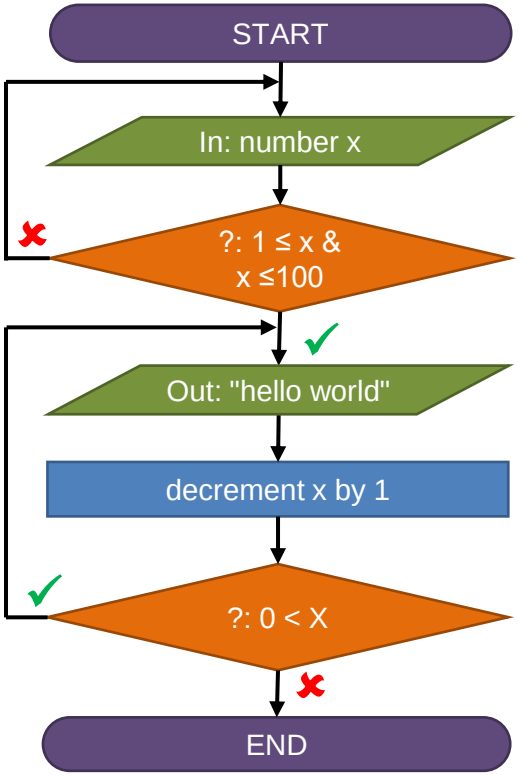
Inspiration by a commercial board game: RoboRally ©: movements need to be "programmed" by shuffled and dealt cards 6 movements ahead

"copy" cards with PP, print and laminate a sufficient number:

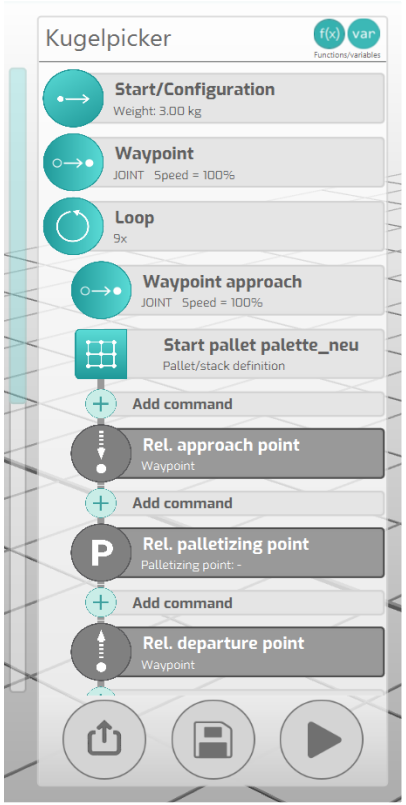


first level exercise: program the robot

Robot Program (RP): a list of sequential commands (~ assembler or BASIC)
loop and *if-then-else* logic by jumps in RP-list



