International Conference NEW PERSPECTIVES in SCIENCE EDUCATION

Novel Approach for Teaching AI in Entry Level Education

Samuel Kübler[1], Christian Bildhauer-Buggle[2], Thomas Schiepp[3]

Furtwangen Hochschule University, Faculty of Medical and Mechanical Engineering, Campus Schwenningen, Jakob-Kienzle-Straße 17, 78054 VS-Schwenningen, Germany[1, 2, 3]

Abstract

The proliferating fielding of artificial intelligence in various but especially industrial applications increases the need for pupils, students and employees in STEM areas and leads to a new job profile, the AI engineer, who does not necessarily require a full study in computer sciences. Extensive efforts are undergone to improve education at the K-12 level, but even at the university level educators are struggling to keep pace with the development of AI. At the same time, a continuous stream of presentations in digital media reveal an unbroken fascination with industrial robots, even though their emergence dates back decades. Our approach to stimulate students' interest in AI education leverages the appeal of physical interaction with an AI via a 6-DOF robot rather than an auditory or purely visual interface. In order to keep access to robot programming as a hurdle as low as possible, we expand the graphic programming ability of an industrial robot user interface with a self-designed game-card concept and combine this with an AI hard- and software that is either pre-trained or trained as part of a problem-solving exercise. We present a step-by-step approach for artificial intelligence in education (AIED) that is characterized by increasing complexity of the exercises and can therefore be adapted to offer different levels of interactive learning environments (ILE). Part of this step-by-step approach is also the presentation of the development of information processing from strictly linear (the robot control) via object-oriented (in the interaction of the AI with the robot) to the AI itself in an overall project that combines all stages and concepts. This interactive learning environment thus enables an adjustment in both the level of difficulty and - by selecting the sub-areas - the amount of time in teaching the application of AI in a real-world scenario.

Keywords: *Science, Technology, Engineering and Mathematics (STEM), Artificial intelligence in education (AIED), Interactive learning environments (ILE)*

