Journal on Data Semantics manuscript No.
(will be inserted by the editor)

Sensor-Based Human-Process Interaction in Discrete Manufacturing

Detection and Validation of Manual Assembly Tasks with Safe Flexibility

Sönke Knoch · Nico Herbig · Shreeraman Ponpathirkoottam · Felix Kosmalla · Philipp Staudt · Daniel Porta · Peter Fettke · Peter Loos

Received: date / Accepted: date

Abstract The rise of Industry 4.0 and the convergence with BPM provide new potential for the automatic gathering of process-related sensor information. In manufacturing, information about human behavior in manual assembly tasks is rare when no interaction with machines is involved. We suggest technologies to automatically detect material picking and placement in the assembly workflow to gather accurate data about human behavior and flexible support of human-process interaction. The detection of material picking is achieved by using background subtraction in combination with scales. For placement detection, two approaches are tested: image classification using convolutional neural networks and object detection using Haar wavelets. The detected fine-grained worker activities are then correlated to a hybrid model of the assembly workflow using BPMN and CMMN, enabling the measurement of production time (time per state) and quality (frequency of error) on the shop floor as an entry point for conformance checking and process optimization. The approach has been evaluated in a quantitative case study recording the assembly process 30 times in a laboratory setup within four hours. Under these conditions, the classification of assembly states using a neural network provides a test accuracy of 99.25% on 38 possible assembly states. Material picking based on background subtraction has been evaluated in an informal user study with six participants performing 16 picks each, providing an accuracy of 99.48%. The suggested method offers a promising approach to easily assess fine-grained timings and error rates of assembly steps which can be used to optimize the corresponding process.

German Research Center for Artificial Intelligence (DFKI), Saarland Informatics Campus, Saarland University

Keywords Manual Assembly · Activity Detection · Computer Vision · Process Enhancement · Industry 4.0.

1 Introduction

The current trend of automation and data exchange in manufacturing, known under the term Industry 4.0 (Kagermann et al., 2013; Lasi et al., 2014), addresses the convergence of the physical and the virtual world. It comprises the introduction of cyber-physical systems (CPS), Internet of Things (IoT) and cloud computing in a fourth industrial revolution, where manufacturing companies face volatile markets, cost reduction pressure, shorter product lifecycles, increasing product variability, mass customization leading to batch size one and, with rising amounts of data, developments towards a smart factory (Cavanillas et al., 2016).

To plan, construct, run, monitor, and improve a flexible assembly work station or CPS tackling the challenges of Industry 4.0, engineers and managers require detailed information about the assembly workflow in the life-cycle phases of a CPS, see for example Thoben et al. (2014). In workflows with manual tasks, this information contains data on human behavior, such as grasp distance, assembly time, and the effect the workplace design has on the assembly workflow. It can be used to plan and construct efficient assembly work stations and receive information on the executed workflows to establish a continuous improvement process (CIP/Kaizen) within the organization. As of today, data has to be approximated or gathered manually which consumes time and money.

We see a large potential in the technical support and automatization of data gathering processes providing information about a workflow's execution regarding

time and quality. The accurate detection and validation tasks in the assembly workflow, and as a consequence, the generation of meaningful events correlated to that workflow, is a challenging task. On the hardware side, the selection of appropriate sensors, their integration into the assembly system and the robustness against changing conditions in the manufacturing context have to be considered. On the software side, these sensors have to be used to reliably detect activities, deliver results fast, and provide complete information in a homogeneous data format. If the physical setup, the software configuration, and the operation of such a system efficiently support the workflow execution, this will add value to the organizations deploying them: (1) Process models will be enriched with detailed information about the assembly steps. (2) Live workflow tracking enables online conformance checking. (3) Further analysis of workflow traces can be used to adapt and optimize the workflow execution.

In a first iteration, an artifact, fast and easy to set up in terms of configuration and instrumentation, was developed integrating multiple sensors to analyze hand and body posture, as well as material picking. For this we used ultrasonic distance sensors, as well as cameras based on infrared, RGB+Depth, and RGB images (Knoch et al., 2016). This proof-of-concept implementation was then extended and evaluated in a case study with 12 participants in a second iteration (Knoch et al., 2018). In (Knoch et al., 2019), we reduced the sensor setting to the most promising sensors (two RGB cameras and one hand sensor) applying new methods from computer vision and machine learning to achieve a high resolution in recognizing assembly tasks. Task-related events were correlated to process tasks in a BPMN model.

In this article, which is an extension of (Knoch et al., 2019), we added a brief introduction about the context of use and the relevant terminology in Section 2. Section 3 describes related work from the field of BPM and activity recognition. It is extended with work from BPM at the intersection with IoT. Section 4 introduces the concept split into the activity detection (4.1) and the process model (4.2) part. We add scales to consistently check the inventory at the assembly work station, investigate the potential of object detection for state classification, and use hybrid process models controlling the manual assembly workflow flexibly based on capability descriptions. Thereby, workers interacting with the process achieve more freedom during assembly and managers can adapt the process faster using prepared process snippets controlling the detection and validation in a flexible but - through activity detection - safe way. Using a combination of imperative (BPMN) and declarative (CMMN) process notations flexible work step orders are supported by defining only the checkpoints a worker needs to pass by. Different worker roles such as trainees and experts are allowed to have a different level of freedom following a different model during assembly without loosing the safety of sub-task validation. Sections 5 and 6 contain the implementation and evaluation of this adapted concept, followed by a discussion of the results in Section 7. The paper is concluded in Section 8.

2 Context

A discrete manufacturing process produces distinct items and can be executed individually. The product in such a process is made from single or multiple inputs, which are material parts, components, and sub-assemblies. Manufacturing follows the job, batch, or flow production principle. Manual assembly work stations can be found in both job (individual assembly of items) and batch (assembly of items in batches) production. Manual assembly work stations are part of assembly lines or arranged in slots where the product is assembled partially or completely. Independent from the applied production principle, manual assembly work is used to repair defective products after negative inspection. During assembly, workers are assisted by worker guidance systems (WGS), providing instructions on different levels of granularity based on the worker's skill (Kerber and Lessel, 2015). It is common practice that work steps have to be confirmed by the worker manually. While the level of process standardization is high during production, during a repair process, assembly workflows are more flexible due to the variance in errors. The process model controlling such an assembly has to support both flexible and strict processes.

Planning an assembly workflow involves the definition of task and flow structures, and the elicitation of assembly times. The structure of the task divides a work-step into a reasonable number of sub-tasks. The structure of the flow defines order and relations between work steps, such as parallel and sequential assembly, in a precedence graph. During execution, short-termed controlling and monitoring tasks, such as failure handling, supply of materials, and elicitation of information about the manufacturing progress, have to be carried out by the controlling process instance. The process model needs to fulfill both the requirements of describing the workflow and handling events during run-time. Abstract process descriptions based on the required capabilities make these models more robust against changes regarding the concrete hardware and software.

A manual assembly task is split into different subtasks. In the field of assembly time elicitation, a standardized terminology is used to label such a manual task. Methods Time Measurement (MTM)[1] defines five basic motions within an assembly step: reaching, grasping, moving, positioning, and releasing a material part. The activity detection modules suggested in this work are embedded in the flow of basic motions defined by MTM. In the process model, they are used to define the capability that is requested in one task. Both enables the gathering of timing for basic motions and the abstract description of process tasks based on a standardized terminology (MTM). Each activity module implementing such a capability is automatically mapped to the process without any adaptions in the model that would become necessary, for example when the activity module has changed or the model is executed on a different work station.

Detection of human activities and their correlation to the correct process instance support automatic time elicitation, workflow tracking, exception handling, and just-in-time material supply. Therefore, data from order management, the work plan, and the bill of materials have to be correlated with activities detected by sensors equipped to an assembly work station. The selection, positioning, and accuracy of the sensors strongly affect the tracking resolution and involve set-up and training times (supervised machine learning).

