machines

MDPI

Article

Hyper CLS-Data-Based Robotic Interface and Its Application to Intelligent Peg-in-Hole Task Robot Incorporating a CNN Model for Defect Detection

Fusaomi Nagata [1,*], Ryoma Abe [1], Shingo Sakata [1], Keigo Watanabe [2] and Maki K. Habib [3]

[1] Graduate School of Engineering, Sanyo-Onoda City University, Sanyo-Onoda 756-0884, Japan; abekendou@icloud.com (R.A.); f124607@ed.socu.ac.jp (S.S.)
[2] Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan; watanabe-kei@okayama-u.ac.jp
[3] Mechanical Engineering Department, SSE, The American University in Cairo, New Cairo 11835, Egypt; maki@aucegypt.edu
* Correspondence: nagata@rs.socu.ac.jp

Abstract: Various types of numerical control (NC) machine tools can be standardly operated and controlled based on NC data that can be easily generated using widespread CAD/CAM systems. On the other hand, the operation environments of industrial robots still depend on conventional teaching and playback systems provided by the makers, so it seems that they have not been standardized and unified like NC machine tools yet. Additionally, robotic functional extensions, e.g., the easy implementation of a machine learning model, such as a convolutional neural network (CNN), a visual feedback controller, cooperative control for multiple robots, and so on, has not been sufficiently realized yet. In this paper, a hyper cutter location source (HCLS)-data-based robotic interface is proposed to cope with the issues. Due to the HCLS-data-based robot interface, the robotic control sequence can be visually and unifiedly described as NC codes. In addition, a VGG19-based CNN model for defect detection, whose classification accuracy is over 99% and average time for forward calculation is 70 ms, can be systematically incorporated into a robotic control application that handles multiple robots. The effectiveness and validity of the proposed system are demonstrated through a cooperative pick and place task using three small-sized industrial robot MG400s and a peg-in-hole task while checking undesirable defects in workpieces with a CNN model without using any programmable logic controller (PLC). The specifications of the PC used for the experiments are CPU: Intel(R) Core(TM) i9-10850K CPU 3.60 GHz, GPU: NVIDIA GeForce RTX 3090, Main memory: 64 GB.

Keywords: industrial robot; HCLS data; sequence control; cooperative control; without PLC; peg-in-hole task; misalignment; convolutional neural network; defect detection

check for updates

Citation: Nagata, F.; Abe, R.; Sakata, S.; Watanabe, K.; Habib, M.K. Hyper CLS-Data-Based Robotic Interface and Its Application to Intelligent Peg-in-Hole Task Robot Incorporating a CNN Model for Defect Detection. *Machines* **2024**, *12*, 757. https://doi.org/10.3390/machines12110757

Academic Editor: Dan Zhang

Received: 2 September 2024
Revised: 7 October 2024
Accepted: 24 October 2024
Published: 26 October 2024

1. Introduction

Articulated-type and selective compliance assembly robot arm (SCARA)-type industrial robots have been introduced in automation lines for manufacturing and assembling industrial products. Recently, there has been an increasing need for multiple small-sized robots to dexterously automate such manufacturing, assembling, or working processes that skilled workers have manually dealt with. For example, Sen et al. proposed a robot that could cope with a peg-in-hole task with only a small clearance by detecting the contact state between a workpiece and hole with their originally developed elastic displacement sensor and by adjusting the robot's orientation [1]. Moreover, Piotr and Krzysztof proposed the modular, didactic production system called MDSP, which was equipped with miniatures of typical devices commonly used in automation lines so that the characteristics of manufacturing, storage, transport and assembly subsystems, and quality control could be actually simulated [2]. When robots are used in manufacturing lines, teaching data are generally created by off-line robot programming using a teaching pendant. Still, it requires a language specific to the robot manufacturer, leading to the lack of versatility.

On the other hand, various types of numerical control (NC) machine tools can be standardly operated and controlled based on numerical control data called NC data that can be easily generated using computer aided design (CAD)/computer aided manufacturing (CAM) systems. In the general process to produce NC data, the main processor for CAM generates cutter location source (CLS) data as intermediate data, then the post processor for CAM transforms CLS data to target NC data according to various types of NC machine tools. The technical vocabulary of CLS data is defined in ISO 2806:1994(en). On the other hand, operation environments of widely used industrial robots still depend on the conventional teaching and playback method, in which each maker's original robot language is uniquely used. Traditional robot programming by teaching is still time consuming and not cost-effective. However, it seems that the operation environment of industrial robots has not been well standardized and unified yet like NC machine tools controlled based on common NC data precisely generated by CAD/CAM systems.

As for the integration of an industrial robot with a CAD or CAD/CAM system to reduce the load of the robotic teaching task, several results utilizing CAD systems have been reported. Zeng et al. proposed an automatic trajectory planning based on CLS data to cope with the problem of obtaining trajectory in robotic finishing with complex surfaces in the manufacturing industry, in which KUKA robot path programs are generated from CLS data [3]. Amersdorfer and Meurer proposed a method to generate equidistant coverage paths on a free-formed curved surface, in which a transformation is achieved between 3D Cartesian coordinates and 2D parameters based on the arc-lengths along the surface. In the transformation, the shape is generated by a CAD/CAM or a 3D scan and is directly processed for robotic machining [4]. Molotla et al. proposed a hybrid integral system that can perform subtractive and additive manufacturing by 3D printing using robotic arms [5], in which it is reported that CAD/CAM design is essential for manufacturing desired workpieces. Furthermore, Ma et al. proposed an actual welding path generation method from a CAD model, in which point clouds sampled from a CAD model of a workpiece and those captured by vision sensors are reconfigured for generation [6]. However, up to now, there does not seem to be a unified robotic data interface based on CLS data.

