Machine Vision and Applications (2025) 36:39
https://doi.org/10.1007/s00138-024-01636-x

RESEARCH

A computer vision system for recognition and defect detection for reusable containers

Vincent Wahyudi[1] · Cedric C. Ziegler[1] · Matthias Frieß[1] · Stefan Schramm[2] · Constantin Lang[1] · Lars Eberhardt[1] · Fabian Freund[2] · Alexander Dobhan[1] · Martin Storath[1]

Received: 9 August 2024 / Revised: 31 October 2024 / Accepted: 15 November 2024

Abstract
Small load carriers (SLCs) are standardized reusable containers used to transport and protect customer goods in many manufacturers. Throughout the life cycle of the SLCs, they will be collected, manually checked for defects (wear, cracks, and residue on the surface), and cleaned by specialized logistic companies. Human operators in small to medium-sized companies manually evaluate the defects due to the variety and degree of possible defects and varying customer needs. This manual evaluation is not scalable and prone to errors. This work aims to fill this gap by proposing a computer vision system that can recognize the SLC type for inventory management and perform defect detection automatically. First, we develop a camera portal, consisting of standard components, that capture the relevant surfaces of the SLC. A labeled dataset of 17,530 images of 34 different SLCs with their defect status was recorded using this camera portal. We trained a classification model (ConvNeXt) using our dataset to predict the different types of SLCs achieving 100% class prediction accuracy. For defect detection, we explore eight state-of-the-art (SOTA) anomaly detection models that achieved high rankings in the MVTec industrial anomaly detection benchmark. These models are trained using default hyperparameters and the two highest-scoring models were chosen and fine-tuned. The best-fine-tuned models based on "Area under the Receiver Operating Characteristic Curve (AUROC)" are PatchCore (0.811) and DRAEM (0.748). These results indicate that there is still potential for improvement in the automation of defect detection of SLCs.

Keywords Industrial object classification · Industrial anomaly detection · Machine learning · Quality control · Small load carrier · Reusable containers

✉ Vincent Wahyudi
vincent.wahyudi@thws.de

Cedric C. Ziegler
cedric.ziegler@thws.de

Matthias Frieß
matthias.friess@study.thws.de

Stefan Schramm
s.schramm@taf-industriesysteme.com

Constantin Lang
langconstantin@gmail.com

Lars Eberhardt
lars.eberhardt@thws.de

Fabian Freund
f.freund@taf-industriesysteme.de

Alexander Dobhan
alexander.dobhan@thws.de

Martin Storath
martin.storath@thws.de

[1] Technische Hochschule Würzburg Schweinfurt (THWS), Ignaz-Schön-Straße 11, Schweinfurt 97421, Bavaria, Germany

[2] TAF Industriesysteme GmbH, Rottendorf 97228, Bavaria, Germany

1 Introduction

Small load carriers (SLCs) are reusable containers with different shapes and sizes commonly used in industries such as automotive, manufacturing, and electronics. SLCs are reusable, stackable, standardized size for the respective types of SLCs, and protect the goods that are being transported using the SLCs [5]. The reusable nature of SLCs makes them a sustainable solution. Due to the ever-increasing global exchange of goods, the logistics industry is continuously growing and the need for standardized reusable containers such as SLCs and pallets is increasing [9]. Companies in the logistics industry have been paying extra attention to the container management system and status management [22]. The SLC quality assessment, which is a part of the container management system and status management, is an important step in determining whether SLCs are usable or contain defects that might pose risks of damaging customer goods during transport, which can be costly. Besides directly affecting the quality of the customer's goods, it can also affect the efficiency when moving the SLCs (e.g. breakage in the outer wall hinders proper SLC movements) [32]. Despite the critical nature of the inspection and evaluation of the SLCs, logistics companies still rely largely on manual human evaluation for the SLCs which is not scalable [23]. Automation of quality control of SLCs can reduce the risk of human errors during the sorting and checking of the SLCs due to fatigue and the repetitive nature of the job [18]. Human personnel can be allocated to other more important processes in the company such as supervising the system which may increase the general workflow. A reliable SLC quality control system removes the subjective bias of different human operators when evaluating containers that have different degrees of defects [16].

Automation of SLC quality control is not commonly implemented as it requires a trained model for the classification of the type of the SLC and identification of the condition of the SLCs (normal or anomalous) which requires a large amount of SLC data usually an image-based dataset. To our knowledge, there is no automated system for large data collection of SLCs for model training purposes. Another gap between the research field and the industrial field is that anomaly detection (AD) models are mostly tested on large controlled datasets such as MVTec AD [4] that might not represent the defect characteristics in SLCs. So it cannot be expected that AD models trained on benchmark datasets provide the same performance for quality control for SLCs [36]. To address these gaps, this study raises the following research questions: How can we implement a computer vision system for classifying the different types of SLCs and for assessing their conditions? Which of the different state-of-the-art (SOTA) methods for anomaly detection can be implemented in evaluation of SLCs? What are the limitations of the respective methods? How do the hyperparameters affect the output of the respective model? This research aims to fill these gaps by designing a system for data collection of the SLCs that are scalable and can be used to train a model for the classification and anomaly detection of the SLCs as shown in Fig. 1. The main contributions of this research are as follows:

- **Development of a camera portal**: We designed a modular, cost-effective camera portal using standard components to capture key areas of the SLC surfaces. The

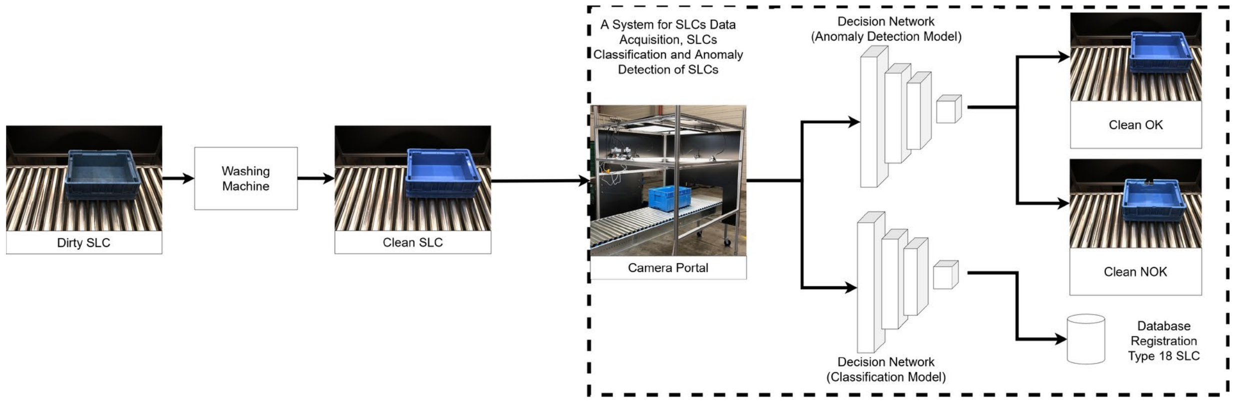
Dirty SLC
Washing Machine
Clean SLC
A System for SLCs Data Acquisition, SLCs Classification and Anomaly Detection of SLCs
Camera Portal
Decision Network (Anomaly Detection Model)
Decision Network (Classification Model)
Clean OK
Clean NOK
Database Registration Type 18 SLC

Fig. 1 General implementation of the cleaning, classification, and anomaly detection of the different types of SLCs. The dotted rectangle on the right side of the figure is the implementation of trained models for the classification of the type of SLC and the anomaly detection pipeline. Dirty SLCs are first collected and transported on conveyor belts to automated washing machines. The SLCs are cleaned and further transported to the camera portal. Images from all sides of the SLCs are taken and further evaluated using neural networks to predict the type of the SLCs and the condition of the SLCs (OK or NOK). The explanation for the NOK SLCs can be given as a heatmap evaluation of the potential point of anomaly based on the respective images. SLCs with different conditions are later separated using an automated conveyor system for further transport and evaluation. The predicted type of SLCs are later stored in a database

camera portal can be installed in industrial production lines without disrupting the flow of the production.

- **Creation of an extensive labeled dataset**: We compiled a dataset of 17,530 images for 34 different SLC types. Each image is annotated for both classification (type) and defect detection (OK/NOK). Of these, a subset of 9,260 images from the whole SLC dataset, covering 18 SLC types, that does not contain customer's credentials and is made publicly available.
- **Effectiveness of classification model on the dataset**: We demonstrate that fine-tuning an off-the-shelf deep convolutional neural network provides highly accurate results for the classification of the SLC type.
- **Comparison of SOTA anomaly detection methods**: We conduct a performance comparison of eight state-of-the-art anomaly detection methods for the SLC defect detection task, and identify the two most promising candidates for further evaluation.
- **Analysis of optimized models**: We provide both quantitative and qualitative analysis on hyperparameter-tuned versions of these candidates. We conduct the quantitative analysis by evaluating various performance metrics. The models' predicted image outputs are assessed to determine their effectiveness in accurately identifying anomalous sections within the images.

1.1 Organization of the paper

The structure of this paper is as follows: Sect. 2 discusses the prior and related works on the building of the system for data acquisition and the models for the classification of SLC types and identification of the SLC defects. Section 3 describes the design decisions of the system for data acquisition which is the camera portal. Section 4 describes the dataset collected from the designed system and the preprocessing steps taken before model training. Section 5 discusses the classification algorithm used for the SLC dataset. Section 6 discusses the different SOTA algorithms used for anomaly detection that are commonly implemented for industrial implementation. Exploratory analysis of the SOTA algorithms is conducted along with hyperparameter tuning to adapt the models to the present data set. Qualitative and quantitative analyses are conducted on the best models. Section 7 discusses the result of the experiments and explores different sectors of the experiment that can be improved.

2 Prior and related work

Prior and related work of this study will be divided into three main parts: data acquisition and evaluation system in industry (Sect. 2.1), image classification (Sect. 2.2), and image-based anomaly detection for industrial objects (Sect. 2.3). Each part further discusses what has been done on other studies for the respective topics.

2.1 Data acquisition and evaluation system in industry

A reliable system for data acquisition of the SLCs is critical in developing a consistent object detection and anomaly detection system. Considerations such as the physical features of the SLCs (whether the surface of the object is reflective when shined with lights), the on-site hard constraints such as the maximum dimension of the system, the number of cameras needed, the movement of the SLCs during production, the lighting condition on-site and the computing power needed are all important details that need to be considered before designing the system [48]. Careful planning of the camera setup and the lighting condition for the image acquisition is crucial as background noise or other objects that are not relevant to the quality control process can affect the evaluation of the SLC [46]. Poss et al. [36] describe the characteristics of the SLCs that they use in the production line as well as the hard constraints such as cycle time of the line that must be kept as efficient as possible. The evaluation system must conform to the hard constraints of the line to avoid slowing down production.