inspiration



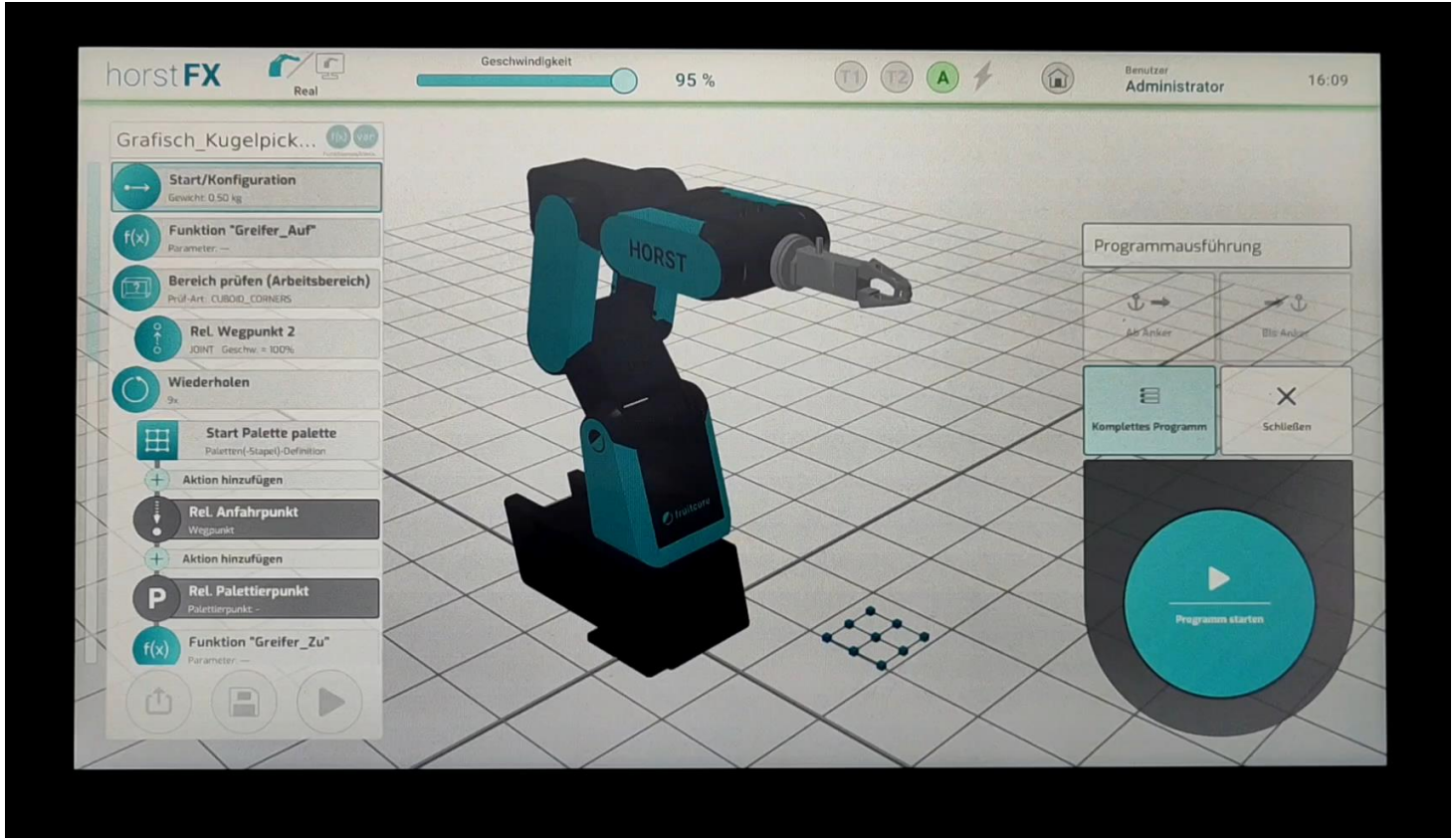
inspiration

step1:
create RP with cards

step2:
follow cards to create RP

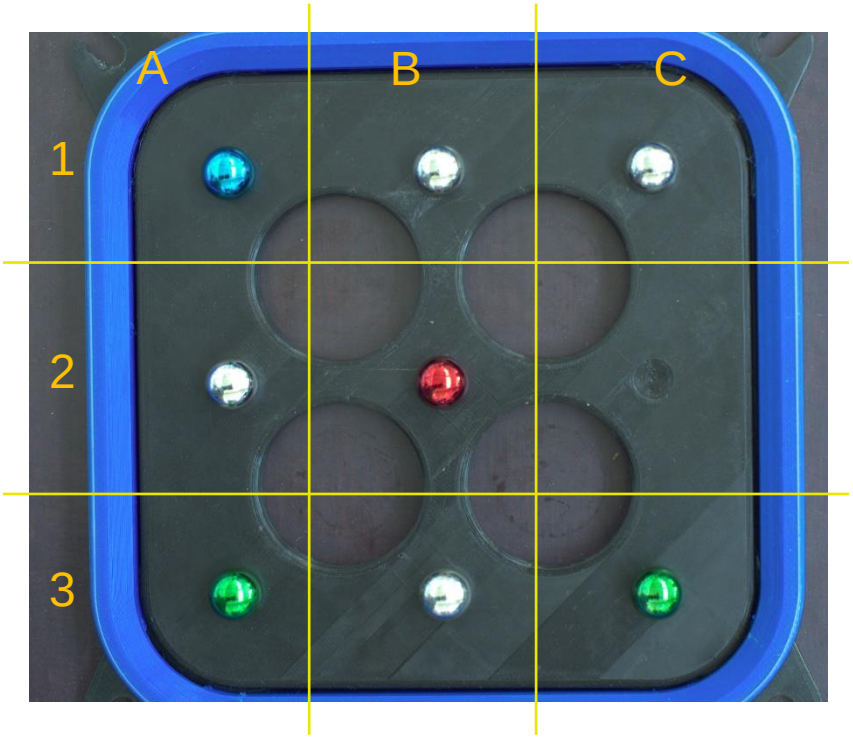
AI enhanced robot

Exercise typically completed in 45 minutes, including short introduction



AI enhanced robot

task for DeepLearning



pick only "good" colored balls: SilverBall