1. Introduction

The demand for qualified AI professionals is high and steadily increasing. Yet particularly in Germany, industry has issues in recruitment of skilled personnel to staff mid-level technical positions already since years which is aggravated by the local demographic situation of an aging population [1]. Therefore, education in STEM subjects competes with other fields of study for a small pool of possible students. In order to increase the attractiveness of STEM studies and boost the popularity of AI in university level studies, new approaches for AI education in K-12 level are sought after already since the late 1980's [2]. The primary hurdle is to attract students and those interested in studying these subjects in the first place and many efforts are pointed in this direction [3], [4], [5], [6], [7], [8], [9], [10]. Given the high public interest in technological innovations, artificial intelligence and the popularity of robots among young people [11], we use this attractivity as focal points of our teaching strategies and try a hands-on-approach. This is not dissimilar to approaches using mobile robots [12], [13], [14], the main difference is our employment of a professional, industrial robot, which stresses the connection to industrial installations with robots and attempts to “solve real-world problems” [15] but with the focus on the application not on the engineering of a robot. A common AI application in an industrial context is the identification of faulty parts within a large set, such as those that are not of the correct colour or have visible defects, an area where weak-AI is dominated by machine vision (MV) and deep-learning (DL). By integrating such real-world applications, young people can understand not only the theoretical foundations but also the practical relevance and the impact of AI on future industry and production, summarized under the term industry 4.0. Such image recognition systems are often used in post-production quality control, in order to remove faulty parts from an assembly line, before they are packaged. For the removal, robots are not the first choice, since removed parts are to be discarded and the more cost-effective solutions are preferred for that purpose. With their great precision in pick-and-place applications, robots are ideal to collect components from trays or palettes on which such components are shipped, and place them into an assembly line. This application is so common that the programming interfaces of most if not all robot manufacturers offer a purpose-made function “depalletize” [16]. Such a function cycles the pick-up coordinates for a robot in a pick-and-place process through all available slots - typically in a 2D array - until the tray is empty, which removes the necessity to program 2 nested loops from the user. This scenario is only one out of many, and it does not require AI, if the programmer can assume, that all positions in the palette are occupied with good parts in the correct orientation. While this application is also in the field since decades, the manufacturers of robots strive to improve the interface to their robots such, that the necessary training requires ever less skill and time for the shop-floor personnel to create such simple applications without help from more qualified – and expensive - experts. Since robots operate mostly in shop-floor environments and spend by far most of their time without any interaction with a human operator, there is no need for desktop-like control computers. In order for the operator to observe a robot’s actions closely, the controller is a handheld device wired to the robot control electronics, which also contains safety related switches, which mandates cable connections. The improvements in display quality, size, resolution and the advent of large touch screens has allowed such robot control handheld units to develop from handheld teach pendants (HTP) with a single line, black-and-white LCD display and the look-and-feel of oversized pocket calculators via HPT’s with large, full-colour TFT screens to tablet teaching pendants (TTP) with touchscreens rivalling tablet PC’s and only a few physical buttons remaining for safety reasons. Sensors within the robot arms have allowed the rise of COBOTS and yet another method of robot programming, but those are not discussed in this paper. As the computing power of the processors within the units are also dramatically rising, more effort can and is being dedicated to improving ease of use via graphical representation of a robot’s functionality and programming with the abovementioned goal of ease-of-access for less skilled personnel. These modern operating concepts allow users to create program sequences through graphical elements without deep prior knowledge, similar to the program flow diagrams like in DIN 66001, but simpler. As a side effect, this makes such robot-GUIs also suitable for use in teaching exercises with pupils within K-12 education, as we intend to verify in our setup. Since a robot can physically execute only one command at a time with the synchronous motion of all 6 motors to move from one pose to another counted as one, this also suits the explanation of the workings of procedural programming. Here, program flows are handled linearly, which bears similarities to, for example, assembler programming. It has become a common improvement to a robot’s capabilities to outfit it with machine vision (MV) and AI, so the robot can perform less well-defined tasks as the abovementioned depalletizing within a fully defined environment. With the help of AI and MV, a robot can pick objects from an unsorted heap or a conveyor belt and arrange them into a palette, to “palletize” parts. To enable this functionality, software must organize the data flow between the MV, the AI and the robot, which is typically done with object-oriented text-based programming languages. Offering the creation of this software as a task to students poses a significantly higher challenge, since here prior knowledge is required. Compared to this, the teaching of the AI itself is easier, since the algorithms for teaching of an AI are hidden behind a convenient user interface. Also here, commercial providers strive to improve ease-of-use to their customers, so that less skilled personnel can perform this task with as little training effort as possible. Such interfaces offer all the comfort of standard desktop applications with menus, graphical items, intricate graphical representation and analysis of the AI’s performance. This level of ease-of-use again makes it possible to teach pupils, how to configure and teach an AI. In this paper, we present our newly set up integrated learning environment (ILE) which incorporates a 6-degree-of-freedom (6DOF) robot, machine vision (MV) with a deep-learning AI and all necessary hard- and software to handle data flow between them. Borrowing from the user-concept of the robots GUI, we introduce also a card-game of robot-programming. The idea is, to offer as much physical interaction as possible both on the input as well as on the output side.

2. Methodology

2.1 Setup

For the experimental setup, we have installed an industrial robot model Horst600 (Fruitcore Robotics GmbH, see fig. 1a [17]) with a graphical user interface (HorstFX, see fig. 1b). In order to replicate the GUI functionality along with its look-and-feel and make a cooperative working environment possible, we created laminated cards of about 30 cm length with a selection of robot commands and provide them in sufficient numbers. The purpose of this is to circumvent the limitation of the TTP's size of 30 by 20 cm which allows only one person at a time to use the panel. With the larger cards, it is possible to request groups of pupils to discuss cooperatively, which command-cards to choose and in which

