The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-023-12622-4

ORIGINAL ARTICLE

Cut-paste image generation for instance segmentation for robotic picking of industrial parts

Jonas Dirr[1] · Johannes C. Bauer[1] · Daniel Gebauer[1] · Rüdiger Daub[1]

Received: 11 June 2023 / Accepted: 27 October 2023

Abstract
Vision-based robotic picking enables automation of commissioning and sortation of disordered parts. To locate parts for grasping, state-of-the-art approaches rely on convolutional neural networks for instance segmentation in 2D images. However, this requires sufficiently large training datasets, which are expensive to capture and annotate. Therefore, training with synthetic data is promising as the data can be generated automatically. We present an approach for the cut-paste method to create synthetic images for industrial use cases. With this approach, an end-user first prepares the image generation with just a smartphone and about 20 minutes of manual effort. Then, a versatile dataset with instance segmentation labels is generated automatically. In addition, a procedure for grasp pose computation is applied to enable robotic picking based on instance segmentation. For evaluation, training data is generated for a wide range of rigid parts and deformable linear objects. Testing with real-world data and practical experiments demonstrates the effectiveness of the proposed cut-paste method for industrial applications.

Keywords Synthetic training data · Data generation · Copy-paste · Bin picking · Deformable linear objects · Cable

1 Introduction

Robotic picking systems enable the automation of non-value-adding commissioning and sortation. During these tasks, parts are typically provided in an unordered manner, for example in small load carriers. Therefore, part localization is required for precise grasping. State-of-the-art approaches apply convolutional neural networks (CNNs) for instance segmentation of parts in 2D images [1]. However, training CNNs requires large amounts of annotated data. Manually photographing parts and labeling images is highly time-consuming and costly. This manual work is not feasible, especially for industrial applications with frequently changing variants.

One approach to reduce these efforts is the rendering of synthetic images from simulations with tools like blender [2]. This enables the automatic generation of images and corresponding labels. Nevertheless, the so-called domain gap between synthetic and real-world images [3] is an open challenge for this approach. In addition, many implementations require expert knowledge or a geometric model of the part [4].

In this work, a different approach for the generation of synthetic training data is investigated: The cut-paste method follows a simple procedure, where real-world images are used as a source for the generation of synthetic images with labels [5]. An advantage of the cut-paste method is that source images from the target environment can be used, which helps to reduce the domain gap compared to rendered images [3]. However, the preparation and execution of the cut-paste method can require considerable manual effort by the end-user. Hence, we propose boundary conditions and an approach for the cut-paste method to significantly reduce the manual effort. Thus, training data can be generated cost-efficiently for new part types and variants to quickly implement CNN-based localization. Experiments demonstrate the effectiveness of the proposed method for instance segmentation and robotic picking in cluttered scenes (Fig. 1). Thereby, various types of rigid parts and deformable linear

✉ Jonas Dirr
jonas.dirr@iwb.tum.de

[1] Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstraße 15, Garching 85748, Germany

Published online: 23 November 2023

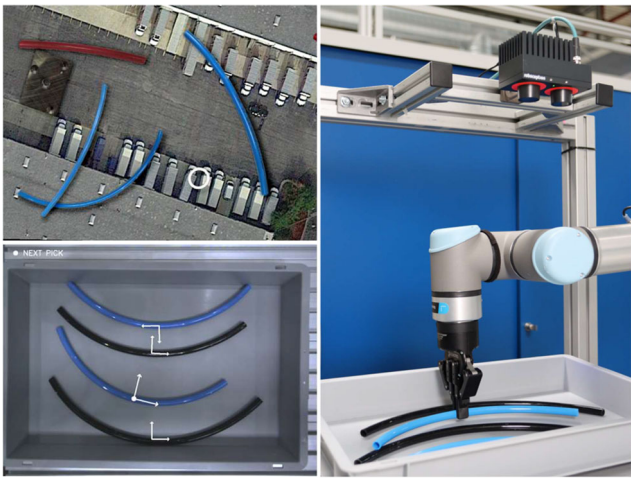

Fig. 1 The proposed cut-paste method generates synthetic training data for instance segmentation (top left). Consequently, grasp poses (bottom left) are computed based on instance segmentation of real images to enable robotic picking (right)

objects (DLOs), such as cables and hoses, are taken into account.

The following summarizes the contribution of this work: (1) We propose a pipeline for the generation of synthetic training data based on the cut-paste method, which is specifically tailored towards industrial applications. (2) Boundary conditions are introduced to generate synthetic datasets with few source images, which further reduces the manual effort for training data generation. (3) We demonstrate the usage of smartphone-based training data for vision-based picking with an industrial camera.

2 Related works

The following introduces the basic principle of the cut-paste method and summarizes preceding works. For other synthetic image generation methods, see the review in [6].

The goal of the cut-paste method is to generate new image datasets from source images using pixel operations [5]. For this purpose, objects in the source images are cropped from the background in the cut step (Fig. 2). In the subsequent paste step, the cropped objects are composed with different backgrounds to new images. In contrast to classical image augmentation, new scene compositions can be created by the variable arrangement of different source images [6]. Cropping the object from the source image also creates an object mask. Therefore, instance masks of the synthetic images can be computed automatically. With the repeated application of the cut-paste method, versatile image datasets can be generated for training CNN-based image segmentation models.

Recent works show different approaches to reduce the manual effort and generate better-performing training data. A summary of related works on source image acquisition as well as the cut and paste step is provided next. So-called hybrid methods, in which source images are rendered from a simulation [8], are not considered below.