skip the "bad" ones (BlueBall, RedBall, GreenBall)
and ignore empty slots !

use a commercial Deep-Learning-ai software which is designed for fault recognition in quality control

user must define patterns, for which the ai-module searches

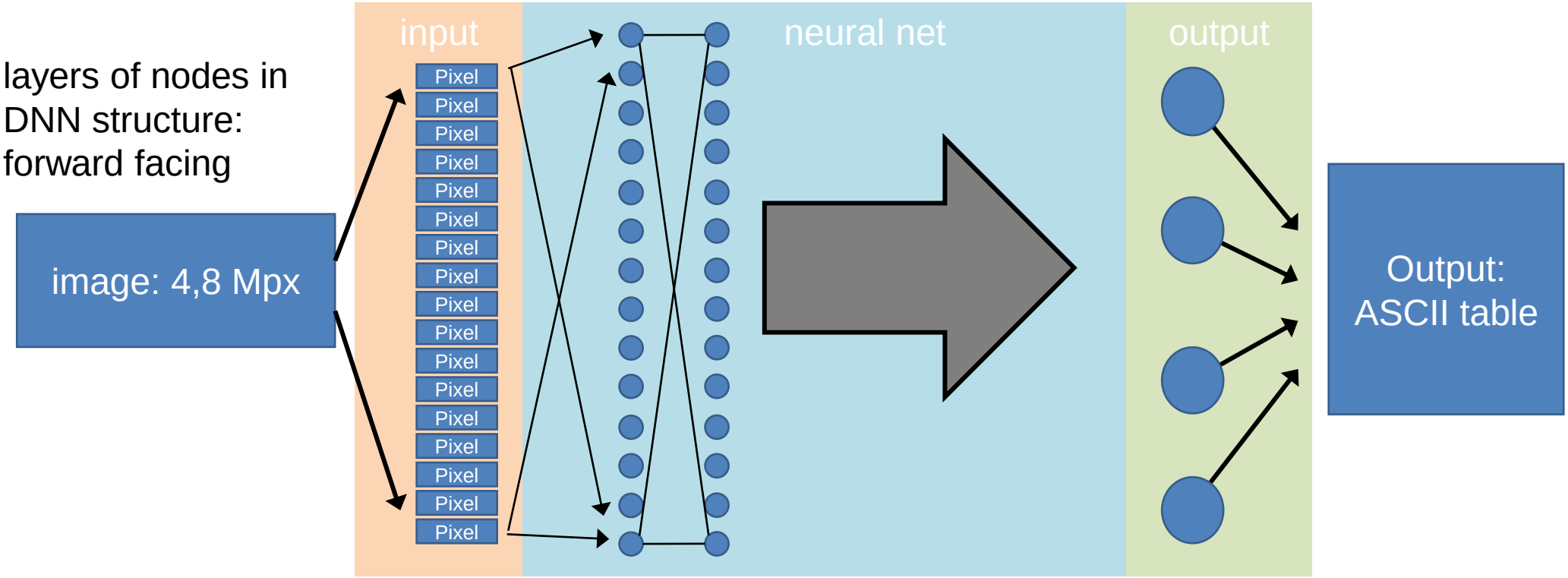
ai will return found patterns in an ASCII table
user-defined-name | *X-position** | *Y-position**

