Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Deep Learning-based Object Recognition for Counting Car Components to Support Handling and Packing Processes in Automotive Supply Chains

A. Börold*. M. Teucke.**
A. Rust*. M. Freitag******

**BIBA – Bremer Institut für Produktion und Logistik at the University of Bremen, Bremen, D-28359, Germany (e-mail: bor@biba.uni-bremen.de)*
***BIBA – Bremer Institut für Produktion und Logistik at the University of Bremen, Bremen, D-28359, Germany (Tel: 49-421-218-50159; e-mail: tck@biba.uni-bremen.de)*
****Faculty of Production Engineering, University of Bremen, Bremen, D-28359, Germany (e-mail: rus@biba.uni-bremen.de)*
*****BIBA – Bremer Institut für Produktion und Logistik at the University of Bremen, Bremen, D-28359, Germany Faculty of Production Engineering, University of Bremen, Bremen, D-28359, Germany (e-mail: fre@biba.uni-bremen.de)}*

Abstract: Complex distributed supply chains, e.g., in the automotive industry, need to cope with high product variety. Digital image processing can use specific geometric and optical properties of parts and components for determining their type and thus needs no external markers. It is thus well applicable to supply chain processes that involve direct handling of many different product components and need no individual identification of items. An example of such a process is counting items of different product types during packing. In this paper, we use deep learning-based digital image processing methods in order to distinguish and count the number of objects of two different types of automotive components in standardized transport bins, detected by time-of-flight (ToF) depth sensors. Classical watershed object counting methods are adapted to depth data and support the fast generation of training data for the deep learning-based classification methods. The proposed method is applied to an automotive supply chain, and it is demonstrated that car components can be counted with good reliability during packing into transport bins. Thus, digital image processing can be useful to supplement auto-identification and sensor technologies and complete digital end-to-end monitoring of supply chains.

***Keywords**:* Supply Chain Monitoring, Supply Chain Transparency, Digital Image Processing, Deep Learning

1. INTRODUCTION

Operations in global production networks are influenced by disruptive events, like product mismatches, quality deviations and delays. These may result in the requirement of post-treatment, additional emergency transports, downtimes and production loss as well as changes to the final product (Ivanov et al., 2016). Enhanced digitization and information exchange can increase transparency in production networks, which leads to faster identification and elimination of disruptions (Lanza and Treber, 2019). Auto-identification technologies (Finkenzeller, 2006) can identify individual objects, which allows real-time tracking of products (Musa et al., 2014). However, these technologies require external codes or tags. This is disadvantageous in several ways: they may adversely modify the quality and properties of the objects; the objects may be too small for their attachment, or be subject to conditions that damage them. For these reasons, external codes and tags may not be applicable in many processes that involve direct handling of individual objects.

With the use of deep learning, image recognition can use inherent, characteristic, optically distinguishable properties, or features, of the objects for classification of objects (decision, whether an individual object belongs to a specific type of objects) without external tags. For this reason, deep learning based image recognition is well suited to monitor supply chain processes that involve direct handling of individual parts and require distinguishing only the type or variant of a part, but not to identify a unique individual part. Examples of such operations are sorting, picking, or packing.

In this context, this paper makes the following contribution: Based on a case study from the automotive industry, we investigate the suitability of deep learning based image processing for automated classification (determining the type) and counting (determining the number) of different product components in transport bins, and in addition, the checking of their correct alignments. This requires the simultaneous detection of numerous, unsorted, tightly arranged objects at arbitrary positions. Our approach uses 3D geometric data generated by active time-of-flight (ToF) depth sensors and combines deep learning with a conventional watershed algorithm based segmentation method for the fast generation of training data for the CNN (Fig. 2). Depth sensors are particularly suited to distinguish between parts, which differ only in size, but not in color or geometric form. Using only color (intensity) data instead would make separation and distinction between many densely arranged objects of the same

color very difficult. In comparison to previous methods, CNN shows a markedly improved reliability and thus promises better performance.

