Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 581–588

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Emerging Data and Industry 4.0 (EDI40)
March 23 - 26, 2021, Warsaw, Poland

Object Detection for Smart Factory Processes by Machine Learning

Lukas Malburg[a,*], Manfred-Peter Rieder[a], Ronny Seiger[c], Patrick Klein[a], Ralph Bergmann[a,b]

[a]*Business Information Systems II, University of Trier, 54296 Trier, Germany*
[b]*German Research Center for Artificial Intelligence (DFKI)*
Branch University of Trier, Behringstraße 21, 54296 Trier, Germany
[c]*Institute of Computer Science, University of St.Gallen, 9000 St.Gallen, Switzerland*

Abstract

The production industry is in a transformation towards more autonomous and intelligent manufacturing. In addition to more flexible production processes to dynamically respond to changes in the environment, it is also essential that production processes are continuously monitored and completed in time. Video-based methods such as object detection systems are still in their infancy and rarely used as basis for process monitoring. In this paper, we present a framework for video-based monitoring of manufacturing processes with the help of a physical smart factory simulation model. We evaluate three state-of-the-art object detection systems regarding their suitability to detect workpieces and to recognize failure situations that require adaptations. In our experiments, we are able to show that detection accuracies above 90 % can be achieved with current object detection methods.

Keywords: Process Monitoring; Object Detection; Computer Vision; Machine Learning; Industry 4.0; Cyber-Physical Production Systems

1. Introduction

The industry is in a transformation towards more autonomous and intelligent manufacturing with machines, products, and humans that are increasingly connected via information technology. This process is known as *Fourth Industrial Revolution* (Industry 4.0) [13]. Developments in context of Industry 4.0 promise more efficient and flexible manufacturing processes, optimized supply chains, reduced downtimes and maintenance efforts for machines as well as cost reductions [25]. In general, production environments should act autonomously in order to react flexibly to changes in their corresponding environment [20].

* Corresponding author. Tel.: +49-651-201-2855; Fax: +49-651-201-3396.
E-mail address: malburgl@uni-trier.de

10.1016/j.procs.2021.04.009

The monitoring of the process execution in shop floors is indispensable to detect errors within the production at an early stage and, if possible, to correct them automatically to ensure on-time completion [20]. In current research work (e. g., [26, 10, 34]), mainly IoT data in form of discrete sensor streams (e. g., the state of light barriers, RFID readers, etc.) is used to monitor and detect erroneous situations in production processes. However, this type of sensor data is not always sufficient for deriving process-related events in every situation, e. g., for manual working steps or failure situations in which typical spatially restricted sensors can only hardly be used. In addition, the quality of sensor streams is sometimes insufficient and noisy [32], and the processing of large amounts of sensor data in near real time is a demanding task. Video-based methods such as *Object Detection (OD)* [21] can contribute to an enhanced monitoring of processes especially in failure situations. These methods do not require changes in the surface of objects such as QR codes, can analyze a larger contiguous area, and allow to complement or even replace other sensors. Nevertheless, video-based methods are still in their infancy and only used in specific areas (e. g., to detect manual working steps [12] or to evaluate the quality of produced pieces [30, 8]), and they are not used for monitoring of higher level processes.

The contribution of this paper is twofold: First, we present a general framework for using video-based methods to monitor the execution of manufacturing processes that can be combined with other methods for the data-driven analysis of smart manufacturing environments at runtime (e. g., *Complex Event Processing (CEP)* [29]). We then evaluate three state-of-the-art OD systems regarding their suitability as the core of the monitoring framework to detect workpieces in our physical factory simulation model. As a testbed for Industry 4.0 research, we use a physical factory simulation model from Fischertechnik (FT) (cf. [11, 16, 18, 26]). It enables us to simulate and monitor manufacturing processes taking into account actual physical properties of the production environment–especially w. r. t. runtime behavior and ad-hoc interactions with the physical world [4].