2.1 Source image acquisition

In a step preceding the actual cut-paste method, source images must be captured or collected. While the images can be taken manually by the user [7], other approaches use open source datasets as source images [9]. Alternatively, internet

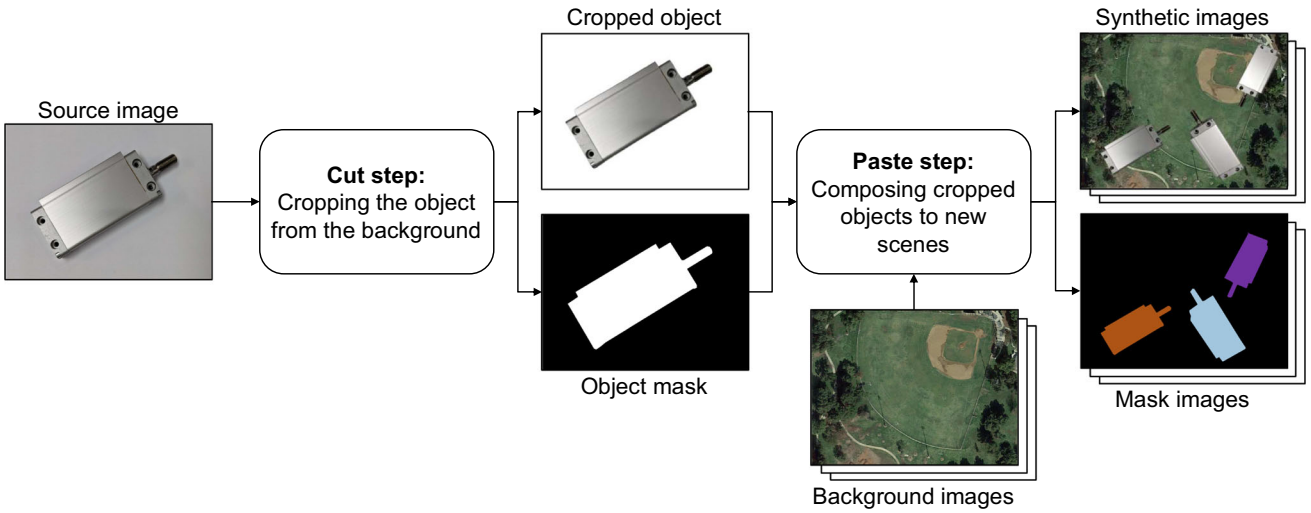

Fig. 2 Basic procedure of the cut-paste method: In the cut step, an object is cropped from the background of the source image. This generates a cropped object and an object mask. In the paste step, new synthetic images are generated by pasting the cropped object variably in new backgrounds. The corresponding mask images are computed automatically. A number of synthetic images can be generated by repeated execution of the method. (This figure is inspired by [5, 7].)

search engines are applied to collect existing images [10]. In [11], a robotic arm positions a camera to capture parts from multiple viewpoints. Zanella et al. [12] record a video sequence and extract image frames to collect a large set of source images.

2.2 Cut step

Once the source images are captured, the objects of interest must be separated from the image background, which can be performed manually, similar to labeling [13]. If the source images are taken in front of a green screen [14], chroma keying is applicable for background separation with little manual effort [12]. However, if there are multiple objects in one image, they cannot be distinguished from each other [12]. Such a semantic segmentation mask is only partially useful, since a segmentation on instance level is required for robotic picking. Dwibedi et al. [5] utilize a pre-trained neural network to automatically determine the object masks. Practically, even semi-automatic and automatic approaches require manual inspection of the generated masks to reject corrupted object masks.

2.3 Paste step

During the paste step, the cropped objects are fused in new background images. While in [12], only the background is exchanged, most approaches compose new image scenes by variable arrangement of multiple objects [7].

Various blending methods are used to reduce optical artifacts that may occur when objects are pasted in the new backgrounds [5]. In addition, image augmentation is performed to increase the diversity of the synthetic images. On the one hand, augmentation is applied before insertion to the cropped objects [5] and backgrounds [12]. On the other hand, the new images are augmented after insertion. In [5] and [9], the effects of different blending and augmentation methods on the object detection performance are investigated. Thereby, it is found in [5] that the application of multiple augmentations and additional insertion of distractor objects, which are labeled as background, can improve the detection performance. In contrast to the preceding steps, pasting can be performed fully automatically.

2.4 Summary

The cut-paste method has been tested for various environments. Examples include household objects in kitchen scenes [5, 9] as well as outdoor scenes with vehicles and living beings [15]. Thus, apart from [12], where synthetic images of cables are generated, hardly any industrial use cases are considered.

Industrial applications of machine vision strongly differ from the scenarios considered so far. Unlike in outdoor scenes, environmental conditions such as backgrounds are constant and well-defined for stationary industrial applications, e.g. bin picking [4]. This enables the usage of domain-specific datasets for industrial use cases [4]. Kitchen scenes show characteristic objects, making object detection transferable across households [16]. In contrast, industrial applications show specific parts requiring individual training data [16]. Therefore, source images must be captured and cropped repeatedly for new part types and variants, which requires hardware setups [11] or manual work [10]. Thereby, the manual effort scales with the number of source images.

Existing approaches use several hundred [5, 10] to thousands [12] of source images. Thus, the manual effort required can be an obstacle to the application of the cut-paste method in industry.

3 Methodology

3.1 Overview

We propose an approach for the cut-paste method, which can be applied with minimal manual effort and without expert knowledge. In addition, we apply a procedure for the computation of grasp poses that enables picking parts based on instance segmentation masks.

The following presents an overview of the methodical procedure and the structure of the successive sections: The starting point is the cut-paste method, which is used to create synthetic images and corresponding labels (Sect. 3.2). This data is used to train CNN-based models for instance segmentation (Sect. 4.3). In Sect. 4.4, the trained models are evaluated on real-world test images.

Moreover, the trained segmentation models are integrated into a robotic system. Here, the segmentation results serve as the basis for the grasp pose computation (Sect. 3.3). Finally, the picking performance of the robotic system is tested in Sect. 4.5.

3.2 Cut-paste method

The proposed approach is based on the basic principle of the cut-paste method (Fig. 2). Below, we introduce additional boundary conditions, a novel data acquisition procedure, and adapted cut and paste steps.