Sobottka et al. [45] use CCD (Charged-Coupled Device) cameras in their experiment for detecting dust and other types of debris on SLCs. They emphasize the importance of having multiple camera setups from different angles and a stable light source preferably in an all-enclosed tunnel-like structure to avoid external uncontrollable lighting. The usage of conveyor belts and switch plates to separate different SLC conditions reflects the general use case of our needed system. Despite all the suggestions, Sobottka et al. simulated the setup in a simulation model layout which can be different in real use scenarios. Bohm et al. [6], Xu et al. [54], and Noceti et al. [34] emphasize the importance of multiple camera setups (with Noceti et al. [34] emphasizing the importance of top view camera position). It is more practical to have a "view insufficiency" situation where images have overlapping information in environments with a lot of movement.

Pierer et al. [35] experimented on a scalable multi-camera (13 monochrome camera setup) inspection approach for industrial press line applications with parts moving on top of conveyor belt and four high-powered LED illumination bars. Zhang et al. [57] conducted experiments on object detection with the subject apples on modified YOLO (You Only Look Once) models with different illumination. Different illuminations (front, side, and backlight) change the appearance of the apple which affects the ability of the model to correctly identify the apples in the image. Most of the prior experiments emphasize the benefit of using a multi-camera setup with controlled lighting conditions to have the best image quality acquired from the object.

2.2 Image classification

SLCs are divided into different classes based on their physical properties (colors, dimensions, and geometry). The collected images from the data acquisition system of the SLCs are labeled with the respective SLC type, creating a labeled dataset for normal instances. This can be used to train neural networks for common classification problems. Different models can be used for common classification problems such as vision transformers (ViT) [15] and convolution neural networks (CNNs). ViT may perform better generally on complicated problems with more data. However, for this particular use case of classifying SLCs with limited datasets, CNNs might be easier to train with the limited data and easier to fine-tune for better classification results. ConvNeXt [30] model is a modernization of ResNet (Residual Network) which is inspired by the design of Vision Transformers while maintaining the simplicity and efficiency of CNNs. General architecture design of ConvNeXt that differentiates it from traditional CNNs includes substituting the stem cell in ResNet with a simpler "patchify" layer as in ViT (4x4 non-overlapping convolution), implementation of depthwise convolution (a special grouped convolution where the number of groups equals the number of channels popularized by MobileNet [24]), inverted bottleneck block which was popularized by MobileNetV2 [42], implementation of 7x7 depthwise conv in each block, substitution of Rectified Linear Unit (ReLU) [33] with Gaussian Error Linear Unit (GeLU) [21], implementing single GeLU in each block, and changing the batch normalization [26] into layer normalization [2].

2.3 Image-based anomaly detection for industrial objects

Anomaly detection (AD) refers to the detection of patterns in data that do not conform to expected behavior; it is widely applied in various fields, such as fraud detection in banking, identifying spam in emails, fault detection in industrial systems, and detecting abnormalities in medical images [8]. The key challenge in AD lies in accurately distinguishing between normal variations and genuine anomalies, especially in complex datasets. To mention a few examples, Huang et al. [25] implemented visual language models for AD in medical images that can detect anomalies from different kinds of medical images. AD is also pivotal in the predictive maintenance of machinery, where algorithms analyze sensor data or machine logs to identify unusual patterns indicative of potential faults; for example, Wang et al. [52] used a reinforcement learning-based method for fault estimation of the actuators. In the field of networks, anomaly detection is used to identify malformed packets [7].

This study focuses specifically on anomaly detection for image-based data types in the industrial domain. It is a complex problem as anomaly instances in the industry can be diverse, unpredictable, rare, and irregular. The rare nature of anomalies made it impractical and costly to intentionally produce them in industry. We next provide a brief overview of anomaly detection for industrial images related to the present setup; for a broader discussion on the field we refer to the recent survey by Liu et al. [29]. Bai et al. [3] discussed methods for solving the problem of imbalance datasets due to the rare nature of anomalies in the industrial field.

Ruff et al. [41] explain that anomaly detection approaches can be classified based on the feature maps and models. Feature maps can be differentiated as deep or shallow methods. Models can be classified into classification, probabilistic, and reconstruction-based. The categorization of the anomaly detection approaches is not discrete as some approaches lie between different models. Other anomaly detection approaches can rely purely on distance-based methods.

Several implementations have been previously implemented in the industrial fields. Liang et al. [28] use an industrial camera to capture the inkjet printing of production codes containing information (expiry date, batch number) on plastic beverages to check for defects such as missing letters. ShuffleNet V2 network is used to classify if the printed code is defective or normal. Singh et al. [44] uses pretrained ResNet-101 for feature extraction and multi-class support vector machines (SVMs) to detect surface defects of tapered rollers. He et al. [20] implemented an end-to-end steel surface defect detection approach using a fusion of multiple hierarchical features for potential real-time detection. Dlamini et al. [14] use the YOLOv4 model for textile industry quality control. The implementation of enhanced feature extraction YOLO industrial small object detection algorithm [47] can also be implemented for detecting specific anomaly types in close-up SLC images. Würschinger et al. [53] developed a low-cost machine vision system for piston rod quality evaluation using transfer learning.

A lot of the implementations of AD in industrial objects are for new production objects which have mostly a fixed degree of the normal object. Reusable objects such as SLCs are rarely in perfect condition when returned for inspection.

This introduces another complexity level as the surface of the normal SLCs can slightly differ while still considered normal. Truong et al. [49] use a camera to evaluate reusable food packaging (namely plastic cups) for defects and contamination. They implemented background removal using U-Net [39] to highlight the object from the background and trained the model based on an auto-encoder-based framework for anomaly detection. Despite the similar reusable condition of the object, this study has some slightly different conditions. SLCs have different physical features such as size and color which is different from only evaluating one type of object. As SLCs are not transparent, all the inner and outer surfaces of the SLCs must be evaluated, thus requiring more than one view for a sufficient quality image.

3 Design of the camera portal

Designing the camera portal is further divided into three different parts: requirements (Sect. 3.1), geometrical calculation (Sect. 3.2), and final build of the camera portal and hardware used (Sect. 3.3). The requirements section discusses the basic requirements needed for the camera portal to function and collect data properly. The geometrical calculation elaborates on the positioning of the cameras and how it affects the images taken. The final build of the camera portal and the specific materials used for building the prototype are further elaborated in the last part.

3.1 Requirements

The data acquisition system needs to solve both the classification of the SLC types and determine whether it is defective. While the first task (classification) can be solved with a single camera view, the second task (defect detection) requires a complete view of all the surfaces (inside and outside) of the SLC as the defect can be anywhere on the surface of the SLC. The bottom surface is excluded as it is rarely defective and the defects usually have little negative impact on the usability of the SLC. The system design needs to be under certain constraints such as not slowing down the SLC cleaning process, need to fit into the facilities, and being economical while maintaining quality. For cost and simplicity reasons, a setup based on a single camera moving around the object was ruled out as the mechanism would be very complicated. The geometry of the containers suggests that at least three (static) cameras are needed; two cameras on the diagonals of the SLC and one camera for the top view. The three-camera setup is cost-effective for capturing images but a preliminary test revealed that the diagonally placed camera setup caused occlusion for higher containers, exhibited a notable perspective distortion, and a relatively high sensitivity to shift of the SLC location. Adding two more cameras and placing the four cameras at the parallel sides of the SLCs and one camera on the top of the SLC mitigated these issues. Implementing two additional cameras means more cost but is still justifiable. To summarize the camera portal uses a five-camera setup where four are placed on the parallel sides of the SLCs and one at the top of the portal frame (providing the top view).

3.2 Geometrical calculation

Choosing the appropriate camera characteristics and selecting the proper position and orientation of the camera for SLC evaluation is crucial for capturing important details of the object. Geometrical calculations help identify potential blind spots that might affect the performance of models. Higher camera resolutions are considered in capturing more detailed images for better model performance. The spatial resolution needed to capture the details for anomalous sample can be set as 1 pix/mm as the nature of anomalous parts in SLCs are larger in dimensions. Furthermore, considering an additional safety factor of three, the required spatial resolution is 3 pix/mm. For choosing the camera resolution, the calculation must consider the largest dimension of the SLC which is 280 mm (height), 600 mm (length), and 400 mm (width). The calculated minimum sensor dimension for the given distance of the portal is 1800 × 840 pixels for the longer sides of the SLCs. This paper uses U3-3880CP-C-HQ Rev.2.2 from IDS Imaging Development Systems GmbH which is an RGB camera with 6.41 MP resolution (3088 × 2076 pixels), frame rate 59.0 fps, and C-Mount type. As the position of the camera will be static and the position of the incoming SLC will be approximately in the same spot (with a little variation on the exact positioning), the initial setup of the camera must be chosen to avoid lens and perspective distortion. The calculation of the field of view (FOV) depends on the camera sensor width or height and the focal length used. The determined FOV also affects the working distance of the camera. When SLCs are placed closer than the working distance, then some part of the SLC might be cut off. The working distance is related to the focal length, where a longer focal length means a longer working distance. The size constraints of the camera portal setup are 1.5 × 1.5 m (width × length). If the SLC is placed in the center of the camera portal setup, the longer side of the largest SLC has a constraint of a maximum working distance of 450 mm. The shorter side of the largest SLC has a constraint of a maximum working distance of 550 mm. Due to the physical size constraints, the camera needs to be set higher to mitigate the longer working distance. Setting a higher camera position will have lower visibility on the sides of the SLCs and capture more background that might not contribute to the performance of the model. Considering all the constraints present, the focal length of 8 mm is used for the cameras and the height is 500 mm from the edge of the highest SLC (780 mm above the rollers of the conveyors). The inclination of the camera is set at approximately 47.5°.