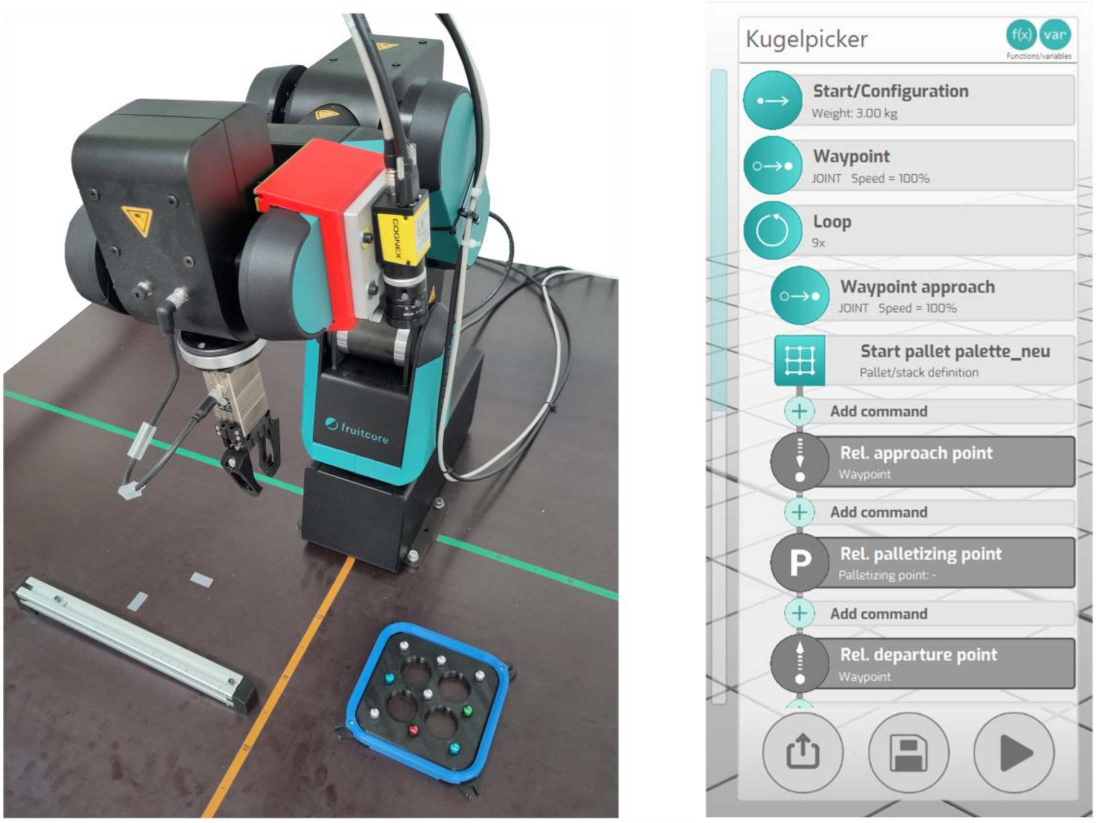

Fig. 1a (left): Setup with robot, camera, target palette with balls and repository (raised profile)
Fig. 1b (right): partial screenshot of a GUI program, notice the sequential structure of commands

order to arrange them, in order to plan the entire program before entering it on the TTP. This can result in a column of dozens of cards arranged on the floor with pupils walking around them, which is supposed to stimulate a cooperative working pattern for the group. As objects for a depalletizing task serve a sufficient number of metallic balls from a commercial provider (Gravitrax by Ravensburger) with 12 mm diameter in 4 different colours. As a palette containing these balls we engineered and 3D-printed a palette with 9 fixed positions in a 3x3 grid (spacing 50x50 mm). This palette is fixed to the base platform on which the robot rests as well (see on the right in fig. 1a). A pre-defined drop-off point is provided by a simple construction profile that is mounted on the base platform at a slight angle to make the balls roll down towards a mechanical stop (see on the left in fig. 1a); this rail can serve also as a fixed position to pick up balls when the lowest position of balls next to the mechanical stop is chosen. Here all balls are dropped off at roughly the mid-point of the rail. As an effector the robot is equipped with an electromechanical gripper, which is modified with a pair of spoon-like grippers shaped to ideally match the balls used here (see fig. 1a). To enable machine vision an industrial camera system is mounted at the robot's last arm before the effector flange (with the gripper). The robot model offered no means of fixation, so we engineered a 3D printed clamp-like base with an aluminium plate that serves as a heat sink for the camera. We did not choose a position of the