3 Related Work

Related work can be found at the intersection of BPM with cognitive computing, context-awareness, IoT, and human activity recognition, here mainly vision-based.

Cognitive BPM (CBPM) CBPM comprises the challenges and benefits of cognitive computing in relation to traditional BPM. Hull and Motahari Nezhad (2016) suggest a cognitive process management system (CPMS) to support cyber-physical processes enabled by artificial intelligence (AI). Marrella and Mecella (2017) propose the concept of such a CPMS which automatically adapts processes at run-time, taking advantage of the AI's knowledge representation and reasoning. Similar to our system the alignment between real physical and expected modeled behavior can be measured. In *context-aware BPM*, the user behavior is set in relation to situations affecting the behavior. Transferred to our approach, user behavior corresponds to detected worker activities in the context of a concrete assembly task from a process model instance executed at a concrete assembly work station. Jaroucheh et al. (2011) apply linear temporal logic and conformance checking from process mining to compare real with expected user behavior. Since the collection of high resolution data for the aforementioned systems is not a trivial task, augmenting the process with information captured by cognitive computing, such as computer vision, is the focus within this research activity.

IoT and BPM The manifesto by Janiesch et al. (2017) indicates potential benefits of IoT from BPM and vice-versa. Our approach tackles many of the challenges defined by the authors, such as 'placing sensors in a process-aware way', 'detecting new processes from data' and 'bridging the gap between event-based and process-based systems'. Most approaches in that field focus on process discovery and due to data availability in the smart living domain skipping the step of gathering data discussed in our work. Cameranesi et al. (2018) and Sora et al. (2018) present an approach to discover process models from activities in an ambient assisted living scenario and in the field of smart spaces. A data set from a smart home scenario was chosen to extract daily activities. In Cameranesi et al. (2018), aggregated macro-activities characterizing daily user behavior have been chosen as input for a process mining algorithm. In a similar fashion Carolis et al. (2015) apply first-order logic to learn daily routines of users in smart home environments. In their approach, the authors suggest a new process mining technique able to learn complex models efficiently and in a way that is applicable on-line. Since we assume an assembly workflow with restricted degrees of freedom, our focus is the handling of undesired behavior (errors) and the analysis based on predefined metrics measuring time and quality across process instances. The application of process mining remains an interesting field for future work.

Vision-based human activity recognition (HAR) In HAR the main focus is on 2D video data and applies various machine learning methods (for an overview see Poppe (2010)). HAR applications in manufacturing are sparse but occur more frequently since the rise of IoT, industrial internet, and Industry 4.0. Within the field of human robot collaboration, Lenz et al. (2011) use 3D video data to determine the hand position of a worker sitting in front of an assembly work table and collaborating with an assistive robotic system applying Hidden Markov Models (HMM). Two cameras are mounted to the assembly workstation and are calibrated to each other. High calibration effort facilitates the application of the method only in restricted setups. Similarly, Roitberg et al. (2015) apply hierarchical HMMs in a

[1] `mtm-international.org`

multi-modal sensor setting within the same domain. They combine RGB-D (Kinect 2) and IR (Leap Motion) cameras to detect fine-grained activities (e.g. assembly, picking an object, fixing with a tool) and gestures (e.g. pointing, thumb up) and analyze the results of the sensors individually and in combination. Combinations of sensors show the best average recognition results for activities in most cases. Unlike these approaches focusing on gesture detection, we concentrate on the detection of assembly states in the form of image classes allowing the connection to state transitions in process models. It is an example how AI can be applied to ensure process quality and adherence to time constraints.

HAR using wearable sensors The authors in Grzeszick et al. (2017) use convolutional neural networks (CNN) on sequential data of multiple inertial measurement units (IMU). Three IMUs are worn by workers on both wrists and the torso. This way, an order picking process in warehouses can be analyzed, fusing data from all sensors to classify relevant human activities such as walking, searching, picking and scanning. In a similar fashion Stiefmeier et al. (2008) apply different on-body sensors (RFID, force-sensitive resistor (FSR) strap, IMU) in a "motion jacket" worn by workers, and environmental sensors (magnetic switches and FSR sensors) to detect activities in the automotive industry, such as inserting a lamp, mounting a bar using screws and screwdriver, and verifying the lamp's adjustment. We decided to use contactless activity detection and avoid the instrumentation of workers, because we expect limited user acceptance when wearing sensor equipment. In addition, using wearables the reloading of batteries and mechanical signs of fatigue during usage may be issues the operator wants to avoid. Thus, an easy and light setup is suggested, applying state-of-the-art AI technology to the problem of workflow tracking in manual assembly enabling conformance checking and optimization.

4 Concept

BPM has a high potential to support the implementation and adaption of process models which control and sense the assembly workflow. We provide a concept showing how recognized events from the shop floor can be correlated to tasks in the model based on capability descriptions, allowing the supervision of time and quality constraints defined ex-ante in the form of reference times and material states. In addition, the collection of timing information and error frequency supports the optimization of assembly workflows, e.g. through conformance checking. To enable this kind of supervision and optimization, the detection of critical activities within one work step is necessary. The setup is designed as light-weight and can be easily used to equip any assembly work station in a short time.

4.1 Activity Detection

Figure 1 shows the three-tiered activity detection: first, grasps to container boxes are detected to analyze which part the worker assembles next (*Grasp Detection*). In parallel, the amount of pieces removed from the respective box is verified by the *Inventory Control*. After performing the actual assembly step, the worker places his hands next to the workpiece, where they are detected by the *Hand Detection* module. At this point in time, an image is captured, as there cannot be any occlusions or motion blur. This image is then used for *Material Detection*, analyzing which workstep the assembly led to, i.e. if the step was correct and the next assembly step can be taken or whether one of several possible failure states is reached and a recovery strategy needs to be applied.

4.1.1 Grasp Detection

Grasp Detection involves the appearance of a foreground object (user's hand) over a stationary background (formed by the image of the container box) seen by the camera observing the container box from the top. It is assumed that this kind of activity occurs when a part is removed from its designated container box. Therefore, activity zones have to be predefined within the image when setting up the camera to detect the worker's hand. In the process model, material grasping and grasp detection form the first step after receiving the instruction for the current assembly task. Such a system facilitates the supervision of assembly workstations equipped with a lot of similar-looking materials without the effort of hard-wiring the system with the workstation, as in case of classical pick-supervision.

4.1.2 Inventory Control

Inventory or stock control is capable of checking the number of material parts in stock. An inventory system permanently tracking the number of parts can be used to verify the removal of the correct amount of pieces needed within one assembly task. Since bulk material is hard to detect with optical sensors, another sensor has to be employed allowing the accurate measurement of small materials such as screws or nuts. A high precision scale carrying a material box supports such a relative

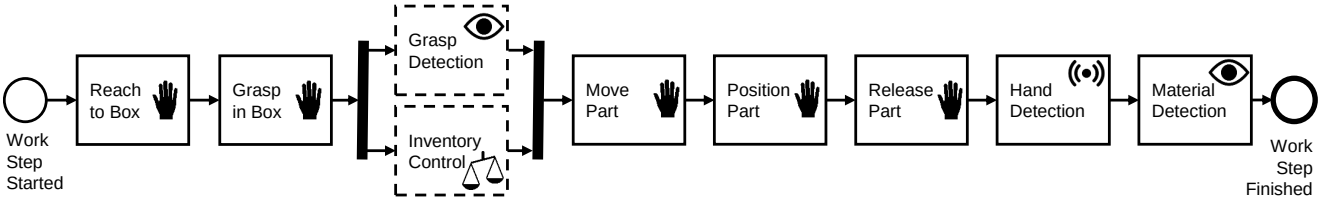

Fig. 1 Flow chart representing the activity detection process within one work step following the MTM basic motion flow; dashed rectangles indicate optional validation (depends on worker skill).

measurement once the weight per piece is known and approximately equal among material pieces of one type. If these constraints are met, the weight-based counting of pieces can be solved using a scale with an adequate resolution. The remaining challenge is the filtering of noise created by vibrations during material removal. Although the noise is an indicator for the grasp activity, grasp detection remains necessary to achieve more accurate time estimates.