*with positions in pixel-count

HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

AI: DeepLearning in neural net

layers of nodes in DNN structure: forward facing

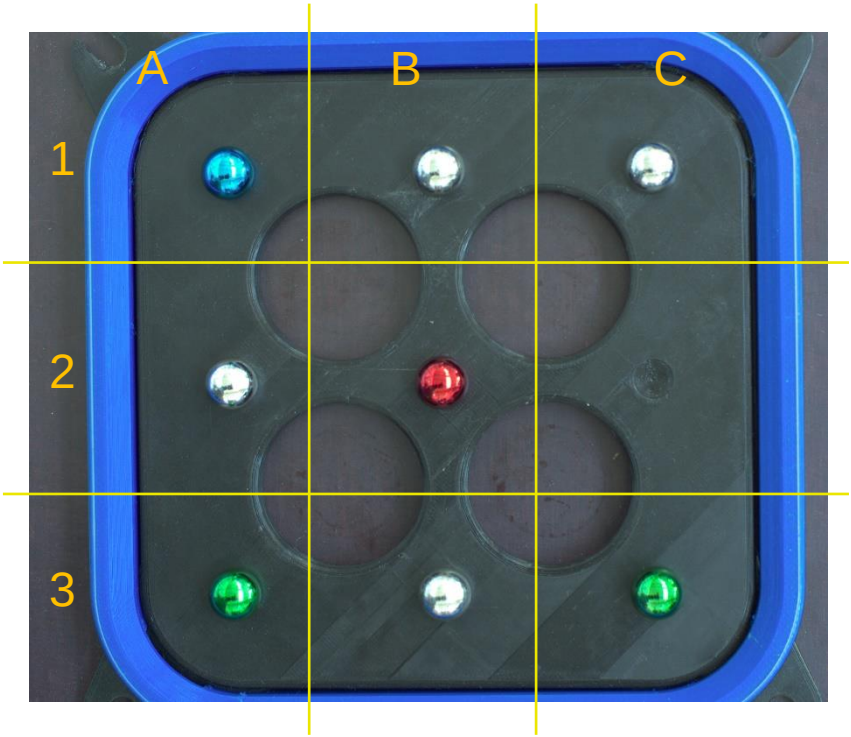


AI enhanced robot

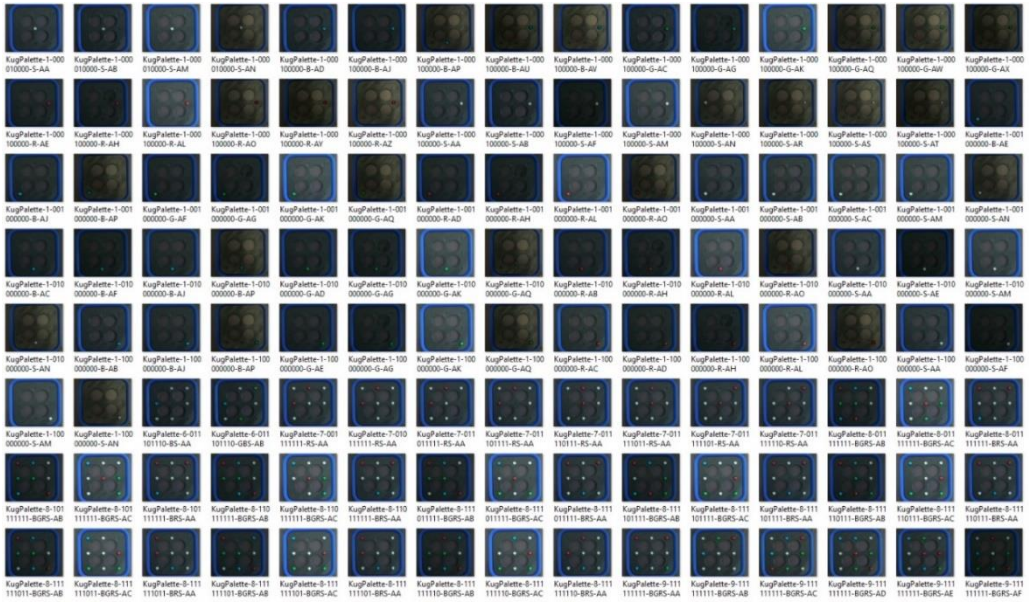
HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

task for DeepLearning

pick only "good" colored balls: SilverBall
skip the "bad" ones (BlueBall, RedBall, GreenBall)
and ignore empty slots !



