Sensor-based Human-Robot Collaboration for Industrial Tasks

Alexandre Angleraud[a], Akif Ekrekli[a], Kulunu Samarawickrama[a], Gaurang Sharma[a] and Roel Pieters[a,*]

[a] *Unit of Automation Technology and Mechanical Engineering, Tampere University, Korkeakoulunkatu 6, Tampere, Finland*

ARTICLE INFO

Keywords:
Human-robot collaboration
Visual perception
Deep learning

ABSTRACT

Collaboration between human and robot requires interaction modalities that suit the context of the shared tasks and the environment in which it takes place. While an industrial environment can be tailored to favor certain conditions (e.g., lighting), some limitations cannot so easily be addressed (e.g., noise, dirt). In addition, operators are typically continuously active and cannot spare long time instances away from their tasks engaging with physical user interfaces. Sensor-based approaches that recognize humans and their actions to interact with a robot have therefor great potential. This work demonstrates how human-robot collaboration can be supported by visual perception models, for the detection of objects, targets, humans and their actions. For each model we present details with respect to the required data, the training of a model and its inference on real images. Moreover, we provide all developments for the integration of the models to an industrially relevant use case, in terms of software for training data generation and human-robot collaboration experiments. These are available open-source in the OpenDR toolkit at `https://github.com/opendr-eu/opendr`. Results are discussed in terms of performance and robustness of the models, and their limitations. Although the results are promising, learning-based models are not trivial to apply to new situations or tasks. Therefore, we discuss the challenges identified, when integrating them into an industrially relevant environment.

1. Introduction

Collaborative robots (co-bots) can improve the safety, work efficiency and productivity of industrial processes by acting as flexible and reconfigurable tool to human operators. Within Industry 4.0, co-bots have a core role to contribute to the transition from traditional manufacturing to digital manufacturing [3, 13]. Co-bots can be easily programmed and reconfigured, and are safe for interaction, due to their small form-factor and incorporated sensor systems that can detect collisions [57]. Co-bots are also to be found in high-payload form, where protective covering can be complemented by sensor-based safety features. Human-robot collaboration (HRC) is typically possible in two ways [62]: 1. Off-line programming of robot tasks by demonstration (also known as hand-guiding or kinesthetic teaching), and 2. On-line interaction between human and robot, enabled by external sensor systems. While off-line programming is an established method of collaboration, on-line interaction still typically requires great efforts in development and its success depends highly on the sensor system. That is, if the external sensor system is not robust or has high latency, this reflects negatively on the performance of the collaboration.

Nevertheless, the role of humans and industrial robots in smart factories is often emphasized [13] and future roadmaps state clear benefits on utilizing collaboration between humans and robots [57]. The practical requirements and tools needed, however, are often underestimated or given little attention, resulting in great interest from industry and SMEs, but not many practical implementations [62]. To be realistic, successful integration of perception tools in human-robot collaboration requires considerable effort towards the selection of suitable detection tools, the preparation of suitable data for training and the actual training of a detection model, followed by its implementation in the robotic system. In this work we address these issues, and present the following contributions:

1. Identification of challenges for deep learning-based visual perception in HRC
2. Practical integration details for three deep learning-based visual perception tools in HRC
3. Open-source software templates for sensor-based HRC
4. Validation of the sensor-based HRC framework with an industrial use case

The problems we aim to address in this work are the current limitations in perception models and situational awareness for industrial human-robot collaboration. Perception and situational awareness of robot systems can be enhanced, such that fluent and responsive collaboration between human and robot is possible. We believe that perception models, based on deep learning, are ideal for this, as they can be accurate, reliable and fast to execute. These can then provide the required sensory input for interaction, such as the human body and its pose, human actions or gestures, and the pose of objects and targets in the scene. Developing and integrating such models for robotics in industry are hard tasks, often requiring expertise from many different areas [49]. Therefore, we additionally provide a general HRC software framework, based on ROS [40], which can be utilized to replicate our developments. The framework is build around OpenDR [38], a deep learning toolkit for robotics, and has the perception tools integrated for a practical and industrially relevant use case in agile production. The visual perception tools are human skeleton detection, human action recognition and the

*Corresponding author
✉ `roel.pieters@tuni.fi` (R. Pieters)

detection and pose estimation of objects and targets in the scene.

In the following section, the current challenges of perception for HRC are identified, when considering deployment in industrial environments.

1.1. Challenges for sensor-based HRC

The first two identified challenges relate to typical and well-known issues of learning-based perception [29], i.e., perception model selection and training data collection. The last two identified challenges relate to the applicaton and integration of such models to an industrial environment.

1. **Model selection and training -** The choice of perception model depends mostly on what needs to be detected. Many well-performing models exist, e.g. for common objects households objects [26] or humans [34, 56]. However, simply selecting the model with the highest accuracy is usually not the best approach. For example, a model that detects humans in an automotive scenario would not perform well in industrial scenario. All relevant context and properties of the model needs to be considered, as it will affect the performance with respect to the intended use case. Moreover, properties such as model size and inference time are of practical importance for human-robot collaboration where delay and responsiveness of the interaction matter greatly.

2. **Data collection -** The performance of a detection model is directly influenced by the quality and quantity of the data used for training. Data and its annotation need to include enough variability that could occur in the real use case, without enlarging the dataset unnecessarily. While in certain areas large datasets exist (e.g., household objects [23]), in other cases the dataset needs to be collected or generated from scratch. Collecting real data is usually preferred, as it captures the realistic content of the target object as well as the sensor, however, synthetic data has also shown suitable performance in many cases [36]. One additional problem for data collection is the annotation of the data with the ground truth, for example, object classes or 6D object poses. For real data, annotation is difficult and time-consuming, and in some cases near impossible (e.g., object poses). In this case simulation and the generation of synthetic data has the benefit of knowing exactly where an object is rendered in the virtual world [50].

3. **Reliability and safety -** Deep neural networks (DNN) are known as black-box models, implying that their inner workings cannot (easily) be understood [5]. Explainable AI aims to provide explanations to models, even though there is no general consensus of what is meant by explainable and/or interpretable [20]. In case of safety-critical applications (e.g. autonomous driving or human-robot collaboration), DNN cannot provide required reliability and safety levels [18]. Moreover, model performance, failure probability and their uncertainty are difficult to determine and can drift during long-term operation. While continual learning might prove useful in this regard, developments are still in early stages [31].

4. **Integration -** Deploying DNNs to a real environment requires integration efforts that depend on the model and its intended outcome. Clear differences can be identified between models that provide input for on-line decision making and models that provide diagnostics for off-line monitoring [54]. For example, in manufacturing environments, the detection of obstacles and humans needs to provide timely input to machinery for halting processes. As such, the operating equipment needs to be shut down and tested extensively to ensure reliable working of the developed tools [45]. Predictive maintenance, on the other hand, only provides recommendations and does not interfere with running processes. Data collection and installation of models can, therefore, often be done while machinery is in operation or without rigorous testing protocols [59]. One additional challenge is the availability of state-of-the-art DNN tools. While most developments are open-source available and can even be commercialized, there is no guarantee for code-quality and its maintenance [21]. Support for the software is typically not offered by the tool developers, and tools quickly become obsolete due to, for example, general software updates. As industrial systems are operational for extended time periods (years), investment in upgrading is not a regular occurrence.

These identified challenges are broad research topics, and cannot be tackled by individual research efforts, but require community effort to push boundaries forward. We therefore do not claim in this work that we provide a solution to these challenges but offer directions in the specific area of human-robot collaboration how the challenges can be taken into account. The remainder of this paper is organized as follows. In Section 2 we provide an overview of related work in human-robot collaboration and relevant perception tools. As a result of this overview, several perception tools are selected for implementation and explained in further detail in Section 3. Section 4 describes the industrial assembly use case, the software framework as well as integration details needed to replicate the research developments. The results of the perception modules and the human-robot collaboration experiments are presented and discussed in Section 5. Finally, Section 6 concludes the work.

