Selection of Standard Parts under the Influence of Deep Learning

Despite the automation trend, manual assembly still represents an essential manufacturing step, which is associated with high time and cost expenditure. To meet these challenges, various optimization approaches for manual assembly are investigated. New potentials exist through the integration of object detection algorithms. Object detection is a subfield of computer vision and is concerned with determining the content and position of objects based on image properties (features). Deep learning applies when these features are extracted from diverse data by deep neural networks. These networks are trained on images containing all relevant information about the object to be recognized. In this way, images of all components and typical assembly defects can be integrated into an object detection model to monitor the assembly process. Overall, the application of Deep Learning holds great optimization potential for manual assembly. However, the question arises whether current products are appropriately designed for the use of such systems. Only if object detection algorithms can identify assembly components, their use in manual assembly is reasonable. Existing design guidelines do not consider this aspect yet. This research project investigates which properties standard parts should have, to enhance the accuracy of object detection algorithms.

Keywords: *Deep Learning, Object Detection, Computer Vision, Product Engineering, Design Guidelines*

Introduction

Despite the ongoing trend towards automation, most companies still assemble manually, resulting in high time and cost expenditure. In addition to the high labor costs in the European region, the growing diversity of variants presents companies with major challenges. Employees have to adapt more and more flexibly and quickly to new products and variants. The situation is further exacerbated by the increasing shortage of skilled workers.

To ensure that European companies can continue to hold their own in global competition, they must constantly optimize their processes. To support them in this, potential optimization approaches for manual assembly are being investigated in numerous ongoing research projects. One possibility to optimize manual assembly lies in the use of object detection algorithms. By capturing the assembly process using a camera and subsequent object detection, it is possible to determine the position, location, and type of components so that information about work steps, assembly errors, or the current state of the product can be derived.

Object detection is a subfield of computer vision and is concerned with identifying the content and position of various objects based on certain image properties (features). Deep learning takes place when these features are extracted from data by deep neural networks (Khan et al., 2018). Convolutional neural networks (CNNs) can be used for this purpose. For object detection, a CNN is trained with training images, on which the position and class of the objects of interest are marked. It is trained for a certain number of iterations and validated over a validation dataset until the detection quality is optimal. In this way, images of all components and typical assembly defects can be integrated into the object detection model.

For datasets such as COCO, CIFAR or ImageNet, the performance of deep learning models has been widely evaluated (Bochkovskiy et al., 2020). These datasets include various object classes relevant to everyday life (e.g., animals, buildings, flowers, or furnishings). However, objects from industrial assembly have different characteristics. While cats, for example, can take on countless shapes, objects in an industrial context are usually clearly defined. In order to recognize them reliably, it may be necessary to optimize the architecture of the neural network or to modify the shape of the objects. In the following, the second approach will be investigated. This approach has rarely been considered in the literature because the shape of animals, plants or buildings cannot be modified. In contrast, industrial products are easier to revise. Design guidelines, for example, could be used to provide guidance on how to design products in a way that is appropriate for object detection.

Overall, the use of object detection algorithms based on Deep Learning in manual assembly holds great potential for optimization. Assembly times could be reduced, employees could be relieved and assembly errors could be avoided. However, to ensure that the use of object detection algorithms in manual assembly is sensible, industrial products and all associated components should be recognized with reliability. To achieve this, a high level of detectability by object detection systems should already be considered during product design. At the moment, there are numerous design guidelines for an assembly-compatible design, but these do not address the aspect of integrating object detection.

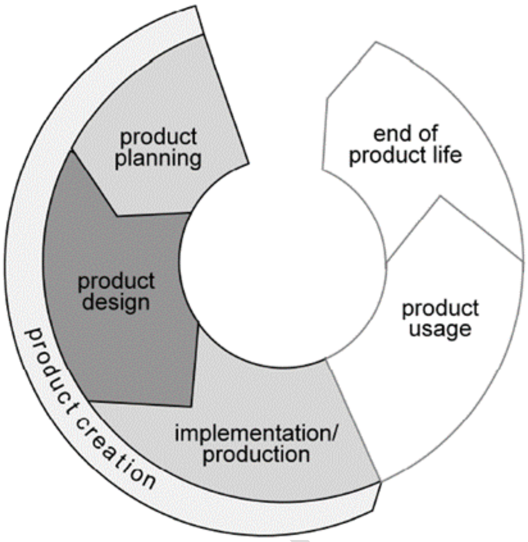
In order to fully exploit the potential of object detection algorithms based on deep learning in manual assembly, product design should be focused on this use case. For this reason, the research project investigates which design properties products and their components must have in order to enable optimal object detection. Since standard parts (especially screws) are installed in most industrial products, the research concentrates on this group of products.

