2nd Conference on Production Systems and Logistics

Introducing a Decision Support System for Use-Case Specific Object Detection Methods in Production Systems

Kai Müller[1], Stephan-Andrés Saß[1], Christoph Greb[1]

Institut für Textiltechnik (ITA) of RWTH Aachen University, Otto-Blumenthal-Str.1, 52074 Aachen, Germany

Abstract

Object detection has the potential to facilitate the automation of optical quality inspection, achieving a significant reduction of human error. However, the gap between needed expertise to understand and integrate complex object detection systems into production environments and the availability of computer scientists, is hindering its use in the manufacturing industry. A support system for decision-makers unrelated to the subject is therefore required to promote industry utilisation of object detection and effectively manage this otherwise unused opportunity of profitable knowledge. Within the Cluster of Excellency "Internet of Production (IoP)", such a support system has been developed. Lowering the implementation hurdle of object detection systems is achieved by translating complex information about existing methods into tangible factors, such as quality and cost. In this work we aim to structure relevant object detection techniques and employ a decision tree to provide a user-support based on a use-case-oriented framework. The use-case's basic conditions and requirements serve as input for the framework. The decision tree gives the suitable object detection method as output, accordingly. For traditional object detection techniques, the characteristics are translated into basic requirements, which are then used as input, e.g., for the template matching method the comparison of a source image with a reference image is translated to the ability to guarantee images for all possible quality deviations. The deep learning (DL) methods are consolidated in the project management triangle consisting of quality, cost, and time. Firstly, an introduction into object detection is given. Secondly, traditional methods are clustered and deep learning methods classified. A description of the decision tree is then presented, before testing results conclude this paper. The developed support system enables decision-makers to evaluate object detection methods for individual use-cases and consequently achieve increased production planning efficiency. The system's universal design allows for application across manufacturing industries and use-cases.

Keywords

Object Detection; Computer Vision; Deep Learning; Quality Management; Digitalisation; Automation; Internet of Production

1. Introduction

Several quality management tasks in the manufacturing environment are currently performed manually. As fatigue increases and focus decreases, the occurrence of human error implies that a constantly high working standard cannot be guaranteed [1]. Introducing object detection into these use-cases helps to minimise human errors, as machine performance is constant. Historically however, this approach is only an option for enterprises that employ programmers for this specific task and have access to large amounts of computing power and data needed for computer vision tasks. With the increased amount of available computing power, object detection becomes more realistic for most enterprises. However, the recent development in computing power and amount of readily available data has increased object detection possibilities at a higher pace than computer scientists have been educated [2]. This leads to a gap between potential and fulfilled opportunity, due to the scarcity of expertise that is able to perform potential tasks. In this paper we propose a support system based on a decision tree to translate the complex parameters of object detection methods into more tangible factors to support closing this gap.

2. Related Work

Object detection aims to detect instances of objects from given categories and return their spatial location [3]. Historically, so-called traditional object detection methods have been used. With advancements in data obtainability and availability of computing power, several deep learning (DL) methods have been introduced in recent years [4]. The following chapters will give a brief overview and explanation of these two types of methods.

2.1 Traditional object detection methods

Traditional object detection methods can be divided into three stages: *image segmentation*, *feature extraction* and *classification* [5]. Additionally, we consider the *image pre-processing* for a holistic approach that covers the entire process from the captured image to the detected objects. In traditional object detection each of the aforementioned stages is performed separately. The goal of the stages leading up to the classification is to extract features in the form of a feature vector that can consequently be used in the classification to determine which objects are depicted in each image [6]. This happens by observing sets of pixels through a sliding window and moving that window across the entire image. This approach has two limitations: Firstly, each relevant feature needs to be manually selected to perform the classification task. Secondly, these methods become computationally intensive and slow because each image is analysed incrementally window by window [6,7].

