

Simplified Object Detection for Manufacturing: Introducing a Low-Resolution Dataset

Jonas Werheid ¹, Shengjie He ¹, Tobias Hamann ¹, Anas Abdelrazeq ¹,
Robert H. Schmitt ¹

1. Chair of Intelligence in Quality Sensing (WZL-IQS), RWTH Aachen University, Aachen.

**Date Submitted:**

2024-03-22

Date Received:

2024-04-04

Date Accepted:

2024-12-02

Date Published:

2025-01-20

DOI:

doi.org/10.48694/inggrid.4133

Reviewers:

2 anonymous reviewers

License:

This work is licensed under [CC BY 4.0](#) 

Keywords:

Dataset, Computer Vision, Object Detection, Quality Classification, Manufacturing

Data availability:

Data can be found here: <https://zenodo.org/records/10731976>

Software availability:

Software can be found here: https://git.rwth-aachen.de/zukip-ro/yolov5_for_plastic_brick_quality_classification

Corresponding Author:

Jonas Werheid
jonas.werheid@wzl-mq.rwth-aachen.de

Abstract.

Machine learning (ML), particularly within the domain of computer vision (CV), has established solutions for automated quality classification using visual data in manufacturing processes. Object detection as a CV method for quality classification provides a distinct advantage in enabling the assessment of items within the manufacturing environment, regardless of their location in images. However, substantial challenges remain regarding labeled data availability in manufacturing contexts, training examples, data imbalance, and the complexity of incorporating these methods into real-world applications. Furthermore, real-world datasets often lack adherence to FAIR principles, which limits their accessibility and interoperability, especially for small- and medium-sized enterprises (SMEs) working to integrate object detection into their manufacturing processes. In this article, we present a low-resolution 640x640 dataset based on plastic bricks for object detection, featuring two quality labels to identify minor surface defects as an example of quality classification. We analyze the dataset using a YOLOv5 model on three different dataset sizes, while accounting for class imbalance, to demonstrate the accuracy of an object detection model in a simple manufacturing use case. The mean Average Precision mAP@0.5 for correctly identifying instances in our testing dataset ranges from 0.668 to 0.774, depending on dataset size and class imbalance. While our focus is on demonstrating object detection with low-resolution images and limited data availability, the generated data and trained model also adhere to FAIR principles. Therefore, these resources are made available with proper metadata to support their reuse and further investigation into object detection tasks for similar quality classification use cases in manufacturing.

1 Introduction

Object detection and pose estimation are key capabilities in unstructured or less structured environments to enable smart manufacturing applications, such as autonomous robots or process monitoring [1]. However, these areas in computer vision (CV) including advanced machine learning (ML) techniques are still in their infancy [2]. Although research reveals a robust understanding of ML and applications, notably small- and medium-sized enterprises (SMEs) show low maturity with only 8 percent of SMEs in Germany having deployed ML technologies

in a questionnaire done in 2020 [3]. Also, a further study with 368 german SMEs revealed in 2021 that just 5.8 percent of them developed AI solutions by themselves [4]. The governmental project "Mittelstand Digital" identified insufficient data as the second most significant obstacle among nine barriers to AI adoption in SMEs. Furthermore, the preparation of best practices and examples was highlighted as the most suitable public measure among 16 factors that support SMEs in AI integration [5].

These challenges and circumstances underscore the critical necessity for open-source ML datasets and pre-trained models, serving as illustrative examples to articulate best practices and facilitate the transfer of research into the industry for SMEs to deploy ML techniques such as object detection and foster their manufacturing processes. Additionally, such open-source publications must encourage FAIR principles to ensure efficient integration and interoperability of presented best practices for SMEs and stakeholders [6].

Recent approaches introduced various object detection datasets, in diverse domains, such as for detection of industrial tubes or safety helmets in different scenarios [7],[8]. Moreover, the existing research contributes to datasets provided with a focus on object detection in the context of defect detection or quality classification of industrial goods, such as metal parts, printed circuit boards, or insulator components for electricity supply [9], [10], [11]. Also, datasets incorporating plastic bricks are available as artificial use cases [12], [13]. These serve as learning resources and provide realistic synthetic image datasets for training object detection methods in an understandable context [12].