The paper is structured as follows: Sect. 2 presents the basics for our work and related work. The framework for video-based monitoring of processes is introduced in Sect. 3. The selection of OD systems from literature, the dataset and the training of the detection models, and the training results are presented in Sect. 4. A conclusion is given and future work is discussed in Sect. 5.

2. Foundations and Related Work

2.1. Physical Factory Simulation Model of an Industry 4.0 Environment

For the simulation of an Industry 4.0 manufacturing environment[1], we use a physical *Fischertechnik (FT)* factory simulation model. Such models are referred to as *Learning Factories* [1] and are used for education and Industry 4.0 research purposes (e. g., [11, 18, 26]). This allows to develop research prototypes and to assess their suitability for potentially use in practice. In contrast to purely virtual simulation models (e. g., based on Digital Twins [3]), we assume that physical simulation models are closer to the real world especially w. r. t. emerging runtime behavior and unanticipated ad-hoc interactions (e. g., by humans) not considered in the virtual models [4].

Our model consists of two shop floors each having 4 workstations with 6 identical and 1 individual machine as shown in Fig. 1. Each shop floor is equipped with several discrete sensors–light barriers, switches, capacitive sensors, and RFID readers–for monitoring purposes. The workpieces used for simulating manufacturing processes have the same cylindrical shape and size (height = ~1.4 cm, diameter = ~2.6 cm) and only differ in their color (red, white, blue). During the simulation of manufacturing processes, several failures can occur or be simulated, e. g., a workpiece may fall off a conveyor belt or a workpiece is positioned incorrectly on its lateral surface. Such errors are difficult to identify due to spatially restricted sensors of individual machines. Of particular interest for this work is the camera that is mounted centrally at a distance of 70 cm above the factory. The 2D camera module from Raspberry Pi V2 with 8 MP continuously records the entire factory from a static bird's-eye view with a viewing angle of approx. 90°. This camera provides the input video for the object detection system that is a central part of our process monitoring framework with an adequate resolution and quality to monitor almost all areas of the production line and its environment. In general, we use a service-oriented architecture based on semantically enriched RESTful web services to access the manufacturing capabilities of the smart factory on a (business) process-oriented level [16, 18].

[1] See https://iot.uni-trier.de for more details.

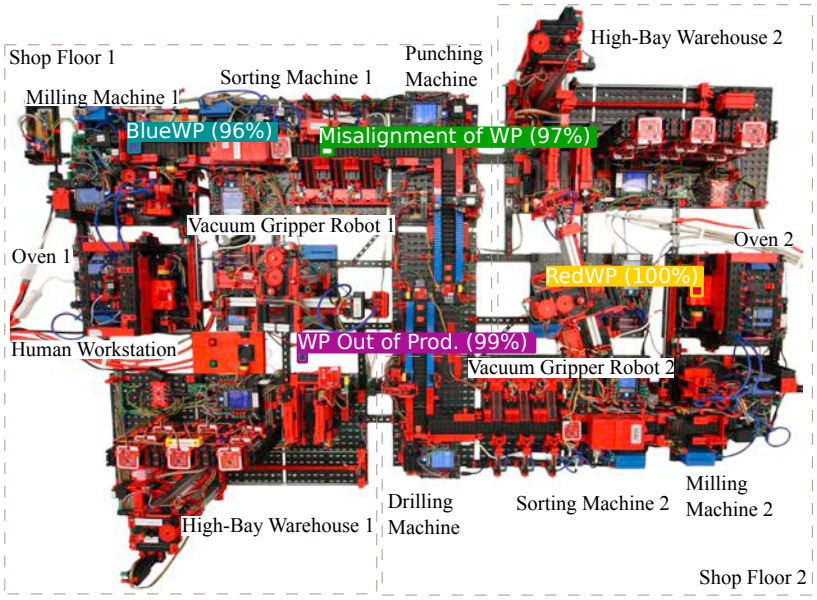

Fig. 1: The Two Fischertechnik Factory Simulation Models and Results of the Proposed Object Detection.