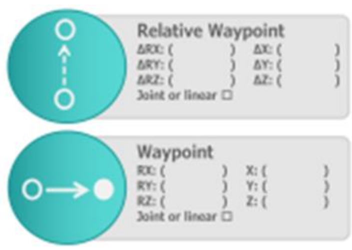

Fig. 2: Print-template for cards

camera next to the effector, since it's intense motion would stress the data cables whereas the standard imaging position is always the very one, shown in fig.1. To acquire an image for use in the AI, the robot must move into this pose and trigger the camera via a digital pulse from the DIO-interface of the robot control computer. The camera data connection is wired to an industrial computer (IPC-631, I-Mation GmbH [18]) with appropriate machine vision hard and software. This software passes the image to an AI module, that is running on an internal, dedicated, powerful graphics card and retrieves the information, reported back from that AI. The analysis of the reported data, essentially a list of discovered objects, their classification and their position within the image is performed by a series of C# scripts that execute on the normal CPU of the computer. These scripts can access a DIO (Digital Input/Output) interface, which is wired to the DIO hardware of the robot control hardware. By setting the TTL states, the system can thus convey information from the AI to the robot. The AI module is loaded into the graphics card dedicated to the AI, but must be pre-trained and provided as a local file. In order to train this AI module, we use a commercial software package (Cognex Deep Learning Studio, Cognex LLC [19]). This is done solely using image files from the hard-disk and is performed locally on the computer. The robot and the machine vision hardware at this stage is only needed to acquire a sufficient number of images beforehand, but not during the AI training. Theoretically, the AI training can be performed even on a separate computer, or multiple ones as a general AI training exercise in a PC equipped class room, with the image files and the trained AI being transferred by LAN or USB.

2.2 Procedural Program (for pupils in school education)

The entry-level for pupils the task is to create a procedural program with the graphical user interface of the robot to depalletize all balls from the palette to the drop-off point. To provide some background knowledge, a brief and concise introduction with a duration of about 15 minutes to the basic concepts of robot operation is offered in a classical lecture like format. The basics deal with the role of robotics in industry and society, robot electromechanics, different movement possibilities, coordinate systems, orientation of the effector in 3D-space and the basic structure of the robot GUI. We demonstrate these concepts by allowing the students to use the direct control interface to try out motions of the robot in the direct drive mode. Here, each student can move and rotate the effector while observing the GUI displaying position and orientation of the effector. This inevitably leads occasionally to collisions of the gripper with the platform, but the 3D-printed parts have weak-points engineered into them, which protect the rest of the hardware; eventually, we replace the broken piece in about a minute. Then, the task is explained and the cards from fig.2 are handed out. The lecturer keeps in the background during the discussion and allows the students to find the right order of instructions by themselves. The entire sequence of commands is laid out on the floor, sometimes over a distance of about 3 meters. A crucial detail introduced in this phase is the implementation of a loop, which must automate the traversal of the nine fields of the palette with a counter. This loop allows the process of moving and gripping to be repeated, with the target point being adjusted with each pass. This cycling through is provided by the HortsFX GUI through the "pallet" function. This representation allows students to visualize and understand the sequence of commands on an algorithm level and how individual steps are sequentially processed to achieve the set goal. The developed algorithm is transferred to the robot interface's control, once the teacher deems it appropriate. This can still take some time, since all positions of the effector must the taught one by one. If successful, program execution will steer the robot to depalletize all balls correctly, offering the pupils an element of suspense and joy. Also, an interesting detail can then be demonstrated: in this operation mode, the robot will execute the pick-up and drop-off sequence even if no ball is located at the pick-up point, or any other mishap like a collision happens.

