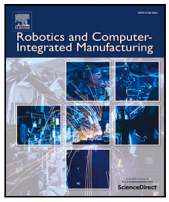
Robotics and Computer–Integrated Manufacturing 85 (2024) 102608

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Full length article

A deep learning-enhanced Digital Twin framework for improving safety and reliability in human–robot collaborative manufacturing

Shenglin Wang [a], Jingqiong Zhang [a,*], Peng Wang [c], James Law [b], Radu Calinescu [d], Lyudmila Mihaylova [a]

[a] Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, United Kingdom
[b] Department of Computer Science & The Advanced Manufacturing Research Centre, The University of Sheffield, Sheffield, S1 3JD, United Kingdom
[c] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, M15 6BH, United Kingdom
[d] Department of Computer Science, University of York, York, YO10 5GH, United Kingdom

ARTICLE INFO

Keywords:
Safe human–robot collaboration (HRC)
Intelligent sensing
Digital Twin
Semi-supervised deep learning framework

ABSTRACT

In Industry 5.0, Digital Twins bring in flexibility and efficiency for smart manufacturing. Recently, the success of artificial intelligence techniques such as deep learning has led to their adoption in manufacturing and especially in human–robot collaboration. Collaborative manufacturing tasks involving human operators and robots pose significant safety and reliability concerns. In response to these concerns, a deep learning-enhanced Digital Twin framework is introduced through which human operators and robots can be detected and their actions can be classified during the manufacturing process, enabling autonomous decision making by the robot control system. Developed using Unreal Engine 4, our Digital Twin framework complies with the Robotics Operating System specification, and supports synchronous control and communication between the Digital Twin and the physical system. In our framework, a fully-supervised detector based on a faster region-based convolutional neural network is firstly trained on synthetic data generated by the Digital Twin, and then tested on the physical system to demonstrate the effectiveness of the proposed Digital Twin-based framework. To ensure safety and reliability, a semi-supervised detector is further designed to bridge the gap between the twin system and the physical system, and improved performance is achieved by the semi-supervised detector compared to the fully-supervised detector that is simply trained on either synthetic data or real data. The evaluation of the framework in multiple scenarios in which human operators collaborate with a Universal Robot 10 shows that it can accurately detect the human and robot, and classify their actions under a variety of conditions. The data from this evaluation have been made publicly available, and can be widely used for research and operational purposes. Additionally, a semi-automated annotation tool from the Digital Twin framework is published to benefit the collaborative robotics community.

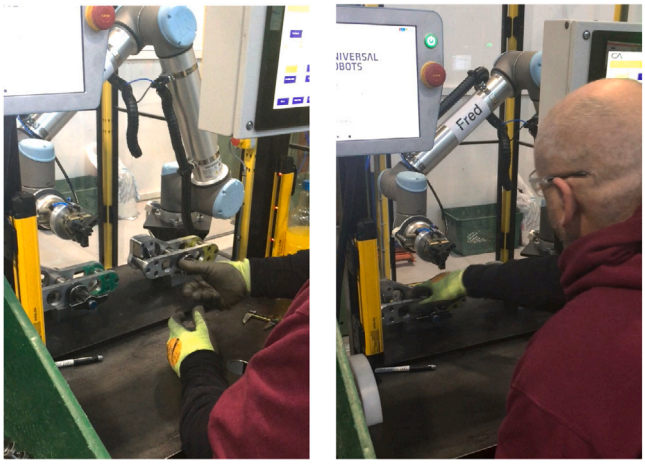
1. Introduction

Collaborative robots (cobots) [1] are playing an increasingly important role in the smart manufacturing and Industry 5.0 era, as they have the potential to boost productivity, ensure safety, and liberate humans from labor-intensive activities [2–4]. The concept of human–robot collaboration (HRC) in Industry 5.0 is mostly conveyed by smart manufacturing where cobots work alongside humans in close proximity in a shared workspace and they are pre-programmed to interact with humans to carry out various tasks. However, human safety is a key prerequisite for the deployment of such robots. Traditional approaches to ensure robot safety in manufacturing require deployment of cages, as shown in Fig. 1. Physical barriers, light gates, and laser rangefinders prevent direct contacts of cobots and humans [5]. These safety measures protect human workers, but they are bulky, inflexible (preventing true collaboration), and expensive.

In recent years significant research has been carried out to develop cage-free and more flexible safety solutions. Collision avoidance based solutions have been proposed in [6–8], where the pre-programmed trajectory of cobots are adapted to avoid collisions with dynamic obstacles, e.g., humans and other objects in the shared workspace. Unfortunately, these solutions lack the ability to distinguish 'humans' from other objects, which could subsequently cause severe consequences. In addition, these solutions rely on the alignment of digital cobots designed by Computer-Aided Design (CAD) tools [9] to re-built digital cobots from Red, Green, Blue plus Depth (RGB-D) camera data. CAD

* Corresponding author.
E-mail address: jingqiong.zhang@sheffield.ac.uk (J. Zhang).

https://doi.org/10.1016/j.rcim.2023.102608
Received 12 June 2022; Received in revised form 16 April 2023; Accepted 28 June 2023
Available online 16 July 2023



(a) (b)

Fig. 1. Figs. 1(a) and 1(b) shows the configuration of an industrial HRC process, where an operator exchanges components with a cobot at a shared handover location. The robot cell is open on one side, allowing staff to enter the cell under specific circumstances.

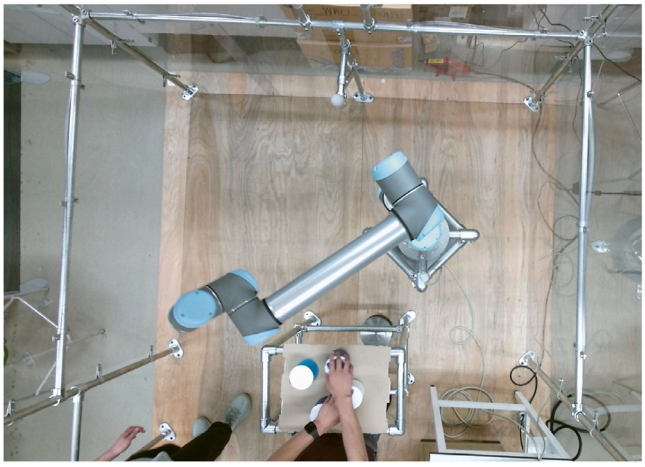


Fig. 2. A HRC cell is available in the Sheffield Robotics Lab at the University of Sheffield, UK. An HRC cell is shown in this picture, where there is an operator desk in front of the cobot and the operator exchanges components with the cobot on the desk. A Kinect sensor is mounted on the top of the cell to monitor the HRC operation.

models of cobots were combined with the data captured by RGB-D sensors. This leads to an easy separation of robots from surrounding objects and also from humans. The alignment between a CAD model and the caged cobot is typically done with the assistance of hand-eye calibration [9,10]. However, the calibration quality is critical in determining the accuracy of alignment.

Besides CAD models, augmented and mixed reality techniques which integrate computer-generated virtual information into real-world scenes can help users to enhance their understanding and awareness to support safe interaction in HRC tasks [11–13]. Meanwhile, thanks to the rapid development of deep learning and computer vision techniques, a series of modern approaches have been proposed [14,15], demonstrating success in scene understanding and visual perception, such as classification, object detection and segmentation.

Furthermore, Digital Twins of cyber–physical systems provide a real-time digital representation of physical collaborative manufacturing systems. This can greatly improve the systems' intelligence regarding design, production, operation, evaluation, health management and performance optimization [4,16]. Digital Twins can contribute to a range of different aspects in challenging HRC systems [17], including to simulation, modeling, performance analysis, process monitoring, data collection, data mining, data fusion, interaction as well as cognitive service [18,19]. This makes Digital Twins and intelligent solutions promising in avoiding the complex calibration process and in achieving identification of cobots and other objects in HRC without calibration at all.

Different definitions [18] of Digital Twins have been proposed and developed over time. According to [20], a Digital Twin is a set of coupled computational models and methods that evolve over time to persistently represent the structure, behavior, and context of a unique physical asset such as a component, system or process. A Digital Twin represents a real system, e.g. a city, cobot, aircraft, and acts as a coupled duplicate of the real world. It has several important characteristics: (i) it is **universal** and can be applied to several domain areas, (ii) it has a **modular structure**, which can be updated, expanded and developed further, (3) it is **connected with data** – both computer generated and from the real system. It can be used for a number of purposes – design, increasing safety and autonomy, and others, including for new functionalities. Fu et al. [18] point out four stages in the development of Digital Twin, with an increasing usage of data in the last two stages, including remotely, when data could be stored on a cloud and accessed via the Internet of Things (IoT) technologies. The surveys [21,22] systematically review the recent developments of artificial intelligence-driven Digital Twin in the areas of cutting-edge robotics and smart manufacturing. Besides, multi-access edge computing was incorporated into Digital Twins, facilitating manufacturing processes towards smart and flexible [23,24].

Having in mind these recent trends [25,26], one can identify several gaps between the research in Digital Twin techniques and their applications in industry: (i) Digital Twins need further developments in order to represent manufacturing systems in a wide range of complex environmental conditions and diverse production stages, (ii) In the majority of cobot systems, safety is guaranteed via caged environments or additional safety sensors when the cobots are operated at higher speeds so as to meet production demands of end users, (iii) the level of autonomy varies across different applications and is on the increase thanks to recent developments in intelligent sensing, computer vision and artificial intelligence techniques.

Aiming at contributing towards bridging these gaps, this paper proposes an intelligent Digital-Twin-based safe human–robot collaboration framework. A Digital Twin is built to simulate the physical HRC system which is shown in Fig. 2. A communication framework is further designed so that the Digital Twin can be synchronized with the physical HRC platform with the support of the Robot Operating System (ROS) [27]. Consequently, information including robot poses and kinematics can be shared between the digital and the physical systems flexibly and in a real-time manner. Owing to the Digital Twin's ability to create photo-realistic digital cobots and maintaining holistic cobot parameters, a diverse amount of synthetic cobot data with accurate labels are generated by the digital system. These data combined with human data from the COCO repository [28], are used to train deep learning models to monitor interactive operations of robots and humans. The challenges stemming from the simulated Digital Twin environment and the real environment are addressed by further proposing a semi-supervised deep learning detector. Our Digital Twin system is applied to analyze and validate how the environment, e.g. the lighting conditions, affect the performance of the deep-learning action-recognition system. With the proposed deep learning detector, humans and robots are monitored in the physical environment to ensure their safe separation. Therefore, by adopting a Deep Learning-enhanced Digital Twin Framework, this work contributes towards cost-effective and flexible systems for intelligent sensing and decision making.