2.2. Object Detection (OD) Systems

The task of OD is to locate and classify predefined objects in a larger image by assigning each detected instance a *Bounding Box (BB)* [15] (cf. Fig. 1). OD systems can be divided into traditional systems using shallow, feature-engineered machine learning and systems based on *Deep Learning (DL)* [35]. As DL based detectors are the state-of-the-art since 2014 [9], we focus on this type of OD system. In DL based systems mainly *Convolutional Neural Networks (CNNs)* [15] are used as *backbone*, which have been typically pre-trained for image classification to learn meaningful general features (e. g., edges, lines, etc.), modified for OD, and then retrained to detect specific objects. For the actual OD, a *detector head* [9] is attached on top of the CNN backbone in order to re-use the feature representations extracted by the CNN as input for its BB predictions. These DL based detectors can be further distinguished between (i) two-stage detectors and (ii) one-stage detectors [33]. A two-stage detector such as Faster R-CNN [24] formulates OD as a classification problem by first selecting certain regions and then classifying them. In contrast, one-stage detectors such as YOLO [22] have no separate phase in which they generate region proposals. The backbone considers the entire input image and locates and classifies objects in one step. This means that OD is formulated as a regression problem that converts pixels into BB coordinates with class probabilities [22]. By dividing the detection process into two steps, two-stage detectors usually achieve higher detection accuracy but require longer time for processing [33] and thus could have a negative impact on real-time applications, which is an important requirement for our work.

2.3. Use Case and Requirements

The goal of our work is to develop a framework for video-based monitoring of manufacturing processes that can be used in the context of a smart factory. The main use case addresses the detection of workpieces and to determine their position at runtime in the production line–here of our smart factory model (cf. Sect 2.1). From this, exceptions as well as faulty and deviating situations in the shop floor can be derived in near real-time, e. g., situations in which a workpiece has fallen off a conveyor belt. Furthermore, these error and position information related to workpieces within the production environment can build the basis for adaptations to the production processes to resolve errors automatically [18, 27]. In the following, *requirements (RQs)* are described that must be met by OD systems to be part of our framework:

RQ 1 – Online Object Detection. The OD system supports the usage of a live stream as input for online detection.

RQ 2 – Recognition of Workpieces and their Color. The detected workpieces should be distinguished by characteristic features (here: their color).

RQ 3 – Detection of Misalignments and Position Errors within the Production. The OD system should be able to reliably detect faulty and erroneous situations. These include misalignments (e. g., when workpieces are standing on their lateral surface) or incorrect positions of the workpieces (e. g., outside of the production environment if they have fallen off conveyor belts).

RQ 4 – Visualization of Position Information of Detected Workpieces. The detected workpieces should be marked with their BBs. It should be possible to determine the position of workpieces within the production environment.

2.4. Related Work

Computer vision methods are already used in industrial contexts: (i) in quality control: Villalba-Diez et al. present an approach for industrial optical quality inspection in the printing industry [30]. This approach allows errors to be reliably detected and thus to reduce manual quality inspection costs. In [8], the authors use computer vision methods to monitor the assembly process of automotive lights in a reconfigurable work cell to detect failures in this process. Both approaches use computer vision to detect errors and quality issues in manufacturing steps. However, they do not aim at detecting individual workpieces and link potential errors to the entire production process–instead they focus on single process steps. (ii) in robotics: Mallick et al. present an approach and comparison of object detection methods for robotic picking tasks [19]. More precisely, they detect and localize products to provide the robot with accurate positioning information. (iii) for electronic components: in [7] and [14], the efficiency of state-of-the-art OD systems for detecting custom components on electrical boards (e. g., capacitors or inductors) is compared. Despite achieving suitable results w. r. t detection accuracy, the approaches do not use the OD methods for detecting failures that could occur during production. (iv) for detection of manual working steps: Knoch et al. [12] use the OD system YOLO to analyze a video stream to detect the hands of workers in manual assembly processes. The goal is to identify single process steps and to discover manual assembly processes. In contrast, we use an OD system to detect workpieces as a basis for monitoring entire manufacturing processes in a shop floor, i. e., the single process steps are distributed across several workstations. Additionally, erroneous situations during execution of processes should be detected to enable an automatic repair (cf. Requirements in Sect. 2.3).