Boundary conditions The goal is to define boundary conditions that create a solution space for the industrial application of the cut-paste method where a small number of source images is required. The following cut-paste approach is designed for flat and slim parts. This covers parts that have an aspect ratio of a minimum of 5:1. In addition, only appli-

cations are considered in which the camera is mounted in the top view and its distance to the work surface is significantly larger than the part height.

Source image acquisition It is the objective to perform the source image acquisition with resources of high availability and low system complexity. Thus, the parts are photographed with smartphones or tablets. First, one part is positioned in front of a neutral background, which provides contrast. Then, the camera device is positioned above the object as it is expected for the segmentation task. In this way, images are taken while the object poses within the image plane and the dominant orientations are varied. By this, different configurations of object, light, and camera are imaged.

Cut step Sect. 2 describes advanced methods for separating objects and backgrounds in source images. However, here an approach with high availability and low complexity is required. For this reason, web applications are used for automatic cropping of source images. Nevertheless, depending on the object, its shadow cast, and the camera perspective cropping may fail for some images. For such cases, the web applications offer tools to manually post-process the images in order to fix local error spots. Alternatively, the faulty images are sorted out.

Paste step This approach uses open-source images for the background and strongly augments the cropped objects to compose new images. In the first step, a background image is randomly selected from over 400 images of the iSAID dataset [17]. Then, with a uniform distribution between two and five cropped objects are added to the background to compose a new scene.

Both the cropped objects and the newly generated images are augmented. For this purpose, the object position and orientation in the background image are randomized. Additionally, the objects are scaled and sheared. Further, the brightness and contrast of the objects are varied. For each object added to the background, a randomly oriented shadow and, with a probability of 50 %, a distractor are inserted. The distractors are captured, cropped, and augmented in the same way as the objects. All objects and distractors are blended with Gaussian blur into the background image. Gaussian blur and Gaussian noise are added to 25 % of the resulting images. Finally, the corresponding mask images are computed automatically.

3.3 Grasp pose computation

Given a 2D image, the grasp pose computation is considered a three-degree of freedom problem. Therefore, for each instance in the image, the x- and y-position as well as the orientation α of the grasp pose P_i are computed in pixel coordinates.

Algorithm 1 summarizes the grasp pose computation building up on our previous work [18]. Starting from an image I, instance segmentation is performed, to obtain the instance contours C for example as splines. From C the instance mask image M is derived, whereby thresholds for the mask size can be applied and the number of instances N in the image is determined. Consequently, for each instance, a binary mask image M_b is computed from M.

Due to the wide range of parts considered in the validation (Sect. 4), slim objects are distinguished during the grasp pose computation. For slim objects, the skeleton of the object mask is computed S_i, while for other object shapes the principal axis A_i is determined. Based on S_i or A_i, the center point and the angle of inclination in the vicinity are determined as grasp pose P_i. At last, one prioritized grasp pose P_p is selected per image from the list of grasp poses P.

Algorithm 1 Grasp pose computation

Input: image I
Output: prioritized grasp pose P_p

$C \Leftarrow instance\_segmentation(I)$
$M \Leftarrow mask\_generation(C)$
$N \Leftarrow number\_of\_instances(M)$
for $i \Leftarrow 1$ to N **do**
  $M_b \Leftarrow binary\_mask\_extraction(M, i)$
  if mask shape is slim **then**
    $S_i \Leftarrow skeletonization(M_b)$
    $P_i \Leftarrow grasp\_pose\_generation(S_i)$
  else[mask shape is not slim]
    $A_i \Leftarrow line\_fitting(M_b)$
    $P_i \Leftarrow grasp\_pose\_generation(A_i)$
  end if
  $P.append(P_i)$
end for
$P_p \Leftarrow grasp\_pose\_selection(P)$

This procedure ensures that the grasp poses are located on the object mask despite different part shapes. If the computation of a skeleton or principal axis fails, no grasp pose is returned for this instance. The presented procedure is independent of the segmentation method.

4 Experimental setup and results

4.1 Use cases

For the subsequent experiments, different use cases are defined by the combination of parts and attributes such as the presence of distractors, the camera system, and the geometric arrangement. Thus, the evaluation covers a wide range of different challenges. During the evaluation, industrial parts are provided in small load carriers or in front of neutral backgrounds. Each scene contains one of the following five part types: pneumatic cylinder, milled plate, USB flash drive board, electric cable, or pneumatic hose.

Table 1 Conditions during training data generation: Device used for source image acquisition and number of correctly cropped objects during the cut step

Part type	Device	Cropped images
Cylinder	Google Pixel 4a	44
Plate	Google Pixel 4a	44
USB	Google Pixel 4a	50
Cable	iPad Pro	43
Hose	Samsung S20	27

All instances of the types cylinder, plate, and USB are identical. In contrast, for hoses and cables multiple variants, which differ in length, diameter, and color, exist. In addition, cables can have an individual shape in each scene as a result of deformation. The dimensions of the parts fulfill the conditions defined in Sect. 3.2. Only the cylinder has an aspect ratio of 5:2. Due to the limited space in the scenes, distractors occur only in scenes with the compact rigid parts, but not in scenes with DLOs. Overlapping between parts or with distractors is not permitted in the scenes.

4.2 Training data generation

Based on the description in Sect. 3.2, one dataset with 5000 images (720 × 540 pixels) and corresponding mask labels is generated for each part type. During source image acquisition multiple devices are used to photograph the parts in front of white paper (Table 1). On average, four source images are captured per minute. Afterward, these source images are cropped using the web application removebg[1], which takes about five minutes for 40 images. Rejecting incorrectly cropped images during a manual inspection can lead to a varying number of images that are used subsequently (Table 1). The paste step is implemented using Python 3.6 and the libraries OpenCV[2] and NumPy[3]. Figure 3 shows exemplary synthetic images generated with the proposed cut-paste method.

