ing.grid preprints

DATASET DESCRIPTOR

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:
2023-01-01

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/zukipro/yolov5_for_plastic_brick_quality_classification

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, there are substantial challenges regarding labeled data availability in manufacturing contexts, training examples, and the complexity of incorporating within the subject. Real-world datasets present challenges in high resolutions and task specificity that hinder the adoption of object detection by small- and middle-sized enterprises (SMEs) for their manufacturing processes. In this article, we present a simple 640x640 low-resolution dataset based on plastic bricks for object detection, featuring two quality labels to identify minor surface defects in some instances as an example of quality classification. Analyzing our dataset with a YOLOv5 model on four different dataset sizes, we aim to demonstrate the accuracy of a common object detection model in a simple manufacturing use case, showcasing object detection with low-resolution images and the impact of varying data availability. The mean Average Precision mAP@0.5:0.95 in correctly identifying instances improved from 0.786 to 0.833 as we moved from the smallest data size of 485 instances to the complete dataset of about 1500 instances. While our interest is specifically in showcasing object detection for manufacturing with low-resolution images and limited data availability, the generated data and trained model can serve as a common basis to further investigate object detection tasks on a wider variety of similar quality classification use cases in manufacturing.

1 Introduction

In unstructured or less structured environments, object detection and pose estimation are key capabilities 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 found does not describe the specific subject area under investigation. 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. Our primary discovery centers around achieving high accuracy levels despite limited data availability and suboptimal camera resolutions, emphasizing the critical interplay between data volume, 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. 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.

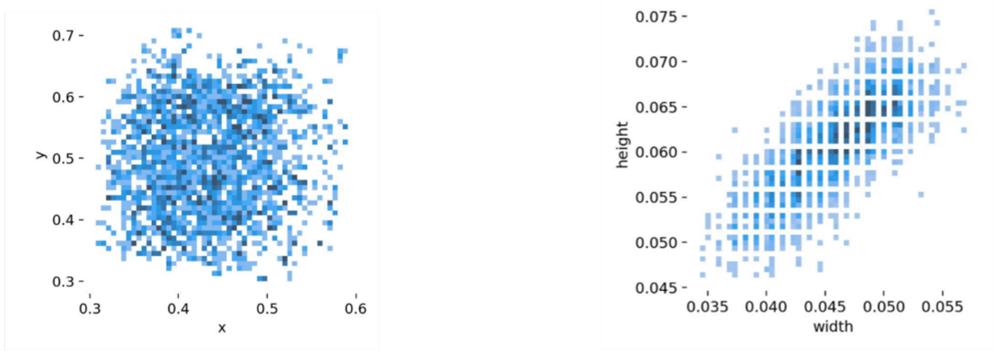
2.1 Data Description

The dataset provides images of plastic toy bricks with surface damages 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.

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 pixels. 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 center and indicates the dimension of an object. The linear distribution occurs due to the quadratic geometry of all plastic bricks used.

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 bonding 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.