3. Framework for Multi-modal Monitoring of Production Processes

State-of-the-art process control systems in industrial *Internet of Things (IoT)* settings mostly rely on sensors emitting discrete values (e. g., internal status of individual production machines, light barriers, switches, smart tags, etc.) to monitor the progress of the overall high-level processes within a production line [26, 10, 34]. While these sensor events are usually sufficient to detect the start and end of specific production steps or local failures related to individual machines, exceptions and errors regarding the workpieces at arbitrary points of the entire production line cannot be detected due to an insufficient sensor coverage. Here, video-based methods can be used complementary to the existing sensor infrastructure to provide a more comprehensive and robust process monitoring based on multiple modalities.

As one part of our contribution, we present a framework for video-based monitoring of manufacturing processes. Our proposal for such a multi-modal monitoring framework of production processes is depicted in Fig. 2. Complementary to the IoT sensor data emitted by the smart factory, the video stream(s) from available cameras is fed into an *Object Detection System* that processes the incoming video to detect the individual workpieces (cf. Fig. 2). Information about the workpiece detection is then forwarded to a *Tracking System* that adds unique identifiers to the individual objects and thus enables to distinction and tracking of the workpieces in the production line.

By combining the tracking results with an analysis of the IoT sensor data from other available smart factory sensors using e. g., *Complex Event Processing (CEP)* [26] to derive higher-level process knowledge, it is possible to reliably correlate individual workpieces to their corresponding process-level events and activities with a higher accuracy (cf. Fig. 2). In case of failures that cannot be captured by the local sensors or in cases of uncertainty about the overall process state, the object tracking system provides valuable additional information about the workpieces. This information can then be used to adjust and synchronize the current execution state of the process within the

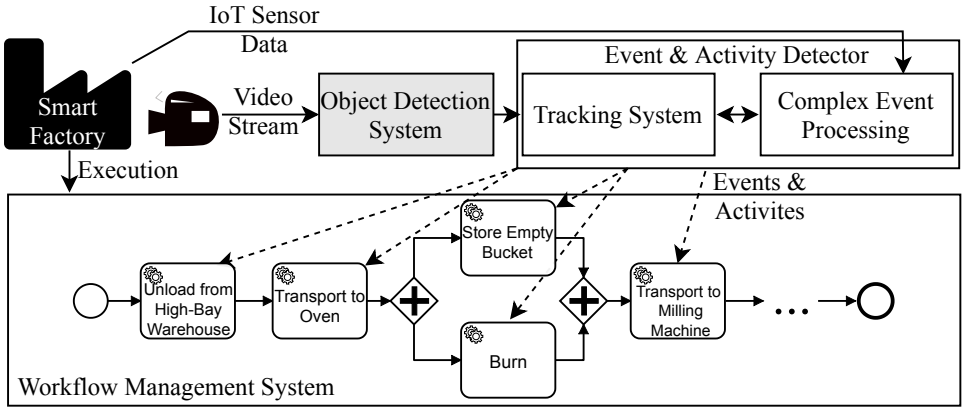

Fig. 2: Framework for Monitoring of Processes based on Multi-modal Sensors.

workflow management system executing and monitoring the high-level processes [27]. In case of faulty or deviating situations, automatic adaptations or optimizations of the processes can be derived and executed [18, 28].

While existing frameworks for process monitoring currently only consider discrete IoT sensors deployed in the smart factory [26], we propose to extend the process analysis with video-based monitoring. In the following section, we investigate the potential of state-of-the-art OD systems and evaluate them as a first step towards a holistic video-based process monitoring approach that can be easily integrated as additional modality into the proposed framework.