3.3 Final build of the camera portal and hardware used

The final prototype of the camera portal was built using aluminum structural beams with mountings for the camera and underneath the portal is an industrial roller conveyor for the movement of the SLCs. Diffused lights were installed on the top of the camera portal to reduce shadowing in the SLCs. Opaque plastic sheets are used to cover the camera portal to provide stable lighting conditions (enclosed tunnel-like structure). Additional tube lights are installed on the horizontal aluminum structural beams to provide additional lighting. Cameras are set to trigger all at the same time to provide a uniform condition on all the images. The movement of the SLC in the conveyor is controlled by a Programmable Logic Controller (PLC). Sensors and PLC are used to control the position of the SLCs to the center of the camera portal and the movement after the images are taken. Figure 1 (Camera Portal) shows the setup of the camera portal prototype. The detailed types of hardware used are shown in Table 1.

Table 1 The hardware details used to build the camera portal for SLC data acquisition

Part name	# of Items	Type	Material	Details
Roller conveyor	1	Driver roller and support roller	Galvanized steel	636 mm width
PLC controller	1	CPU 1211C DC/DC/DC	–	–
Controller	2	ConveyLinx Ai2	–	–
Sensors	3	OPT1508	–	Reflex sensor
Portal structure	1	–	Aluminum bars	–
Light tube	4	Glass	LED	–
Diffused lights	2	50-50-Sled-1-VA2-18W	Acryl 2 mm white	–
Camera	5	U3-3880CP-C-HQ Rev 2.2	–	3088 × 2076
Partition	5	PVC hard foam	PVC	–

Each part has its total number of items used, the specific type name, the material used for the particular part, and additional important details. The # symbol stands for number

4 Dataset acquisition and preprocessing

This section explains the details of the dataset collected and further used for training the AD models. The total collected images are 17,530 images of 34 types of SLCs with an unbalanced number of images for each class of SLCs. The SLCs can be further differentiated into different shapes such as covers, inlays, and boxes. The SLCs consist of different colors and can have a mixture of two colors in an SLC box such as orange with yellow stripes.

The SLCs were labeled as normal or anomalous by an expert familiar with SLC defects. Each SLC corresponds to five images and anomalies may only be present on one of the sides. Thus, the dataset is further preprocessed by manually choosing the OK views from the anomalous SLCs and moving it to the normal view SLCs. Approximately 17.9% of the total number of recorded images were labeled as anomalies after the additional preprocessing. Anomalies can range from scratches, cracks, broken parts on the edges of the SLCs, the remainder of old stickers, and the presence of foreign objects inside the SLCs. For the training purposes of the models, resizing of the original images is carried out from 3088 × 2076 to 256 × 256. For AD training pipelines, some models are pre-processed on an additional background removal pipeline to remove unwanted background noise and see the performance difference between the models with and without background removal. The method used for the background removal is U^2-Net [37] which is a pre-trained model that processes the input image into an output image that only contains the SLCs without the backgrounds. Normalization of the images is done based on ImageNet mean and standard deviations.

For our experiments, we use the following four variants of the dataset: all SLC with background removal, all SLC without background removal, sampled SLC without background removal, and inference dataset. The sampled SLC dataset is a subset of the whole dataset where it is sampled using a stratified sampling method. This sampled dataset contains 7430 total images with 2970 images as the normal train images. Approximately 42.3% of the total sampled SLC dataset are considered anomalies. The sampled dataset was built for hyperparameter tuning due to time and resource constraints. The inference dataset is a custom-made dataset containing a mixture of normal and abnormal views of the SLCs. Each of the 34 types of SLCs are included in this dataset. The total number of images is 265 which 155 are normals and 110 are anomalies. 11 out of the 34 types of SLCs do not have defective samples.

5 Classification of SLC type

The experiments start with collecting the dataset as explained in Sect. 4. The collected images of the SLCs are used as training images for the classification model to differentiate the different types of SLCs. The model used for the classification problem is the ConvNeXt [30] model which is a modernization of ResNet towards the design of Transformers. Details of the parameters used for the training and other additional processes are discussed further in Sect. 5.1. Further analysis of the model for the inference times to evaluate the performance of the model are elaborated in Sect. 5.2.

5.1 ConvNeXt training and additional preprocessing

The collected SLC dataset is further divided into train, test, and validation datasets based on the 60:20:20 split. The dataset used for the training, testing, and validation of the classification model has a mixture of OK and NOK images of each type of SLC. The training of the dataset used the following hyperparameters: base-ConvNeXt model, $4 \cdot 10^{-3}$ for the learning rate with AdamW optimizer [31], and batch size is 64. Training is executed for 250 epochs. Normalization parameters for the images (mean and standard deviation for the RGB (red, green, blue) values) are from ImageNet.

Table 2 Inference testing for the classification model is implemented using two different datasets: inference dataset which contains 265 images as explained in Sect. 4 and a 5-image inference which a folder contains five different views of the same SLC

Inference setup	Hardware	Average time (secs)	Inference time per image (secs)
Inference dataset (265 images)	CPU-only (i5-1335U)	65.78	0.25
Inference dataset (265 images)	CPU-only (Ryzen 3970X)	36.38	0.14
Inference dataset (265 images)	GPU (RTX 3090)	18.34	0.07
5-image inference	CPU-only (i5-1335U)	1.13	0.24
5-image inference	CPU-only (Ryzen 3970X)	0.67	0.13
5-image inference	GPU (RTX 3090)	0.51	0.10

There are two hardware used to test the model. First is the hardware used in the study as explained in Sect. 6.2 and the second one is the CPU-only laptop. This shows how different hardware and configurations (using CPU-only or with GPU) can affect the average time and inference time per image. All tests are executed five times and the average time is taken

5.2 ConvNeXt inference and analysis

Inference tests for the trained ConvNeXt are shown in Table 2. The inferencing is done with the same hardware as explained later in Sect. 6.2 and additional hardware laptop with Intel i5-1335U CPU. The inferencing dataset used the same inference dataset as explained in Sect. 4 and a 5-image inference to replicate the real-life use case where inferencing is done using 5 cameras in the camera portal. Each run is done five times for the same task and the average time of the five runs is taken. As of now, the recognized SLC types are stored locally. In a later stage, they will be sent to a central database which will allow analysis of the throughput and the stock in an enterprise resource planning (ERP) system. The integration into the ERP system will be reported elsewhere.

6 Defect detection

Next, we investigate camera-based defect detection. There are different kinds of possible defects such as residue of old stickers, rust stains, oil spillage, surface wear, cracks, and breakage that can occur. Currently, the defect type is of low importance. The number of defective SLCs in real-world scenarios is very low (a single-digit percentage or less compared to the total number of SLCs evaluated). Thus, modeling this task as a visual anomaly detection is reasonable. Defect detection are divided into three subsections: exploration of the SOTA AD methods and hyperparemeter tuning (Sect. 6.1), experimental setup (Sect. 6.2), and experimental results (Sect. 6.3).

6.1 Explorative analysis of SOTA anomaly detection methods and hyperparameter tuning

To keep the implementation effort manageable, this study focuses on anomaly detection methods that can be invoked in a standardized manner using the Anomalib Python library [1]. Table 3 categorizes SOTA deep anomaly detection models based on their taxonomy, loss function, and the pre-trained model used if any. The table format and explanations are inspired by the work of Liu et al. [29]. Student-teacher model approaches depend on pre-trained models such as ResNet [19] and VGG [43] as the teacher network. The teacher networks are trained on large datasets such as ImageNet [13] for normal feature extraction. A particular layer of the pre-trained network is selected as a parameter for the teacher network. The teacher network then teaches a simpler student network on extracting normal features. Anomalies are detected when the features extracted differ vastly between the teacher and student networks. An anomaly score is generated by comparing the features generated

Table 3 SOTA anomaly detection models based on the taxonomy, methods, loss functions, and the usage of pre-trained networks
Source: Based on overview of SOTA AD models [29]

Model taxonomy	Method	Loss function	Pre-trained
Student–teacher	Reverse distillation	Cosine similarity	ResNet
Student–teacher	STFPM	MSE	ResNet
Reconstruction	DRAEM	SSIM and Focal loss	–
Distribution map	FastFlow	Log-likelihood	ResNet
Distribution map	CFlow-AD	Log-likelihood	ResNet
Memory bank	PaDiM	–	CNN
Memory bank	PatchCore	–	ResNet
Memory bank	CFA	SVDD	ResNet

Most of the models rely on pre-trained neural networks (CNN-based). Student-teacher models generally compare the result of the teacher and student model with the loss function for learning. Reconstruction models do not use pre-trained models for feature extraction. Distribution maps-based models rely on statistical concepts such as log-likelihood to fit the distributions to the data. Memory bank models mostly use distance-based algorithms as a replacement for loss functions

by the student and the teacher network. Example methods of student-teacher taxonomy in AD include Reverse distillation [12], and Student Teacher Feature Pyramid Matching (STFPM) [51]. Reverse distillation introduces one class bottleneck embedding (OCBE) as the input for the student network instead of raw images. The different architecture between the student and the teacher network helps improve anomaly detection. STFPM is an extension of the vanilla student-teacher network with feature pyramid matching (hierarchical structure). Several bottom layer groups of the model are used as feature comparisons between the student and teacher model, the differences between the feature generated are considered to be anomalies.

Reconstruction-based models, unlike student-teacher approaches, do not rely on robust pre-trained models. Reconstruction-based models have encoders and decoders in the reconstruction model that are self-trained to replicate the original normal images. Anomalies are reconstructed images that deviate from the original normal images based on a prediction from the comparison model. They perform worse in image-level anomaly detection as they cannot extract high-level semantic features; however, they excel in pixel-level anomaly detection [29]. An example of a reconstruction-based model for AD is DRAEM [56]. DRAEM is composed of a reconstructive (encoder–decoder architecture) and a discriminative sub-networks (U-Net-like architecture). The reconstruction sub-network is trained to reconstruct the original image from an artificially corrupted version by a simulator (generated by a Perlin noise generator). The output of the reconstructive sub-network is concatenated with the original image and used as input for the discriminative sub-network. The appearance of the reconstructed and the original image will differ significantly in anomalous cases.

Distribution maps are anomaly detection approaches that require a robust pre-trained network to extract features from normal images. The extracted features are mapped to a distribution usually a Gaussian distribution. Anomalies will have features that deviate from the normal feature Gaussian distribution. Examples of distribution maps AD models are FastFlow [55] and CFlow-AD [17]. FastFlow consists of a feature extractor (either Vision Transformers (ViT) or ResNet) and the FastFlow model. It is inspired by Normalizing Flow (NF) [38] techniques. During the inference, anomalous data should be out of distribution and have a lower likelihood than normal images. CFlow-AD consists of a discriminative pre-trained encoder and multi-scale generative decoders for estimating the likelihood of the encoded features. Features extracted by the encoder are processed by sets of decoders that are independent for each *k*th scale. The decoder is a conditional normalizing flow network with feature input and conditional input.