2. Related work

2.1. Human-robot collaboration

Collaboration between human and robot has been an ongoing trend since the advent of smart manufacturing [13] and Industry 4.0 [57]. Formal definitions of collaboration, working zones and operating modes are common [53] and standards provide requirements and design guidelines to ensure safety for operators. [55] provides an overview of symbiotic human-robot collaborative assembly and highlights future research directions. Methods presented include voice processing, gesture recognition, haptic interaction, and even brainwave perception. In most cases deep learning is used for classification, recognition and context awareness identification. Computer vision-based approaches are the most popular, as presented in [14]. This reports a systematic review of computer vision-based holistic scene understanding in HRC scenarios, which mainly takes into account the cognition of object, human, and environment. Subsequently, visual reasoning can be used to gather and compile visual information into semantic knowledge for robot decision-making and proactive collaboration. Other overviews of human-robot collaboration approaches can be easily found, for example, towards the topics of robotic vision [43] and machine learning [45], indicating the popularity of the topics, either individually, or combined. Proactive collaboration between human and robot is highlighted in [22], with emphasis on cognitive, predictable and self-organizing perspectives. Current challenges are found, which call for future research direction that address real-world applications.

2.2. Human detection

The detection of humans, individual body parts and their actions based on visual information has been a long-standing problem in computer vision [34].

Human presence detection - Detecting the presence of a person in the robot work space has been an active area of research, mainly to ensure safety of the human [63]. Different visual modalities can be used to detect humans [24]. In [35], a depth sensor is utilized, producing data in the form of a point cloud. From this, a convex hull of the human point cloud is created and background removal detects any moving objects/subjects in the scene. Similar is the work in [15], where a depth map is utilized to detect a person's presence, but also to allow interaction with a projected graphical user interface. A dynamically updated workspace model is, therefore, required. Depth cameras are also used in [28] for the detection of a person in the work space and to compute their distance to the robot. In addition, laser scanners at leg-level are included to detect an operator's presence. It is noted that both sensing systems work in parallel and do not fuse information together, allowing a redundancy for safety. 3D LiDAR-based detection of humans is presented in [61], which utilizes a learning-based approach for human classification. The work, however, targets large indoor public spaces and a mobile service robot. In [24] a comparison is made between the performance of state-of-the-art person detectors for 2D range data, 3D LiDAR, and RGB-D data, as well as selected combinations thereof, in a challenging industrial use case. Multi-modal approaches have also gained interest [39], however, most works only consider larger environments for mobile robots (or cars) [19], making their suitability for small and dense industrial environments questionable. Human pose estimation goes beyond human detection by estimating 3D poses of humans and their individual skeleton joints. Well-known approaches are OpenPose [6] and VoxelPose [51], which can utilize single as well as multiple cameras.

Gesture detection - Detection and recognition of human gestures has also been of interest to robotics. In [25], a comprehensive review is given of different gesture recognition approaches for human-robot collaboration. Besides visual perception, the review also includes non-image based approaches, such as wearables. [33] demonstrates real-time human-robot interaction with robust background invariant hand gesture detection. The approach presents a method to collect a training dataset for static hand gestures, taken from letters and numbers from American sign language.

Human action recognition - As an extension to the detection of humans and their gestures, the methods of human action recognition consider the behavior of a person, i.e., their actions or motions, to be detected [56, 48]. This implies an image sequence to be used for recognition, as compared to single images in e.g., human detection. Recent progress has been achieved by deep learning approaches that take as input an image sequence in RGB-D format, extracts the 2D or 3D skeleton pose and performs action classification [60]. In relation to human-robot collaboration, research on action recognition has also focused on industrial activities [10, 8] and pose forecasting [44], including actions such as picking, placing, assembling, polishing, etc.

2.3. Object detection and pose estimation

State of the art deep neural networks have shown impressive performance for generic object categories [26]. Real-time object detection is an active research problem to allow adoption to robotics applications, and many works can be found that have utilized detectors for tasks such as robot grasping [11]. Popular approaches are for example, Faster R-CNN [42], Yolo [41] and SSD [27]. Pose estimation of objects considers to estimate the 6D pose of an object. Similar to object detection, different approaches exist, such as correspondence-based methods 3DMatch [64], template-based methods such as PoseCNN [7] and voting based methods such as DenseFusion [55]. For both object detection and pose estimation, datasets can be found, for example, Pascal VOC [12] and COCO [23] for 2D object detection, and, more recently, Objectron [1] and T-LESS [16] for 3D objects and 6D pose estimation. It is important to mention a crucial difference between these methods of object detection and pose estimation, as compared to human detection and pose estimation. In general, most human perception approaches are successful with a large variety in humans. That means existing dataset are sufficient to be used in new areas with new humans. In contrast, most object perception approaches do not scale well to novel objects and additional data should be generated to train a model and achieve successful detection. In this work, results were achieved in a similar manner.

2.4. Other interaction modalities

Speech - Utilizing speech as interaction modality has the benefit of not requiring physical actions for the human, allowing work-related tasks to be uninterrupted [32]. As a research field, the maturity has increased significantly recently, due to advancements of speech recognition technologies, with respect to recognition performance and robustness against noise [52]. However, despite the maturity in speech recognition performance, the connection of speech commands to robot actions and/or higher-level goals requires internal representations that need to be developed as well [30]. For tasks that are low in complexity (e.g., pick-and-place, hand-overs) such knowledge representation is manageable [4], but with increasing conversational capabilities in natural language perception, knowledge representation requires careful and extensive modelling.

***Graphical user interface* -** The most common modality for programming industrial robots is a graphical user interface (GUI) [53]. Robot tasks and motions can be achieved by either robot hand-guiding and a teaching pendant, or by low-level programming with suitable programming language and software toolbox. In both cases a GUI is utilized to assist in the programming and/or teaching of robot tasks. GUIs are typically developed with ease-of-use in mind and, recently, user perceptions such as user experience, user effort and understanding are actively taken into account as well [9]. As a graphical tool, GUIs offer great capabilities, such as visualization and simulation, integrated as part of the robot programming stage. Limitations, however, have been identified as well, such as a higher cognitive burden needed for end-users [2]. While GUIs are beneficial for the programming of robots, they are not well suited for interaction during task execution. Human-robot collaboration requires responsiveness of the robot to human cues, which is difficult to achieve with a GUI alone.

2.5. Comparison to our approach

From this brief overview of related work, a few observations can be made. Most perception tools are developed and presented without robotics in mind, aiming for general target groups (see Section 2.2-2.3). This implies that specific characteristics relevant for human-robot collaboration in industrial environments are not included or tested, making their suitability for this questionable. For example, manufacturing environments can be dirty and noisy, and specific conditions, such as lighting, can be difficult to adjust, in contrast to laboratory and domestic environments. In addition, while the adoption of perception tools is often possible by open-source software, details on integration are usually limited to just the tool itself [33] and not to a robotics framework [35] (e.g., ROS). This is also found in other works, where different perception tools are reviewed to detail the state of the art, e.g. for robotic vision [43] and machine learning [45]. What these works do not cover is the challenges and issues faced with respect to data collection and the practical integration of the tools to a robot. While [14] and [22] do include challenges, these are not related to technical integration. Our work aims to fill this gap, by focusing on three different visual perception tools. We provide details on how to replicate our work, from dataset generation and training tools, to code examples (Python, ROS) for individual perception tools and as an integrated use case with a collaborative robot. These are available open-source in the OpenDR toolkit[1].

3. Visual recognition modules

All three integrated visual recognition modules utilize color images for perception. Depth perception was intentionally excluded such that models can run at high update rate, ideally in real-time (i.e., 20 FPS or higher). Especially for the detection of a person and their gestures this is needed to have a responsive system with short delay time.