Assembly-oriented product design

The Product Design Process

Assembly-oriented product design is an important part of the product creation process. This is part of the technical product life cycle and can be divided into the three phases of product planning, product design and implementation/production (VDI 2221 Part 1, 2019), see **Figure *1***. While product planning aims at the general design of the company's offering, product design is an interdisciplinary company process for the development of a marketable product, based on a definition of initial goals and requirements for the product, which are continuously developed and iteratively adapted during the process.

Figure 1. *Technical Product Lifecycle*



Source: VDI Richtlinie 2221 – Blatt 1, 2019

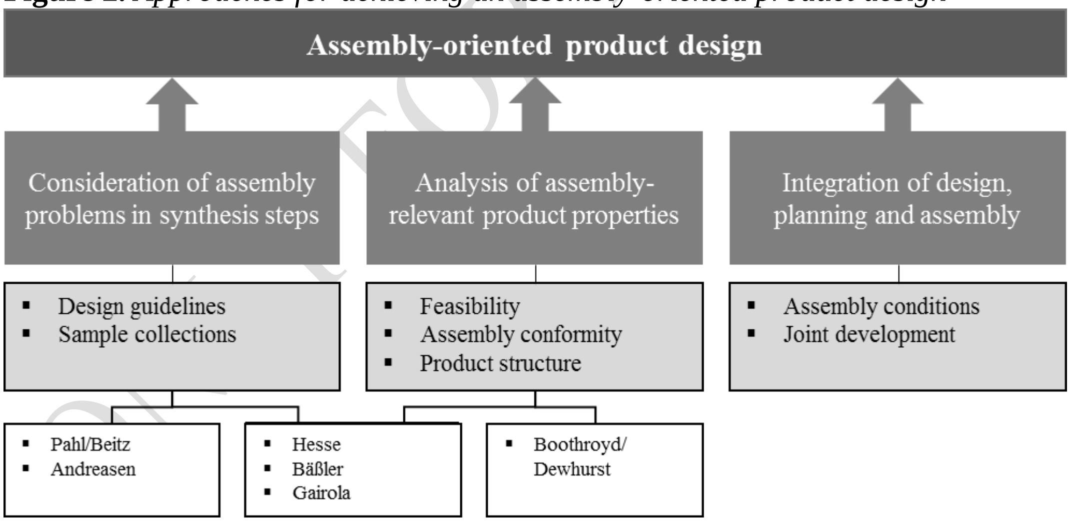
In order to increase the comprehensibility of the extensive and complex product development process, efforts have been made since the end of the nineteenth century to systematize this process. Reuleaux made the first considerations at the end of the nineteenth century by developing a model procedure for kinematic synthesis (Reuleaux, 1875). Further efforts to systematically structure the product development process were made by Wögerbauer (1942), Kesselring (1954), Hansen (1965), Hubka (1976), Koller (1976), Pahl et al. (1977), Roth (1982) and Rodenacker (1991). The aim of all these considerations was to provide design engineers with tools to develop products more efficiently. In order to establish a German standard and harmonize scientific findings, the first edition of VDI 2221 was published in 1986 (VDI 2221, 1986), which was last revised in 2019 due to new findings (VDI 2221 – Part 1, 2019). In addition to the primary guideline VDI 2221, the authors refer to additional guidelines such as VDI 2222 Part 1 (1997), VDI 2222 Part 2 (1982), VDI 2223 (2004) and VDI/VDE 2206 (2021), which specify special aspects and details of methodical product development and solution finding.

For example, VDI 2223 recommends the use of design guidelines, also known as design rules. These give designers concrete indications of potential design weaknesses and suggest suitable improvements. They can serve as a checklist for reviewing designs before they are released for elaboration. In literature, design rules are usually documented as collections of solutions or in design catalogs. Design guidelines can refer to a wide range of aspects. For example, there are rules for stress-appropriate, cost-appropriate, welding-appropriate or production-appropriate design. However, for manual assembly, the information on product design for assembly is particularly relevant.

Approaches for an assembly-oriented product design

Product design has a decisive influence on the future assembly process. For example, the product design influences the structure of the assembly systems, the future joining processes and the options for feeding, storing and separating components. For this reason, the assembly processes that will be required later should already be considered during the design phase. In literature, different approaches can be found to realize an assembly-oriented product design, see **Figure *2***.

Figure 2. *Approaches for achieving an assembly-oriented product design*



Source: Own illustration based on Ehrlenspiel, 1995

On the one hand, concrete design guidelines for assembly-oriented design are provided in form of collections of examples and design guidelines. The authors recommend that assembly problems should already be taken into account during the synthesis steps in the product development process. Well-known collections of examples can be found in Pahl et al. (2007), Andreasen et al. (1983), Hesse (1995), Bäßler (1988) and Gairola (1981).