4.3 Model training

The synthetic dataset of each part type is used to train one CNN-based instance segmentation model. Thereby, the implementation in the mmDetection framework [19] is used for the SOLOv2 architecture [20] with a ResNet-50 as backbone. The models, which are pretrained on the COCO dataset [21], are trained with a batch size of 8, using the Adam optimizer with a learning rate of 1e-5 and a cosine decaying learning rate schedule. The training on a NVIDIA RTX 3090 GPU is stopped after 10 epochs, which is sufficient for the models to converge. Since the datasets are synthetically generated, only basic data augmentation is applied, including random horizontal flips and random resize crops.

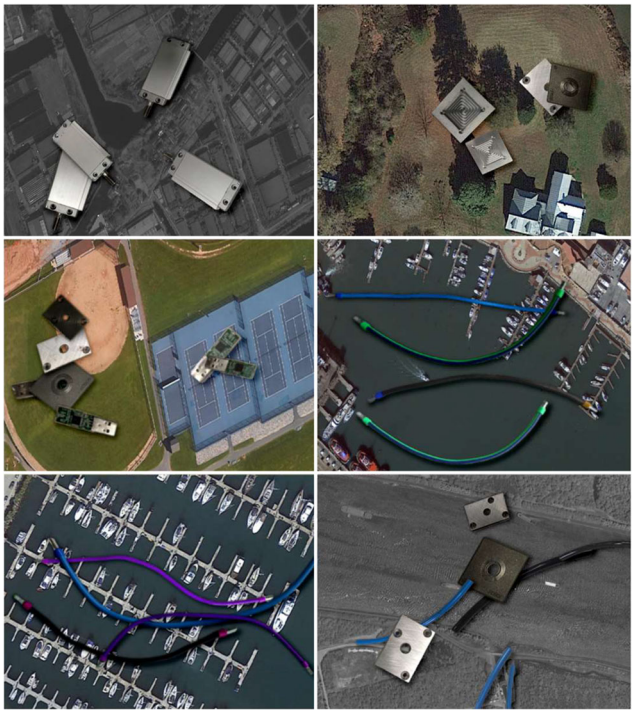

Fig. 3 Exemplary synthetic images generated by the proposed cut-paste method

4.4 Experiment 1: instance segmentation

This first experiment evaluates the performance of the instance segmentation trained with cut-paste images. For this purpose, the models from Sect. 4.3 are tested using real-world images, and the COCO evaluation metrics [21] are computed.

One test dataset of about 80 images with diverse scenes and backgrounds is available for each part type. The test images of the rigid parts are captured with the Google Pixel 4a smartphone, while cables [22] and hoses are imaged with the industrial camera *rc_visard 65 color* from Roboception. The corresponding labels are prepared manually using Hasty[4] and labelme[5].

For evaluation, the segmentation results from the models are compared against the ground truth labels. Thus, the intersection over union (IoU) of the segmentation result and the corresponding ground truth label is calculated. Based on a threshold, the results are classified as true or false to compute

[1] https://www.remove.bg/

[2] https://opencv.org/

[3] https://numpy.org/

[4] https://hasty.ai/

[5] https://github.com/wkentaro/labelme

Table 2 Instance segmentation results for real-world test images with parts in front of various backgrounds

Part type	$AP_{0.50}$	$AP_{0.50:0.95}$	$AR_{0.50:0.95}$
Cylinder	0.989	0.952	0.969
Plate	0.988	0.889	0.900
USB	0.977	0.699	0.742
Cable	0.869	0.261	0.368
Hose	0.973	0.499	0.599

precision and recall. The average precision $AP_{0.50}$ applies an IoU threshold of 50 %. For the computation of $AP_{0.50:0.95}$ and $AR_{0.50:0.95}$, the results of all APs and ARs between 50 % and 95 % with an interval of 5 % are averaged. Table 2 shows the quantitative results of this experiment for all five part types using the widely accepted AP and AR metrics.

Instance segmentation works best for cylinders and plates. This demonstrates that precise segmentation based on the proposed cut-paste method is feasible. In comparison, the results for hoses and especially for cables are lower. This difference is even stronger for $AP_{0.50:0.95}$ and $AR_{0.50:0.95}$.

Figure 4 shows exemplary mask predictions for the instance segmentation of all types. In the test dataset, all cylinders and plates are localized and no distractors are segmented despite similar shape or texture. On the other hand, a few USB drives are not segmented at all. If the cylinders and plates are placed in front of a metallic background such as a punched plate or slotted table, few masks protrude beyond the edges. For cables, false positive instances occur due to the groove slots in the background, which have similar dimensions. In contrast, there are much less false positive pixels for hoses. However, similar to cables, several submasks are predicted for one hose instance. Overall, small segmentation errors occur in the contour area of all part types. For example, in plenty cases the predicted mask is scaled by a few pixels and thus, too large. This results in a number of false positive pixels along the instance contour.

4.5 Experiment 2: robotic picking

Following the evaluation on test images, the segmentation models are integrated into a robotic system for practical picking experiments. This allows to evaluate the performance of the cut-paste method for industrial picking applications. The following describes the hardware setup and picking routine, before the experimental setup and results are shown.

