Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 434–439

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

# Image-Bot: Generating Synthetic Object Detection Datasets for Small and Medium-Sized Manufacturing Companies

Lukas Block[a]*, Adrian Raiser[b], Lena Schön[c], Franziska Braun[a], Oliver Riedel[b]

[a]*University of Stuttgart, Institute of Human Factors and Technology Management (IAT), Nobelstraße 12, 70569 Stuttgart, Germany*
[b]*Fraunhofer IAO, Fraunhofer Institute for Industrial Engineering IAO, Nobelstraße 12, 70569 Stuttgart, Germany*
[c]*Eberhard Karls University Tübingen (Student), Geschwister-Scholl-Platz, 72074 Tübingen, Germany*

* Corresponding author. Tel.: +49-711-970-2173; fax: +49-711-970-2299. *E-mail address:* lukas.block@iat.uni-stuttgart.de

**Abstract**

Training datasets for image recognition are poorly available for small and medium-sized manufacturing companies, due to the specialized products they work with, and the disproportionate investment to generate their own ones. Thus, we investigate a new approach: The Image-Bot consists of a physical apparatus and a processing pipeline to generate training datasets from real-world objects easily. It takes pictures of the objects in front of a green screen and blends them with random backgrounds. The approach was tested with 23 objects and a YOLOv5 algorithm. It creates a state-of-the-art training dataset with about 2,000 images per object in under 45 min.



*Keywords:* Image Recognition; SME; Object Detection; Synthetic Training Data; Chroma Keying; Low-Effort

## 1. Introduction

The recent progress in machine learning based image recognition offers vast potentials in manufacturing applications. Some examples are defect detection (e.g. [1]), robotic positioning and grasping (e.g. [2, 3]), or picking and commissioning (e.g [4]) [5, 6]. Yet, small and medium-sized manufacturing companies (SME) benefit from this progress only to a limited extent [7]. Often training datasets for supervised learning are poorly available, according to literature [3, 5] and following the feedback from the participants of our artificial intelligence training workshops [8, 9]. Complementary, a recent scientific study shows that the lack of suitable solutions is ranked as the second highest obstacle for SME in Germany to employ machine learning (ML) [10].

On the one hand, most of the SME conduct tasks with highly specialized objects and cannot use publicly available or commercial training datasets. On the other hand, creating their own training dataset is not viable from an economic perspective. The suitability of ML algorithms to the specific manufacturing task is ex-ante unknown and must be evaluated. The disproportionate investment for a small or medium-sized company to create a real-world training dataset is thus not lucrative (see [3, 5, 10]).

Different approaches exist, to generate such datasets synthetically. Some examples are 3D modelling and photorealistic rendering, photogrammetry, the employment of generative adversarial networks, roto scoping and image augmentation [11]. Yet, these approaches still require significant effort and specialized knowledge, which is often not present in small or medium-sized companies [10].

Consequently, the lack of easily available training data creates a barrier to the wide-spread use of machine-learning-based algorithms in image recognition for small and medium-sized manufacturing companies.

Thus, we investigate a new approach to generate training datasets for supervised object detection and image segmentation algorithms. Its application should not require any specialized knowledge, it should be low-effort and only rely on artefacts already present at the SME to overcome the stated


10.1016/j.procir.2022.05.004

obstacles in generating image recognition datasets. Our solution for such an approach is the Image-Bot. The Image-Bot is a physical setup and a software pipeline, which uses a common technique from the movie industry [12, 13]: Pictures of objects are taken in front of a green screen from different perspectives. The green parts of the image are removed (so-called chroma keying) and the objects are inserted into random backgrounds (see Fig. 1). We further extend this approach with occlusion and augmentation of the image. The major advantage of this solution is, that it is based on real-world objects but still needs relatively few images to be taken. Certain features of the object to be learned, like defects for example, are only present in real-world objects. It is not necessary to (virtually) recreate them. Furthermore, real-world objects are easily available and can directly be used. Yet, the application of chroma keying is not new to the generation of training datasets. Additionally, in its basic form it is also not low-effort.