Another approach is the analysis of assembly-relevant product properties. Here, a systematic product analysis is used to assess the suitability for assembly and to identify weak points. Concrete methods can be found in Hesse (1995), Bäßler (1988), Gairola (1981) and Bothroyd et al. (1983).

The third approach concerns the integration of design, planning and assembly. Organizational measures can improve cooperation between these various departments so that the flow of information is optimized. An integrated process should improve planning quality, reduce development time and minimize costs. The discussed approaches are not comparable, but they are highly complementary. Ideally, all three approaches are combined (Ehrlenspiel, 1995).

In the context of the paper, the first approach will be pursued. The aim is to derive concrete design guidelines from practical tests. These shall serve designers as a tool to design products in a way that they are reliably recognized by object detection algorithms based on Deep Learning.

Deep Learning based Object Detection

Object detection has attracted increasing attention in recent years due to its wide range of applications and recent technological breakthroughs. This task is being intensively studied both in academia and in practice, e.g. in security surveillance, autonomous driving, traffic monitoring, drone scene analysis and robot vision. Among the many factors and efforts that have led to the rapid development of object detection techniques, notable contributions should be attributed to the development of deep convolutional neural networks and the computational power of GPUs. Currently, the deep learning models are widely used throughout the computer vision field, including general object detection and domain-specific object detection. Most state-of-the-art object detectors use deep learning networks as a backbone and recognition network to extract, classify or localize features from input data.

The task of identifying and localizing objects on RGB-image data is crucial for many applications in the industrial environment. Typically, deep learning-based methods achieve this goal by using features extracted by convolutional neural networks (CNNs) (Lecun et al., 1998). CNNs are built by alternately stacking convolutional and pooling operations with learnable weights. Convolutional layers convolve one or more parameterized kernel with the respective input data and thus create feature maps. These contain low-level features like edges and textures in the networks first layers and high-level features like object information in the deeper layers.

To reduce the number of parameters in the networks, the spatial feature map size is scaled down using pooling operations such as max pooling or average pooling. The pooling operation is typically performed over a 3 x 3 sliding window of the input data (Wu, 2017). This is one of many applied feature-space dimensionality reduction methods that lead to a more stable training and better generalization of the trained model.

Typically, the output of each networks layer is applied to an activation function to introduce the necessary non-linearity of a neural network. The mostwidespread activation function is the rectified linear unit (ReLU) f(x)=max(0, x) (Nair et al., 2010).

By stacking pooling and convolutional layers, a network can be built that enables a hierarchical and evolutionary development of pixel data towards meaningful feature representations. These can be used for the mentioned downstream tasks such as classification and localization of objects. Many of the downstream applications use Residual Networks (He et al., 2016) as backbone, whereas other types of neural networks like Transformers (Dosovitskiy et al., 2021) have shown large success as feature extractors.

Object detection methods can mainly be divided in two-shot methods, where object regions are predicted first and classified in a separate neural pathway (Girshick et al., 2014) and one-shot methods where location and class of objects are derived from the same features (Redmon et al., 2016). In the present work, a region proposal-based method (Ren et al., 2016) coupled with a feature pyramid network (Lin et al., 2017) is used. Two-shot approaches typically show a higher object detection accuracy at a higher computational cost than single-shot-detectors.

Faster-RCNN introduces the concept of a region proposal network (RPN) to determine regions of interest (RoI) that will be classified. The RPN is a fully convolutional network to predict object bounds and objectness scores at each image position. The RPN slides of selected feature maps of the backbone CNN, obtaining a low-dimensional vector at each position. This vector is fed into a classifier and regressor to classify if there is an object and regress its position. Each region then is parameterized relative to a reference anchor box. The distance between the predicted box and the ground truth position is measured to only output the offset between the predicted box and the anchor.

Feature pyramids built upon image pyramids have been widely applied in many object detection systems to improve scale invariance. The FPN consists of a bottom-up pathway and a top-down pathway of differently sized featured maps from the backbone CNN. It so combines low-resolution and semantically strong features with high-resolution and semantically weak features.

Due to its architecture, a vanilla Faster-RCNN network will output multiple, slightly shifted bounding boxes for each object. These must be filtered for duplicates to achieve useful prediction results. NMS (non-maximum suppression) is a heuristic method which selects only the object of the highest classification score, otherwise the object will be ignored. This method is commonly used in almost all object detection architectures, despite there is active research toward NMS-free methods (Hosang et al., 2017).

The resulting network can be trained end-to-end using bounding-box annotated image data. Typically, this is done using stochastic gradient descent on a combined loss function to both minimize the bounding-box position error and the logistic classification error. Typically, the network is trained on a large dataset such as Microsoft Common Objects in Context (Lin et al., 2014). COCO contains 80 objects classes and 328K images. The pre-trained network is then finetuned for a specific task such as detection of screws or other relevant objects.