However, the literature does not describe object detection datasets as best practices for SMEs in the context of exemplary manufacturing applications and their adherence to FAIR principles for easy reuse. Demonstrating a tangible object detection use case in manufacturing with low-resolution image data and development showcases considering limited data availability is not addressed in the literature. Exemplary model development showcases, illustrating best practices for developing algorithms of the corresponding datasets, are either not provided or lack description. Also, findability and descriptions of access licenses are not described, indicating an insufficient fulfillment of FAIR principles. For example, Digital Object Identifiers or Metadata are typically not provided within these resources. FAIRness evaluation software, such as F-UJI, evaluates the FAIRness of the cited resources with a score below 65 percent [14]. This highlights a significant gap in FAIR datasets and showcases that could offer tailored best practices for SMEs in manufacturing to foster their AI integration.

Building upon the context of research challenges and existing approaches, we develop a simple low-resolution object detection dataset based on plastic bricks with some having minor surface defects. Furthermore, we train a current ML model of the YOLO series to detect the bricks and whether they show defects. Different sizes of datasets are used to assess how performance varies depending on the availability of data. Moreover, class imbalance, a common challenge in manufacturing, is considered to highlight its impact on detection precision [15]. Our primary discovery centers around achieving good accuracy levels despite limited data availability, class imbalance, and suboptimal camera resolutions, emphasizing the critical interplay between data, resolution, and the specific use case under consideration.

We structure these by presenting the dataset and its properties first, then explaining its creation

and methods in Section 2. In Section 3, we analyze the dataset with the open-source object detection model YOLOv5 and provide a pre-trained architecture including insights and analytics of the training with varying dataset sizes and class imbalances. Hence, the data and model are published regarding FAIR principles with metadata ensuring the transferability of this publication to stakeholders, such as developers in SMEs. Finally, the contribution and its limitations will be discussed in the conclusions.

2 Dataset

The dataset encapsulates the complexities of surface defect detection with plastic toy bricks as objects. It comprises multiple plastic bricks of different colors and sizes within a single frame, that are either defective or valid. Defective bricks have indentations and deformations on the surface, aiming to resemble common surface defects in industrial manufacturing. The following section provides a comprehensive overview of the dataset, including insights into the collection methods and employed tools. Section 2.1 delves into the fundamental details and properties of the dataset, while Section 2.2 outlines the process of image collection and annotation creation.

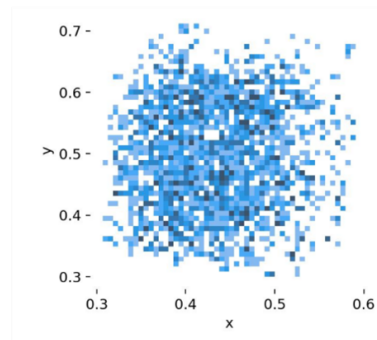
2.1 Data Description

The dataset provides images of plastic toy bricks with surface damage caused by a hammer as objects to inspect. While the bricks occur in multiple colors and sizes, the labels are provided binary with valid bricks and defective ones having damages on their surfaces. The dataset consists of 1500 images containing a total of approximately 4400 objects. Among these objects, there are roughly 2000 instances representing defects and 2400 representing valid instances. This balanced distribution of labels within the dataset serves to counteract possible biases and prevent models from learning disproportionately toward any particular class and therefore simplify the object detection task. Nevertheless, the dataset can be manipulated to introduce class imbalance by utilizing the metadata on class distribution to select a subset of the data, thereby making the task more challenging, as demonstrated in Section 3.

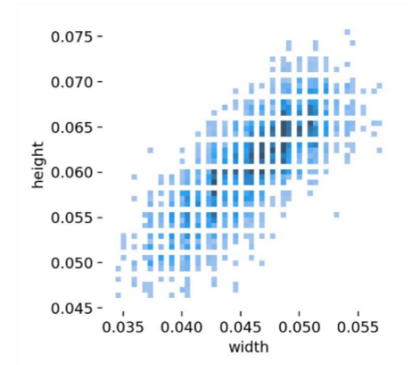
Each image has a corresponding label. Table 1 shows all information provided by a label. The coordinates x-center and y-center are normalized and refer to the coordinates of the center point of a bounding box, that labels an object to inspect. Width and height represent the dimensions of the bounding box in normalized pixels, where pixel values are scaled between 0 and 1, relative to the image dimensions. Lastly, the label indicates the two classes valid and defective. Figure 1 overlaps the labels of each image. Figure 1a shows x-center and y-center. The uniform distribution counteracts any specific patterns in the locations of objects. Further, Figure 1b represents the height and width of each bounding box center and indicates the dimension of an object. The linear distribution occurs due to the quadratic geometry of all plastic bricks used. Defective instances in the testing set do not appear in the training or validation set images.