2.2 Deep learning-based object detection methods

DL is a subfield of Machine Learning (ML) and employs an Artificial Neural Network (ANN) to recognise and learn the patterns in a given dataset [8]. Images serve as input and the classification as output. To train an ANN a set of labelled data is needed. In the case of object detection, a labelled data set consists of images in which the relevant objects are marked and given the label of the class that this object belongs to [9]. The ANN learns how to classify objects within a given image by comparing the predicted class with the labelled objects in the image and adjusting the biases of the neurons and the weights of the connections between them [8]. In comparison to traditional object detection methods a disadvantage of DL methods is, that they follow a black box model, making the process from input to output difficult to comprehend. However, research is currently being done to improve the transparency of deep learning models [10,11].

3. Proposed concept

The initial aim of the developed support system presented in this paper was to suggest an algorithm or DL model that is best suited for the use-case at hand. After having reviewed fundamental literature, it has become clear, that little to no research has been done on this concept. An overwhelming number of algorithms and DL models represent a barrier for the development of a decision tree comprising the entire spectrum. Instead, in this paper we suggest the foundation for the development of an exhaustive decision support system, by clustering the algorithms and DL models according to the similar characteristics they exhibit. Traditional object detection algorithms can be grouped together based on the input they receive and the output they generate. These groups can further be clustered according to the techniques the algorithms use to achieve the output. A similar approach for DL models is presented in 3.2

3.1 The traditional object detection process

The traditional object detection process is divided into the here defined stages *image pre-processing*, *image segmentation*, the *feature extraction* and the *classification* [5]. Since different algorithms with different parameters apply to each of these four stages, the algorithms are clustered for each stage individually. This way the relevant cluster can be suggested based on the given use-case.

3.1.1 Image pre-processing

Image pre-processing is a processing step in which the image is transferred into a new image. The new image is similar to the original version, but is adapted to facilitate further processing, e.g. by increasing contrast. [12] Image pre-processing algorithms are further clustered according to their functionality into *photometric* and *geometric* image processing. *Photometric* processing uses the brightness of each pixel or the average brightness of adjoined pixels in the original image to calculate the brightness of the pixel in the new image. [12] *Geometric* processing keeps the grayscale value constant, but changes the coordinates of the pixels in the original image to form the new image [13].

3.1.2 Image segmentation

In the stage of image segmentation, regions of interest are defined and isolated [12]. Regions of interest could be the contour of a borehole or letters of the alphabet. This stage can be clustered into four types of segmentation algorithms: *grayscale segmentation*, *contour tracing*, *template matching* and *corner detection*. [7] The *grayscale segmentation* algorithms rely on thresholds of grayscale values. Through the calculation or pre-set of a threshold value, the processed image is turned into a binary image. This means that objects are isolated according to whether they are above or below a pre-set grayscale value. [7] The contour of an object is the organised order of the objects edge points. *Contour tracing* is thus well suited for the detection of large objects with high contrast. [7] In *Template Matching*, a given template from an image block is searched for in a larger target image with the prerequisite that the template may only experience translation or rotation to fit the object in the target image [7]. Template Matching can therefore be used for quality control, if the analysed product must fit a certain template [14]. An *edge detector* searches for the specific location of an edge when the region of that edge is clear. An edge is located according to its direction, height and length, where the direction is defined as whether the grayscale value increases or decreases in the direction of search, the height is difference in grayscale value between neighbouring pixels and the length is described as the number of pixels between which the set height has to occur. [12]

3.1.3 Feature extraction

After having isolated the regions-of-interest, characteristic features are calculated. This process is called feature extraction. During feature extraction, a set of descriptors are computed that help to classify and label objects into categories. [15] Such a set is known as a feature vector. Based on the types of objects isolated, feature extraction algorithms can be clustered into two types: *area-based* features and *contour-ba*sed features. *Area-based algorithms*: These algorithms extract information related to the area of the object. An example for this is the area itself. In some cases, however, the area might not be as relevant as for example the centre of gravity, a characteristic related to the area of an object. [12] *Contour-based* algorithms: Algorithms belonging to this group analyse the contour of the objects. Similar to the area-based approach, information about characteristics such as the centre of gravity can be gathered. However, only the contour is considered in this calculation. The results are of comparable quality to the area-based approach but achieved with less computing power. [12]