4.1.3 Hand Detection

Hand Detection allows the active confirmation of work steps by the worker without reaching to distant touch screens or buttons. With correct and complete execution, this enables positive feedback and forms the point in time when image data for material detection is gathered. For the worker, this procedure offers more safety, since subsequent work steps are only started after successful confirmation of the previous ones. In the case of a mistake, the operator is supported with fine-grained assistance. For later material detection, this approach ensures that no hands occlude the workpiece and that the workpiece does not move, thereby avoiding motion blur.

4.1.4 State Classification

State Classification can be used on the acquired image to confirm whether the material parts being grasped are correctly assembled in the desired way. This involves verifying if each part is correctly located in the expected orientation. A camera is used to monitor the assembly region (where assembly of product is being performed) which facilitates verification and quality assurance based on the image information. This can be achieved in two ways: (1) *image classification* considers the whole image for estimating the product's state, and (2) *object detection* detects each individual object's position and orientation in an image to deduce the overall assembly state. Both approaches require the gathering of training data.

Process-based Gathering of Image Data Training of a classifier or object detector usually requires manual labeling of a large amount of data samples. This might become an obstacle when the suggested method for quality assurance based on image data is applied in an organization. Thus, an approach is needed to quickly get the system running in practice. The approach of process-controlled activity detection suggested in this paper can solve this challenge. A slightly modified model can be used to control the training process and, due to the correlation between process instances and sensors, to automatically gather and annotate image data. Remotely controlled light bulbs equipped to the assembly work station are used to generate different lighting conditions which is necessary to achieve a robust material detector. The training images can be acquired either in a separate gathering process (as done for the presented evaluation), or confirmations/correction input to the WGS can be used to label images with corresponding classes, thus supporting cost-neutral data capture in the wild.

Image Classification Image classification is a well known task with many out of the box solutions. The validation of an assembly state can be solved with image classification which has a very vivid community generating competing results of high quality, for example within the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). Each material state in the process model, including all error states, is considered as a separate class. For each of these classes, a set of images is acquired to train a classifier according to the approach described above. Whenever an image is captured after hand detection, this image is classified to the corresponding state in the BPM, which allows jumping to the next step in the WGS, a system showing instructions about the current assembly task on a screen, or intervening in case of error states.

Object Detection The verification of the assembly of the parts can also be treated as a detection and localization problem. The entire image or a predefined region is searched for the specific part. A separate object detector is trained for each part independent of

the other parts. Therefore, the labeled training data from above can be used in a slightly modified way by cropping the respective part from the image. Based on the label from the process, the name and orientation of the work piece are known, allowing the definition of a bounding box's position and size. Once defined, the cropping is processed automatically by applying the coordinates and dimensions to all other images (since the camera and the assembly zone are fixed). During execution of the detector, since the material of interest is known from the grasp signal, only the respective object detector is run on the image to improve efficiency. Information about the location of the detected part is used to check the correctness of the assembly, taking into consideration the previous process step.

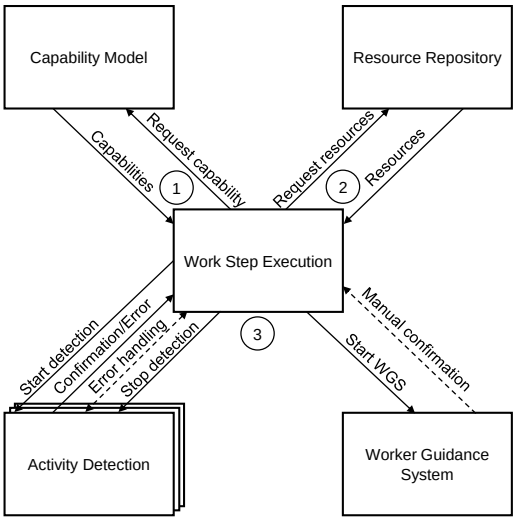

Fig. 2 Logical capability checking (1), operational matching (2), and device control (3).

4.2 Process Model and Correlation

A correlation between a physical thing and a modeled workflow can be tight or loose according to (Wombacher, 2011). In a tight correlation, process model and sensors depend on each other, while in a loose correlation, process steps are only used as synchronization points. We are aiming at a tight correlation, since it allows the process to control the senors and to enhance the process model with fine-grained information about the process execution. Nevertheless, it requires the model to be more complete regarding error handling and case modeling.

At the beginning of each process, order information and work plan data from manufacturing information systems is fetched and used to set variable properties such as material types and the amount to assemble in each step. The model contains the logic and is configured with a set of properties in a graphical process model editor, which allows the coupling of events to the model without the specification of concrete devices and due to the generic implementation without further software implementation effort. The binding between model and components in the assembly workflow is based on a topology describing the value range of the property variables necessary to run a concrete workflow.

4.2.1 Abstract Process Descriptions

Abstract descriptions of manufacturing processes consist of a sequence of capability requirements. These capability requirements have to be mapped to the manufacturing equipment on the shop floor. An efficient automatic alignment of capability requirements and resource type warrants is achieved, applying light-weight semantic procedures from transaction processing in dialog systems, such as algorithms for unification and pattern matching.

Figure 2 shows the three phases necessary to execute a work step from an abstract assembly process. First, a set of potential operational resources is determined based on descriptions in the capability model. The resulting resource asserting the requested capability are checked in a second phase for the availability and state. Finally, if a resource was selected the detection and guidance process can start exchanging the messages described in Subsection 4.2.4.

CapabilityRequests are divided in WorkerAssistance and WorkStepConfirmation requests. Both requests contain one of the HandlingCapabilities Grasp, Release, or Position, according to the MTM terminology. Based on the data from the bill of material in the instantiated process, both these capabilities are instantiated with a list of materials relevant in the current work step. The grasp capability receives additional information about the required material quantities to validate the removal of the correct number of pieces.

4.2.2 Process Flexibility

Workflow and business process notations, such as BPMN, were designed for modeling processes without much variation as known from mass production. Models described in such flow-based, imperative process notations become very confusing if many cases are modeled and can end up in so called spaghetti models. Constraint-based, declarative process notations are suited for the modelling of flexible processes by focusing on rules and constraints of the process instead on the actual flow. In combination with BPMN, we use the Case Man-

agement Model and Notation (CMMN) based on the Guard-Stage-Milestone model. Modeling work steps in such a flexible way, allows for the definition of tasks during assembly without specific definition of their logical or temporal relation to other tasks in one step. Workers following such a flexible process can assemble the product without satisfying a strict order. In addition, the strictness of the model reflected in the formulated constraints can be varied, e.g. based on the experience level of the worker.

4.2.3 Activity-to-Model Correlation

In order to match an activity to an instance of a task two approaches exist: cost-based and key-based matching. Cost-based matching uses information about the distance between an activity and a task instance based on the task description and the time when the event occurred. A strong content-based similarity between the description of the activity and the task, and, on the other side, a strong proximity between the occurrence in time of both activity and task instance, indicate a match. Key-based matching uses an identifier to correlate activities and task instances, thereby avoiding potential matching errors as in the first approach. In the following, we rely on key-based matching to achieve robustness against any form of matching errors.

