Computers in Industry 130 (2021) 103482

Contents lists available at ScienceDirect

Computers in Industry

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

Interactive Process Automation based on lightweight object detection in manufacturing processes

Amolkirat Singh Mangat[a,*], Juergen Mangler[b], Stefanie Rinderle-Ma[b]

[a] Research Group Workflow Systems and Technology, Faculty of Computer Science, University of Vienna, Waehringerstrasse 29, 1090 Vienna, Austria
[b] Chair of Information Systems and Business Process Management, Departments of Informatics, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 22 December 2020
Received in revised form 29 March 2021
Accepted 5 May 2021
Available online 15 May 2021

Keywords:
Interactive Process Automation
Object detection
Synthetic training images
Deep learning
Manufacturing processes

ABSTRACT

Interactive Process Automation refers to the idea of supporting the interaction of humans in processes through physical objects. This is particularly promising for human/cobot collaboration tasks where the communication is fuzzy. A typical example is a picking and placing scenario. Here, a "picking area" can serve as a user interface, i.e., objects are freely placed in a defined area, and then identified and transferred to specific positions, where deterministic processes can use them. If, for example, object A is placed at position pos_A by the human, automatically, the robot is instructed to pick A and place it at position pos_B on a tray. Realizing Interactive Process Automation for picking and placing tasks in manufacturing processes requires (i) a lightweight and flexible object detection approach and (ii) a human–machine interface design for Interactive Process Automation. This work proposes (i) an object detection approach that works solely based on synthetic training data. The object detection is embedded into (ii) generic process models that are implemented based on an existing manufacturing orchestration framework and a camera-equipped cobot. The approach is prototypically implemented and evaluated based on several experiments including a pick and place cobot station.

1. Introduction

Assisting human labour and replacing mundane repetitive activities through collaborative robots (cobots) is gaining increasing interest in the manufacturing domain (Weckenborg et al., 2019). Contrary to the widespread fear and belief that autonomous robots are replacing human workers (Wewerka et al., 2020), research is focusing on collaborative human–robot solutions that free human workers for tasks robots cannot perform autonomously in the near future. With the gained free time, human workers can make a transition into, e.g., a supervisory and maintenance oriented role, and are enabled to engage into creative and purposeful/value-added activities that are beneficial to enterprises (Syed et al., 2020).

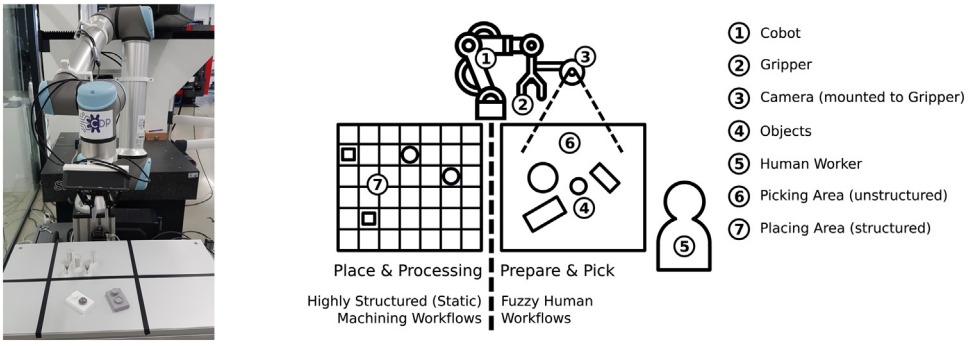
The idea of automating business processes by delegating work to machines is not new. In manufacturing repetitive and structured activities have been automated with the help of industrial robots over the past decades. Well-known examples include vehicle assemblies, palletization and filling and packaging of goods.

The focus of this paper lies on the human–cobot collaboration for a picking and placing scenario, where the nature of the communication between these collaborators is fuzzy. An interface for such collaborations in manufacturing that resides between humans and robots, is the loading station (see Fig. 1a). The purpose of the loading station is to bridge the communication gap between humans and robots to empower robots to work with semi-structured/fuzzy instructions provided by the human. We refer to this way of collaboration as *Interactive Process Automation*.

Interacting with a cobot from the human perspective in the realm of a picking and placing scenario can be illustrated based on the following request: *"Take objects $a_1, \ldots, a_n$ and produce item A. Once done, place the A at position pos_A."*. This set of instructions resembles the interaction humans would use when engaging with other humans. In our scenario the role of the human worker is to provide the necessary objects (e.g., raw materials or tools), while the task of the cobot is to recognize and process the objects based on the human input (i.e., objects provided by the human). The provisioning and recognition of objects mark the fundamental steps that initiate an activity or series of activities of a business process. Fig. 1b provides an overview of the components and participants

* Corresponding author.
E-mail addresses: amolkirat.singh.mangat@univie.ac.at (A.S. Mangat), juergen.mangler@tum.de (J. Mangler), stefanie.rinderle-ma@tum.de (S. Rinderle-Ma).

https://doi.org/10.1016/j.compind.2021.103482



(a) Loading Station

(b) Participants and Components of the Loading Station

Fig. 1. Loading station (a) and human/cobot interaction (b).

that together constitute a loading station. The fundamental piece of the communication interface is the picking area (6). Here a human worker (5) interacts with the cobot (1) by providing the objects (4). Note that the placement of objects in the picking area may be arbitrary, similar to the way humans would hand over objects to other humans, e.g., on the cashier counter. The cobot participates in this interaction by using the attached camera (3) to recognize and interpret the objects on the picking area to derive its next task. Using the gripper (2) the cobot can grab objects and place them in the structured area (7) for further automated processing.

Collaborative human–robot solutions have gained huge interest lately[1] due to the availability of affordable and programmable collaborative robots (Ranz et al., 2018), but also due to the remarkable advances in the computer vision research. Although this creates an opportunity for small to medium sized manufacturing enterprises to benefit from these technological advances, they often do not have the time and resources, and most significantly lack the knowledge to implement such solutions. Furthermore the introduction into the day to day business processes poses a serious challenge with regards to the human factors (e.g., resentments and rejection due to the fear of being replaced/losing the job or major changes to the familiar work habit), but also organizational factors such as the time to adapt and adjust the manufacturing process without causing substantial production downtimes.

This calls for solutions that are (a) affordable, (b) adaptable/flexible for ever-changing business requirements, (c) easy to integrate into the daily business processes, and (d) user friendly. A key challenge is to design the human–robot interaction similar to the human-to-human experience, to ensure a seamless collaboration between humans and robots.

Thus, this work investigates the development of a picking and placing solution – a commonly required step in manufacturing processes – in a collaborative and interactive fashion between human and robots. To address (a) we use a non-commercial, state of the art deep learning approach, and, automatically generated and labelled synthetic training images of objects with a low fidelity representation from 3D model artefacts. Synthetic training data can save labour costs, as one can limit or completely circumvent data collection and labelling by humans. We address (b) and (c) by leveraging a process engine to flexibly orchestrate activities for the realization of human–cobot interface, and to enable the possibility to flexibly substitute object detection models to accommodate for new requirements. This includes the provision of generic process models in standard BPMN[2] notation for fostering a broad applicability. Finally for (d) we employ a simple interaction between the human and cobot by using objects and a loading station as means to determine and execute the intended task for the cobot.