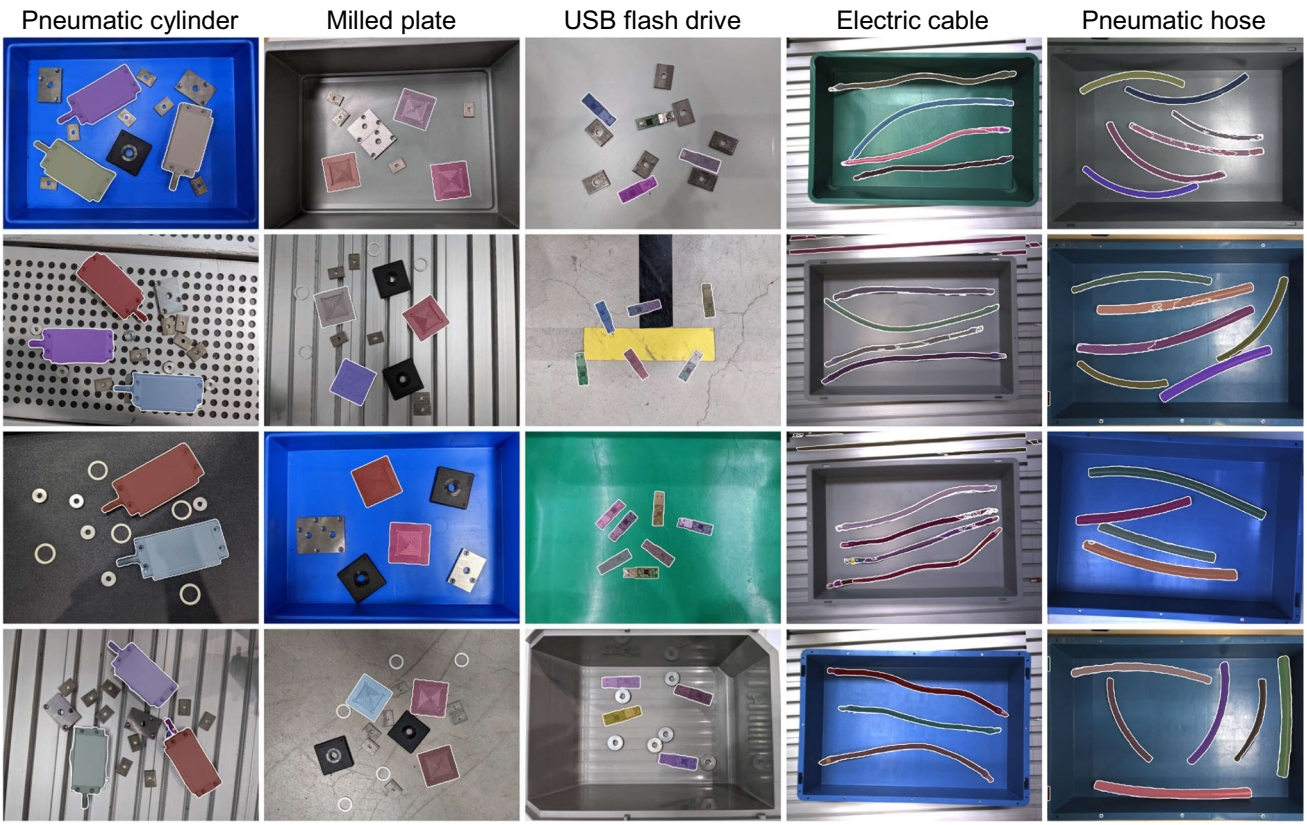

Fig. 4 Exemplary instance segmentation results for scenes with pneumatic cylinders, milled plates, USB flash drives, electric cables, and pneumatic hoses in front of diverse backgrounds

The hardware setup is defined by the boundary conditions of the presented cut-paste method. In order to image the scene, the industrial camera *rc_visard 65* is mounted above the region of interest (ROI). Grasping and handling are performed with the *Universal Robots UR10e* and the two-finger gripper *Robotiq 2F-85*. In addition, a non-commercial vacuum gripper with three suction cups is used. Figures 1 and 5 show the experimental setup with the two-finger gripper and the vacuum gripper, respectively. Computation is performed on an industrial computer (Intel Xeon Core i7 and NVIDIA Quadro RTX 4000 8GB).

During the picking routine, first, the industrial camera takes a 2D image. Then, the trained model outlined in Sect. 4.3 performs instance segmentation to output the instance contours C of the parts in the image. Given the erroneous submasks (Sect. 4.4), instances with a circumference smaller than 150 pixels are sorted out during mask generation. By implementation of the Algorithm 1 the grasp poses P_i are computed. For the DLOs, grasp poses with a larger distance to the nearest instance are preferably selected to prevent collision of the gripper. With the help of the predefined z-coordinate, the grasp pose P_p is transformed from 2D pixel coordinates into 3D real-world coordinates. Finally, the robot is navigated there to pick one part out of the ROI. This routine is repeated until no parts remain in the ROI.

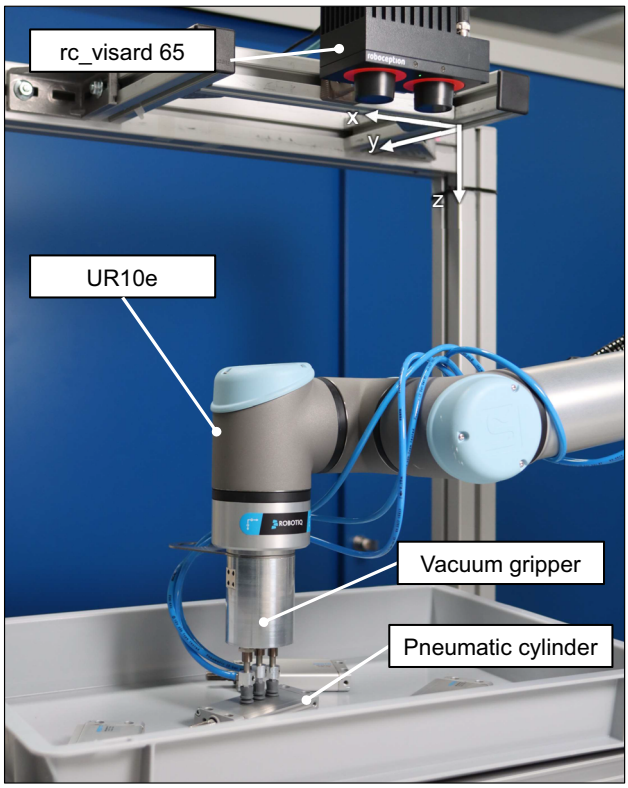

Fig. 5 Technical setup and hardware components for practical evaluation with the robotic picking experiment

For the experiment, scenes with four to six parts are set up. In each new scene, the number and arrangement of parts as well as the small load carriers and backgrounds are altered. In addition, the type and number of distractors are varied for the rigid parts. For DLOs new variants, which are not shown in the source images, are added to the scenes.