Our principal aim is a test employing a case study, whether our method works. Consequently, we apply the method to a use case involving the automated detection and counting of automotive components in standardized transport bins. The motivation is to automate the inspection in operations like packing, picking, warehouse entry, and warehouse exit, which are part of component sourcing processes and involve the handling of numerous identical or similar objects. It may serve to improve the order conformity of component sourcing processes in production networks in the automotive industry and decrease the related costs.

2. RELATED WORK

Existing image recognition methods for the detection of numerous tightly packed objects either use conventional segmentation, but not machine learning (Rahman and Islam, 2013), or do not classify the objects, but use object density maps to estimate the number of objects (Onoro-Rubio and López-Sastre, 2016).

Compared to classic image recognition algorithms, convolutional neural networks (CNN) provide enhanced capabilities for extracting deep information from image data (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014, LeCun et al., 2015). Applications include simple object detection (Ren et al., 2015), as well as more comprehensive tasks, like process monitoring (Staar et al., 2018) and real-time environmental analysis (Börold and Freitag, 2019). An exemplary application of CNN based image recognition in logistic processes is the automated classification of turbine spare parts in assembly and disassembly processes (Krüger et al., 2019). This application realizes the automated inspection of a single salient component and the determination of the type it belongs to, but not the detection of multiple densely packed objects in one image, as in our case.

DESCRIPTION OF THE CASE STUDY

A use case from the automotive industry is employed for testing the method. Sourcing processes in automotive production networks involve high product variety (Klug, 2018). A single car includes several thousand different parts, sourced from many external suppliers. The transport of many smaller parts and components is done in standardized transport bins of fixed size. Predefined packing schemes regulate the location of parts inside the bins.

Counting of homogeneous objects is a trivial task that requires no identification and can be done, e.g., by weighing or using light barriers. Counting becomes complicated if several different types of objects are present so that establishing the number of objects has to be combined with verification of the correct type. It becomes even more complicated if the correct position and orientation of parts need verification as well. By applying double verification, an automated assistance system can improve the manual verification of a human operator, so that both assist each other in detecting packaging defects. In this case, a low error rate of the assistance system may be tolerable if it reduces the probability of errors compared to the non-assisted employee and increases the packing productivity. Today's supply chain processes with a large number of process variants for individual customers of a supplier and frequent modifications of the parts and products require high flexibility and quick adaptability of the assistance system. In particular, the system should be able to recognize new types and variants of parts as well as different positions of these parts after limited training efforts.

Our case study consists of the classification and the counting of coolant pumps, which are sourced by the car manufacturer from a component supplier. At the supplier, the ready-made pumps are packed into standardized transport bins for the transport to a logistic hub. After inserting the pumps, the filled bins are manually inspected, to verify that the number and alignment of the pumps conform to the orders. At the hub, an external logistic services provider (LSP) collects incoming components from different suppliers and consolidates them as well as other components into containers for further transport to the OEM by truck, container vessel, and train. The LSP does not handle individual pumps, but only the filled bins as a whole. At the manufacturer's goods receipt, incoming bins are rechecked on a sample basis. Finally, after intermediate storage, the bins are provided at the corresponding stations at the assembly line, where the pumps are manually taken out and assembled into cars. The problem we address is verifying at the supplier as well as at the OEM that the required numbers of pumps are packed into the transport bins (Fig. 1).

.

Fig. 1. Standardized automotive transport bin with pumps

The pumps are available in two different sizes, with mostly identical shapes and colors. A bin may contain between 20 and 40 tightly packed pumps of either type in just one layer. Thus, some pumps may cover parts of others, but the essential structural characteristics of the pumps remain visible. Each pump has to be checked for correct alignment in the bin, too.

4. IMAGE RECOGNITION APPROACH

4.1 Experimental setup

The set-up and algorithm flow are illustrated in Fig. 2. For data acquisition, we use the consumer-grade ToF and intensity sensor Microsoft Kinect v2. The sensor has to be located at a minimum distance of 0.70 m above the bin, so that the bin falls completely within the visible range. The components in the bin must not be layered so that all components can be detected.