3.1.4 Classification

Classification is the central process step in object detection and can be described as the mapping of features from a continuous feature space onto a discrete class space. Examples for classification tasks are the recognition of graphic characters in a segmented image and the designation of a screw in a sorting system with different types of screws according to length and width of the screw head. According to the specific use-case, classification algorithms can be clustered into two categories: *multi-reference* classifiers and *functional* classifiers. [12] *Multi-reference classifiers* need a certain number of prototype patterns for each class. To classify an object, the calculated feature vector is compared to the prototype patterns and assigned to the class with the highest similarity. [12] *Functional classifiers* similar to multi-reference classifiers use classified training patterns. However, these prototypes are not used as patterns, but rather do the functional classifiers attempt to approximate the decision function that maps the feature vector from the feature space to the class space. [12]

3.2 Classification of deep learning methods

DL object detection methods follow a more holistic approach, in which the object detection is done from inputting an image to classifying and localising an object in one or two steps, depending on the chosen type of algorithm. Thus, a clustering of the methods as seen for the traditional methods is not of use here. Instead, the DL methods are clustered according to their functionality into One-Stage Frameworks, Two-Stage Frameworks and object detection methods that are offered by online services such as Google’s AutoML or Microsoft Azure’s Custom Vision. In order for a subject unrelated person to be able to choose, which of these three types of methods is most suitable for their use-case, the methods are assessed according to the project management triangle consisting of quality, cost and time.

3.3 The project management triangle

The aim of the developed support system is to translate the complexity of object detection methods into factors that are assessable by decision makers unrelated to the field of computer vision. The project management triangle shows, which factors characterise a project [16]. The factors represented by this triangle are quality, cost, and time [17]. Since these are characteristics by which projects are measured, they must be tangible and thus represent an adequate target system for the evaluation of deep learning methods.

3.4 Development of an evaluation model for deep learning-based object detection methods

The factors *quality*, *cost*, and *time* with regards to deep learning-based object detection methods are defined as follows:

Quality: The quality of a DL-based object detection method is represented by the mean average precision (mAP). The most common metric to assess the performance of DL-based object detection methods is average precision (AP) [3]. However, since the AP is computed for each object category separately, the mAP is adopted as a final measure of performance, as this is calculated across all object categories [3,18]. The quality spectrum is continuous, where a high mAP value suggests a high quality.

Cost: The cost it takes to implement a DL-based object detection method is derived from the labour cost of the developer and the computing cost to train the model. As these are highly variable factors, in this paper the cost is defined as binary, where a method is classified as high cost, when a developer is needed to program a model and train it, and classified as low cost, when a cloud service can be used to train a model externally.

Time: In this paper time is also defined as a binary value, differentiating between a method capable of real-time object detection and non-real-time object detection.

3.5 Method clusters and their features

Deep Learning methods are typically divided into two types of frameworks: *One-Stage-Frameworks* and *Two-Stage-Frameworks* [19]. This paper also suggests and includes a third type of framework, that does not fit into the typical groups: *Cloud Services*. The following chapter presents these three frameworks and how they are mapped onto the project management triangle.

One-Stage Frameworks or Single-Stage-Detectors (SSD) are DL-based object detection methods, that perform the object detection in one single step with one artificial neural network [20]. Historically, these SSDs lagged in performance compared to the upcoming Two-Stage-Frameworks [21]. However, since the localisation and classification are done in a single step, SSDs reach very high processed frame rates and are thus eligible for real-time applications [19]. The methods belonging to this group of frameworks are therefore suggested for use-cases, where real-time image processing is a basic condition. Examples for this framework include: RetinaNet, YOLO (v1-v3), Single-Shot Detection and Gradient Harmonising Mechanism (GHM) [21,22].