Class	X-Center	Y-Center	Width	Height
Defective	0.43984375	0.43125	0.0375	0.0546875
Valid	0.44765625	0.5921875	0.0390625	0.05625

Table 1: The content of the label file corresponding to the example image in Figure 3b



(a) X-center and y-center are the normalized coordinates of the bounding boxes around objects



(b) Height and width represent the size of a bounding box indicating the dimension of an object and its distance to the camera

Figure 1: The distribution in both figures is nearly uniform and therefore counteracts specific patterns in object locations

The correlogram in Figure 2 shows a detailed correlation of all data properties. It is a group of 2-dimensional histograms showing each axis of the data against each other axis. The correlation statistics indicate the position, width, and height of the bounding boxes of the objects. The figure indicates that the dataset properties are balanced in each label combination with no clusters visible. The distributions of single labels present approximately normal distribution. Notably, outliers are infrequent, and those present are rare points rather than data values that significantly deviate from the expected pattern.

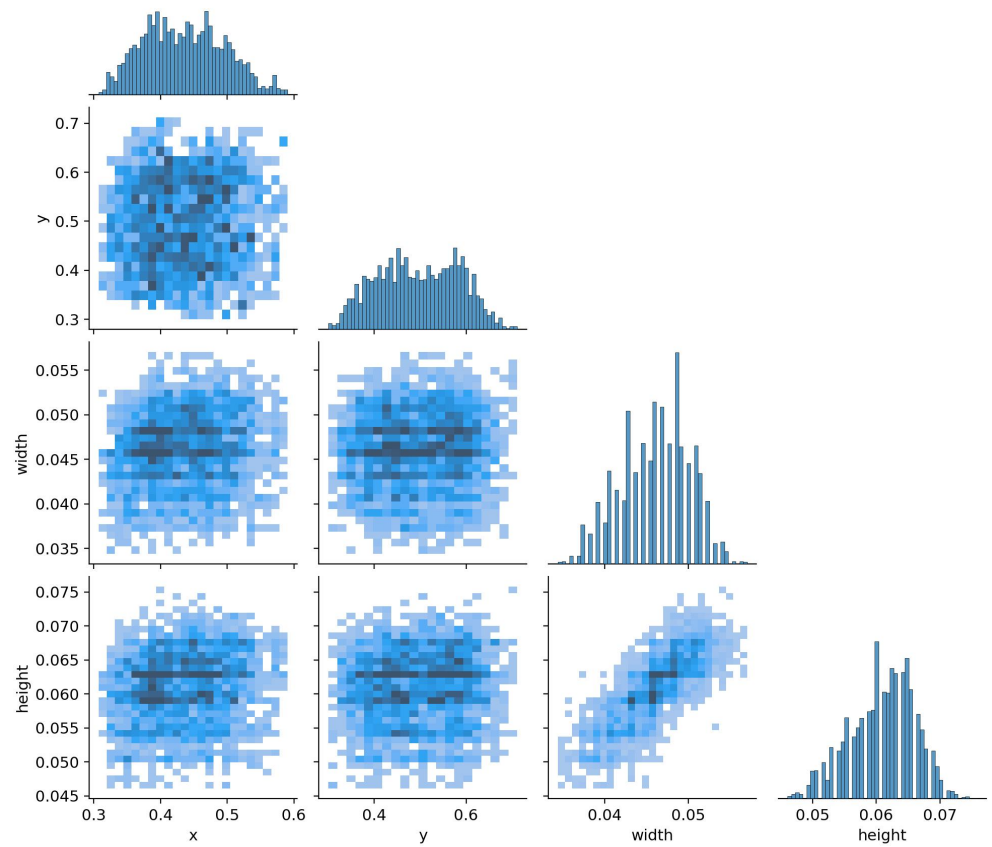


Figure 2: The correlation of all labels to each other shows an approximately normal distribution and balance in the data

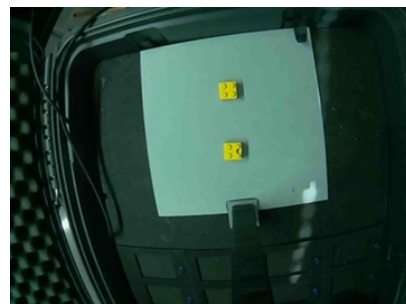
Each image is saved in JPG/.jpg format with a size ranging between 35 and 40 kilobytes. These images maintain a consistent shape of 640x640 pixels. The corresponding labels for these images are stored in a separate file in TXT/.txt format, which also includes metadata on how many instances of each class are present in the image, allowing for the creation of imbalanced subsets of the data. The file paths for both the images and the labels are specified within a file in YAML/.yaml format. As a result, all files collectively occupy a total size of 58.2 megabytes. The files are available on Zenodo and linked in Section 4: [Usage Notes](#).