The main contributions of this work are as follows: (i) a semi-supervised framework for object detection is proposed by adopting a faster region-based convolutional network [29]; (ii) a Digital Twin of a physical HRC system is developed that generates synthetic robot data to train deep learning models for monitoring human–robot collaborative behaviors. (iii) the performance of the developed Digital Twin system is validated and evaluated over both synthetic and real data sets, demonstrating that it can achieve accurate recognition of human–robot behaviors for safety assurance. Research outputs include publicly available datasets generated by our Digital Twin of a Universal Robot 10 (UR10) robot [30], and a semi-automated annotation tool [31].

The remainder of this paper is organized as follows. Section 2 gives an overview of related work from three perspectives: (*i*) needs and challenges in the manufacturing industry, (*ii*) machine learning methods for solving complex cobot tasks, (*iii*) Digital Twins of HRC systems. Section 3 describes the developed framework for safe and reliable HRC in detail. Section 4 describes the real and synthetic datasets along with the semi-automated annotation tool used in this work. Section 5 presents evaluation and validation of the detection and classification results under different lighting conditions, whilst explaining safety criteria for decision making and demonstrating how to implement or adopt our framework into practical cases. Finally, Section 6 summarizes the results and discusses future work.

2. Related work

2.1. Digital twins for HRC safety and resilience in manufacturing

Simulation models play a variety of roles in designing, testing and delivering products in industry. However, the frequently changing demands, the need for real-time process monitoring, and the need for cost-effective production [32] pose new challenges to simulation techniques. Digital Twins extend traditional simulation approaches by taking real-time and historic data from their counterpart physical systems into consideration. Such techniques have drawn significant attention from both industry and academia [19,20], especially for production lines in which humans work in shared spaces with robots. Digital Twins can combine cyber and physical information together throughout a product lifecycle, and are recognized as one of the most prospective tools for the design, maintenance and monitoring in smart manufacturing [19]. Thanks to the advances of the artificial intelligence, cyber–physical system, big data, information fusion and advanced sensing, Digital Twin technology is developing and shaping the manufacturing industry towards intelligent HRC [18].

Malik and Brem [3] propose a framework that applies a Digital Twin to industrial assembly systems. The system adaptability and dynamics due to the human presence are represented within the Digital Twin which significantly improves the safety in HRC. In [33], a machine learning-enhanced Digital Twin is proposed as an experimental platform to verify the proposed deep learning model for path planning before further actions in physical environments. This is particularly beneficial for scenarios where humans are involved in validation as any unaddressed issues could lead to injuries to humans. Besides, deep learning methods can be embedded into a rich Augmented Reality (AR) environment by mapping the virtual and physical objects during the multi-functional interaction, and can have a better preview of target objects [34]. Park et al. [35] design a hands-free interaction system in mixed reality environments with the assistance of a Digital Twin.

In contrast to previous studies, our Digital Twin is able to assist in the training of deep learning models by generating training datasets in addition to testing and validating the model. This enhances the efficiency of training the deep learning model in terms of both time and labor costs.

The next subsection presents an overview of recent advances in deep learning methods for object detection.

2.2. Fully-supervised and semi-supervised deep learning for object detection

Several fully-supervised object detection algorithms have been proposed and one of the famous series are region-based convolutional neural networks, also named two-stage detectors, including R-CNN [36], Fast R-CNN [37] and Faster R-CNN [29]. In these two-stage methods, the first stage is to extract image features through backbone networks, for instance, ResNet [38]. The second stage generates region proposals for further localization and classification of objects. During the evolution of region-based convolutional neural networks, the computation costs of region proposal generations decrease significantly from a selective search [39] to the Region Proposal Network (RPN) in faster R-CNN [36]. The RPN can achieve real-time performance and it has made great progress in detection accuracy.

The object detection algorithms reviewed above belong to the group of fully-supervised algorithms which means that they require huge amount of labeled data during the training process. In contrast to fully-supervised algorithms, semi-supervised ones use partly labeled data and partly unlabeled data or pseudo-labeled data, which can significantly reduce the volume of labeling data to some extent. Pseudo-label based approaches adopt the teacher-student model in which a teacher model firstly is trained to generate pseudo-labels. Unlabeled data combined with the pseudo-labels are then used to train the target student model. In FixMatch [40], Sohn introduces weakly-augmented data for generating pseudo-labels and then the same strongly-augmented images are applied to predict whether the results match the weakly-augmented one. In [41] pseudo-labels are generated by using data augmentation and high efficiency is achieved compared with fully-supervised faster R-CNN [36]. Xu et al. [42] propose a soft teacher model which performs pseudo-labeling on weakly augmented data. The teacher model is updated by using the student model which applies an exponential mean average (EMA) strategy.

In previous studies on semi-supervised solutions [43], data preprocessing and data augmentation are required to train a well-performed detector. Additionally, the detector only considers a single domain where both the labeled and unlabeled data are from the same domain. It may result in detection accuracy degradation, especially in new unseen environments. Our semi-supervised object detection approach considers both domains from the physical environment and the simulation, and hence can achieve satisfactory performance in new real environments.

2.3. Bridging the gap between simulation and the physical world

It is expensive to collect and annotate huge amounts of data for training a deep learning model. Especially, there is no public dataset available that can be used to train deep learning models in new environments, especially for manufacturing purposes, for instance, using a deep learning-based detector to detect robots and humans in HRC. Digital Twin is an efficient option in which simulations in the digital system can generate a great amount of labeled data [17,26]. These generated data can be used for training deep learning models and applied to real-world environments. Techniques known as Simulation to Real (Sim2Real) [44–46] could be utilized in such tasks. In addition, they enable only the (simulated) virtual world to be employed during the training, validation, and testing phases of the deep neural network (DNN) models. However, there are still cases showing that such models tend to perform inaccurately when evaluated in real world applications, due to the discrepancies between the simulated virtual and the real worlds.

The main objective of [47] is to detect objects on the table in a real-world environment and to estimate the object's position. To satisfy the requirements of transferring the model trained in the simulator to the physical world, Tobin et al. [47] randomized with respect to distractors, objects, backgrounds and lighting conditions. The model was trained directly on the simulator, and succeeded in estimating the position

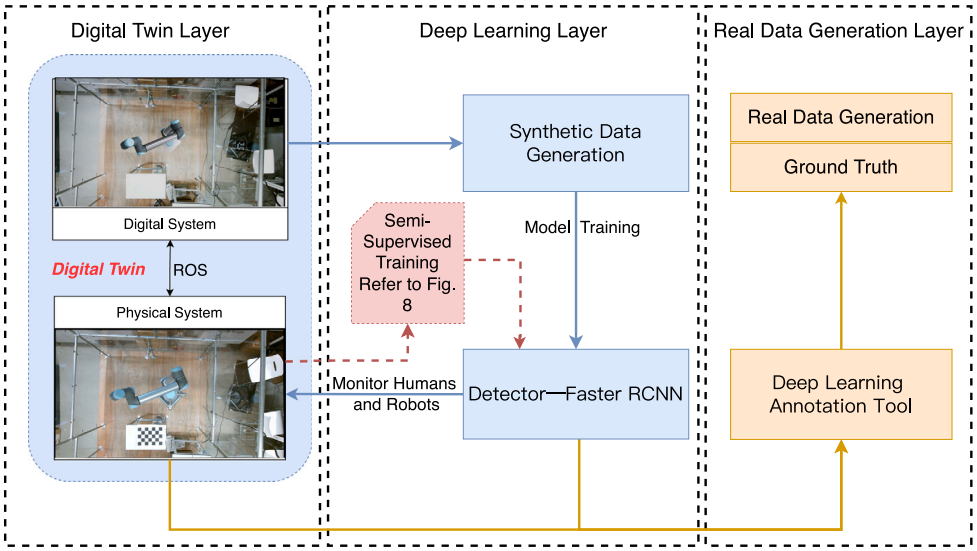


Fig. 3. Theoretical framework of using deep learning and Digital Twin techniques for monitoring Cobots towards safety and reliability. The framework is comprised of three layers: (i) Digital Twin layer, (ii) deep learning layer, and (iii) real data generation layer. Digital Twin layer illustrates the Digital Twin in which a ROS-based communication system is designed for information transmission including robot pose, the orientation and position of the camera, etc. between the digital and the physical system. Deep learning layer represents how the synthetic dataset with accurate annotations is generated, then the detector is trained with the dataset. The detector is applied to monitor humans and the cobot in the physical system. In the meanwhile, it also illustrates how a semi-supervised detector is trained which will be explained in Section 3.4. In the real data generation layer, a deep learning-based annotation tool is developed to assist to collect and annotate real data.

of various shape-based objects on the table in the physical world. With respect to object detection, Tremblay et al. [48] applied similar strategies as in [47] to detect real objects in complex backgrounds. Compared with the method developed in [47], they introduced a new component called flying distractors which improve the accuracy of detection. Furthermore. Tremblay et al. investigated the importance of each randomization parameters. During the training process, environment parameters are uniformly randomized in the simulation model, but the sample complexity grows with the increasing number of randomization parameters [47–49]. However, it is difficult to find out what could cause failures during this randomization process. To solve the problems described above, Mehta et al. [50] find out the most informative environment variations in the range of given randomization parameters.

Domain Randomization is adopted in the digital system of the proposed Digital Twin in this paper. A simple and efficient scheme generates the synthetic dataset. The digital system has the same configuration as the physical system which is shown in Fig. 3, besides the randomization parameters. Furthermore, previous studies only considered the synthetic data but ours also adopt the unlabeled real data so as to minimize the gap in Sim2Real [44–46].