The main contributions of this paper are as follows:

- Firstly, we show how the presented concepts contribute to the idea of interactive process automation, i.e., leveraging physical objects such as the loading station as collaborative interfaces between human and cobot to automatically enact, execute, and complete manufacturing process tasks.
- Secondly, we investigate the feasibility of synthetic training images in combination with supervised deep learning approaches to train object detection models, to locate low-texture objects in real-world monocular images within a manufacturing context.

The remainder of this paper is structured as follows: Section 2 presents and discusses related work, including efforts of the industry for vision based cobot solutions. Section 3 presents our approach for a pick and place scenario using cobots and a process driven approach. Section 4 describes the implementation details for our approach presented in Section 3.

The evaluation of our approach is presented in Section 5. Finally Section 6 discusses and summarizes the presented approach and obtained evaluation results, followed by a conclusion in Section 7.

2. Related work and industry efforts

Interactive Process Automation can be classified into the field of "Internet of Things (IoT) meets process technology" (Janiesch et al., 2020) with a strong focus on the human in the loop. A first contribution towards interactive process automation is the automatic task completion and documentation through NFC-equipped physical objects such as a toothbrush applied in the care domain (Stertz et al., 2020).

The interactive process automation approach for picking and placing in manufacturing processes presented in this work relies on **object detection**, i.e., the ability to locate and classify objects of interest in images. Classic approaches to object detection employ a sliding window paradigm in combination with supervised machine learning techniques. The training data for the machine learning techniques (e.g., Ada-Boost (Freund and Schapire, 2021) or Support Vector Machine (Boser et al., 1992)) is generated using feature description techniques, e.g., Haar wavelets (Viola and Jones, 2001), Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) and Scale Invariant Feature Transform (SIFT) (Lowe, 2021). A trained detection model is then applied to various regions of an image. Later (Felzenszwalb et al., 2008) introduced the framework Deformable Parts Model that utilizes the concept of pictorial structures to model

[1] https://ifr.org/ifr-press-releases/news/record-2.7-million-robots-work-in-factories-around-the-globe Last Accessed: 01.03.2021.

[2] www.bpmn.org Last Accessed: 09.12.2020.

the appearance of objects as a collection of individual interconnected parts.

However, the above mentioned approaches suffer from their inherent inability to capture more diverse significant features. Also these approaches require upfront expert knowledge to identify key features and special techniques to extract them. A solution that addresses both these issues are convolutional neural networks (ConvNet). First demonstrated for handwritten cypher recognition (Lecun et al., 2021), ConvNets have been shown to perform exceptionally well as feature extractors for multi-object object classification (Krizhevsky et al., 2021; Simonyan and Zisserman, 2021; Szegedy et al., 2021; He et al., 2016). This has led to the adoption of ConvNets also for the more challenging object detection use cases. Earlier approaches focused on multi-stage detectors, which use two separate networks, one to propose regions containing the object, and another network to detect and refine the bounding boxes (Sermanet et al., 2021; Girshick et al., 2014; Girshick, 2015). In contrast, single-stage detectors utilize one unified network architecture for predicting bounding boxes directly from a feature extraction network (Liu et al., 2016; Redmon et al., 2016; Redmon and Farhadi, 2017). To deal with objects occurring at multiple scales, predictions are done at various layers of feature extraction network (Lin et al., 2021; Redmon and Farhadi, 2021). In our approach we utilize the general purpose deep learning object detection architecture YOLOv3 (Redmon and Farhadi, 2021) that provides a good balance between object detection and inference speed.

Synthetic training data. Well-labelled and sufficiently available training data is a fundamental requirement for supervised machine learning algorithms. However, the availability of training data cannot always be guaranteed, e.g., due to legal obligations (data privacy), unavailability of human resources, cost of labeling, etc. An approach to overcome the lack of sufficient real-world training data is to supplement training data with synthetic training data.

A common practice for training images is data augmentation that pursues the goal of transforming available labelled images into different variations of the original images. Transformations include operation such as cropping, distortions, colour manipulation, etc. (Dodge and Karam, 2016). In Cubuk et al. (2021) a ConvNet based solution is proposed, capable of inferring performance improving data augmentations strategies.

Another prominent technique is image composition. The goal here is to compose new images from existing labelled images, e.g., by placing crops of objects onto arbitrary backgrounds. Instead of random image compositions, Gupta et al. (2016) proposes a geometry aware integration technique to place texts onto natural scenes such that they align realistically with their background. Generative Adversarial Networks (GAN) (Goodfellow et al., 2021) represent a special class of neural networks that learn a generative model to produce synthetic outputs of a target domain. Shrivastava et al. (2017) have proposed a GAN that refines synthetic images into more realistic images.

The above mentioned approaches assume availability of labelled (real-world) training images. The alternative approach is to substitute real-world training images entirely with synthetic images. To represent objects, 3D models are used. A natural choice is to generate images with photorealistic rendering (Proenca and Gao, 2021; Movshovitz-Attias et al., 2016) to acquire training images with high resemblance to their real-world counterparts. However, achieving high photorealism requires advanced skills and experience, and thus can be challenging. Hence, alternative approaches have been proposed that attempt to use less-photorealistic, low-fidelity representations of objects in training images (Sun and Saenko, 2021; Peng et al., 2015; Tremblay et al., 2018; Hinterstoisser et al., 2018) in combination with random image composition. In Tobin et al. (2017) the authors take a slightly different approach to synthetic image generation by proposing to entirely randomize the appearance of objects. They argue that a broad randomized diversification of the object appearances and backgrounds can compensate for the lack of photorealism in the generated synthetic images. For the approach presented in this paper we employ low-fidelity synthetic training images comparable to Tobin et al. (2017), which we generate from readily available 3D models of objects.

Industry efforts. Enabling robots and humans to work alongside in *a shared workspace* has been the target of several robot manufacturers in the recent years. Major contributors include Universal Robots and Rethink Robotics that offer collaborative robots *(cobots)* (see Table 1). The key difference between traditional industrial robots and the cobots is the focus of cobots on human safety that permits humans to work while cobots are operating. The cobots are built to cease operation in case of accidental collisions with humans to prevent injuries.

This paper has a particular focus on cobots with the ability to detect objects from images using a vision based system to drive processes through human–cobot interaction. Thus, we investigated solutions offered by cobot manufacturers. The cobot manufacturers have been determined using the Google search engine and the search terms "*cobot manufacturer OR collaborative robot manufacturer*". We searched[3] the first 30 results for manufacturers that (a) have a company website (b) offer cobot solutions and (c) currently exist. Then we scrutinized the websites of the cobot manufacturers to determine if a vision based object detection solution is offered and whether it provides the possibility to learn new objects for detection and utilizes synthetic training images. We also considered manufacturers that utilize third-party object detection solutions. The results of our findings are presented in Table 1 in alphabetical order.