Moreover, as for recent functional extensions of industrial robots, many research results can be found. For example, Su et al. presented a robotic visual-based controller with a time-delay compensator to deal with the problem of inserting a moving peg [7]. The time-delay compensator was configured based on a radial basis function neural network, and a feedback compensator was incorporated for eliminating the tracking errors caused by the time delays. Park et al. developed a robotic bin picking system with cluttered objects using human guidance and convolutional neural network (CNN) [8], in which a robot bin picking system that uses a CNN model trained by human data is proposed. The robotic system increases the accuracy of the model by continuous training by the robot itself.

In the realm of defect detection, advancements such as the integration of singular spectrum analysis (SSA) with hierarchical hyper-Laplacian prior prototypes (HHLP) have demonstrated efficient flaw extraction methods in industrial robot operations, leading to enhanced fault diagnosis and system performance monitoring [9]. Likewise, dual-metric neural networks with attention guidance have shown remarkable success in surface defect detection, mainly when limited training data are available, making such systems highly applicable in real-world industrial settings [10]. Choi et al. proposed a multimodal 1D CNN that can predict delamination defects from time series multi-sensor data [11], in which it is reported that the predictive accuracy and real-time operational capability, i.e., inference time, are superior to the previously proposed methods in industrial settings. Wang et al. proposed a surface roughness prediction method based on deep learning for a large shaft grinding robot, in which three types of signals, including spindle current, vibration, and acoustic emission, were extracted for inputs [12]. The multiple process signals are fused through an attentional CNN-based long short-term memory (LSTM) architecture.

Programmable logic controllers called PLCs have played an important role in enhancing the performance of industrial robots, mechanical systems, and automation lines, e.g., for sequence control and cooperative control. For example, Jacob and Felipe identified the lack of a modernized version targeting the functionality of an Industry 4.0-oriented control system as a problem and proposed a lot-PLC with control functions, fog-computing as filtering and field data storage, and multiple wireless interfaces managed independently. This allowed them to conduct loop reconfiguration without restarting the controller, thus increasing flexibility [13]. In addition, Ossama et al. identified a problem where PLC programs used in large applications in the food, oil, and gas industries make troubleshooting very difficult and require very competent engineers. To solve this problem, machine learning technology and add on instruction (AOI) were applied to the PLC program to create an easier structure, reduce the number of rungs, reduce the processor memory, and speed up execution time, in contrast to conventional ladder diagrams [14].

Moreover, as for the cooperative control of multiple industrial robots, Edgar et al. presented a mechatronics system to couple two industrial robots to a parallel kinematic system to temporarily increase the mechanical stiffness property. It is reported that the coupled two robots enable load sharing, higher process forces, and eventually, higher precision [15]. Fathi et al. systematically reviewed assembly-line balancing studies targeted at assembly lines with industrial and collaborative robots [16], in which it is observed that balancing assembly lines with collaborative robots has been receiving more attention in the last five years with the emergence of Industry 4.0. For example, if a robot is not able to execute a wide working range task since the position of the target workpiece is out of the movable area, then cooperation by multiple robots is effective in coping with the task. Up to now, open architecture types of industrial robots (OAIRs) have been developed and released to product manufacturers so that the developments of original robotic applications become possible on the user side [17]. Generally, OAIRs are provides with each maker's software development kit (SDK), such as dynamic link library (DLL); however, it seems that the available number of connections between a PC and robot is limited to only one owing to the specification of the provided SDK. Hence, some kind of PLC device is needed to implement sequence control or cooperative control of multiple robots while taking each motion timing. For an actual implementation, it is forced to the user side to apply some sequential control or cooperative control to multiple robots.

Collaborative robots (COBOTs) have also seen substantial improvements, with modern control technologies integrating artificial intelligence (AI) and machine learning, enabling more flexible and efficient cooperation between robots and humans in various industrial contexts [18]. Moreover, recent surveys on industrial defect detection using computer vision and machine learning techniques have highlighted the growing importance of automated quality control [19]. In contrast, our approach introduces a unified data interface that integrates cooperative control, defect detection using a CNN model, and automatic teaching point generation, systematically described through HCLS data. This resolves many of the limitations found in previous systems and provides an efficient, scalable solution for complex industrial tasks.

As the authors have surveyed, it seems that an easy implementation and operation environment for a machine learning model such as CNN, visual feedback controller, and cooperative controller for multiple robots have not been realized for industrial robots in an integrated manner yet. In this paper, a hyper cutter location source (HCLS)-data-based robotic interface is proposed to cope with the above issues without using any PLC. Due to the HCLS-data-based robot interface, the robotic operation sequence can be visually and unifiedly described as NC codes. In addition, for example, visual feedback control for object tracking, CNN model for defect detection, and cooperative control for multiple robots can be systematically described in HCLS data and executed. The effectiveness and validity of the proposed system are demonstrated through a cooperative pick and place task using three small-sized industrial robot MG400s and a peg-in-hole task while checking undesirable defects in workpieces by a CNN model.

While our previous work in Abe et al., 2023 [20], introduced the core functionality of the HCLS data interface for sequence control of multiple robots without a PLC, this manuscript presents significant extensions and new scientific contributions. The novel aspects of this work include the following:

1. Implementing cooperative control for multiple robots performing large-scale workpiece handling tasks, expanding the application of the HCLS interface.
2. Integration of a neural-network-based defect detection system (CNN) for peg-in-hole tasks, leveraging ONNX models converted from MATLAB and running on Python. This builds on recent developments in defect detection techniques, including singular spectrum analysis (SSA) and AI-driven defect recognition methods [9,10].
3. Introducing an automatic teaching point generator resolved misalignment issues between robot and work coordinate systems, thus improving efficiency and accuracy in teaching tasks.
4. Demonstrating these capabilities in an industrial setting highlights the proposed system's scalability, cost-effectiveness, and real-world application.

These contributions advance the field of collaborative robotics by enhancing the functionality, scalability, and practical applicability of industrial robot systems. They address current challenges in cooperative control, defect detection, and automated system design.