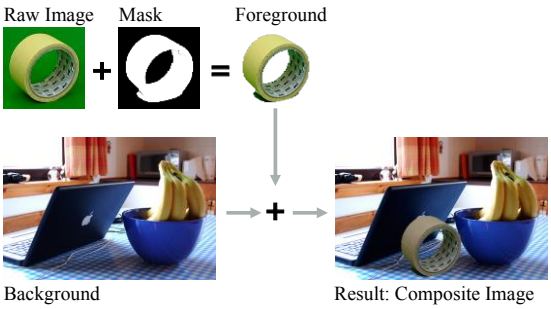


Fig. 1. Image masking and blending to insert a foreground image into a background image (background image from [20]).

## 2. Related Work

Within this paper, a training dataset is defined as a set of training images and their associated ground truth annotations (labels), because most state-of-the-art algorithms in image recognition are of supervised nature. There is an overwhelming amount of literature that deals with the generation of synthetic training datasets. Nikolenko [11] and Schraml [14] provide a state-of-the-art overview on work dealing with this subject. Yet, most of these approaches either require prerequisites like 3D models to be present (see e.g. [3]), need specialized knowledge (see e.g. [5]), certain software pipelines or do not reduce the effort sufficiently for small and medium-sized companies to take advantage of the approach. Only few approaches exist, which build upon capturing real-world physical objects.

One technique to lower the number of images to be taken is image augmentation [11, 15]: Only few images are captured and labeled. They are then multiplied and altered in a variety of ways (e.g., geometric transformations, occlusion, chromatic alternations) to generate a diverse set of training images from a small dataset. They support generalization especially for small datasets [15]. However, image augmentation methods are based on oversampling. They rely on the object's representation in the existing image context.

A possible extension to image augmentation is to mask the objects under consideration (see e.g. [16–18]). The masked image can then be placed in a variety of different backgrounds (see Fig. 1). Combined with augmentation of the masked image or of the composed image a multitude of different training images can be generated [12]. Real object textures and shapes are used, which is one major advantage over computer generated images [18]. Furthermore, object placement and perspective can be controlled to a certain extend [18, 19]. The labels for object detection or image segmentation are derived from the mask. The masked images can be created by hand, through image segmentation techniques (e.g. [17]) or via chroma keying (e.g. [12]). Chroma Keying techniques have been proven to work well and with reasonable effort for arbitrary real-world objects.

LeCun et al.'s work [21] is the earliest research to propose an approach similar to chroma keying: They capture model objects on a grey turntable under controlled lighting conditions with a stereo camera mounted on a swiveling arm. The grey turntable is then digitally replaced with various textured backgrounds to create a dataset for image recognition. They show, that a Convolutional Neural Network (CNN) can be trained with this dataset to recognize objects in real pictures. Sapp et al. [12] apply a real chroma keying approach to segment the foreground object from the green background. The objects are rotated by hand. They exchange textures on the objects and employ image augmentation techniques on the foreground image. Thereby, they find that background exchange plays an important role for more robust algorithms and that augmentation of the foreground image can reduce the number of necessary training images to one third.

Varatharasan et al. [17] and Nguyen et al. [18] apply chroma keying to generate datasets of air vehicles. Nguyen et al. [18] employ a robotic setup to move the camera around the object. A commercial video editing software is used to remove the green screen manually. Adjustments like gaussian filters [17, 18], tonality [17] and brightness [18] adjustments or Seamless Image Cloning [18] are applied to the foreground image to better blend with the background. The generated datasets are evaluated by training a YOLOv3 algorithm. In both cases they perform close to or outperform real-world training datasets, if they pretrain with the green screen data (see also [22, 23]).

Similar approaches to the ones described exist (e.g., [16, 24–27]). However, for all present approaches, neither software code nor construction manuals for the image taking apparatus are publicly available. In Nguyen et al. [18] and LeCun et al. [21] image capturing happens automatically through robots and turntables. They build up a highly controlled, sophisticated, and thus complicated capturing environment, which does not seem to be robust against any changes in its setup. Furthermore, setup costs are probably high.

As such, all reviewed works differ from our goals in that most of them try to generate high quality training images to train or pretrain networks under participation of machine learning experts. Time as a proxy for effort is only considered by Sapp et al. [12]. For our use case, machine learning experts

are seldom present and effort is crucial due to the uncertainty, whether the aimed ML approach is beneficial for the company.