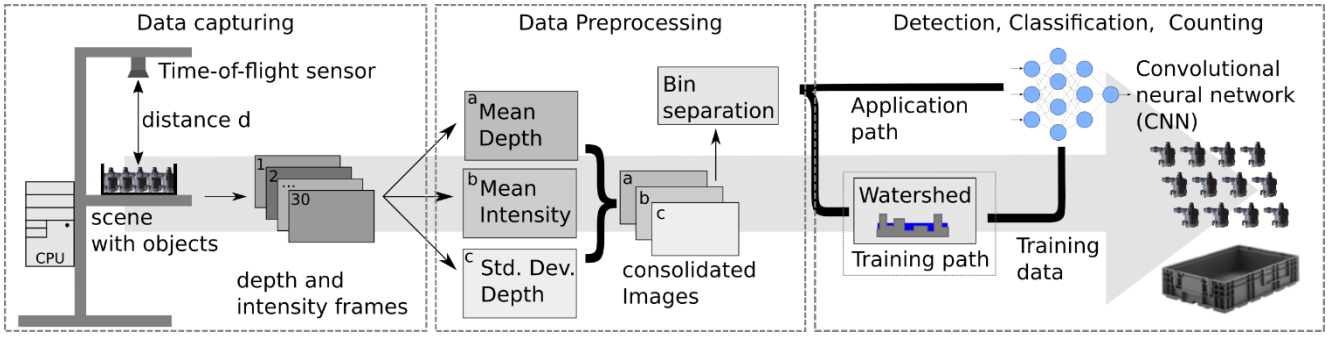

Fig. 2. Experimental set-up and algorithm workflow

The components are placed close together in the bin. There are occlusions between objects, but the essential visual characteristics of the pumps remain visible. Since the components also have to be checked for correct alignment, it is necessary to determine the exact position of each pump.

Altogether, 130 sequences, each of 30 frames, are generated. It is vital that the training data contains as much data variation as possible. From a set of 30 frames of the ToF sensor data, a consolidated depth image is created by calculation of mean intensity and depth and standard deviation of depth. After segmentation, the depth image data is fed into a deep learning net (CNN), where objects are detected and classified and subsequently counted. Training data is generated using a watershed algorithm combined with several thresholds. The mentioned aspects are described in the following sections.

4.2 Pre-processing of sensor data

Due to its discrete nature, the photon shot noise of the ToF sensors corresponds to a Poisson distribution (Büttgen et al., 2005). Since the number of interacting photons per pixel is vast and the sensor puts out an internal mean depth value computed from several measurements rather than discrete measurements, the said noise distribution can also be approximated by a Gaussian distribution.

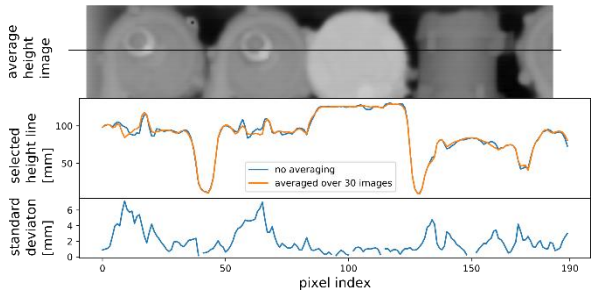

Fig. 3. Depth image, depth and standard deviation profiles for indicated exemplary horizontal pixel line

Due to the static structure and the fixed number of frames, no online functions are needed, and all calculations are performed after data capturing. In comparison to the data from a single depth frame, we can achieve a higher data quality by temporal filtering and calculate additional information for the object recognition algorithm. First, by calculating the mean depth image over the entire sequence, we significantly improve surface estimation quality and object recognition. Second, we calculate the standard deviation for each pixel as a standard deviation image.