The dataset offers a wide range of possibilities for diverse tasks, including object localization, object classification, object counting, semantic segmentation, and scene understanding. However, the dataset's provided labels and the identifiable damages on the objects make it particularly well-suited for tasks related to object detection and quality classification, specifically in identifying surface defects often encountered in manufacturing industries.

2.2 Data Collection

The data collection was done in a defined procedure. Images were captured with a microcontroller board and a compatible camera. An Arduino UNO was chosen with an OV7670 300KP VGA Camera. Arduino embedded systems are widely available and used for prototype purposes. They benefit from an active online community helping to lower development challenges [16]. Moreover, the setup includes fluorescent lighting directed toward the objects under inspection, with the camera positioned on a tripod to maintain a fixed distance from the ground, where all the objects are placed. The distance between the camera and the objects is determined by the angle each object has relative to the camera. On the software side, Python code controls the capturing process. The collection started from single objects with different colors, angles, and positions, as well as defects on some objects. Later on, multiple objects were placed in one image with the same differences described. Each defect is generated by a hammer manually and therefore individual with a varying degree of surface damage. This supports the diversity of surface damages that are labeled as defective.

The annotation of the images is based on the software Roboflow [17]. Features, polygon bounding boxes, and labels are provided with this software. Besides, Roboflow is used for auto-orient to discard common rotations by metadata and standardize pixel ordering, as well as resizing the images to a frame of 640x640 pixels from the original camera resolution of 1640x1232 pixels. This resolution of 640x640 is often suggested to facilitate the convenient use of object detection models, such as YOLOv5 [18]. Figure 3a shows an exemplary image before annotation and Figure 3b shows the same image after annotation. The purple box indicates the valid object, while the red box indicates the defective one. Table 1 shows the corresponding label information of Figure 3b. All boxes are applied comprehensively around the relevant objects, ensuring that occluded objects are always fully included. Besides, we aimed to minimize the spaces between the bounding box borders and the objects to ensure that only the relevant objects are enclosed within the box.



(a) Original image



(b) Labeled image

Figure 3: Exemplary image of the dataset consisting of two objects with one valid and one defective instance

Finally, the captured images and labels are stored in Zenodo and saved with a Data Management Plan (DMP) created with RDMO [19]. The DMP includes information about metadata, data formats, as well as technical insights to enhance scientific reuse within FAIR principles. F-UJI scored the resource with a FAIRness of 75 percent.

3 Object Detection and Quality Classification Showcase

While the presented dataset provides possibilities to perform various tasks, this section aims to demonstrate the dataset's suitability for object detection and quality classification through binary defect detection of the surface damages occurring on the objects. This showcase shall be a best practice to learn and facilitate additional exploration. Additionally, training is conducted on varying dataset sizes and class imbalances to demonstrate the performance and its relationship with both the quantity of data and the degree of imbalance used during training. The variation in dataset size and class imbalance is intended to address challenges faced by SMEs with limited and imbalanced data. Therefore, we first explain the metrics used for this task, introduce the algorithm trained, and then present its results across different dataset sizes and class imbalances.

3.1 Metrics

As the task consists of binary defect detection on objects that need to be detected first, several metrics need to be used. The object detection is measured by Intersection over Union (IoU), as suggested by literature [20]. This metric is based on the ratio of the area of intersection of two bounding boxes to the area of union of two bounding boxes as shown in the Formula 1

$$IoU = \frac{\text{Area of Intersection of two bounding boxes}}{\text{Area of Union of two bounding boxes}} \quad (1)$$