3. The deep learning-enhanced digital twin framework

Traditional solutions to prevent hazardous human activities with cobots include physical safety barriers, proximity sensors, and light gates, which have major disadvantages of big size, difficult maintenance, inability to adapt under various operating conditions, and sometimes high cost [51,52]. To meet the high requirements for cobots towards safety and reliability, this paper proposes an intelligent and flexible deep learning-enhanced Digital Twin framework for monitoring the human–robot collaboration with a high level of autonomy in manufacturing.

The performance of our framework is demonstrated and evaluated on a Universal Robots UR10 platform using a Microsoft Kinect V2 sensor as shown in Fig. 2. The framework does not require any complicated and time-consuming sensor calibration.

Fig. 3 shows the Digital Twin including the proposed deep learning model which consists of three layers: (i) Digital Twin layer, (ii) deep learning layer, and (iii) real data generation layer. In the Digital Twin layer, a virtual robot in the digital system captures the pose of the physical robot in the physical space during the working process via the ROS, so that the virtual robot performs in the same way as the physical robot. The virtual visual sensor in the digital system has a different function — to capture synthetic data of the robot with random position and orientation. The data annotation information is also generated automatically along with the collection of the synthetic data. During the synthetic data preparation, Domain Randomization as described in Section 3.2.2 is applied to the digital system with the aim of bridging the reality gap between the real world and the simulation.

In the Deep Learning layer, the synthetic data from the digital system is provided for training a faster R-CNN detector. The detector combined with the deep learning annotation tool is applied to collect the annotated real data in the Real Data Generation layer. With the real data, a semi-supervised method described in Section 3.4 is implemented to train a new detector. This semi-supervised detector monitors the interactions between humans and robots in the physical system of the Digital Twin layer to achieve a safe HRC.

This framework provides a cost-efficient solution to generate data with accurate annotations and other types of sensor information such as mask, bounding boxes, RGB, and depth information. A semi-supervised deep learning model is presented to narrow the gap between the digital system and the physical system. Consequently, the detector proposed in this work can achieve more accurate detection, compared to those fully supervised detectors which are purely trained with real or synthetic data.

3.1. Communication design of the digital twin

In traditional simulators, e.g., Gazebo [53] and CoppeliaSim [54], all designs, simulations and experiments are finished in such a closed environment without connecting to any other physical systems. However, a Digital Twin requires not only simulation but also a physical test. Consequently, to satisfy this requirement, a data transmission framework is needed.

In our Digital Twin, bidirectional data transmission is enabled between the physical system and the digital system, which should be

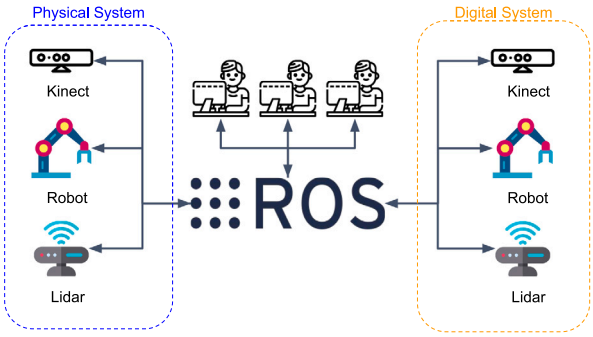


Fig. 4. A ROS based communication framework is designed for the Digital Twin. In this framework, cameras, cobots and users are regarded as nodes. In a ROS framework, nodes communicate with each other through topics, services, and actions provided by ROS.

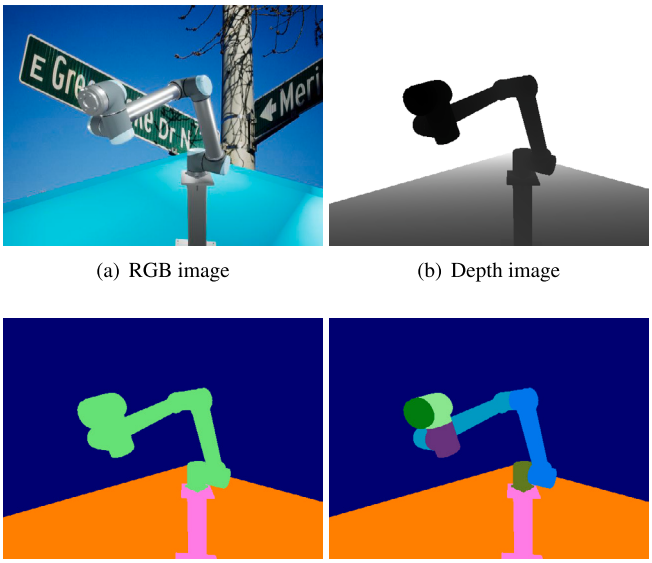
capable of performing multiple processes in a real-time manner. UnrealCV [55] built a file transfer protocol (FTP)-based communication system that only listens to a single socket and the one-way transmission allows only one pack of control data during the whole transmission. Consequently, it cannot support a multi-user control at the same time, i.e., the camera and the cobot cannot be controlled in parallel. A higher level communication design is required to meet the synchronous data transmission between the digital system and the physical system in the Digital Twin.

ROS has been used to facilitate the implementation of the overall system. ROS is a distributed system where a synchronous data transmission can be achieved when the digital system and physical system do not need distant communication. A ROS based communication framework is built for the Digital Twin to achieve data transmission among multiple clients. Fig. 4 shows how the communication framework is implemented in the Digital Twin. Clients such as cameras, cobots and users are regarded as nodes. Different nodes communicate with each other through topics, services, and actions provided by ROS. For instance, a node can publish defined messages (data) onto a topic, and other nodes subscribing to the topic can receive the message. In our case, joint angles of the physical cobot in the physical system are published, and joint angle data are subscribed by the digital cobot in the digital system (see Fig. 3). As a result, both physical and digital cobots move synchronously and keep the same poses. In the meantime, the digital robot can also publish verified robot poses and trajectories to the physical system so that the physical robot can implement specific task without further tests and trials.

3.2. A digital twin for synthetic and real data acquisition

3.2.1. Data acquisition and data types

Unreal Engine 4 (UE4) [56] is a powerful gaming engine that has the capability to simulate a physical world realistically. To some extent, the usage of UE4 can minimize the reality gap due to its photorealism. The developed Digital Twin framework uses UE4 as a digital system environment to generate the synthetic data with annotation information for training the developed faster R-CNN [36] and validating its performance for detection of the areas of the human and of the cobot and making decisions on whether the safety standards are satisfied. With the assistance of the communication framework in the Digital Twin, users can control the camera mounted on the top of the physical robot cell and the physical robot in the physical system to collect the real data as well. In the physical system, to capture images of how the robot carry out its task, the robot arm is moving from one pose to another. At the same time, users can control the camera to collect data from frame to frame. The size of collected images is 1920 × 1080. Furthermore, with the trained detector and an annotation



(a) RGB image (b) Depth image

(c) Object mask (d) Object mask with different components

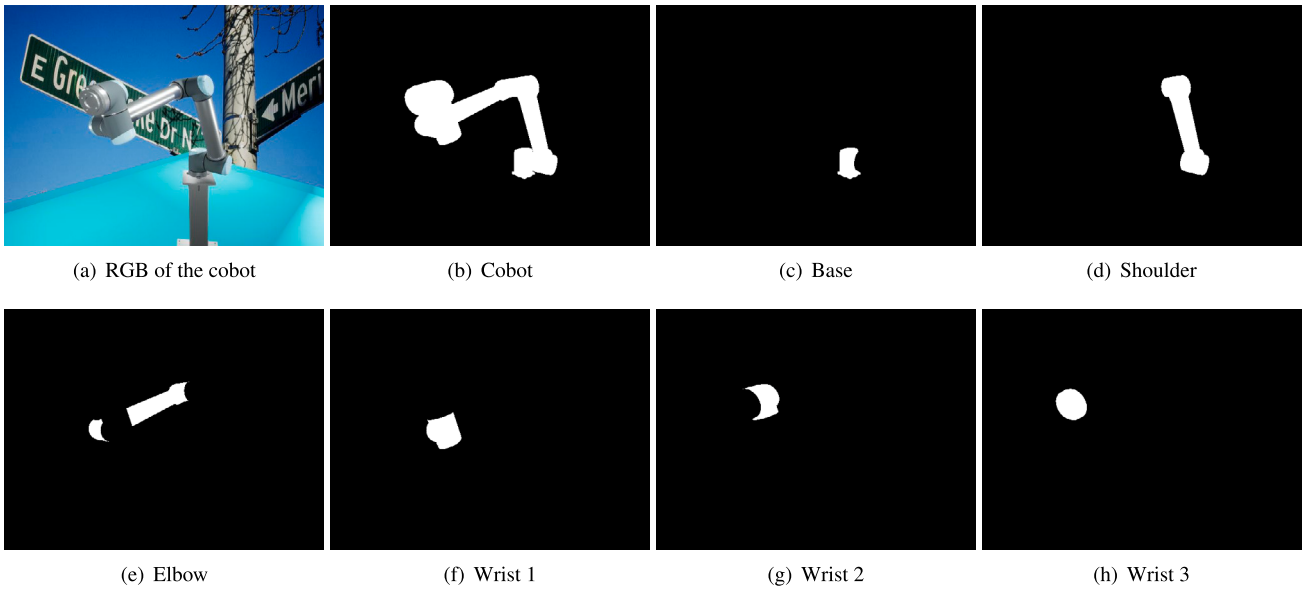
Fig. 5. The synthetic data including different types of sensing information of the cobot generated from the Digital Twin.

tool described in Section 4.1, can collect and annotate the raw data efficiently and reduce manually labeling time effectively compared to traditional manual data acquisition and annotation.

Compared to in stock sensors such as RGB cameras that provide specific types of data, the Digital Twin system is more efficient and flexible in obtaining various sensing information with the help of UE4. Fig. 5 displays examples of different types of sensing information generated by UE4. UE4 renders objects with their original colors to generate RGB images as shown in Fig. 5(a) and it also provides depth information in Fig. 5(b). Depth information gives rich 3D information which is of benefit to get the location and orientation of objects. With additional user-defined color information, UE4 can also render an object with a defined single color. Consequently, the annotation of the object can also be obtained with the defined color. The accurate annotation, as demonstrated in Fig. 5(c) and (d), is useful for instance segmentation and object detection.