## 3. Image-Bot Approach

The goal of this paper is to research, develop and evaluate the Image-Bot (see Fig. 2). The Image-Bot is a physical apparatus and an associated software pipeline. It creates training datasets for object detection and image segmentation of real-world physical objects with low-effort and does not require any specialized knowledge or preparation to be operated. The following requirements are to be fulfilled by the Image-Bot. They are defined based on the initial problem description and insights from our artificial intelligence training workshops [8, 9].

1. Overall time to generate the training dataset must be lower than 45 min. Human operator involvement should be shorter than 15 min. Consequently, major parts of the process must be automated efficiently (low effort).
2. Costs for the apparatus must be lower than 1,000 €. The components must be easy to acquire (low effort).
3. The Image-Bot must rely on physical objects already present at the SME with a height of up to 30 cm.
4. It should be controlled by an "average" computer present at a SME, e.g. an office computer. It should have an easy-to-use software installation process and use a graphical user interface to control it (low effort, no specialized knowledge).
5. The process should generate suitable training datasets under different apparatus setups and divers types of objects (robustness). None to few parameters to fine tune the process should exist (robustness, no specialized knowledge).

To realize the Image-Bot, we follow the ideas of Sapp et al. [12], Varatharasan et al. [17], Nguyen et al. [18] and LeCun et al. [21] due to their low effort in application and high quality in the generated datasets. However, we also extend the approaches, to make the Image-Bot applicable and robust for SME under different circumstances with little prior knowledge in synthetic data generation. From a research perspective we suggest applying a new methodology for image blending due to a better and more robust performance.

### *3.1. Physical setup*

To fulfill requirement 2, the Image-Bot apparatus builds up on a set of off-the-shelf components. The physical setup (see Fig. 2) consists of a camera, mounted on a small robot, and an embedded controller for the robot. A personal computer performs the image capturing and processing task and provides a graphical user interface to control the process. Furthermore, a green curtain is attached to an L-Shaped support structure. Lighting happens through two soft boxes.

Typical parameters, which should be variated for robust training datasets are lighting, camera position and object position (see req. 5, [5]). The robot moves the camera vertically around the object. The horizontal rotation of the object is conducted by hand through the operator. Initially, we also planned to use a turntable for vertical rotation as LeCun et al. [21] and Nguyen et al. [18] did. However, it casted unfavorable shadows on the green background. Thus, either the robustness of the chroma keying algorithm would suffer, or the physical setup would become less robust to minor derivations (see req. 5). As such we decided to omit the turntable idea in favor of robustness for our approach.

The overall cost for the setup is approximately 500 €. It excludes the personal computer, which we assume to be already present at the SME. The physical setup can scan objects with a height of up to 30 cm (see req. 2 and 3).

Fig. 2. Physical setup to capture and process the images (personal computer is not on the picture).

### *3.2. Data generation process and software architecture*

The data generation process is depicted in Fig. 3. A human operator is still needed to rotate the object horizontally and pick the green value to mask the object. The green value might change for setups in different companies.

The personal computer is the interface to the human operator. It controls the camera as well as the robot and processes the images after capturing. The object's chroma keying mask is automatically generated by calculating a thresholded and normed difference between the operator-picked green value and the rgb-vector of each pixel. Additionally, it is smoothed via edge detection. Object augmentation then alters the foreground image via geometric transformations and by further cropping or occluding. This step is based on the insights from Sapp et al. [12], who state that foreground alternations can contribute to a significant reduction in the number of necessary real-world images (see req. 1). Subsequently, image blending (i.e. foreground and background composition) and composite image augmentation also happens automatically (see section 3.3). To fulfill the timing requirements, the image processing pipeline follows the pipe-and-filters-architecture and is parallelized to run on multiple processors or a GPU, if present (see req. 1).

### *3.3. Image Blending and augmentation*

Image blending happens via Poisson Image Editing as described in Pérez et al. [28]. Poisson Image Editing is more