Therefore, greater IoU values signify increased overlap and an improved prediction. To eliminate redundant boxes encompassing the same object, IoU typically employs Non-Maximum Suppression. This method operates on the criterion that predictions with IoU lower than the confidence threshold are ignored, while only boxes with IoU values exceeding this threshold are retained. Here, the confidence threshold denotes the minimum score at which the model considers a prediction to be valid. Furthermore, Precision (P) and Recall (R) as classification metrics are applied to measure the accuracy of fault detection within detected objects. Generally, an image typically contains a wealth of information, including both relevant and irrelevant objects. To clarify this, P is introduced to only indicate relevant ones. It measures the proportion of correctly recognized objects out of all detected objects. R , on the other hand, measures the proportion of relevant objects that were correctly recognized by the model out of all relevant objects. The mathematical definitions of P and R are shown in Formula 2 and Formula 3. True Positive (TP) represents correct detections ($IoU \geq \text{confidence threshold}$), False Positive (FP) represents a wrong detection ($IoU < \text{confidence threshold}$), and False Negative (FN) represents a wrong misdetection.

$$\text{Precision}(P) = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall}(R) = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{TP}{TP + FN} \quad (3)$$

P and R offer a trade-off that it is graphically represented in the PR curve by varying the classification threshold. The area under this curve provides the average precision for each class

(AP_i) for the trained model. The average of this value across all classes is referred to as the mean Average Precision (mAP), which is used to evaluate performance in object detection and quality classification in this showcase, as it combines all introduced metrics. The equation is shown in Formula 4. The equation is shown in Formula 4.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

N corresponds to the total number of object classes. mAP has different categories, varying in their parameter settings. We select the most common ones $mAP@0.5$ and $mAP@0.5:0.95$. $mAP@0.5$ is used across several benchmark challenges on datasets such as Pascal VOC or COCO. It interpolates with 101 recall points with (IoU) threshold = 0.5, which means that IoU values greater than or equal to 0.5 are considered TP , while values less than 0.5 are considered FP predictions. $mAP@0.5:0.95$ uses the same interpolation method as $mAP@0.5$, but averages the APs obtained from using ten different IoU thresholds (0.5, 0.55, ..., 0.95). The introduced metrics P , R , $mAP@0.5$ and $mAP@0.5:0.95$ measure the performance of the algorithm during training and in tests after training in this showcase.

3.2 Algorithm and Training

An algorithm of YOLO series is selected as an example real-time object detection algorithm commonly used in research and industry. YOLO series object detection algorithms use a one-stage neural network to directly complete detection object localization and classification without using pre-generated region proposals [21], [22]. They are widely used for their good balance between high speed and high accuracy, easy implementation, and low-cost maintenance. YOLOv5, proposed by Jocher Glenn [18], is selected as the YOLO version after consideration of computing resources, layers of the network, model parameters, detection accuracy, inference time, deployment ability, and algorithm practicability. The specific model YOLOv5 is used for its properties of lightweight and relatively high speed. Since the size of the dataset in this showcase is relatively small and the background information is fixed, real-time detection and high accuracy can be ensured by YOLOv5s at the same time.

Training is conducted on smaller subsets of the dataset, as well as with class imbalances, to demonstrate the model's performance to the number of images and the degree of class imbalance used for training. Three different dataset sizes, with a class imbalance in the first two, are used as shown in Table 2. To create the imbalanced datasets, images with fewer instances of the 'valid' class were selectively removed, resulting in a final dataset with around 65% of images containing valid parts. The sizes of the training datasets are 310, 378, and 1050, respectively. The validation and testing set sizes are 16% and 20% for the 1st and 2nd datasets, and 20% and 10% of the total data for the complete dataset. The algorithm is trained 300 epochs with a batch size of 32 using YOLOv5s default hyperparameters.

	Training Set	Validation Set	Testing Set
1st(normal)	310	78	97
1st(imbalanced)	310	78	97
2nd(normal)	378	94	118
2nd(imbalanced)	378	94	118
3rd	1050	300	150

Table 2: Split of Training set, Validation set, and Testing set for all dataset sizes used