4.2.4 Messages

Messages are divided into (a) *instructions* published by the model controlling software components and (b) *activities* published by software components detecting material picking and placing or user input from the WGS. Every message contains the business key identifying the process instance with an universally unique identifier (UUID) and a time stamp. Thereby, key-based matching is supported since the key is stored in the receiving Activity Detection application and sent back when the activity was detected. Instructions are subdivided according to the destination (*WGS*/*ActivityDetection*).

WGS instruction An instruction controlling the WGS contains the mandatory property work step index (used to gather descriptive information and media to explain and present the current assembly task to the user), a list of expected materials, its amount, and a list of expected orientations. Optionally, an error is set when the list of found materials or found orientations does not reflect the expected ones, e.g. when *WrongMaterial* or *WrongOrientation* occurs. Then, a correction instruction is presented on the WGS:

```
{
  "businessKey":"3af444fd
     -5132-4211-9391-b1d1a027a390",
  "timestamp":"1488412740302",
  "workStep":3,
  "expectedMaterials":[{
        "materialName":"
           ConnectingBoard",
        "amount":1},{
        "materialName":"
           ApplicationBoard",
        "amount":1}],
  "expectedOrientations":
        ["East","West"],
  "error":{
    "errorName":"WrongMaterial",
    "foundMaterial":"Mainboard",
    "foundAmount":1
  }
}
```

ActivityDetection instruction An instruction controlling the activity detection contains the expected materials, orientations and the assembly activity, e.g., $grasp_t$, $insert_t$, or $rest_t$, where t is the reference time for one activity. The time t can be used to measure the deviation from the manufacturing time planned. To control software components, we introduce the *action* property that contains the state of the respective component, e.g. “start” or “stop”. The sample message starting the grasp detector can be seen in the following listing:

```
{
  "businessKey":"3af444fd
     -5132-4211-9391-b1d1a027a390",
  "timestamp":"1488412681045",
  "material":"ConnectingBoard",
  "activity":"grasp",
  "action":"start"
}
```

ActivityDetection event is sent when the activity detection was started by the controlling process instance and an activity was detected. Then, the activity detection module delivers the properties material (expected & detected), orientation (expected & detected), activity, and a type, which indicates if the source of the detection was from an *automatic* (activity detection) or *manual* (confirm button) origin. Since activity detection might fail, it is always possible to confirm an activity by pressing a button on the WGS screen as a

fallback. In the following listing the message representing an automatically detected grasp into the material box containing connecting boards is shown:

```
{
  "businessKey": "3af444fd
     -5132-4211-9391-b1d1a027a390",
  "timestamp": "1488412770298",
  "expectedMaterials":[{
        "materialName":"
           ConnectingBoard",
        "amount":1},{
        "materialName":"
           ApplicationBoard",
        "amount":1}],
  "foundMaterial":{
        "materialName":"
           ConnectingBoard",
        "amount":1},
  "activity":"grasp",
  "type":"automatic"
}
```

5 Implementation

The implementation of the concept presented in the previous section requires an example assembly workflow which was set up in the lab (Subsection 5.1). It involves an hybrid process model implemented in BPMN 2.0 and CMMN 1.1 controlling and tracking this workflow (Subsection 5.2), and four activity detection modules detecting material grasping, inventory changes, hands, and material positioning realized in two approaches (Subsections 5.3–5.8).

5.1 Assembly Workflow and Apparatus

To test and iteratively improve our system, we designed an assembly workflow consisting of four assembly steps. The product to be assembled consists of three printed circuit boards (PCB) to be connected and placed in one 3D-printed case. The four materials are provided in small load carriers (SLC), boxes common in manufacturing when dealing with small parts. In steps 1 to 3, material is removed from the SLCs and assembled in the work area on top of the workbench in front of the worker. In step 1 the case is removed and placed with the open side up in the work area. In step 2 one PCB is removed and inserted into the case. In step 3 the two remaining PCBs are removed, connected and inserted into the case. The removal of two parts in parallel is common practice to improve the efficiency in assembly workflows. Finally, within step 4 the assembled construct is removed from the work area and inserted into a slide heading to the back of the assembly workstation. The whole workflow is supported by a basic worker guidance system (WGS) running on a touch screen and showing textual instructions and photos of the relevant materials and of the target state in the respective assembly step.

The assembly workstation, shown in Figure 3, is made from cardboard prototyping material and instrumented with two consumer-electronics RGB cameras: a Logitech C920 HD Pro and a Logitech BRIO 4K Ultra HD. The first camera is mounted on top of the assembly station and pointed at the four SLCs loaded with material. The second camera is mounted to the shelf carrying the SLCs and is aimed at the work area. Four Mettler Toledo weighting pads with a resolution of 1 mg are placed under the SLCs to permanently track the material weight.

5.2 Process Model

A process model, modeled in the BPMN language and shown on top of Figure 4, controls the execution of an assembly order and the respective activity detection modules and the WGS supporting the current order. It consists of service, call, send and receive tasks. Service tasks fetch order and work plan information and initialize a new work step setting the relevant parameters necessary for the subsequent work step. Here, we set the product ID, variant name, materials from the BOM and the index i of the step. The call task referes to a CMMN 1.1 model describing the assembly of one work step in a flexible way. Send tasks control components necessary to execute the work step and generate *instruction* messages sent to a target component c. Receive tasks wait for acknowledge events confirming an *activity* a in the current work step. The model is instantiated for one product and loops over all work steps defined in the work plan.

The call task initializes the work step CMMN model shown on the left of Figure 4. It consists of optional and non-optional process tasks all linked to a specialized BPMN process model controlling the activity detection processes in detail (models on the right). In the case model for a work step, grasp validation is modeled as an optional task. The release and position validation tasks are modeled as mandatory tasks, since they are used to ensure the quality of the final product. Reflecting the fact that the worker has to confirm the assembly state manually by placing her hands before the material state is checked, hand detection has to occur before material

Fig. 3 Assembly workstation equipped with 1 touch screen, 2 cameras and 6 light sources (middle); example material states (of the 38) and 7 light scenes for one part (left); inventory control with weighting pad, and grasp detection using background subtraction with the 'U'-shaped activity zones (right).

detection can be conducted. This is expressed by the entry criterion (rhombus) interconnecting both tasks.

Each activity detection module, except grasp detection which is combined with inventory control, is controlled by one BPMN process model consisting of send and receive tasks. Within a send task of type start, the activity to observe is defined. When an activity occurs, two options exist: (1) the correlated event occurs as expected and the activity is stopped by the subsequent send task, or (2) a correlated activity occurs that detects a behavior that does not match the expected state. Then, the exception handling is started, instructing the worker to correct his activity. Again, a call task in the BPMN refers to a CMMN model handling the error in case the wrong material or the wrong quantity was grasped or in case a wrong material was detected after positioning and releasing it. Finally, when the process ends, the process task in the CMMN is informed and continues. Figure 4 on the right shows the model controlling hand, material and grasp detection with inventory control. The case models for exception handling can be seen on the left.

Error handling is important when processes and sensors are tightly correlated and allows the system to intervene when states are detected that have been classified as erroneous. In total 38 classes cover all relevant states including errors that may occur during assembly at this work station with the existing material parts by rotating parts on their positions. Thereby, every assembly state is covered and can be handled by the process. For training, every state can be generated walking step by step through the process to gather all data required for this classification task. The BPMN model was redesigned to instruct the user about the arrangement of materials in the training phase of the system. Within each instruction step the material state is confirmed such that images of this state can be taken under a variety of lighting conditions automatically generated.

5.3 Grasp Detection

The activity zones, shown in Figure 3, are marked outside the interior of the container boxes because a background subtraction algorithm adapts to a changing background. The background (interior of the box) changes, when a part is removed from its container box. The activity zones are 'U'-shaped since the direction of approach of the hand is not always perpendicular to the breadth of the container box. The red regions are marked a single time during system setup and the yellow activity zones are automatically identified as a corollary.