Two-Stage-Frameworks consist of two separate stages, where in the first stage, so-called candidate proposals that show all objects within the image are generated and most non-relevant locations are filtered out. In the second stage, these candidates are then classified. [21,19] The run-through of two separate ANN means that the framerate is lower than that of SSDs, making real-time processing non-viable [23]. When comparing the mAP of SSDs with that of Two-Stage-Frameworks, it can be seen, that Two-Stage Frameworks operate at a higher quality than SSDs [3]. Examples for Two-Stage-Frameworks include R-CNN, Fast R-CNN, Faster R-CNN, Grid R-CNN, and Double-Head-R-CNN [21,22].

The final option of DL-based object detection is represented by Cloud Services. Building a high-quality DL-based object detection model requires expertise. Cloud Services like Google AutoML, Google LLM, Mountain View, USA or Microsoft Azure Custom Vision, Microsoft Corporation, Redmond, USA are introduced to the market to support people with little or no ML knowledge [24]. These services do not require a developer with expertise but can be executed by using the online service that is offered as a subscription model. The omittance of a developer and its replacement by a temporal subscription to the service result in this being the most cost-effective option out of the frameworks. According to Google, their service also includes real-time image processing support [25]. However, the nature of an online service bares the drawback, that it cannot be used if sensible data is involved in training the model, as this data may be compromised. Even though these cloud services do not represent a certain ANN architecture, it is still considered here, as they represent an equally viable solution along with One- and Two-Stage-Frameworks.

3.5 Description and visualisation of the developed support system

The first step towards finding the most suitable object detection method is to determine whether a deep learning-based or a traditional approach is applicable for the specific use-case, by employing the decision tree developed at Institut für Textiltechnik (ITA) at RWTH Aachen University [26]. A classification method for the individual use-cases may be taken from this paper as well.
Once it has been determined whether a traditional method or a DL method is more suitable, the specific clusters of algorithms need to be established. The traditional approach follows the process steps of object detection, suggesting which cluster is most suitable depending on what type of object is to be detected. The decision tree, as seen in Figure 1, is set up so that the user can plug in the data from their specific use-case and will be given a suggestion for the most suitable cluster of methods.