Memory banks are anomaly detection approaches that require robust pre-trained networks for feature extraction from normal images. Features extracted are stored in a memory bank and used during the inference step. Memory bank approaches require large memory space to store the features and require less training time. Anomalies are determined when the features are distanced far away from the normal features extracted from the memory bank. Examples of memory banks AD models are PaDiM [11], PatchCore [40], and CFA [27]. PaDiM consists of a pre-trained CNN for feature extraction of the patches from different semantic levels. The features are embedded and the assumption for the normal training image distribution is based on a multivariate Gaussian distribution. Mahalanobis distances are used as a comparison of whether a particular patch is anomalous, where high scores indicate anomalous areas. PatchCore uses pre-trained networks on ImageNet dataset such as ResNet and WideResNet for feature extraction. Features extracted are stored in a memory bank which is later subsampled using greedy coreset selection. A sample is considered anomalous when the nearest neighbor search of the features is far from neighboring samples. CFA uses pre-trained ResNet to extract features, features are interpolated and concatenated from different spatial resolutions to form patch features. Extracted features are stored in the memory bank and nearest neighbor searches are initialized to differentiate the normal and abnormal features. Features that are outside of the hypersphere of memorized and normal features are considered anomalies.

To choose a proper AD model, we first looked at the performance of the models on the MVTec anomaly detection dataset (MVTec AD) [4] which was generated for benchmarking anomaly detection methods with a focus on industrial inspection. The experiments are implemented using the Anomalib Python library [1] for uniform implementation of the models. From the Anomalib library, eight different anomaly detection models are trained using default hyperparameters from the library. Default hyperparameters are changed when there are limitations such as limited GPU VRAM for training a particular model. All models are implemented with early stopping with the AUROC as the main evaluation metric. All models are trained with two datasets (all SLC with background removal and without background removal) as explained in Sect. 4. Based on the result of the training of the anomaly detection models, comparisons of different metrics between the trained models are made (further explained in Sect. 6.2). The top two models that performed best based on the metrics were further explored for hyperparameter tuning. The hyperparameter that was chosen and tuned will be explained in Sect. 6.3.2. Note that the chosen best models can have different hyperparameters from the other models.

6.2 Experimental setup

The experimental setup covers the hardware used for the whole experiment and the metrics used for choosing the two best models based on the results of the explorative analysis of the SOTA anomaly detection models.

The hardware used for the preprocessing of the data and the training of the models for the experimental setup is a desktop PC with the following specifications: CPU AMD Ryzen Threadripper 3970X 32 Core Processor, NVIDIA GeForce RTX 3090 24 GB VRAM, 72 GB RAM. Codes are written in Python. Python libraries used are Anomalib (v 0.7.0), mmpretrain for ConvNeXt (classification model) [10]. The NVIDIA CUDA Toolkit version is 12.2.

The anomaly detection methods are compared based on quantitative analysis and qualitative analysis. The quantitative analysis of the models is based on the Area Under the Receiver Operating Characteristic (AUROC), F1 score, precision, recall, and accuracy. AUROC is the area under the curve of ROC which plots the true positive rate against the false positive rate at various threshold settings. AUROC with a value of 0.5 indicates a classifier with no discriminative power, equivalent to random guessing, while an AUROC value of 1.0 corresponds to a perfect classifier. Models with the top two highest AUROC in the initial explorative analysis comparison were chosen for hyperparameter fine-tuning. The qualitative analysis is implemented when comparing the top two anomaly detection methods to see how the anomaly detection models can highlight the anomaly in a given image. Models that have more precise highlights of the anomaly parts during the qualitative analysis are considered better compared to the other models.

6.3 Experimental results

The experimental results will be mainly divided into three parts: results of explorative analysis SOTA anomaly detection models (Sect. 6.3.1), hyperparameter tuning of top candidates on sampled SLC dataset (Sect. 6.3.2), and results for tuned hyperparameter on whole SLC dataset (Sect. 6.3.3).

6.3.1 Results of explorative analysis SOTA anomaly detection models

Table 4 shows the result of the explorative analysis of SOTA anomaly detection models for the whole SLC dataset. Methods with '(nob)' are datasets that are preprocessed with a background removal pipeline before being used for training. The training time given here is in hh:mm:ss. The Reverse Distillation method is abbreviated to 'RD'. Methods such as PatchCore and PaDiM are memory bank taxonomy models that only need 1 epoch for feature extraction. Based on the AUROC metric, PatchCore performs best (0.786) followed by DRAEM (0.773) for the original dataset. For the dataset with the background removal pipeline, the best performers are PatchCore (0.781) followed by RD (0.726). The F1 score metric also reflects similar results to the AUROC metric for the best-performing models. Based on this explorative analysis of the AD models, the experiments without the background removal pipeline perform consistently better. The next training will be conducted without the background removal pipeline.

6.3.2 Hyperparameter tuning of top candidates on sampled SLC dataset

Quantitative analysis of top candidates on sampled SLC dataset For the sake of time efficiency, hyperparameter tuning of the best-chosen models is done using only the sampled SLC dataset. All of the hyperparameter tunings have global seed 4 to ensure reproducibility.

Hyperparameter tuning for PatchCore is conducted by changing the backbone of the feature extractor (ResNet-18 or WideResNet-50), the layers configuration, the Limit Train Batches, the coreset subsampling percentage, and the number of neighbors. The layer configurations can be chosen between layer 2, layer 3, or layer 2+3. The Limit Train Batches size can be lowered to account for limited VRAM. The possible choices include 0.2, 0.4, 0.8, or 1.0. Using Limit Train Batches size as 1.0 means maintaining the whole training dataset during the training. Table 5 shows the result of the hyperparameter tuning of the PatchCore model.

Table 4 Explorative analysis of SOTA AD models for the whole SLC dataset

Method	AUROC	F1 score	Recall	Precision	Accuracy	Training time	Trainable parameters (M)
RD	0.710	0.647	0.757	0.565	0.657	3:19:59	80.6
RD (nob)	<u>0.726</u>	<u>0.633</u>	0.829	0.512	0.602	2:46:54	80.6
STFPM	0.699	0.618	0.736	0.533	0.624	5:57:35	49.7
STFPM (nob)	0.697	0.611	0.713	0.535	0.624	5:53:40	49.7
DRAEM	<u>0.773</u>	<u>0.658</u>	0.728	0.601	0.687	35:21:34	97.4
DRAEM (nob)	0.663	0.591	0.782	0.478	0.551	21:54:06	97.4
FastFlow	0.698	0.634	0.885	0.499	0.584	3:50:55	124
FastFlow (nob)	0.617	0.592	0.973	0.425	0.444	3:43:44	124
Cflow-AD	0.575	0.584	0.974	0.417	0.425	11:28:10	236
Cflow-AD (nob)	0.610	0.592	0.998	0.421	0.431	5:50:13	236
PaDiM	0.634	0.606	0.914	0.454	0.509	0:20:39	24.9
PaDiM (nob)	0.566	0.587	0.942	0.426	0.451	0:18:17	24.9
PatchCore	**0.786**	**0.678**	0.810	0.583	0.681	1:18:07	4.1
PatchCore (nob)	**0.781**	**0.674**	0.798	0.584	0.680	1:17:20	4.1
CFA	0.536	0.585	0.964	0.420	0.435	0:34:11	31.3
CFA (nob)	0.504	0.588	0.964	0.423	0.441	0:33:20	31.3

(nob) means that the models are subjected to a background removal pipeline. AUROC is used as the main evaluation for the models. Best performing models based on AUROC metric are given in **bold** followed by <u>underline</u>. Original and (nob) has its ranking based on the AUROC metric. The F1 score metric also depicts similar results to the AUROC metric for the best-performing models. For the whole dataset, the models with the background removal pipeline consistently perform worse compared to the original dataset. Training time is given as hh:mm:ss (hours, minutes, seconds). Note that some models may only need one epoch to train

Table 5 The result of hyperparameter tuning for PatchCore model on sampled SLC dataset

PatchCore							
Hyperparameter			Metrics				
Backbone	Layers/ C/N	LTB	AUROC	F1 score	Accuracy	Training time (hh:mm:ss)	Trainable parameters (M)
ResNet-18	3/0.1/9	1.0	0.747	0.832	0.729	00:18:09	2.8
WideResNet-50	3/0.1/9	1.0	0.772	0.833	0.742	00:18:13	24.9
WideResNet-50	2/0.1/9	0.8	<u>0.808</u>	0.852	0.769	01:37:12	4.1
WideResNet-50	2/0.1/12	0.8	<u>0.808</u>	0.852	0.769	01:37:15	4.1
WideResNet-50	2/0.1/6	0.8	0.807	0.850	0.761	01:37:12	4.1
WideResNet-50	2/0.1/3	0.8	0.793	0.843	0.759	01:37:13	4.1
WideResNet-50	2/0.05/9	0.8	**0.811**	0.851	0.762	00:53:38	4.1
WideResNet-50	2/0.01/9	0.8	0.790	0.843	0.757	00:18:50	4.1
WideResNet-50	2+3/0.1/9	0.4	0.748	0.830	0.726	00:32:09	24.9

Due to hardware limitations (GPU VRAM), some of the models can only be trained on lower Limit Train Batches to avoid out-of-memory errors. For PatchCore models, the WideResNet-50 backbone performs better compared to the ResNet-18 backbone. Best performing models based on the metrics are marked with **bold** followed by <u>underline</u>. C in the hyperparameter refers to the Coreset and N refers to the number of neighbors. LTB refers to the Limit Train Batches. hh:mm:ss means the time in hours, minutes, and seconds

Based on the results of the hyperparameter tuning for PatchCore, choosing the WideResNet-50 backbone has a slight improvement in the metrics compared to the ResNet-18 backbone. Choosing only layer 3 in the layer hyperparameter has worse metrics compared to the other models. However, it is significantly faster in training time which can be useful for rapid model training or prototyping. Increasing the neighbors above 9 does not increase performance. Decreasing the number of neighbors has a slight decrease in performance while still maintaining approximately the same training time. Choosing a smaller coreset subsampling percentage sacrifices a little performance but significantly improves the training time (coreset 0.05 to 0.01).