Given a video sequence $V(t)$, the frame denoted as $I(t)$ is compared pixel-wise against a pixel-wise model built by the background subtraction algorithm by Zivkovic and Van Der Heijden (2006). The model is updated every frame so that it captures the recent history of values taken by the pixels. This history is controlled by the parameter α. The background subtraction algorithm shown in Algorithm 1 marks each pixel as active or inactive. The procedure of detecting the start and end of a grasping activity is as follows: by default there is inactivity in the activity zones. When the number of foreground pixels $|F(t)|$ (pixels classified as active) exceeds the total number of pixels P in the activity zone a by a certain user-defined value ($p_a = 40\%$), activity is said to occur in the respective zone. To filter noisy signals, the start and end of an activity is detected once there is a considerable number of video frames with activity/inactivity respectively. Hence in a user defined recent history of frames (set to 10) denoted by the pa-

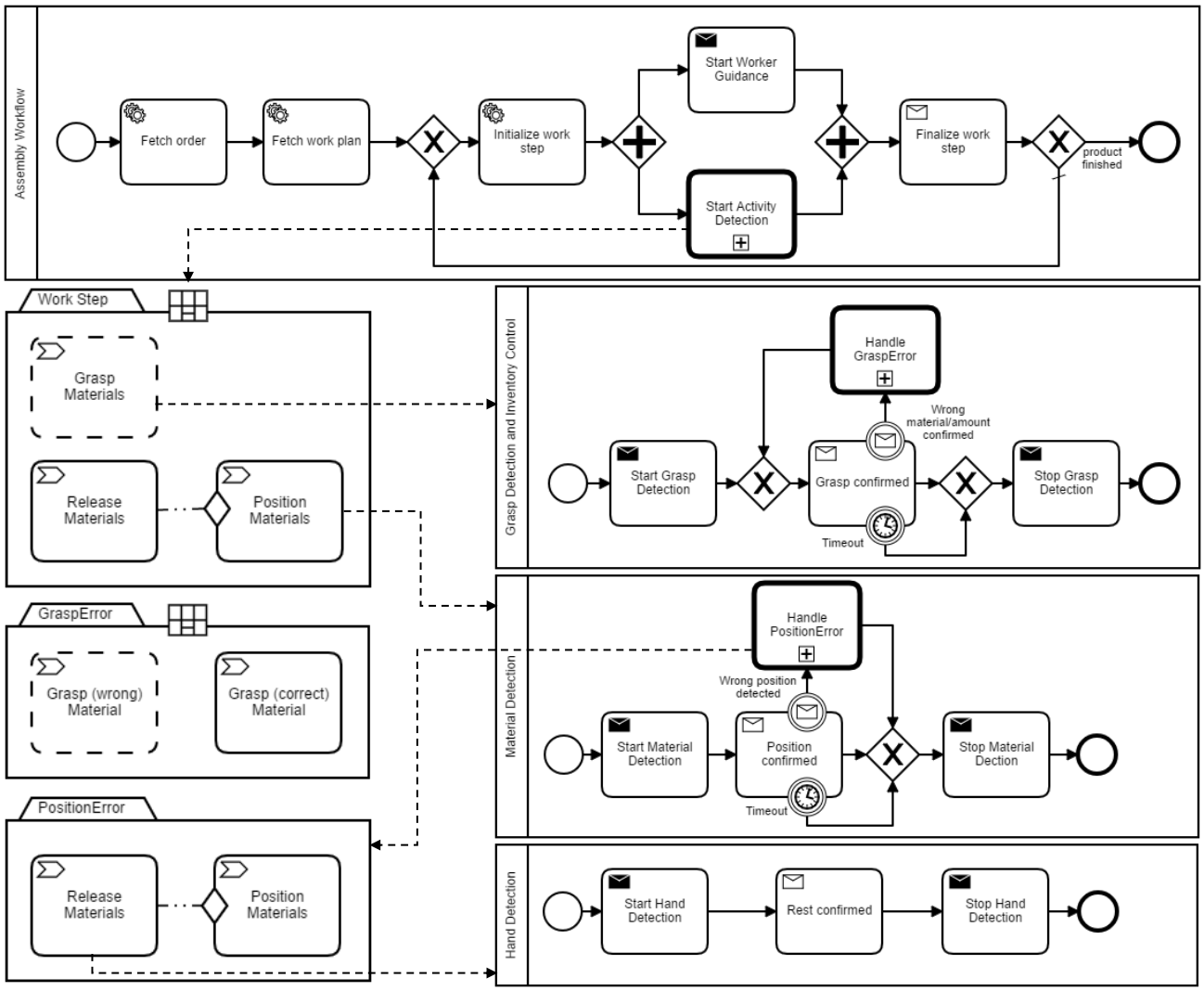

Fig. 4 Assembly workflow (top) and activity detection processes (bottom right) modeled in BPMN; validation of work steps and error handling modeled in CMMN (bottom left).

rameter H_F (note that this history is different from that defined by the parameter α), only when a certain user-defined percentage $p_H(60\%)$ of frames contains activity or inactivity, grasp start or end is indicated. The variable c_a counts the number of frames where activity is detected and c_i counts the number of frames where a lack of activity is detected after a grasp start event. The activity flag, keeps track of the state of the system.

```
while hasNext(V(t)) do
    if (|F(t)| > p_a * |P(a)|) then
        c_a++;
        c_i = 0;
        if (c_a > p_F * H_F|) then
        send(grasp_start), activity=true;
    end
    else if (activity) then
        c_i++;
        c_a = 0;
        if (c_i > p_F * H_F) then
        send(grasp_end), activity=false;
    end
end
```

Algorithm 1: Background subtraction algorithm to detect start and end of grasping material.

5.4 Inventory Control

Selecting the appropriate weighting sensor to gather accurate weight values requires the consideration of the target environment, whether it is dry, dusty, or wet. Further, range of weights to be covered (min and max value), and the weighing tolerance. Once the appropriate sensor is selected, the software needs to provide basic functionality to zero the scale, gather the tare value and the number of reference pieces to enable counting

of material based on the relative measurement. After configuration, the software fetches weight values from a set of scale pads in a fixed clock cycle.

$$\hat{\sigma}_t(n) = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}(w_{t-i} - \hat{\mu}_t(n))^2} \quad (1)$$

When a worker grasps into the box, the weight value rises caused by the pressure forced by the worker's hand collecting the material parts from the box, which induces noise. If no noise is generated by grasping, the new number of materials can be immediately adapted. Otherwise, to filter out the noise generated by this activity and to detect the step from one amount of pieces to another, statistics can be applied to the time series of weight values. Therefore, the rolling standard deviation $\hat{\sigma}$ is computed for windows with width $n = 2$ using the 2 most recent weight observations w_t and w_{t-1}. If the value of $\hat{\sigma}$ comparing these to observations exceeds a certain threshold motion is detected. The threshold is set to the weight of one reference piece. During motion, the old number of pieces is kept. When motion ends, the number of pieces is updated.

Compared to grasp detection the motion signal delivers similar information about when a worker enters a material box. Since the effect on the weight occurs only when pressure is measurable, grasp detection will lead to more accurate time information about when a worker actually enters and leaves the area defined around the box. In combination, both systems deliver detailed data about grasping material.

5.5 Hand Detection

After the user has assembled the grasped material, he or she actively confirms the work step by putting his hands on designated areas of the workstation close to the assembly area, which have electrically conductive metal plates integrated into the assembly worktop. Those plates are connected to a capacitive sensor. The sensor measures the capacitance of the capacitor, which is formed by the electrodes "metal plate" and "operator". By laying down both hands, the measured capacity reaches a characteristic value, which initializes the manual confirmation of the operator's work step and forms the point in time when the photo for material detection is taken. This kind of work step confirmation is a common procedure in manual assembly and helps in the our implementation to avoid occlusion and motion blur during material detection. As these metal plates are integrated directly next to the assembly area, there is no need to reach distant displays or buttons.