2.3 AI-assisted (for entry level students)

The task in this level is to repeat the task to depalletize as described above, but now the robot should skip over empty slots and avoid "bad" parts. In our exercise environment we introduce the distinction of "good" vs. "bad" part through the colour of the balls, arbitrarily choosing one as "good" and all others as "bad". This builds on the program flow created before and that program can be expanded upon. Firstly, the camera must take an image of the palette, so the robot must move the camera into the correct position and trigger the image acquisition. The camera transfers the image automatically to the machine-vision PC, which triggers also the C# scripts mentioned above. However, this might require a second to start. The scripts will set output flags in the DIO, to signal the robot, that processing is underway, and reset the flag, once the processing is complete. The robot must wait a second unconditionally for processing to start, and then continue to wait until processing is completed, i.e. the according flag is reset. Otherwise, the robot might query the DIO outputs before they are correctly set.

As mentioned before, the AI module is executed within a dedicated graphics card. This module is either pre-trained by the supervisor or trained by students as explained in chapter 2.4. It is noteworthy that the AI module only identifies objects and returns this information in an ASCII data table. The decision, which slot in the palette should be visited or ignored by the gripper has to be programmed in the C# script that analyses this table. The identification of each object is done via a string of letters, that the user defines while training the AI, so the scripts essentially does an ASCII comparison. Part of this task is also to identify the position of the balls at least by a coarse estimate sufficient to assign a colour to a certain tray slot or identify an empty one. The last change in the robot program concerns the activation of the gripper; the robot must query the inputs of the robot DIO interface, which convey the result of the AI analysis. We implement this by using 9 digital lines for each of the 9 slots in the palette. The graphical programming method does not offer a simple way to skip over a slot to be visited, but the decision of closing or not closing the gripper can be easily linked to the DIO lines. The creation of the object-oriented C#-scripts that handle the data-flow between the camera, the DIO-interface and the AI can also serve as a programming exercise itself. This represents a challenge in coding and requires some skills in software development and data representation. Therefore, we provide this code ready-made for pupils in the groups 1 and 2; it is however an appropriate task for students at university level.

2.4 Training the AI (for advanced students / bachelor candidates)

As mentioned above, we choose the colour of the balls as the distinction with arbitrarily one being defined as “good” i.e. to be picked up. Finding all “good” ones is the key task of the AI-module. It is not strictly necessary to identify also all other colours and empty slots, but we pose the task with this requirement anyway, because then it is easy to switch from one “good” colour to another one within the scripts handling the AI output. On discovering one or more objects (an empty slot is to be considered an object as well) the AI returns an ASCII-table of those objects and their position within the pixels of the image. This AI module must be trained prior to using it in conjunction with the robot. This task is executed via a commercial GUI that is primarily dedicated to machine vision purposes and offers extensive functionality and a convenient interface for deep learning. The modules of the DL application offer a larger number of AI functions; however, for the task described herein, only the object recognition feature is required. The task for the students next to getting a grip on the interface and training the AI, is to collect a sufficiently large set of images of the target palette with the installed camera system featuring a sufficient diversity in the settings. These images should vary in exposition time, to emulate different light conditions, experience actually changed lighting the room to modify reflections and intentionally slightly misalignment of the camera. These images can be taken independently from the DL-GUI and be stored on a hard drive for later use. As the DL-GUI runs independently with these stored images, all training can be conducted by students independently of the robot. Once the AI module performs satisfactory within the GUI, it can be exported into a separate file, which can be activated and used independently from the DL-GUI. Once the AI module is activated and works cooperatively with the robot, the actual experiment can be run, i.e. a few - up to 9 - balls of different colours, can be placed on the palette and the robot program can be launched. If the task is correctly solved, the robot will depalletize only the balls with the chosen colour, which again adds an element of suspense and enjoyment to the students, if the robot operates correctly.

3. Student’s response

We envision 3 levels of students, out of which we gained experience with 2, where both are from secondary level school education. One group of students is comprised of about 13 years old, the other of about 17 years old school pupils, the third is comprised of bachelor level students. At the time of the editing of this paper, we gained experience only with the first 2 groups, but we are currently planning to work with experienced students in the next semester and further expand our experience with larger groups. Both managed to solve the card game preliminary exercise and program the robot to successfully depalletize the pallet and solve the task without AI assistance within less than 45 minutes. We retrieved feedback from these pupils by handing out questionnaires that were returned with about 2 months delay. The replies indicated great satisfaction and “fun” with the exercise and confirmed our hope that the interaction with the robot is more joyful than with an internet-based text AI. To our surprise, the pupils requested even more advanced explanations in the initial lecture part, seemingly, we have underestimated the pupils’ skills in computing concepts. It remains to be seen, if this holds for all groups, since the total number of subjects in our teaching experiment is below 30, and more statistics is desirable.