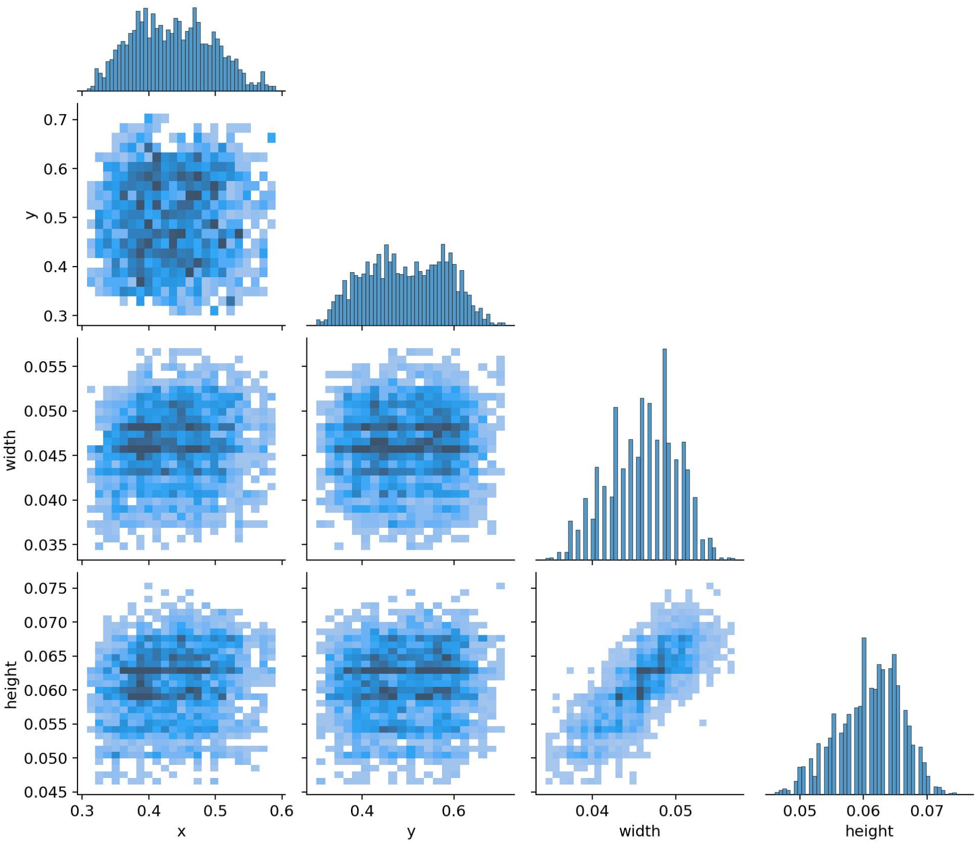


Figure 2: The correlation of all labels to each other shows an approximately normal distribution and balance in the data

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. 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.

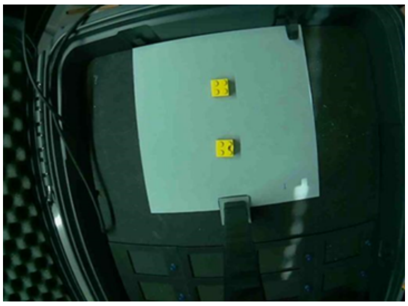
The dataset offers a wide range of possibilities for diverse tasks, including object localization, object classification, object counting, semantic segmentation, and scene understanding. However, the dataset's provided labels and the identifiable damages on the objects make it particularly well-suited 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 [15]. Moreover, the configuration involves fluorescent light directed toward the objects under inspection, with a constant distance between the camera and the objects supported by a tripod. 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 [16]. 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 images to a frame of 640x640 pixels. This shape is often suggested to facilitate the convenient use of object detection models, such as YOLOv5 [17]. 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: Examplary 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 [18]. 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. Further, training is conducted on different dataset sizes to demonstrate performance and its relationship with the amount of data used for training. The variation in dataset size is intended to address challenges faced by SMEs with limited data. Hence, we first explain the metrics used for that task, then introduce the algorithm trained and consequently show its results on the dataset along different dataset sizes.

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 [19]. 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

$$IoU = \frac{Area\ of\ Intersection\ of\ two\ bounding\ boxes}{Area\ of\ Union\ of\ two\ bounding\ boxes} \tag{1}$$

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 represents the number of objects correctly recognized by the object detection model divided by the total number of objects. R is also introduced to indicate all the relevant objects. It measures the number of relevant objects that were correctly recognized by the model. The mathematical definitions of P and R are shown in Formula 2 and Formula 3. True Positive (TP) represents correct detections ($IoU \geq$ *confidence threshold*), False Positive (FP) represents a wrong detection ($IoU <$ *confidence threshold*), and False Negative (FN) represents a wrong misdetection.

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

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

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

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4}$$

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 [20], [21]. 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 [17], 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.

The training is conducted by also taking smaller sizes of the dataset provided to show the model's performance regarding the number of images used for training. Four different dataset sizes are used as shown in Table 2. The sizes of the training datasets are 35, 140, 350, and 1050, respectively. The size of the validation and testing set is the same in all four dataset sizes, 300 and 150 respectively. The algorithm is trained 300 epochs with a batch size of 32 and default hyperparameters.

	Training Set	Validation Set	Testing Set
1st	35	300	150
2nd	140	300	150
3rd	350	300	150
4th	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 correlates with the dataset size. As the size increases, so does the value of evaluated metrics, indicating an improvement in the models' performance. Regarding the entire dataset, the development model achieves a *mAP@0.5* of 0.995 and a *mAP@0.5:0.95* of 0.833. The visualized comparison between the size of the dataset can be seen in Figure 4 for the validation data and in Figure 5 for the testing data. Despite this, the performance increase remains relatively modest, suggesting that even with the smallest dataset, satisfactory performance above 0.95 in *mAP@0.5* is achieved. This indicates the simplicity of the underlying visual task, as the dataset is intended to be easily manageable.

	Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Inference Time [ms]
1st	All	0.954	0.952	0.977	0.786	156.8
	Defective	0.961	0.942	0.978	0.776	
	Valid	0.947	0.963	0.976	0.796	
2nd	All	0.992	0.986	0.994	0.818	157.8
	Defective	0.997	0.978	0.995	0.816	
	Valid	0.987	0.994	0.993	0.821	
3rd	All	0.997	0.996	0.995	0.828	156.5
	Defective	0.996	0.998	0.995	0.823	
	Valid	0.998	0.995	0.995	0.832	
4th	All	0.998	0.999	0.995	0.833	156.4
	Defective	0.997	1	0.995	0.828	
	Valid	1	0.998	0.995	0.839	

Table 3: Precision, Recall, mAP@0.5, mAP@0.5:0.95 and Inference Time in ms for the Validation Set

	Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Inference Time [ms]
1st	All	0.965	0.957	0.977	0.809	188
	Defective	0.973	0.944	0.978	0.81	
	Valid	0.956	0.971	0.977	0.807	
2nd	All	0.99	0.984	0.988	0.83	157.4
	Defective	0.991	0.979	0.987	0.832	
	Valid	0.989	0.988	0.989	0.829	
3rd	All	0.986	0.994	0.995	0.843	156.7
	Defective	0.979	0.995	0.995	0.842	
	Valid	0.992	0.993	0.995	0.844	
4th	All	0.998	0.992	0.995	0.854	159.1
	Defective	0.999	0.995	0.995	0.849	
	Valid	0.996	0.99	0.995	0.859	

Table 4: Precision, Recall, mAP@0.5, mAP@0.5:0.95 and Inference Time in ms for the Testing Set

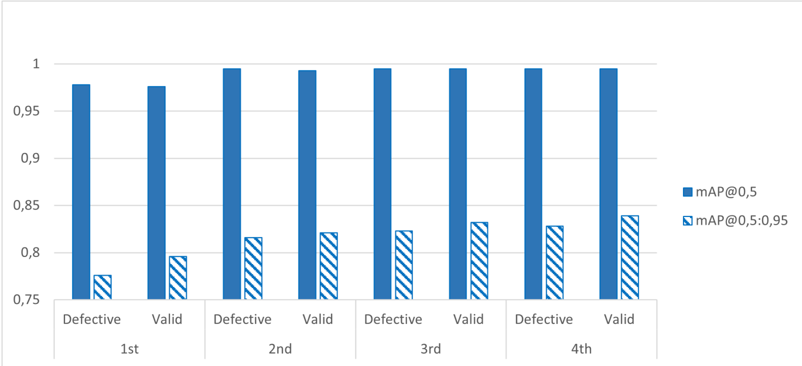


Figure 4: *mAP@0.5* and *mAP@0.5:0.95* metrics of validation set of all four dataset sizes

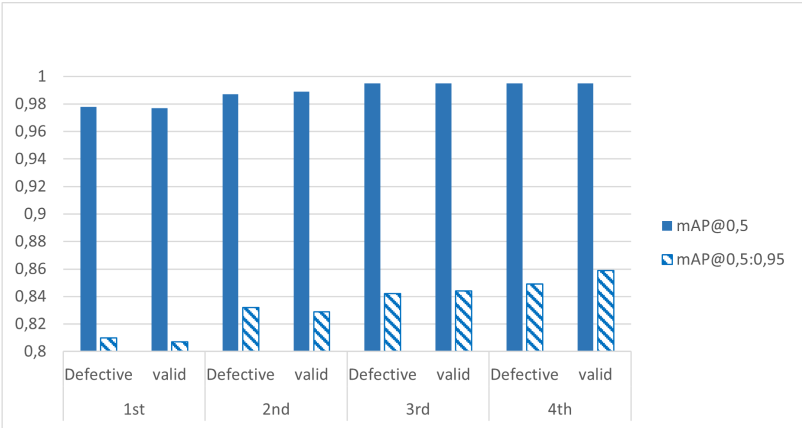


Figure 5: *mAP@0.5* and *mAP@0.5:0.95* metrics of testing set of all four dataset sizes

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 simplification of the dataset with low resolution and a limited amount of instances, efforts regarding typical challenges of SMEs were addressed. 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@0.5:0.95* score up to 0.833. 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. 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 (10.5281/zenodo.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 at https://git.rwth-aachen.de/zukipro/yolov5_for_plastic_brick_quality_classification and licensed under GNU Affero General Public License v3.0.

6 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).

7 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 & Supervison

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. [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$_v 2.0 computervisionproject$.” (), [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] 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.

[16] B. Dwyer, J. Nelson, and J. e. a. Solawetz. “Roboflow (version 1.0) [software].” (), [Online]. Available: https://roboflow.com.

[17] G. Jocher, *Ultralytics yolov5*, version 7.0, 2020. DOI: 10.5281/zenodo.3908559. [Online]. Available: https://github.com/ultralytics/yolov5.

[18] “Research data management organiser.” (), [Online]. Available: https://rdmorganiser.github.io/.

[19] H. Rezatofighi, 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.

[20] 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.

[21] 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.