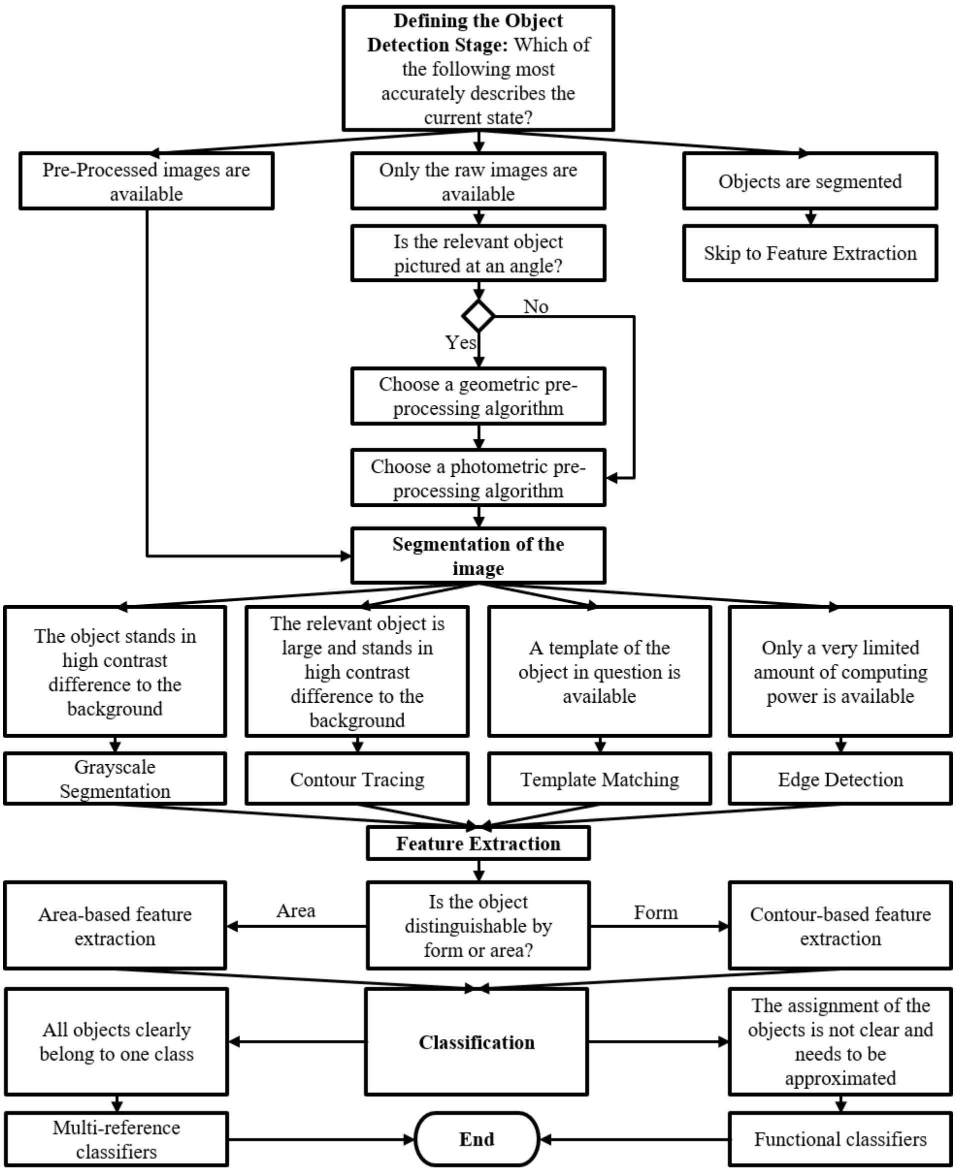

Figure 1: Decision Tree for choosing traditional object detection methods

The decision tree for deep learning-based methods is displayed in Figure 2. The main characteristics by which the cluster of methods is determined are quality, cost, and time. According to these dimensions of the project management triangle, a suitable cluster of methods is suggested for the given use-case. The complexity of the different deep learning-based object detection methods has thus been translated into more tangible factors in which the scope of a project is generally described. It must be noted that there is no concrete outcome for the combination of prioritising a low-cost solution with sensible data. This can be justified with the definition of cost within this paper, as cost is seen as a relative dimension, where the cost of manual labour exceeds that of an online service and thus a cost-effective alternative to the online service would result in a contradiction.