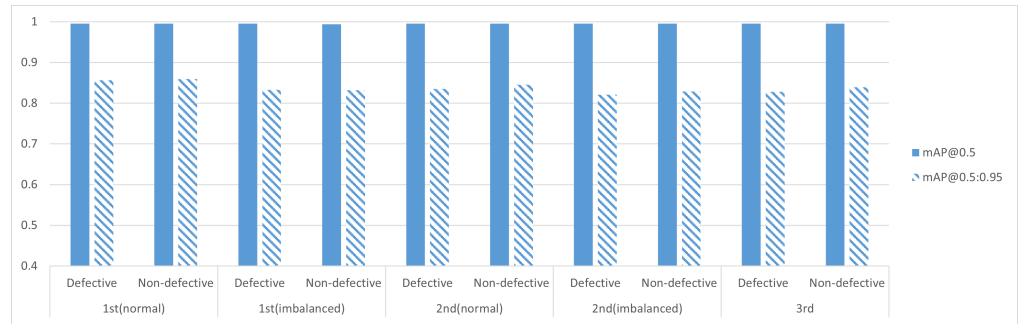
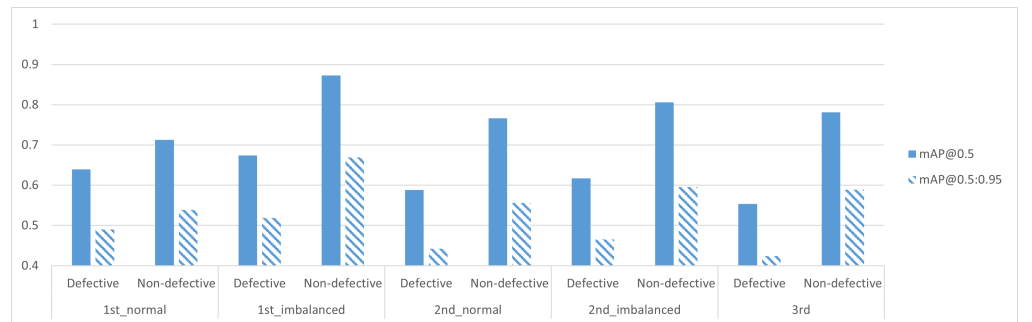
3.3 Evaluation

As introduced, the results are presented with P , R , $mAP@0.5$ and $mAP@0.5:0.95$ for validation and testing set of the dataset and visualized in Table 3 and Table 4. The performance on the validation set exceeds that of the testing set, indicating overfitting during training. Overall, the performance on the testing data varies depending on dataset size and class imbalance. Regarding the entire dataset, the trained model achieves a validation $mAP@0.5$ of 0.995 and test a $mAP@0.5$ of 0.668. The visualized comparison between the size of the dataset can be seen in Figure 4 for the validation data and in Figure 6 for the testing data. Despite this, there is no significant performance increase, suggesting that even with the smallest dataset, satisfactory performance in training, but not in testing is achieved. However, it is possible that more advanced models, such as newer versions of YOLO, could achieve better test performance.

	Class	Precision	Recall	$mAP@0.5$	$mAP@0.5:0.95$
1st(normal)	All	0.99	1	0.995	0.858
	Defective	1	0.999	0.995	0.856
	Valid	0.98	1	0.995	0.859
1st(imbalanced)	All	0.994	0.988	0.994	0.833
	Defective	1	0.975	0.995	0.833
	Valid	0.988	1	0.994	0.832
2nd(normal)	All	0.989	0.983	0.995	0.84
	Defective	0.978	0.99	0.995	0.835
	Valid	1	0.976	0.995	0.845
2nd(imbalanced)	All	0.995	0.996	0.995	0.825
	Defective	1	0.992	0.995	0.821
	Valid	0.99	1	0.995	0.829
3rd	All	0.998	0.999	0.995	0.833
	Defective	0.997	1	0.995	0.828
	Valid	1	0.998	0.995	0.839

Table 3: Precision, Recall, $mAP@0.5$ and $mAP@0.5:0.95$ for the Validation Set

	Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
1st(normal)	All	0.763	0.708	0.676	0.514
	Defective	0.547	0.986	0.639	0.49
	Valid	0.979	0.43	0.713	0.539
1st(imbalanced)	All	0.733	0.794	0.774	0.594
	Defective	0.541	0.995	0.647	0.519
	Valid	0.926	0.592	0.873	0.669
2nd(normal)	All	0.687	0.714	0.677	0.499
	Defective	0.477	1	0.588	0.442
	Valid	0.897	0.429	0.766	0.556
2nd(imbalanced)	All	0.762	0.712	0.711	0.531
	Defective	0.543	1	0.617	0.465
	Valid	0.982	0.425	0.806	0.596
3rd	All	0.707	0.708	0.668	0.507
	Defective	0.473	0.997	0.554	0.424
	Valid	0.941	0.419	0.781	0.589