Current research in the field of unsupervised object classification and detection shows the general viability of training networks without annotated data. These approaches allow to use much larger amounts of training data (and thus more accurate networks) and use models in even more application field, where data annotation is inefficient. Especially methods such as contrastive learning (Chen et al., 2020) have gained latest attention as standard CNNs trained under this unsupervised method achieve similar accuracy compared to supervised training procedures.

The broad variety of deep learning methods led to a large number of different methods for localizing and classifying objects. Faster-RCNNs have been around for many years and thus are well understood in both their technical details and real-world behavior. Newer methods such as shifting window transformers (Liu et al., 2021) or the YOLO family with YOLOR (Wang et al., 2021) as state-of-the-art method might outperform Faster-RCNN in terms prediction performance. The research objective in this work however is a real-world problem, where not latest research trends are deployed but well understood methods.

Materials and Methods

Null hypotheses and alternative hypotheses

In order to develop design guidelines, the first step was to formulate theses concerning an optimal selection of standard parts. The basic functionality of object detection algorithms was used as the basis for the theses. The object detection algorithms available on the research stage are based on CNNs, which consist of several layers arranged one after the other. The receptive field of the individual layers increases the deeper they are in the network. During the training process, the first layers learn to recognize simple image features such as corners and edges. Deeper layers respond to more complex contours and higher dimensional features. If image features are assigned to an incorrect object class due to their high similarity, the object detection algorithms can no longer reliably identify the objects to be detected. On this basis, the hypotheses listed in Table 1 were formulated.

Since a hypothesis cannot be statistically proven, but only disproven, null and alternative hypotheses were formulated. If a null hypothesis can be refuted by the experiments described below, the respective alternative hypothesis can be accepted. (Siebertz et al., 2017)

Table 1. *Null hypotheses and alternative hypotheses*

No.	Null hypothesis	Alternative hypothesis
1	Screws of the same type are detected even if their nominal size differs by only one size increment. $mAP > 75$	Screws of the same type are confused if their nominal size differs by only one size increment. $mAP \leq 75$
2	Screws with similar screw heads will be detected reliably. $mAP > 75$	Screws with similar screw heads will be confused. $mAP \leq 75$

The Mean Average Precision (mAP) is a quantitative specification for the recognition accuracy of an object detection model and will serve as an indicator in the experiments. Due to the high complexity and individuality in applications of object detection algorithms, no concrete minimum value for the mAP is defined in the literature. Depending on the use case, a higher or lower mAP may be sufficient for reliable object detection. In the context of this paper, a mAP higher than 75% is assumed to be sufficient.

Dependent variables and independent variables

The hypotheses were tested by experiments. The basis of the experimental design was the determination of independent and dependent variables. As described before, the mAP was used as the dependent variable. The definition of mAP used in this work is analogous to the metric mAP@[50%:95%] used in the COCO dataset. It describes the average of the variant Average Precision values under the assumption that a positive detection is evaluated at 50%, 55%, ..., to 95% agreement between prediction and actual position of the object. The Average Precision is obtained from the area under the Precision (Recall) function, where Recall is defined as the correct positive rate (sensitivity) and Precision is defined as the positive prediction value (accuracy) (Henderson et al., 2017).

In this way, the mAP evaluates both the quality of classification (Are the correct objects detected?) and of position determination (Is the position of the objects on the images detected correctly?) by the prediction model. With regard to the theses under investigation, it seems appropriate to provide a meaningful value on detection accuracy. (Henderson et al., 2017)

Regarding the independent variables, a distinction must be made between influencing variables which are intended to be varied deliberately (factors) and influencing variables which are kept constant (control variables). Since the optimal standard part selection for high detectability is supposed to be investigated by object detection algorithms, the screw diameter as well as the head shape of the screws were defined as factors to be varied.

Object detection models are usually trained and validated on a database consisting of images of the objects to be detected. Since these are essential for the recognition accuracy, all variables that influence the image characteristics have to be defined as independent variables and kept constant during the tests. This ensures that only the influence of the standard part selection on the mAP is measured. Besides the illumination intensity, image background, image quality, image perspective as well as the number of training and validation images used have to be kept constant. Finally, there are a variety of different object detection algorithms that can be used for object detection. Since the type of model used also has an impact on mAP, one algorithm must be selected and applied for the experiments.

Experimental Setup

After defining the independent and dependent variables, an experimental setup was planned which allows the control variables to be kept constant while the screws characteristics are varied.

The principle procedure of the investigation is shown in Figure 3. First, screws with different diameters and screw heads were selected. In addition to grub screws, cylindrical head, countersunk head and lens head screws were procured in nominal sizes M3 to M10. Since screws in assemblies usually occur in a bolted state, elements were designed to house the screws. These were simple aluminum cubes which were provided with threaded holes. In order to investigate the previously established hypotheses, similar screw diameters and screw heads were selected in each case and illuminated whether they could be distinguished by an object detection algorithm. The selected screw pairs can be taken from Table 2.