In total 100 picking trials are performed for each part type. A pick is considered successful if the robot removed the part from the ROI. All other cases are classified as unsuccessful, e.g., trials without segmentation results or grasp pose and grasping failures. For such failed trials, the affected part is removed manually and consequently, the picking routine is continued. This permits only one picking trial per instance in the initial scene. Table 3 summarizes the grippers applied for picking and which scenes contain distractors. In addition, the share of successful picks is shown for each part type.

These results show that all cylinders and almost all plates could be removed from the ROI using the vacuum gripper. By applying the two-jaw gripper, about 90 % of the DLOs could be picked successfully. Examples of successful picking sequences with cables and hoses are shown in Fig. 6. It includes camera images and segmentation masks with corresponding grasp poses before each pick. Thereby, the grasp pose of the next pick is highlighted. Compared to DLOs the system performs significantly worse for picking USB drives because in hardly any scene all instances are segmented. Exemplary Fig. 6 shows one USB drive left of the image center that is not detected in any of the frames. Due to the failed segmentation, no picking of this instance is possible. One explanation for the high number of undetected USB drives is that some instances reflect due to the lighting in the bin picking setup. These reflections hardly occur in the preceding training and test images. The different gripper types show no significant effect on the picking performance because the failed attempts are almost entirely caused by absent segmentation masks.

Table 3 Properties and results of the experiment 2: Gripper type applied for picking trials, presence of distractors in the scenes, and share of successful picks out of 100 trials

Part type	Gripper	Distractors	Success rate
Cylinder	Vacuum	Yes	100%
Plate	Vacuum	Yes	99%
USB	Two-finger	Yes	65%
Cable	Two-finger	No	89%
Hose	Two-finger	No	94%

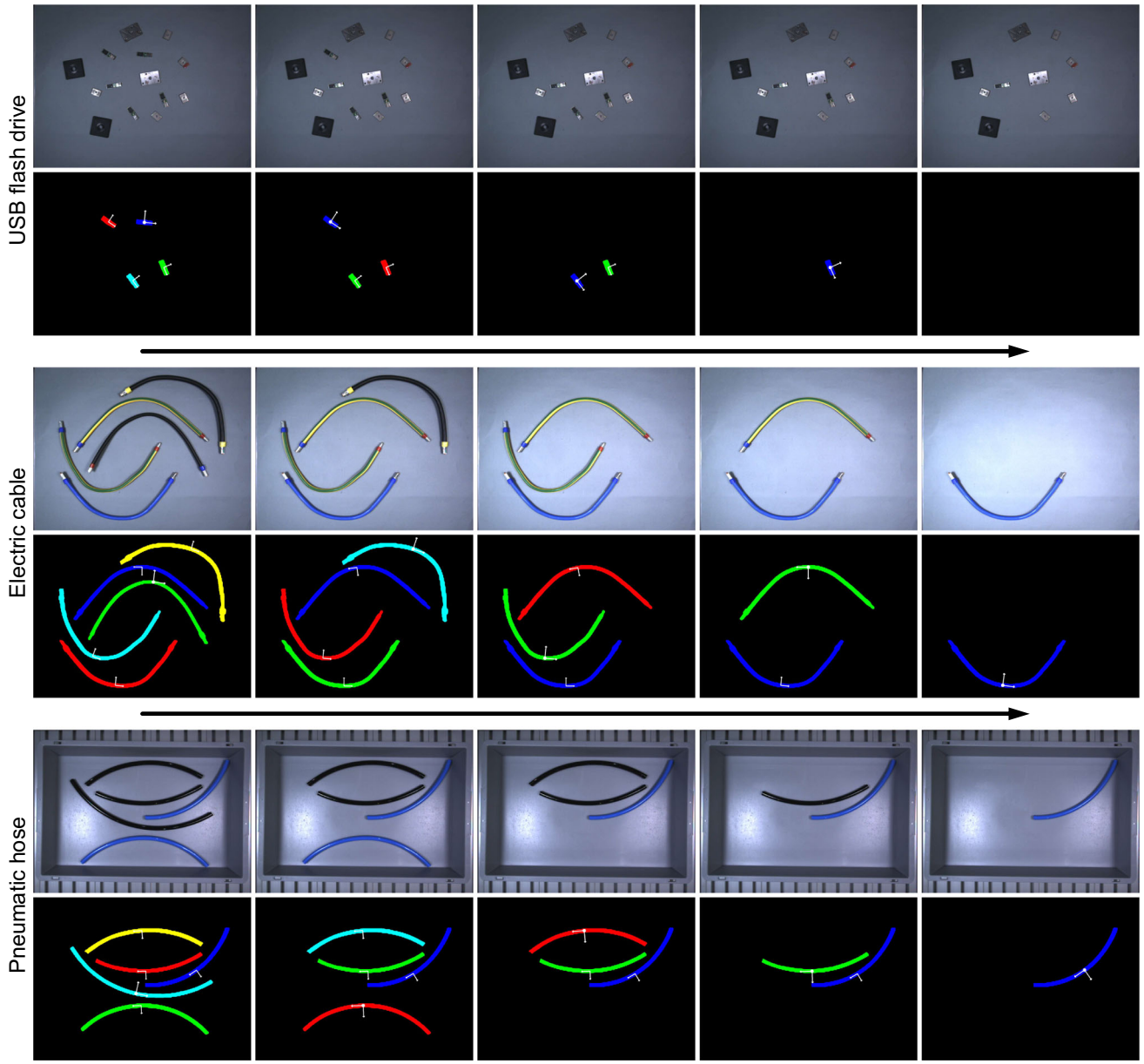

Fig. 6 Sequential input images from the industrial camera and output instance masks with grasp poses during the iterative picking procedure: The routine starts with five parts in the first image for each use case. For electric cables and pneumatic hoses, all instances are segmented and removed from the ROI iteratively. For USB flash drives, one instance is not detected at all. Consequently, it cannot be picked, causing the last attempt to fail. The images and masks show the scenes before each pick

4.6 Discussion and limitations

The presented boundary conditions (Sect. 3.2) ensure that only minor perspective effects arise from the part's pose in the image. In addition, the flat and slim parts typically have a small number of preferred orientations around the x- and y-axis, so that fewer perspective views of the parts occur during the application. Thus, the number of source images can be greatly reduced compared to the state of the art, where many different views on volume objects are mapped in the source images [5].