The remainder of this paper is organized as follows. In Section 2, we describe the design and implementation of the HCLS data interface and its integration with cooperative robot systems. Section 3 details the methodology for developing the automatic teaching point generator and integrating the defect detection system. Section 4 discusses the experimental setup and presents the results of the cooperative pick-and-place task and the peg-in-hole task with defect detection. Finally, Section 5 concludes the paper by summarizing the essential findings and outlining potential directions for future research.

2. Hyper CLS-Data-Based Robotic Interface

2.1. Robot Arm MG400

Major industrial robot manufacturers provide a variety of small-sized industrial robots. However, it is not easy for small- and medium-sized manufacturers in terms of introduction cost to introduce multiple robot systems to meet the need for the automation of segmented work processes. To cope with this problem, the authors decided to employ low-cost 4-DOF robot arms. The robot is Dobot MG400 (from now on referred to as MG400) provided by Shenzhen Yuejiang Technology Co., Ltd. (Shenzhen, China) [21]. It is equipped with a suction cup or an electromagnetic gripper as an end-effector. This robot has only 4-DOFs, less than the 6-DOFs generally required for industrial robots. However, this is not a problem when the robot is used only for tasks in which controlling the roll and pitch angles of the end-effector is not needed. In addition, the robot manufacturer provides an easy-to-use teaching interface [22], by which position data for the end-effector can be manually obtained. However, a CAD/CAM interface like for NC machine tools is not provided. Furthermore, it seems that visual feedback control for object tracking, cooperative control for multiple robots, and the CNN model for defect detection are not supported yet. The main objective of this study is to give MG400s such functionalities based on HCLS data.

2.2. Proposal of Hyper CLS Data Interface

The authors have introduced the concept of an HCLS-data-based robotic interface, in which not only conventional teaching points but also advanced statements for a camera control, convolutional neural network called CNN, visual feedback control [23], and sequence control of multiple robots [20] can be described. The concept of HCLS data allows conventional teaching and playback-type robots to enhance their practical potential ability.

The software for robot control based on HCLS data is developed on Python, which allows users to control the robot through the dialogue application shown in Figure 1. For example, an HCLS data file is loaded by clicking "(1) HCLS data load"; an HCLS data file loaded is executed by clicking "(2) Execute"; a CNN model saved with the open neural network exchange (ONNX) format is loaded for defect detection by clicking "(3) CNN (ONNX model) load"; the availability of the camera for visual feedback control is checked by clicking "(5) Camera check"; a YOLOv2 model saved in the ONNX format is loaded for defect detection by clicking "(7) YOLOv2 (ONNX model) load"; 10×10 positions and the corresponding 100 GOTO statements are automatically generated by clicking "(6) Position(XY) for MG400" or "(12) Position(XYZ) for MG400" after only giving four corner positions.

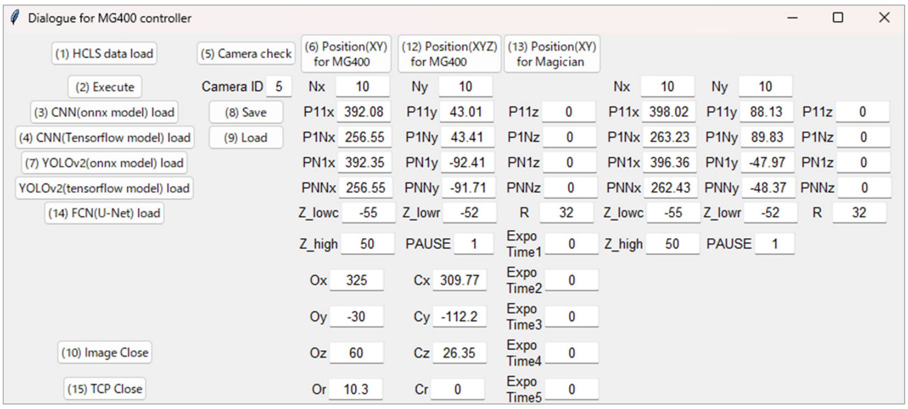

Figure 1. Developed dialogue application running on Python to cooperatively control multiple small-sized robot arms based on HCLS data, in which the position generation function considering the misalignment of robot and work coordinate systems is essentially incorporated.

In the case of MG400, essential target trajectories of the end-effector, such as for pick-and-place tasks, can be created using the teaching software named DobotStudio provided by the robot maker. When DobotStudio is used for a teaching task for the MG400, the taught data are not exported as a single playback file but as seven related files. Although these files (.lua) are composed of text codes and can be modified with a simple text editor such as Notepad, it is not easy for users to understand the coordinate values and their meanings intuitively.

To deal with this problem, the data converter implemented by the authors can systematically generate the corresponding HCLS data file from two of seven data files, as shown in Figure 2. One is src0.lua containing interpolation instructions. The other is point.json.lua containing teaching points. Although HCLS data basically have a structure based on the CLS data generally used in CAD/CAM systems, the superiority is its extendability of function. Table 1 shows the proposed HCLS statements currently available for the MG400 manipulator. For example, each statement from SNAPSHOT to CNN_DEFECT, SNAPSHOT_DIFF, and WEIGHT_CHECK provides us unique benefits that are not supported by NC codes.

Table 1. Examples of the proposed HCLS statements currently available for the MG400 manipulator.