Overall the approaches employed by the cobot manufacturers and third-party vendors of vision systems for cobots can be distinguished in 2D image and 2.5–3D depth image based object detection. Based on our findings above, the state-of-the-art vision system approaches, for instance by Cognex and ABB, apply a user-guided learning mechanism for object detection, which requires human operators. The training images are collected by an operator using the provided vision system's training software and camera that is attached to the cobot. During the image collection process the images are manually labelled in the software. In contrast, 2.5–3D image based approaches, which includes the solution by Scape, requires 3D models of objects that require a virtual 3D representation of real world objects. The detection then focuses on aligning 3D models with real-world objects. Overall the learning approach is similar to the 2D image approaches with regards to the training data collection. It also requires a human operator to train cobots in place.

The approach presented in this paper differs from the vision systems employed by industrial approaches as follows: It employs non-commercial learning techniques and uses synthetic training images generated from readily available 3D models. Therefore, the presented approach does not require manual labelling of objects.

3. Interactive Process Automation: interface design

The key piece in the human–cobot collaboration scenario supported by interactive process automation is the loading station. The loading station is the interface responsible for exchanging information between the human and its robotic counterpart – the cobot – and for initiating the next task for the cobot. Fig. 2 represents our

[3] Search Date: 15.03.2021.

Table 1
Cobot manufacturers and the supported vision systems.

Cobot manufacturer	Object detection solution	Learning solution	Synthetic training images	Description vision system
ABB	√	√	×	Integrated Vision
Doosan	×	n.a.	n.a.	n.a.
Kassow Robots	×	n.a.	n.a.	n.a.
Kuka Robotics	×	n.a.	n.a.	n.a.
Rethink Robotics	√	√	×	Cognex
Techman Robot	√	n.a.	n.a.	n.a.
Universal Robots	√	√	×	Scape, Dalsa Vision, Keyence
Yaskawa	×	n.a.	n.a.	n.a.
Yuanda	×	n.a.	n.a.	n.a.

√ = applies, × = does not apply, n.a. = data not available.

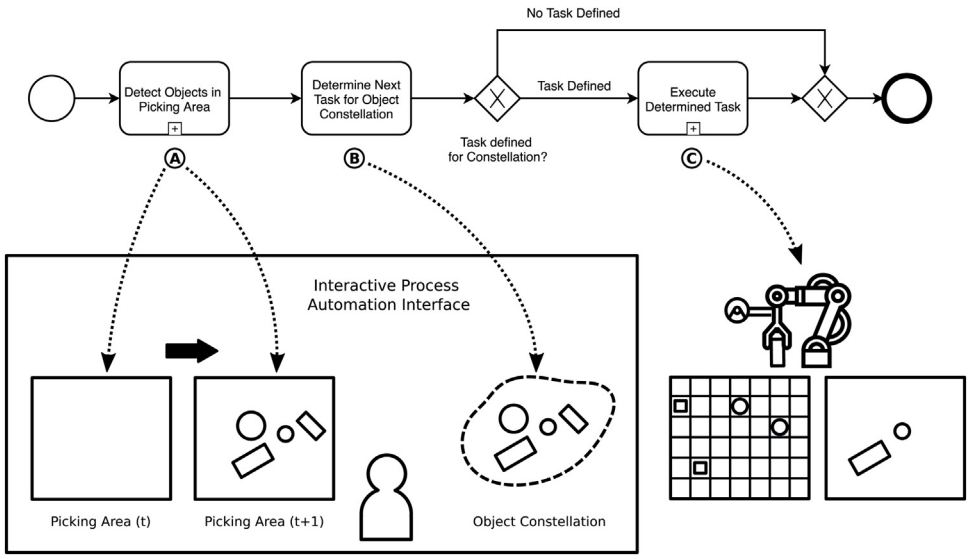


Fig. 2. Automatic task initiation driven by human–machine collaboration.

conceptual approach for the realization of the interactive process automation for the human–cobot interaction using a process driven approach.

The interface is designed through a set of process models that realize the interaction of human, cobot, and system, i.e., the process engine. Figs. 2–4 provide these process models in a generic manner using the standard Business Process Modeling and Notation (BPMN[1]) in order to foster a broad applicability and interoperability. The process model at the top of Fig. 2 describes the fundamental activities/sub processes and flow of control to drive the human–cobot collaboration and automatic task initiation in terms of a super process that employs several sub processes. Activities *Detect Objects in Picking Area* (A) and *Determine Next Task for Object Constellation* (B) constitute the interactive process automation interface. (A) is responsible for observing changes to the picking area, i.e., locating and identifying objects a human worker has placed on the picking area. After objects on the picking area have been noticed, (B) is triggered to determine if a task for the current object constellation on the loading has been defined. If a task for the constellation can be determined, it is carried out by activity *Execute Determined Task* (C). After the completion of a task or if no task can be determined the process terminates. Subsequently the entire process is repeated.

A detailed description of activity *Detect Objects in Picking Area* (A) is presented in Fig. 3. At first, activity *Bring Camera into Position* positions the camera attached to the cobot in order to capture the picking area from the desired perspective. Then, sub process *Detect Objects* locates objects on the picking area. The object detection approach is described at the end of this section. This step involves taking an image of the picking area and then determining if objects of interest are present in the image. The process of locating objects is repeated until objects of interest are detected in the picking area.

To give human collaborators sufficient time to place items in the picking area and to avoid acting prematurely, a mechanism is required to ensure the intended task(s) for the provided object is carried out. We propose a time based approach, as this does not require any additional impulse from the human in form of a physical confirmation (e.g., pressing a button) or a complex vision/sensor based confirmation (e.g., human is not in the proximity of the loading station). Alternative approaches are further discussed in Section 6. Therefore an arbitrarily configurable delay X in seconds is introduced (represented by the time event) to give sufficient time for object placement or removal. After the delay the objects in the picking area (activity *Detect Objects*) are determined again. If changes in the object constellation are observed with regards to the previous object detection results, the process of waiting and checking for changes in the object constellation is repeated.

If no change in objects in the picking area is observed, the overall process transitions to activity (B) to determine the next task the cobot needs to carry out based on the current object constellation. If a task is found that has been defined for the current object constellation, then the task is executed as indicated by activity (C) in Fig. 2. The determination of the objects in the picking area and the look-up of a corresponding task represent the fundamental prerequisites for the realization of interactive process automation. A detailed and generic representation for activity (C) is depicted in Fig. 4.