The quantitative results for the instance segmentation indicate that the presented cut-paste method is well suited for the segmentation of industrial parts fulfilling these boundary conditions. Nevertheless, the results vary strongly among the five-part types, especially for higher IoU thresholds. It is evident from the scenes with cables and hoses that the segmentation is less precise for the DLO use cases. Possible

reasons for these poorer results are the DLO-specific properties, the diversity of variants, and the change of the camera system between source and test images. The observed subpart masks are common and a known issue for the segmentation of slim objects such as DLOs with general models [13].

The false-positive segmentation results along the instance contour possibly arise from inaccuracies in the manually labeled ground truth masks and the blending of objects during the paste step. When evaluating with IoU-based metrics, it should be taken into account that the IoU is biased towards large objects [23]. Thus, for a dilation error with the same number of pixels, smaller types such as the USB drives are evaluated worse than the larger cylinders and plates [23].

In the preceding work in [24], synthetic training data is generated by simulation and rendering with blender. There, the same electric cable dataset is used for testing as in Sect. 4.4 and the following results are obtained: 0.979 for $AP_{0.50}$, 0.574 for $AP_{0.50:0.95}$ and 0.637 for $AR_{0.50:0.95}$. These instance segmentation results based on [24] are better than the results based on the cut-paste method, especially for higher IoU thresholds. A possible reason for this is that the simulation can generate many expressions for the DLO shape and deformation. In comparison, the shape and deformation of the electric cables in the synthetic images of the cut-paste method are limited to the spectrum of the few source images. However, the cut-paste method has the advantage over the simulation and rendering in [24] that it is much simpler to implement and does not require expert knowledge to apply. In summary, an automatic pipeline for the simulation and rendering of synthetic training data can be set up with more effort to achieve better results. On the other hand, the cut-paste method can be implemented with little effort, but still requires some minor manual effort and performs slightly worse. Due to their specific advantages and disadvantages, it strongly depends on the use case which of the two methods is more suitable.

The second experiment demonstrates that a robotic system can be enabled for robust picking even with a small number of source images and a change of the camera system between source image acquisition and testing. The results for DLOs show that variants, which are similar but not identical to the source images, can still be segmented and picked robustly. Nevertheless, the characteristic properties of DLOs and the variety of variants pose additional challenges compared to the rigid and uniform cylinders and plates. A possible reason for the poor picking results for UBS drives, besides the already worse performance on the test images (Table 2), is that the connector has specular or diffuse reflection depending on the ambient light. On the one hand, this changes the parts' appearance compared to the source images. On the other hand, the automatic exposure is affected by highlights in the image, which can change the brightness of the entire image. Thus, strongly reflective parts and significant changes of the settings between source images and applications limit the usage of the proposed cut-paste method.

Comparing the two experiments, the first evaluates the masks of all instances in a single image per scene with an IoU-based metric. In contrast, during picking only one part needs to be segmented sufficiently well per picking trial. Due to this difference, the conclusions from the instance segmentation results (Sect. 4.4) to the picking performance (Sect. 4.5) are limited. In addition, a comparison of the part types performance in both experiments is not significant, because different attributes apply for the use cases.

5 Conclusion

Generation of training data for CNN-based instance segmentation is a challenge, especially for industrial applications, because specific datasets are required for specific part types. The manual effort required for acquiring and inspecting several hundred to thousands of source images with existing cut-paste methods is not feasible for industrial end-users. To address this challenge, we present boundary conditions for the cut-paste method. With these constraints, only a small number of source images is required, so the manual effort is significantly reduced. Although the introduced boundary conditions limit the applicability of the proposed method, they cover a wide range of industrial applications such as bin picking.

The first experiment indicates that the presented method is effective in generating training data for various part types. For DLOs, however, less precise masks are obtained compared to the rigid parts. The second experiment demonstrates that using instance segmentation, which was trained exclusively with synthetic data, and the procedure for grasp pose computation, about 90 % and more of the instances can be picked in the first attempt. By filtering unfavorable masks, the presented approach is also applicable to DLOs.

Not having to use the same camera system for source and test images, smartphones can be used to capture the former even for industrial applications. Thus, an end-user can take source images independently of the application and does not require special technical setups or expert knowledge. Due to the reduced number of source images, the manual effort required to prepare one dataset is only about 20 minutes and is thus significantly lower compared to preliminary work. Limitations of the presented cut-paste method appear for the USB drives, most likely due to their partly specular reflection.

In future work, we will extend the grasp pose computation to test the presented method for overlapping objects and more DLO types. In addition, we will investigate whether the presented method can be combined with the pipeline for the simulation and rendering [24] to take advantage of both approaches.

Acknowledgements We thank Cong Xu B.Sc. and Julius Deyle B.Sc. for their engaged support during experiment preparation and execution.

Author contribution JD: conceptualization, methodology, software, validation, investigation, data curation, writing—original draft, visualization, project administration. JCB: software, validation, formal analysis, data curation, writing—review and editing. DG: methodology, writing—review and editing. RD: resources, supervision, funding acquisition.