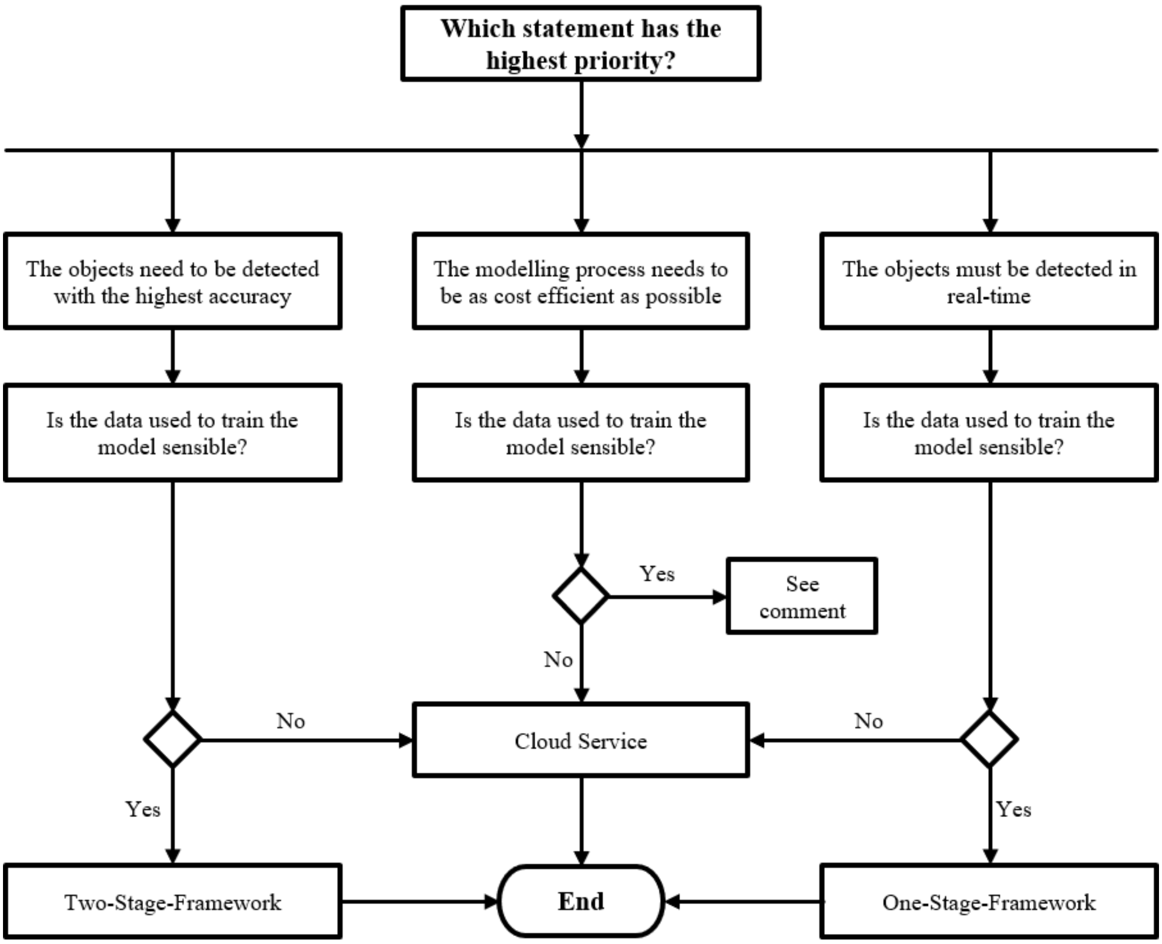

Figure 2: Decision Tree for Deep Learning Methods

4. Use Case

The proposed support system has been applied to a use-case at the Digital Capability Center (DCC) Aachen. In the use-case, the painting of car doors of an OEM was simulated. For this model there is a choice of four lacquers: red, green, blue, and yellow. After the paint has hardened, there are three different outputs: quality okay, white tarnish, and paint runners, resulting in a total of 12 classes that needed to be detected, as seen in Figure 3. The conditions set for this use-case were, that the development of the model had to be cost-efficient, and no real-time object detection was needed.