Different from UnrealCV [55], our Digital Twin framework provides more flexibility for users in how the annotation of an object is represented. In UnrealCV [55], masks of different objects can be obtained when the specific color is known. For instance, the blue color often represents the background, the green is the robot and the orange is the ground floor in Fig. 5(c). However, it is impossible for users to get component masks of an object, because their mask rendering solution only queries from object to object when rendering an object mask scene. This means that the components of the object are not queried during rendering and they cannot be rendered as different colors. Fig. 5(c) illustrates that the robot is rendered with single color. In our digital system, the rendering logic is different from the UnrealCV [55] where the digital system both queries what objects exist in a scene but also checks the components of the objects during rendering of a mask scene. Consequently, it can render the components with defined colors which are specified by users when generating mask annotation in which the components of the cobot is rendered with different colors as shown in Fig. 5(d), compared to Fig. 5(c). Furthermore, different components can be identified in one object and the component masks can be obtained once the colors are known as shown in Fig. 6.

Through our Digital Twin, it is easy and efficient to get these types of information which are expensive in traditional manual annotation. The flexibility of the data generation in our Digital Twin is able to



(a) RGB of the cobot (b) Cobot (c) Base (d) Shoulder

(e) Elbow (f) Wrist 1 (g) Wrist 2 (h) Wrist 3

Fig. 6. (a) represents a RGB image of the cobot, while the masks of the cobot and its components are illustrated from (b) to (c). The digital system can generate different component masks which is defined by users. The cobot mask can be separated into different components and components can be combined as the one. Consequently, users can obtain masks based on their requirements to meet different tasks.

meet different tasks including robot detection, robot grasping, pose estimation, etc.

3.2.2. Domain randomization

Bridging the reality gap between the physics simulators and the real world is challenging. The aim of the Sim2Real tool is to transfer the virtual models to the real world situations. One approach to generating high quality realistic virtual images is to deploy high-quality rendering simulators such as Unity3D [57], UE4 [56] and OpenGL [58]. In the next paragraphs, we introduce the main mathematical notations and concepts that are needed for the description of the DNN model.

A domain, defined as D, is composed of a d-dimensional feature space $\mathcal{X} \subset R^d$ with a marginal probability distribution $P(\mathbf{X})$, and the task in this domain is defined as $\mathcal{T}$. Given a training set $\mathbf{X} = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$ and its labels $\mathbf{Y} = \{y_1, \dots, y_n\}$ of the label space $\mathcal{Y}$, the conditional probability distribution is $P(\mathbf{Y} \mid \mathbf{X})$.

In Sim2Real, it is assumed that there are two domains: a source domain (i.e. simulator) $D^s = \{\mathcal{X}^s, P(\mathbf{X}^s)\}$ with a task $\mathcal{T}^s = \{\mathcal{Y}^s, P(\mathbf{Y}^s \mid \mathbf{X}^s)\}$ and target domain (i.e. physical world) $D^t = \{\mathcal{X}^t, P(\mathbf{X}^t)\}$ and its corresponding task $\mathcal{T}^t = \{\mathcal{Y}^t, P(\mathbf{Y}^t \mid \mathbf{X}^t)\}$.

The solution introduced by Tobin et al. [47] is a simple but powerful technique for training models on simulated images, i.e. on the source domain D^s. The model is able to be transferred to the physical world, i.e. target domain D^t. The basic concept of [47] is that by randomizing and rendering in the simulator, the model trained with the randomized synthetic images can be adapted to the real images. Tobin et al. [47] assumed a hypothesis that a set of randomization parameters can be controlled in the simulator. In the process of generating simulated data, if the variability of the simulator is diverse enough, the physical world may appear as another variation in the simulator, i.e. $D^t \subset D^s$ and $P(\mathbf{Y}^t \mid \mathbf{X}^t) \subset P(\mathbf{Y}^s \mid \mathbf{X}^s)$. Consequently, the model trained in the simulator will generalize to the physical world with no cost of additional adjustment on training and represent well the real world.

In our framework, Domain Randomization is applied in the digital system to generate abundant samples with the aim of bring the simulated images close to the real ones. It is demonstrated that the model trained over the synthetic data with Domain Randomization has accurate performance under different lighting conditions which will be illustrated in Section 5. The Domain Randomization helps improving the deep learning detector and its ability to work under a variety of conditions. Advantages of the digital system are its flexibility, ability to annotate images accurately and to diversify inputs in the feature space. Limitations exist in generating a real dataset with respect to sample diversity. These limitations are linked to a number of factors such as the fixed orientation and position of sensors, unchanged lighting conditions and unchanged backgrounds. These limitations may cause inadequate generalizations and lack of model adaptation in new environments. However, these limitations can be regarded as changeable variations with respect to randomization parameters in the simulators. The randomization parameters considered in the digital system are the following: strength and color of the direct light, position and orientation of the direct light, position and orientation of the camera, images of backgrounds which are from the COCO dataset [28] along with poses of the robot. With these randomization parameters, different kinds of samples can be easily generated, with different appearances, Consequently, the generated dataset can be diverse enough to help the source domain (simulation) to get closer to the target domain (real). It is difficult to collect such different kinds of samples in the real world system due to device limitations.

3.3. A digital twin for intelligent sensing and machine vision tasks in changeable environments

The Digital Twin framework proposed in this paper adopts the faster R-CNN [29] as a detector to verify the performance of the model trained with different synthetic and real data, under different lighting conditions. The architecture of the considered faster R-CNN [36] is presented in Fig. 7. The Faster R-CNN [36] consists of two stages: (1) feature extraction from the input image, and (2) generation of potential region proposals where the location of the object of interest is, calculated with a region proposal network (RPN). As shown in [29], Faster R-CNN can achieve accurate detection in real-time performance. By using the Non-Maximum suppression operation [59], proposals with low confidence are filtered. The remaining proposals and feature maps are refined by the next layer for the Region of Interest (RoI) Pooling stage. The corresponding proposals are classified as different objects as well as their bounding boxes are predicted.

The architecture of faster R-CNN [29] includes ResNet-50 [38] for extracting features from images. The residual block is defined as

$$\mathbf{y} = \mathcal{F}\left(\mathbf{x}, \{W_i\}\right) + \mathbf{x}, \tag{1}$$

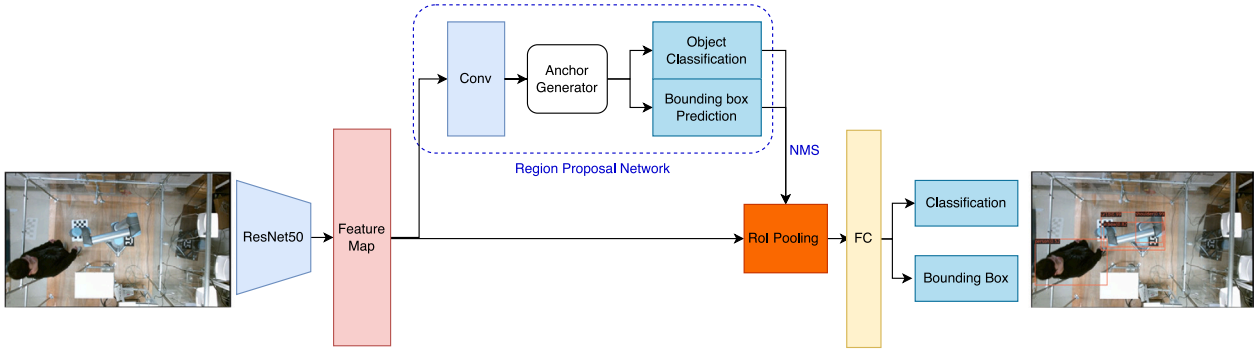


Fig. 7. Architecture of the faster R-CNN. ResNet-50 extracts feature maps from the input image. In Region Proposal Network, regions of interest are generated. RoI Pooling processes the regions of interest and their corresponding feature maps to get new feature maps with fixed size. The FC (Fully connected layer) predicts the classes and the bounding boxes for these feature maps.

where **x** is the input image for the residual block, **y** is the output image feature map which is coming out of the residual block. The function $\mathcal{F}$ represents the residual mapping and $\{W_i\}$ denote the weights of layers in the residual block. The detector block includes two sub-tasks: object classification and bounding box regression for object detection. In the two-stage detector, both loss functions in the Region Proposal Network (RPN) and the final Region of Interests (RoI) results are considered.

In the Region Proposal Network (RPN) [29], the loss function L_{RPN} of the RPN is defined as:

$$L_{RPN}\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* R\left(t_i - t_i^*\right), \tag{2}$$

where p_i is the predicted probability of the ith anchor, which is a binary result characterizing whether the anchor is an object or not, and t_i is the corresponding bounding box prediction, N_{cls} is the normalized parameter for the classification. The classification loss in RPN is denoted as L_{cls}, and p_i^* is the corresponding ground-truth, whose value is 1 (positive) or 0 (negative). The balanced parameter is denoted as λ while the N_{reg} are normalized parameters of the regression. The bounding box is optimized with the smooth L1 regression loss function R, and t_i^* is the ground-truth of the bounding box of anchor i. The smooth L1 loss function R is defined in the form:

$$R\left(t_i - t_i^*\right) = \begin{cases} 0.5\left(t_i - t_i^*\right)^2 & \text{if } \left|t_i - t_i^*\right| < 1 \\ \left|t_i - t_i^*\right| - 0.5 & \text{otherwise.} \end{cases} \tag{3}$$

For the classification loss function in the RPN, a binary cross entropy loss is adopted

$$L_{cls} = p_i^* \log(1 - p_i) + (1 - p_i^*)\log(p_i). \tag{4}$$

In the final RoI area, the cross entropy loss L_{cls}^{roi} for object classification and the smooth L1 loss L_{bbox}^{roi} for bounding box regression are introduced, so the total loss function L required to be minimized is

$$L = L_{RPN} + L_{cls}^{roi} + L_{bbox}^{roi}, \tag{5}$$

where *bbox* denotes the bounding box regression.