Since the number of photons emitted depends on the surface inclination and material properties, it is related to the noise level (Fig. 3). Features such as edges and material changes are, therefore, detectable in the standard deviation image. Thus, the calculations mentioned above provide additional value for detection. The data is calculated as follows:

Since the ToF sensor returns a zero value if the depth measurement was not successful, a map $\mathcal{M}$ containing the number of valid depth measurements for each pixel in the sequence of all depth images D_1 to D_{30} must be created first. The parameter p_{xy}^{M} denotes the corresponding count allocated to the pixel at position (x, y) according to equation (1):

$$p_{xy}^{\mathcal{M}} = \sum_{k=1}^{n=30} f\left(p_{xy}^{D_k}\right) \text{ with } f\left(p_{xy}^{D_k}\right) = \begin{Bmatrix} 1, if\ \ p_{xy}^{D_k} > 0 \\ 0, else \end{Bmatrix} \quad (1)$$

The values of map $\mathcal{M}$ are then used to calculate the averaged depth image $\mathcal{R}$. The arithmetic mean depths of all valid pixels are calculated according to equation (2). The parameter p_{xy}^{R} denotes the mean depth of the pixel at the position (x, y), while $p_{xy}^{D_k}$ denotes the depth of the corresponding pixel of the k-th individual frame.

$$p_{xy}^{R} = \frac{\sum_{k=1}^{n=30} p_{xy}^{D_k}}{p_{xy}^{M}} \quad (2)$$

In addition to the mean value, the standard deviation d_{xy}^{D} of each depth point is calculated for the map S by using equation (3). Again, only valid measurements are considered.

$$p_{xy}^{S} = \sqrt{\frac{\sum_{k=1}^{n=30}\left(p_{xy}^{D_k} - p_{xy}^{R}\right)^2}{n-1}} \quad (3)$$

Analogously the intensity data is calculated. No spatial filtering is used because the general idea of deep learning is to use raw data wherever possible so that the network can adapt to this data.

The experimental set-up ensures that all components placed in the observed bin are located entirely within the visible range of the sensor. The efficiency and robustness of the counting method are increased by considering only the relevant area within the bin. All depth points of the upper edge of the bin are located at approximately identical distances from the sensor plane. The binary mask *K* is created according to equation (4), by considering only those pixels from the consolidated depth image R, whose depth values belong to the defined depth interval $[s_1, s_2]$, with s_1 as lower and s_2 as upper limit. These include all measuring points *K* of the edge as well as additional points within the bin.

$$K = [s_1, s_2] \cap \mathcal{R} \quad (4)$$

A floodfill algorithm segments the bin using *K*. The remaining pixels contain the area within the bin. The bounding box, i.e., the smallest possible rectangle that encloses all remaining pixels, now borders exclusively on the area where the objects to be counted are expected. In the further course, only this cropped data area is used. The following pre-processed data is used for later training and recognition: The depth data from the averaged depth image R with corresponding standard deviations S and the averaged infrared image. All three data channels are normalized and lossless stored.

4.3 Deep learning based detection of unsorted objects

To test the suitability of CNN to detecting densely arranged objects, we rely on CNN detector and classification networks with a proven record in object detection. As a first CNN architecture, we combine Faster Regional Convolutional Neural Networks (R-CNN) (Ren et al., 2015) with ResNet101 (He et al., 2016) and alternatively with Inception v2 (Szegedy et al., 2016) base networks. Generally, a CNN detects different objects in an image in two steps: First, areas are marked, where objects can be found. Subsequently, these region proposals are classified. Faster R-CNN networks contain a specialized proposal network that uses a previously created feature map to mark areas with objects. Subsequently, each region proposal is passed forward into the classifier network. The Faster R-CNN networks offer excellent results on the COCO data set [19].

As a second CNN architecture, we combine a Single Shot MultiBox Detector (SSD) network (Liu et al., 2016) with Inception v2 and alternatively with MobileNet (Howard et al., 2017) as base networks. SSD networks require only one internal forward pass through the CNN with short computation times. Detections are here determined by testing a fixed set of bounding boxes of different locations, sizes and aspect ratios. ResNet101 and Inception v2 are classification networks that have achieved good results in combination with both Faster R-CNN and SSD. MobileNet is a classification network designed for low memory requirements, which we include for test purposes, as it is suited for mobile devices.

