Designing a computer-vision-based artifact for automated quality control: a case study in the food industry

Felix Xiong[1] · **Niklas Kühl**[1,2] · **Maximilian Stauder**[3]

Abstract

Reducing waste through automated quality control (AQC) has both positive economical and ecological effects. In order to incorporate AQC in packaging, multiple quality factor types (visual, informational, etc.) of a packaged artifact need to be evaluated. Thus, this work proposes an end-to-end quality control framework evaluating multiple quality control factors of packaged artifacts (visual, informational, etc.) to enable future industrial and scientific use cases. The framework includes an AQC architecture blueprint as well as a computer vision-based model training pipeline. The framework is designed generically, and then implemented based on a real use case from the packaging industry. As an innovate approach to quality control solution development, the data-centric artificial-intelligence (DCAI) paradigm is incorporated in the framework. The implemented use case solution is finally tested on actual data. As a result, it is shown that the framework's implementation through a real industry use case works seamlessly and achieves superior results. The majority of packaged artifacts are correctly classified with rapid prediction speed. Deep-learning-based and traditional computer vision approaches are both integrated and benchmarked against each other. Through the measurement of a variety of performance metrics, valuable insights and key learnings for future adoptions of the framework are derived.

Keywords Computer vision · Quality control · DCAI · Deep learning · Packaging

✉ Niklas Kühl
  kuehl@uni-bayreuth.de; niklas.kuehl@kit.edu

Felix Xiong
  felix.xiong@alumni.kit.edu

Maximilian Stauder
  maximilian.stauder@mail.schwarz

[1] Karlsruhe Institute of Technology, 76133 Karlsruhe, Germany

[2] University of Bayreuth, 95444 Bayreuth, Germany

[3] Schwarz IT KG, 74172 Neckarsulm, Germany

1 Introduction

Food production is based on long and complex supply chains (Serdarasan 2013; Gunasekaran 1996; Haji et al 2020). The inspection of food packaging is a main production bottleneck (Nandakumar et al 2020) due to manual checks leading to inevitable human error and process inefficiencies (Kang et al 2018; Vergara-Villegas et al 2014). Following the *Theory of Constraints*, reducing or eliminating those bottlenecks strongly improves production productivity (Hoseinpour et al 2020, 2021). However, companies struggle to switch to automated quality checks (Razmjooy et al 2012), e.g. leveraging Artificial Intelligence (AI)-based technologies (Kühl et al 2022) such as deep learning-based computer vision (CV) models (Zhu et al 2021). This work proposes a CV-based quality control framework including a solution blueprint and a training pipeline focused on adaptability—underlined through a use case implementation focused on food packaging. Food packaging quality has a major impact on consumer buy-decisions (Ansari et al 2019; Popovic et al 2019), and defective packaging is a main cause for food waste (Williams et al 2012; Poyatos-Racionero et al 2018). In addition, wrong package information may lead to fines, recalls, or even health risks for customers (Thota et al 2020). However, since packaging fulfills multiple purposes (marketing, protection, and information display) (Ansari et al 2019), it needs to be assessed on multiple quality factor types (visual flawlessness, correct information, etc.).

Hence, beside it's focus on adaptability, the framework also tackles the lack of quality control of multiple quality factor types in comparable works (Zhu et al 2021). Developing a (deep learning-based) CV quality control system evaluating surface as well as informational quality features of packaged artifacts has not been conducted on a scientific basis before. Without this ability, CV-based quality control systems cannot reflect, and therefore replace, thorough human quality inspection yet. In order to tackle the complexity of capturing multiple quality aspects, this work leverages the concept of *Data-Centric AI (DCAI)* (Jakubik et al 2024) to focus on data acquisition instead of model optimisation during development. So far, there have not been any DCAI focused approaches to the problem of multi-feature-based quality control yet. In a scenario with multiple regions of interest, a DCAI-focused approach allows to test different angles and recording parameters such as resolution, lightning and angle. Underlining the adaptability focus of this framework, these learnings can then be transferred into the deployment setup.

Overall, this work aims at answering the following research question:

How can an efficient multi-feature, end-to-end computer vision-based quality control system be designed?

Instead of just describing a problem-specific solution like other predecessors in the scientific field of CV-based quality control, this work proposes an end-to-end holistic architecture concept, the *quality control framework*, to be transferred onto other use cases. In this, end-to-end refers to the solution covering the whole process from image acquisition to classification result and post-processing of prediction data. The framework focuses on modularity and allows for the measurement of visual as well as informational quality features. It includes the blueprint of a potential solution architecture design as well as an exemplary training pipeline, describing

the acquisition of data and the subsequent DCAI-based data engineering. In order to prove the whole solution's viability, an implementation of the proposed solution is developed based on a use case in the field of coffee packaging.

In the following, related work in the area of quality control in packaging with regard to multi-feature classification, architectural concept and data focus is presented (Sect. 2). To further explain the approach of this work of answering the above-stated research question, the methodology is discussed in more detail (Sect. 3). From there, the framework's architecture design blueprint and the training pipeline for the integration of deep learning models into the solution architecture is outlined (Sect. 4). Afterwards, an exemplary use case including the respective data (Sect. 5), as well as the actual use case-based implementation are described to prove the viability of the solution design (Sect. 6). This is followed by the test results of the implementation, an analysis of the results, and the evaluation of alternative design approaches (Sect. 7). In the end, derived learnings, contributions, and potential future extensions are summarized (Sect. 8).

2 Related work

Despite packaging being a crucial production step, scientific works researching the application of CV-based quality control for packaging are limited. A collection of related works is presented and compared in this chapter. Generally, CV-based quality control approaches examine extractable visual information of images such as pixel intensities. These can either be analyzed by leveraging traditional CV methods, e.g. edge detection algorithms, or by training deep neural networks (Mahony et al 2020). For the latter, the detection and definition of image features relies predominantly on Convolutional Neural Networks (CNNs) (Biswas et al 2018).

For packaging, both approaches are leveraged in scientific works. Depending on the use case, traditional CV-based algorithms work very well, e.g., through color normalization to count the amount of canules in a package (Erwanto et al 2017), or by analyzing pixel histograms to compare images in order to detect outliers (Sa et al 2020). Still, traditional CV-based algorithms have their limitations in terms of flexibility and feature complexity (Mahony et al 2020). Hence, applying CNN-based architectures became increasingly popular for quality control use cases (Voulodimos et al 2018). In the packaging industry, application cases include pattern recognition (Sa et al 2020), area segmentation (Ribeiro et al 2018), and optical character recognition (Thota et al 2020). CV tasks can be accomplished on different wavelengths of light, called spectrums. Quality assessment can be done in the visible spectrum but also in the spectrum of infrared or X-ray with either a mono spectrum—i.e. grayscale images—or a multi-spectral approach. Recently, also hyperspectral imaging sensors are tested on quality control tasks (Medus et al 2021). Other approaches include the usage of out-of-the-box, proprietary software tools which come with various downsides, especially from a customization perspective (Huaiyuan et al 2013). However, all these approaches are fixed to their domain and not extendable by potential adopting users. On top of that, the

solutions developed in these approaches exclusively detect surface errors based on single quality features.

More advanced approaches conduct automated quality control based on not one, but multiple features. Nandi et al (2014) and Blasco et al (2009a) classifiy fruits based on shape, surface defects, and maturity defects using a weighted score aggregation. Alternatively to that, Blasco et al (2009b) shift away from extracting multiple features from one image, and instead use a multi-spectral approach analyzing fluorescence, near-infrared (NIR), and RGB images of citrus fruits respectively. In the packaging domain, Banús et al (2021) look at the different surface inconsistencies of thermoforming food packages and classify the packages according to different regions of interest (ROIs), using three cameras to analyze the packages from different camera angles with respect to the individual ROIs. Benouis et al (2020) scan food trays using object detection algorithms to detect 11 different classes of foreign materials. Another example is Wang et al (2012)'s approach in classifying cheese packages based on their deformation, as well as on potential cheese leakage. However, even if multiple errors are checked, all approaches are solely focused on multiple errors of similar types, e.g. based on visual appearance. Although there are attempts to include contextual information, e.g. by leveraging optical character recognition (OCR) to identify and extract expiry dates of packaged food (Ribeiro et al 2018; Thota et al 2020), these attempts do not take visual appearance factors into account.