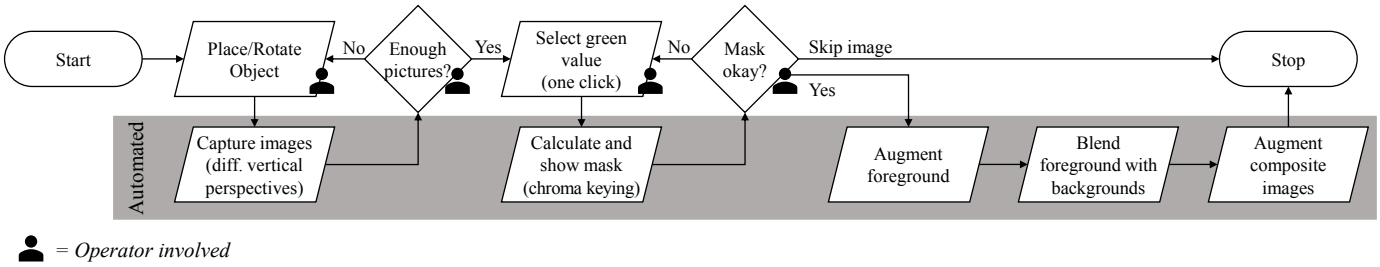


Fig. 3. Synthetic data generation process of the Image-Bot.

sophisticated than the blending algorithms present in the existing approaches. It adjusts the foreground image for illumination and color of the background image. Additionally, it can work with rough object masks (see req. 5). However, we found that even slight green spill effects (i.e., slightly greenish colors at the edge of the masked object) lead to major and obviously wrong color adjustments of the foreground image. Poisson Image Editing blends foreground and background image based on the pixel value derivation of the neighboring pixels. Furthermore, we noticed in first evaluation trainings, that even slight color abnormalities due to the green screen reflections were used as features by the tested object detection algorithms. Thus, we decided to only work with grayscale images in a first step, to maintain robustness (see req. 5).

The background images for image blending are randomly sampled from a subsample of 5,000 images from the COCO 2017 dataset [20]. The composite images are again augmented. This time foreground and background images are augmented together. Geometric transformations are applied and supplemented by further augmentation techniques to simulate frequent occlusions and distortions, typical for an industrial environment like fog or a dirty camera lens.

Overall, this results in at least 20 (=5x2x2) different training images from one captured green screen image. If necessary, this number can easily be scaled up by blending one image with more backgrounds. Yet, we found it to be sufficient.

### *3.4. Implementation, installation process and dissemination*

The software is written in Python using C/C++ implemented libraries like numpy to comply with the performance requirements (see req. 1 and 4). Thus, portability to different personal computer operating systems is possible in general. Dependencies are automatically installed and resolved via the package installer pip. A “README.md” walks the operator through the physical and software setup. It also describes, how to use the Image-Bot. The construction manual as well as the software can be found in our Github Repository [29].

## 4. Evaluation

The Image-Bot approach is tested by generating synthetic training datasets for 23 random, industrial and everyday objects (see Fig. 4). Overall, a variety of challenging attributes like generic shape (e.g. a simple box), holes (e.g. duct tape), surface reflections and filigree, variable shape (e.g. stepper motor with cables) as well as transparency (e.g. plastic bottles) were captured. The images were taken with four different setups of camera type, green screen curtain, and lighting, to check the algorithms’ robustness against changes in the setup (see req. 5). The dataset is publicly available under [30].

Fig. 4. Exemplary objects for which a training dataset was generated.

Between 75 and 200 pictures per object were captured. Capturing took about 10 min on average per 100 pictures. The selection of the green value took about 4.5 min. Learning effects regarding the capturing time and time to pick the green value can be observed on the operator’s side. Table 1 describes the time effort for the first three objects (order: left to right). Overall processing time for 100 pictures was about 31 min (average). This resulted in about 2000 to 3000 training images per 100 pictures captured, depending on the number of skipped images (see Fig. 3). Image processing happened on an institute’s office computer with an Intel i7-8550U processor with 4x2.0GHz and no GPU acceleration.

Table 1. Effort required to generate the synthetic training datasets for the first three objects captured.

| Object | Box | Duct Tape | Stepper Motor |
|---|---|---|---|
| No. of captured images | 83 | 75 | 105 |
| Image capture duration | 12 min | 6 min | 10 min |
| Image Processing *(Operator engagement)* | 7 min *(6 min)* | 4 min *(3 min)* | 9 min *(6 min)* |
| Total duration | 38 min | 26 min | 39 min |
| No. of training images | 1660 | 1488 | 1871 |

Subsequently, the generated datasets were tested by training a YOLOv5s network. We used pretrained weights from the COCO 2017 dataset [31] and trained for 50 to 20 epochs per 2,000 to 5,000 synthetic training images with the default parameters. Evaluation of the trained algorithm happened with about 40 real-world images for each of the objects (see Fig. 5).