4. Application of Object Detection Systems in a Production Line

Since we intend to perform online object detection (cf. **RQ 1**), a fast inference but still a good accuracy (cf. **RQ 2** and **RQ 3**) of the OD system is crucial. However, this consideration results in a trade-off as in general, the accuracy decreases with increasing processing speed [6]. For this reason, we use the survey of Wang et al. [31] to identify and select suitable systems: *PANet (SPP) CSPResNeXt50* [31], *YOLOv3 (SPP) DarkNet53* [23], and its successor *YOLOv4 CSPDarknet53* [2]. These systems are all one-stage detectors and could presumably fulfill the four requirements from Sect. 2.3, which makes them applicable as OD system to be used in the proposed framework.

4.1. Datasets

The OD systems are typically trained and evaluated on the COCO dataset[2] that consists of common object classes like persons, birds, and bottles. Since these types of classes are quite different from our presented use case in the FT factory, we need to generate our own training data by recording videos with the Raspberry Pi camera mentioned in Sect. 2.1. Here we use the 4:3 format with a resolution of 1640 x 1232 pixels with a 10 FPS frame rate. The selected format allows to reduce the number of pixels to be filled with zero padding, lets the FT factory system appear relatively large, and ensures to be useful if the typical OD model input size is increased from 608 x 608 pixels.

Our data is divided into three sets: a training, a validation, and a test set. We ensured that the training set contains all common positions of a workpiece in the model factory that occur during the error-free execution of production processes. The validation and test sets are designed in such a way that it is possible to check how well the requirements **RQ 2** (detection of workpieces and their color) and **RQ 3** (detection of misalignments and position errors of workpieces) can be fulfilled. We divide the datasets according to the workpiece colors into three classes. Furthermore, we simulate two different types of failures–*workpiece out of production* and *misalignment of workpiece*–that occur frequently in our production environment and that are mostly not recognizable without video-based methods (cf. Fig. 1). All in all, the two failure types and the three different colors of workpieces result in five different classes for detection:

[2] https://cocodataset.org/

RedWP, *BlueWP*, *WhiteWP*, *WP Out of Production*, and *Misalignment of WP*. The basic training set consists of 780 images containing instances of the five different classes. Every class has approximately 600 instances (cf. Table 1).

Table 1: Number of Class Instances for Different Datasets.

Dataset	Number of Class Instances				
	RedWP	**BlueWP**	**WhiteWP**	**WP Out of Production**	**Misalignment of WP**
Training	560	574	574	628	598
Validation	684	592	625	75	75
Test	504	457	611	75	75

Similar to the training set, the validation set with a size of 450 images contains all common positions of a workpiece in the FT factory that occur during the error-free execution of production processes. The instances of workpieces differ from the training set because we want to investigate how well the model can detect workpiece positions that occur during regular process execution. There are about 600–700 instances for each color. The validation set contains 75 instances for each failure class, i. e., ∼10 % of all instances of a failure class for the entire dataset. 10 % is a typical proportion for the validation set. The test set is constructed similar to the validation set and contains 496 images.

To increase the training dataset artificially and make it more robust against visual variations such as brightness, data augmentation techniques are used. Huang et al. [7] distinguish between two different augmentation strategies: geometric operations and color operations. Since the FT factory is captured from a stationary camera perspective and class recognition is position-dependent, geometric operations for data augmentation are not considered valuable. For this reason, we only use the four color operations proposed in Huang et al. [7]: brightness modification, contrast variation, noise addition, and motion blur[3]. These four color operations are performed sequentially on each training image by randomly selecting the transformation parameters from a fixed interval defined by an expert to ensure that workpieces are still recognizable. In the end, on each image brightness is increased or decreased, contrast is increased or decreased, and noise or motion blur is added. From these transformations, the training set was extended with 780 additional images resulting in 1560 images for the full training set.

Furthermore, we used the image annotation tool *LabelImg*[4] to create BB annotations in the YOLO format that is used by the three trained detection models. To annotate the validation and test sets, pseudo-labelling was chosen to reduce the labelling effort by using a supervised learning model learned by the training data to label unlabeled data.