Step1: gather appropriate collection of images



AI enhanced robot

HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

task for DeepLearning

Step2 import all images into DL-GUI

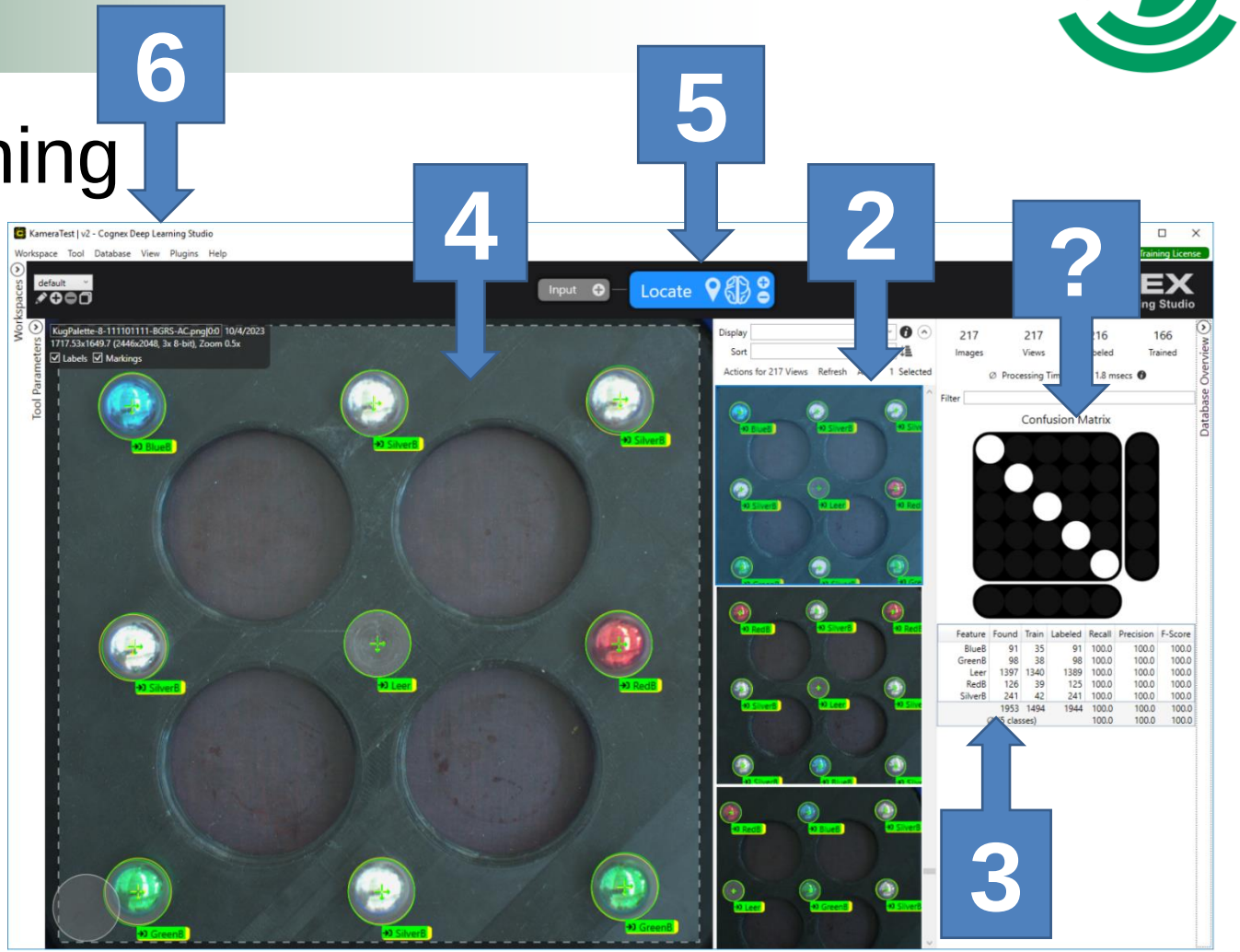
Step3 define objects with names