Another distinctive factor throughout scientific works in this domain is the acquisition of data and—closely tied to that—the hardware used. Some approaches record their own data either through static images (Erwanto et al 2017), or by recording videos leveraging a conveyor belt. While requiring additional initial effort, the data acquired using video cameras in combination with conveyor belts depicts production scenarios more accurately. Data extraction techniques are either continuously filming one area and extracting frames (Banús et al 2021), or by using line scanning technologies (Benouis et al 2020). Alternatively, developers may reuse already existing data. This can be derived by leveraging publicly available data sets (Thota et al 2020), or by acquiring proprietary data (Ribeiro et al 2018).

While all previously mentioned approaches present high performance scores, most of them are tested on data similar to the development data. Some researchers attempt to include flexibility in their solutions to provide for changing conditions by increasing the variety of packaged artifact shapes and types (Ribeiro et al 2018; Benouis et al 2020), or by using different recording parameters for the test set Banús et al (2021). Thota et al (2020) propose a solution that allows including additional datasets in the context of expiry-date detection of food packages—although without explicitly describing how to integrate the additional data.

In order to underline the generalizability of the developed solutions, describing the developed software architectures is common in most of the mentioned works (Banús et al 2021; Thota et al 2020; Ribeiro et al 2018; Benouis et al 2020; Banus et al 2021). However, the focus lies on the description of use case-specific applied solutions rather than on an architectural blueprint. Also, the development process and data processing steps are only described on a very high level. In addition, some works list the integration of additional quality control metrics in

their outlook but do not describe how to extend the respective evaluation systems. Thus, the integration and adaption of the proposed solutions onto new use cases is hardly possible.

The above mentioned works and their different approaches to CV-based quality control of packaged artifacts are listed in the table below (Table 1). In the next chapter, this work's approach to fill the existent research gap of multi-feature quality control through leveraging the DCAI paradigm is explained.

Table 1 Related packaging focused CV-based quality control approaches

	Sa[a]	Banus[b]	Thota[c]	Ribeiro[d]	Erwanto[e]	Wang[f]	Benouis[g]	Banus B[h]	This work
Data Acquisition		x			x	x		x	x
Product Varience			x	x				x	x
Conveyor Belt	x	x			x			x	x
DCAI[i]									x
Architecture		x	x	x			x	x	x
E2E[j]		x	x	x				x	x
Modularity			x					x	x
Surface-based	x	x			x	x	x	x	x
Information-based			x	x					x
Multiple Cameras		x							x
Runtime Measuring	x						x	x	x
Deep Learning	x	x	x	x			x	x	x
SIFT[k]									x
OCR/OCV[l]			x	x					x

[a]Sa et al (2020)

[b]Banús et al (2021)

[c]Thota et al (2020)

[d]Ribeiro et al (2018)

[e]Erwanto et al (2017)

[f] ang et al (2012)

[g]Benouis et al (2020)

[h]Banus et al (2021)

[i]Data-Centric AI

[j]End-to-End - The whole classification process is displayed from data acquisition, processing and classification to the finalized result with respective output

[k]Scale Invariant Feature Transform
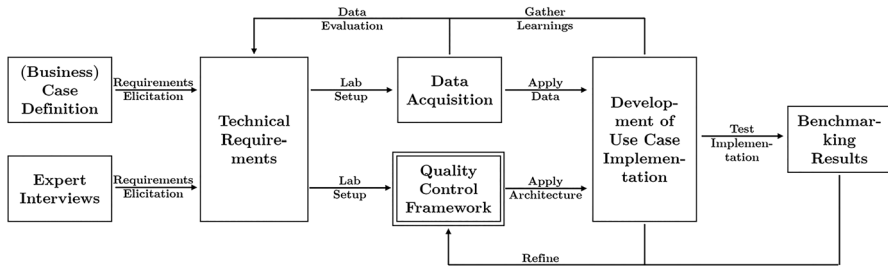
[l]Optical Character Recognition/Verification

Fig. 1 The methodology of this work with the goal of deriving the quality control framework by developing it in DCAI-focused feedback cycles

3 Methodology

As previously mentioned, conveyor belt and assembly line production processes are homogeneous in their nature. Thus, the framework of this work is applicable to all industries leveraging conveyor belts in their processes. Generally, artifacts are quality checked on a variety of quality factor categories. Therefore, the framework focuses on flexibility in terms of seamlessly adding or removing quality factors and their respective classification logic. In this, the framework must not be limited to one quality factor category. Instead, it should be feasible to perform quality checks on multiple error categories such as visual or informational errors. In order to be able to record their own training data, the authors of this work have collaborated with a big European food producer. This allows evaluating the viability of this framework by applying it onto a use case of the food producer—following the DCAI paradigm. The data is initially recorded with pre-defined parameters in terms of recording tools, camera settings, and facilitating environment. Throughout the development process, this data as well as the framework is then continuously re-evaluated. If development bottlenecks occur due to insufficient training data, parameters are adjusted and new data acquired. For a high-level overview of the development process of both the framework as well as the following use case implementation see Fig. 1.

3.1 Data acquisition

The data acquisition is conducted by setting up a lab environment simulating real use cases. The lab environment consists of a conveyor belt and multiple cameras recording the artifacts on the belt from multiple angles. The cameras have different recording parameters, e.g., resolution or sensor. As output, the cameras record video material, which is split into single frames. These frames are sorted, segmented, and partially labelled, so that they can be used as training input. Videos are thus recorded with respect to different regions of interest (ROIs) of the artifacts. These are determined by the respective quality control types and their position on the package. The underlying use case is focused on the quality classification of coffee packages (Sect. 5). These have multiple ROIs with respect to multiple quality factors—the lot number, the expiry date, the barcode, and the sticker on top of the package. Thus,

videos are recorded with focus on these ROIs. Each ROI has different requirements as they vary strongly in classification logic. The lot number and expiry date have to be read out and evaluated logically. As a result, recordings of these ROIs require a high resolution to enable the extraction algorithms to work properly. The top sticker on the other hand merely needs to be identified, so it does not require as strong of a camera focus as the other ROIs. Following the DCAI paradigm allows trying different recording settings and testing the acquired data during development for the respective quality factor classification—and potentially re-recording data with new settings.

3.2 Quality control framework

Concurrently with the data acquisition, a framework for the development of an automated CV-based quality control system is designed. The framework includes a proposal for a solution architecture following a pipeline shape in order to mirror the linear process of manual quality control. In addition, it describes a training process to train the leveraged deep learning models in this architecture. The solution architecture is designed as a collection of microservices (Dmitry and Manfred 2014) with a facilitating process logic to connect them. This allows easy adoption and tailoring for other use cases. The training pipeline is designed to automate the training process, making it scalable and minimizing the need for manual labeling. In enables users to obtain and integrate sufficient training data with little manual work. Through that, it is enabling the DCAI-focused development. Overall, the goal of this framework is to support the development and implementation of an automated quality control system by providing a blueprint for a solution architecture and a streamlined training process.

3.3 Use case implementation

The overall framework is not only described theoretically, but also implemented based on the use case described in Sect. 5. In the use case, various computer vision techniques are compared, and the finalized solution trained and tested on the use-case-related, self-acquired data. Based on input of subject matter experts, the solution's requirements (e.g., detection speed, most common error types,...) are elicited and continuously adjusted throughout the development process. During development, the proposed solution architecture blueprint is used as foundation for the developed solution. This not only includes the classification solution itself, but also data acquisition, model training, and testing. The training pipeline of the framework is leveraged during the model training phase as it enables following the DCAI paradigm without the task of manual relabelling. By testing the example solution after development and training, performance results of the solution are obtained and evaluated. In order to identify performance factors of the proposed solution, additional alternative implementation choices, both deep learning model as well as traditional CV approachesare developed and compared against each other.

3.4 Benchmarking

Common CV metrics are used as benchmarking metrics for a comparison between the alternative approaches. The accuracy of the object detection of the packaged artifact as well as the classification ROIs is measured by using the *Intersection over Union (IoU)* (Rezatofighi et al 2019). This metric essentially calculates the percentage of overlap between both bounding boxes—the predicted and the actual one.

To measure classification accuracy, confusion matrices (Ting 2017) are the foundation for more complex metrics. Thus, the confusion matrices for the overall and ROI classifications are calculated during the tests. From there, accuracy, precision and recall (Vakili et al 2020) can be derived.