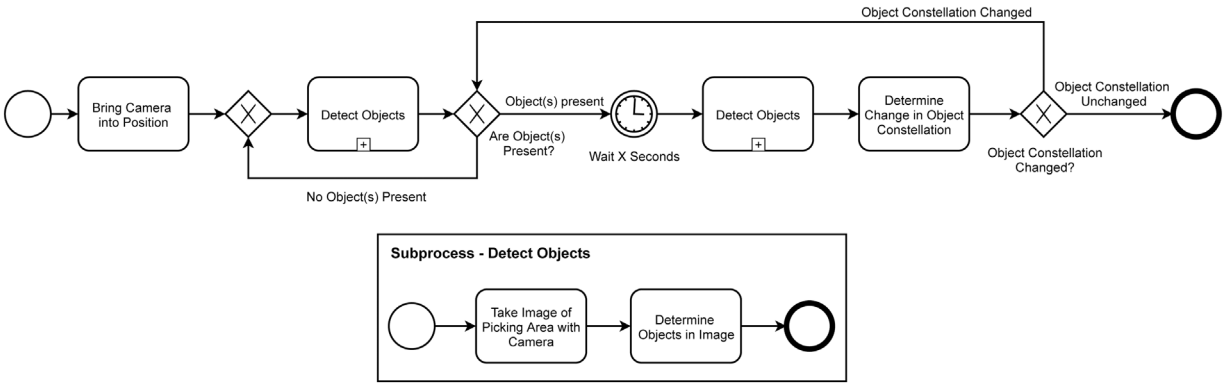


Fig. 3. Sub process *detect objects in picking area* for enabling interactive process automation.

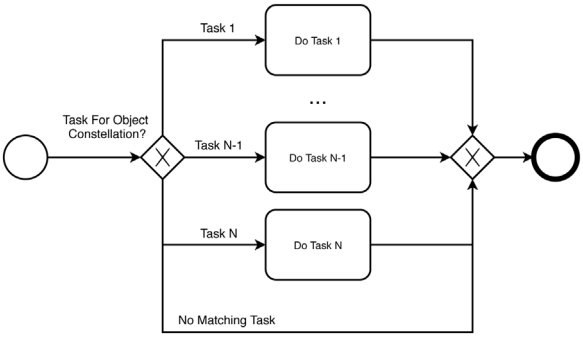


Fig. 4. Sub process *execute determined task*.

Depending on the object constellation and task look-up a matching branch with regards to the determined task is executed. If no match is found for an object constellation or a determined task has been completed, the process terminates.

4. Object detection using synthetic images

This section details the proposed synthetic image generation tool chain and the object detection solution used to realize the interactive process automation interface (Fig. 3).

The common language of our participants (human and cobot) are the objects. The cobot having no natural means of communication with human, receives its real-world input from a camera. To interpret the visual cues in the images provided by the camera, the cobot is supported by an object detection model that can detect the object class and the location of objects in the image. In this work we rely on a deep learning model trained with synthetic training images with supervised-learning. The input of this model are coloured images. The output of the model are coordinates of bounding boxes for located objects and class labels. We rely on synthetic training images to circumvent time-consuming manual collection and labelling of training images.

4.1. Synthetic image synthesis

We use non-photorealistic rendered training images and synthetic images generated from cropped real-world object images, i.e., object images with transparent background. In Sun and Saenko (2021), Peng et al. (2015) and Tobin et al. (2017) the authors report synthetic training data to deliver competitive performance in comparison to real world training images for object detection tasks.

For the generation of purely synthetic training images we implemented a small library in the Python programming language around the graphics rendering engine Povray[4] and by using 3D models to represent the real-world objects. An overview of our approach is presented in Fig. 5. Povray is capable of generating low-fidelity to photo-realistic three-dimensional graphics. The rendering engine provides a declarative scene description language to specify the visible space and user perspective, the illumination, backgrounds, objects, and the appearance of the objects. We use a custom configuration file to describe a Povray scene, which we then use to generate the synthetic images with our library. The details for the scene configurations are covered in Section 5.2.1.

For the objects we used 3D models in the STL format that needed to be converted into Povray's format. Although the generation of synthetic training data in general is cheap, being able to eliminate cases which are unlikely to occur (i.e., poses of objects) can (a) reduce the time to generate images, (b) shorten the training time when using deep learning approaches and (c) limit the complexity of the detection model. In our scenario we focus on objects that are placed on a flat surface. Thus, we can safely ignore certain object poses and only consider physically realistic poses that we define using the Extensible Markup Language (XML).[5] We refer to our description language as Object Pose Description XML (see Fig. 5), which defines relevant poses as rotations around the origin in 3D-space with respect to the initial pose of the object in the STL model.

To automatically obtain the labels for the bounding boxes for each rendered image, we generated additional images as PNGs with transparent backgrounds that only contain the object. The bounding box coordinates are computed considering the minimal and maximal non-transparent pixels in the vertical and horizontal direction of the image.

For creating synthetic images from real-world background images and real-world objects we also implemented an image composition tool in Python with automatic labelling capabilities. The tool randomly rotates and pastes the object images onto the background images. The objects are expected as PNG images with the background pixels set to transparent in the alpha channel. The bounding boxes can be computed from the silhouette represented by the alpha channel of the PNG images.

4.2. Object detector

Before we can grip an object we require the location, and the class and pose labels of objects in an image. For our object detectors we employ the state-of-the-art YOLOv3 architecture (Redmon and Farhadi, 2021) with spatial pyramid pooling. YOLOv3 is a single-

[4] http://povray.org Last Accessed: 09.12.2020
[5] https://www.w3.org/TR/xml11 Last Accessed: 09.12.2020.

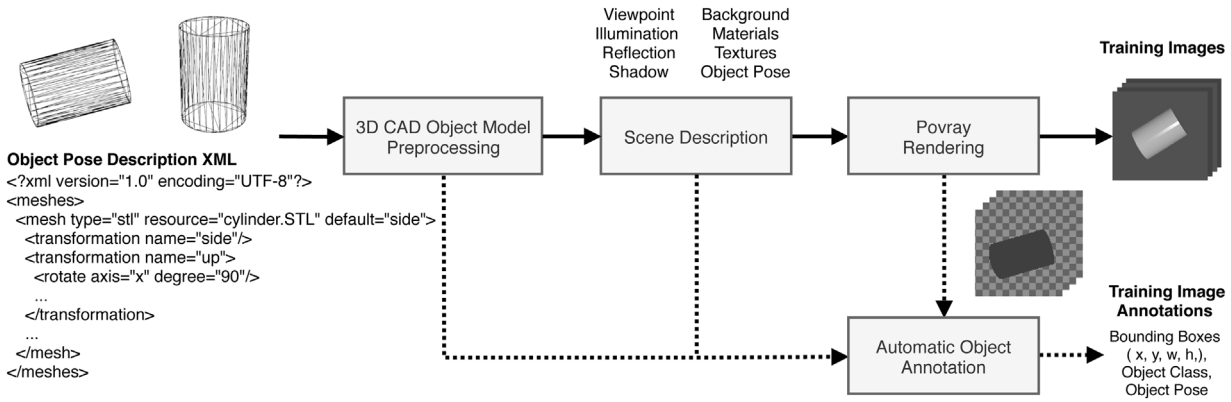
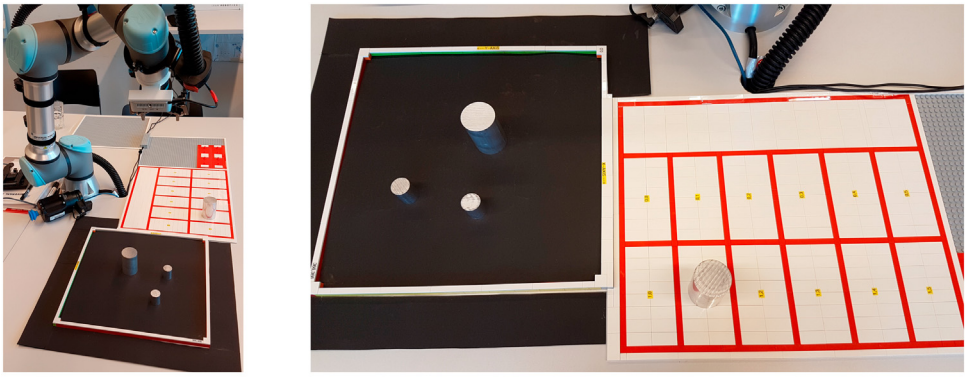