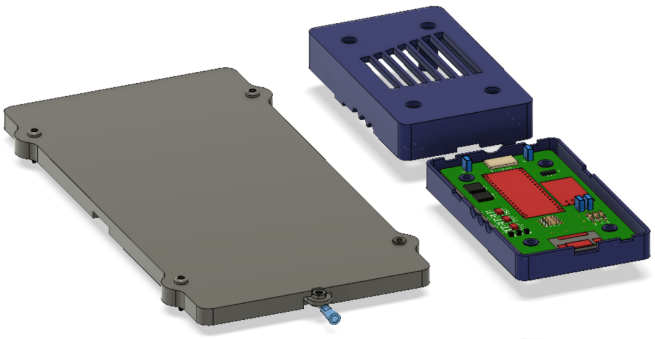

Fig. 5 3D model of the hand detection controller (right) wired to 2 capacitive sensor plates (left).

The prototype hand detection assembly shown in Figure 5 controls the confirmation process of the operator's currently active work step. The controller unit Printed Circuit Board (PCB) was designed in Autodesk EAGLE and packaged - as well as the sensor plates - into 3D modelled casings which were designed in Autodesk Fusion 360, and 3D-printed.

5.6 Process-based Gathering of Image Data

For the system described in this paper, it is essential to validate the assembly state as correct or identify an error as a part of quality control. We used the BPMN model to automatically label the assembly steps: a custom model which, in combination with the worker guidance system, not only instructed the participant to perform the usual assembly steps, but also directed her to generate erroneous states (i.e. wrong part placement or orientation). This made it possible to semi-automatically (the participant still had to place her hands next to the workpiece after each step) take a picture of every state of the BPMN-model and label it at the same time. Since different lighting conditions easily occur in a real life setting, we also took these into account during the training data acquisition: to simulate various lighting condition, six Philips Hue lights were placed in different positions around and on the assembly workstation. Using the Philips Hue API, we programatically changed the lighting by switching different bulbs on and off or changing their color and intensity, which results in different shadows and highlights on the pictures of the workpieces. Whenever the participant placed the hands next to the workpiece, we captured a total of 7 images using different light scenes. This automatic light adjustment dramatically speeds up the capturing of training data including a wide variety of lighting conditions, thereby improving the robustness towards light changes in the resulting system.

5.7 Image Classification

One approach to detect the current state is *Image Classification*, which learns a function taking as input an image captured as defined above and outputs the current state in the BPM. Vast improvements in image classification results using deep convolutional neural networks have been achieved in recent years; therefore, we decided to also use convolutional neural nets for our task. Since our task is comparatively simple, we do not apply complex and extremely deep convolutional networks like ResNet (He et al., 2016), but instead design our own very simple network: we use 5 (convolutional, convolutional, max pooling) blocks with filter size of 3 by 3, pool size of 2 by 2, and relu activation, followed by a dense layer of 512 (relu activation), and the final classification dense layer performing softmax. Dropout is used to reduce overfitting.

To minimize the amount of images required to achieve high classification accuracies, we perform an initial training procedure based on parts of the ImageNet dataset (Russakovsky et al., 2015). For this, we downloaded a total of 419 classes of the set, including 178 classes we considered roughly related to PCB assembly, such as 'electrical circuit', 'printed circuit', and 'circuit board'. We resize all images to 224 x 224 pixels and train the network for 300 epochs using a batch size of 64, RMSProp optimization, and a learning rate of 0.0001. These weights are stored and used as a basis for the training process on the images specific to our assembly task. This pretraining should reduce the amount of images required, as basic features like edges, shapes, and colors can already be learned from the ImageNet data. The dense and classification layers at the end are randomly initialized when fine-training on our dataset, as these can be considered specific to the ImageNet data.

5.8 Object Detection

One of the oldest and highly successful object detectors was proposed in 2001 by Viola and Jones (2001). Though it was demonstrated initially for faces, the framework was later used to propose accurate classifiers for other classes of objects like cars and pedestrian detection (Lee and Kanade, 2007; Monteiro et al., 2006). Even though the field of object detection/classification has progressed vigorously to state-of-the-art performance by Neural Nets, we decided to start with the approach from Viola and Jones since we considered a small scale task with a controlled scenario. The approach also had some additional benefits: (1) the interpretability of the features learnt; (2) the existence of highly optimized training and testing codes (OpenCV).

Training and detection times are important characteristics of any learning based application. Here, we train an object detector for each part and hence the training time is directly proportional to the number of parts. In our scenario, due to the small number of parts, fast training times are irrelevant, however in theory the number of parts could be much higher. Hence another application could benefit from fast training times. Fast detection time is however critical irrespective of the quantity of parts and therefore also relevant for our use case. The final application is expected to fire a result with minimal delay after the part has been assembled. So a fast detector helps in minimizing this time delay.

Given the above described requirement, even though the original Viola-Jones detector used Haar wavelet like features, we decided to explore the Local Binary Pattern (LBP) based variation proposed by Ahonen et al. (2006). This LBP-based approach, though less accurate (still the overall accuracy is above 90%), provides fast training and test times.

6 Evaluation

We conduct an evaluation, testing the individual parts of the activity detection module and analyzing to what extent these can be used to automatically gather insights into the assembly process to optimize it in later steps.

6.1 Grasp and Hand Detection

The grasp and hand detection is evaluated using a series of experimental runs involving 6 users (1 *f*, 5 *m*) with different hand sizes (circumference: $\bar{x} = 21.58$, $\sigma = 1.64$ cm; length: $\bar{x} = 18.83$, $\sigma = 1.77$ cm; span: $\bar{x} = 21.67$, $\sigma = 2.29$ cm). During the experiment, the user grasps a part, places it on the worktable and confirms by resting the hands. The user then returns the part and confirms by resting the hands. This procedure is repeated for each part and box with right and left hand alternating 2 times per part in four repetitions comprising all parts. This leads to a 2 (hands) $*$ 2 (remove and return part) $*$ 4 (parts) $*$ 4 (repetitions) $=$ 64 grasp start, stop and rest events. The activity start and stop events are monitored by a supervisor with an annotating application to generate ground truth information. The rest detection failed only in 2 cases where two users curved their palms, leading to no sensor response. In total, an accuracy of 99.22% was achieved. For grasp detection, 2 pairs of start and stop events failed where one user approached from angles where the activity zone is least disturbed, such that the threshold of 40% foreground

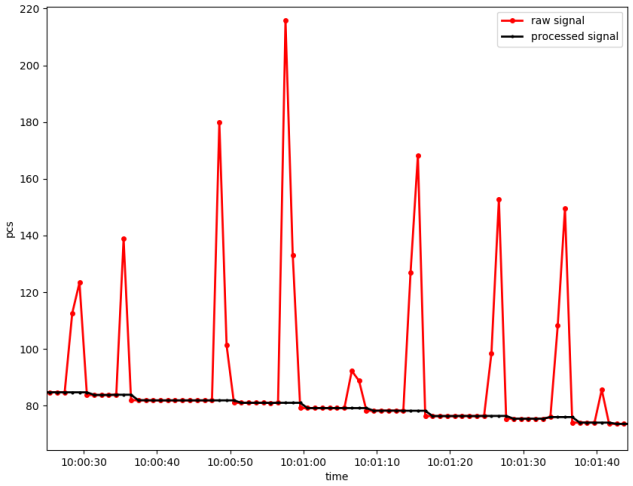

Fig. 6 Excerpt of the validation of weight values using a sliding standard deviation with n=2 counting steel nuts.

pixels was not reached in the activity zone. In total, an accuracy of 99.48% was achieved for grasp detection.