3.4. A semi-supervised teacher-student detector for Sim2Real

A detector trained with the synthetic data can achieve an effective performance in the real world environment. It still needs to be validated whether the detector using both synthetic and real data would have accurate performance within the Digital Twin. A semi-supervised solution is proposed to train a detector of human actions and the whole framework is shown in Fig. 8. Our semi-supervised method is based on the faster R-CNN [29] framework. Our solution consists in a teacher-student model to train a student model through semi-supervised training. The teacher model is trained with synthetic data $D^{syn} = \{\mathbf{X}^{syn}, \mathbf{Y}^{syn}\}$. Once the teacher model is trained, the real data is input without the ground-truth $\mathbf{X}^{real}$ to the teacher model during the testing mode Then the model will give predicted labels of $\mathbf{X}^{real}$, which is denoted as $\widetilde{\mathbf{Y}}$. However, the real data with its predicted labels $\left\{\mathbf{X}^{real}, \widetilde{\mathbf{Y}}\right\}$ cannot be used to train the student model directly, because some redundant and low-quality results exist in its prediction $\widetilde{\mathbf{Y}}$. To filter these redundant and low-quality results, the Non-Maximum suppression operation [59] is implemented. The faster R-CNN predicts objects in an image with their bounding boxes and classes with confidence which are regarded as the predicted label $\widetilde{\mathbf{Y}}$ for an input $\mathbf{X}^{real}$. In the Non-Maximum suppression operation, the bounding boxes of each class are ranked by their confidence. The bounding boxes of each class with the highest confidence are remained which are $\hat{\mathbf{Y}}$ while the rest are filtered. After $\widetilde{\mathbf{Y}}$ being filtered, pseudo labels $\hat{\mathbf{Y}}$ are obtained.

In the next step, the real data with its pseudo labels $D^{pseudo} = \left\{\mathbf{X}^{real}, \hat{\mathbf{Y}}\right\}$ is applied to train the student model. The weight of the teacher model will be frozen as a pre-trained for training the student model. The student model is the final model that is applied to monitor the interactions between the robots and humans in the physical system. It achieves more accurate and more robust results under changing lighting conditions compared to the fully-supervised faster R-CNN. The performance of our framework is evaluated in Section 5.

3.5. Relevance to the standards and regulations for HRC

Digital Twin technology provides an enormous potential for incorporating health and safety regulations into cobot systems and vice versa, the Digital Twin can impact the standards and regulations towards higher safety and reliability of these systems. Some of the main safety regulation documents [51,52,60], especially applicable to manufacturing, do not consider various levels of autonomy for the needs in different industrial applications. A part of the technical challenge is to identify and assess the underlying hazards and risks of these cobot systems when not being operated in power and in force limiting (PFL) mode. Particularly, this is especially important in highly automated manufacturing industry which employs intelligent sensing and artificial intelligence systems. In the considered UR10 cobot system, traditional sensors were used for which the current standards and regulations [51,52] have well specified safety rules. These safety rules include proximity and light gates, to avoid hazardous humane–robot collisions.

This work proposes an autonomous decision-making framework utilizing vision cameras, with the advantage of being able to rapidly adapt to dynamic environments. Additionally, in consideration of the

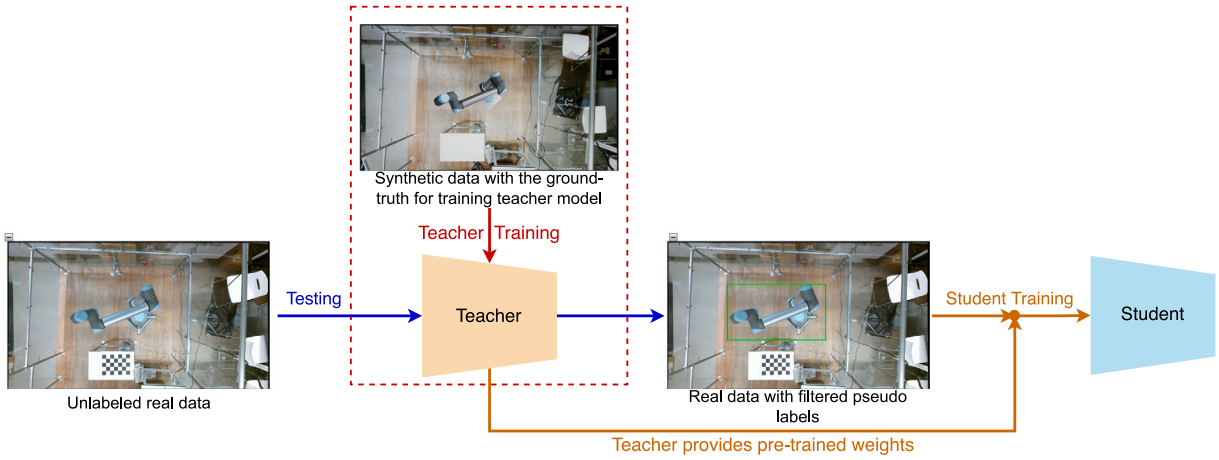


Fig. 8. Framework of the semi-supervised method applied to train a detector. A teacher model is firstly trained with the synthetic data. The unlabeled real data is fed to the teacher model and the teacher model generates pseudo labels for the unlabeled real data during the testing mode. The pseudo labels are further filtered. Next a student model is trained with the real data with filtered pseudo labels.

physical reconfiguration of safety sensors as robot movements are reprogrammed for conducting different tasks, the positioning and installation of vision sensors are relatively easy to achieve, compared with light gates and physical fences.

According to the relevant sensing standards [61] which illustrate the requirements for equipment using vision based sensors, several environmental factors should be considered when implementing such sensors into real industrial applications, including optical occlusion, various ambient temperatures and lighting conditions. Due to practical considerations of complex industrial conditions, our detector for actions recognition of human–robot interactions is tested under different lighting conditions in terms of the accuracy of object detection as depicted in Section 5. In practice, our detector can be embedded both in a Digital Twin platform and in the control algorithms of a cobot system. Accordingly, if dangerous scenarios such as unsafe interactions or abnormal operations are detected successfully, the operator would be alerted by relevant warnings whilst brake signals would be sent to the controller to delay or stop the robot movements for guaranteeing safety.

4. Datasets

To build a deep learning-based detector, two different datasets for training and testing are required. Table 1 gives details about the two different datasets used: the synthetic dataset for training and the real dataset for testing, described in Section 5.2. Benefiting from the efficient synthetic data generation, a synthetic dataset along with the annotation information is created within the digital system of the Digital Twin for model training purpose. With respect to the testing data, real datasets are collected from the physical system of the Digital Twin under the real working environment. With the assistance of the semi-automated annotation tool, the process of annotating the raw RGB data can be speed up for constructing the real dataset.

4.1. Semi-automated annotation tool

It is usually time-consuming and labor-intensive to annotate real data for each single image. Several commercial annotation tools are available, such as V7 [62] and Labelbox [63], supply AI functions to aid in data annotation. However, one major drawback is that these predefined AI models usually only work well in very limited scenarios, for example, detecting cars and humans for autonomous driving tasks. It cannot meet various demands of annotating specific objects such as the robot UR10, and is not applicable to diverse industrial scenes.

The deep learning model in this framework is working with a semi-automated annotation tool which we developed [31] based on Labelme [64]. The Digital Twin generates synthetic data and then the deep learning model is trained and tested using these data. Furthermore, the deep learning model is deployed with the annotation tool for acquisition and annotation of the real data from the physical system.

4.2. Real data

In order to validate this framework, a real dataset is acquired by a Kinect V2 sensor based on a UR10 platform and the dataset is publicly available on [30]. To simulate a real HRC scenario, three operators dressed in different clothes took part in the test whilst the Kinect V2 camera was mounted horizontally on the ceiling, looking down over the workspace. In this case, the field of view of the camera can capture one or two operators at the same time. Fig. 9 depicts that when a robot is working in a cell, an operator is moving into the cell and then interacting with the robot.

The real data was collected under various experimental conditions, by changing illumination levels and operators (humans). There are 4 different illumination levels and 2700 images were recorded respectively at each illumination level. Besides, 1653 images were saved with different operators. Totally this real dataset contains 12 453 images.

4.3. Synthetic data

The synthetic datasets include robot images that are generated using the proposed Digital Twin technique whilst operator data is gathered from the COCO database [28]. Fig. 10 shows people with different appearances and robot images that are fed into training a detector. With respect to the robot images generated from the digital system, Domain Randomization techniques such as different lighting conditions and different robot poses are applied during the data generation. To make the synthetic robot data looks similar to the physical system, the background of the synthetic data is captured from the physical system.

The reason for merging human samples with annotation information from COCO data [28] with the robot data is that COCO [28] is a public dataset for object detection research and has collected abundant human images. It brings the advantage that the detector can learn diverse human actions from the training data set to improve its generalization. The detector is also capable of detecting different operators with different appearance. This is irrespective of how many operators get into the robot cell since it has learnt enough human data during the training process. Consequently, it can be considered as an effective way

(a) (b) (c) (d)

Fig. 9. Images with annotation information in the real dataset. From (a) to (d), the whole process of human–robot collaboration is captured from the Kinect V2 sensor mounted on the top of the UR10.

(a) (b) (c) (d)

Fig. 10. Images with annotation information in the synthetic dataset. (a) and (b) shows human images from COCO database [28], while (c) and (d) are robot images generated from the digital system of the Digital Twin.

Table 1
Numbers of images in different datasets. To guarantee the real data that contains different light factors, the data is collected under different lighting conditions: full light, semi-light, semi-dark and dark where the lighting condition is changing from light to dark. With respect to the synthetic data, the data is generated without identifying lighting conditions.

Datasets	Full light	Semi-light	Semi-light	Full dark	Total
Real data	4861	2877	2977	3211	13,926
Synthetic data	–	–	–	–	20,823

to construct a training dataset for HRC scenario without extra data collection and annotation. This synthetic dataset is randomly split into two parts, including 20 823 images for training and 5206 images for validation.

The image database used in this research is shared online, including the real dataset as well as the synthetic dataset.

5. Performance evaluation and validation

5.1. Evaluation metrics

The performance of the proposed framework has been evaluated and validated over synthetic and real data under different lighting conditions.