Fig. 5. Samples from the evaluation dataset.

## 5. Results and Discussion

We were able to generate a dataset of about 2,000 images per selected object in under 45 min (req. 1, see Fig. 6). Operator engagement was about 18 min per 100 images captured. This is comparable to Sapp et al. [12]. With respect to the training and evaluation results, the real-world performance of the trained algorithms is within the same range or even exceeds previous chroma keying approaches (e.g. [17, 22], see Table 2).

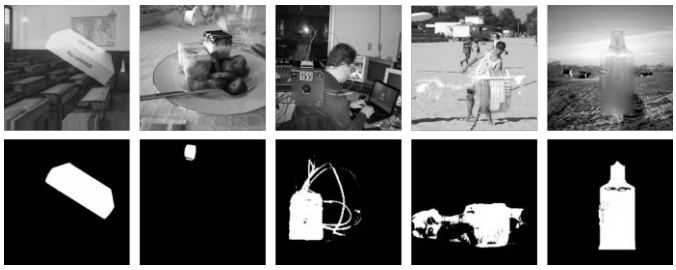

Fig. 6. Samples from the generated dataset (synthetic image and mask).

Table 2. YOLOv5s results for the generated datasets (detailed results in [30]).

| | Precision | Recall | mAP@0.5 |
|---|---|---|---|
| Test (synth. test dataset, 2:1 split) | Avg. 0.958 [0.939-0.981] | Avg. 0.877 [0.804-0.920] | Avg. 0.930 [0.881-0.962] |
| Evaluation (first run) | Avg. 0.611 [0.282-0.903] | Avg. 0.554 [0.308-0.848] | Avg. 0.522 [0.154-0.878] |
| Evaluation (blur / persp. corrected) | Avg. 0.860 [0.669-1.000] | Avg. 0.687 [0.509-0.895] | Avg. 0.737 [0.503-0.936] |
| Evaluation (final result, fine-tuned) | Avg. 0.853 [0.714-0.954] | Avg. 0.807 [0.613-0.956] | Avg. 0.824 [0.650-0.974] |

However, first application of the Image-Bot to real-world evaluation datasets revealed much worse results (2nd row of Table 2). The chosen Poisson Image Editing algorithm blurs objects with no significant texture, if the edge between the green screen and the object is masked. Thus, we disabled mask cropping in our algorithm. Furthermore, the operators missed certain perspectives or states of the objects while capturing them. Common examples were top perspectives and states of variable parts. However, these perspectives and states were present in the real-world evaluation dataset because it was generated independently. Thus, we adjusted our Image-Bot manual for the operators to point this out. The missing perspectives or states were added. In one case, the original object was not available anymore and the not captured states were removed from the evaluation dataset for consistency. The real-world evaluation results in row 3, Table 2 were obtained.

Yet, they were still worse than the expected results, implied by the synthetic test dataset. We found that the algorithms were trained on a mixture of real-world features and artefacts in the synthetic training dataset. Examples of such artefacts are shadows and remaining parts of the greenscreen. The real-world evaluation dataset's precision, recall and mAP@0.5 increased proportionally to the values of the synthetic test dataset in the first five to ten epochs of training. Afterwards, the evaluation dataset's values randomly alternated around the final value without convergence, even when the synthetic test dataset's values were still decreasing. The amplitude of the random alternations was linked to how severe the artefacts were. Consequently, the algorithms were fine-tuned with a smaller set of real-world training data to balance out the artefacts (see e.g. [18]). They were trained with half of the evaluation dataset (about 20 images) for five epochs. The other half of the evaluation dataset was used for testing. Satisfactory results in real-world evaluation compared to the synthetic test dataset were obtained (4th row of Table 2).

Consequently, the Image-Bot's synthetic training datasets are not sufficiently good for a production-ready application. Yet, they build a solid basis, which can easily be fine-tuned with only few real-world images. YOLOv5 algorithms only trained on the same amount of real-world images delivered much worse result (mAP@0.5 < 0.1).

In terms of robustness against minor changes in the physical setup, the chroma keying step masked the images equally well for the different setups (req. 5). No effect with respect to the final results was observed for partial translucent, reflective and transparent surfaces. The only prior knowledge required to operate the Image-Bot was how to operate the software and how to setup the physical system. This is clearly described in the software repository and the construction manuals (see [29]).