Table 6 shows the result of the hyperparameter tuning of the DRAEM model. The experiments are conducted by changing the lambda value, the learning rate, and the batch size. Lambda values are called loss balancing hyperparameter that controls how much of the loss SSIM (Structural Similarity Index Measure) between neighboring patches affects the loss function.

Qualitative analysis of top selected models on sampled SLC dataset Figures 2 and 3 show the qualitative analysis of the two tuned models from PatchCore and DRAEM based on the AUROC metrics on different types of defects. The left image is the original image with the respective defects. Defects included are foreign objects, stickers, dirt on the surface of the SLCs, oily surfaces, breakage, and

Table 6 The result of the hyperparameter tuning for DRAEM model on the sampled SLC dataset

DRAEM							
Hyperparameter			Metrics				
Lambda	Learning rate	Batch size	AUROC	F1 score	Accuracy	Training time (hh:mm:ss)	Trainable parameters (M)
None	0.0001	4	0.717	0.830	0.723	07:49:01	97.4
None	0.0001	16	0.732	0.832	0.728	10:55:33	97.4
0.0	0.0001	8	<u>0.742</u>	0.838	0.743	09:12:07	83.0
0.5	0.0001	8	0.730	0.814	0.723	10:04:10	83.0
1.0	0.0001	8	0.729	0.832	0.729	09:50:44	83.0
None	0.0001	8	**0.748**	0.834	0.739	05:58:27	97.4
None	0.001	8	0.702	0.832	0.733	05:47:19	97.4
None	0.01	8	0.664	0.828	0.720	05:58:42	97.4

The optimal learning rate is 0.0001 with batch sizes 8. The average time per epoch is around 3 min 30 s, which is uniform throughout the different hyperparameter settings. Best performing models based on the metrics are marked with **bold** followed by <u>underline</u>. hh:mm:ss means the time in hours, minutes, and seconds

liquid spillage. Every two columns are the predicted heat map from the model and the confidence level of whether the SLC is anomalous or normal. A higher anomaly score in a particular spot of the SLC has darker red spots, and a lower anomaly score has lighter yellow colors. PatchCore models generally perform best for most of the anomaly types. It can highlight the location in which the anomaly is most likely located in Figs. 2 and 3 compared to DRAEM tuned models. PatchCore layer 2 generally has tighter groupings of the anomalous region as shown in the foreign object (big) anomaly. PatchCore layer 3 has much-spread groupings of anomalous regions. PatchCore performs slightly worse in oily surface defects. DRAEM models work best on surface anomalies such as dirt on the surface and oil on the surface. Heatmaps from DRAEM perform best on surface anomalies and do not highlight the breakage of the SLC walls as seen in breakage, breakage and liquid, and major breakage in Fig. 3. Despite having high confidence in predicting major breakage, the heatmap does not give a proper visualization of which part of the SLC is anomalous. This is however expected as DRAEM models are built for pixel-level detection. Based on the qualitative experiment, the PatchCore model performs consistently better compared to the other models and has a better-predicted heat map on the localization of the defects.

6.3.3 Results for tuned hyperparameters on whole SLC dataset

Based on the quantitative and qualitative analyses conducted during the hyperparameter tuning for the sampled SLC dataset, we use the best combination of hyperparameters to train it on the whole SLC dataset without background removal. The total number of images in the whole SLC dataset is twice as high as the sampled SLC dataset. This prevents us from directly using the same hyperparameters from the hyperparameter tuning due to GPU VRAM limitations. Changes in specific parameters will be specified during the quantitative and qualitative analysis of the models. Table 7 shows the quantitative comparison of the hyperparameter-tuned models with the default parameter models. The hyperparameter-tuned PatchCore model performs better and is faster than the default hyperparameter PatchCore. The hyperparameter-tuned PatchCore may perform better, if there is no limitation to the GPU VRAM, as the hyperparameter used here is the second model from Table 5 with lower Limit Train Batches (0.7). When higher GPU VRAM is available, using the Patchcore layer 2 models with 1.0 Limit Train Batches may increase the performance of the model. The hyperparameter-tuned DRAEM performs better compared to the default parameter DRAEM. The training time needed to train the tuned model is shorter. The inference time for the model to do inferencing on the inference dataset (265 images of normal and anomalous SLCs) is on average the same.

Figures 4 and 5 show the qualitative analysis and comparison between the hyperparameter-tuned whole SLC dataset models and the default parameter models. Figure 4 shows the first three defects on different colors and types of SLCs. The first two columns are from breakage on the top left side of the blue SLCs. The next two columns show a yellow SLC with a breakage defect on the top left corner walls. This represents one of the most challenging cases, as the defect is difficult to detect for the naked human eyes. The final two columns are one of the simpler cases of defects (a large hole in the center of the SLC). Patchcore hyperparameter-tuned models are more confident in the larger defects with red heat map highlights on the location of the defect. However, the PatchCore model (both default and hyperparameter-tuned) still struggles to accurately classify the yellow SLCs as defects and highlight the location of the defects. DRAEM models have high anomaly scores on obvious defects such as the dark blue SLC defects in Fig. 4. However, the highlighted anomaly locations are not

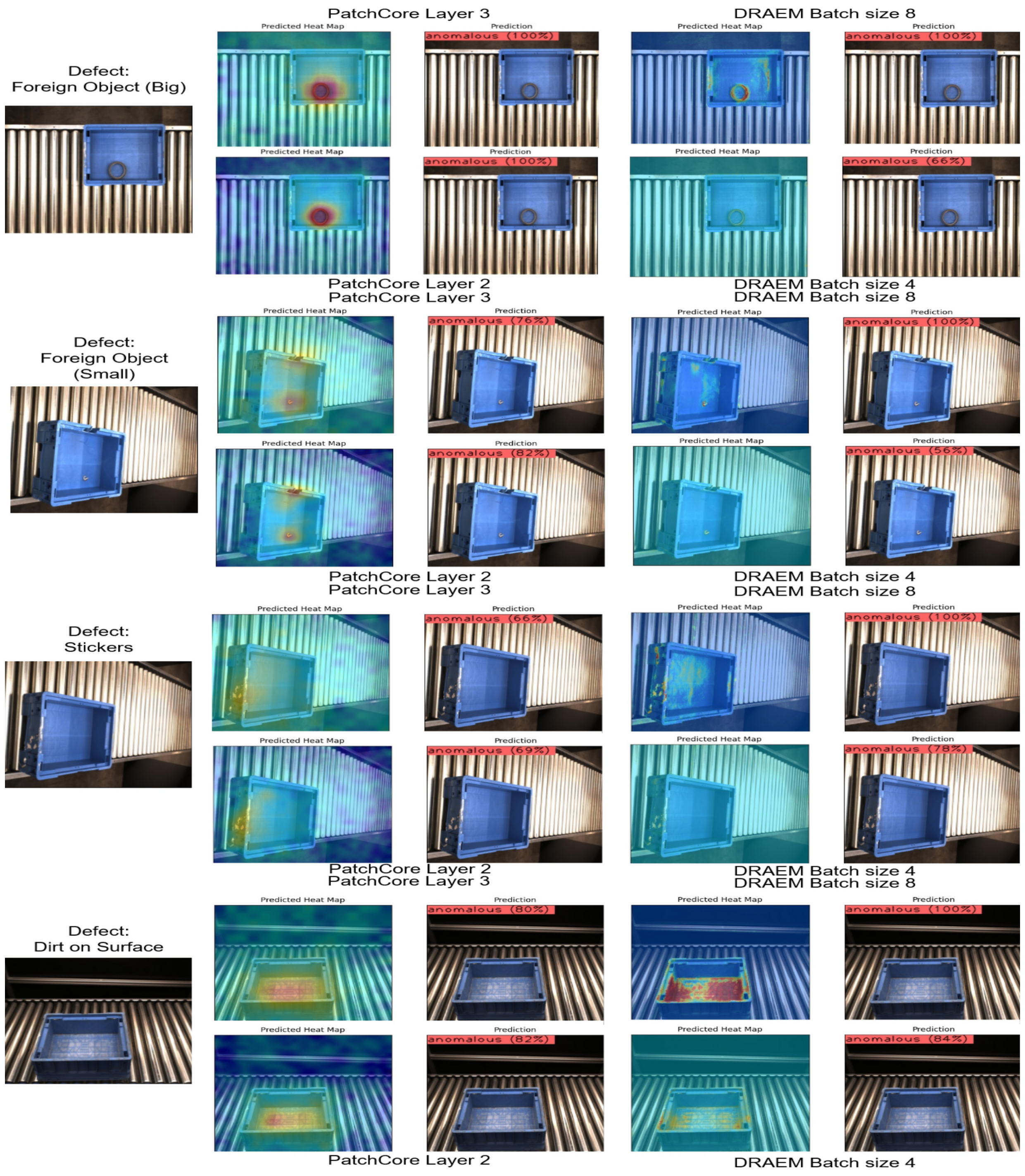

Fig. 2 The qualitative analysis of the two tuned models from PatchCore and DRAEM models is based on the explorative analysis of SOTA anomaly detection. Heatmaps from PatchCore better isolate the anomalies compared to the DRAEM models. Patchcore Layer 2 produces heatmaps that are well-localized at the true anomalous areas. Patchcore Layer 3 produces more spread-out anomaly scores near the anomalous areas. Layer 2 also has darker red spots which means a high anomaly score which indicates a high likelihood of anomaly in that particular region. DRAEM performs best on surface-based anomalies such as dirt on the SLC surface. DRAEM with batch size 8 performs better compared to batch size 4 in most of the cases