Figure 3. *Experimental Setup*

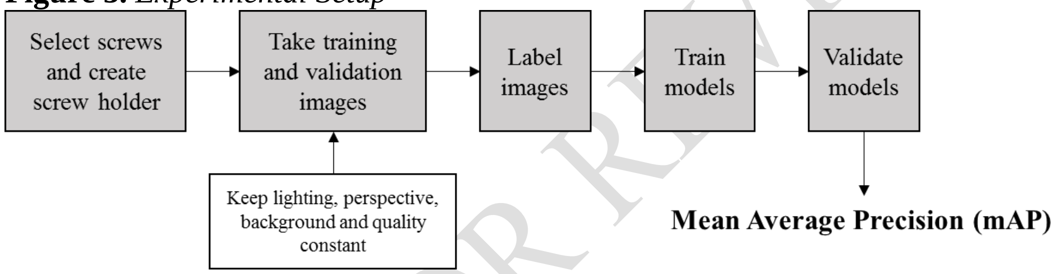
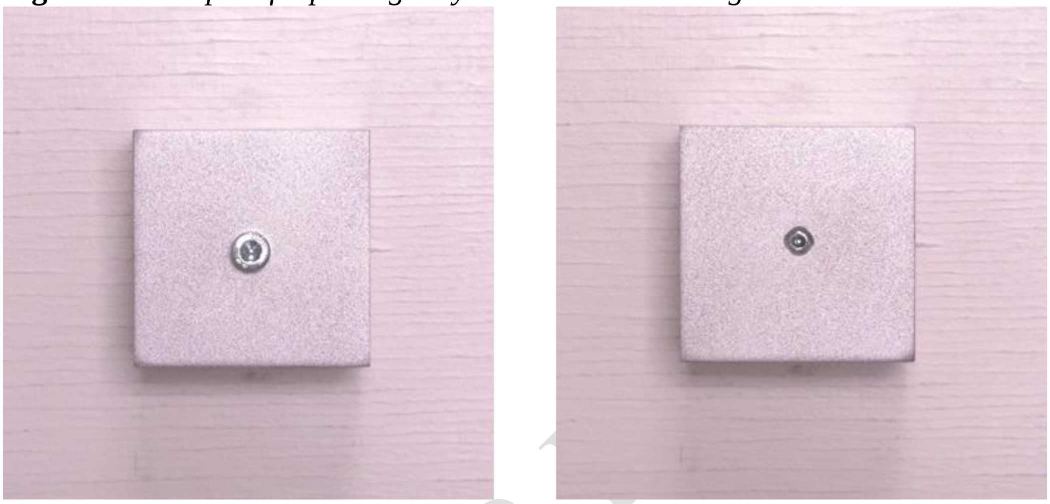


Table 2. *Selected Screw Types - The object detection models are each trained with images of both types and are designed to differ between the two types.*

Hypothesis	Screw type 1	Screw type 2
1	Cylinder head screw M3	Cylinder head screw M4
	Cylinder head screw M5	Cylinder head screw M6
	Cylinder head screw M8	Cylinder head screw M10
	Lens head screw M3	Lens head screw M4
	Lens head screw M5	Lens head screw M6
	Lens head screw M8	Lens head screw M10
	Countersunk screw M3	Countersunk screw M4
	Countersunk screw M5	Countersunk screw M6
	Countersunk screw M8	Countersunk screw M10
	Grub screw M3	Grub screw M4
	Grub screw M5	Grub screw M6
	Grub screw M8	Grub screw M10
2	Cylinder head screw M3	Lens head screw M3

	Countersunk screw M3	Grub screw M3
	Cylinder head screw M6	Lens head screw M6
	Countersunk screw M6	Grub screw M6
	Cylinder head screw M10	Lens head screw M10
	Countersunk screw M10	Grub screw M10

Figure 4. *Example of a pairing - Cylinder head M3 and grub screw M3*



Afterwards, images of the inserted screws were created. **Figure *1*** shows an example of a pair consisting of a cylinder head M3 and a grub screw M3 (hypothesis 2), which should be distinguished. In order to keep the illuminance constant, these images were recorded in a room illuminated exclusively with artificial light, and the illuminance was continuously monitored using a calibrated lux meter. The objects were placed on an electric turntable, which has a rotation speed of 60 seconds per revolution and a white surface, at the same position each time. In addition, a camera (Sony Alpha 6300 with SELP1650 lens) was focused on the objects of interest. To capture images of the screws, the turntable was set in rotation and, starting at a defined start mark, an image was recorded every second. The settings as well as the position of the camera to the object were kept constant during the recording of each test object.