4.2. Network Training

The official Darknet implementations of YOLOv3 (SPP), YOLOv4, and PANet (SPP) CSPResNeXt50 are used[5]. We use 10,000 training iterations[6] with a default input format of 608 x 608 pixels. Hyperparameters such as learning rate, momentum term, (L2) regularization parameter, etc. are kept as default values. We use transfer learning by initializing the training with a pretrained Darknet53 backbone [23] for YOLOv3, a pretrained CSPDarknet53 backbone [31] for YOLOv4, and a pretrained CSPResNeXt50 backbone [31] for PANet. To avoid overfitting, we use early stopping by selecting the training weights that achieve the best performance on the validation set. The models are trained and evaluated on an Ubuntu environment with one Nvidia Tesla V100 with 32 GB graphics RAM, two Intel Xeon Gold 6138 processors with a total of 40 cores and 768 GB RAM. The approximate training time for a model takes 20 hours.

4.3. Results

We evaluate the OD systems by using the most frequently applied performance metric: the *Average Precision (AP)* [35]. The AP is calculated for a single class of a dataset that is why the mean value of the AP values of all

[3] We applied the image augmentation library *imgaug* (https://pypi.org/project/imgaug/).
[4] https://github.com/tzutalin/labelImg
[5] See https://github.com/AlexeyAB/darknet for more details.
[6] According to https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects for five classes.

classes–the *mean Average Precision (mAP)*–is applied to evaluate the accuracy over all classes. The AP summarizes the shape of the precision-recall curve in a key figure. We use the definition of the AP according to PASCAL VOC [5] that is common in practice and corresponds to the $AP^{IoU=0.5}$ from the set of COCO metrics.

Table 2: Average Precision for OD Systems.

OD System	AP of Non-Failure Classes			Mean	AP of Failure-Classes		Mean
	RedWP	BlueWP	WhiteWP		WP Out of Prod.	Misalignment of WP	
YOLOv3 (SPP)	0.9478	0.9255	0.9850	0.9528	0.8728	0.9854	0.9291
YOLOv4	0.9893	0.9250	0.9997	0.9713	0.9529	0.9665	0.9597
PANet (SPP) CSPResNeXt50	0.9588	0.9479	0.9988	0.9685	0.9163	0.9615	0.9389

Table 2 shows the results of the trained models according to the test data. For the different detectors, we compare the mAP of the three color classes and the mAP of the two failure classes to evaluate how well the requirements (detection of workpieces **RQ 2** and detection of misalignments and position errors of workpieces **RQ 3**) are fulfilled. In general, YOLOv4 is the most accurate detector with a mAP for the non-failure classes of 0.9713 and a mAP for the failure classes of 0.9597. PANet achieves a mAP for the non-failure classes of 0.9685 and a mAP for the failure classes of 0.9389. YOLOv3 is the least accurate detector with a mAP for the non-failure classes of 0.9528 and a mAP for the failure classes of 0.9291 in our experiment. However, all detectors are comparatively accurate and the differences between the mAPs are relatively small, i. e., only 3 % at maximum. Compared to related work for detecting objects in industrial context (cf. [7] with a mAP of 0.9521 and [14] with a mAP of 0.9307), we achieve similar detection accuracies. The ranking of the detection systems corresponds to the expected order since it corresponds to the ranking of detectors for detecting common objects based on the COCO test dataset (cf. [31, 2]). Regardless of the detector, workpiece colors are easier to recognize than failure classes. Blue workpieces are more difficult to recognize than red and white workpieces. White workpieces are by far the easiest to detect. This is probably due to the higher contrast between the objects and the background, i. e., the better the contrast, the better the colors should be recognizable. Workpieces located outside the production environment are more difficult to detect than misaligned workpieces. The difference between the APs of both failure classes decreases with increasing mAP for failure classes of the detector. However, this does not apply to the color classes, since PANet has a smaller difference between the APs of the three color classes than YOLOv4, although YOLOv4 achieves a higher mAP for the color classes than PANet. An exemplary workpiece detection is shown in Fig. 1 and a demo video showing how the detection and the later video-based process monitoring work can be found in [17].