6.2 Inventory Control

The scales in the setup were designed to support a relative measurement necessary for inventory control. Therefore, a simple validation of the counting of bulk material was conducted with two parts: 100 steel nuts, class 4, zinc-plated, size M4x0.7 mm, and 46 black-oxide screws, size M3x0.5 mm, 25 mm long. At the beginning of each run, all parts were filled into the box and used as reference parts. Afterwards, we started removing 10 parts one-by-one, 5 times first 1 then 2 parts, and 5 times first 2 then 4 parts, in total grasping 55 pieces per material with ca. 5 second between each grasp. The screws were refilled before running empty after the second time grasping the 2-4-pair.

Figure 6 shows an excerpt of the nut grasping experiment. The red line indicates the non-validated number of pieces and the black line the validated number of pieces. It can be seen that after stabilization, when two almost equal measurement points exist, the number of pieces is reduced to the new validated value. Since no surprising behavior was observed, we stopped our study at this point. Due to the time it takes for the scale to stabilize after a hand motion inside a box, the timings recognized using grasp detection are more accurate than using the timings of the scale directly.

6.3 Process-based Gathering of Image Data

During data gathering of images for state classification, the remaining lighting conditions were kept stable (shutter closed, room and workstation lights on). We used seven different simulated lighting conditions (including all lights off, i.e. room lighting) per assembly state. The rationale behind this was to allow for different lighting preferences of the individual workers. Each iteration consisted of 19 states, generating $19 * 7$ images and with a duration of approximately 8 minutes (25 seconds per assembly step). Two types of iterations were considered, orienting all parts in two directions (left and right) leading to 38 classes in total. These classes cover all possible assembly states due to the assembly area where the parts are inserted oriented to the left or right.

With this setting, we were able to generate 3990 images in 30 runs within 4 hours. To avoid irrelevant parts of the surroundings, such as hands or tools being present in the image, the camera was fixed and the assembly area was cropped from each image. Overall the goal of state classification (the following two sections) is to quantify how well the approach works to determine whether material was assembled correctly. For this, we use 50% of the labeled image data for training, 25% for validation, and 25% for testing.

6.4 Image Classification

As described in Section 5.7, the neural network used to classify the various assembly states was pre-trained on a subset of ImageNet, and then all except for the dense layers are fine-tuned on the training data specific to our task. For this, we again resize all input images to 224 by 224 pixels and train for 25 epochs using a batch size of 8. As expected due to the simple nature of the problem (in comparison to competitions like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)), the classification performance is very good: a test accuracy of 99.25% was achieved on this 38-class problem.

6.5 Object Detection

An object detector is trained for each relevant part in its 19 states in each orientation. For the detector for each state, the images belonging to all the other states were considered as negatives. The images from one orientation are rotated to match with the other orientation. Thereby, the same classifier can be used on the flipped input image to detect the two different orientations. If the part was detected in the image and the location of the part fits the target location the result indicates a correct assembled material part. The result of the object detection combined with the information from the

process about the correctness of the last state results in the correctness of the current assembly state.

To prepare the training data, the material parts were cropped from the images automatically possible due to the fixed position of the camera and assembly area, and labels from the training process. Informal tests resulted in difficulties detecting ApplicationBoard and ConnectingBoard. Therefore, data augmentation has been integrated. In a first step, the part images of ApplicationBoard and ConnectingBoard were rotated about the three axis (X and Y along the width and height, and Z into the plane of the image), scaling the images and changing the brightness. These are standard ways for data augmentation. The distorted images were then placed in varying background images, as can be seen in Figure 7. The result was used to train the cascade classifier based on the unifying features found in these images. As negative images, background images and object images (in total 4000) downloaded from various databases based on Fei-Fei et al. (2007) were used for training object detectors. For each part about 1000 cropped images of the other three parts were added as negatives.

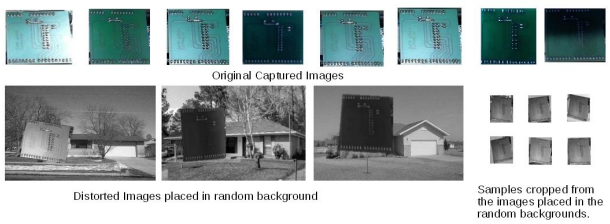

Fig. 7 Steps in ConnectingBoard training images generation.

The detection accuracy for the states within the first two work steps (involving the TopCasing and Mainboard assembly) were above 95%. The detection for the states in the next work step involving ConnectingBoard and ApplicationBoard were challenging, since these parts have a lot of reflecting feature points in them and were often confused with one another. To tackle this challenge, we focused on these two parts. After careful augmentation with respect to part orientation, degree of rotation, and range of scaling, a test accuracy of 95% was achieved for the states within this work step.

7 Discussion

7.1 Material Detection

The results indicate the competitiveness of both approaches object detection and image classification. Even this very simple convolutional neural network achieved high accuracies, however, for tasks with even more classes than our test case one might need to increase the depth of the network. The object detection approach has the added advantage of providing localization, however, in an application of the type discussed localization is not necessary. Nevertheless, in tasks where the assembly area is not fixed, the detection approach could have advantages compared to the classification approach. Using a different Neural Network architecture like MaskR-CNN (He et al., 2017) would also allow localization, and one could use the same data as for the current object detection approach, or use the same simple gathering procedure and data augmentation techniques to acquire more data. Also the class of functions offered by Neural Network is much bigger than Cascade Detectors/Classifiers and hence more scalable for more states and parts. This is a very important aspect since we aim for a solution for a dynamic factory scenario. With the detector approach though high accuracy results can be achieved, a lot of effort is involved in training detectors. When scenarios occur where different states are quite similar, additional effort is required as in the case of the parts ApplicationBoard and ConnectingBoard in the use case at hand. Here, our image classification approach did not have such strong problems and was a lot easier to train.

Comparing object detection with image classification one major drawback of object detection is the strong dependence on previous detections. While an image classifier checks the whole assembly state in each run, the object detector validates the position of objects and assumes the previous states as stable. Errors may be propagated through the process and have to be corrected manually.

7.2 Other Parts of the Pipeline

Furthermore, we see that many steps of the overall detection process can be easily realized using sensors like capacitive sensors, high-precision scales, or simple background subtraction upon camera images. While we see how machine learning and specifically deep learning can yield very good results on more complex problems like material detection, there simply is no need to use such approaches for other parts of the pipeline. The process-based combination of sensors, as suggested for grasp detection in combination with inventory control, lead

to very precise results and provides interesting details about the assembly process.

7.3 Link between Process and Data

Using a very simple data gathering procedure, which could also be conducted during day-to-day work, we were able to train a system that can be used to get accurate timings for the individual steps in the process model, and to also automatically detect which state the worker is currently in, including error states. Thus, it can be used to (a) perform process optimization based on the timings, (b) analyze commonly occurring errors and figure out options to avoid these errors or at least provide aid to the worker on how to resolve the error.

The correlation between processes and activity detection based on capability descriptions and the modelling of specific process snippets allows process modelers to reuse the models and quickly combine them to new processes. Flexible process validation, for example to support experienced workers, can be achieved by combining CMMN and BPMN models. This allows a dynamic variation of the process strictness adapting to the worker's skill (Ullrich et al., 2016) in the particular task for an efficient human-process interaction.

7.4 Limitations

A limitation of our analysis is that we only explored the approach for the production of a single item. While we believe that the chosen test case is reasonably complex to realistically evaluate the advantages and disadvantages of our approach, we will explore more test cases in the future. Furthermore, while we tested our approach on a separate test set, we did not explore it in the wild with real working conditions and potential problems arising from this.

Since flexible processes are rather new, the CMMN support is not fully covered in process engines available on the market by now. In addition, the power of a business process engine controlling manufacturing workflows is limited when it comes to real-time critical processes.