After all objects had been recorded from one perspective, the camera changed position in order to record all objects from another perspective and thus generate a larger number of images. In total, images were recorded from ten different perspectives. The described procedure ensured that the object images differed only in their visual appearance.

For each screw type, a total of 360 images were recorded and divided into training and validation images. Every fifth image was defined as a validation image, while the remaining images were used as training images. Subsequently, both the training and validation images were labeled. For this purpose, it was manually marked on the images which screw is contained and at which position it is located. Afterwards, the model was trained.

For this purpose, a Faster-R CNN model with ResNet-50 backbone and a feature pyramid network head was used, which represents a good compromise between detection accuracy, training time, inference time, and model size at the time of the study. The model architecture, as well as a model pre-trained over ~37 epochs on the COCO dataset, were taken from the PyTorch framework detectron2 developed by FAIR. (Wu et al., 2021; Ren et al., 2016). Each model contained two object instances, for which in each case 240 images were used for training and 60 images for validation.

The detectron2 framework enables a low development effort for object detection methods, offers a variety of different model architectures, and is more performant in terms of training and inference time than a pure PyTorch implementation. Using a pre-trained model is common practice because it eliminates learning simple contours of the first network layers. This has a positive effect on the training time and the number of frames needed for training.

On the pre-trained model, the training images were trained locally for 40,000 iterations and validated using the validation dataset. The obtained results are explained in the following section.

Results

The results indicate that all screws considered can be identified by the object detection with high reliability. For all models, the mAP is significantly higher than the value of 75 described in Materials and Methods.

In order to obtain statistically valid results, the experiment was repeated three times and the one-sided one-sample t-test was performed. For this purpose, the t-value for each model was calculated according to Equation 1 and compared with the critical t-value for $\alpha = 0.05$ ($t = 2.920$) according to Graf et al. (1998). If the calculated t-value is higher than the critical t-value, the difference between $\bar{X}$ and μ_0 is significant (Schneider, 2020).

Table ***3*** shows that for all models, T > t.

$$T = \frac{\bar{X} - \mu_0}{S}\sqrt{n} \qquad \textit{Equation 1}$$

$\bar{X}$ Mean average of test results (= $\overline{mAP}$)

μ_0 Value tested against (= 75)

S Standard deviation

n Size of sample (= 3)

Table 3. *Experimental Results*

Hypothesis	**Model**	$\overline{mAP}$	T
1	Cylinder head M3 + Cylinder head M4	98.14	49.2
	Cylinder head M5 + Cylinder head M6	98.92	39.4
	Cylinder head M8 + Cylinder head M10	98.89	34.8
	Lens head M3 + Lens head M4	97.65	93.4
	Lens head M5 + Lens head M6	99.55	123.4
	Lens head M8 + Lens head M10	99.52	266.2
	Countersunk M3 + Countersunk M4	91.68	10.9
	Countersunk M5 + Countersunk M6	91.62	4.8
	Countersunk M8 + Countersunk M10	93.40	7.0
	Grub screw M3 + Grub screw M4	81.64	5.5
	Grub screw M5 + Grub screw M6	84.16	6.0
	Grub screw M8 + Grub screw M10	85.77	3.1
2	Cylinder head M3 + Lens head M3	96.18	26.4
	Countersunk M3 + Grub screw M3	85.56	5.6
	Cylinder head M6 + Lens head M6	98.69	42.1
	Countersunk M6 + Grub screw M6	88.55	7.0
	Cylinder head M10 + Lens head M10	99.40	40.7
	Countersunk M10 + Grub screw M10	89.45	9.4

Despite the very high overall detection accuracies, some models detect screws more accurately if they have larger nominal diameters. For example, grub screws are better distinguished from one another the larger their diameter (81.64 < 84.16 < 85.77). Considering the models with countersunk screws and grub screws, it becomes clear that screws achieve higher mAP when their absolute size increases (85.56 < 88.55 < 89.45). These results lead to the assumption that some screws are better detected the wider their absolute diameter tends to be.

In addition, the test results show that the models containing grub screws and countersunk screws have slightly lower recognition accuracies compared to the other models. Unlike lens head and cylinder head screws, these two types of screws are fully sunken into the component. This leads to the assumption that screws are better detected if their head does not entirely sink into the component.

However, even the model with the lowest mAP (81.64) achieves a very high recognition accuracy and is able to identify the contained objects with high reliability. Therefore, the present investigation does not result in any design restrictions with regard to the selection of standard parts.

Conclusions

No restrictions for designers with regard to the selection of standard parts can be derived from the results of the investigation. All screws considered in the investigation were detected reliably by the algorithm used. This shows that a conventional Faster-R-CNN model has no problems recognizing screws with very similar optical properties. Thus, neither an optimization of the architecture of the model nor of the selection of standard parts is necessary.