Based on confusion matrices and IoU, the most common metric for object detection and classification is the *Mean Average Precision (mAP)* (Henderson and Ferrari 2016). This metric calculates the mean of all interpolated average precisions (APs) per recall values for each class with respect to a certain IoU threshold. Here, interpolated precisions are the local maxima of the precisions per recall.

The mAP is calculated for the packaged artifacts and the classification ROIs extraction. Additionally, the average classification time (t) for the overall pipeline is measured to also compare the alternative approaches with the current classification speed.

The results of benchmarking are then analyzed, discussed and further potential improvements are proposed. The overall goal of the benchmarking process is to underline the viability of the developed framework and its implementation. In addition, insights regarding edge cases and potential problem sources can be identified. These learnings are then used to re-evaluate and propose future improvements to the overall framework and the developed solution. In the next chapter, the generalized, CV-based quality control framework is described in more detail.

4 Quality control framework

The quality control framework consists of two parts—a potential solution architecture design concept and a training process logic. The latter enables the solution's development to follow the DCAI paradigm with large amounts of data. The solution architecture design is strongly focused on modularity. It allows users to apply their use cases seamlessly. It is hence to be understood as a blueprint which can modified for use case-specific adjustments. The training pipeline enables users to integrate object detection models in this architecture concept. The models are used for the package and the ROI detection during the quality control process. In the following, the architecture concept and its services are presented. Afterwards, the training pipeline is described.

4.1 Architecture

The solution architecture design follows a pipeline shape due to the linear process of quality control checks. With focus on modularity, it follows a service oriented architecture concept (Fig. 2). This means the individual steps of the pipeline are segmented into single microservices (Perrey and Lycett 2003). Each microservice has its own purpose and can be seen as an independent development entity block. The independence of the blocks allows use-case specific modifications. This can be derived both on service internal level, as well as the addition or removal of existing services. Only the input and output requirements of the existing services needs to be considered.

Acquistion service. When the solution is ran, the *acquisition service* first sets all camera parameters according to pre-defined values. From there, the *acquisition service* iteratively pulls frames from the camera on a pre-defined time interval. Each frame is then checked on the appearance of the artifact, and wether all ROIs are detectable. If not, the next frame is pulled. If an artifact and it's ROIs are detectable, the acquired raw frame is converted to the required image format and sent to the *processing service*.

Processing service. The *processing service* is responsible for the pre-processing of the image, preparing it for the succeeding *classification service*. Its task is to extract the classification ROIs from the raw frame. As a first step, the raw frame is pre-processed in multiple steps including pixel format transformation, size adjustments, or noise reduction among others. Then, the packaged item is identified, extracted, and labeled. All these steps are then repeated to extract the ROIs from the cut out frame. These are forwarded as input parameters to the *classification service*.

Classification service. To perform the quality control of all factors, multiple evaluations based on the extracted ROIs are performed in the *classification service*. Each evaluation is performed individually in parallel, so quality factors can be easily added or removed. Thus, each quality factor also requires its own classification logic. The applied logic depends on the detectable error types (e.g. textual syntax). All individual classification model output scores are aggregated based on a use-case
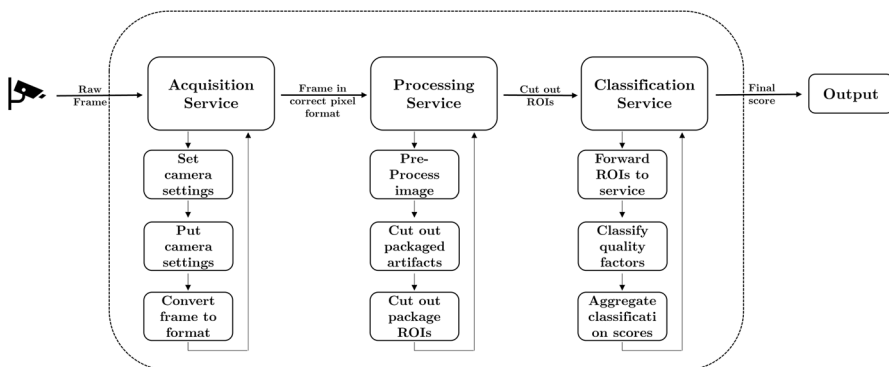

Fig. 2 The quality control pipeline solution architecture

dependent aggregation logic. Examples of aggregation logic are Boolean-like logic, weighted sums, or an average over all classification scores. Through the aggregation logic, various error types of different error categories can be jointly evaluated.

Output relay service. In the last step, the use case requirements decide which output channels to use. It is possible to simply store classification data in raw form, to calculate metrics and store the results in databases, or to have follow up actions defined, e.g., a flashing LED or the interruption of the packaging process.

After describing the architecture on a high level, the proposed solution design architecture is implemented based upon an industry use case in Sect. 6.

4.2 Multi-source approach

The above described architecture classifies a package based on a single frame. This might not always be feasible. To be able to capture all ROIs of an artifact's package—even if they are positioned on opposite sides of the package—the framework allows classification based on multiple camera sources. The classification logic fundamentally stays the same. Solely the acquisition and processing of multiple frames is conducted differently through multiple, parallel processes Fig. 3. Profiting from the framework's modularity, the *acquistion* and *processing service* can be tailored for each camera source individually. The architecture enables users to seamlessly add or remove sources.

4.3 Training pipeline

To enable data-centric development, iteratively recorded training data for models used in the *classification* and *processing service* can be seamlessly integrated. The presented training pipeline focuses on deep learning-based object detection models leveraged in the *processing service* (Fig. 4). Focusing on flexibility, the architecture allows users to test and integrate different object detection models based on on their respective requirements.
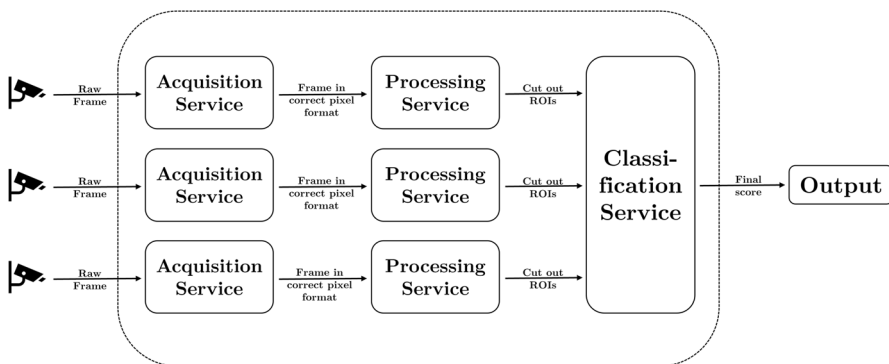

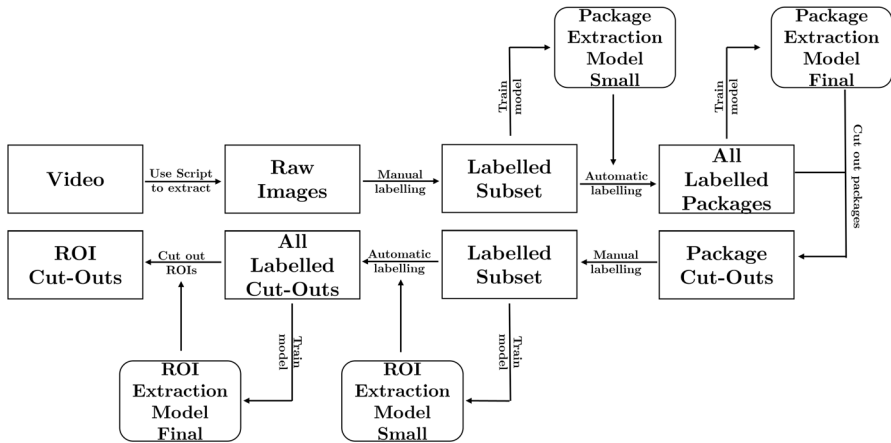**Fig. 3** The multi-source approach of the solution architecture

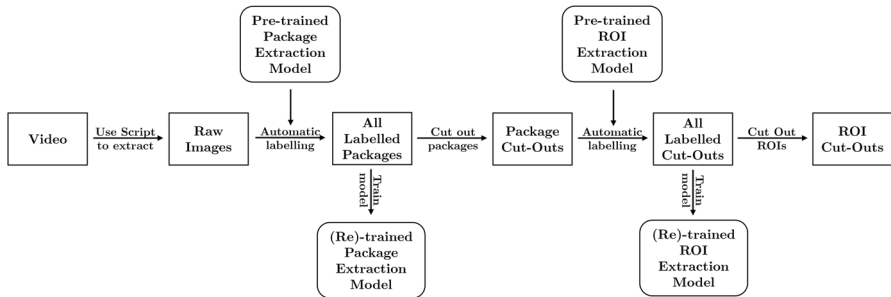**Fig. 4** The framework's training pipeline for semi-automatic labeling

Fig. 5 Process of re-using the pre-trained models for automatic labeling

Recorded video data is cut into frames by a script, and stored as training and test data. A subset of frames per sorting category (artifact types, fraudulent or flawless, input source, etc.) needs to be manually labelled. After training an object detection model with this subset, the remaining unlabeled images are automatically labelled as well by leveraging this pre-trained helper model. Through that, the labeling process is semi-automated. The same process can now be repeated for the ROI extraction model. Following this training scheme, users can operationalize huge amounts of training data.