3.1. Human skeleton detection

Method - Detection of a human in the scene is done with OpenPose [6], a real-time multi-person human pose detector. OpenPose is capable of detecting up to a total of 135 human body, foot, hand, and facial key points, from a single or multiple image/camera sources. The lightweight version of OpenPose is selected [37], as it achieves detections in realtime. For a successful detected human pose the method returns a list 18 2D image key points of the human skeleton with associated key point abbreviation.

Data generation and model training - The method in this work utilizes the pretrained MobileNet model as explained in [37], which was trained and evaluated with the COCO 2017 dataset [23] under default training parameters.

3.2. Human action recognition

Method - Recognition of human actions is done with ST-GCN [60], a real-time skeleton-based human action recognition framework, as it can utilize the lightweight OpenPose model [37]. The method takes the location of the human joints in every image, and generates a sequence of detected human skeleton graphs, connected both spatially and temporally. Depending on the dataset the method can detect a large number of different human actions, ranging from daily activities to complex actions with interactions.

Data generation and model training - The smallest training dataset is selected (NTU-RGB+D [46]), as it contains the most relevant human action classes (60 classes in 56,000 human action clips). For each image human skeleton joints are annotated in 3D, with respect to the camera coordinate system. The pretrained model from the original authors, with default training parameters, is used for inference.

3.3. Assembly object and target detection

Method - Mask R-CNN from Detectron2 [58] was selected for object and target detection in the scene, as performance was preferred over inference time. Mask R-CNN combines a Region Proposal Network (RPN) with the CNN model, to simultaneously predict object bounds and objectness scores at each position. After detection, orientations are estimated in each bounding box by the second order moment from a segmented object or target.

Data generation and model training - As the assembly objects and targets are novel with respect to existing datasets, a custom dataset needed to be generated. For this, 200 images of eight object and target classes were annotated with segmentation polygons, as depicted in Fig. 1. The object classes included rocker arms, bolts and pushrods, and the target classes included the Diesel engine, small and big

[1] https://github.com/opendr-eu/opendr

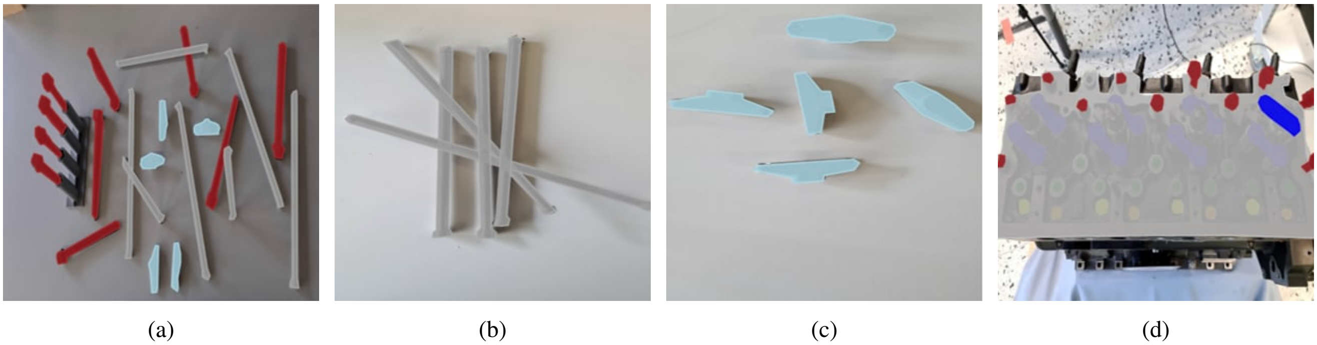


Figure 1: Image annotations for assembly objects, including bolts (red), pushrods (grey) and rocker arms (light blue); (a), (b) and (c), and targets objects, including Diesel engine (grey), small (yellow) and big (orange) pushrod holes, bolt holes (green) and rocker arm locations (dark blue); (d). Annotations are done with segmentation polygons in different colors, for different object classes. A total of 200 images with eight object and target classes were utilized for augmentation and dataset generation.

pushrod holes, bolt holes and rocker arm locations. This data was augmented to include a broad variation in noise and lighting conditions, to form the custom dataset of around 280,000 images [47]. The methods for data generation and annotation are available in the OpenDR toolkit[1].

4. Industrial assembly use case

4.1. Diesel engine assembly

The manufacturing of Diesel engines involves assembly steps that are hard to automate, such as contact placement and manipulation of parts with various degrees of freedom. For example, rocker arm placement, push rod insertion and bolt fastening all have different constraints with respect to the final manipulation of the part to the engine. Rocker arms can be moved freely in 3D task space before placements, push rod insertion requires vertical motion into a pushrod hole and bolt fastening requires rotational motion and compliance orthogonal to vertical motion. In addition, parts to assemble are complex in shape, metallic and require lubricant for assembly and for operation. This means traditional robotic operations for picking and placing are not suitable for assembly and manual actions are the standard approach for manufacturing. A promising alternative, however, is to utilize the robot as assistant and assign tasks to it that support the assembly procedure and the ergonomy of the human operator. These are easy, but repetitive tasks, such as pick and placement, and actions for operator assistance such as hand-overs of parts and tools.

The scenario for human-robot collaboration is depicted in Fig. 2 and includes the Diesel engine, a table with parts and tools, the human operator and a collaborative robot. To demonstrate and validate our developments, we constructed a use case in which the robot picks and places parts from the table to the engine (push rods) and hands-over parts from the table to the operator (rocker arms and bolts). Visual perception is used as input to robot actions (object and target detection) and for human task coordination (human skeleton detection and human action recognition).

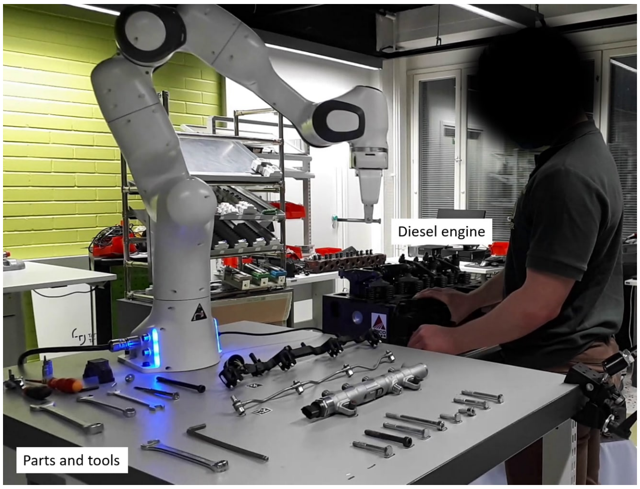


Figure 2: Experimental setup with a collaborative robot (Franka Emika), Diesel engine and parts for assembly tasks.

4.2. Integration

All developments are integrated in the OpenDR[1] toolkit [38] with ROS/ROS2[2] nodes of the perception tools and ROS moveit2[3] scripts for the human-robot collaboration scenarios. A description of the use case, and the individual perception modules, has been documented[4], enabling to easily replicate (and extend) our work. For robot and perception hardware, we utilize the Franka Emika collaborative robot[5] and two Intel Realsense D435 cameras, one on the end-effector of the robot and one front-facing to the person for human perception. Computations are done on a Ubuntu PC with Nvidia GTX 1080 Ti GPU, running ROS Noetic.

A Python script example of a visual recognition module is shown in Listing 1, demonstrating its usage. Here, a pretrained model for Detectron2 is loaded and the model

[2] https://www.ros.org/

[3] https://moveit.picknik.ai/

[4] https://trinityrobotics.eu/use-cases/sensor-based-human-robot-collaboration/

[5] https://franka.de/

inference is run on an input image. The prediction results of the model are drawn as boxes on the image as well. It should be noted that other tools of the OpenDR toolkit, i.e., human skeleton detection, human action recognition, as well as other perception tools, datasets and trained models, can be utilized in a similar manner [38]. For example, in the case of object and target detection, a custom dataset was generated, as explained briefly in Section 3.3. This included image annotation and augmentation, with the open-source tools Label Studio[6] and Albumentations[7], respectively. These are also integrated into the OpenDR perception tools, in form of Python scripts and Jupyter notebooks[8].