However, in addition to the theses considered, there are a number of other product features which are not addressed in the present study. For example, it is necessary to examine the influence of certain assembly characteristics on recognition accuracy and the extent to which similar design elements (e.g. grooves, bores, etc.) can be distinguished. Since the screws in the present investigation were deliberately tested in simple aluminum cubes in order to minimize interference, it is also necessary to determine whether the results can be transferred to standard parts in real products. The product environment - e.g. the assembly workplace, tools or body parts of assembly workers - represents a significant optical interference influence and could affect the detection capability of the object detection algorithms. Experience shows that environmental conditions have a great influence on the quality of object detection algorithms, so that different results can be expected here with high probability. For this reason, further tests must be carried out in a real product environment to verify the results.

Overall, the research project represents a first step towards deriving design guidelines for the use of object detection algorithms in manual assembly, which must be followed by further investigations. Only after a comprehensive examination of the topic a statement can be made whether the use of object detection algorithms in manual assembly results in restrictions with regard to the product design.

References

Andreasen M, Kähler S, Lund T (1983) *Design for Assembly*. Berlin: Springer

Bäßler R (1988) *Integration der montagegerechten Produktgestaltung in den Konstruktionsprozess* [Integration of assembly-oriented product design into the design process]. Berlin: Springer. DOI: 10.1007/978-3-662-07930-0

Bochkovskiy A, Wang CY, Liao HY (2020) *YOLOv4: Optimal Speed and Accuracy of Object Detection*. https://arxiv.org/pdf/2004.10934.pdf, published on 23.04.2020, called up on 07.04.22

Bothroyd G, Dewhurst P (1983) *Design for assembly: a designer's handbook.* Amherst, Ma: University of Massachusetts, Dept. of Mechanical Engineering

Chen T, Kornblith S, Norouzi M, Hinton G (2020) *A Simple Framework for Contrastive Learning of Visual Representations.* In: Proceedings of the 37th International Conference on Machine Learning. DOI: 10.5555/3524938.3525087

Dosovitskiy A, Kolesnikov A, Weissenborn D, Heigold G, Uszkoreit J, Beyer L, Minderer M, Dehghani M, Houlsby N, Gelly S, Unterthiner T, Zhai X (2021) *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. In: The Ninth International Conference on Learning Representations. DOI: 10.48550/2010.11929

Ehrlenspiel K (1995) *Konstruktionstechnik München* [Construction technology Munich]. Volume 21. Munich, Vienna: Carl Hanse

Gairola A (1981) *Montagegerechtes Konstruieren – Ein Beitrag zur Konstruktionsmethodik* [Designing for Assembly - A Contribution to Design Methodology]. Doctoral Thesis. Technische Hochschule Darmstadt

Girshick R, Donahue J, Darell T, Malik J (2014) *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. DOI: 10.1109/CVPR.2014.81

Graf U, Henning HJ, Stange K, Wilrich PTh (1998) *Formeln und Tabellen der angewandten mathematischen Statistik* [Formulas and tables of applied mathematical statistics]. 3rd Edition. Berlin: Springer. ISBN: 978-3-540-16901-7

Hansen F (1965) *Konstruktionssystematik: Grundlagen für eine allgemeine Konstruktionslehre* [Design systematics: basics for a general design theory]. Berlin: Verlag Technik

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. DOI: 10.48550/1512.03385

Henderson P, Ferrari V (2017) *End-to-end training of object class detectors for mean average precision.* https://arxiv.org/pdf/1607.03476.pdf, published on 16.03.2017, called up on 08.04.22

Hesse S (1995) Montage-Atlas [Assembly atlas]. Wiesbaden: Vieweg+Teubner

Hosang J, Benenson R, Schiele B (2017) *Learning non-maximum suppression*. https://arxiv.org/pdf/1705.02950.pdf, published on 09.05.2021, called up on 08.04.2022

Hubka V (1976) *Theorie der Konstruktionsprozesse: Analyse der Konstruktionstätigkeit* [Theory of construction processes: Analysis of construction activity]. Berlin: Springer. DOI: 10.1007/978-3-642-81035-0

Kesselring F (1954) *Technische Konstruktionslehre* [Technical engineering design]. Berlin: Springer

Khan S, Rahmani H, Shah SA, Bennamoun M (2018) *A Guide to Convolutional Neural Networks for Computer Vision*. Williston, VT: Morgan & Claypool

Koller R (1976) *Konstruktionsmethode für den Maschinen-, Geräte- und Apparatebau* [Design method for machine, device and apparatus construction]. Berlin: Springer. DOI: 10.1007/978-3-642-96295-0

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) *Gradient-based learning applied to document recognition*. In: Proceedings of the IEEE. Volume 86. DOI: 10.1109/ 5.726791