Table 4: Precision, Recall, mAP@0.5 and mAP@0.5:0.95 for the Testing Set**Figure 4:** $mAP@0.5$ and $mAP@0.5:0.95$ metrics of validation set of all five dataset**Figure 5:** $mAP@0.5$ and $mAP@0.5:0.95$ metrics of Testing set of all five dataset

4 Conclusion

SMEs in the manufacturing sector lag behind their larger counterparts in the adoption of ML technologies like object detection. This is influenced by factors including insufficient data, high complexity, and a scarcity of tangible examples. We presented a simple low-resolution dataset based on plastic bricks with different surface defects to address a typical use case of object detection in manufacturing. By using a low-resolution dataset with a limited number of instances and accounting for class imbalances, we aimed to address typical challenges faced by SMEs. A showcase provided with a YOLOv5 model indicated sufficient performance with different metrics. Our findings show that maintaining simplicity does not compromise performance, demonstrating the effectiveness of straightforward open-source object detection methods and achieving an $mAP@0.5:0.5$ score up to 0.995 in training and 0.774 in testing. These findings were published ensuring FAIR principles and achieved an FAIR score of 75 percent in F-UJI. The provided data and YOLO model can be reused for learning purposes and establish the groundwork for transferring knowledge to object detection tasks with similar surface damages on the objects to inspect. However, it's important to note that the limitation lies in the inability to directly apply such models or data to unrelated tasks. The consideration of the specific context is fundamental for the transferability of the presented methods. Additionally, it is important to recognize that industrial damages can significantly differ in the complexity of their defects. Future research should focus on investigating more universally applicable resources, facilitating direct transfer for use cases at SMEs through interoperable research approaches.

5 Usage Notes

The dataset generated for this research is accessible on Zenodo via DOI (<https://zenodo.org/records/10731976>). The dataset is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). The developed algorithm is available on RWTH Aachen Gitlab (https://git.rwth-aachen.de/zukipro/yolov5_for_plastic_brick_quality_classification) and licensed under GNU Affero General Public License v3.0.

6 Appendix

	Class	mAP@0.5	mAP@0.5:0.95
1st (normal)	Defective	0.995	0.873
	Non-defective	0.995	0.875
1st (imbalanced)	Defective	0.995	0.865
	Non-defective	0.995	0.873
2nd (normal)	Defective	0.995	0.851
	Non-defective	0.995	0.849
2nd (imbalanced)	Defective	0.995	0.858
	Non-defective	0.995	0.859
3rd	Defective	0.995	0.868
	Non-defective	0.995	0.874

Table 5: mAP@0.5 and mAP@0.5:0.95 for the Training set

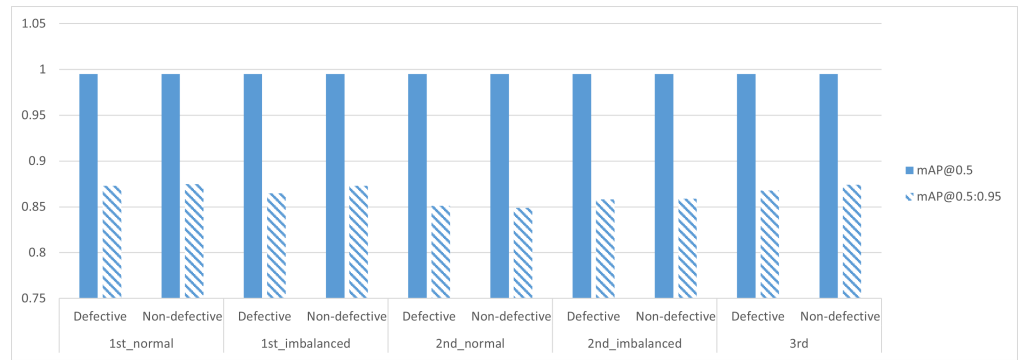


Figure 6: mAP@0.5 and mAP@0.5:0.95 metrics of Training set of all five dataset

7 Acknowledgements

The project "ZUKIPRO" is funded as part of the "Future Centers" program by the Federal Ministry of Labour and Social Affairs and the European Union through the European Social Fund Plus (ESF Plus).