All in all, the three evaluated OD methods meet all four requirements (see Sect. 2.3): Live streams are supported as input (cf. **RQ 1**), which is needed to enable online video-based monitoring of processes. Workpieces are recognized by their color (cf. **RQ 2**) that could be of interest to quickly determine the number of individual product variants. Typical failures such as misalignments and position errors that could occur during production (cf. **RQ 3**) are detected. This information provides the basis for resolving errors in production processes automatically (cf. Use Case 2 in [18]). Finally, detected BB information is visualized (cf. **RQ 4**) that gives a descriptive explanation to humans and that can be used to determine the position of workpieces in the production environment.

5. Conclusion and Future Work

In this work, we presented a general framework for multi-modal monitoring of manufacturing processes, and deployed and evaluated three state-of-the-art OD systems as a first step towards video-based monitoring and tracking. Our experimental evaluation shows that detection accuracies above 90 % for custom objects as well as failure positions of these objects in an industrial research context can be achieved. Based on these results, we expect that OD systems provide reliable and useful information that make them suitable for combination with other sensors in the context of IoT-based production process monitoring.

In future work, we will integrate the OD system with the other modules of our proposed process monitoring framework (cf. Fig. 2). The next step is to use the OD predictions and develop a tracking system for the workpieces that enables us to distinguish and identify different process instances. Additionally, the combination of our video-based

methods with other IoT sensor data such as light barriers and RFID reader/writers to track workpieces and to match them to a concrete process instance will be investigated. The use of semantic knowledge (e. g., an ontology [11]) in this process is also be part of future work.