Fig. 5. Synthetic training image generation for supervised object detection model training.



(a) Cobot in the Camera Mode. (b) Human-Cobot Interface (Left Black Area) and Structured Area (Right Area).

Fig. 6. Experimental loading station setup for the evaluation.

stage-detector that comprises of a feature extraction component and a regression component which predicts the bounding boxes and class labels for objects in an input image. The feature extraction backbone uses a deep architecture with 52 convolution operations. The regression component applies a mixture of convolution and upsampling operations to feature maps, i.e., at different stages of the network the feature maps are condensed and enlarged. To increase the overall exposure to features for additional information, YOLOv3 also concatenates earlier fine-grained feature maps with upsampled maps of later stages. The regression component of this network predicts at three different scales, to enable the detection of small and large objects. For the implementation of YOLOv3 we used the library Darknet (Redmon, 2021).

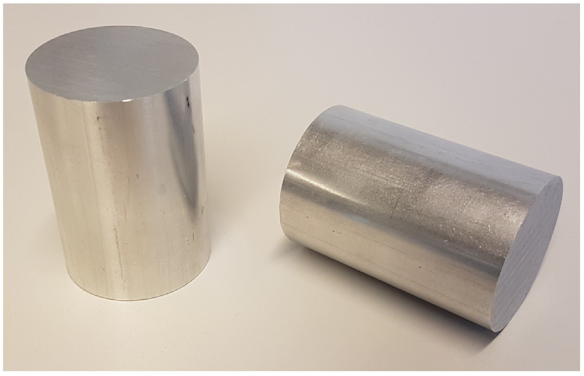


Fig. 7. Steel cylinders in the lying and upright pose.

5. Experiments

5.1. Experimental setup

The experimental setup closely resembles a loading station in a manufacturing environment. The loading station consists of a Universal Robot 10 (UR10) cobot equipped with an Intel Realsense D415 camera that produces images of 1920 × 1080 resolution and a gripper with two fingers (see Fig. 6a), a picking area (see Fig. 6b) for the realization of the human–cobot interface for picking and preparing, and, a structured area for the placing and processing of objects by the cobot. We chose a matte black background to eliminate shadow and background reflections.

To demonstrate the detection of objects placed by humans on the picking area, we use steel cylinders depicted in Fig. 7. We consider both the upright and lying pose of the cylinder. In addition we also expect to detect the objects at different scales. Using the software of the UR10, the cobot has been programmed to allow us to move in the birds-eye view to capture the picking area from above, hover to a specific location above the picking area when provided the coordinates of an object's location in the image, and to grab objects vertically from above and place them to a specific location in the structured area as show in Fig. 6.

All three operations are available as REST services. They can be invoked by the process depicted in Fig. 2 that is implemented using the open source process engine Cloud Process Execution Engine (CPEE).[6]

[6] https://cpee.org Last Accessed: 09.10.2020.

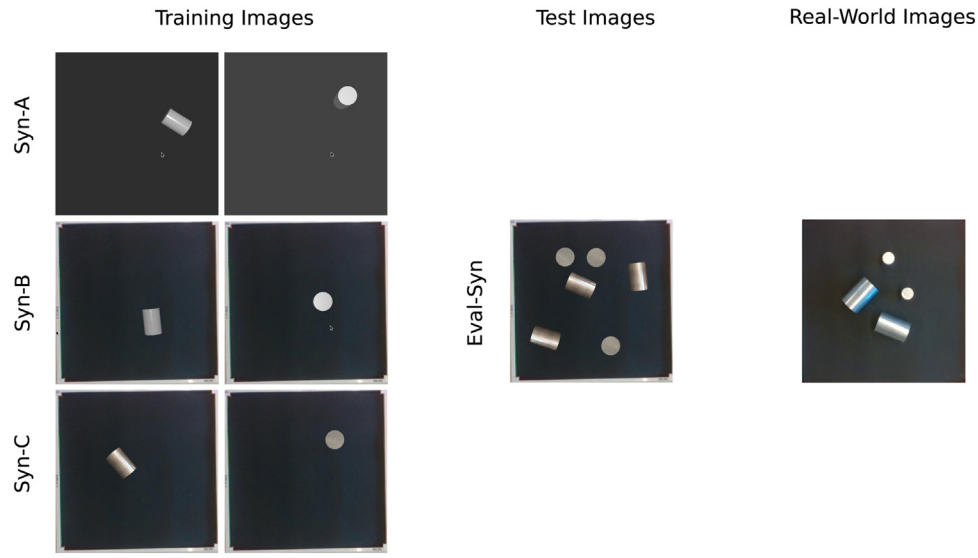


Fig. 8. Synthetically generated and composed training and test images for the detection of objects in real-world images.

We opted for the CPEE as it provides a process modelling and execution environment, and among others supports the invocation of HTTP based web services. Thus, the CPEE allows orchestrating the execution of the Interactive Process Automation activities that include taking images with the camera, invoking the object detection model, determining the task based on the detected object constellation, and running the determined task which invokes a particular program of the cobot.

5.2. Object detection

Recall that the purpose of the object detection step is to detect the object and its pose. This section evaluates the feasibility of the proposed object detection solution that we trained solely with synthetic training data for the realization of the interactive human–cobot interplay.

5.2.1. Synthetic training datasets

In total we generated three synthetic data sets *Syn-A*, *Syn-B* and *Syn-C* for training, using our image generation approach presented in 4. All data sets comprise of three thousand RGB images for each object-pose. Each image contains a single object. We use a birds-eye perspective that resembles our experimental setup with a distance of 45cm between the camera and the table. The visible scene for the detection models is limited to the human–cobot interaction zone (see Fig. 8).