Ultimately, the training pipeline enables developers to follow the DCAI paradigm. Existing models are reused for the automated labeling of new training data in scale. By making the object detection models as robust as possible, they can be reused in every new iteration of data acquisition as displayed in Fig. 5. The figure shows the process of automated labeling in the case of newly captured training data. The video data is segmented into images which are then labelled by the previously trained models. This labeled data is then re-used to further solidify the robustness

of these models through retraining—allowing even more precise automated labeling for future data acquisition iterations.

5 Case description

Quality control in packaging processes are generally very similar. Unpackaged artifacts enter the process, packaging applied, and the evaluation result then monitored (Poyatos-Racionero et al 2018). Hence, this work considers an example use case to be very expressive for the general applicability of the proposed framework. In this specific use case, the framework is tested on the automated quality control of coffee packages. The coffee packages contain 500 g vacuumed coffee each and have a rectangular shape. The vacuum packaging is aluminum-based with a paper cover wrapped around. The cover is held together through a paper sticker glued on package's top side. The use case evaluates the packaging of five different types of coffee beans. Each type has a different design in terms of paper cover and top sticker. However, they are all similar in shape. An illustration of the coffee packages can be found in Fig. 6.

First, a list of relevant errors of coffee packages is created and discussed with experts working at production site. The most important errors in terms of error frequency and impact are identified:

In terms of visual features, the top sticker needs to be evaluated. The top sticker being incorrect would make the package unsaleable in stores. Regarding informational features, the expiry date needs to be accurate due to potential implications of faulty dates (Zielińska et al 2020). Also, the lot code as well as the barcode are controlled since they are crucial elements of the packages logistic processes. An overview of potential errors and their categories can be found in Table 2 (the errors checked in this work in bold). Coffee packages may be classified as bad because of a single, (Fig. 9) or multiple (Figs. 7 and 8) fraudulent quality factors. The three
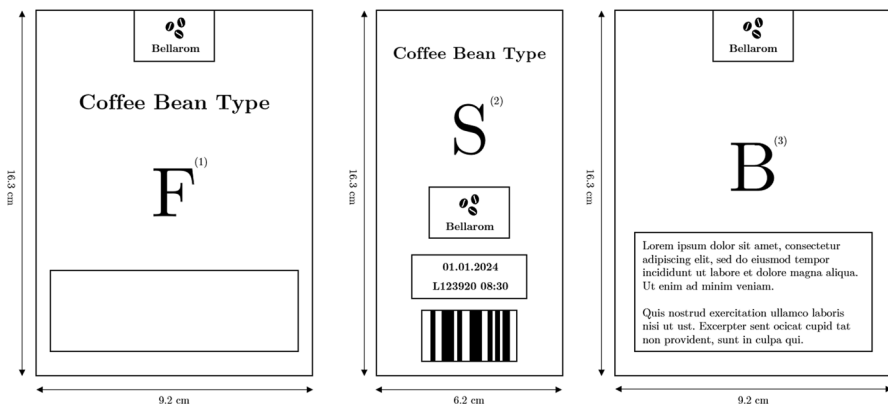

Fig. 6 The coffee packages are filmed from the side (2). Depending on their position on the belt, the front (1) and back (3) are visible as well. However, the ROIs are all positioned on the side (and top) view

resulting ROIs are shown in Fig. 6. Each ROI and its possible errors are listed in Table 3.

In order to capture all error types, different requirements are necessary. To record training and test data, a laboratory setup including conveyer belt and multiple cameras is constructed. This allows the re-recording of coffee packages with respect to different camera types, angles, and parameters.

Video data is recorded with two different cameras from horizontal (Fig. 7), diagonal, i.e. 30 and 60 degrees to the horizontal axis (Fig. 8), and vertical (Fig. 9) angles.

Table 2 Potential error types with their respective category (error types checked in thus work displayed as bold)

Quality factor	Quality factor category
Top Sticker Missing	**Visual**
Top Sticker position[a]	Visual
Top Sticker rotation[a]	Visual
Expiry-by-date not readable	**Visual**
LOT not readable	**Visual**
Barcode not readable	**Visual**
Paper cover incorrectly folded[a]	Visual
Outside damage[a]	Visual
Back seam of package crooked[a]	Informational
Expiry-by-date semantically incorrect	**Informational**
Wrong Paper cover language[a]	Informational
Barcode semantically incorrect	**Informational**

[a]Errors which occur but are not covered in this work

Table 3 Checked potential errors per quality factor

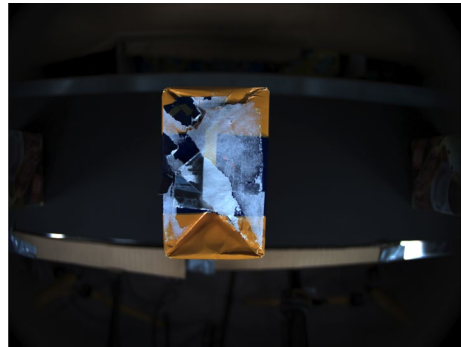| Expiry date	Lot number	Barcode	Top sticker
Not readable	Not readable	Not readable	Missing
Wrong format	Wrong format	Wrong Barcode	Wrong Top Sticker
In the past	Wrong Lot		
Below threshold			

Fig. 7 Expiry date, lot number and barcode defect

Fig. 8 Expiry date and lot number wrong

Fig. 9 Top sticker missing

Videos are iteratively recorded with different cameras, processing hardware, camera settings, and light adjustments.

In total, 1.7 Terabyte of video data is recorded in *avi* format. In every iteration, packages of each coffee type are recorded separately without any modifications, flipped vertically and horizontally to extend the training base, mixed together for test data, and a sample of packages per type is recorded with simulated quality flaws on the packages. The videos are split into single frames and semi-automatically labeled leveraging the framework's training pipeline concept. It hence follows the concept of *semi-supervised deep learning*—i.e. using a small set of labeled data and a large set of unlabeled data (Zhu and Goldberg 2009)— for the labeling process. This concept is widely applied in recent years targeting efficient training processes with large amounts of data (van Engelen and Hoos 2020). In the first iteration, videos of the packages on the conveyer belt are recorded from four different angles. In the second iteration, the frames per second are reduced and the resolution is increased. In the last iteration of recordings, the illumination is adjusted to particularly improve the top sticker classification. Therefore, videos are only recorded from a 60 and a 90 degrees angles. All numbers regarding data acquisition can be observed in Table 4. Here, videos are being recorded in different camera angles (*Camera Degrees*), and with approximately the same amount of flawless packages (*Non-Error Images*) of each of the five coffee package types (*N-E Images per Type*). Overall (*Total Images*),
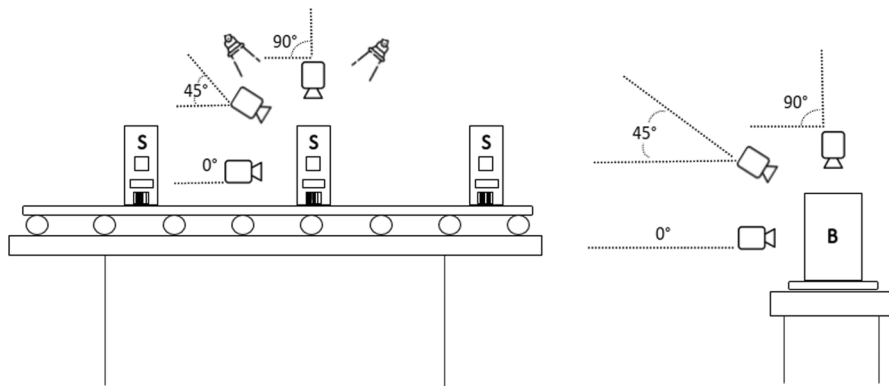
Fig. 10 Data Acquisition Setup from side (left) and front (right) view of the conveyor belt

Table 4 Acquired images extracted from recorded videos

	1st Iteration	2nd Iteration	3rd Iteration
Camera degrees	4	3	2
Non-Error (N-E) images	≈ #5400/deg	≈ #3500/deg	≈ #1200/deg
N-E Images per type	≈ #1100/deg	≈ #700/deg	≈ #600/deg
Error Images	≈ #3500/deg	≈ #3600/deg	≈ #600/deg
Total Images	≈ #35, 600	≈ #21, 300	≈ #3600

around 80,000 images of flawless as well as deliberately damaged (*Error Images*) packages are extracted from the video recordings.