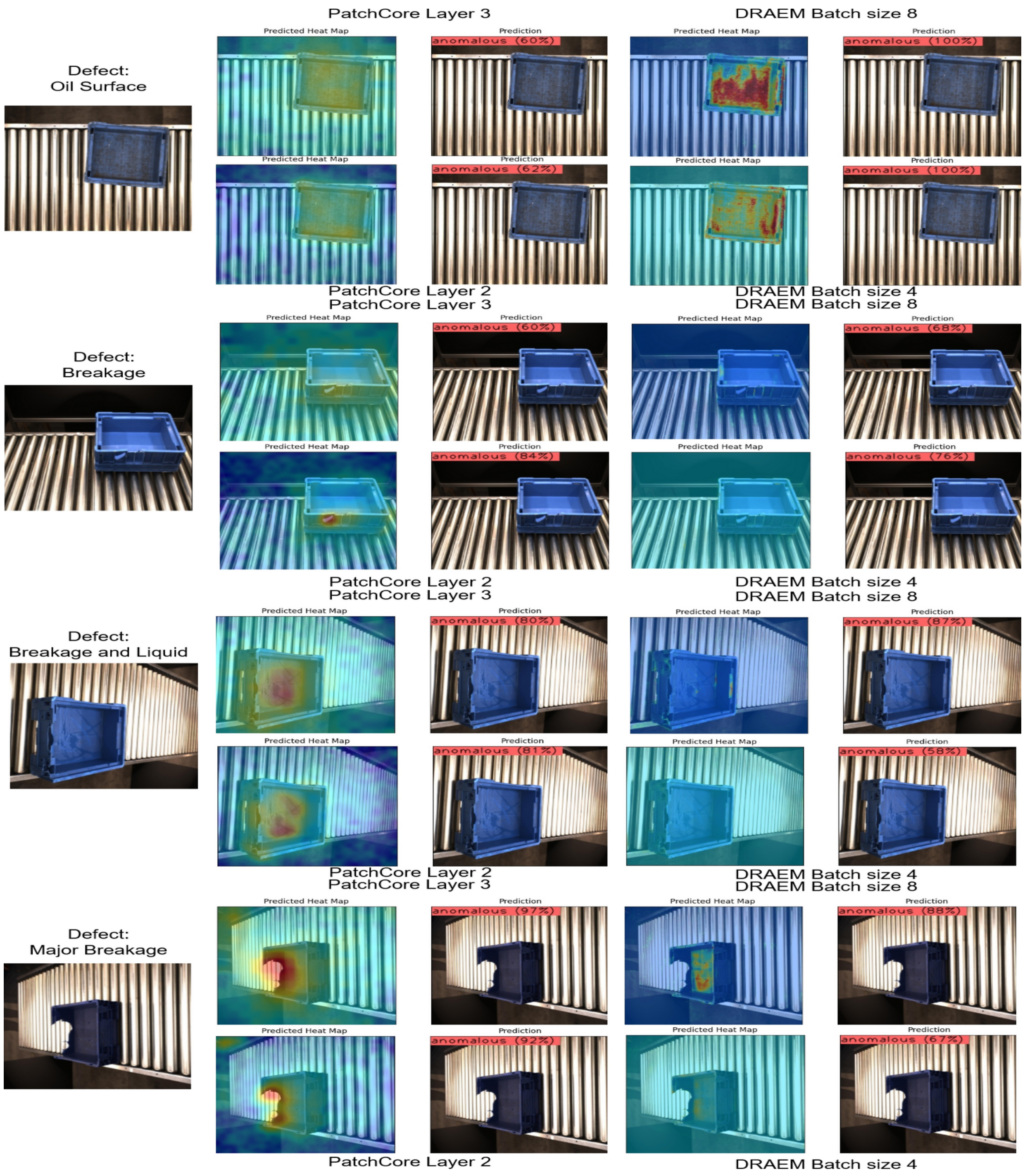

Fig. 3 Qualitative analysis of the two tuned models from PatchCore and DRAEM models. This covers the rest of the anomaly types that are not covered in Fig. 2. Despite the high confidence of the DRAEM model on major breakage and breakage anomalies, the model fails to highlight the area of the breakage anomaly properly. Patchcore layer 2 consistently highlights the main point of anomaly as in breakage defect and major breakage SLCs

properly shown or misguiding, especially in the dark blue SLC. This may be unfavorable in industrial settings as the result of the defect classification cannot be backed with a proper indication of the location of the defects. The tuned DRAEM model misclassifies the yellow SLC into a normal class. Figure 5 shows additional qualitative analysis of the models on another three different colored and shaped SLCs. The first two columns show a blue cover SLC with a cracks on the upper left part of the cover SLC. All models except the DRAEM default model identify the cover SLC

Table 7 Quantitative comparison between the default (noted by def on the end of the model name) and hyperparameter-tuned models on the whole SLC dataset

Comparison default and best hyperparameter-tuned models on whole SLC dataset							
Model	AUROC	F1 Score	Recall	Precision	Accuracy	Training time (hh:mm:ss)	Inference time (mm:ss)
PatchCore	0.812	0.700	0.736	0.672	0.741	01:02:27	01:22
PatchCore (def)	0.786	0.678	0.810	0.583	0.681	01:18:07	01:24
DRAEM	0.789	0.672	0.886	0.541	0.642	28:57:19	01:24
DRAEM (def)	0.773	0.658	0.728	0.601	0.687	35:21:34	01:23

Metrics are given as AUROC, F1 score, recall, precision, accuracy, training time, and inference time. The training time is given as the total training time. Whereas the inference time is the total inference time needed to process the inference SLC dataset (which contains 265 images of normal and anomalous SLCs). The hyperparameter-tuned PatchCore and DRAEM performed better based on the AUROC metric. PatchCore model is not performing to its maximum potential due to limitations in the VRAM. hh:mm:ss means the time in hours, minutes, and seconds

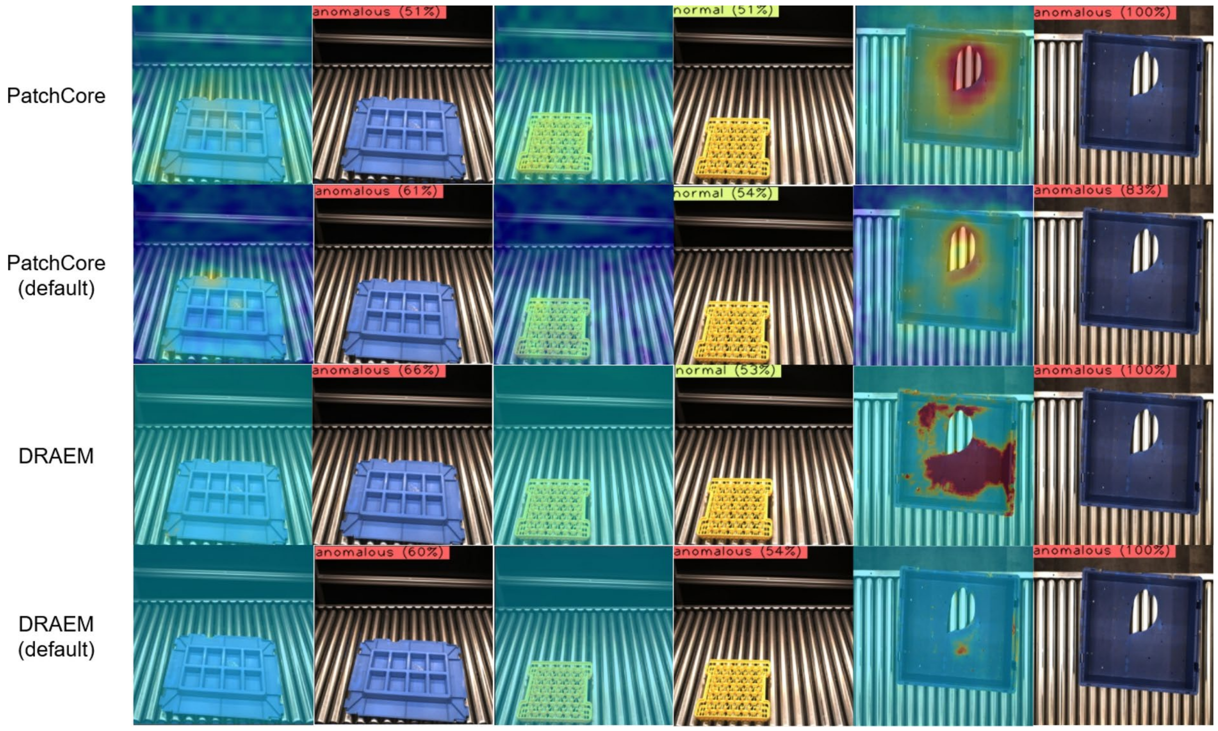

Fig. 4 The qualitative analysis and comparison of the two best models based on the default parameter and the hyperparameter-tuned models on the whole SLC dataset. Patchcore on both default and hyperparameter-tuned models performs better in highlighting the position of the anomaly compared to DRAEM (as seen in the left and right column heatmap). DRAEM models can mostly classify the SLC as a defect but fail to properly highlight the defect location in the heat maps. The blue SLC cover (left column) has breakage on the top left side of the image. All models can accurately classify the image as anomalous. However, only the default PatchCore can properly pinpoint the anomalous spot. The dark blue SLC (right column) are simpler defect case where breakage on the surface of the SLC is easily observed with the naked human eye. Despite the high confidence score of the DRAEM models, the highlighted regions do not represent the anomalous part of the SLC. The yellow SLC (center) exhibits a more complex structure which makes defect detection considerably more difficult. The defect for the yellow SLC is on the top left corner walls of the SLC (breakage). Both Patchcore and DRAEM fail to identify the anomalous part in the yellow SLC

successfully as a defect class. The difference between the Patchcore default (layer 2) and tuned Patchcore (layer 3) can be seen in the blue cover SLC, where Patchcore with layer 2 has a higher anomaly score in the region of the anomaly affecting the confidence in the prediction. The next two columns show a black SLC with a deformed left outer wall of the SLC. All models classify properly that the black SLC belongs to the defect class. None of the model have successfully highlighted the location of the deformed left outer wall. The final two columns show a breakage on the top right corner of the yellow SLC and the part broken lies on the center of the SLC. This is another harder defect case where the defect location is not as obvious. Both PatchCore models successfully highlight the location of the breakage and the location of the foreign object with the default having better heatmap result due to the layer 2 hyperparameter. The default DRAEM model fails to highlight anything significant, thus the misclassification of the yellow SLC. The