To summarize, the requirements 1 to 5 are fulfilled. Yet, the Image-Bot was only tested by training one object detection algorithm without any specific use case. Real-world and on-site tests regarding commissioning, sorting and fault-part detection were planned, but not possible due to SARS-CoV-2.

## 6. Conclusion

Within this paper we presented the Image-Bot. It is a physical apparatus and software pipeline to generate synthetic image recognition training datasets fast and with low effort from real-world physical objects. It should support SME in generating their individual training datasets to leverage the potentials of machine-learning-based image recognition. Thus, on-site evaluation with SME is for sure the next step, to validate that the Image-Bot is indeed a solution to the problem described. Furthermore, we plan to evaluate the training datasets with further machine-learning algorithms, because performance might differ significantly (see e.g. [3]).

Regarding the technical setup, the current simple approach for chroma keying can be extended with more advanced background removal techniques (e.g. [12, 17, 32, 33]). They

might provide a better solution to the existing challenges regarding artefacts, green spill and grayscale images. As such, robust solutions to process color images and capture them from more diverse perspectives are currently under investigation. Domain-knowledge for context accurate placement can be incorporated to improve the results even further (see e.g. [5]).

To the best of our knowledge, the Image-Bot is the first chroma keying approach for synthetic dataset generation to utilize Poisson Image Blending algorithms. We argue that this is one of the reasons for the good evaluation results besides the simple and robust processes. Suitable training datasets for object recognition were created in 45 min per object. No prior knowledge about synthetic dataset generation was necessary.

## Acknowledgements

The research conducted in this paper is part of the research project Morphoa, funded by the German Federal Ministry of Education and Research (BMBF). The authors are responsible for the content of this publication.

## References

[1] Chen, X., Chen, J., Han, X., Zhao, C. *et al.*, 2020. A Light-Weighted CNN Model for Wafer Structural Defect Detection. IEEE Access *8*.

[2] Farag, M., Ghafar, A.N.A., Alisibai, M.H., 2019. Real-Time Robotic Grasping and Localization Using Deep Learning-Based Object Detection Technique, in *2019 IEEE International Conference on Automatic Control and Intelligent Systems: I2CACIS 2019*, IEEE, Piscataway, NJ, p. 139.

[3] Manettas, C., Nikolakis, N., Alexopoulos, K., 2021. Synthetic datasets for Deep Learning in computer-vision assisted tasks in manufacturing. Procedia CIRP *103*, p. 237.

[4] Rennie, C., Shome, R., Bekris, K.E., Souza, A.F. de, 2016. A Dataset for Improved RGBD-Based Object Detection and Pose Estimation for Warehouse Pick-and-Place. IEEE Robot. Autom. Lett. *1*, p. 1179.

[5] Schoepflin, D., Holst, D., Gomse, M., Schüppstuhl, T., 2021. Synthetic Training Data Generation for Visual Object Identification on Load Carriers. Procedia CIRP *104*, p. 1257.

[6] Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. *Object Detection in 20 Years: A Survey*.

[7] Kaul, A., Schieler, M., Hans, C., 2019. *Künstliche Intelligenz im europäischen Mittelstand: Status quo, Perspektiven und was jetzt zu tun ist*.

[8] Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Business Innovation Engineering Center (BIEC). KI einfach machen! https://biec.iao.fraunhofer.de/de/Transfer/ki-einfach-machen.html. Accessed 29 May 2021.

[9] Fraunhofer Institute for Industrial Engineering IAO, Business Innovation Engineering Center (BIEC). Students teach Professionals. https://biec.iao.fraunhofer.de/de/Transfer/Students-teach-Professionals.html. Accessed 29 May 2021.

[10] Bauer, W., Ganz, W., Hämmerle, M., Renner, T., Dukino, C., Friedrich, M., Kötter, F., Meiren, T., Neuhüttler, J., Schuler, S., Zaiser, H., 2019. *Künstliche Intelligenz in der Unternehmenspraxis: Studie zu Auswirkungen auf Dienstleistung und Produktion*. Fraunhofer-Verlag, Stuttgart.

[11] Nikolenko, S.I., 2019. *Synthetic Data for Deep Learning*.