References

[1] Abele, E., et al., 2017. Learning factories for future oriented research and education in manufacturing. CIRP Ann. 66, 803–826.
[2] Bochkovskiy, A., Wang, C., Liao, H.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR abs/2004.10934.
[3] Boschert, S., Rosen, R., 2016. Digital Twin—The Simulation Aspect, in: Mechatron. Futur.. Springer, pp. 59–74.
[4] Broy, M., Cengarle, M.V., Geisberger, E., 2012. Cyber-Physical Systems: Imminent Challenges, in: Large-Scale Complex IT Syst. Dev., Operat. and Manag. - 17th Monterey Workshop, Springer. pp. 1–28.
[5] Everingham, M., et al., 2010. The Pascal Visual Object Classes (VOC) Challenge. Int. J. Comput. Vis. 88, 303–338.
[6] Huang, J., et al., 2017. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors, in: Conf. on Comput. Vis. and Pattern Recognit., IEEE. pp. 3296–3297.
[7] Huang, R., et al., 2019. A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network. Electronics 8, 825.
[8] Ivanovska, T., et al., 2018. Visual Inspection and Error Detection in a Reconfigurable Robot Workcell: An Automotive Light Assembly Example, in: 13th Int. Conf. on Comput. Vis., Imaging and Comput. Graphics Theory and Appl., SciTePress. pp. 607–615.
[9] Jiao, L., et al., 2019. A Survey of Deep Learning-Based Object Detection. IEEE Access 7, 128837–128868.
[10] Kammerer, K., et al., 2020. Process-Driven and Flow-Based Processing of Industrial Sensor Data. Sensors 20, 5245.
[11] Klein, P., Malburg, L., Bergmann, R., 2019. FTOnto: A Domain Ontology for a Fischertechnik Simulation Production Factory by Reusing Existing Ontologies, in: Proc. of the Conf. LWDA, CEUR-WS.org. pp. 253–264.
[12] Knoch, S., Ponpathirkoottam, S., Schwartz, T., 2020. Video-to-Model: Unsupervised Trace Extraction from Videos for Process Discovery and Conformance Checking in Manual Assembly, in: BPM. Springer. volume 12168 of *LNCS*, pp. 291–308.
[13] Lasi, H., et al., 2014. Industry 4.0. BISE 6, 239–242.
[14] Li, J., Gu, J., Huang, Z., Wen, J., 2019. Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection. Appl. Sci. 9, 3750.
[15] Liu, L., et al., 2020. Deep Learning for Generic Object Detection: A Survey. Int. J. Comput. Vision 128, 261–318.
[16] Malburg, L., Klein, P., Bergmann, R., 2020a. Semantic Web Services for AI-Research with Physical Factory Simulation Models in Industry 4.0, in: Proc. of the Int. Conf. on Innov. Intell. Ind. Prod. and Logis. (IN4PL), SciTePress. pp. 32–43.
[17] Malburg, L., Rieder, M.P., Seiger, R., Klein, P., Bergmann, R., 2020b. Demo Video: Object Detection for Smart Factory Processes by Machine Learning. https://doi.org/10.6084/m9.figshare.13240784.
[18] Malburg, L., Seiger, R., Bergmann, R., Weber, B., 2020c. Using Physical Factory Simulation Models for Business Process Management Research, in: Del Río Ortega, A., Leopold, H., Santoro, F.M. (Eds.), Business Process Management Workshops, Springer. pp. 95–107.
[19] Mallick, A., Del Pobil, A.P., Cervera, E., 2018. Deep Learning based Object Recognition for Robot picking task, in: 12th Int. Conf. on Ubiquitous Inf. Manag. and Commun., ACM. pp. 1–9.
[20] Monostori, L., et al., 2016. Cyber-physical systems in manufacturing. CIRP Ann. 65, 621–641.
[21] Pathak, A.R., Pandey, M., Rautaray, S., 2018. Deep Learning Approaches for Detecting Objects from Images: A Review, in: Prog. in Comput., Anal. and Netw.. Springer. volume 710 of *Adv. in Intell. Syst. Comput.*, pp. 491–499.
[22] Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: Conf. on Comput. Vis. and Pattern Recognit., IEEE. pp. 779–788.
[23] Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. CoRR abs/1804.02767.
[24] Ren, S., He, K., Girshick, R.B., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Trans. Pattern Anal. Mach. Intell. 39, 1137–1149.
[25] Rüßmann, M., et al., 2015. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. Boston Consult. Group 9, 54–89.
[26] Seiger, R., et al., 2020. Towards IoT-driven Process Event Log Generation for Conformance Checking in Smart Factories, in: 24th Int. EDOC Workshop, IEEE. pp. 20–26.
[27] Seiger, R., Aßmann, U., 2019. Consistency and synchronization for workflows in cyber-physical systems, in: Proc. of the 10th ACM/IEEE Int. Conf. on Cyber-Physical Syst., pp. 312–313.
[28] Seiger, R., Huber, S., Heisig, P., Aßmann, U., 2019. Toward a framework for self-adaptive workflows in cyber-physical systems. Softw. Syst. Model. 18, 1117–1134.
[29] Soffer, P., et al., 2019. From event streams to process models and back: Challenges and opportunities. Inf. Syst. 81, 181–200.
[30] Villalba-Diez, J., et al., 2019. Deep Learning for Industrial Computer Vision Quality Control in the Printing Industry 4.0. Sensors 19, 3987.
[31] Wang, C., et al., 2020. CSPNet: A New Backbone that can Enhance Learning Capability of CNN, in: IEEE Conf. on Comput. Vis. and Pattern Recognit., IEEE. pp. 1571–1580.
[32] Wieland, M., et al., 2009. Towards Integration of Uncertain Sensor Data into Context-aware Workflows, in: Informatik, GI. pp. 2029–2040.
[33] Wu, X., Sahoo, D., Hoi, S.C.H., 2020. Recent advances in deep learning for object detection. Neurocomputing 396, 39–64.
[34] Zhong, R.Y., Wang, L., Xu, X., 2017. An IoT-enabled Real-time Machine Status Monitoring Approach for Cloud Manufacturing. Procedia CIRP 63, 709–714.
[35] Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. Object Detection in 20 Years: A Survey. CoRR abs/1905.05055.