6 Implementation

Leveraging the acquired data, the quality control framework proposed in Sect. 4 is used as foundation for an exemplary use case implementation. The applied quality control pipeline is developed using the PyTorch (Paszke et al 2019) framework with use case specific design decisions for each of the individual services.

Processing service As the first prototypical solution, a *YOLOv5s* (Jocher et al 2022) model is used in order to detect and classify the coffee packages as well as the ROIs. Models from the *YOLO* family (Redmon et al 2016) are single stage detectors, hence faster compared to two-stage detectors. In addition to that, Meta *Detection Transformer (DE:TR)* (Carion et al 2020) as a transformer-based approach, and a traditional CV algorithm in *scale invariant feature transform (SIFT)* (Lowe 1999) are used to compare between different object detection algorithmsto compare deep learning-based and traditional CV approaches at the coffee package and ROI detection step (Fig. 10).

Table 5 Technologies used in the use case implementation of the quality control framework

Coffee extraction	ROI extraction	Barcode class	OCR class	Top sticker class
YOLOv5	YOLOv5	Pyzbar	Pytesseract	SIFT
DE:TR	DE:TR		YOLOv5	YOLOv5
SIFT	SIFT			CED

Classification models For the purpose of this use case, no manual training of the classification models is required. Instead, pre-defined algorithms and pre-trained models are leveraged.

The *Expiry Date and Lot Number Classification Model* requires syntactical and semantical checks based on optical character recognition (OCR). For this work, Google's pre-trained Tesseract engine (Smith 2007) for image character recognition is leveraged. To enable the OCR process, the frame is put through multiple pre-processing steps, e.g., grayscaling, Gaussian blurring (Gedraite and Hadad 2011), binarization (Palumbo et al 1986), dilating (Soille 2004), binning, (Jin and Hirakawa 2012) and smoothing (Lee 1983). The process is visualized in Fig. 11. The extracted information is both checked syntactically and semantically.

In the case of the *Barcode Classification Model*, we are able to use existing models by including the Python distribution of *zBar* (Sourceforge 2011) called *pyzbar*. Pyzbar enabbles the implementation of a pre-trained CNN focused on decoding barcodes. As the barcode needs to be horizontal, multiple pre-processing steps including Hough's line transformation (Illingworth and Kittler 1988) are applied. The barcode content is then checked on readability and information.

To verify the existence and correctness in the *Top Sticker Classfication Model*, the SIFT algorithm is used. It extracts keypoints (Fig. 12) of an image, compares it with the keypoints of a reference image, and counts the matches. In this case, extracted package frames are compared with standalone cutouts of each type's top sticker text. By matching the cutout and the text, top sticker existence and coffee type are determined using a certain threshold of keypoints. If enough keypoints

(1)   (2)   (3)   (4)

Fig. 11 OCR extraction process for expiry date and lot number classification from the extracted ROI (1), over pre-processing incl. binarization (2), resizing and dilation (3) to the contour extraction and separation of the individual text lines (4)
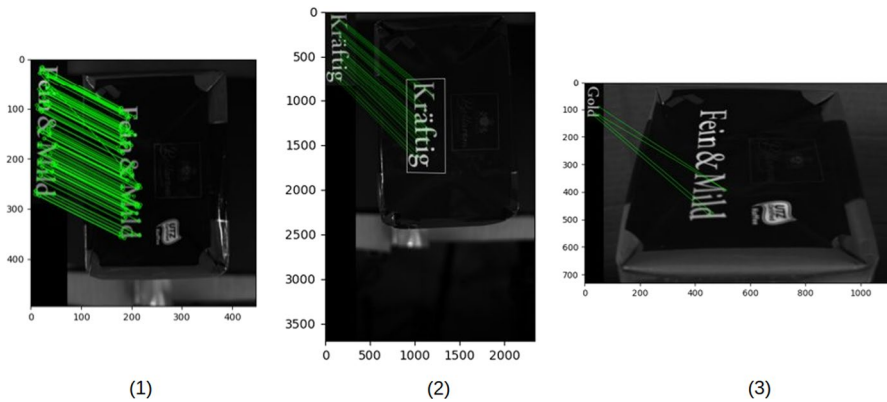
Fig. 12 SIFT-based top sticker classification showing how many matches a long coffee bean type name (1) has in comparison to a short coffee bean type name (2). In addition, a wrong top sticker classification based on too few matches is shown (3). Note: The coffee bean type *Fein & Mild* is referred in this work as *Light* and *Kräftig* as *Strong*

are detected, the top sticker is classified as being flawless, and it is tested if the detected top sticker matches the rest of the package.

An overview of technologies and approaches included in the implementation can be found in Table 5.

For the *Aggregation and Output step*, the use case demands a Boolean-like classification. This means, if one of the quality factors is classified as insufficient, the whole package is classified as faulty. The outputs are stored in a local database, including information about which quality factor is responsible for the negative classifications. On top of that, performance metrics are continuously calculated and stored in the database as well.

For the models used in the *processing service*, the proposed training pipeline is leveraged. During training, emphasis is being put on the variety of training data with regard to position, recording angle, and lightning among other factors, to reach robust models. Continuously feeding more images into the training process during the DCAI-based re-recordings of training data further improves the models robustness. For every new iteration of data acquisition, the new images are labeled using the already-trained models. Through that, little to no manual label efforts are necessary with the labeling process quasi-automated. Consequently, even if new recordings with new parameters would be required, labeling them, training the models, and testing the pipeline with the new data foundation is easily possible. This streamlines the development process significantly and shows the advantages of DCAI.

7 Evaluation

For the quantification of results, the performance of the framework's implementation is evaluated. To test the processing service, the YOLOv5s, the DE:TR, and the SIFT object detection algorithm are ran against each other on a subset of data under equal

Fig. 13 Confusion matrix of YOLO + YOLO model combination

	Actually Positive	Actually Negative
Predicted Positive	30 (0.60)	2 (0.04)
Predicted Negative	0 (0.00)	18 (0.36)

Fig. 14 Confusion matrix of SIFT + SIFT model combination

	Actually Positive	Actually Negative
Predicted Positive	27 (0.54)	0 (0.00)
Predicted Negative	3 (0.06)	20 (0.40)

Fig. 15 Confusion matrix of DE:TR + DE:TR model combination

	Actually Positive	Actually Negative
Predicted Positive	30 (0.60)	3 (0.06)
Predicted Negative	0 (0.00)	17 (0.34)

setup conditions. The subset consists of 50 image pairs from different angles, 30 of them being flawless packages. Regarding the other 20, certain quality factors are fraudulent—either one, multiple, or all of them. For the overall classification performance, common prediction metrics are used to test different algorithm combinations.

7.1 Results

Exemplary confusion matrices (Figs. 13, 14, 15) of all three object detection approaches show that the solution framework classifies the 50 image pairs mostly correct. However, false positive and false negative classifications occur as well. Possible explanations for these are discussed in Sect. 7.2.

During benchmarking, the *Intersection over Union (IoU)* is calculated for the coffee package and ROI detection respectively. Regarding coffee package detection, the SIFT algorithm outperforms the deep learning-based CV algorithms for both degrees. The YOLO and the DE:TR approach derive very similar scores.