Absolute	Header of HCLS
GOTO/x,y,z,r,0.0,0,1	Go to a position
SNAPSHOT	Take a photo using a camera
ORIENTATION	Pose estimation by image processing
CNN_ORIENTATION	Pose estimation using CNN
VF_CONTROL	Visual feedback control
CNN_DEFECT	Defect detection or feature extraction using CNN
GRIPPER_DISABLE	Gripper power off
GRIPPER_OPEN	Gripper open
GRIPPER_CLOSE	Gripper close
PAUSE	Wait a pause time as 'PAUSE 2'

Table 1. *Cont.*

MOVZ	Z-directional motion as 'MOVZ 10'
SET	Sent a number to a MG400 as 'SET 10'
GET	Obtain a number from a MG400 as 'GET 20'
SPEED	Set motion rate [%] as 'SPEED 50'
OFFSET	Camera offset in x-direction
GRIPPER_SUCK	Sucking using a suction cup
GRIPPER_BLOW	Blowing using a suction cup
R	Tool, e.g., gripper rotation angle [degree] as 'R 45'
DO 12 1	Set DO port No. 12 to 1 (ON)
DO 5 0	Set DO port No. 5 to 0 (OFF)
GRIPPER_OPEN	Electromagnetic gripper open
GRIPPER_CLOSE	Electromagnetic gripper close
LOOP_START 3	Repeating to LOOP_END for 3 times
LOOP_END	End position of LOOP_START
SNAPSHOT_DIFF 3000 2	To next statement in 2 s after 3000 pixels area is changed
WEIGHT_CHECK	Monitoring of current torques of 4 joints

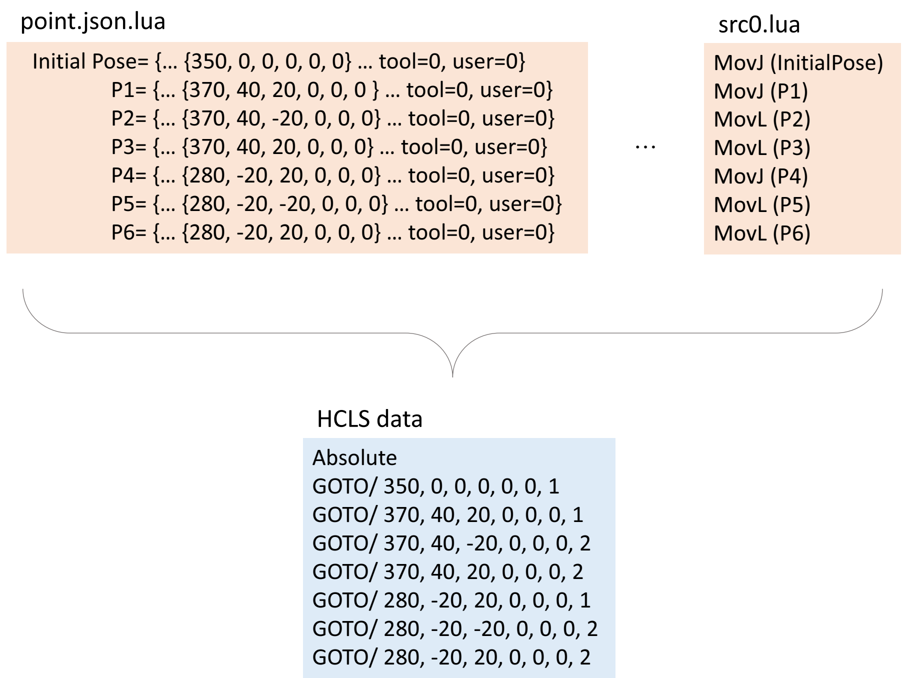

Figure 2. Example of an HCLS data file converted from original seven playback files.

2.3. Handling of Multiple MG400s Using TCP/IP

This study has been conducted to cope with the needs of an industrial product manufacturer in Japan who wants to apply multiple MG400s to a production line for automation. The problems to be identified have been investigated, and the solutions concerning coordinate misalignment and multiple robots control have been considered. The standard user interface DobotStudio provides several functions such as teaching playback, script language editing, and language editing with diagrams named DobotBlockly. In addition, a dynamic link library (DLL) is optionally provided for application developments on the user side. However, even when using DobotBlockly and DLL, only one robot can be operated from one application; therefore, sequential or cooperative operation by multiple robots cannot be directly achieved.

On the other hand, the proposed HCLS data interface easily realizes the cooperative pick and place task, as shown in Figure 3, by using remote control commands through TCP/IP protocol [24], for example, only by giving the HCLS data listed below.

⋮

```
TCP1_29999_OPEN:192.168.2.7
TCP1_30003_OPEN:192.168.2.7
TCP2_29999_OPEN:192.168.2.8
TCP2_30003_OPEN:192.168.2.8
TCP3_29999_OPEN:192.168.2.9
TCP3_30003_OPEN:192.168.2.9
TCP1_SV_ON
TCP2_SV_ON
TCP3_SV_ON
SNAPSHOT_DIFF 5000 2
TCP1_GOTO/308.0,-20.47,-10.0,22.74,1,1
TCP2_GOTO/308.0,-20.47,-10.0,22.74,1,1
TCP3_GOTO/308.0,-20.47,-10.0,22.74,1,1
```

⋮

where SNAPSHOT_DIFF is the statement to take a timing of the three robots. In this example, after detecting the changes in values over 5000 pixels in an image captured by the camera attached to the head of the second robot, if the situation continues for 2 s, the sequence moves to the next statement as "TCP1_GOTO" for synchronous control.

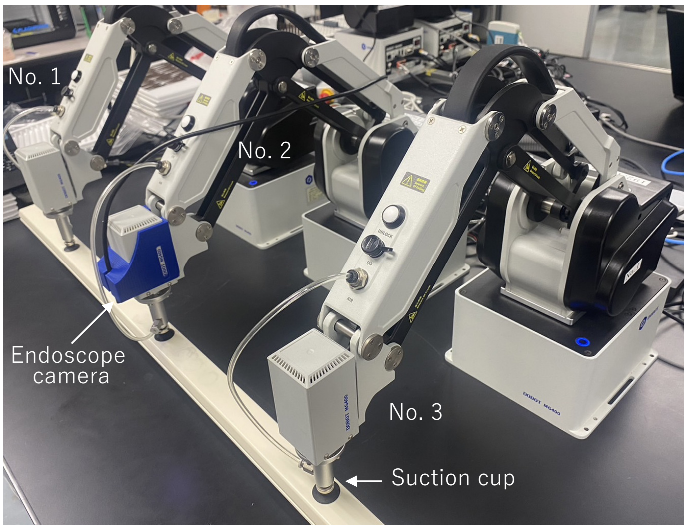

Figure 3. Cooperative control by three 4-DOF robot arms named MG400 for picking a long-size workpiece, which can be operated based on HCLS data by the developed application shown in Figure 1. The second robot has an endoscope camera to detect a long-size workpiece.