4.4 Generation of training data

CNNs require thorough training to achieve reliable object detection. Our training data consists of pre-processed sensor data from the actual process, which are annotated with labels. Each label indicates the position and class of an object. To generate labeled data, we have implemented a watershed algorithm and several threshold procedures. This procedure is similar to (Raman and Islam, 2013), but it takes the potential of the depth data into account. The resulting labels may contain errors, but these are easy to correct manually afterward.

Of the available data set of almost 4,000 image instances, 80% have been used for training and 20% for the subsequent evaluation of the CNN. As already stated in chapter 4.1, the training data must contain as much data variation as possible. Therefore, in each sequence of 30 frames, the position and orientation of the bin changes, the number and orientation of the components vary, and even some entirely different, random objects are included. In addition, pumps are included that lie upside down, sideways, or above another pump in the bin. For each new sequence, all objects have been completely rearranged.

The algorithm works as follows: First, the average depth image $\mathcal{R}$ is segmented by a threshold value s into the image S. It colors areas within a component white and gaps between the components black. A distance transformation then calculates the shortest Euclidean distance to a black pixel from image S as image D for each white pixel. By filtering with the threshold value d, areas are created in image E, which represent separate object proposals in averaged depth image $\mathcal{R}$. These object proposals have larger radii than d. The larger the smallest surface diameter, the larger the diameter of the proposed object. All object suggestions in the original image $\mathcal{R}$ are used as a basis for the watershed method (Fig. 4).

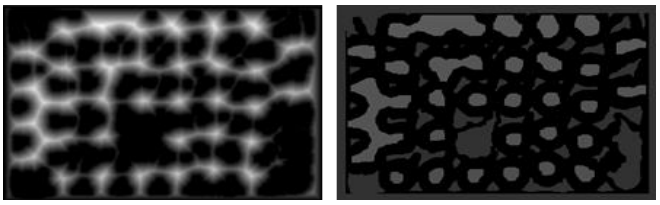

Fig. 4. Left: Distance image D, Right: Object Proposals E

The boundaries of the objects grow stepwise until they meet other objects. The black areas of map S, i.e., the gaps between the components, are set as markers. They prevent the gaps from being added to the segment of a component. The result is a fully segmented image (Fig. 5). The algorithm has to be run separately for each object type (i.e., twice). First, the large pumps are identified and then removed from the depth image. Second, this calculation is repeated for the small pumps. Objects of the precisely same size cannot be distinguished since the components are identified by their rough geometry. The algorithm segments overlapping pumps correctly in most cases. Also, there are problems with the distance transformation in very dense arrangements where several objects merge (in the example left and right side). Within the scope of the capabilities, the algorithm is used to create annotations for components very quickly.

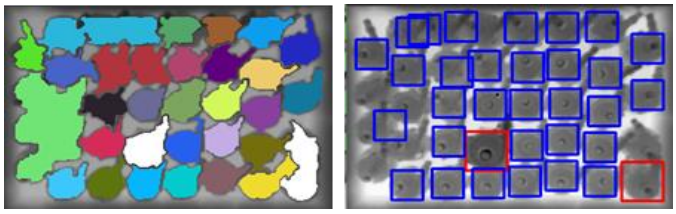

Fig.5. Segmented depth image, Labels (red: big, blue: small)

Usually, labels contain the smallest possible rectangle around all pixels belonging to a segment. As in our case, this includes protruding parts such as the pump connections. Labels would always contain parts of the background. In object detection, learning through a large number of training images and different backgrounds allows the algorithms to learn the object independently of the background or surrounding objects. This principle cannot be used here since the labels always cover parts of other pumps as well as the same background. Tests have shown (Onoro-Rubio et al., 2016) that a net trained with such labels was not able to distinguish between the pumps, and therefore counting was not possible. In our scenario, both the components and the distance of the sensor to the component are always identical. Hence we arbitrarily determined a smaller label size and placed them in the area center of gravity of the segments (cf. Fig. 5). The bounding boxes are written into an XML file according to the annotation form of the Pascal VOC dataset (Everingham et al., 2010). Missing or incorrect annotations are corrected manually afterward.