4. Conclusions and outlook

We present an ILE that combines an industrial robot and an AI with a game-card-like teaching approach to robot programming. Using a physical interaction rather than an internet-based text or speech-based AI greatly enhances the interest and 'fun' factor in the lecture. Furthermore, our approach is recreating one of the most common real-life industrial application. On completing the exercise, students have acquired basic skills in programming a robot, and have understood, how an AI is integrated with an industrial setup. Our approach has received very positive feedback and serves as a basis for further expansion of this program. The course will be expanded into a 1 semester course for introduction to AI in industrial applications on bachelor level, where the herein described setup will expand a traditional lecture with a real-life-exercise.

References

[1] Niggemann, M. "Künstliche Intelligenz in Produktion und Maschinenbau", VDE Verlag, 2022.

[2] Brady, M. „Artificial intelligence and Robotics," *Artificial Intelligence,* p. 79, 1985.

[3] Kandlhofer, M. et al. „Artificial Intelligence and Computer Science in Education: From Kindergarten to University," in *IEEE Frontiers in Education Conference (FIE)*, 2016, p. 16.

[4] Dodds, Z. et al. "Components, curriculum, and community: robots and robotics in undergraduate AI education," *AI Magazine,* p. 11, 2006.

[5] Roll, I. et al. „Evolution and Revolution in Artificial Intelligence in Education," *Int.J.Artif.Intell.Educ.,* p. 582, 2016.

[6] J.E.Aoun, J.E. "Robot-proof: higher education in the age of artificial intelligence," Cambridge: MIT Press, 2017.

[7] Chassignol, M. et al. „Artificial Intelligence trends in education: a narrative overview," *Procedia Computer Science,* p. 16, 2018.

[8] Chen, L. et al. „Artificial Intelligence in Education: A Review," *IEEE Access,* p. 75264, 2020.

[9] Huang, J. et al. „A Review on Artificial Intelligence in Education," *Academic Journal of Interdisciplinary Studies,* p. 206, 2021.

[10] Tedre, M. et al. „Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education," *IEEE Access,* p. 110558, 2021.

[11] Grubbs, M. „Robotics intrigue middle school students and build STEM skills," *Technology and Engineering Teacher,* p. 12, 2013.

[12] „Robocup," [Online]. Available: http://www.robocup.org/. [Zugriff am 2023].

[13] R.Raj, R. et al. „A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives," *Applied Sciences (MDPI),* p. 6951, 2021.

[14] Chu, S.T. et al. „Artificial intelligence-based robots in education: A systematic review of selected SSCI publications," *Computers and Education: Artificial Intelligence,* p. 100091, 2022.

[15] Khanlari, A. „Robotics integration to create an authentic learning environment in engineering education," *IEEE Frontiers in Education Conference,* p. 1, 2016 .

[16] Shake, M. „Ten popular industrial robot applications," Jabil, 2023. [Online]. Available: https://www.jabil.com/blog/ten-popular-industrial-robot-applications.html.

[17] FruitCore Robotics, [Online]. Available: https://www.fruitcore-robotics.com/en/. [Zugriff am 2024].

[18] I-Mation, [Online]. Available: https://www.i-mation.de/. [Zugriff am 2024].

[19] Cognex, [Online]. Available: https://www.cognex.com/. [Zugriff am 2024].

[20] Raj, M. et al. „Primer on artificial intelligence and robotics," *Journal of Organization Design,* p. 8, 2019.

[21] Asemi, A. et al. „Intelligent libraries: a review on expert systems, artificial intelligence, and robot," *Library Hi Tech,* p. 412, 2020.

[22] Ahmat, S.F. et al. „Artificial Intelligence and Its Role in Education," *Sustainability (MDPI),* p. 12902, 2021.