8 Roles and contributions

Jonas Werheid: Conceptualization, Writing – original draft, Writing – review & editing

Shengjie He: Conceptualization, Writing – original draft

Tobias Hamann: Writing – review & editing

Anas Abdelrazeq: Writing – review & editing

Robert H. Schmitt: Funding acquisition & Supervision

References

- [1] M. Rudorfer, *Towards Robust Object Detection and Pose Estimation as a Service for Manufacturing Industries*. Fraunhofer IRB Verlag, 2021.
- [2] L. Malburg, M.-P. Rieder, R. Seiger, P. Klein, and R. Bergmann, "Object detection for smart factory processes by machine learning," *Procedia Computer Science*, vol. 184, pp. 581–588, 2021, The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.04.009>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050921007821>.
- [3] M. Bauer, C. van Dinther, and D. Kiefer, "Machine learning in sme: An empirical study on enablers and success factors," 2020.
- [4] P. Ulrich and V. Frank, "Relevance and adoption of ai technologies in german smes – results from survey-based research," *Procedia Computer Science*, vol. 192, pp. 2152–2159, 2021, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.08.228>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050921017245>.
- [5] C. M. Martin Lundborg, "Künstliche intelligenz im mittelstand: Relevant, anwendungen, transfer." (), [Online]. Available: https://www.mittelstand-digital.de/MD/Redaktion/DE/Publikationen/kuenstliche-intelligenz-im-mittelstand.pdf?__blob=publicationFile&v=5.
- [6] H. van Vlijmen, A. Mons, A. Waalkens, *et al.*, "The Need of Industry to Go FAIR," *Data Intelligence*, vol. 2, no. 1-2, pp. 276–284, Jan. 2020, ISSN: 2641-435X. DOI: [10.1162/dint_a_00050](https://doi.org/10.1162/dint_a_00050). [Online]. Available: https://doi.org/10.1162/dint%5C_a%5C_00050.
- [7] T. DATA. "Safety helmet detection dataset." (), [Online]. Available: <https://www.kaggle.com/datasets/trainingdatapro/helmet-detection/>.
- [8] TubeData1. "Tube object detection image dataset." (), [Online]. Available: <https://universe.roboflow.com/tube/tube-object-detection/dataset/1>.
- [9] Ruben. "Aughmanity_v2.0 computer vision project." (), [Online]. Available: https://universe.roboflow.com/ruben-8bqya/aughmanity_v2.0.
- [10] H. Z. Jianfeng Zheng Hang Wu. "Insulator-defect detection dataset." (), [Online]. Available: <https://datasetninja.com/insulator-defect-detection>.
- [11] G. L. Runwei Ding Linhui Dai. "Augmented pcb defect dataset." (), [Online]. Available: <https://datasetninja.com/augmented-pcb-defect>.
- [12] M. Gribulis. "Synthetic lego brick dataset for object detection." (), [Online]. Available: <https://www.kaggle.com/datasets/mantasgr/synthetic-lego-brick-dataset-for-object-detection/>.

- [13] DREAMFACTOR. “Largest lego dataset (600 parts).” (), [Online]. Available: <https://www.kaggle.com/datasets/dreamfactor/biggest-lego-dataset-600-parts/data>.
- [14] “F-uji.” (), [Online]. Available: <https://www.f-uji.net/?action=test>.
- [15] A. de Giorgio, G. Cola, and L. Wang, “Systematic review of class imbalance problems in manufacturing,” *Journal of Manufacturing Systems*, vol. 71, pp. 620–644, 2023, ISSN: 0278-6125. DOI: <https://doi.org/10.1016/j.jmsy.2023.10.014>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612523002157>.
- [16] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, “A systematic literature review on prototyping with arduino: Applications, challenges, advantages, and limitations,” DOI: <https://doi.org/10.1016/j.cosrev.2021.100364>.
- [17] B. Dwyer, J. Nelson, and J. e. a. Solawetz. “Roboflow (version 1.0) [software].” (), [Online]. Available: <https://roboflow.com>.
- [18] G. Jocher, *Ultralytics yolov5*, version 7.0, 2020. DOI: [10.5281/zenodo.3908559](https://doi.org/10.5281/zenodo.3908559). [Online]. Available: <https://github.com/ultralytics/yolov5>.
- [19] “Research data management organiser.” (), [Online]. Available: <https://rdmorganiser.github.io/>.
- [20] H. Rezaatoughi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “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 (CVPR)*, Jun. 2019.
- [21] X. Xie, K. Chen, Y. Guo, B. Tan, L. Chen, and M. Huang, “A flame-detection algorithm using the improved yolov5,” *Fire*, vol. 6, no. 8, 2023, ISSN: 2571-6255. [Online]. Available: <https://www.mdpi.com/2571-6255/6/8/313>.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.