Funding Open Access funding enabled and organized by Projekt DEAL. The research leading to this publication has received funding from the Bavarian Ministry of Economic Affairs, Regional Development, and Energy (StMWi), as part of the project 'RoMaFo' (DIK-1908-0002// DIK0109/01).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Grard M (2019) Generic instance segmentation for object-oriented bin-picking. Dissertation, Université de Lyon, Lyon. https://tel.archives-ouvertes.fr/tel-03081227
2. Denninger M, Sundermeyer M, Winkelbauer D, Olefir D, Hodan T, Zidan Y, Elbadrawy M, Knauer M, Katam H, Lodhi A (2020) BlenderProc: Reducing the reality gap with photorealistic rendering. In: International conference on robotics: sciene and systems, RSS 2020
3. Hinterstoisser S, Lepetit V, Wohlhart P, Konolige K (2018) On pre-trained image features and synthetic images for deep learning. In: European conference on computer vision – ECCV 2018 workshops, pp 682–697.https://doi.org/10.1007/978-3-030-11009-3_42
4. Eversberg L, Lambrecht J (2021) Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization. Sensors 21(23). https://doi.org/10.3390/s21237901
5. Dwibedi D, Misra I, Hebert M (2017) Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: 2017 IEEE International conference on computer vision (ICCV), pp 1310–1319. https://doi.org/10.1109/ICCV.2017.146
6. Tsirikoglou A, Eilertsen G, Unger J (2020) A survey of image synthesis methods for visual machine learning. Comput Graph Forum 39(6):426–451. https://doi.org/10.1111/cgf.14047
7. Toda Y, Okura F, Ito J, Okada S, Kinoshita T, Tsuji H, Saisho D (2020) Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. Commun Biol 3(173). https://doi.org/10.1038/s42003-020-0905-5
8. Magaña A, Wu H, Bauer P, Reinhart G (2020) Posenetwork: Pipeline for the automated generation of synthetic training data and cnn for object detection, segmentation, and orientation estimation. In: 2020 25th IEEE International conference on emerging technologies and factory automation (ETFA), pp 587–594. https://doi.org/10.1109/ETFA46521.2020.9212064
9. Georgakis G, Mousavian A, Berg AC, Kosecka J (2017) Synthesizing training data for object detection in indoor scenes. Robot Sci Syst (RSS). https://doi.org/10.15607/RSS.2017.XIII.043
10. Naumann A, Hertlein F, Zhou B, Dorr L, Furmans K (2022) Scrape, cut, paste and learn: Automated dataset generation applied to parcel logistics. In: 2022 21st IEEE International conference on machine learning and applications (ICMLA), pp 1026–1031. https://doi.org/10.1109/ICMLA55696.2022.00171
11. Block L, Raiser A, Schön L, Braun F, Riedel O (2022) ImageBot: Generating synthetic object detection datasets for small and medium-sized manufacturing companies. Procedia CIRP 107:434–439. https://doi.org/10.1016/j.procir.2022.05.004
12. Zanella R, Caporali A, Tadaka K, Gregorio D, Palli G (2021) Auto-generated wires dataset for semantic segmentation with domain-independence. In: 2021 International conference on computer, control and robotics (ICCCR), pp 292–298. https://doi.org/10.1109/ICCCR49711.2021.9349395
13. Feng Y, Yang B, Li X, Fu C-W, Cao R, Chen K, Dou Q, Wei M, Liu Y-H, Heng P-A (2022) Towards robust part-aware instance segmentation for industrial bin picking. In: 2022 International conference on robotics and automation (ICRA), pp 405–411. https://doi.org/10.1109/ICRA46639.2022.9811728
14. Sapp B, Saxena A, Ng AY (2008) A fast data collection and augmentation procedure for object recognition. In: Proceedings of the twenty-third conference on artificial intelligence (AAAI), pp 1402–1408
15. Dvornik N, Mairal J, Schmid C (2018) Modeling visual context is key to augmenting object detection datasets. Computer vision - ECCV 2018:375–391. https://doi.org/10.1007/978-3-030-01258-8_23
16. Schoepflin D, Holst D, Gomse M, Schüppstuhl T (2021) Synthetic training data generation for visual object identification on load carriers. Procedia CIRP 104:1257–1262. https://doi.org/10.1016/j.procir.2021.11.211
17. Waqas Zamir S, Arora A, Gupta A, Khan S, Sun G, Shahbaz Khan F, Zhu F, Shao L, Xia G-S, Bai X (2019) iSAID: A large-scale dataset for instance segmentation in aerial images. https://doi.org/10.48550/arXiv.1905.12886
18. Dirr J, Siepmann A, Gebauer D, Daub R (2023) Evaluation metric for instance segmentation in robotic grasping of deformable linear objects. Procedia CIRP 120. https://doi.org/10.1016/j.procir.2023.09.066
19. Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, Sun S, Feng W, Liu Z, Xu J, Zhang Z, Cheng D, Zhu C, Cheng T, Zhao Q, Li B, Lu X, Zhu R, Wu Y, Dai J, Wang J, Shi J, Ouyang W, Loy CC, Lin D (2019) MMDetection: Open mmlab detection toolbox and benchmark. https://doi.org/10.48550/arXiv.1906.07155
20. Wang X, Zhang R, Kong T, Li L, Shen C (2020) SOLOv2: Dynamic and fast instance segmentation. In: Advances in neural information processing systems (NeurIPS), vol 33, pp 17721–17732. https://doi.org/10.48550/arXiv.2003.10152
21. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: Common objects in context. European conference on computer vision - ECCV 2014:740–755. https://doi.org/10.1007/978-3-319-10602-1_48
22. Dirr J, Yao J, Siepmann A, Gebauer D, Daub R (2022) Dataset for instance segmentation of deformable linear objects. Dataset, Tech-

nical University of Munich, Munich. https://doi.org/10.14459/2022mp1690303
23. Cheng B, Girshick R, Dollár P, Berg AC, Kirillov A (2021) Boundary IoU: Improving object-centric image segmentation evaluation. In: 2021 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 15329–15337. https://doi.org/10.1109/CVPR46437.2021.01508
24. Dirr J, Gebauer D, Yao J, Daub R (2023) Automatic image generation pipeline for instance segmentation of deformable linear objects. Sensors 23(6). https://doi.org/10.3390/s23063013