The Average Precision (AP) [65] is adopted as the main evaluation metric, which is defined as

$$\mathrm{AP} = \int_0^1 p(r)dr, \tag{6}$$

where p denotes the precision function and r - the recall function [66, 67]

$$\begin{aligned} \text{Precision } p &= \frac{TP}{TP+FP}, \\ \text{Recall } r &= \frac{TP}{TP+FN}, \end{aligned} \tag{7}$$

where TP represents the true positive values, FP is the false negative and FN is the false negative [65].

The average precision (7) represents the area under the precision–recall curve. The average precision has a high value when both precision and recall are high, and it has a small value when either of precision or recall is small. While the average precision AP is calculated for each class, the mean average precision (mAP) is calculated by taking the average of average precision across all the considered classes.

The IoU is defined as follows [66,67]

$$\mathrm{IoU} = \frac{A \cap B}{A \cup B}, \tag{8}$$

where A is the predicted bounding box of an object and B is the corresponding ground-truth bounding box.

The mean AP (mAP), AP at the Intersection over Union (IoU) over 50% (AP50) and the AP at the IoU over 75% (AP75) [28] are used to evaluate the performance of the CNN trained over different datasets and under different lighting conditions.

5.2. Experiment setting

The Digital Twin is an excellent physical-virtual integrated system which can be used to study the impact of different environmental conditions, including the potential factors which may affect object detection, human action recognition and decision making. This Section 5.2 presents results over real and synthetic data with the faster R-CNN described in Section 3.3. Two faster R-CNN models are trained with different datasets: one is trained only with real data, the other is trained only with synthetic data. Then the performance of our semi-supervised model is also evaluated which is described in Section 3.4 which considers both the real data without the ground-truth and the synthetic data with the annotation. The teacher block within the semi-supervised model is firstly trained with the synthetic data and next the student model is trained with real data without the ground-truth. These models are trained on four Tesla V100 GPUs. The three models have been trained with the same strategy, with a stochastic gradient descent (SGD) algorithm.

For the distributed training, 16 samples per GPU are selected with a total of 64 batch size and the overall convergence of the stochastic gradient process takes up to 7 h. The model trained by real data takes less than a half an hour. linear warmup, a learning rate schedule, is applied for training with an initial learning rate of 0.08 and the learning rate rises linearly after 500 iterations. Together with the stochastic gradient, a technique called momentum is used. Instead of using only the gradient of the current step in the search, the momentum uses the gradient of the past steps to determine the next direction to move. A weight decay of 0.0005 and momentum of 0.9 are applied during the training process.

Table 2
Results under different lighting condition, mAP, AP50 and AP75 is utilized to evaluate three object detection models. Real represents the faster R-CNN model trained with the real data. Synthetic represents the faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.

	Full light			Semi-light			Semi-dark			Dark		
	mAP	AP50	AP75	mAP	AP50	AP75	mAP	AP50	AP75	mAP	AP50	AP75
Real	0.692	0.98	0.781	0.661	0.974	0.841	0.645	0.965	0.742	0.605	0.968	0.674
Synthetic	**0.789**	0.978	0.913	**0.773**	0.965	**0.930**	0.585	0.844	0.677	0.608	0.904	0.712
Semi-supervised	0.781	**0.993**	**0.924**	0.768	**0.989**	0.928	**0.679**	**0.966**	**0.804**	**0.701**	**0.972**	**0.817**

Table 3
mAP results at UR10 and Human under different lighting conditions. Real represents the faster R-CNN model trained with the real data. Synthetic represents the faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.

	Full light		Semi-light		Semi-dark		Dark	
	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}
Real	0.689	0.695	0.648	0.673	0.596	**0.694**	0.625	0.586
Synthetic	**0.864**	0.714	**0.835**	0.711	0.693	0.477	0.708	0.509
Semi-supervised	0.790	**0.773**	0.768	**0.768**	**0.700**	0.659	**0.792**	**0.611**

5.3. Performance evaluation of detection

Four lighting conditions are considered in the experiment for evaluating the three models. Two faster R-CNN models trained with two different datasets and our semi-supervised model are evaluated under different lighting conditions.

The first evaluation is within a steady manufacturing environment, where a robot repeats the same routine with pre-defined program in the robot cell. The detection algorithm can achieve accurate and steady results by learning from similar scenes to the robot cell, i.e., the training dataset should be diversified to cover as many scenes as those in the robot working routine. Several environmental factors in real manufacturing scenes may affect the performance of a deep learning-based detector negatively, such as image noise, illumination, unseen objects [68,69]. Among these factors in the robot cell, the room illumination has the greatest influence on the performance of the detection algorithm. The change of illumination may results from the sunlight or the lighting conditions of the factory which are unpredictable.

Table 2 shows that the semi-supervised solution achieves the best performance compared to those trained only with the real or synthetic data under four lighting conditions. From Tables 2 and 3, it is evident that when the lighting condition is becoming worse, the APs of the three models decline demonstrates that the lighting conditions is a critical factor that affects the performance of faster R-CNN. Compared to the model trained with the real data, the model trained with the synthetic data and the semi-supervised model have better performance when the lighting is sufficient (full light) which are roughly 10% better than the model trained with real data. Especially in good lighting conditions (full light and semi-light), both the model trained with synthetic data and the semi-supervised model achieves over 76% mAP.

With respect to AP50 and AP75, AP75 gives closer matching between the predicted bounding box and the ground-truth compared to the AP50 metric. From what AP50 and AP75 of these model in full light and semi-light is illustrated, the faster R-CNN trained with the synthetic data and the semi-supervised model are above 91%, while the faster R-CNN trained with the real data in the full light condition only has 78% AP75 in full light and 84% in semi-light. The performance of the model trained with the real data drops over 10% from AP50 to AP75, while the other two show smaller reduction in the average precision which means that the predicted bounding boxes of these models are more accurate and closer to the ground-truth bounding boxes.

However, when the lighting is insufficient (semi-dark and dark), the model trained on synthetic data shows a significant reduction in its performance. The APs of the semi-supervised model drop less compared to the model trained with synthetic data, when the lighting conditions change. The semi-supervised model also outperforms the faster R-CNN model trained with the real data. Hence, the semi-supervised solution is robust to changes in the lighting conditions.

Table 3 gives the results for the mAP of UR10 robot and human under different lighting conditions. The faster R-CNN trained with the synthetic and the semi-supervised model also achieves better performance than the network trained on real data with good lighting conditions (full light and semi-light). However, mAPs with respect to both UR10 and humans declines when the lighting is reduced.

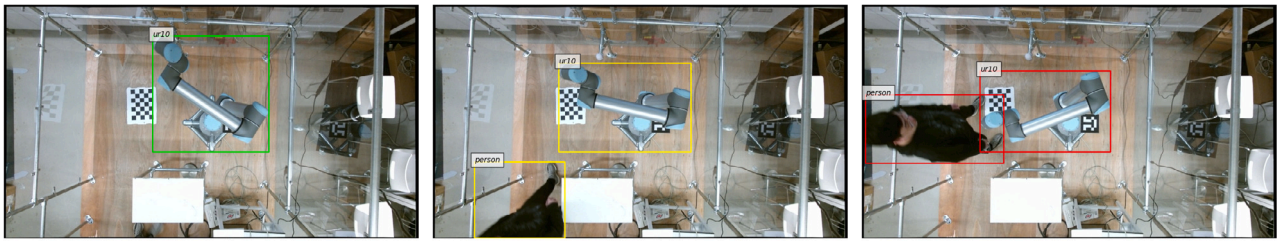
Even when the faster R-CNN is purely trained with the synthetic data, it achieves a remarkable mAP_{UR10} under full and semi-lighting conditions. The semi-supervised model shows more robust behavior when the lighting conditions are changing. Furthermore, with respect to mAP_{human}, the semi-supervised algorithm has the best score compared to those models that are only trained with the real or the synthetic data. The mAP_{human} is above 77% under full lighting and also achieves 61% under the dark situation.

5.4. Decision making for safe HRC

The detection algorithm is implemented on a laptop with Nvidia RTX 2070 GPU. When monitoring the robot and the operator in the physical system, it can achieve the detection speed at about 20 frames-per-second (fps), which meets the real-time monitoring requirement in this case. However, the cumulative time delay due to data transmission and model inference time may lead to negative effects on the monitoring a safe HRC. In the meanwhile, some detection failures cannot be ignored, even though it rarely happens.

To enhance the reliability and the safety of the HRC, three decision making criteria are defined to minimize the negative effects described above. A faster R-CNN detector has the capability to detect objects of interest and their locations in an image. Because the camera used to monitor the interaction between operators and robot is mounted on the top of a HRC cell, it provides a horizontal two-dimensional vision space [70]. With such a spacial relationship between camera frame and the world frame, the detection information (bounding boxes) can indicate how close between the operator and the robot and help to make a safe decision making. The safety decision making criteria can be defined as: (i) **Safe**: Only the robot is detected and no operator enters the robot cell, the robot moves at normal moving speed as shown in Fig. 11(a). (ii) **Potential**: In Fig. 11(b), the operator enters the robot cell and the bounding boxes of both the operator and the robot are detected and two bounding boxes are not overlapped. And the robot reduces its speed to the half of the original speed. (iii) **Dangerous**: If two bounding boxes are overlapped as shown in Fig. 11(c), it means the operator is quite close to the robot. Therefore, the robot should stop immediately to avoid collision with the operator.

With different speed settings, the detection algorithm can efficiently reduce the risk of the collision when the operator is getting close to the robot. The robot firstly can be aware of the presence of the operator, then the robot reduces its speed. When an overlap between



(a) Safe (b) Potential (c) Dangerous

Fig. 11. Three safety criteria for safety decision making.

the bounding boxes surrounding the human and the robot end-effector occurs, the robot stops immediately. This allows the operator to have enough reaction time to potential danger.

By calibrating the camera parameters, the camera is positioned at 3 meters height from the ground. The horizontal distance between the operators and the robot is about 20 cm when their bounding boxes are overlapping at the beginning. In our work, the human can keep a safe distance to the robot with the designed criteria based on the bounding box information. This is a different solution compared to the approach proposed by Liu and Wang [71] which is a collision-free HRC approach, requiring the position information for both the human and robot. The approach of Liu and Wang [71] requires extra sensor-robot coordinate calibration for the purpose of collision sensing which is not necessary in our case.