Listing 1: Object and target detections script in OpenDR[1]

```
from opendr.engine.data import Image
from opendr.perception.object_detection_2d import
     ↪ Detectron2Learner

# load model and run inference on image
detectron2 = Detectron2Learner(device=”cpu”)
detectron2.download(”.”, mode=”pretrained”)
detectron2.load(”./detectron2_default”)
img = Image.open(”input_image.jpg”)
predictions = detectron2.infer(img)

# draw bounding boxes of predictions on image
boxes = BoundingBoxList([box for kp,box in predictions])
draw_bounding_boxes(img.opencv(), boxes, class_names=
     ↪ detectron2.classes, show=True)
```

A python script example of robot actions is shown in Listing 2, demonstrating how to define a pick and place task with several concatenated actions. These low-level actions are based on Moveit2[3] and therefore robot-agnostic. In the example, motions are defined in task space as 2D planar motion parallel to the table (`2D_action`) and 1D motion vertical to the table (`1D_action`), to perform grasping. Additional actions include end-effector rotations (`rotate_EE`) and gripper actions (`move_gripper`) and can take input from visual modules, as shown by the inclusion of `object` and `place`.

Listing 2: Robot actions script in OpenDR[1]

```
def Pick_and_Place(object,place):
    # Move and align robot above object
    2D_action(pose=[object.x, object.y], slow=False)
    rotate_EE(angle=object.angle)
    # Move robot down and grasp object
    1D_action(z_pose=0.35, slow=True)
    move_gripper(speed=20.0, width=0.02)
    # Move robot to place and release object
    1D_action(z_pose=0.2, slow=True)
    2D_action(pose=[place.x, place.y], slow=False)
    1D_action(z_pose=0.35, slow=True)
    move_gripper(speed=20.0, width=0.08)
```

A python script example for human-robot collaboration is shown in Listing 3, demonstrating how to combine the visual recognition modules and the robot actions. In the example, whenever a visual recognition module publishes a message, i.e., when a successful detection is made, a callback function is called with successive robot actions. This can therefore be used for human coordination of the assembly process, by triggering, halting and/or resuming robot actions.

Listing 3: Human-robot collaboration script in OpenDR[1]

```
def AR_callback(AR_data):
    if AR_data.id == 37 and AR_data.score > 0.80:
    # Stop robot motion when 'salute' is detected
        stopAction()
    elif AR_data.id == 39 and AR_data.score > 0.80:
    # Continue when 'cross hands in front' is detected
        continueAction()

def OD_callback(OD_detections):
    # Get bolt and bolt_hole pose
    bolt_id = detections.find_object(”bolt”)
    bolt_pose = detections.get_pose(bolt_id)
    bolt_hole_id = detections.find_object(”bolt_hole”)
    bolt_hole_pose = detections.get_pose(bolt_hole_id)
    # Call pick and place action
    Pick_and_Place(bolt_pose,bolt_hole_pose)

if __name__ == '__main__':
    # subscribe to action_recognition topic
    rospy.Subscriber(”/opendr/action_recognition”,
         ↪ ObjectHypothesis, AR_callback)
    # subscribe to object_detection topic
    rospy.Subscriber(”/opendr/object_detection”,
         ↪ ObjectHypothesisWithPose, OD_callback)

    rospy.spin()
```

5. Results and Discussion

Results are described for each individual visual recognition module and for the utilization of the modules in human-robot collaboration experiments. Integration, limitations and future work are described in the discussion as well.

5.1. Visual recognition performance

Table 1 and 2 provides details of the different perception modules, their corresponding datasets for training and inference, and their prediction accuracy results. In the case of human skeleton detection and human action recognition, pre-generated datasets were utilized, as these provided sufficient performance for detection. A disadvantage, however, is that the datasets cannot be easily extended by adding additional data and/or classes. We explain this and other practical limitations in more detail for each recognition module.

[6] https://labelstud.io/

[7] https://albumentations.ai/

[8] https://jupyter.org/

Table 1
Perception models and datasets utilized to enable human-robot collaboration. Performance is reported in terms of frames per second (FPS) and prediction accuracy on custom test data, recorded for evaluation.

	Training			**Inference (GTX 1080 Ti)**			
Perception module	**Method**	**Dataset**	**Dataset size**	**Model size**	**Image size**	**FPS**	**Prediction accuracy (%)**
Human skeleton detection	Lightweight OpenPose [37]	COCO 2017 [23]	25 GB	1.2 GB	1920x1080 1280x720 960x540	30 30 60	91
Human action recognition	ST-GCN [60]	NTU-RGB+D [46]	1.3 TB	47 MB	1920x1080 1280x720 960x540	20 30 31	87
Object and target detection	Detectron2 [58]	Custom [47]	65 GB	0.5 GB	1920x1080 1280x720 960x540	2.6 4.5 6.0	93

Table 2
Confusion matrix of the object and target detection tool, evaluated on 2340 images with 39530 instances of the 8 classes. Actual classes are shown as column heads and predicted classes as row heads. The prediction accuracy is shown as last column.

Classes	**Rockerarm target**	**Bolt hole**	**Big pushrod hole**	**Small pushrod hole**	**Engine**	**Bolt**	**Pushrod**	**Rockerarm object**	**Background**	**Prediction accuracy**
Rockerarm target	**4802**	0	0	0	0	0	0	0	13	**99.7**
Bolt hole	0	**12398**	0	1	0	0	0	0	157	**98.7**
Big pushrod hole	0	0	**2234**	47	0	0	0	0	193	**90.3**
Small pushrod hole	0	11	28	**2453**	0	0	0	0	196	**91.3**
Engine	0	0	0	0	**995**	0	0	0	0	**100**
Bolt	0	0	0	2	0	**6451**	24	63	504	**91.6**
Pushrod	0	0	0	1	0	269	**2579**	47	517	**75.6**
Rockerarm object	0	0	0	0	0	0	0	**3817**	25	**99.3**

Human skeleton detection

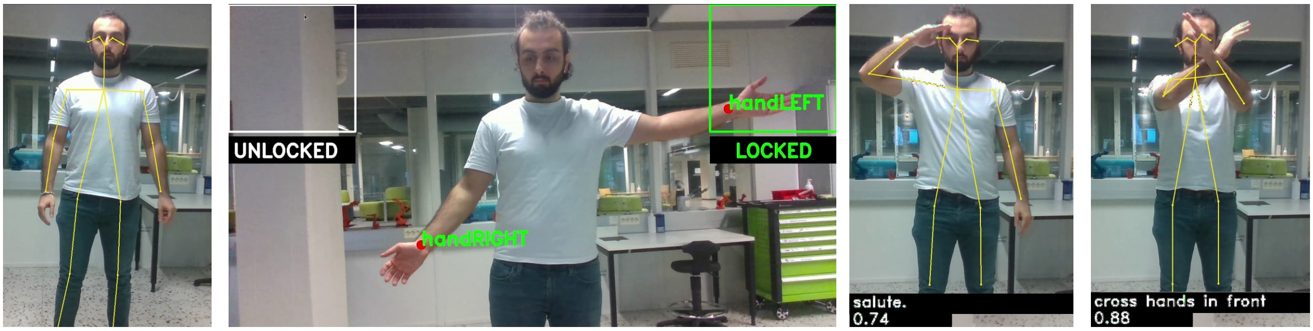
The human skeleton detection method (LightWeight OpenPose [37]) with pretrained model [23] is evaluated on a custom test dataset of 1950 images, in which a person performs different actions in the field of view. Human actions included are similar to actions to be recognized in the human action recognition tool. The prediction accuracy of a human skeleton detected correctly, such that it performs human action recognition, was found to be 91%. Fig. 3 depicts the skeleton detection and draws it over the person in the scene. In terms of computational performance, the module achieves 30 frames per second, for high resolution camera image input (1920 × 1080) and even higher for lower resolution images (see Table 1).