[12] Sapp, B., Saxena, A., Ng, A.Y., 2008. A fast data collection and augmentation procedure for object recognition, in *Proceedings of the 23rd AAAI*, AAAI Press, Menlo Park, Calif.

[13] Smith, A.R., Blinn, J.F., 1996. Blue screen matting, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, p. 259.

[14] Schraml, D., 2019. Physically based synthetic image generation for machine learning: A review of pertinent literature, in *Photonics and Education in Measurement Science 2019*, SPIE, Bellingham, Washington, p. 51.

[15] Shorten, C., Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. J Big Data *6*.

[16] Georgakis, G., Mousavian, A., Berg, A., Kosecka, J., 2017. Synthesizing Training Data for Object Detection in Indoor Scenes, in *Robotics: Science and System XIII*, Robotics Science and Systems Foundation.

[17] Varatharasan, V., Shin, H.-S., Tsourdos, A., Colosimo, N., 2019. Improving Learning Effectiveness For Object Detection and Classification in Cluttered Backgrounds, in *The 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems: RED-UAS 2019*, IEEE, Piscataway, New Jersey, p. 78.

[18] Nguyen, T., Miller, I.D., Cohen, A., Thakur, D., Prasad, S., Taylor, C.J., Chaudrahi, P., Kumar, V., 2020. *PennSyn2Real: Training Object Recognition Models without Human Labeling*.

[19] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M. *et al.*, 2019. ImageNet-trained CNNs are biased towards texture: increasing shape bias improves accuracy and robustness, in *Proceedings of the 7th International Conference on Learning Representations: ICLR 2019*.

[20] Lin, T.-Y., Maire, M., Belongie, S., Hays, J. *et al.*, 2014. Microsoft COCO: Common Objects in Context, in *Computer vision - Part V: ECCV 2014*, Springer, Cham, p. 740.

[21] LeCun, Y., Huang, F.J., Bottou, L., 2004. Learning methods for generic object recognition with invariance to pose and lighting, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition: CVPR 2004*, IEEE Computer Society, Los Alamitos, Calif., p. 97.

[22] Ryan, K., Bahhur, B., Jeiran, M., Vogel, B.I., 2021 - 2021. Evaluation of augmented training datasets, in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXXII*, SPIE, p. 19.

[23] Nowruzi, F.E., Kapoor, P., Kolhatkar, D., Hassanat, F.A., Laganiere, R., Rebut, J., 2019. *How much real data do we actually need: Analyzing object detection performance using synthetic and real data*.

[24] Wei, X.-S., Cui, Q., Yang, L., Wang, P., Liu, L., 2019. *RPC: A Large-Scale Retail Product Checkout Dataset*.

[25] Follmann, P., Böttger, T., Härtinger, P., König, R. *et al.*, 2018. MVTec D2S: Densely Segmented Supermarket Dataset, in *Computer Vision: ECCV 2018*, Springer, p. 581.

[26] Zanella, R., Caporali, A., Tadaka, K., Gregorio, D. de *et al.*, 2021 - 2021. Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence, in *2021 International Conference on Computer, Control and Robotics: ICCCR*, IEEE, p. 292.

[27] Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, 2018. *Adaptive 3D Object Recognition for Service Robots*, Stuttgart.

[28] Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing. ACM Trans. Graph. *22*, p. 313.

[29] Block, L., Raiser, A. Image-Bot Git Repository. https://github.com/FraunhoferIAO/Image-Bot. Accessed 14 March 2022.

[30] Block, L., Schön, L., Raiser, A. Image-Bot: Everyday/Industrial Objects: Dataset. https://doi.org/10.17632/4nn2w8rvx3.2.

[31] Jocher, G. Yolov5 Releases: v5.0. https://github.com/ultralytics /yolov5/releases/tag/v5.0. Accessed 30 November 2021.

[32] Sengupta, S., Jayaram, V., Curless, B., Seitz, S.M. *et al.*, 2020. Background Matting: The World Is Your Green Screen, in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Conference Publishing Services, Los Alamitos, California, p. 2288.

[33] Rosas-Romero, R., Lopez-Rincon, O., David Rojas, E., Jacobo, N.-P., 2016. Learning matte extraction in green screen images with MLP classifiers and the back-propagation algorithm, in *2016 International Conference on Electronics, Communications and Computers: CONIELECOMP*, IEEE, Piscataway, NJ, p. 14