2.4. Automatic Generation of Teaching Points Considering Misalignment Between Robot and Work Coordinate Systems

When an industrial robot tries to automate a peg-in-hole task into many holes, the teaching task is much more complicated. Furthermore, a misalignment between the robot and work coordinate system as shown in Figure 4 cannot be ignored. Such a misalignment unfortunately tends to cause serious troubles such as breakage of workpieces. However, empirically, it seems that such a misalignment problem sometimes occurs by over- and under-fastening the screws and bolts used to fix jigs on a working table and tends to cause serious troubles such as breakage of workpieces. It also tends to occur more frequently as the end-effector moves around the work table's edge. This undesirable phenomenon is caused by the relative position and orientation misalignments between the two coordinate systems.

To solve the problem, the authors have proposed a generation function considering the misalignment that automatically generates teaching points for picking and placing by only giving the actual four corner positions of a working table. The function is available on the application developed in Python, as shown in Figure 1.

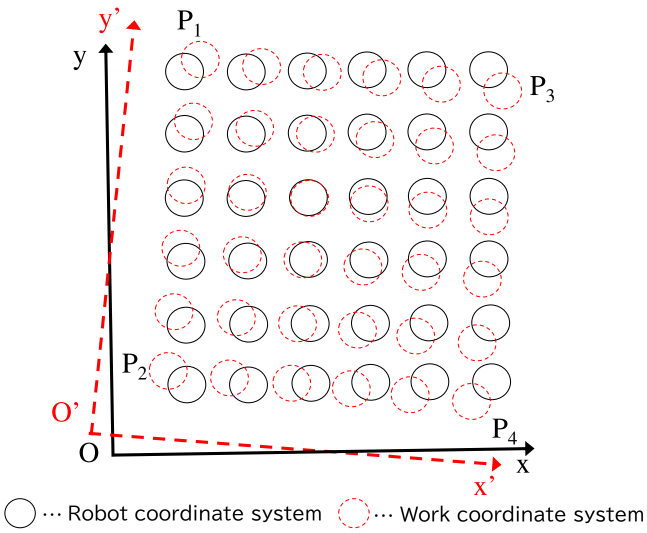

Figure 4. Image to explain undesirable misalignment between a robot and work coordinate systems in x and y dimensions. Actually, z direction must be noted.

In an example of the peg-in-hole task shown in the next section, which handles many small resin-molded articles, the clearance between an article and a hole on a jig side is about 1.0 mm. Therefore, picking and placing cannot be performed normally if the robot coordinate system and work coordinate system are misaligned, as shown in Figure 4. The proposed method can deal with the problem by setting only four corner positions.

3. Robotic Peg-in-Hole Task Cooperatively Using Two MG400s

Figure 5 shows the overall configuration framework of the experimental system, in which any PLC is not needed. The endoscope camera attached to the MG400-2 is used for object detection in Figure 3. Furthermore, the Basler GigE camera is used for the defect detection experiment in Section 3. Figure 6 shows an experiment scene of a peg-in-hole task using two MG400s to evaluate the effectiveness of the proposed system, in which two MG400s named No. 1 and No. 2 are cooperatively used. One hundred prototypes of cylindrical resin-molded articles with the length: 44 mm, the diameter of insertion part: 8 mm, and the diameter of grasping part: 7 mm were made using a 3D printer. On the jig side, two work tables with 100 holes are prepared and named work Tables 1 and 2. The diameter of each hole is 9 mm and the adjacent holes are located at an interval of 15 mm. In addition, a small jig with three holes of 9 mm diameter is placed between the two work tables, where articles are temporarily placed for a next sequence.

The workflow of this peg-in-hole task is as follows.

1. No. 1 chucks the article placed on work Table 1 and places it on the small jig called temporarily placed position.
2. No. 2 chucks the article placed on the temporary position and puts it in the target hole on the work Table 2.
3. Chucking and placing operations are synchronously and alternately conducted by No. 1 and No. 2 while considering the efficiency to reduce the task execution time and the safe timing to avoid a collision.

The authors confirmed two results from the experiment. One is that HCLS data consisting of 100 GOTO statements with coordinate values of 100 workpieces were generated by the proposed position generation function while considering the misalignment. The other is that due to the HCLS data, the peg-in-hole task by the two MG400s was successfully performed without using any PLC.

Table 2. Network parameters and training condition of transfer-learning-based CNN (VGG19-based CNN).

Item	Value or Setting
Model name	VGG19-based CNN
Number of total layers	47
Number of total weights	139,578,434
Training images	156 (OK), 196 (NG)
Test images	156 (OK), 196 (NG)
Resolution of input images	$224 \times 224 \times 3$
Epoch size	50
Mini batch size	32
Initial learning rate	0.0001
Optimizer	SGDM
Loss function	Cross entropy loss
Convergence time [s]	871 (average)
Forward calculation time [s]	0.07 (average)

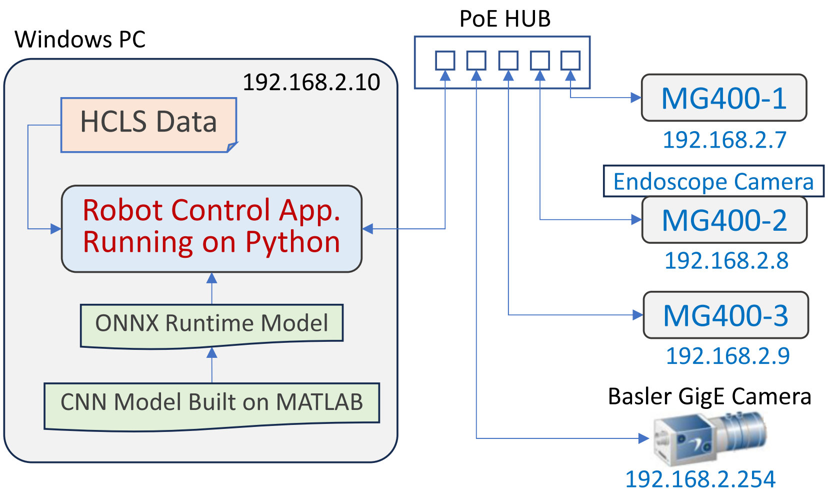

Figure 5. Overall configuration framework of the experimental system without using a PLC.