The industrial environment and the scenario of engine assembly leaves practical limitations on how the human skeleton detection tool can be utilized. For example, the camera cannot capture the human in full, but only the upper body. For human-robot collaborative tasks the detection of a person's left and right wrist was therefore chosen for the interaction, as these could be detected reliably, while allowing free motion in the entire camera view. The detection of both wrists in predefined areas in the image can then be utilized to trigger robot actions, and to halt and resume them. Requiring both detections simultaneously in both areas increased the robustness to false positive detection with a single wrist, when the person was doing assembly actions on the engine. A sequence of screenshots of human skeleton and wrist detection can be seen in Fig. 3 and Fig. 5.

Human action recognition

The human action recognition method (ST-GCN [60]) with pretrained model [46] is evaluated on a custom test dataset of 1950 images, in which a person performs different actions in the field of view, i.e., '*salute*' (ID:37), '*put the palms together*' (ID:38) and '*cross hands in front*' (ID:39). Each action was performed for 30 seconds, leading to >600 images per action. Recognition results were evaluated manually afterwards. Results indicate that a reasonably high prediction accuracy can be achieved (89%, 81% and 91% for the three actions, respectively).

Fig. 3c and 3d depict actions recognized and their confidence score printed on the image. As the action recognition tool utilizes skeleton detection, this is drawn over the image as well. In terms of computational performance, the module achieves 20 frames per second, for high resolution camera image input (1920 × 1080) and even higher for lower resolution images (see Table 1). Similar to human skeleton detection, the industrial scenario imposed limitations as datasets for human action recognition mostly cover daily actions [46], not relevant for industrial tasks.



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

Figure 3: Human visual recognition modules. (a) and (b) depict results of human skeleton detection with the skeleton-based tracker Lightweight OpenPose [37]. (b) demonstrates that skeleton detection can be used for human-robot collaboration by detecting human wrists (handLeft and handRight) in certain image areas. (c) and (d) depict results of human action recognition with the real-time skeleton-based human action recognition framework ST-GCN [60]. Recognized actions are '*salute*' (c) and '*cross hands in front*' (d), with their corresponding confidence score.

Object and target detection

The object and target detection method (Detectron2 [58]) with custom trained model achieves satisfactory performance, for non-overlapping objects. Fig. 4 depicts the objects detected on the table (a) and the targets detected on the engine (b). To create the dataset [47], 200 images of the eight objects and targets, in various configurations, were recorded and all objects and targets in the images were annotated with segmentation polygons in their correct class. Distractor objects, such as Diesel fuel lines, common rails and other tools, were included, as would be expected in a real scene. This data was then expanded with augmentations to a full datatset of around 280,000 images. Training of the model was done until convergence of the loss function (sum of losses due to classification and bounding box regression), which took around 20,000 epochs. With this method, the trained model achieved detection confidences for real camera images of more than 90%. While more data could be added and more training could be done, results are sufficient to perform reliable experiments for picking and placing, and human-robot collaboration. In terms of computational performance, the module cannot run in real-time, but achieves 2.6 frames per second for high resolution camera input (1920 × 1080). As the objects and targets are static in the scene, real-time performance is not required. The implemented object and target detection tool enables both continuous detection (images are processed consecutively) and detection requests from a single image, with a function call. In the human-robot collaboration scenario a detection request is utilized to save computational performance of the GPU machine. It is expected, though, that both approaches would work equally well in terms of object pick and placement performance.

5.2. Human-robot collaboration

The visual perception modules were utilized to enable human-robot collaboration, in several different ways, with the detection modules utilized as interaction tool. Certain tools are more suited to specific tasks, due to their detection or computational performance. For example, human skeleton detection is very reliable and fast, while human action recognition is less reliable and slower. This time performance difference is due to the fact that human action recognition relies on the human skeleton detection as input and requires a considerable number of detected frames (300) for successful recognition. In practise this means that human action recognition has more false detections as well. The following experiments were tested in detail.

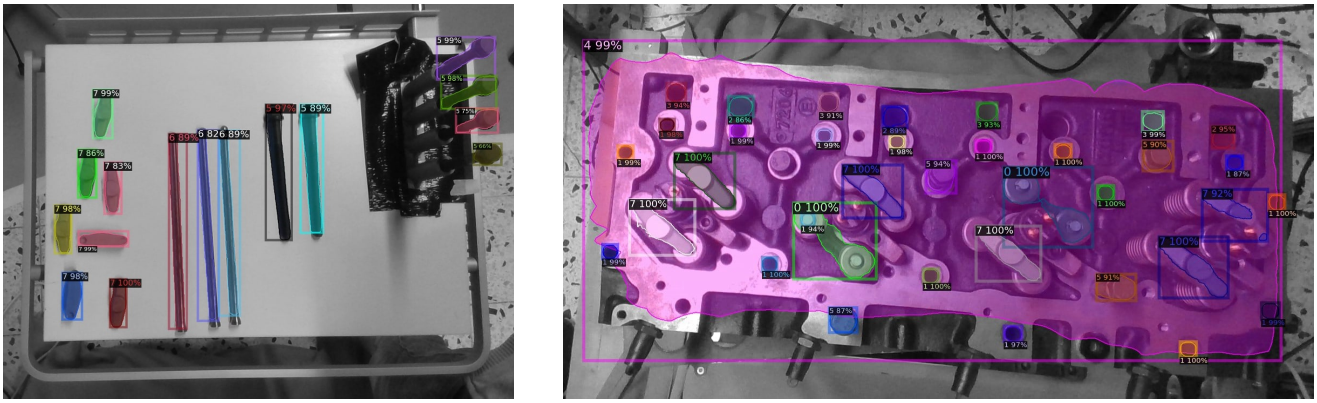
Human task coordination

The shared assembly task can easily be coordinated by the human with visual perception. Human skeleton detection (i.e., wrists in certain location) or human actions can be used for starting and/or stopping robot actions, thereby setting the pace for the assembly task and performing corrective actions, in case a robot has misplaced a part. Human visual perception is not required to have high performance for this, as the detection tools can be run at a high rate (i.e., >30 FPS). This implies that few false negative detections have no significant negative impact in the collaboration. For the object and target detection tool, real-time performance is not required either, as pick and place actions are called on request. These coordination experiments, by human wrist detection, are depicted in Fig. 5 and in the recorded video[9]. Robot actions are the assembly (pick and placement) of pushrods and bolts (six in total) to the Diesel engine and human actions are the placement of rocker arms, after their hand-over from the robot.

Robot-human hand-overs

As explained in Section 4, certain tasks for assembling a Diesel engine are too difficult for a robot to execute. However, as assistive tool, the robot can hand-over parts located on a table to the person executing complex assembly tasks. This is demonstrated in Fig. 6a and Fig. 6b, as well

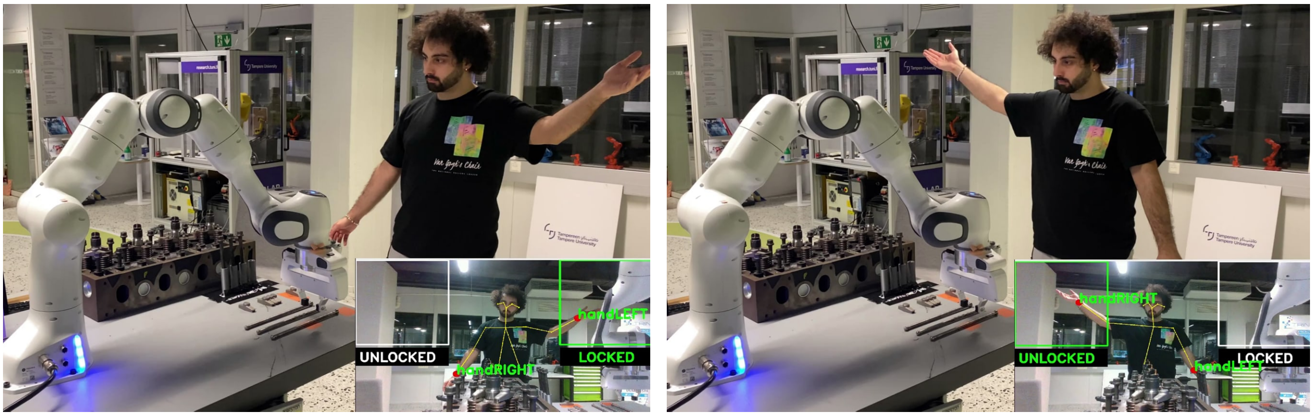
[9] https://youtu.be/3z3yiLdznrY



(a)