The same phenomena can be observed when calculating the *mean average precision (mAP)* based on an IoU of 0.5 for the coffee package detection (Table 6). Again, the YOLO and DE:TR models achieve very similar results. However, this time they outperform the SIFT algorithm on both camera angles. Especially for the 90 degrees camera angle, their respective mAPs are significantly higher than the SIFT's mAP. Generally, some package designs are detected with better mAP scores than others. For the 0 degree camera angle, *Decaf* and *Light* have lower mAP scores than the other three types for all object detection approaches. Also, the two coffee been types with golden packaging—*Biogold* and *Gold*—achieve lower mAP scores compared to the other three types from the 90 degrees camera angle. Another takeaway from the data is that the 0 degree camera angle is very accurate for all three models, while for 90 degrees scores are lower.

The overall solution classifies the coffee packages very fast (Table 7). The quickest classifications are achieved when the package detector is a neural network, either DE:TR or YOLO, combined with the SIFT algorithm as the ROI extractor. However, all model combinations classify within less than 0.6 s per package in average. Next, the findings are analyzed, interpreted and discussed.

Table 6 Mean average precision for package extraction with an IoU of 0.5 (mean Average Precision of individual package types in bold)

Average Precision	DETR 0° (%)	SIFT 0° (%)	YOLO 0° (%)	DETR 90° (%)	SIFT 90°(%)	YOLO 90° (%)
Biogold	100	100.00	100	76.56	65.00	68.06
Decaf	91.67	91.67	91.67	98.18	90.00	86.00
Light	93.75	81.54	96.67	90.00	73.64	86.00
Gold	100	100.00	100	77.78	39.68	77.78
Strong	100	91.67	100	92.31	76.92	84.62
mAP	**97.08**	**92.97**	**97.67**	**86.97**	**69.05**	**80.49**

Model combination	Overall classification
($Package_$ROIs)	Time
DETR_DETR	0.5788s
DETR_SIFT	0.3961s
DETR_YOLO	0.4485s
SIFT_DETR	0.5385s
SIFT_SIFT	0.5549s
SIFT_YOLO	0.5078s
YOLO_DETR	0.5739s
YOLO_SIFT	0.3655s
YOLO_YOLO	0.5960s

Table 7 Average classification time per model combination

7.2 Discussion

In this work, the use case implementation including the DCAI-focused approach of semi-automated object detection model training shows the general adaptability of the proposed framework. The overall performance of the continuously re-trained object detection models can be extracted from the derived mAP values for each model combination. Also, with regard to the derived confusion matrices, it is shown that most coffee packages are correctly classified as positive or negative. Furthermore, the overall prediction time is close to matching industrial conveyer belt speeds according to experts at this work's industry partners' production site. This is particularly interesting considering the limited amount of actions taken to increase the classification speed. However, while showing the feasibility of the implemented solution and thus of the framework, further improvements and learnings based on the benchmarking results are discussed in the following.

As shown by the confusion matrices (Figs. 13, 14 and 15), the prototypical implementation classifies most coffee packages correctly. However, there are still false positive (FP) and false negative (FN) classified coffee packages. Some of these errors are very hard to erase. E.g., the expiry date of a sample package (Fig. 16) looks fraudulent to the human eye. However, the OCR algorithm classifies the ripped part as the digit *four* due to its shape and the backside of the paper having the same color as the font—the result is a false positive classification. Another example is the top sticker model, which classifies the correctness based on the amount of related key points. However, even if the top sticker is damaged, the algorithm still detects key points and might classify it as positive, as can be seen in Fig. 17. But not only the FPs, also the FNs are often based on problems during OCR extraction. Especially the digits *1*, *7* and *4* are mixed up by the OCR engine due to their similarity in this specific font.

The co-operating company's production benchmark of 0.5 seconds per coffee packages is nearly matched with the first use case implementation—without any focus on runtime reduction. Interestingly, although it does not have multiple

Fig. 16 False positive due to wrong OCR recognition by mistaking the ripped part for the digit 4

Fig. 17 False Positive due to SIFT algorithm detecting enough key points

hidden layers, the SIFT algorithm is not the fastest in the package extraction step. The keypoint calculations and comparing with 5 reference images is computationally intense. However, the ROI extraction with only two loops (two ROIs on horizontal level) is quicker than YOLOv5 and DE:TR. The major reason for that is the image size as for SIFT, the input image size is the decisive factor in terms of detection speed. Hence, in both detection steps, the query image can be drastically resized due to the relative size of the objects compared to image size to increase processing speed.

Another benchmarking observation are the IoU results of the coffee package extraction. In order to always include the whole coffee package in the cutout frame, the YOLO and the DE:TR models *learn* to create a padding around the coffee packages. Therefore, the IoU scores are lower than the SIFT scores, since the ground truth bounding boxes are smaller. However, this generally helps the following object detector to detect the ROIs. This is underlined through the necessity of manually adding a padding to the very precise SIFT cutout frames. Otherwise, details on the borders of the package such as the barcode may be partially missing. However, the IoU score of the SIFT algorithm indicates that it is performing very well solely based on this specific metric.

Another takeaway during ROI detection is the importance of training data. The DE:TR does not always detect the best-by date and lot number ROI for two specific package designs. These are colored golden with a white font—hence offer relatively

Table 8 mAP of SIFT ROI detection in comparison to exclusively deep learning-based-CV extraction (mean Average Precision of individual package types in bold)

Average Precision	DETR_DETR (%)	DETR_SIFT (%)	YOLO_SIFT (%)	YOLO_DETR (%)
Expiry Date	98.04	61.04	79.43	96.00
Barcode	95.29	95.58	89.60	96.37
mAP	**96.67**	**78.31**	**84.52**	**96.19**

Table 9 Examplary confusion matrices including SIFT-based ROI extraction show that the SIFT model has difficulties extracting all Expiry Date/Lot Number-ROI areas based on the False Negative values

YOLO_SIFT	TP	FP	FN	TN
Expiry Date	30	0	5	5
Lot Number	31	0	6	3
Barcode	37	2	1	5
Top Sticker	38	2	0	10
Overall	25	2	5	18
DETR_SIFT	TP	FP	FN	TN
Expiry Date	30	0	5	5
Lot Number	31	0	6	3
Barcode	38	1	0	5
Top Sticker	38	2	0	10
Overall	26	2	4	18

little contrast. The YOLOv5 model does not seem to have any difficulty with that. This could be due to the YOLOv5 including multiple data augmentation steps at training, including advanced augmentation techniques such as mosaic augmentation. This makes it very suitable for cases with smaller amounts of training data. The DE:TR developers, on the other hand, propose a larger amount of training data for their models than it is available for each ROI model with respect to each coffee package type.

Since the SIFT algorithm is not a deep learning-based CV algorithm, it does not require any training which minimizes the initial efforts. Also, if a deep learning-based CV model has extracted the coffee packages first, the following detection of the ROIs is quicker using SIFT than with the YOLO or DE:TR models. However, no required training and this increase of detection speed is a trade-off with accuracy. Table 8 shows how the mAP decreases as soon as the SIFT algorithm is used for ROI detection. This is also highlighted in the respective confusion matrices, Table 9 displays that exemplarily. The high numbers in false negatives (FN) and small number in false positives (FP) indicate that the error is not due to the OCR algorithm, but because the respective ROIs could not be identified correctly by SIFT. Hence, SIFT performs very well on clearly segmented areas (Top Sticker and Barcode), but should be used more carefully for areas with less contrast.

Initially, two different approaches for ROI extractions were compared. First, a single overall ROI extraction model including images of all package variations was trained. As comparison, multiple individual models per coffee bean type were trained to be more accurate depending on the identified type. However, when observing the data, it became obvious that training multiple models per type does not increase the metrics significantly, and is hence not worth the additional effort. An exemplary comparison is listed in Table 10.

During implementation, the development of the acquisition and processing services was replicable and intuitive. However, developing the classification service and the classification models themselves proved to be the most complex task. As an example, difficulties came up for the OCR process with respect to detecting different fonts and resolution requirements. The pre-processing of the ROI cutouts appeared to be the decisive factor. As a solution, parameterizing the pre-processing steps allows testing different combinations. Also, it enables tailoring the configurations to different designs, while maintaining generalizability. The remaining parts of the pipeline, however, can be taken as-is and may be transferred to other use cases without major modifications—hence underlining the goal of this work.