Inspired from [72], the Kalman filter and Hungarian matching method are used here to improve the reliability of the inference process. The state of each detection box is defined as $\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$, where (u, v) is the center of the bounding box in an image, s is the scale parameter and the r is the ratio of the height to width of the bounding box. The other variables $\dot{u}, \dot{v}, \dot{s}$ denote the respective speeds of the center coordinates and scale of the bounding box. When a bounding box is detected by the detector, it is applied to update its corresponding target state with the Kalman filter. The IoU distance between the detected and predicted box of an existing target that is tracked, is calculated. The assignment between the current and predicted box is performed by the Hungarian matching algorithm. To reduce the delays from the inference time of the detector, the frequency of detection is reduced to detect an image every 4 frames. It significantly improve the speed from 20 fps to 100 fps by sampling detection results when updating the states of the tracking with this post-processing.

A detector may fail to detect objects in some frames which may reduce the reliability of the monitoring process and could raise risks of danger in HRC. Thanks to the Kalman filter, the negative effects of such detection failures can be eliminated to a great extent. For the multi-object tracking problem [72], occlusions are also key factors that could reduces the quality of the tracking performance. Thanks to the monitoring camera mounted on the top of the robot cell, some occlusions can be avoided.

Although in [13], a similar deep learning approach is proposed, it applies the Mask R-CNN [73] to extract mask information. The mask information helps to reconstruct 3D relationship between the human and the robot in order to calculate the direct distance for safe decision making. Compared to [73], our inference speed outperforms the speed reported in [73]. Mask R-CNN [38] which is an extension version of faster R-CNN [29] requires extra computation cost to predict mask information. It would be difficult to achieve a real-time performance without extra post-processing, even though a real-time calculation is reported in [38]. In our case, the Kalman filter has improved the performance of action detection as well as the calculated speed. Consequently, the improved method leads to a robust solution and safe HRC.

5.5. Discussion and demonstration

From the analysis presented above, several advantages of the proposed framework are evident. First, it is easy to setup and deploy the proposed framework in manufacturing. Within the proposed Digital Twin, users can simply build a Digital Twin of the physical manufacturing workspace by introducing CAD models of real objects into the Digital Twin. The communication between the physical and digital systems can be established by the ROS.

Traditional deep learning application in manufacturing usually requires huge data collection and expensive manual annotation work. However, these can be avoided in our proposed framework by implementing efficient data generation and with semi-supervised method using the Sim2Real technique.

Besides, flexibility is another significant advantage of the proposed framework. This generative framework is not limited to detect humans and robot actions. It can also be extended to other objects by introducing new objects through adding their CAD models into the digital system. In the meanwhile, users also can specify annotation method to meet their requirements described in 3.2.1. Moreover, faster R-CNN can be replaced with another detection model within the semi-supervised method. The adoption of the efficient data transmission scheme between the digital and the physical systems, together with the automatic annotation generation, can allow users to implement other tasks, such as reinforcement learning [74] and Augmented Reality (AR) [75] in HRC.

This work also evaluates and discusses the effect of one key environment factor, lighting condition, on detection performance. Additionally, by introducing the Kalman filter and the Hungarian algorithm, the detector is enhanced to avoid detection failures whilst the inference speed is also improved. With these post-processing and decision making rules, the safety distance between the human and robot is maintained which enhances the reliability of the HRC environment.

Digital Twins combined with artificial intelligence have a huge potential to make a difference in smart manufacturing. This was also demonstrated in [76,77]. Moreover, the inclusion of cloud computing services in Digital Twins can lead to cyber–physical cloud manufacturing systems [78]. Digital Twin can be served as a platform for reinforcement learning training [75,79], and meanwhile, reinforcement learning is promising to lead the next generation of Digital Twins.

6. Conclusions and future work

This work explores the feasibility of a Digital Twin in smart manufacturing. It proposes a deep learning-enhanced Digital Twin for detecting and classifying human and robot actions for enhancing safety in manufacturing systems. A Digital Twin is designed for human–robot collaborations which generates synthetic data directly in the digital system. This helps with the generation of real data in the physical system with accurate annotation. The Digital Twin is an efficient tool for studying different levels of safety and to design decision making and control algorithms for manufacturing purposes.

The Robot Operating System is used to provide synchronous communication with, and real-time control of, the robot. The Digital Twin corresponding to the physical system is designed with the help of Domain Randomization and the powerful photorealistic Unreal Engine 4. Training of the developed deep learning algorithms is achieved successfully with synthetic data. A fully-supervised detection algorithm is shown to achieve successful detection results in the real environment. To ensure reliability of the system under different lighting conditions, a semi-supervised detector is proposed to take both synthetic and real data into the training and detection process, which helps in bridging the gap between the two systems in detecting humans and robots.

Future work will focus on more challenging cases with multiple robots and multiple operators. Apart from object detection, other tasks, such as gesture recognition and pose estimation will be considered in order to recognize both the actions of human operators and robots. This will enable more complex decision-making and control, boosting additional flexibility as well as enhancing the system resilience in complicated tasks.

CRediT authorship contribution statement

Shenglin Wang: Conceptualization, Methodology, Data collection, Software, Validation, Writing – original draft, Visualization. **Jingqiong Zhang:** Conceptualization, Methodology, Data collection & curation, Validation, Writing – refining, Visualization. **Peng Wang:** Conceptualization, Methodology, Data collection, Writing – refining. **James Law:** Funding acquisition, Project administration, Writing – review & editing. **Radu Calinescu:** Conceptualization, Funding acquisition, Writing – review & editing. **Lyudmila Mihaylova:** Conceptualization, Methodology, Supervision, Funding acquisition, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data from this evaluation have been made publicly available on Figshare ORDA. The semi-automated annotation tool is published on GitHub.

Acknowledgments

This work was supported by the Assuring Autonomy International Programme, a partnership between Lloyd's Register Foundation and the University of York; UK EPSRC project EP/V026747/1 (Trustworthy Autonomous Systems Node in Resilience); UK EPSRC [Grant No. EP/T013265/1, NSF-EPSRC: "ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven Systems"] and the USA National Science Foundation [Grant No. NSF ECCS 1903466]; the "Towards Turing 2.0" project under the EPSRC Grant EP/W037211/1 and the Alan Turing Institute; and Research England via the University of Sheffield's Internal Knowledge Exchange Scheme, "Towards Improved Safety and Reliability of Cobots" project. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

We are grateful to Nicholas Hall from the Health and Safety Executive (HSE) for his invaluable advice during different stages of this research, especially in the aspects of regulations, safety and standards for cobot systems. We also acknowledge the support from the RKE "Constructing accurate 3D human with deep learning in digital twin environments for safe human-robot collaboration" project under grant No. 914175. We are also grateful to the Associate Editor and Reviewers for the constructive suggestions helping us to improve this work.

References

[1] E. Magrini, F. Ferraguti, A.J. Ronga, F. Pini, A. De Luca, F. Leali, Human-robot coexistence and interaction in open industrial cells, Robot. Comput.-Integr. Manuf. 61 (2020) 101846, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584518303338.

[2] V. Villani, F. Pini, F. Leali, C. Secchi, Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications, Mechatronics 55 (2018) 248–266.

[3] A.A. Malik, A. Brem, Digital twins for collaborative robots: A case study in human-robot interaction, Robot. Comput.-Integr. Manuf. 68 (2021) 102092.

[4] X. Ma, F. Tao, M. Zhang, T. Wang, Y. Zuo, Digital twin enhanced human-machine interaction in product lifecycle, Proc. Cirp 83 (2019) 789–793.

[5] E. Matheson, R. Minto, E.G.G. Zampieri, M. Faccio, G. Rosati, Human–robot collaboration in manufacturing applications: A review, Robotics 8 (4) (2019) [Online]. Available: https://www.mdpi.com/2218-6581/8/4/100.

[6] F. Flacco, T. Kroeger, A. De Luca, O. Khatib, A depth space approach for evaluating distance to objects, J. Intell. Robot. Syst. 80 (1) (2015) 7–22.

[7] B. Schmidt, L. Wang, Depth camera based collision avoidance via active robot control, J. Manuf. Syst. 33 (4) (2014) 711–718, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612514000417.

[8] J.-H. Chen, K.-T. Song, Collision-free motion planning for human-robot collaborative safety under cartesian constraint, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 4348–4354.

[9] R.Y. Tsai, R.K. Lenz, et al., A new technique for fully autonomous and efficient 3D robotics hand/eye calibration, IEEE Trans. Robot. Autom. 5 (3) (1989) 345–358.

[10] R. Horaud, F. Dornaika, Hand-eye calibration, Int. J. Robot. Res. 14 (3) (1995) 195–210.

[11] C.Y. Siew, S.-K. Ong, A.Y. Nee, A practical augmented reality-assisted maintenance system framework for adaptive user support, Robot. Comput.-Integr. Manuf. 59 (2019) 115–129.

[12] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, J.-K. Kämäräinen, AR-based interaction for human-robot collaborative manufacturing, Robot. Comput.-Integr. Manuf. 63 (2020) 101891.

[13] S.H. Choi, K.-B. Park, D.H. Roh, J.Y. Lee, M. Mohammed, Y. Ghasemi, H. Jeong, An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation, Robot. Comput.-Integr. Manuf. 73 (2022) 102258.

[14] L. Alzubaidi, J. Zhang, A.J. Humaidi, A.Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J. Big Data 8 (1) (2021) 53, [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8.

[15] J. Fan, P. Zheng, S. Li, Vision-based holistic scene understanding towards proactive human–robot collaboration, Robot. Comput.-Integr. Manuf. 75 (2022) 102304.

[16] C. Cimino, E. Negri, L. Fumagalli, Review of digital twin applications in manufacturing, Comput. Ind. 113 (2019) 103130.

[17] Q. Qi, F. Tao, Y. Zuo, D. Zhao, Digital twin service towards smart manufacturing, Proc. CIRP 72 (2018) 237–242, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118302580, Proceedings of the 51st CIRP Conference on Manufacturing Systems.