(b)

Figure 4: Results of visual perception for object and target detection utilizes Detectron2 [58]. (a) depicts detection of objects (three classes): rocker arms, bolts and pushrods, and (b) depicts detection of targets (five classes): engine, bolt holes, pushrod holes and rocker arm location. Each detection is labeled with the detected class and their corresponding confidence score.



(a)

(b)

Figure 5: Results of human-robot collaboration experiments. (a) and (b) depict human task coordination by visual detection of the left wrist (handLeft), for halting the robot and performing manual assembly actions (a), followed by right wrist detection (handRight) for resuming robot actions (b).

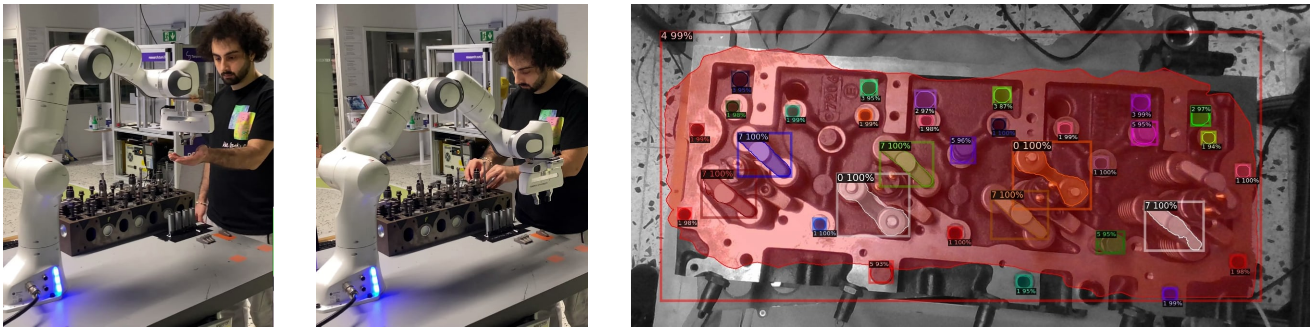
as in the recorded video[9], for the assembly tasks of rocker arm placement. The objects are detected with the same detection model and all detected parts are handed over in sequence to a hand-over point, close to the human. By human gestures (visual perception tools) the person can request for the initiation of the hand-over task (i.e., pick an object and move to the hand-over location) and trigger the actual hand-over action. After the rocker arm is handed over, the human can continue the assembly action, while the robot fetches another part.

In theory, human-robot collaboration by human coordination can improve the fluency of collaboration fluency measures [17]. This implies the reduction of idle time for both human and robot, as well as the robot's functional delay, leading to higher task efficiency. While this work serves to demonstrate the functionality of the visual perception modules, a thorough analysis and evaluation for fluency measures has not been carried out.

5.3. Additional features

Besides their respective detection output and enabling human-robot collaboration, the visual perception tools can also be used for additional, higher-level features. For example, perception tools can provide information relevant to the shared task and its progress, such as keeping track of objects in the scene and whether they are assembled or not, or determining the time durations of (individual) assembly steps. To demonstrate this, an application was developed that tracks the progress of the Diesel engine assembly task, by detecting which and how many objects are placed in the correct location and which objects are not placed yet.

While there are many ways how this could be implemented, a simple but effective implementation was done as follows. As the entire engine block is detected as well, it can be easily checked whether certain assembly objects (rocker arms and bolts) are detected inside the detected engine bounding box. For this, the image dataset included the images of assembly objects assembled on the engine.



(a)

(b)

(c)

Figure 6: Results of robot-human hand-over and assembly tracking experiments. (a) depicts the hand-over of a rocker arm from robot to human. (b) depicts the human assembly action of the rocker arm by the human, while the robot fetches another rocker arm. (c) depicts the assembly tracking results, with several objects (rocker arms, class 7; bolts, class 5) and their locations (rocker arm location, class 0; bolt holes, class 1) detected inside the detected Diesel engine bounding box (class 4). Each detection is labeled with the detected class and their corresponding confidence score.

Output of the assembly progress tracking tool then returns the number of objects assembled and/or whether the task is completed or not. Fig. 6c depicts the detection of different objects (rocker arms and bolts) inside the detected Diesel engine bounding box. In this time instance, six of the eight rocker arms are placed, however, only five are detected (class 7), while two rocker arm locations are detected (class 0). This means one detection is missing for either a rocker arm or rocker arm location. In addition, four of the 22 bolts are placed and detected (class 5), while sixteen bolt holes are detected (class 1) and thus empty. In total, 22 bolts should be assembled to the engine block, meaning two detections are missing for either bolts or bolt holes. It can then be concluded that assembly progress is around 5/8 or 62% and 4/22 or 18% for the rocker arms and bolts, respectively.

This example of assembly progress tracking demonstrates how additional features can be integrated by utilizing the perception tools. However, it does reveal some challenges that need to be addressed to provide a robust solution. In particular as shown by the example, missing or false detections might provide inaccurate estimates for tracking assembly progress and human supervision is required to verify correct assembly steps.

5.4. Discussion

Limitations - The first limitation of the explored perception modules relates to the relevance of the (training) data for industrial context. As most tools are developed for humans and objects in domestic or outdoor environments, success in other areas is not guaranteed. In certain cases this is not a major issues (e.g., humans look similar in a broad context), but in some cases it can be a problem, as classes are unsuitable (e.g., multi-human actions in a single human use case) or simply do not exist (e.g., novel objects or human actions to detect). One obvious solution to this would be to extend an existing dataset or create a new dataset from scratch, however, this is not a trivial task [29], [49]. Collecting data is complex, and expensive in resources and equipment, even when synthetic data generation approaches exist [36] [50]. In this work, the data generation tools for object and target detection are open-source available through the OpenDR toolkit.

Utilizing perception tools for human safety, in particular by DNN-based visual perception models, is not recommended. The reaction time of a safety system, in order to stop robot motion, should be small, which cannot always be guaranteed. Some models used in this work can be executed in real-time (see Table 1), and even faster (60 FPS), meaning that it takes at least 17*ms* for a detection, assuming a prediction is accurately made. Other models are simply not suited for fast detection or recognition, as they require a set of images, instead of single images (e.g., 300 in the case of [60]) and/or rely on another detection tool as input (e.g., skeleton detection in the case of [60]). In addition, as reported in [18], quantifying the reliability of machine learning and DNN-based perception tools is still a challenge and performance might drift over time. The time-delay of perception and its performance uncertainty should then be taken into account when calculating the minimum separation distance between human and robot [53, 63].

Hardware limitations concern the computation hardware and the visual sensors utilized. Naturally, a GPU similar to the ours (Nvidia GTX 1080 Ti) needs to used to achieve the same performance as reported in Table 1. However, the toolkit is compatible for both GPU and CPU systems to train and run all models, limiting only the run-time performance. Placement of the visual sensors is challenging to accommodate due to the different moving parts in the scene, i.e., robot and human. In our case, the visual sensors were placed on the robot end-effector and behind the robot facing to the person. This led to situations were objects are either not in the camera's field of view or humans are occluded by the robot, limiting the time that suitable perception can occur. While different solutions can be developed that would better distribute cameras or avoid occlusion [43, 62], our camera setup did not cause limitations in performance or drawbacks in fluency of collaboration, as demonstrated in the recorded video[9].