4.4 Training of the neural networks

The training data set and associated annotations enable the training of neural networks of different architectures. The nets should be able to detect small and large pumps as well as misalignments and foreign objects. Consequently, the classes "large pump", "small pump" and "wrong" are used.

All networks have to be adjusted for training. First, the number of classes is reduced from 80 (number of COCO dataset classes) to three. As a measure to reduce the training effort and the amount of specific training data, we use a method called transfer learning, a proven and frequently used procedure. It is capable of improving the performance of the CNN, particularly its capability to generalize objects (Yosinski et al., 2014). Accordingly, we pre-train the CNN, using the freely available COCO dataset (Lin et al., 2014). This process is feasible in our case, as the features of the dataset are general enough to serve as input. Consequently, the first layers of the networks already recognize low-level features, and the subsequent layers are fine-tuned with our own pump training dataset. Due to different memory requirements, training parameters are adjusted separately for each network to achieve optimal training speed.

The most critical adjustment in terms of efficiency is the adjustment of the batch size, which determines how many images can be trained as a batch at the same time. Both SSD based networks require relatively little memory and can be trained with a GTX 1080 8GB with a batch size of 24. In contrast, the Faster R-CNN with ResNet101 can only be trained with a mini-batch of two images due to higher memory requirements. As a result, training all the networks on a GTX 1080 equipped workstation took between 6 and 8 hours.

5. RESULTS

The trained nets are evaluated with the remaining 20% of the data set. Correct detection means that a box has been defined for a class c where a ground truth bounding box is present (Fig. 6). The mean Average Precision (mAP) is used as a measure of reliability. It was developed for the Pascal VOC Challenge and indicates the precision of a detector in a single scalar value in the interval mAP $\in [0, 1]$. For comparison, both the Average Precision for each class and the mAP are used to get an insight into how well the networks work.

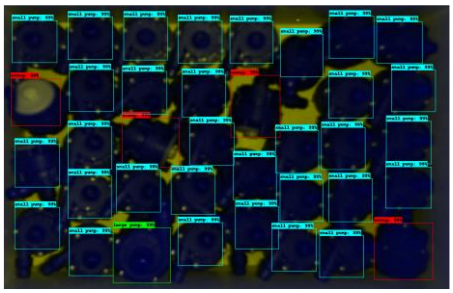

Fig. 6. Detected pumps and misaligned objects (Faster R-CNN with Resnet 101)

Table 1 illustrates the test results of the CNNs. Faster R-CNN with Inception v2 delivers the best result with a marginally better mean Average Precision. The SSD detector with the Inception v2 base network delivers comparable results but requires slightly less computing time. Since the measurements require one second, the time gain is not relevant in choosing this network over the other networks. The Faster R-CNN with the ResNet 101 base network only shows a slightly better performance with the larger pumps.

Table 1 Evaluation results of tested CNN

Network combinations	Faster R-CNN		SSD	
	R101	V2	MNet	V2
Computing time on GTX 1080 [ms]	106	58	**30**	42
Correct detection of large Pumps [%]	**99.8**	99.5	88.9	99.5
Correct detection of small Pumps [%]	99.7	**100**	52.3	99.6
Correct detection of unknown objects and misaligned positions [%]	99.3	99.9	73.2	**100**
mAP [%]	99.6	**99.8**	71.5	99.7

Mnet = Mobile Net, V2 = Inception V2, R101 = ResNet101,
mAP = mean Average Precision; Best values are marked **fat**.

The SSD with MobileNet as base network does not produce any useful results. MobileNet is a small network that designed for mobile devices with low memory requirements, but a lower detection rate than that of the slightly slower SSD with Inception v2. In terms of bounding box quality, Faster R-CNN networks produce more accurate results than SSD-based networks.