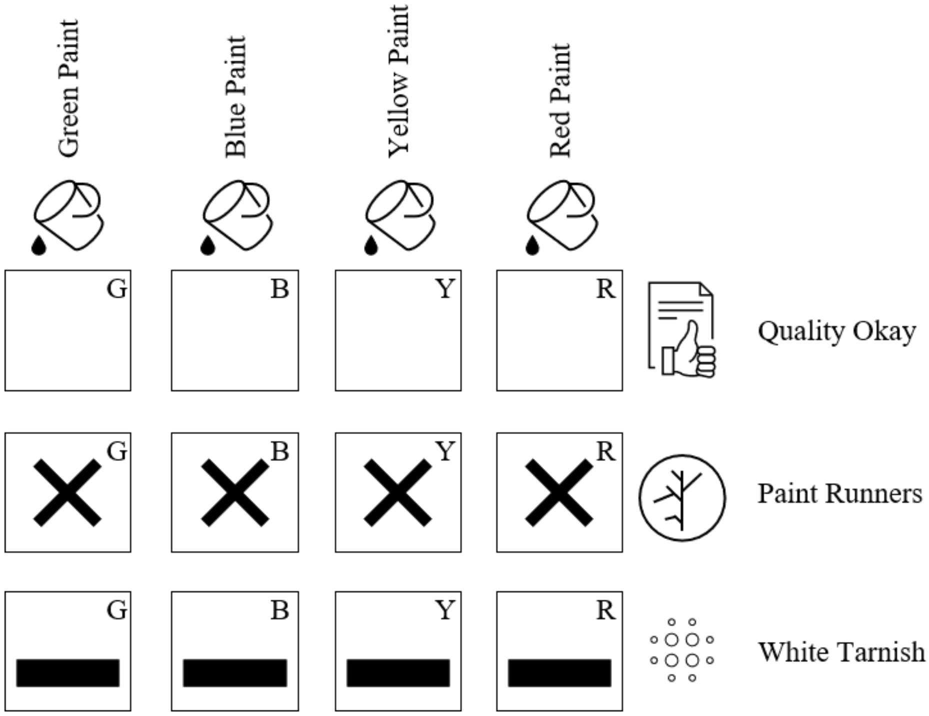

Figure 3: Class types for the DCC use-case

Since this simulation did not encompass client's data, it was not sensible. All objects were readily available at the DCC, thus as much data as needed could be gathered. The availability of high-quality images enabled a deep learning-based approach. Seeing that the data was not sensible, it was possible to use a method offered by an online service. The omittance of a developer and its replacement by a temporal subscription to the service result in this being the most cost-effective option out of the frameworks. With the given parameters of the use-case, the model was then trained with labelled data and Azure's Custom Vision Service. The results of the tested model can be seen in Table 1. To guarantee the comparability of the trained models' performance, the table shows the metrics used by Microsoft Azure, which are commonly used to assess a trained model's performance, where 100% is the optimal outcome for each value. For further information about these metrics, the reader may be referred to [27,28]. This example displays the value of the developed support system. Within a few questions and minimal effort, an object detection method has been suggested, that fits the given use-case and delivered promising results.

Table 1:Testing results for the trained model chosen by the developed decision support system for the DCC use-case

	Precision	Recall	mAP
Value	95.5%	95.5%	98.9%

5. Conclusion and discussion

The application of the decision support system on the DCC use case shows, that with a small number of steps the vast landscape of object detection methods was distilled to a solution that fit the use-case at hand. The support system was able to break down the complex possibilities to solve the problem into a few questions that could be answered by only having knowledge about the use-case and not about the underlying technology. In a next step, to further optimise the presented support system, the clusters need to be granulated further, so that the decision tree outputs not a cluster, but rather a suggestion for an exact algorithm. To optimise the fulfilment of each use-cases potential, this decision tree can be implemented into an existing, user friendly framework developed at ITA [26], that is made to suggest the optimal hardware for a given use-case, thus building an enhanced support system.

Acknowledgements

Gefördert durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen der Exzellenzstrategie des Bundes und der Länder – EXC-2023 Internet of Production – 390621612. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

[1] Myszewski, J.M., 2010. Mathematical model of the occurrence of human error in manufacturing processes. Qual. Reliab. Engng. Int. 26 (8), 845–851.

[2] Jacobs, K., Kuhlmey, A., Greß, S., Klauber, J., Schwinger, A., 2020. Mehr Personal in der Langzeitpflege - aber woher? Springer Open, Berlin, 336 pp.

[3] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., 2020. Deep Learning for Generic Object Detection: A Survey. Int J Comput Vis 128 (2), 261–318.

[4] future. OMahony - Deep Learning vs Traditional Computer Vision.

[5] Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2018. Object Detection with Deep Learning: A Review, 21 pp. http://arxiv.org/pdf/1807.05511v2.

[6] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2019. Deep Learning vs Traditional Computer Vision.

[7] Süße, H., Rodner, E., 2014. Bildverarbeitung und Objekterkennung. Springer Fachmedien Wiesbaden, Wiesbaden, 667 pp.

[8] Chollet, F., 2018. Deep learning with Python. Manning Publications Co, Shelter Island New York, xxi, 361 pages.

[9] Suzuki, K., 2011. Artificial Neural Networks: Methodological Advances and Biomedical Applications.

[10] Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Muller, K.-R., 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. Proc. IEEE 109 (3), 247–278.

[11] Serena Booth, Yilun Zhou, Ankit Shah, Julie Shah. Bayes-TrEx: A Bayesian Sampling Approach to Transparency by Example.

[12] Demant, C., Streicher-Abel, B., Springhoff, A., 2011. Industrielle Bildverarbeitung. Springer Berlin Heidelberg, Berlin, Heidelberg, 420 pp.