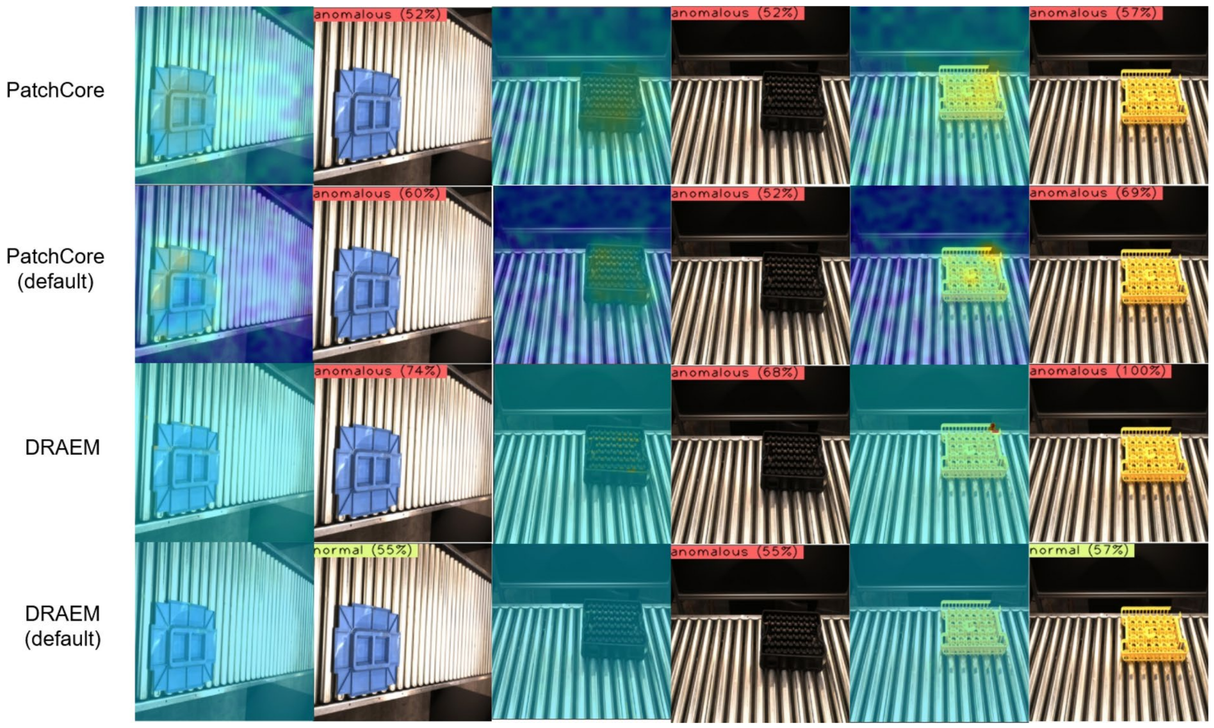

Fig. 5 Additional qualitative analysis and comparison of the two best models based on the default parameter and the hyperparameter-tuned models on the whole SLC dataset. The blue SLC cover (left column) contains cracks on the upper left part of the SLC cover. The default DRAEM model failed to classify this as anomalous. Both PatchCore models can classify this as anomalous and have some highlights in the region of the cracks. The black SLC (center column) has a defect (deformed plastic) in the left outer wall of the SLC. All of the models successfully classify the SLC as defective. However, none of the models properly highlights the defective left wall. The yellow SLC (right column) contains two defects. One breakage defect is on the top right corner of the SLC and the other defect is a foreign object in the center of the SLC. PatchCore models perform consistently better compared to other models with good highlighting of the location of the defect. The default DRAEM model produced a miss classification in the yellow SLC case as it fails to detect the foreign object and the breakage on the upper right walls. We can see the weakness of the hypertuned PatchCore model where the layers used in the hyperparameter (layer 3) have worse performance in highlighting the defect in the yellow SLC

tuned DRAEM model highlights the breakage on the wall in dark red, however, fails to highlight the foreign object in the center of the yellow SLC. Based on all the qualitative analysis, PatchCore models perform more consistently compared to DRAEM model and can more reliably highlight the exact location of the defects.

7 Discussion and conclusion

The goal of this work was to create a computer vision system that can recognize the SLC type for inventory management and perform defect detection automatically bridging the gap between research and the industrial field for anomaly detection systems. The camera portal was built based on careful consideration of the geometry of the SLCs while keeping space requirements and costs minimum. The whole setup consists of the camera portal with 5 cameras angled at approximately 47.5° with a placement height of 500 mm from the edge of the highest SLC and a PLC-controlled conveyor belt.

This work implemented SOTA classification and anomaly detection models, which are commonly used for benchmarking on controlled datasets such as MVTec Dataset, on real SLC datasets.

The accuracy of the finetuned classification model ConvNeXt is 100% for the SLC test data set. This exceptionally high accuracy can be attributed to the fact that the SLCs are standardized products, with a set of easily distinguishable features such as color, size, and geometry. So the classification of the SLC type can be considered as a solved problem.

The anomaly detection is considerably more challenging than the SLC-type classification. The top two performers using the default hyperparameter in the explorative experiment step for anomaly detection are PatchCore and DRAEM. Despite the high metric score of DRAEM, it performs worse in image-level anomaly detection compared to PatchCore qualitatively. PatchCore performs better in localizing the defects in the heatmaps which helps to explain why an object is classified as defect. Besides this, PatchCore has consistently lower training times compared to DRAEM. Uncertainties during the implementation of the classification and the anomaly detection model in real-life production line applications include the stability of the code containing the classification and anomaly detection model. As a work in progress, the proposed computer vision system is tested in a longer continuous run.

Limitations of the proposed study can be divided into theoretical and practical limitations. We identify three theoretical limitations such as the implementation of the background removal algorithm, the exclusion of pixel-level anomalies, and the exclusion of potential dependencies between images. The background removal used in our study is U^2-Net which might not be enough for our case of background removal. The exclusion of pixel-level anomalies hinders the performance of some models such as DRAEM. We have further identified the practical limitations in our study. This includes the training method done locally with the RTX 3090 which has limited VRAM. Data collection for our dataset are currently done in a controlled manner (no abnormal conditions such as sudden change in lighting conditions or wetness of the SLC surface) which may affect the performance of the classification and AD models. The current dataset separates the SLC as defect or normal without the detail of what and where the anomalies are present in the image. This can be solved with future annotations of the dataset. The final limitation of this study is the possibility of slower real on-site inference time that can be affected by external factors.

The limitations in our study provide opportunities for future work. A potential novel implementation may combine the AD and classification into one model. Besides this, a more detailed annotation on the location of the anomalies in the images could help improve the model training. VRAM limitations during the training can be tackled in two ways. First, the training can be done on clusters with higher VRAM. Second, it can be considered a practical solution where the anomaly detection model part is trained for each particular type of SLC. This means that the whole dataset for that particular SLC can be trained without the VRAM limitation as that particular SLC type is only a subset of the whole dataset. However, this adds another layer of operations during inferencing in real-life implementation as the system needs to know which SLC is detected by the camera and then do the inferencing based on that particular SLC type. Potential problems may include the inference time for the anomaly detection model must wait for the classification model's output, which can affect production cycle time. Another problem is during the inference of the classification, if the output is wrong, then the queried model for the anomaly detection will also be the wrong model. The current practical implementation of the inference is done per image. Each SLC will have exactly five images from the five cameras. SLCs are considered defective if one of these views is considered defective. This method has its limitations and consideration of future work in multi-view AD methods can be an interesting direction. Techniques such as image stacking, anomaly score aggregation, or multi-view convolutional models could be explored to enhance performance and provide a more accurate analysis of the SLCs.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00138-024-01636-x.

Acknowledgements Supported by DIBCO funded by the research program information and communication techniques of the Bavarian Ministry of Economic Affairs, Regional Development and Energy (DIK-2105-0044/DIK0264)). Supported by the publication fund of the Technical University of Applied Sciences Würzburg-Schweinfurt.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availibility The image data utilized in this study was collected at a facility dedicated to reprocessing SLCs. Due to the presence of sensitive customer credentials in some of the SLCs, only a selected portion of the data can be shared publicly. The published data set contains 18 types of SLCs with a total of 9260 images, and can be accessed via Hugging Face [50]. Additional details are provided in the Supplementary materials.

References

1. Akcay, S., Ameln, D., Vaidya, A., et al.: Anomalib: A deep learning library for anomaly detection. In: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, pp 1706–1710 (2022)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. ArXiv e-prints arXiv:1607.06450 [stat] (2016)
3. Bai, D., Li, G., Jiang, D., et al.: Surface defect detection methods for industrial products with imbalanced samples: a review of progress in the 2020s. Eng. Appl. Artif. Intell. **130**, 107697 (2024). https://doi.org/10.1016/j.engappai.2023.107697
4. Bergmann, P., Fauser, M., Sattlegger, D., et al.: Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, CA, USA, p 9584-9592, (201) https://doi.org/10.1109/CVPR.2019.00982, https://ieeexplore.ieee.org/document/8954181/