The results show that the method can detect, with good reliability, the number, positions, as well as the alignment of components belonging to a very limited number of different types in dense arrangements. The CNN architectures we used show high robustness and can separate many different object

classes. Therefore, we expect that the method will also be capable of distinguishing between a more significant number of different component types, e.g., all types of pumps that are inserted into bins at one work station. The method, however, cannot reliably distinguish between very similar sized and shaped components.

6. CONCLUSIONS AND OUTLOOK

We have investigated the suitability of deep learning and active depth cameras for counting of supply components. The results demonstrate that the chosen approach can achieve a sufficiently reliable counting of densely arranged objects in a short time to assist workers in counting different automotive components while they are put into standardized transport bins. The counting of components during packing into bins complements subsequent tracking of the parts by sensors attached to the transport bins by ensuring that components of the right types are supplied in the right quantities. Such enhanced, automated tracking of supply components can increase transparency within supply chains and thus their robustness against production, supply, and transportation disruptions.

Future work will deal with distinguishing between a greater number of different product types. In addition, it will focus on the simulation-based creation of synthetic training data in order to reduce the costs for training data acquisition.

ACKNOWLEDGEMENT

This work was funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) as part of the collaborative research and development project 01MA16004 “SaSCh – Digital Services for Shaping agile Supply Chains”.

REFERENCES

Börold, A. and Freitag, M. (2019). Real-time environmental analysis for industrial vehicles based on synthetic sensor data and deep learning. *Procedia CIRP*, 81, pp. 252-257. Amsterdam: Elsevier B.V.

Büttgen, B., et al. (2005). CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. In *Proceedings of the 1st range imaging research day*, September 8/9, 2005, Zürich, pp. 21-32.

Everingham, M., et al. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, *88* (2), pp. 303-338.

Finkenzeller, K. (2006). *RFID-Handbuch*. 4th ed., Munich: Hanser.

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.

Howard, A. G., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Ivanov, D., Dolgui, A., Sokolov, A., Ivanova, M. (2016). Disruptions in supply chains and recovery policies: state-of-the art review. *IFAC-PapersOnLine, Vol. 49*, pp. 1436–1441. Amsterdam: Elsevier B.V.

Klug, F. (2018). *Logistikmanagement in der Automobil-industrie*. Wiesbaden, Germany: Springer Vieweg,

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Krüger, J., Lehr J., Schlüter, M., Bischoff, N. (2019). Deep learning for part identification based on inherent features. *CIRP Annals*, Vol. 68, Issue 1, pp. 9–12. Amsterdam: Elsevier B.V.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *nature*, 521(7553), pp.436-444.

Lin, T. Y., et al. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

Musa, A.; Gunasekaran, A. & Yusuf, Y. (2014). Supply chain product visibility: Methods, systems and impacts. *Expert Systems and Applications*, 41, pp. 176–194.

Onoro-Rubio, D. and López-Sastre, R. J. (2016). Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pp. 615-629. Springer, Cham.

Rahman, M. S., & Islam, M. R. (2013, February). Counting objects in an image by Marker Controlled Watershed Segmentation and Thresholding. *Advance Computing Conference (IACC), 2013 IEEE 3rd International* (pp. 1251-1256). IEEE

Ren, S., et al. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, pp. 91-99.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR, vol. abs/*1409.1556*.

Staar B., Lütjen M. & Freitag M. (2018). Anomaly Detection with Convolutional Neural Networks for Industrial Surface Inspection. *Procedia CIRP, Vol. 79, pp. 484-489*. Amsterdam: Elsevier B.V.

Szegedy, C., et al. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826.

Teucke, M., et al. (2018). Using Sensor-Based Quality Data in Automotive Supply Chains. *Machines*, 6 (4). DOI: 10.3390/machines6040053.

Yosinski, J., et al. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems,* pp. 3320-3328.