[18] Y. Fu, G. Zhu, M. Zhu, F. Xuan, Digital twin for integration of design-manufacturing-maintenance: An overview, Chin. J. Mech. Eng. 35 (1) (2022) 1–20.

[19] F. Tao, H. Zhang, A. Liu, A.Y. Nee, Digital twin in industry: State-of-the-art, IEEE Trans. Ind. Inform. 15 (4) (2018) 2405–2415.

[20] M.G. Kapteyn, J.V. Pretorius, K.E. Willcox, A probabilistic graphical model foundation for enabling predictive digital twins at scale, Nat. Comput. Sci. 1 (5) (2021) 337–347.

[21] Z. Huang, Y. Shen, J. Li, M. Fey, C. Brecher, A survey on AI-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics, Sensors 21 (19) (2021) 6340.

[22] M.M. Rathore, S.A. Shah, D. Shukla, E. Bentafat, S. Bakiras, The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities, IEEE Access 9 (2021) 32030–32052.

[23] C. Zhang, G. Zhou, J. Li, F. Chang, K. Ding, D. Ma, A multi-access edge computing enabled framework for the construction of a knowledge-sharing intelligent machine tool swarm in Industry 4.0, J. Manuf. Syst. 66 (2023) 56–70.

[24] C. Zhang, G. Zhou, Q. Xu, Z. Wei, C. Han, Z. Wang, A digital twin defined autonomous milling process towards the online optimal control of milling deformation for thin-walled parts, Int. J. Adv. Manuf. Technol. 124 (7–8) (2023) 2847–2861.

[25] S. Honig, T. Oron-Gilad, Understanding and resolving failures in human-robot interaction: Literature review and model development, Front. Psychol. 9 (2018) [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2018.00861.

[26] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, A. Nee, Enabling technologies and tools for digital twin, J. Manuf. Syst. 58 (2021) 3–21, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861251930086X, Digital Twin towards Smart Manufacturing and Industry 4.0.

[27] Stanford Artificial Intelligence Laboratory et al., Robotic operating system, 2018, [Online]. Available: https://www.ros.org.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[29] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.

[30] J. Zhang, S. Wang, P. Wang, L. Mihaylova, J. Law, A vision data repository for human-UR10 robot interactions in manufacturing, 2022, http://dx.doi.org/10.15131/shef.data.16669315.v1, https://doi.org/10.15131/shef.data.16669315.v1.

[31] S. Wang, J. Zhang, P. Wang, L. Mihaylova, Semi-automated labelme, A deep learning based annotation tool, 2022, http://dx.doi.org/10.5281/zenodo.6393953, [Online]. Available: https://github.com/wongsinglam/Semi_Labelme, On Github.

[32] J. Friederich, D.P. Francis, S. Lazarova-Molnar, N. Mohamed, A framework for data-driven digital twins for smart manufacturing, Comput. Ind. 136 (2022) 103586.

[33] K. Dröder, P. Bobka, T. Germann, F. Gabriel, F. Dietrich, A machine learning-enhanced digital twin approach for human-robot-collaboration, Proc. Cirp 76 (2018) 187–192.

[34] Y. Ghasemi, H. Jeong, S.H. Choi, K.-B. Park, J.Y. Lee, Deep learning-based object detection in augmented reality: A systematic review, Comput. Ind. 139 (2022) 103661, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522000586.

[35] K.-B. Park, S.H. Choi, J.Y. Lee, Y. Ghasemi, M. Mohammed, H. Jeong, Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality, IEEE Access 9 (2021) 55448–55464.

[36] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[37] R. Girshick, Fast r-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[39] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[40] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E.D. Cubuk, A. Kurakin, H. Zhang, C. Raffel, FixMatch: Simplifying semi-supervised learning with consistency and confidence, 2020, arXiv:2001.07685 [Cs, Stat] [Online]. Available: http://arxiv.org/abs/2001.07685.

[41] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, T. Pfister, A simple semi-supervised learning framework for object detection, 2020, arXiv preprint arXiv:2005.04757.

[42] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, Z. Liu, End-to-end semi-supervised object detection with soft teacher, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021, pp. 3040–3049, [Online]. Available: https://ieeexplore.ieee.org/document/9710144/.

[43] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J. Big Data 8 (2021).

[44] W. Zhao, J.P. Queralta, T. Westerlund, Sim-to-real transfer in deep reinforcement learning for robotics: a survey, 2020, arXiv preprint arXiv:2009.13303.

[45] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Proceedings of European Conference on Computer Vision, ECCV, in: LNCS, vol. 9906, Springer International Publishing, 2016, pp. 102–118.

[46] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Proceedings of Conference on Robot Learning, PMLR, 2017, pp. 1–16.

[47] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 23–30.

[48] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, S. Birchfield, Training deep networks with synthetic data: Bridging the reality gap by domain randomization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 969–977.

[49] F. Sadeghi, S. Levine, CAD2rl: Real single-image flight without a single real image, 2016, arXiv preprint arXiv:1611.04201.

[50] B. Mehta, M. Diaz, F. Golemo, C.J. Pal, L. Paull, Active domain randomization, in: Proceedings of Conference on Robot Learning, PMLR, 2020, pp. 1162–1176.

[51] ISO/TS 15066:2016 Robots and robotic devices — Collaborative robots, Vol. 2000, Standard, International Organization for Standardization, Geneva, CH, 2016.

[52] ISO 10218-1:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots, Vol. 2011, Standard, International Organization for Standardization, Geneva, CH, 2011.

[53] C. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. Rivero, J. Manzo, E. Krotkov, G. Pratt, Inside the virtual robotics challenge: Simulating real-time robotic disaster response, IEEE Trans. Autom. Sci. Eng. 12 (2) (2015) 494–506.

[54] E. Rohmer, S.P.N. Singh, M. Freese, Coppeliasim (formerly V-REP): a versatile and scalable robot simulation framework, in: Proceedings of the International Conference on Intelligent Robots and Systems, IROS, 2013, www.coppeliarobotics.com.

[55] W. Qiu, A. Yuille, Unrealcv: Connecting computer vision to unreal engine, in: Proceedings of European Conference on Computer Vision, Springer, 2016, pp. 909–916.

[56] E. Games, Unreal engine, 2019, [Online]. Available: https://www.unrealengine.com.

[57] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, D. Lange, Unity: A general platform for intelligent agents, 2018, arXiv preprint arXiv:1809.02627.

[58] M. Woo, J. Neider, T. Davis, D. Shreiner, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2, Addison-Wesley Longman Publishing Co., Inc., 1999.

[59] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, in: Proceedings of 18th International Conference on Pattern Recognition (ICPR'06), Vol. 3, IEEE, 2006, pp. 850–855.

[60] ISO 20218-2:2017 Robotics — Safety design for industrial robot systems — Part 2: Manual load/unload stations, Vol. 2017, Standard, International Organization for Standardization, Geneva, CH, 2017.

[61] IEC/TS 61496-4-3:2015 Safety of machinery - Electro-sensitive protective equipment - Part 4-3: Particular requirements for equipment using vision based protective devices (VBPD) - Additional requirements when using stereo vision techniques (VBPDST), Vol. 2015, Standard, The British Standards Institution, London, UK, 2015.

[62] AI data platform for Automated Annotation, [Online]. Available: https://www.v7labs.com/.

[63] The leading training data platform for data labeling, [Online]. Available: https://labelbox.com/.

[64] K. Wada, Labelme: Image polygonal annotation with Python, [Online]. Available: https://github.com/wkentaro/labelme.

[65] M. Everingham, S.M.A. Eslami, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The Pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136.

[66] R. Padilla, S.L. Netto, E.A.B. da Silva, A survey on performance metrics for object-detection algorithms, in: Proceedings of the International Conference on Systems, Signals and Image Processing, IWSSIP, 2020, pp. 237–242.

[67] R. Padilla, W.L. Passos, T.L.B. Dias, S.L. Netto, E.A.B. da Silva, A comparative analysis of object detection metrics with a companion open-source toolkit, Electronics 10 (3) (2021) [Online]. Available: https://www.mdpi.com/2079-9292/10/3/279.

[68] J. Gawlikowski, C.R.N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., A survey of uncertainty in deep neural networks, 2021, arXiv preprint arXiv:2107.03342.

[69] T. Han, Y.-F. Li, Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles, Reliab. Eng. Syst. Saf. 226 (2022) 108648.

[70] P. Wang, S. Wang, J. Law, L. Mihaylova, 2.6.1 – Monitoring RAS operation, 2021, https://www.york.ac.uk/assuring-autonomy/body-of-knowledge/implementation/2-6/2-6-1/cobots/, Accessed: 2021-08-03.

[71] H. Liu, L. Wang, Collision-free human-robot collaboration based on context awareness, Robot. Comput.-Integr. Manuf. 67 (2021) 101997.

[72] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: Proceedings of the IEEE International Conference on Image Processing, ICIP, IEEE, 2016, pp. 3464–3468.

[73] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[74] M. Kaspar, J.D.M. Osorio, J. Bock, Sim2real transfer for reinforcement learning without dynamics randomization, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 4383–4388.

[75] C. Li, P. Zheng, S. Li, Y. Pang, C.K. Lee, AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop, Robot. Comput.-Integr. Manuf. 76 (2022) 102321.

[76] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, A. Calinescu, Digital twins: State of the art theory and practice, challenges, and open research questions, J. Ind. Inf. Integr. 30 (2022) 100383, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X22000516.

[77] Y. Wang, X. Kang, Z. Chen, A survey of digital twin techniques in smart manufacturing and management of energy applications, Green Energy Intell. Transp. 1 (2) (2022) 100014, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2773153722000147.

[78] L. Hu, N.-T. Nguyen, W. Tao, M.C. Leu, X.F. Liu, M.R. Shahriar, S.M.N. Al Sunny, Modeling of cloud-based digital twins for smart manufacturing with MT connect, Procedia Manuf. 26 (2018) 1193–1203, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235197891830831X, 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.

[79] Y. Liu, H. Xu, D. Liu, L. Wang, A digital twin-based sim-to-real transfer for deep reinforcement learning-enabled industrial robot grasping, Robot. Comput.-Integr. Manuf. 78 (2022) 102365.