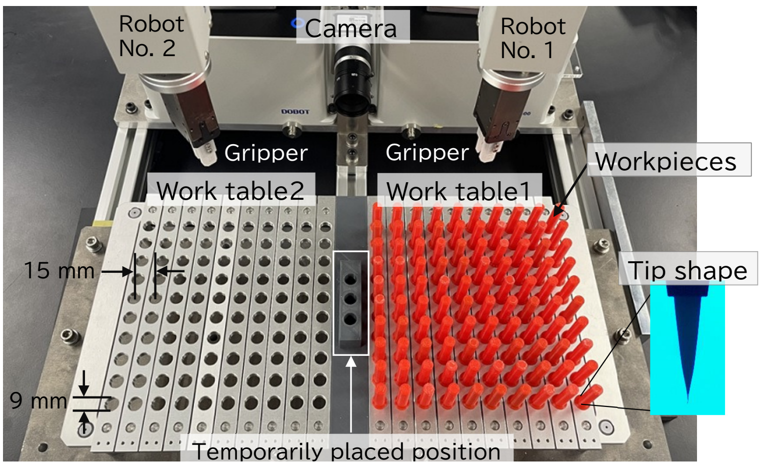

Figure 6. A cooperative peg-in-hole task experiment using two robot arms MG400 controlled by the developed application shown in Figure 1.

4. Advanced HCLS Data Statement for Calling a Defect Detection CNN Model

In this section, an advanced HCLS data statement coded as CNN_DEFECT is introduced to call a CNN model for defect detection of an industrial product. One of promising applications by industrial manipulators is for a robotic peg-in-hole task while checking undesirable defects. Figure 7 shows typical defects seen in the actual production line of this product. These types of defects tend to appear around the tip during the manufacturing process. Due to the statement of CNN_DEFECT, it is expected that undesirable defects around the tips of workpieces can be detected and sorted by the robots.

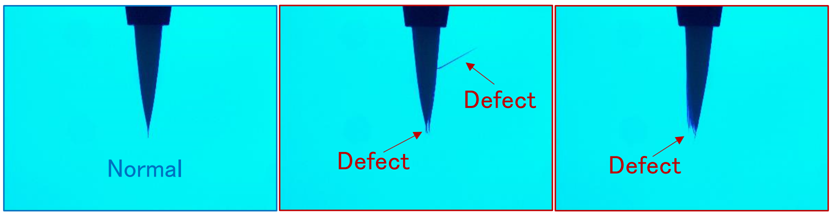

Figure 7. Typical defects of an industrial product.

Figure 8 shows the CNN model consisting of 47 layers for binary classification, i.e., OK or NG, built by the VGG19-based transfer learning method on MATLAB. The model is additionally trained using a training dataset consisting of 156 OK images and 196 NG ones. In designing the VGG19-based transfer-learning CNN model from the original VGG19 for 1000-class classification, we replace only the final fully-connected layers, the softmax layer, and the class output layer, then use the dataset for additional training to deal with the binary-class classification task. The additional training is conducted with a learning rate of 0.0001 up to the 42nd layer and with a learning rate of 0.001 from the 43rd layer to the final one in order that the powerful feature extraction capability originally acquired by the convolutional layers of VGG19 is not declined. Note that the resolution of input images has to be downsized into $224 \times 224 \times 3$ before training and test to fit the input layer of VGG19. Other training conditions are tabulated in Table 2.

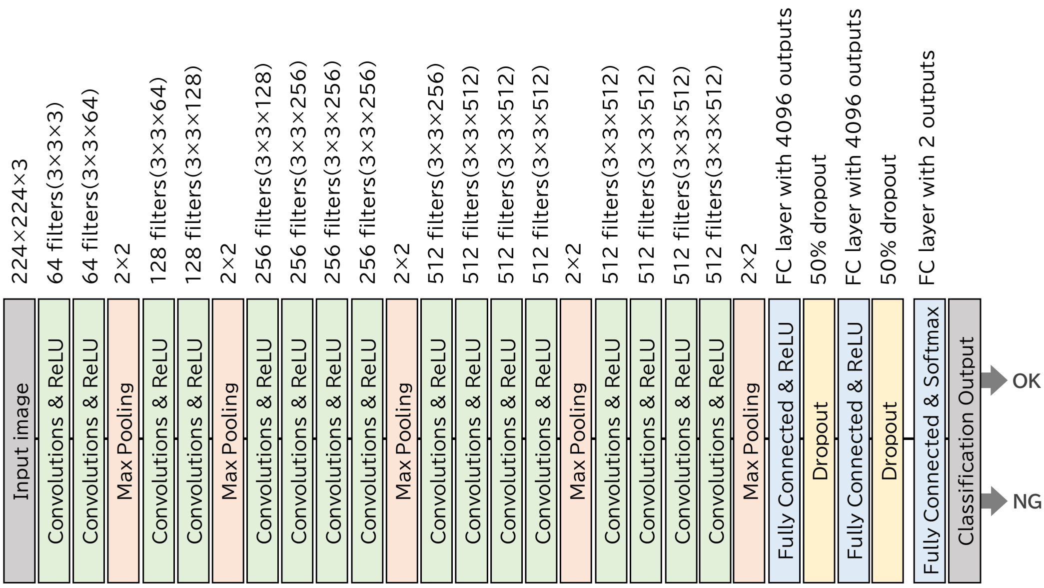

Figure 8. Transfer-learning CNN model designed based on VGG19 for binary classification, i.e., normal or anomaly. The CNN model built on MATLAB can be used in Python due to the interoperability provided by ONNX.