Step4 mark patterns in image as objects (about 1/2 of images), define those as training set

Step5 let network train itself

? if insuficient, back to Step1

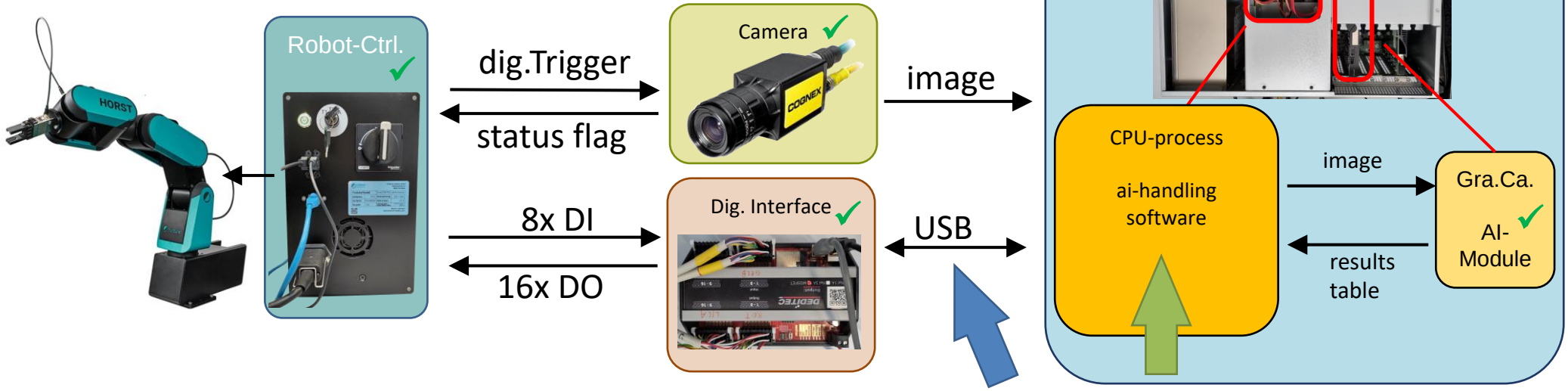
Step6 export ai module to file



dataflow between AI and robot

2 more tasks:

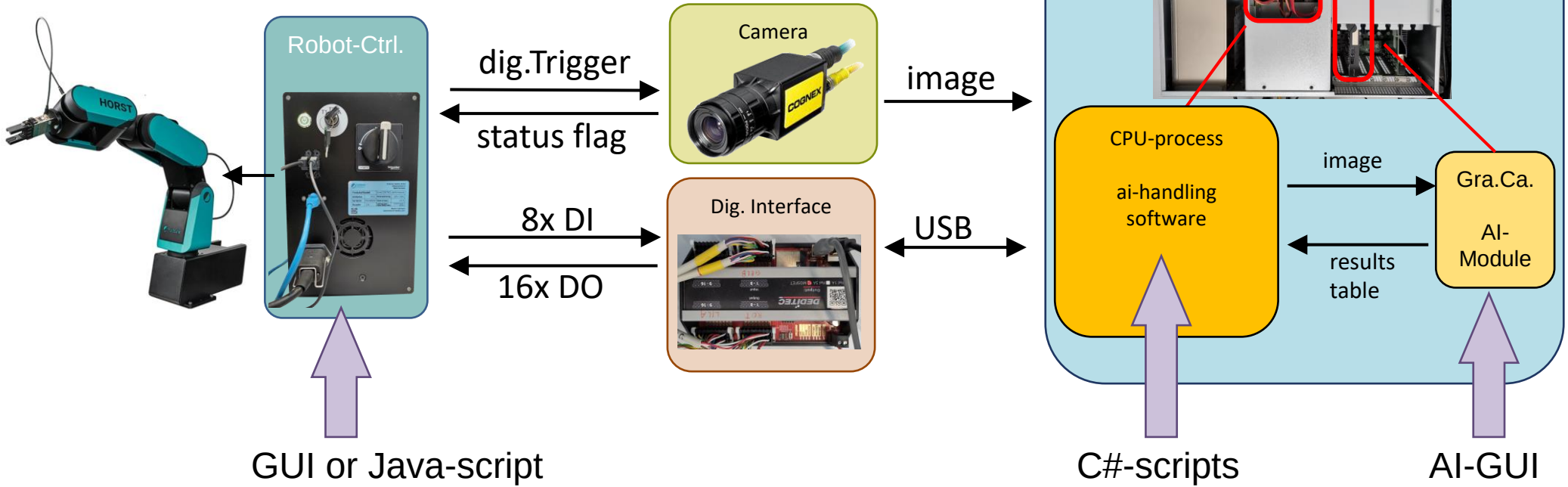
arrange data-flow and **handle ai-results**



dataflow between AI and robot

2 more tasks:

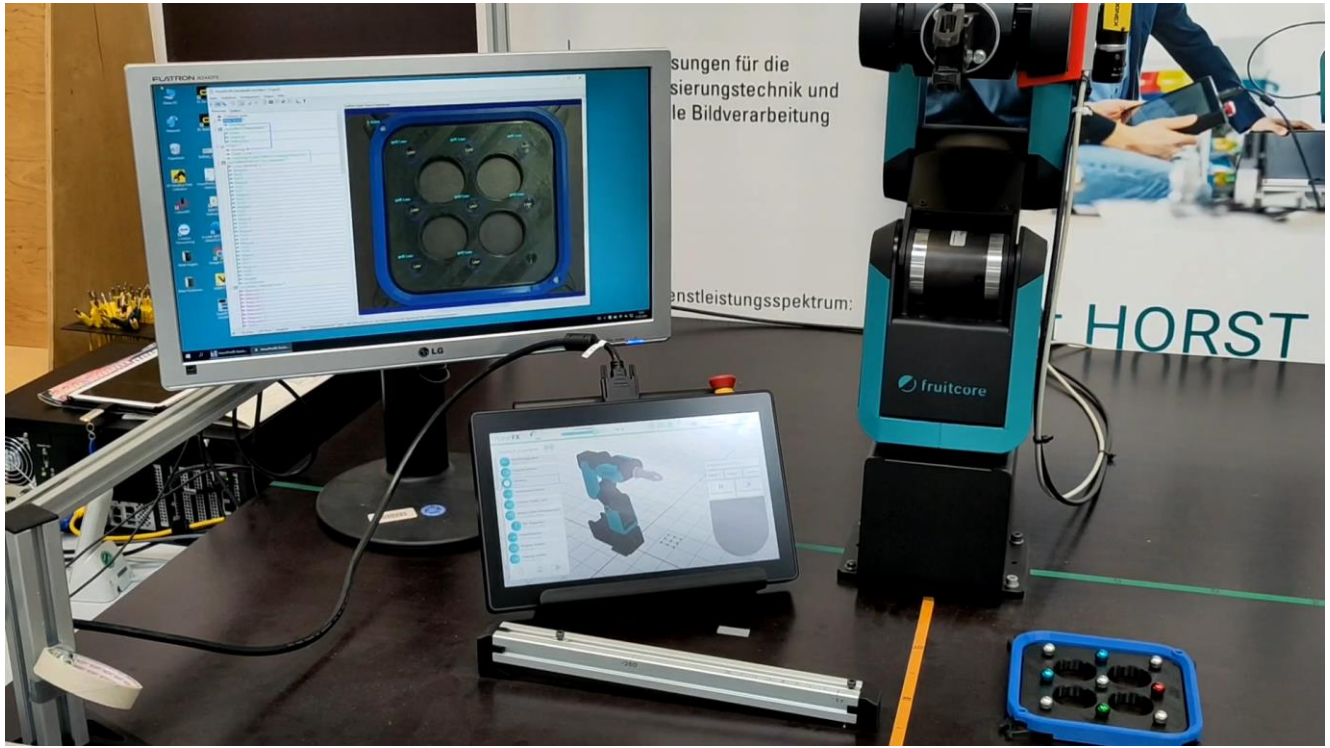
arrange data-flow and **handle ai-results**



AI enhanced robot

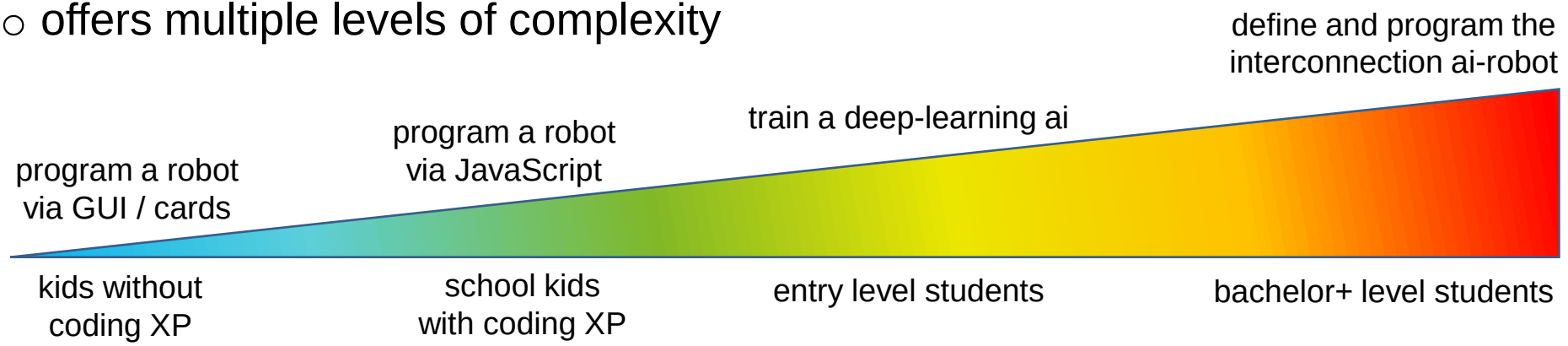
HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

run it all together



Summary

- installed an Integrated Learning Environment with industrial equipment
- emphasis in on application with close to real-life-application (no building)
- offers multiple levels of complexity



- having fun with the outcome