- ***Syn-A***. This dataset consists of purely synthetically created images. Objects have been rendered on a randomized single colour background. This approach is inspired by the domain randomization approach introduced in Tobin et al. (2017). However, we limit the range to colours that matches our experimental setup, i.e., the objects and the background are randomly assigned colours from the pool of gray shades to reflect the natural appearance. In addition we apply randomized phong effects to the object to vary the object finish from a matte, to a reflective and shiny appearance. We applied constant daylight illumination directed at the centre of the visible scene vertically from above.
- ***Syn-B***. In contrast to a purely synthetically rendered image set, we apply a mixed approach using synthetically generated objects and real-world images of backgrounds. We use image composition by randomly pasting rendered object onto real-world image backgrounds. The object appearance and illumination is kept similar to the approach for *Syn-A*.
- ***Syn-C***. For insightful comparisons with images using synthetically rendered image elements, we prepared a third dataset composed of images using real-world crops of objects with transparent background that are randomly pasted onto real-world backgrounds.

5.2.2. Model training

For each training data set we train an object detection model using the YoloV3 network architecture. We use the Darknet-53 ConvNet, pre-trained on the COCO datset, as the feature extraction backbone. Fine tuning pre-trained models for custom datasets have been shown to be a viable approach (Pan and Yang, 2021; Girshick et al., 2014), if the model has been exposed to a large collection of diverse images. The COCO dataset is a large image collection, consisting of over 200-thousand images with 80 object categories.

We set the network input size to 416×416 pixels. We used batch gradient descent with a batch size of 64. We trained for six thousand iterations. For back-propagation we used stochastic gradient descent with a learning rate of 2×10^{-3}, which we decreased in steps, namely after four thousand iterations by 10^{-1} and after four thousand iterations again by 10^{-1}. Furthermore regularization for the weights was applied with a decay of 5×10^{-4}. Momentum was set to 0.9. We kept the network's ability to dynamically change the input size of the network during training enabled, as this step acts as an additional data augmentation mechanism and enables the network to learn at different scales with regards to the object size. The change of the input size is carried out randomly.

We also applied standard data augmentation techniques including random image flipping and cropping, and image distortions (using the HSL colourspace) by randomly adapting hue, saturation and exposure, that are provide by the Darknet library. The saturation and exposure parameters were set to 1.5 and hue to 0.1. The jitter parameter that controls the amount of image cropping during training in the Yolo layers (responsible for predicting the bounding boxes) was set to 0.3.

5.2.3. Evaluation methodology

To measure the object detection performance with regards to the correctness of the class label and bounding box predictions, we use the widely applied metric average precision (AP) (Everingham et al., 2021). AP is defined as a probability density function of ranked

recall values. Precision measures the fraction of predictions that are correct with regards to all prediction results. Recall, or sensitivity, measures the fraction of ground truth/relevant objects that have been correctly localized.

We approximate AP as proposed by Everingham et al. (2021) using the eleven point interpolation scheme

$$AP = \frac{1}{11} \sum_{r \in [0, 0.1, \ldots, 1]} \max_{\tilde{r} \geq r} p(\tilde{r}), \quad (1)$$

where r represent recall measures and $p(r)$ the precision as function of recall.

The AP score is computed for each object class/category individually. To provide an overall score irrespective of the object class, we use mean average precision (mAP) similar to Everingham et al. (2021). The mAP score is computed by averaging the sum of all class specific average precision scores.

We apply the Jaccard index, or intersection over union (IoU), expressed as

$$IoU = \frac{B_{gt} \cap B_p}{B_{gt} \cup B_p}. \quad (2)$$

to determine the correctness of a bounding box prediction. This measure computes the overlap of bounding boxes as fraction by comparing the ground truth set of pixels B_{gt} against the predicated set of pixels B_p within the bounding boxes. We consider bounding box predictions with an IoU score of greater or equal 0.5 as correct.

5.2.4. Model evaluation results

We use synthetic data to train models, but require the models to operate with real-world images as input at inference time. To reliably assess whether the models are capable of bridging the reality gap between the synthetic and real-world data domain, measuring the performance with real-world images is essential. Since one of our goals is to avoid manual labour we have prepared a synthetic test set *Eval-Syn* that comprises of 40 images composed of real world objects and backgrounds as depicted in Fig. 8. Each image contains up to six objects that are randomly pasted onto the background, similar to the approach for the training dataset *Syn-C*.

We refer to the object detection models trained on the training datasets *Syn-A*, *Syn-B* and *Syn-C* as *OD-A*, *OD-B*, and *OD-C* respectively. Table 2 summarizes the first best mAP scores obtained with regards to the number of training iterations for the synthetic real-world evaluation set *Eval-Syn* for all the detection models. During training we saved a checkpoint for every 100th iteration for the first 1000 iterations and after that for every 500th iteration. The development of the mAP during the course of the training is depicted in Fig. 9. We can observe that the models of the training sets that include real-world elements (real-world background, object, or both) converge faster, whereas the model (*OD-A*) trained on purely synthetic training images requires significantly more training iterations to achieve a similar perfect mAP score for the *Eval-Syn* as the models *OD-B* and *OD-C*. This indicates the dependency of training data domain on the run-time data domain.

Table 2
Evaluation results on the dataset *Eval-Syn*.

Model	OD-A	OD-B	OD-C
Iterations	3500	2000	2000
mAP	100%	100%	100%

5.3. Interface evaluation

Our experiments above show that synthetic training data can perform well for synthetic real-world test images. To assess whether this also applies for our real-world object detection application for our picking area, we collected 18 images at our experimental setup using the camera equipped with the cobot. We refer to this dataset as *Eval-R*. Each image in *Eval-R* contains at least two and at most four objects. To also test whether the detection at multiple scales of the cylinder objects works for both the upright and lying pose, we also used cylinder objects with a smaller dimensions. We report the mAP scores for the models *OD-A*, *OD-B*, and *OD-C* with 3500 training iterations. Fig. 8 depicts a few selected samples of evaluated images for each object detection model by row (Fig. 10).

Since missing out on present objects in an image is critical for the realization of the interface we report the results as recall, i.e., the fraction of present objects identified with regards to all objects that are present in the image. Overall *OD-A* achieves recall score of 77.5%, *OD-B* 89.7%, and *OD-C* 83.6%. All detected objects are correctly classified with a precision score of 100%. If further distinguished between the standard cylinders and the small cylinders, then all three models achieve recall scores of 100% considering only the standard sized cylinders, while the recall scores for the small cylinders are 59.2% for *OD-A*, 81.4% for *OD-B* and 70.3% for *OD-C* (Table 3).