The test results underline that it is possible to design an efficient multi-feature, end-to-end CV-based quality control based on the quality control framework. This is shown with the implementation of this framework for the use case of coffee packaging. Following the DCAI paradigm allowed for the classification of a variety of quality factors. It has been a substantial factor for the execution of this work, and one main reason that the achieved results have been as successful. Referencing works have already highlighted the value of data-centric focus during development (Lee et al 2021; Beyer et al 2020; Yun et al 2021). Without the re-recording and adjustments in pre-processing, the OCR extraction as well as the barcode decoding would have hardly been possible due to the initially chosen camera angles and resolution. The low initial resolution in combination with a diagonal camera degree does not allow both models to extract the information properly. Furthermore, erasing the illumination for the top sticker classification during post-processing is very tedious and would have likely resulted in a non-robust model. Therefore, the research question regarding *how* to develop such a system is successfully answered by underlining the functionality of the developed framework through the described use case implementation. This prototypical

Table 10 Exemplary mAP comparison of single and multi ROI model approaches (mean Average Precision of individual package types in bold)

Average Precision	DETR_DETR S[1] (%)	DETR_DETR M[2] (%)	YOLO_DETR S[1] (%)	YOLO_DETR M[2] (%)
Expiry date	98.04	94.00	100.00	96.00
Barcode	95.29	97.62	96.85	96.37
mAP	**96.67**	**95.81**	**98,43**	**96.19**

[1]One single coffee bean type-independent ROI extraction model

[2] Multiple coffee bean type-specific ROI extraction models

implementation can be considered a starting point for further development and allows for many insights to consider for future adoptions of this framework.

7.3 Re-recording of training data

As this work follows the DCAI paradigm, strong focus is put on the data acquisition itself, with the data being captured iteratively (Whang et al 2021). After recording data with the default parameters, these were adjusted based on development difficulties. First, the initially chosen resolution made it difficult to extract expiry date and lot number through the OCR process. As the underlying OCR technology is considered state-of-the-art in the non-proprietary OCR domain, the difficulties are most likely based in the training material itself. Consequently, the camera's frames per second (FPS) parameter is reduced and the resolution increased. In the actual production scenario, frames are pulled every 0.5 seconds, so lower FPS and higher resolution are feasible. These new frames increase the accuracy of the results derived from the OCR process, as well as the barcode extraction, tremendously.

Another iteration of data recording was conducted due to illumination problems with the top sticker classification during the second iteration. Edge detection algorithms such as SIFT were found to be vulnerable to reflections caused by light illuminations. Despite various pre-processing attempts, a robust solution could not be obtained. Hence, following the DCAI paradigm the recording parameters were adjusted. To minimize illumination-triggered reflections, light sources were placed to hit the packages from different angles. This improved the quality of frame cut outs and overcame modeling problems. The insights from these re-recordings can be applied in the actual usage of the developed solution in production.

8 Conclusion and outlook

This work presents an innovative computer vision-based framework for automated quality control in production and manufacturing. It allows examining multiple quality factor categories simultaneously, underlined through an real-life industry use case. As the packaging of artifacts generally fulfills a variety of purposes, multiple quality factor categories (visual, informational, etc.) need to be evaluated. After designing the framework, the exemplary integration of the theoretical framework is implemented into practice based on DCAI development practices.

Through this work, multi-feature quality control of (packaged) artifacts in the production area with respect to multiple error categories in packaging is conducted on a scientific basis for the first time. A generalized and extendable framework with a modular architecture was proposed which is able to aggregate defect classifications of a variety of error categories. As a result, manual human-conducted quality control processes can be represented even more realistically. This allows for advanced research in the field of computer vision-based quality control. On top of that, the framework allows companies to integrate automated quality control using (deep learning-based) computer vision—hence reducing economic inefficiencies.

Future adopters of this framework will profit from its focus on flexible customization to seamlessly integrate their existing solutions. In addition, the straight forward adoption and the benchmarking results are potential starting points for innovations regarding waste reduction. Thus, in combination with the overall societal shift towards ecological awareness, this framework supports the push of increased regulations towards sustainable production processes.

As the scope of this work had only been a prototypical implementation, a lot of improvement potential is given. E.g., additional quality factors such as deformation could be added. Also, the detection speed could be increased even further through additional pre-processing improvements such as image scaling (Růžička and Franchetti 2018), the modification of models through, e.g., layer reduction (van Rijthoven et al 2018) or the usage of even more lightweight models (Adarsh et al 2020; Womg et al 2018). On top of that, additional models, both traditional, such as background reduction focused (Haque et al 2008), as well as deep learning-based ones, e.g. Single Shot Multibox Detector (Liu et al 2016), could be included as well.

This implementation already proves the validity of this work's vision of deriving a generalized, adaptable framework for the automated quality control of packaged artifacts with the help of computer vision. It underlines how this framework can be applied to any computer vision-based quality control approach in the context of conveyor belt production processes—for both packaging as well as other production steps. Ultimately, this framework serves as a catalyst for future approaches and scientific works to further reduce material (and food) waste. Through this framework production companies, the consumers and the environment overall can profit—economically and ecologically.

Declarations

References

Adarsh P, Rathi P, Kumar M (2020) Yolo v3-tiny: object detection and recognition using one stage improved model. In: 2020 6th International conference on advanced computing and communication systems (ICACCS). IEEE, pp 687–694

Ansari A, Usama M, Siddiqui S et al (2019) Packaging features and consumer buying behavior towards packaged food items. SSRN Electron J 7(3):1050–1073

Banus N, Boada I, Bardera A et al (2021) A deep-learning based solution to automatically control closure and seal of pizza packages. IEEE Access 9:167267–167281

Banús N, Boada I, Xiberta P et al (2021) Deep learning for the quality control of thermoforming food packages. Sci Rep 11(1):21887

Benouis M, Medus LD, Saban M et al (2020) Food tray sealing fault detection using hyperspectral imaging and pcanet. IFAC PapersOnLine 53(2):7845–7850

Beyer L, Hénaff OJ, Kolesnikov A et al (2020) Are we done with imagenet? arXiv preprint arXiv:2006.07159

Biswas A, Jana AP, Mohana A et al (2018) Classification of objects in video records using neural network framework. In: 2018 International conference on smart systems and inventive technology (ICSSIT), pp 564–569

Blasco J, Aleixos N, Cubero S et al (2009) Automatic sorting of satsuma (citrus unshiu) segments using computer vision and morphological features. Comput Electron Agric 66(1):1–8

Blasco J, Aleixos N, Gómez-Sanchís J et al (2009) Recognition and classification of external skin damage in citrus fruits using multispectral data and morphological features. Biosys Eng 103(2):137–145

Carion N, Massa F, Synnaeve G et al (2020) End-to-end object detection with transformers. In: European conference on computer vision. Springer, pp 213–229

Dmitry N, Manfred SS (2014) On micro-services architecture. Int J Open Inf Technol 2(9):24–27

Erwanto D, Prasetyowati SAD, Budi Susila EN (2017) Utilization of digital image processing in process of quality control of the primary packaging of drug using color normalization method. IOP Conf Ser Mater Sci Eng 190(1):012043

Gedraite ES, Hadad M (2011) Investigation on the effect of a gaussian blur in image filtering and segmentation. In: Proceedings ELMAR-2011. IEEE, pp 393–396

Gunasekaran S (1996) Computer vision technology for food quality assurance. Trends Food Sci Technol 7(8):245–256

Haji M, Kerbache L, Muhammad M et al (2020) Roles of technology in improving perishable food supply chains. Logistics 4(4):33

Haque M, Murshed M, Paul M (2008) A hybrid object detection technique from dynamic background using gaussian mixture models. In: 2008 IEEE 10th workshop on multimedia signal processing. IEEE, pp 915–920

Henderson P, Ferrari V (2016) End-to-end training of object class detectors for mean average precision. In: Asian conference on computer vision. Springer, pp 198–213

Hoseinpour Z, Amir S, Kheirkhah S et al (2020) The problem solving of bi-objective hybrid production with the possibility of production outsourcing through meta-heuristic algorithms. Int Technol Sci Publ 4:1–17. https://doi.org/10.31058/j.mana.2021.42001

Hoseinpour Z, Taghipour D, Beigi H et al (2021) The problem solving of bi-objective hybrid production with the possibility of production outsourcing through imperialist algorithm, nsga-ii, gapso hybrid algorithms. Turk J Comput Math Educ 12:8090–8111