Table 3 shows the classification results of test images, in which an accuracy over 99% is achieved. The CNN model can be exported to an open neural network exchange (ONNX) model [25] that can be imported into Python. ONNX is the open standard format for machine learning interoperability, so that the exported ONNX model for defect detection can be applied to the robots running on Python.

For example, the statement of CNN_DEFECT in the HCLS data is utilized as listed in the following codes.

```
⋮

SNAPSHOT
R 120
SNAPSHOT
R -120
SNAPSHOT
CNN_DEFECT
PLACE
⋮
```

where SNAPSHOT is for the photo shoot of a workpiece by a camera set between the two robots, as shown in Figure 6. R 120 and R -120 rotate the workpiece with 120° and $-120°$ around the yaw angle using the robot's fourth joint, respectively. Consequently, three images captured from three directions are obtained at the step just before CNN_DEFECT. Then, the CNN_DEFECT statement calls the CNN model, i.e., imported ONNX model, shown in Figure 8, to predict each label of the three images. If all the three images are predicted as OK, the workpiece is classified as normal. Otherwise, the workpiece is regarded as an anomaly. Finally, the PLACE statement controls the robot so that the workpiece is placed into the designated hole according to the classification result.

Table 3. Confusion matrix classified by a trained CNN model transferred from VGG19 (row: predicted labels, column: true labels).

	OK (Normal)	NG (Anomaly)
OK (Normal)	153	0
NG (Anomaly)	3	196

The effectiveness and usefulness of the above HCLS data were demonstrated through a cooperative peg-in-hole task using two small-sized industrial robots, MG400, while alternately checking defects in workpieces by the VGG19-based CNN model.

As for the scalability of size of the available robots, if a target robot is provided by the same maker of MG400, then the Python application shown in Figure 1 can be directly used. Whereas if a target robot is provided by a different maker, regardless of the robot's size, its servo system has only to be technically opened to be able to implement the HCLS-data-based interface. Furthermore, although the available number of robots for cooperative control depends on that of allowable TCP connections, as can be guessed, the response between a PC and multiple robots tends to become slower with the increase in the number of connections.

To objectively evaluate the proposed system, quantitative results related to the performance improvements offered by the HCLS data interface are shown. For instance,

- The use of the automatic teaching point generator approximately resulted in a 50% reduction in teaching time compared to manual programming.
- The CNN-based defect detection system demonstrated an accuracy of over 99% in detecting anomalies during the peg-in-hole task, as shown in the confusion matrix in Table 3.

- The system's PLC-free cooperative control showed a significant reduction in coordination time compared to traditional PLC-based systems, improving the overall efficiency of multi-robot operations.

While directly comparable systems were not found in the literature, as discussed in the Introduction, these quantitative results demonstrate the effectiveness and scalability of the proposed approach. The improvements in time efficiency, accuracy, and overall system performance confirm the practical applicability of our system in industrial settings.

Actually, the authors have presented several industrial robot systems based on original CLS data for automating sanding, polishing, and machining processes, in which conventional position and orientation control and force control were only implemented [26]. The target robots with an open architecture controller were provided by Kawasaki Heavy Industries, Ltd. (Tokyo, Japan) and YASKAWA Electric Corporation (Fukuoka, Japan), and so on. Hence, the hyper CLS-data-based interface will be able to also be applied to other makers' industrial robots if the servo controllers are technically opened by, e.g., SDK or API functions.

5. Conclusions

In this paper, an HCLS-data-based robotic interface has been proposed to enhance the functionalities of industrial robots without using any PLC. Due to the HCLS-data-based robot interface, a robotic operation sequence can be visually and unifiedly described as NC codes. In addition, advanced functions such as visual feedback control for object tracking, a CNN model for defect detection, and cooperative control for multiple robots can be systematically described in HCLS data and stably executed. The effectiveness and validity of the proposed system have been demonstrated through a cooperative pick and place task using three small-sized industrial robots MG400s and a peg-in-hole task while checking for undesirable defects in workpieces by a CNN model.

It was observed from our past conducted experiments that VGG19-based CNN models are able to perform superior defect detection; therefore, in this case also, a VGG19-based CNN model was applied. However, as shown in Table 2, since the number of weights to be trained is more than 139 M, other simpler structured models should be applied. In future work, a lightweight model named WearNet is planned to be implemented in the HCLS-data-based robot interface. It is reported by Li et al. that WearNet could perform an excellent defect detection accuracy with a much smaller weight size of 0.16 M and faster detection speed [27].

While the proposed HCLS data interface offers significant advancements in cooperative control and defect detection, its application is currently limited to robots with open architecture controllers. Additionally, implementing the interface for complex multi-robot scenarios may face scalability challenges as the number of connections increases. Future work will address these limitations by optimizing the communication framework and expanding compatibility.