[13] Steger, C., Ulrich, M., Wiedemann, C. (Eds.), 2018. Machine vision algorithms and applications, 2nd, completely revised and enlarged edition ed. Wiley-VCH, Weinheim, 494 pp.

[14] Swaroop, P., 2016. An Overview of Various Template Matching Methodologies in Image Processing 2016.

[15] Sklansky, J., 1978. Image Segmentation and Feature Extraction. IEEE Trans. Syst., Man, Cybern. 8 (4), 237–247.

[16] Bohinc, T., 2010. Grundlagen des Projektmanagements: Methoden, Techniken und Tools für Projektleiter, 1. Aufl. ed. GABAL, Offenbach, 186 pp.

[17] Pollack, J., Helm, J., Adler, D., 2018. What is the Iron Triangle, and how has it changed? IJMPB 11 (2), 527–547.

[18] Nguyen, K., Huynh, N.T., Nguyen, P.C., Nguyen, K.-D., Vo, N.D., Nguyen, T.V., 2020. Detecting Objects from Space: An Evaluation of Deep-Learning Modern Approaches. Electronics 9 (4), 583.

[19] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal Loss for Dense Object Detection, 10 pp. http://arxiv.org/pdf/1708.02002v2.

[20] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single Shot MultiBox Detector 9905, 21–37.

[21] Groener, A., Chern, G., Pritt, M., 2019. A Comparison of Deep Learning Object Detection Models for Satellite Imagery, 1–10.

[22] Soviany, P., Ionescu, R.T., 2018. Optimizing the Trade-off between Single-Stage and Two-Stage Object Detectors using Image Difficulty Prediction, 6 pp. http://arxiv.org/pdf/1803.08707v3.

[23] Ale, L., Zhang, N., Li, L., 122018. Road Damage Detection Using RetinaNet, in: 2018 IEEE International Conference on Big Data (Big Data). 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA. 10.12.2018 - 13.12.2018. IEEE, pp. 5197–5200.

[24] He, X., Zhao, K., Chu, X., 2021. AutoML: A Survey of the State-of-the-Art. Knowledge-Based Systems 212 (3), 106622.

[25] Google Cloud, 2021. AutoML Tables Beginner's guide | Google Cloud. https://cloud.google.com/automl-tables/docs/beginners-guide. Accessed 13 March 2021.

[26] Müller, K., Gries, T. Beurteilungskriterien für visuelle Inspektionssysteme, in: , VDI-Z, 162(2020), pp. 59–61.

[27] Henderson, P., Ferrari, V., 2016. End-to-end training of object class detectors for mean average precision, 15 pp. http://arxiv.org/pdf/1607.03476v2.

[28] Padilla, R., Netto, S.L., Da Silva, E.A.B., 72020. A Survey on Performance Metrics for Object-Detection Algorithms, in: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil. 01/07/2020 - 03/07/2020. IEEE, pp. 237–242.

Biography

Kai Müller M.Sc. (*1992) studied Mechanical Engineering at RWTH Aachen University and started as a researcher at the Institut fuer Textiltechnik (ITA) in 2019. His research is focused on value added management in textile process chains and quality assurance.

Stephan-Andrés Saß (*1992) is a Mechanical Engineering student at RWTH Aachen University and a student researcher at McKinsey & Co.'s Digital Capability Center Aachen. His research is focused on industry solutions in the field of digital manufacturing.

Dr.-Ing. Christoph Greb (*1979) was head of the composites department and member of the Institute's board at the Institute fuer Textiltechnik at RWTH Aachen University from 2013 – 2017. Since 2018 he is Scientific Director at the Institute fuer Textiltechnik at RWTH Aachen University. Furthermore, he is member of the advisory board of the project futureTEX in the programme "Zwanzig20 – Partnerschaft fuer Innovation" of the Federal Ministry of Education and Research (BMBF) and Expert evaluator for the European Commission under the topic CE-FNR-14-2020 "Innovative textiles – reinventing fashion" of the call H2020-FNR-2020. (Horizon 2020 – Food and Natural Resources).