Huaiyuan S, Chenjie S, Yuehua L (2013) The detection system for pharmaceutical bottle-packaging constructed by machine vision technology. In: 2013 Third international conference on intelligent system design and engineering applications, pp 1423–1425

Illingworth J, Kittler J (1988) A survey of the hough transform. Comput Vis Graph Image Process 44(1):87–116

Jakubik J, Vössing M, Kühl N, Walk J, Satzger G (2024) Data-centric artificial intelligence. In: Business & Information Systems Engineering

Jin X, Hirakawa K (2012) Analysis and processing of pixel binning for color image sensor. EURASIP J Adv Signal Process 1:1–15

Jocher G, Chaurasia A, Stoken A et al (2022) ultralytics/yolov5: v6.1-tensorrt, tensorflow edge tpu and openvino export and inference

Kang CW, Ramzan MB, Sarkar B et al (2018) Effect of inspection performance in smart manufacturing system based on human quality control system. Int J Adv Manuf Technol 94(9):4351–4364

Kühl N, Schemmer M, Goutier M, Satzger G (2022) Artificial intelligence and machine learning. Electronic Markets 32(4):2235–2244

Lee JS (1983) Digital image smoothing and the sigma filter. Comput Vis Graph Image Process 24(2):255–269

Lee Y, Kwon OJ, Lee H et al (2021) Augment and valuate: a data enhancement pipeline for data-centric ai. arXiv preprint arXiv:2112.03837

Liu W, Anguelov D, Erhan D, et al (2016) Ssd: Single shot multibox detector. In: Leibe F (ed) Computer vision—ECCV 2016. LNCS sublibrary: SL6 - Image processing, computer vision, pattern recognition, and graphics. Springer, Cham, pp 21–37

Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2. IEEE, pp 1150–1157

Mahony NO, Campbell S, Carvalho A et al (2020) Deep learning versus traditional computer vision. Adv Comput Vis Proc 9:43

Medus LD, Saban M, Francés-Víllora JV et al (2021) Hyperspectral image classification using CNN: application to industrial food packaging. Food Control 125(107):962

Nandakumar N, Saleeshya P, Harikumar P (2020) Bottleneck identification and process improvement by lean six sigma dmaic methodology. Mater Today Proc 24:1217–1224. https://doi.org/10.1016/j.matpr.2020.04.436

Nandi CS, Tudu B, Koley C (2014) Computer vision based mango fruit grading system. In: International conference on innovative engineering technologies (ICIET 2014), pp 28–29

Palumbo PW, Swaminathan P, Srihari SN (1986) Document image binarization: evaluation of algorithms. In: Applications of digital image processing IX, SPIE, pp 278–285

Paszke A, Gross S, Massa F et al (2019) Pytorch: an imperative style, high-performance deep learning library

Perrey R, Lycett M (2003) Service-oriented architecture. In: 2003 Symposium on applications and the internet workshops, 2003. Proceedings. IEEE, pp 116–119

Popovic I, Bossink BAG, van der Sijde PC (2019) Factors influencing consumers' decision to purchase food in environmentally friendly packaging: what do we know and where do we go from here? Sustainability 11(24):7197

Poyatos-Racionero E, Ros-Lis JV, Vivancos JL et al (2018) Recent advances on intelligent packaging as tools to reduce food waste. J Clean Prod 172:3398–3409

Razmjooy N, Mousavi BS, Soleymani F (2012) A real-time mathematical computer method for potato inspection using machine vision. Comput Math Appl 63(1):268–279

Redmon J, Divvala S, Girshick R et al (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

Rezatofighi H, Tsoi N, Gwak J et al (2019) Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 658–666

Ribeiro FDS, Gong L, Calivá F et al (2018) An end-to-end deep neural architecture for optical character verification and recognition in retail food packaging. In: 2018 25th IEEE international conference on image processing (ICIP), pp 2376–2380

Růžička V, Franchetti F (2018) Fast and accurate object detection in high resolution 4k and 8k video using gpus. In: 2018 IEEE high performance extreme computing conference (HPEC). IEEE, pp 1–7

Sa J, Li Z, Yang Q et al (2020) Packaging defect detection system based on machine vision and deep learning. In: 2020 5th International conference on computer and communication systems (ICCCS). IEEE, pp 404–408

Serdarasan S (2013) A review of supply chain complexity drivers. Comput Ind Eng 66(3):533–540

Smith R (2007) An overview of the tesseract ocr engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007). IEEE, pp 629–633

Soille P (2004) Erosion and dilation. In: Morphological image analysis. Springer, pp 63–103

Sourceforge (2011) Zbar - zbar bar code reader. http://zbar.sourceforge.net/. Accessed 04 June 2022

Thota M, Kollias S, Swainson M et al (2020) Multi-source domain adaptation for quality control in retail food packaging. Comput Ind 123(103):293

Ting KM (2017) Confusion matrix. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning and data mining. Springer, Boston, pp 260–260

Vakili M, Ghamsari M, Rezaei M (2020) Performance analysis and comparison of machine and deep learning algorithms for iot data classification. arXiv preprint arXiv:2001.09636

van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440

van Rijthoven M, Swiderska-Chadaj Z, Seeliger K et al (2018) You only look on lymphocytes once

Vergara-Villegas OO, Cruz-Sánchez VG, de Jesús O-DH et al (2014) Automatic product quality inspection using computer vision systems. Lean manufacturing in the developing world. Springer, New York, pp 135–156

Voulodimos A, Doulamis N, Doulamis A et al (2018) Deep learning for computer vision: a brief review. Comput Intell Neurosci 2018

Wang Y, Cheng Z, Hou J et al (2012) Quality inspection improvement for cheese packaging using machine vision. 2154–0357:1–5

Whang SE, Roh Y, Song H et al (2021) Data collection and quality challenges in deep learning: a data-centric ai perspective. arXiv preprint arXiv:2112.06409

Williams H, Wikström F, Otterbring T et al (2012) Reasons for household food waste with special attention to packaging. J Clean Prod 24:141–148

Womg A, Shafiee MJ, Li F et al (2018) Tiny ssd: a tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In: 2018 15th Conference on computer and robot vision (CRV). IEEE, pp 95–101

Yun S, Oh SJ, Heo B et al (2021) Re-labeling imagenet: from single to multi-labels, from global to localized labels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2340–2350

Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Synth Lect Artif Intell Mach Learn 3(1):1–130

Zhu L, Spachos P, Pensini E et al (2021) Deep learning and machine vision for food processing: a survey. Curr Res Food Sci 4:233–249

Zielińska D, Bilska B, Marciniak-Łukasiak K et al (2020) Consumer understanding of the date of minimum durability of food in association with quality evaluation of food products after expiration. Int J Environ Res Public Health 17(5):1632

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Felix Xiong received the M.Sc. in Information Systems from the Karlsruhe Institute of Technology in 2022. His scientific research focuses on applied artificial intelligence and semantic web technologies. He wrote his bachelor's thesis in co-operation with the Beijing Institute of Technology in the field of secure open-source mobile operation systems, and his master's thesis in co-operation with the Schwarz Group in Germany about computer vision-supported quality control. Currently, he is working as a Technical Program Manager at Microsoft in Dublin, Ireland.

Niklas Kühl received a Ph.D. in information systems with a focus on applied machine learning and his Habilitation in applied computer science. In his studies, he is working on conceptualizing, designing, and implementing artificial intelligence (AI) products, focusing on inter-organizational learning and fair and effective collaboration within human–AI teams. He is a Full Professor of information systems and human-centric AI at the University of Bayreuth. He is also a Group Lead with Fraunhofer FIT for business analytics and a Senior Expert in artificial intelligence with IBM. In the past, he was a Managing Consultant for Data Science with IBM, which complemented his theoretical knowledge with practical insights from the field. He has worked on machine learning (ML) and AI in different domains since 2014. He collaborates internationally with institutions such as the University of Texas, Carnegie Mellon University, and the MIT–IBM Watson AI Laboratory.

Maximilan Stauder received the M.Sc. in Industrial Engineering from the Karlsruhe Institute of Technology in 2017 with distinction. During his studies, his scientific research focused on Machine Learning and Computer Vision. At the end of his studies, he wrote and published his master's thesis in co-operation with the Audi AG. Currently, he is working as head of Artificial Intelligence Industry at the Schwarz Group in Heilbronn, Germany.