Next we measure the accuracy of correctly detected object constellations. We only consider an object constellation as correctly interpreted by the object detection models, if all objects in an image are correctly detected. *OD-A* achieves an accuracy of 55.5%, *OD-B* 77.7%, and *OD-C* scores 72.2%. If only the standard sized cylin-

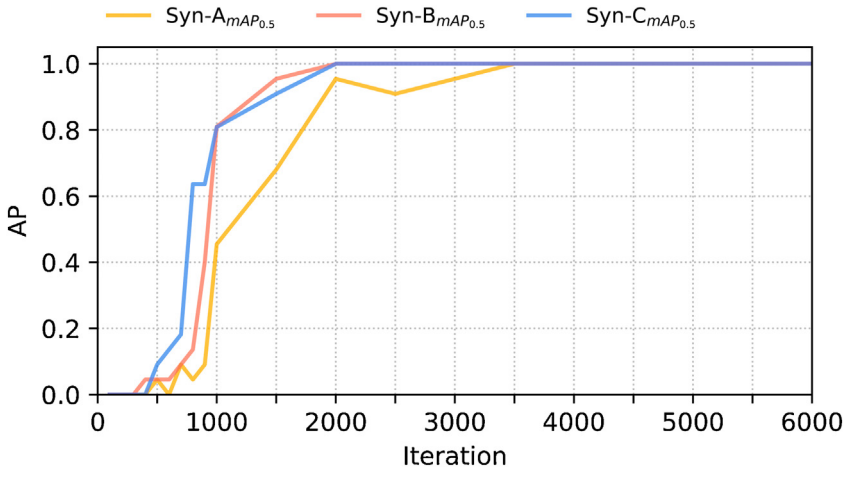


Fig. 9. Average precision scores for the synthetic evaluation set *Eval-Syn* over the course of the model training for the training sets *Syn-A*, *Syn-C* and *Syn-C*.

Fig. 10. Object detection results as bounding boxes for the cylinder in the lying pose (blue) and upright pose (orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Interface evaluation results for *Eval-R*.

Model	OD-A	OD-B	OD-C
F1 Score	87.3%	94.6%	91.1%
Precision	100%	100%	100%
Recall	77.5%	89.7%	83.6%

ders are considered all three models achieve 100% accuracy for the object constellation detection. For the small cylinders the models *OD-A*, *OD-B* and *OD-C* achieve an accuracy score of 27.2%, 54.5% and 45.5% respectively. The results suggest incorporating different variants of the object size directly into the training set, may lead to better results instead of solely relying on the scaling feature of the detection model architecture.

6. Discussion

Reliable Interactive Process Automation. A crucial aspect to reliably ensure the intended process behaviour is to accurately determine the moment a human has completed the placement of objects in the picking area. The approach presented in this work (see Fig. 3) proposes a timer based mechanism, which in regular intervals checks for a stable object constellation. When a predetermined waiting time for changes passes and no changes are detected, the present object constellation is frozen. Objects placed after this point in time are not considered. However, depending on the safety requirements and overall trust in the automatic triggering of processes through human–machine interactions, approaches that require manual triggering (e.g., pressing a button, pushing on a pedal, scanning a card) or require certain conditions to be true (e.g., absence of a person from a safety zone) might be considered more appealing.

Multi-object detection. The approach in this paper focuses on single object, multi-pose detection. The question that arises is, how scaleable is the proposed approach for heterogeneous collections of objects.

State-of-the-art object detection techniques have demonstrated the ability to detect several dozen objects in real-time from 2D images (He et al., 2016; Girshick, 2015; Redmon and Farhadi, 2017). Detecting multiple objects using synthetic training images has also been successfully demonstrated, as laid out in Section 2. Although detecting several dozen objects is likely to scale well, detecting partially occluded objects (Wang et al., 2020) and noise in the input images (Su et al., 2019) still pose a challenge. Since different applications have varying requirements regarding the robustness of the detection results and tolerance levels for errors, a viable approach is to use an ensemble of object detection techniques. This in particular is facilitated by our process driven approach, as it easily allows to incorporate orthogonal object detection approaches in the control flow of the process model. In order to gain further confidence in the detection results, for example, material estimation techniques (e.g., material classification using near-infrared spectroscopy (Tachwali et al., 2007)) can be applied in addition, to validate image based detection results. Likewise multiple object detection models can be applied, where each model focuses on a particular task (e.g., detection of colour, shape, texture, size, or material of objects) and the combination of the detection tasks can be used to achieve more robust results.

Feasibility of synthetic training images. The interface evaluation results demonstrate that purely synthetically generated low-fidelity training images can lead to competitive results when compared to training images that are generated to closely resemble real-world representations of objects. An integral benefit of applying synthetic training images is the control it provides over their appearance. Although manufacturing sites traditionally have a static setup, it is possible to future proof detection models by including random noise to account for unexpected events. In our pick and place scenario with cobots potential noise may include, for example, unexpected items placed onto the picking area, human intervention during an ongoing process (hand gestures over the picking area), colour and texture changes of processed objects, or changes of the surface appearance of the picking area.

However, it is imperative to note that with synthetic training data in general, there is a trade-off between the feasibility and high realism of synthetic data. It is cheap to produce less realistic synthetic images, whereas it is expensive to generate photorealistic images for it requires special knowledge and skills. In this paper, we attempted to maximize the feasibility of synthetic training data by leveraging low-fidelity training images using raw 3D models as the basis. We in particular relied on simple object surface material/texture descriptions that do not require human involvement and are automatable by using pre-existing material/texture description libraries. Using our approach also enables one to incorporate random noise to enrich training images as described above by using arbitrary pre-existing collections of suitable images. Adding random noise is less feasible for the approach taken by cobot manufacturers, which requires a human operator to manually set up scenes for the training images and to collect the images with the cobot for training.

A limitation of our approach is the assumption that 3D models of objects exist. This might not be the case for all small to medium sized manufacturers. Also it is not guaranteed that every type of object can be successfully detected by only using low-fidelity representations of objects as training images. In these two

cases the manual user-guided approach for creating detection models employed by the cobot manufacturers seems to be the more appealing alternative. However, the approach presented in this paper and the user-guided approach offered by the cobot manufacturers are not incompatible. An interesting combination of both approaches could involve starting out with the detection model generated using synthetic training data, and then improving the detection model for wrong detection results by applying the user guided approach with real-world images. The combined approach shows the potential to reduce the overall initial setup effort, provided that 3D models of objects are available.

7. Conclusion

The work presents concepts and a prototypical implementation of an interactive process automation interface that enables humans to communicate with cobots solely through the objects presented on a picking area. The concepts include object detection in combination with a process engine to automatically initiate appropriate processes for a picking and placing scenario. With a particular focus on small to medium sized companies with limited budget and knowledge, the approach for building object detection models is kept lightweight by leveraging low-fidelity synthetic training images generated from 3D models in contrast to using manually labelled real-world images.

The results of the evaluation clearly suggest that it is feasible to use synthetic training data for enabling interactive process automation for human–machine collaboration using object detection. The process driven characteristics of the approach ensures flexibility since individual activities, such as the object detection models, can be substituted in the proposed process models.

Limitations of our experiments include:

- the use of only a single object
- disregarding potential occlusion of objects
- not considering complex shapes of objects with special grabbing requirements
- focus on static object constellation after a task for the determined constellation has been initiated.