Author Contributions: Conceptualization, F.N. and K.W.; methodology, F.N. and R.A.; software, F.N. and R.A.; validation, F.N., R.A. and S.S.; formal analysis, F.N., R.A. and S.S.; investigation, F.N. and R.A.; resources, F.N. and R.A.; data curation, F.N.; writing—original draft preparation, F.N.; writing—review and editing, F.N., M.K.H. and K.W.; visualization, F.N. and R.A.; supervision, F.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SCARA	Selective compliance assembly robot arm
NC	Numerical control

CAD	Computer-aided design
CAM	Computer-aided manufacturing
HCLS Data	Hyper cutter location source data
CNN	Convolutional neural network
PLC	Programmable logic controller
OAIR	Open architecture types of industrial robot
SDK	Software development kit
DLL	Dynamic link library
ONNX	Open neural network exchange

References

1. Wang, S.; Chen, G.; Xu, H.; Wang, Z. A robotic peg-in-hole assembly strategy based on variable compliance center. *Int. J. IEEE Access* **2019**, *7*, 167534–167546. [CrossRef]
2. Jaskolski, P.; Nadolny, K. Characteristics of functional subsystems of modular didactic production system for gear trains. *Int. J. Mech. Energy Eng.* **2019**, *3*, 301–308. [CrossRef]
3. Zeng, G.; Chen, C.Y.; Huang, D.; Zhu, Y. Robotic trajectory planning based on CL data. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 1723–1728. [CrossRef]
4. Amersdorfer, M.; Meurer, T. Equidistant tool path and Cartesian trajectory planning for robotic machining of curved freeform surfaces. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3311–3323. [CrossRef]
5. Molotla, O.; Peña-Cabrera, J.M.; Lomas-Barrie, V. Configurable hybrid integral manufacturing platform: Subtractive-additive process with industrial robot arm, proof of concept results. *IEEE Lat. Am. Trans.* **2023**, *21*, 1227–1235. [CrossRef]
6. Ma, Y.; Fan, J.; Zhao, S.; Jing, F.; Wang, S.; Tan, M. From model to reality: A robust framework for automatic generation of welding paths. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16153–16164. [CrossRef]
7. Su, J.; Wang, L.; Liu, C.; Qiao, H. Robotic inserting a moving object using visual-based control with time-delay compensator. *IEEE Trans. Ind. Inform.* **2024**, *20*, 1842–1852. [CrossRef]
8. Park, J.; Jun, M.B.G.; Yun, H. Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (CNN). *J. Manuf. Syst.* **2022**, *63*, 539–549. [CrossRef]
9. Algburi, R.N.A.; Gao, H.; Al-Huda, Z. Improvement of an industrial robotic flaw detection system. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3953–3967. [CrossRef]
10. Gao, P.; Wang, J.; Xia, M.; Qin, Z.; Zhang, J. Dual-metric neural network with attention guidance for surface defect few-shot detection in smart manufacturing. *J. Manuf. Sci. Eng.* **2023**, *145*, 121010. [CrossRef]
11. Choi, J.G.; Kim, D.C.; Chung, M.; Lim, S.; Park, H.W. Multimodal 1D CNN for delamination prediction in CFRP drilling process with industrial robots. *Comput. Ind. Eng.* **2024**, *190*, 110074. [CrossRef]
12. Wang, D.; Han, C.; Wang, L.; Li, X.; Cai, E.; Zhang, P. Surface roughness prediction of large shaft grinding via attentional CNN-LSTM fusing multiple process signals. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 4925–4936. [CrossRef]
13. Mellado, J.; Nunez, F. Design of an IoT-PLC: A containerized programmable logical controller for the industry 4.0. *Int. J. Ind. Inf. Integr.* **2022**, *25*, 100250. [CrossRef]
14. Rashad, O.; Attallah, O.; Morsi, I. A smart PLC-SCADA framework for monitoring petroleum products terminals in industry 4.0 via machine learning. *Int. J. Meas. Control* **2022**, *55*, 830–848. [CrossRef]
15. Mühlbeier, E.; Bauer, V.; Schade, F.; Gönnheimer, P.; Becker, J.; Fleischer, J. Mechatronic coupling system for cooperative manufacturing with industrial robots. *Procedia CIRP* **2023**, *120*, 744–749. [CrossRef]
16. Fathi, M.; Sepehri, A.; Ghobakhloo, M.; Iranmanesh, M.; Tseng, M.L. Balancing assembly lines with industrial and collaborative robots: Current trends and future research directions. *Comput. Ind. Eng.* **2024**, *193*, 110254. [CrossRef]
17. Chen, L.; Wu, Y.; Du, Z.; Tao, T.; Zhao, F. Development of an industrial robot controller with open architecture. In Proceedings of the 2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Ningbo, China, 19–21 November 2017; pp. 754–757. [CrossRef]
18. Taesi, C.; Aggogeri, F.; Pellegrini, N. COBOT applications—Recent advances and challenges. *Robotics* **2023**, *12*, 79. [CrossRef]
19. Tang, Y.; Sun, K.; Zhao, D.; Lu, Y.; Jiang, J.; Chen, H. Industrial defect detection through computer vision: A survey. In Proceedings of the IEEE International Conference on Data Science in Cyberspace, Guilin, China, 11–13 July 2022; pp. 605–610. [CrossRef]
20. Abe, R.; Nagata, F.; Terasaki, D.; Kato, H.; Ikeda, T.; Watanabe, K. Proposal of hyper CLS data for industrial robots -HCLS statements for sequence control of multiple robots. *Artif. Life Robot.* **2023**, *28*, 750–756. [CrossRef]
21. DOBOT MG400. Available online: https://www.dobot-robots.com/products/desktop-four-axis/mg400.html (accessed on 3 October 2023).
22. DobotStudio Pro 2.8 User Guide (MG400&M1Pro). Available online: https://www.dobot-robots.com/service/download-center (accessed on 23 September 2024).
23. Miki, K.; Nagata, F.; Furuta, K.; Arima, K.; Shimizu, T.; Ikeda, T.; Kato, H.; Watanabe, K.; Habib, M.K. Development of a hyper CLS-data based robotic interface for automation of production-line tasks using an articulated robot arm. *Artif. Life Robot.* **2022**, *27*, 547–553. [CrossRef]
24. TCP. Available online: https://github.com/Dobot-Arm/TCP-IP-Protocol-4AXis/tree/master (accessed on 2 September 2024).

25. ONNX. Available online: https://onnx.ai/ (accessed on 2 September 2024).
26. Nagata, F.; Watanabe, K. *Controller Design for Industrial Robots and Machine Tools: Applications to Manufacturing Processes*, 1st ed.; eBook; Woodhead Publishing Limited: Sawston, UK, 2013; ISBN 9780857094636.
27. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef] [PubMed]