8 Conclusion

Methods from AI, such as computer vision and machine learning, can be tailored to an Industry 4.0 use case and may increase an organization's competitiveness through awareness of error rates and timepass enabling backtracking and intermediate intervention in manual assembly workflows. We have shown that an easy-to-set-up tool set of two cameras, one capacitive sensor, four scales, six light sources and activity detection software trained within four hours of assembly has the ability to generate accurate sensor events. Viewing this data through a "process lens," the measurement of time and quality in manual assembly workflows, and thus the optimization of processes, is enabled.

In the future, we will investigate the potential of unsupervised methods in combination with our approach to discover workflows automatically. This leads to a novel concept of 'sensor-to-model', which deduces the overall process based on data gathered by the sensors. New methods to discover assembly workflows in planning and construction phases, and to monitor or check their conformance online and offline, will provide valuable input for process mining.

Acknowledgements This research was funded in part by the German Federal Ministry of Education and Research under grant number 01IS16022E (project BaSys4.0). The responsibility for this publication lies with the authors. The authors thank Mettler Toledo for providing the hardware setup used for inventory control in this research.

References

Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence 28(12):2037–2041

Cameranesi M, Diamantini C, Potena D (2018) Discovering process models of activities of daily living from sensors. In: Teniente E, Weidlich M (eds) Business Process Management Workshops, Springer International Publishing, Cham, pp 285–297

Carolis BD, Ferilli S, Redavid D (2015) Incremental learning of daily routines as workflows in a smart home environment. ACM Trans Interact Intell Syst 4(4):20:1–20:23, DOI 10.1145/2675063, URL `http://doi.acm.org/10.1145/2675063`

Cavanillas JM, Curry E, Wahlster W (eds) (2016) New Horizons for a Data-Driven Economy: a Roadmap for Usage and Exploitation of Big Data in Europe. Springer, URL `http://dblp.uni-trier.de/db/books/collections/CCW2016.html`

Fei-Fei L, Fergus R, Perona P (2007) Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer vision and Image understanding 106(1):59–70

Grzeszick R, Lenk JM, Rueda FM, Fink GA, Feldhorst S, ten Hompel M (2017) Deep neural network based

human activity recognition for the order picking process. In: Proceedings of the 4th International Workshop on Sensor-based Activity Recognition and Interaction, ACM, New York, NY, USA, iWOAR '17, pp 14:1–14:6, DOI 10.1145/3134230.3134231, URL http://doi.acm.org/10.1145/3134230.3134231

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

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969

Hull R, Motahari Nezhad HR (2016) Rethinking bpm in a cognitive world: Transforming how we learn and perform business processes. In: La Rosa M, Loos P, Pastor O (eds) Business Process Management, Springer International Publishing, Cham, pp 3–19

Janiesch C, Koschmider A, Mecella M, Weber B, Burattin A, Di Ciccio C, Gal A, Kannengiesser U, Mannhardt F, Mendling J, Oberweis A, Reichert M, Rinderle-Ma S, Song W, Su J, Torres V, Weidlich M, Weske M, Zhang L (2017) The internet-of-things meets business process management: Mutual benefits and challenges. Computing Research Repository (709.03628):1–9, URL https://arxiv.org/pdf/1709.03628

Jaroucheh Z, Liu X, Smith S (2011) Recognize contextual situation in pervasive environments using process mining techniques. Journal of Ambient Intelligence and Humanized Computing 2(1):53–69, DOI 10.1007/s12652-010-0038-7, URL https://doi.org/10.1007/s12652-010-0038-7

Kagermann H, Helbig J, Hellinger A, Wahlster W (2013) Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. Forschungsunion

Kerber F, Lessel P (2015) Adaptive und gamifizierte werkerassistenz in der (semi-)manuellen industrie 4.0-montage. In: DeLFI WOrkshops

Knoch S, Kerber F, Pavlov V, Ponpathirkoottam S (2016) Automatic capturing and analysis of manual manufacturing processes with minimal setup effort. In: International Joint Conference on Pervasive and Ubiquitous Computing, ACM, UbiComp, pp 305–308

Knoch S, Ponpathirkoottam S, Fettke P, Loos P (2018) Technology-enhanced process elicitation of worker activities in manufacturing. In: Teniente E, Weidlich M (eds) Business Process Management Workshops, Springer International Publishing, Cham, pp 273–284

Knoch S, Herbig N, Ponpathirkoottam S, Kosmalla F, Staudt P, Fettke P, Loos P (2019) Enhancing process data in manual assembly workflows. In: Daniel F, Sheng QZ, Motahari H (eds) Business Process Management Workshops, Springer, Lecture Notes in Business Information Processing (LNBIP), vol 342, pp 269–280

Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M (2014) Industrie 4.0. WIRTSCHAFTSINFORMATIK 56(4):261–264, DOI 10.1007/s11576-014-0424-4, URL https://doi.org/10.1007/s11576-014-0424-4

Lee DC, Kanade T (2007) Boosted classifier for car detection. unpublished, http://www cs cmu edu/~ dclee

Lenz C, Sotzek A, Roeder T, Radrich H, Knoll A, Huber M, Glasauer S (2011) Human workflow analysis using 3d occupancy grid hand tracking in a human-robot collaboration scenario. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3375–3380, DOI 10.1109/IROS.2011.6094570, URL http://ieeexplore.ieee.org/document/6094570/

Marrella A, Mecella M (2017) Cognitive business process management for adaptive cyber-physical processes. In: Teniente E, Weidlich M (eds) Business Process Management Workshops, Springer, Lecture Notes in Business Information Processing, vol 308, pp 429–439, URL http://dblp.uni-trier.de/db/conf/bpm/bpmw2017.html#MarrellaM17

Monteiro G, Peixoto P, Nunes U (2006) Vision-based pedestrian detection using haar-like features. Robotica 24:46–50

Poppe R (2010) A survey on vision-based human action recognition. Image and Vision Computing 28(6):976 – 990, DOI https://doi.org/10.1016/j.imavis.2009.11.014, URL http://www.sciencedirect.com/science/article/pii/S0262885609002704

Roitberg A, Somani N, Perzylo A, Rickert M, Knoll A (2015) Multimodal human activity recognition for industrial manufacturing processes in robotic workcells. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ACM, New York, NY, USA, ICMI '15, pp 259–266, DOI 10.1145/2818346.2820738, URL http://doi.acm.org/10.1145/2818346.2820738

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3):211–252, DOI 10.1007/s11263-015-0816-y

Sora D, Leotta F, Mecella M (2018) An habit is a process: A bpm-based approach for smart spaces. In: Teniente E, Weidlich M (eds) Business Process Management Workshops, Springer International Publishing, Cham, pp 298–309

Stiefmeier T, Roggen D, Ogris G, Lukowicz P, Troester G (2008) Wearable activity tracking in car manufacturing. IEEE Pervasive Computing 7(2):42–50, DOI 10.1109/MPRV.2008.40, URL http://ieeexplore.ieee.org/document/4487087/

Thoben KD, Poeppelbuss J, Wellsandt S, Teucke M, Werthmann D (2014) Considerations on a lifecycle model for cyber-physical system platforms. In: Grabot B, Vallespir B, Gomes S, Bouras A, Kiritsis D (eds) Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 85–92

Ullrich C, Aust M, Dietrich M, Herbig N, Igel C, Kreggenfeld N, Prinz C, Raber F, Schwantzer S, Sulzmann F (2016) Appsist statusbericht: Realisierung einer plattform für assistenz-und wissensdienste für die industrie 4.0. In: DeLFI Workshops, pp 174–180

Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, IEEE, vol 1, pp I–I

Wombacher A (2011) How physical objects and business workflows can be correlated. In: 2011 IEEE International Conference on Services Computing, pp 226–233, DOI 10.1109/SCC.2011.24

Zivkovic Z, Van Der Heijden F (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters 27(7):773–780