Integration effort - The resources and effort needed to develop, train and deploy perception models for industrial use, is considerable. Even when robust and reliable pretrained models are to be integrated, still effort is needed to comply tools to existing software frameworks with its own datatypes and formatting. While ROS[2] has taken first steps to enable this for robotics, computer vision tools are typically disconnected from this. OpenDR [38] has made efforts to integrate a variety of perception models into ROS, and examples to specific use cases are presented in this work. In the case when pretrained models are not sufficient, additional effort is needed for data collection and training. As it is difficult to estimate how much effort is needed for different models, we report the effort for our custom dataset for object and target detection [47]. A collection of 200 RGB images where taken as base for the dataset and annotations were needed for eight object and target classes. This annotation took considerable time (2-3 days) for the relatively small set of images. Generation of the complete dataset and training a model is time-consuming as well (2 hours for a single training cycle on a Nvidia GTX 1080 Ti GPU), and optimizing to good results requires expertise. Naturally, better performance can be obtained with more powerful computational hardware (e.g., computing cluster or cloud computing), however, these are not always available, and come with additional cost.

Future work - The results of our work demonstrate that deep learning-based perception models can be easily trained and deployed to robotic environments and achieve reliable detection and recognition results. Results also demonstrated that multiple perception models can be utilized simultaneously, enabling the fusion of different sensors or utilizing different detection modules in parallel. As such, this work has established a baseline for future directions. These include the fusion of different sensor information, from similar or dissimilar modalities. This sensor fusion would enable a higher robustness then single sensor models and introduces a redundancy of sensing, for example, in case one sensor fails or is occluded. Exploration of these topics will be done as future work.

6. Conclusions

Visual perception is a common tool for enabling human-robot collaboration, by detection or recognition of relevant objects, features and actions in the scene. The performance and maturity of such tools are usually evaluated by scenarios not related to robotics or manufacturing, limiting their direct utilization in industrial environments. Moreover, in some cases visual perception tools need to be tailored to suit the context of the human-robot collaboration scenario. This means collecting, annotating and augmenting visual data and the training of a perception model.

In this work we have identified these common issues and provide the practical integration details for three different deep learning-based visual perception tools. These are human skeleton detection, human action recognition, and object and target detection in context of the industrial use case of Diesel engine assembly. The tools are integrated open-source in the OpenDR toolkit, with ROS as software platform, providing templates for perception, robot actions and human-robot collaboration, thereby enabling to easily replicate and extend our work.

Declaration of competing interest

There is no known conflict of interest.

Acknowledgements

Project funding was received from European Union's Horizon 2020 research and innovation programme, grant no. 871449 (OpenDR) and no. 825196 (TRINITY).

References

[1] Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., Grundmann, M., 2021. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7822–7831.

[2] Ajaykumar, G., Steele, M., Huang, C.M., 2022. A survey on end-user robot programming. ACM Computing Surveys 54, 1–36. doi:10.1145/3466819.

[3] Alćer, V., Cruz-Machado, V., 2019. Scanning the industry 4.0: A literature review on technologies for manufacturing systems. Engineering Science and Technology, an International Journal 22, 899–919. doi:10.1016/j.jestch.2019.01.006.

[4] Angleraud, A., Sefat, A.M., Netzev, M., Pieters, R., 2021. Coordinating shared tasks in human-robot collaboration by commands. Frontiers in Robotics and AI 8. doi:10.3389/frobt.2021.734548.

[5] Arrieta, A.B., Díaz-Rodríguez, N., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115. doi:10.1016/j.inffus.2019.12.012.

[6] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A., 2019. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[7] Capellen, C., Schwarz, M., Behnke, S., 2019. ConvPoseCNN: Dense convolutional 6D object pose estimation. arXiv preprint arXiv:1912.07333 .

[8] Chen, C., Wang, T., Li, D., Hong, J., 2020. Repetitive assembly action recognition based on object detection and pose estimation. Journal of Manufacturing Systems 55, 325–333. doi:10.1016/j.jmsy.2020.04.018.

[9] Chowdhury, A., Ahtinen, A., Pieters, R., Vaananen, K., 2020. User experience goals for designing industrial human-cobot collaboration, in: Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society, ACM. pp. 1–13. doi:10.1145/3419249.3420161.

[10] Dallel, M., Havard, V., Baudry, D., Savatier, X., 2020. Inhard - industrial human action recognition dataset in the context of industrial collaborative robotics, in: IEEE International Conference on Human-Machine Systems (ICHMS), pp. 1–6. doi:10.1109/ICHMS49158.2020.9209531.

[11] Du, G., Wang, K., Lian, S., Zhao, K., 2021. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review 54, 1677–1734. doi:10.1007/s10462-020-09888-5.

[12] Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88, 303–338. doi:10.1007/s11263-009-0275-4.

[13] Evjemo, L.D., Gjerstad, T., Grøtli, E.I., Sziebig, G., 2020. Trends in smart manufacturing: Role of humans and industrial robots in smart factories. Current Robotics Reports 1, 35–41. doi:10.1007/s43154-020-00006-5.

[14] Fan, J., Zheng, P., Li, S., 2022. Vision-based holistic scene understanding towards proactive human–robot collaboration. Robotics and Computer-Integrated Manufacturing 75, 102304. doi:10.1016/j.rcim.2021.102304.

[15] Hietanen, A., Changizi, A., Lanz, M., Kamarainen, J., Ganguly, P., Pieters, R., Latokartano, J., 2019. Proof of concept of a projection-based safety system for human-robot collaborative engine assembly, in: IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 1–7. doi:10.1109/RO-MAN46459.2019.8956446.

[16] Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X., 2017. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects, in: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 880–888. doi:10.1109/WACV.2017.103.

[17] Hoffman, G., 2019. Evaluating fluency in human–robot collaboration. IEEE Transactions on Human-Machine Systems 49, 209–218.

[18] Jourdan, N., Sen, S., Husom, E.J., Garcia-Ceja, E., Biegel, T., Metternich, J., 2021. On the reliability of machine learning applications in manufacturing environments. arXiv preprint arXiv:2112.06986 .

[19] Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L., 2018. Joint 3D proposal generation and object detection from view aggregation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1–8. doi:10.1109/IROS.2018.8594049.

[20] Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., Baum, K., 2021. What do we want from explainable artificial intelligence (XAI)? – a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. Artificial Intelligence 296, 103473. doi:10.1016/j.artint.2021.103473.

[21] Lavin, A., Gilligan-Lee, C.M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., Lange, D., Baydin, A.G., Sharma, A., Gibson, A., Zheng, S., Xing, E.P., Mattmann, C., Parr, J., Gal, Y., 2022. Technology readiness levels for machine learning systems. Nature Communications 13, 6039. doi:10.1038/s41467-022-33128-9.

[22] Li, S., Zheng, P., Liu, S., Wang, Z., Wang, X.V., Zheng, L., Wang, L., 2023. Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives. Robotics and Computer-Integrated Manufacturing 81, 102510. doi:10.1016/j.rcim.2022.102510.

[23] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context, in: European conference on computer vision (ECCV), pp. 740–755.

[24] Linder, T., Vaskevicius, N., Schirmer, R., Arras, K.O., 2021. Cross-modal analysis of human detection for robotics: An industrial case study, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 971–978. doi:10.1109/IROS51168.2021.9636158.

[25] Liu, H., Wang, L., 2018. Gesture recognition for human-robot collaboration: A review. International Journal of Industrial Ergonomics 68, 355–367. doi:10.1016/j.ergon.2017.02.004.

[26] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. International Journal of Computer Vision 128, 261–318. doi:10.1007/s11263-019-01247-4.