Thus, an interesting aspect for future work is to investigate the effects of a dynamic setup when objects are removed or added from the loading station after a task for a determined object constellation has been initiated and is still ongoing. Further promising directions for future work include addressing the above mentioned limitations with regards to interactive process automation and investigating lightweight/low-effort, semi-automated labelling techniques for real-world images that include humans to leverage expert knowledge for better object detection results.

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.

Acknowledgments

This work has been partially supported and funded by the Austrian Research Promotion Agency (FFG) via the “Austrian Competence Center for Digital Production” (CDP) under the contract number 881843.

References

Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992]. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 144–152.
Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V., 2021]. AutoAugment: Learning Augmentation Policies from Data. arXiv:1805.09501.
Dalal, N., Triggs, B., 2005]. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 886–893, http://dx.doi.org/10.1109/CVPR.2005.177.
Dodge, S., Karam, L., 2016]. Understanding how image quality affects deep neural networks. 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), 1–6, http://dx.doi.org/10.1109/QoMEX.2016.7498955.
Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2021]. The pascal visual object classes (VOC). Challenge 88, 303–338, http://dx.doi.org/10.1007/s11263-009-0275-4.
Felzenszwalb, P., McAllester, D., Ramanan, D., 2008]. A discriminatively trained, multiscale, deformable part model. 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1–8, http://dx.doi.org/10.1109/CVPR.2008.4587597.
Freund, Y., Schapire, R.E., 2021]. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139, doi:10.1006/jcss.1997.1504.
Girshick, R., 2015]. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), 1440–1448, http://dx.doi.org/10.1109/ICCV.2015.169.
Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014]. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 580–587, http://dx.doi.org/10.1109/CVPR.2014.81.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2021]. Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 27. Curran Associates, Inc, pp. 2672–2680.
Gupta, A., Vedaldi, A., Zisserman, A., 2016]. Synthetic data for text localisation in natural images. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2315–2324, http://dx.doi.org/10.1109/CVPR.2016.254.
He, K., Zhang, X., Ren, S., Sun, J., 2016]. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.
Hinterstoisser, S., Lepetit, V., Wohlhart, P., Konolige, K., 2018]. On pre-trained image features and synthetic images for deep learning. In: Leal-Taixé, L., Roth, S. (Eds.), Computer Vision – ECCV 2018 Workshops, Vol. 11129. Springer International Publishing, pp. 682–697, http://dx.doi.org/10.1007/978-3-030-11009-3_42.
Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Ciccio, C.D., Fortino, G., Gal, A., Kannengiesser, U., Leotta, F., Mannhardt, F., Marrella, A., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Serral, E., Song, W., Su, J., Torres, V., Weidlich, M., Zhang, L., 2020]. The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. 6, 34–44, http://dx.doi.org/10.1109/MSMC.2020.3003135.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2021]. ImageNet Classification With Deep Convolutional Neural Networks, Vol. 60., pp. 84–90, http://dx.doi.org/10.1145/3065386.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 2021]. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324, http://dx.doi.org/10.1109/5.726791.
Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2021]. Feature Pyramid Networks for Object Detection. arXiv:1612.03144.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016]. SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, pp. 21–37, http://dx.doi.org/10.1007/978-3-319-46448-0_2.
Lowe, D.G., 2021]. Distinctive Image Features from Scale-Invariant Keypoints, Vol. 60., pp. 91–110, http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.
Movshovitz-Attias, Y., Kanade, T., Sheikh, Y., 2016]. How useful is photo-realistic rendering for visual learning? In: Hua, G., Jégou, H. (Eds.), Computer Vision – ECCV 2016 Workshops. Springer International Publishing, pp. 202–217, http://dx.doi.org/10.1007/978-3-319-49409-8_18.
Pan, S.J., Yang, Q., 2021]. A Survey on Transfer Learning, Vol. 22., pp. 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.
Peng, X., Sun, B., Ali, K., Saenko, K., 2015]. Learning deep object detectors from 3D models. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 1278–1286, http://dx.doi.org/10.1109/ICCV.2015.151.
Proenca, P.F., Gao, Y., 2021]. Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering. arXiv:1907.04298.
Ranz, F., Komenda, T., Reisinger, G., Hold, P., Hummel, V., Sihn, W., 2018]. A morphology of human robot collaboration systems for industrial assembly. Proc. CIRP 72, 99–104.
Redmon, J., 2021]. Darknet: Open Source Neural Networks in C. https://pjreddie.com/darknet/.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016]. You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.
Redmon, J., Farhadi, A., 2017]. a. YOLO9000: better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6517–6525, http://dx.doi.org/10.1109/CVPR.2017.690.
Redmon, J., Farhadi, A., 2021]. b. YOLOv3: An Incremental Improvement. arXiv:1804.02767.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2021]. OverFeat: Integrated Recognition. Localization and Detection using Convolutional Networks. arXiv:1312.6229.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R., 2017]. Learning from Simulated and Unsupervised Images through Adversarial Training. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2242–2251, http://dx.doi.org/10.1109/CVPR.2017.241.

Simonyan, K., Zisserman, A., 2021]. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Stertz, F., Mangler, J., Rinderle-Ma, S., 2020]. Balancing patient care and paperwork automatic task enactment and comprehensive documentation in treatment processes. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. 15, http://dx.doi.org/10.18417/emisa.15.11, 11:1–11:28.

Su, J., Vargas, D.V., Sakurai, K., 2019]. One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. 23, 828–841.

Sun, B., Saenko, K., 2021]. From virtual to reality: fast adaptation of virtual object detectors to real domains. BMVC, 3.

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S.J., Ouyang, C., ter Hofstede, A.H., van de Weerd, I., Wynn, M.T., Reijers, H.A., 2020]. Robotic process automation: contemporary themes and challenges. Comput. Ind. 115, 103162.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2021]. Going Deeper with Convolutions. arXiv:1409.4842.

Tachwali, Y., Al-Assaf, Y., Al-Ali, A., 2007]. Automatic multistage classification system for plastic bottles recycling. Resources. Conserv. Recycl. 52, 266–285.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P., 2017]. Domain randomization for transferring deep neural networks from simulation to the real world. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 23–30, http://dx.doi.org/10.1109/IROS.2017.8202133.

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S., 2018]. Training Deep Networks with Synthetic Data: bridging the Reality Gap by Domain Randomization. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1082–10828, http://dx.doi.org/10.1109/CVPRW.2018.00143.

Viola, P., Jones, M., 2001]. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, IEEE Comput. Soc. I-511-I-518, http://dx.doi.org/10.1109/CVPR.2001.990517.

Wang, A., Sun, Y., Kortylewski, A., Yuille, A.L., 2020]. Robust object detection under occlusion with context-aware compositionalnets. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12645–12654.

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., Spengler, T.S., 2019]. Balancing of assembly lines with collaborative robots. Business Res., 1–40.

Wewerka, J., Dax, S., Reichert, M., 2020]. A user acceptance model for robotic process automation. Enterprise Distributed Object Computing, 97–106, http://dx.doi.org/10.1109/EDOC49727.2020.00021.