5. Bertagnolli, F.: Lean management: introduction and in-depth study of japanese management philosophy. Springer Fachmedien Wiesbaden, Wiesbaden, (2022) https://doi.org/10.1007/978-3-658-36087-0
6. Bohm, M., Ziegenbein, A., Metternich, J.: Bildanalyse zur handhabung von kleinladungsträgern mithilfe künstlicher neuronaler netzwerke. Zeitschrift für wirtschaftlichen Fabrikbetrieb **115**(7–8), 513–516 (2020). https://doi.org/10.3139/104.112296
7. Bykova, M., Ostermann, S., Tjaden, B.: Detecting network intrusions via a statistical analysis of network packet characteristics. In: Proceedings of the 33rd Southeastern Symposium on System Theory, IEEE, pp 309–314 (2001) https://doi.org/10.1109/SSST.2001.918537
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 1–58 (2009)
9. Coelho, P.M., Corona, B., Klooster, R.T., et al.: Sustainability of reusable packaging-current situation and trends. Resour. Conserv. Recycl. **6**, 100037 (2020). https://doi.org/10.1016/j.rcrx.2020.100037
10. Contributors, M.: Openmmlab's pre-training toolbox and benchmark. https://github.com/open-mmlab/mmpretrain (2023)
11. Defard, T., Setkov, A., Loesch, A., et al.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition, Springer, pp 475–489 (2021) https://doi.org/10.1007/978-3-030-68799-1_35
12. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9737–9746 (2022)
13. Deng, J., Dong, W., Socher, R., et al.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Miami, FL, p 248-255 (2009) https://doi.org/10.1109/CVPR.2009.5206848, https://ieeexplore.ieee.org/document/5206848/
14. Dlamini, S., Kao, C.Y., Su, S.L., et al.: Development of a real-time machine vision system for functional textile fabric defect detection using a deep yolov4 model. Textile Res. J. **92**(5–6), 675–690 (2022). https://doi.org/10.1177/00405175211034241
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint https://doi.org/10.48550/arXiv.2010.11929, arXiv:2010.11929 (2021)
16. Dror, I.E.: Cognitive and human factors in expert decision making: six fallacies and the eight sources of bias. Anal. Chem. **92**(12), 7998–8004 (2020). https://doi.org/10.1021/acs.analchem.0c00704
17. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 98–107 (2022)
18. Hachem, C.E., Perrot, G., Painvin, L., et al.: Automation of quality control in the automotive industry using deep learning algorithms. In: 2021 International Conference on Computer, Control and Robotics (ICCCR). IEEE, Shanghai, China, p 123-127 (2021) https://doi.org/10.1109/ICCCR49711.2021.9349273, https://ieeexplore.ieee.org/document/9349273/
19. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778 (2016)
20. He, Y., Song, K., Meng, Q., et al.: An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE Trans. Instrum. Meas. **69**(4), 1493–1504 (2020). https://doi.org/10.1109/TIM.2019.2915404
21. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 [cs] (2023)
22. Hofmann, E., Bachmann, H.: Behälter-Management in der Praxis: State-of-the-Art und Entwicklungstendenzen bei der Steuerung von Ladungsträgerkreisläufen, p 55 (2013)
23. Holm, D.M., Fottner, J.: A concept for camera-based classification of load carriers. ESSN: 2701-6277 (2021) https://doi.org/10.15488/11268, https://www.repo.uni-hannover.de/handle/123456789/11355
24. Howard, A.G., Zhu, M., Chen, B., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 [cs] (2017)
25. Huang, C., Jiang, A., Feng, J., et al.: Adapting visual-language models for generalizable anomaly detection in medical images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11375–11385 (2024)
26. Ioffe, S.: Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. Adv. Neural Inf. Process. Syst. 30 (2017) arXiv:1702.03275 [cs]
27. Lee, S., Lee, S., Song, B.C.: Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access **10**, 78446–78454 (2022) https://doi.org/10.1109/ACCESS.2022.3193699
28. Liang, Q., Zhu, W., Sun, W., et al.: In-line inspection solution for codes on complex backgrounds for the plastic container industry. Measurement **148**, 106965 (2019). https://doi.org/10.1016/j.measurement.2019.106965
29. Liu, J., Xie, G., Wang, J., et al.: Deep industrial image anomaly detection: A survey. Mach. Intell. Res. 21(1):104-135(2024)., [cs] https://doi.org/10.1007/s11633-023-1459-z
30. Liu, Z., Mao, H., Wu, C.Y., et al.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11976–11986 (2022)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 [cs, math] (2019)
32. Mahmoudi, M., Parviziomran, I.: Reusable packaging in supply chains: a review of environmental and economic impacts, logistics system designs, and operations management. Int. J. Product. Econ. **228**, 107730 (2020). https://doi.org/10.1016/j.ijpe.2020.107730
33. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning, (2010) https://api.semanticscholar.org/CorpusID:15539264
34. Noceti, N., Zini, L., Odone, F.: A multi-camera system for damage and tampering detection in a postal security framework. EURASIP J. Image Video Process. (2018). https://doi.org/10.1186/s13640-017-0242-x
35. Pierer, A., Wiener, T., Gjakova, L., et al.: Zero-error-production through inline-quality control of presshardened automotive parts by multi-camera systems. IOP Conf. Series Mater. Sci. Eng. **1157**(1), 012074 (2021). https://doi.org/10.1088/1757-899X/1157/1/012074
36. Poss, C., Ibragimov, O., Indreswaran, A., et al.: Application of open source deep neural networks for object detection in industrial environments. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, Orlando, FL, p 231-236 (2018) https://doi.org/10.1109/ICMLA.2018.00041, https://ieeexplore.ieee.org/document/8614066/
37. Qin, X., Zhang, Z., Huang C., et al. U^2-net: Going deeper with nested u-structure for salient object detection. Pattern Recogn. 106, 107404 (2020), [cs]. https://doi.org/10.48550/arXiv.2005.09007
38. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International conference on machine learning, PMLR, pp 1530–1538 (2015)
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image

computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, Springer, pp 234–241 (2015)
40. Roth, K., Pemula, L., Zepeda, J., et al.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14318–14328 (2022) [cs]
41. Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., et al.: A unifying review of deep and shallow anomaly detection. Proc. IEEE **109**(5), 756-795 (2021).https://doi.org/10.1109/JPROC.2021.3052449, arXiv:2009.11732 [cs, stat]
42. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520 (2018)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015). arXiv preprint arXiv:1409.1556 [cs]
44. Singh, S.A., Desai, K.A.: Automated surface defect detection framework using machine vision and convolutional neural networks. J. Intell. Manuf. **34**(4), 1995–2011 (2023). https://doi.org/10.1007/s10845-021-01878-w
45. Sobottka, T., Sihn, W., Edtmayr, T.: Increasing the efficiency of closed loops of reusable containers in production environments concerning container cleaning. ACTA Technica corviniensis VII:101–110(2014). https://doi.org/10.24406/publica-fhg-238841, https://publica.fraunhofer.de/handle/publica/238841
46. Song, C., Huang, Y., Ouyang, W., et al.: Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, CA, USA, p 3131-3140 (2019)https://doi.org/10.1109/CVPR.2019.00325, https://ieeexplore.ieee.org/document/8953307/
47. Tao, H., Zheng, Y., Wang, Y., et al.: Enhanced feature extraction yolo industrial small object detection algorithm based on receptive-field attention and multi-scale features. Meas. Sci. Technol. **35**(10), 105023 (2024). https://doi.org/10.1088/1361-6501/ad633d
48. Telljohann, A.: Introduction to building a machine vision inspection. Handbook of machine vision pp 35–71 (2006)
49. Truong, A.M., Luong, H.Q.: A non-destructive, autoencoder-based approach to detecting defects and contamination in reusable food packaging. Curr. Res. Food Sci. **8**, 100758 (2024). https://doi.org/10.1016/j.crfs.2024.100758
50. Wahyudi, V.: public_slc_dataset (revision bd1a393). (2024) https://doi.org/10.57967/hf/3306, https://huggingface.co/datasets/vincentw1997/public_SLC_Dataset
51. Wang, G., Han, S., Ding, E., et al.: Student-teacher feature pyramid matching for anomaly detection (2021). arXiv preprint arXiv:2103.04257 [cs]
52. Wang, R., Zhuang, Z., Tao, H., et al.: Q-learning based fault estimation and fault tolerant iterative learning control for mimo systems. ISA Trans. **142**, 123–135 (2023). https://doi.org/10.1016/j.isatra.2023.07.043
53. Würschinger, H., Mühlbauer, M., Winter, M., et al.: Implementation and potentials of a machine vision system in a series production using deep learning and low-cost hardware. Procedia CIRP **90**, 611–616 (2020). https://doi.org/10.1016/j.procir.2020.01.121
54. Xu, C., Tao, D., Xu, C.: Multi-view intact space learning. IEEE Trans. Pattern Anal. Mach. Intell. **37**(12), 2531–2544 (2015). https://doi.org/10.1109/TPAMI.2015.2417578
55. Yu, J., Zheng, Y., Wang, X., et al.: Fastflow: unsupervised anomaly detection and localization via 2d normalizing flows. arXiv:2111.07677 [cs] (2021)
56. Zavrtanik, V., Kristan, M., Skočaj, D.: DRAEM - A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 8330-8339 (2021) [cs]
57. Zhang, C., Kang, F., Wang, Y.: An improved apple object detection method based on lightweight yolov4 in complex backgrounds. Remote Sens. **14**(17), 4150 (2022). https://doi.org/10.3390/rs14174150

Vincent Wahyudi received a B. Eng. from Universitas Tarumanagara, Jakarta, Indonesia in 2019 and an M. Sc. From the Technical University of Applied Sciences Würzburg-Schweinfurt (THWS), Würzburg, Germany in 2023. He is pursuing a Ph.D. degree in Computer Science at THWS. He is currently a researcher on the DIBCO Project at THWS. His research focuses on computer vision and anomaly detection.

Cedric C. Ziegler received his bachelor's degree in business and engineering from the University of Applied Sciences Würzburg-Schweinfurt (THWS) in 2021, and his Master of Science degree in industrial engineering and management from the University of Erlangen-Nuremberg in 2023. He has worked as a researcher at the ERP Lab at the University of Applied Sciences Würzburg-Schweinfurt (THWS). Since 2023, he has been a consultant for IT strategy. His research focuses on computer vision and neural networks in logistics and engineer-to-order environments.

Matthias Frieß received his B.Sc. in Industrial Mathematics from Technical University of Applied Sciences Würzburg-Schweinfurt (THWS) in 2023. He is currently working as an Industrial Data Analyst in Manufacturing. His research interests include machine learning, condition monitoring and predictive maintenance.

Stefan Schramm received his B.Sc. And M.Sc. Degrees from the Technical University of Applied Sciences Würzburg-Schweinfurt (THWS) in 2020 and the Technische Hochschule Nürnberg in 2022, respectively. His research topics were computer vision and machine learning. He is currently working for TAF Industriesysteme as an automation engineer.

Constantin Lang worked as a student assistant at Technical University of Applied Sciences Würzburg-Schweinfurt (THWS) from 2023 to 2024.

Lars Eberhardt received a Dipl. Ing. (FH) from the Technical University of Applied Sciences Würzburg Schweinfurt (THWS) and an M.A. from Technical University Dresden in 2014 and 2016, respectively. He is pursuing the Ph.D. degree in Strategic Decision Making at the University of Bamberg. He is currently a researcher at the DIBCO Projekt at THWS and the owner of a digitalisation company for laywers. His research interests include Strategic Decision Making in the context of digitalization.

Fabian Freund holds a degree in Industrial Engineering and Management, which he earned in 2010 from the University of Applied Sciences Würzburg-Schweinfurt (THWS). In March 2015, he co-founded TAF Industriesysteme GmbH in Rottendorf, where he has been serving as Managing Partner ever since.

Alexander Dobhan has been Professor of Business Processes and Business Applications since 2017 at Technical University of Applied Sciences Würzburg Schweinfurt (THWS). As head of the ERP lab and FIS-SAP lab at THWS, he primarily researches the digitalization of business processes in production and logistics. Previously, he gained practical experience in the digitalization of business processes at Fresenius Medical Care and E.ON Netz GmbH and completed his doctorate at the University of Bamberg in 2012. (Image credit: Stefan Bausewein)

Martin Storath received the Diplom degree in mathematics, the Honours degree in technology management, and the Ph.D. degree in mathematics from Technische Universität München in 2008,2009, and 2013, respectively. He has contributed as a researcher at institutions such as the Helmholtz Zentrum München, the Biomedical Imaging Group at EPFL, and the Image Analysis and Learning Group at Universität Heidelberg.