[27] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single shot multibox detector, in: European conference on computer vision, pp. 21–37.

[28] Magrini, E., Ferraguti, F., Ronga, A.J., Pini, F., Luca, A.D., Leali, F., 2020. Human-robot coexistence and interaction in open industrial cells. Robotics and Computer-Integrated Manufacturing 61, 101846. doi:10.1016/j.rcim.2019.101846.

[29] Marcus, G., 2018. Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631 .

[30] Marge, M., Espy-Wilson, C., Ward, N.G., Alwan, A., Artzi, Y., Bansal, M., Blankenship, G., Chai, J., Daumé, H., Dey, D., Harper, M., Howard, T., Kennington, C., Kruijff-Korbayová, I., Manocha, D., Matuszek, C., Mead, R., Mooney, R., Moore, R.K., Ostendorf, M., Pon-Barry, H., Rudnicky, A.I., Scheutz, M., Amant, R.S., Sun, T., Tellex, S., Traum, D., Yu, Z., 2022. Spoken language interaction with robots: Recommendations for future research. Computer Speech & Language 71, 101255. doi:10.1016/j.csl.2021.101255.

[31] Maschler, B., Pham, T.T.H., Weyrich, M., 2021. Regularization-based continual learning for anomaly detection in discrete manufacturing. Procedia CIRP 104, 452–457. doi:10.1016/j.procir.2021.11.076.

[32] Mavridis, N., 2015. A review of verbal and non-verbal human–robot interactive communication. Robotics and Autonomous Systems 63, 22–35. doi:10.1016/j.robot.2014.09.031.

[33] Mazhar, O., Navarro, B., Ramdani, S., Passama, R., Cherubini, A., 2019. A real-time human-robot interaction framework with robust background invariant hand gesture detection. Robotics and Computer-Integrated Manufacturing 60, 34–48. doi:10.1016/j.rcim.2019.05.008.

[34] Nguyen, D.T., Li, W., Ogunbona, P.O., 2016. Human detection from images and videos: A survey. Pattern Recognition 51, 148–175. doi:10.1016/j.patcog.2015.08.027.

[35] Nikolakis, N., Maratos, V., Makris, S., 2019. A cyber physical system (cps) approach for safe human-robot collaboration in a shared workplace. Robotics and Computer-Integrated Manufacturing 56, 233–243. doi:10.1016/j.rcim.2018.10.003.

[36] Nowruzi, F.E., Kapoor, P., Kolhatkar, D., Hassanat, F.A., Laganiere, R., Rebut, J., 2019. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. arXiv preprint arXiv:1907.07061 .

[37] Osokin, D., 2018. Real-time 2D multi-person pose estimation on cpu: Lightweight openpose. doi:10.48550/ARXIV.1811.12004.

[38] Passalis, N., Pedrazzi, S., Babuska, R., Burgard, W., Dias, D., Ferro, F., Gabbouj, M., Green, O., Iosifidis, A., Kayacan, E., Kober, J., Michel, O., Nikolaidis, N., Nousi, P., Pieters, R., Tzelepi, M., Valada, A., Tefas, A., 2022. OpenDR: An open toolkit for enabling high performance, low footprint deep learning for robotics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12479–12484. doi:10.1109/IROS47612.2022.9981703.

[39] Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J., 2018. Frustum pointnets for 3D object detection from RGB-D data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[40] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al., 2009. ROS: an open-source robot operating system, in: ICRA workshop on open source software, Kobe, Japan. p. 5.

[41] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[42] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks, in: International Conference on Neural Information Processing Systems, p. 91–99.

[43] Robinson, N., Tidd, B., Campbell, D., Kulić, D., Corke, P., 2022. Robotic vision for human-robot interaction and collaboration: A survey and systematic review. ACM Journal of Human-Robot Interaction 12, 1–66. doi:10.1145/3570731.

[44] Sampieri, A., D'Amely, G., Avogaro, A., Cunico, F., Skenderi, G., Setti, F., Cristani, M., Galasso, F., 2022. Pose forecasting in industrial

human-robot collaboration. arXiv preprint arXiv:2208.07308 .

[45] Semeraro, F., Griffiths, A., Cangelosi, A., 2023. Human–robot collaboration and machine learning: A systematic review of recent research. Robotics and Computer-Integrated Manufacturing 79, 102432. doi:10.1016/j.rcim.2022.102432.

[46] Shahroudy, A., Liu, J., Ng, T.T., Wang, G., 2016. NTU RGB+D: A large scale dataset for 3D human activity analysis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1010–1019.

[47] Sharma, G., Pieters, R., Angleraud, A., 2023. Engine assembly dataset. doi:10.5281/zenodo.7669593.

[48] Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J., 2022. Human action recognition from various data modalities: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–20doi:10.1109/TPAMI.2022.3183112.

[49] Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M., Corke, P., 2018. The limits and potentials of deep learning for robotics. The International Journal of Robotics Research 37, 405–420. doi:10.1177/0278364918770733.

[50] Thalhammer, S., Patten, T., Vincze, M., 2019. SyDPose: Object detection and pose estimation in cluttered real-world depth images trained using only synthetic data, in: International Conference on 3D Vision (3DV), IEEE. pp. 106–115. doi:10.1109/3DV.2019.00021.

[51] Tu, H., Wang, C., Zeng, W., 2020. Voxelpose: Towards multi-camera 3D human pose estimation in wild environment, in: European Conference on Computer Vision (ECCV), pp. 197–212.

[52] Vargas, A.M., Cominelli, L., Dell'Orletta, F., Scilingo, E.P., 2021. Verbal communication in robotics: A study on salient terms, research fields and trends in the last decades based on a computational linguistic analysis. Frontiers in Computer Science 2. doi:10.3389/fcomp.2020.591164.

[53] Villani, V., Pini, F., Leali, F., Secchi, C., 2018. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics 55, 248–266. doi:10.1016/j.mechatronics.2018.02.009.

[54] Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018a. Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems 48, 144–156. doi:10.1016/j.jmsy.2018.01.003.

[55] Wang, L., Gao, R., Váncza, J., Krüger, J., Wang, X., Makris, S., Chryssolouris, G., 2019. Symbiotic human-robot collaborative assembly. CIRP Annals 68, 701–726. doi:10.1016/j.cirp.2019.05.002.

[56] Wang, P., Li, W., Ogunbona, P., Wan, J., Escalera, S., 2018b. RGB-D-based human motion recognition with deep learning: A survey. Computer Vision and Image Understanding 171, 118–139. doi:10.1016/j.cviu.2018.04.007.

[57] Weiss, A., Wortmeier, A.K., Kubicek, B., 2021. Cobots in Industry 4.0: A roadmap for future practice studies on human–robot collaboration. IEEE Transactions on Human-Machine Systems 51, 335–345. doi:10.1109/THMS.2021.3092684.

[58] Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R., 2019. Detectron2. https://github.com/facebookresearch/detectron2.

[59] Wuest, T., Weimer, D., Irgens, C., Thoben, K.D., 2016. Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research 4, 23–45. doi:10.1080/21693277.2016.1192517.

[60] Yan, S., Xiong, Y., Lin, D., 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Thirty-second AAAI conference on artificial intelligence.

[61] Yan, Z., Duckett, T., Bellotto, N., 2020. Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods. Autonomous Robots 44, 147–164. doi:10.1007/s10514-019-09883-y.

[62] Yang, C., Zhu, Y., Chen, Y., 2022. A review of human–machine cooperation in the robotics domain. IEEE Transactions on Human-Machine Systems 52, 12–25. doi:10.1109/THMS.2021.3131684.

[63] Zacharaki, A., Kostavelis, I., Gasteratos, A., Dokas, I., 2020. Safety bounds in human robot interaction: A survey. Safety Science 127, 104667. doi:10.1016/j.ssci.2020.104667.

[64] Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T., 2017. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1802–1811.