Lin T, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) *Feature Pyramid Networks for Object Detection*. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA: IEEE Computer Society. DOI: 10.1109/CVPR.2017.106

Lin T, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollar P (2014) *Microsoft COCO: Common Objects in Context*.

https://arxiv.org/pdf/1405.0312.pdf, published on 21.02.2015, called up on 08.04.2022

Liu Z, Lin Y, Cao Y, Hu h, Wie Y Zhang Z, Lin S, Guo B (2021) *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Los Alamitos, CA: IEEE Computer Society. DOI: 10.1109/ICCV48922.2021.00986

Nair V, Hinton G (2010) *Rectified Linear Units Improve Restricted Boltzmann Machines.* In: Proceedings of the 27th International Conference on International Conference on Machine Learning. Madison, WI: Omnipress

Pahl G, Beitz W (1977) *Konstruktionslehre. Handbuch für Studium und Praxis* [Engineering design. Handbook for study and practice]. Berlin: Springer. DOI: 10.1007/978-3-662-02288-7

Pahl G, Beitz W, Feldhusen J, Grote KH (2007) *Pahl/Beitz Konstruktionslehre. Grundlagen* [Pahl/Beitz Engineering design. Basics]. 7th Edition. Berlin: Springer. DOI: 10.1007/978-3-540-34061-4

Redmon J, Divvala S, Girshick R, Farhadi A (2016) *You only look once: Unified, real-time object detection.* In: Proceedings of the IEEE conference on computer vision and pattern recognition. DOI: 10.48550/1506.02640

Ren S, He K, Girshick R, Sun J (2016) *Faster R-CNN: Towards Real-Time Objekt Detection with Region Proposal Networks*. https://arxiv.org/pdf/1506.01497.pdf, published on 06.01.2016, called up on 08.04.22

Reuleaux F (1875) *Lehrbuch der Kinematik* [Course book of kinematics]. Braunschweig: Vieweg

Rodenacker WG (1991) *Methodisches Konstruieren: Grundlagen, Methodik, praktische Beispiele* [Methodical design: Fundamentals, methodology, practical examples]. 4th Edition. Berlin: Springer. DOI: 10.1007/978-3-642-87484-0

Roth K (1982) *Konstruieren mit Konstruktionskatalogen* [Designing with design catalogs]. Berlin: Springer. DOI: 10.1007/978-3-662-08152-5

Schneider M (2020) *Datenanalyse für Naturwissenschaftler, Mediziner und Ingenieure* [Data analysis for scientists, medics and engineers]. Berlin: Springer. DOI: 10.1007/978-3-662-61866-0

Siebertz K, Bebber D, Hochkirchen T (2017) *Statistische Versuchsplanung, Design of Experiments (DoE)* [Statistical design of experiments (DoE)]. 2nd Edition. Berlin: Springer Vieweg

VDI/VDE 2206 (2021) Entwicklung mechatronischer und cyber-physischer Systeme [Development of mechatronic and cyber-physical systems]. Duesseldorf: VDI-Verlag

VDI 2221 (1986) *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte* [Systematic approach to the development and design of technical systems and products]. Duesseldorf: VDI-Verlag

VDI 2221 Part 1 (2019) *Entwicklung technischer Produkte und Systeme. Modell der Produktentwicklung* [Design of technical products and systems. Model of product design]. Duesseldorf: VDI-Verlag

VDI 2222 Part 1 (1997) *Konstruktionsmethodik. Methodisches Entwickeln von Lösungsprinzipien* [Design engineering methodics. Methodic development of solution principles]. Duesseldorf: VDI-Verlag

VDI 2222 Part 2 (1982) *Konstruktionsmethodik. Erstellung und Anwendung von Konstruktionskatalogen* [Design engineering methodics. Setting up and use of design catalogues]. Duesseldorf: VDI-Verlag

VDI 2223 (2004) *Methodisches Entwerfen technischer Produkte* [Systematic embodiment design of technical products]. Duesseldorf: VDI-Verlag

Wang CY, Yeh IH, Liao HY (2021) *You only learn one representation: Unified network for multiple tasks*. https://arxiv.org/pdf/2105.04206.pdf, published on 10.05.2021, called up on 08.04.2022

Wögerbauer H (1942) *Die Technik des Konstruierens* [The technique of construction]. Munich: Oldenbourg

Wu J (2017) *Introduction to Convolutional Neural Networks*. https://www.semanticscholar.org/paper/Introduction-to-Convolutional-Neural-Networks-Wu/450ca19932fcef1ca6d0442cbf52fec38fb9d1e5, called up on 08.04.2022

Wu Y, Kirillov A, Massa F, Lo WY, Girshick R (2019) *Detectron 2*. https://github.Com/facebookresearch/detectron2, called up on 22.07.2021