A Weakly Supervised Semi-Automatic Image Labeling Approach for Deformable Linear Objects

Alessio Caporali, Matteo Pantano, Lucas Janisch, Daniel Regulin, Gianluca Palli, *Senior Member, IEEE*, and Dongheui Lee, *Senior Member, IEEE*

***Abstract*—The presence of Deformable Linear Objects (DLOs) such as wires, cables or ropes in our everyday life is massive. However, the applicability of robotic solutions to DLOs is still marginal due to the many challenges involved in their perception. In this letter, a methodology to generate datasets from a mixture of synthetic and real samples for the training of DLOs segmentation approaches is thus presented. The method is composed of two steps. First, key-points along a real-world DLO are labeled by employing a VR tracker operated by a user. Second, synthetic and real-world datasets are mixed for the training of semantic and instance segmentation deep learning algorithms to study the benefit of real-world data in DLOs segmentation. To validate this method a user study and a parameter study are conducted. The results show that the VR tracker labeling is usable as other labeling techniques but reduces the number of clicks. Moreover, mixing real-world and synthetic DLOs data can improve the IoU score of a semantic segmentation algorithm by circa 5%. Therefore, this work demonstrates that labeling real-world data via a VR tracker can be done quickly and, if the real-world data are mixed with synthetic data, the performances of segmentation algorithms for DLOs can be improved.**

***Index Terms*—Deformable linear objects, dataset generation, spatial labeling, usability, image segmentation.**

Manuscript received 2 September 2022; accepted 23 December 2022. Date of publication 6 January 2023; date of current version 17 January 2023. This letter was recommended for publication by Associate Editor Z. Min and Editor C. C. Lerma upon evaluation of the reviewers' comments. This work was supported in part by European Commission's Horizon 2020 Framework Programme With the Project REMODEL - Robotic Technologies for the Manipulation of Complex Deformable Linear Objects - under Grant 870133 and in part by Siemens AG. *(Alessio Caporali and Matteo Pantano contributed equally to this work.) (Corresponding author: Alessio Caporali; Matteo Pantano).*

Alessio Caporali is with the Department of Electrical, Electronic and Information Engineering, University of Bologna, IT-40136 Bologna, Italy (e-mail: alessio.caporali2@unibo.it).

Matteo Pantano is with Technology Department, Siemens Aktiengesellschaft, D-81739 Munich, Germany, and also with the Human-Centered Assistive Robotics, Department of Electrical and Computer Engineering, Technical University of Munich, D-80333 Munich, Germany (e-mail: matteo.pantano@siemens.com).

Lucas Janisch is with Technology Department, Siemens Aktiengesellschaft, D-81739 Munich, Germany, and also with the Institute for Factory Automation and Production Systems, Friedrich-Alexander-Universität Erlangen-Nürnberg, D-91054 Erlangen, Germany (e-mail: lucas.janisch@faps.fau.de).

Daniel Regulin is with Technology Department, Siemens Aktiengesellschaft, D-81739 Munich, Germany (e-mail: daniel.regulin@siemens.com).

Gianluca Palli is with the Department of Electrical, Electronic and Information Engineering, University of Bologna, IT-40136 Bologna, Italy (e-mail: gianluca.palli@unibo.it).

Dongheui Lee is with the Autonomous Systems, Technische Universität Wien, AT-1040 Vienna, Austria, and also with the Institute of Robotics and Mechatronics, German Aerospace Center, D-82234 Wessling, Germany (e-mail: dongheui.lee@tuwien.ac.at).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3234799, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3234799

I. Introduction

Deformable Linear Objects (DLOs) are part of our everyday life in the form of wires, cables, ropes, tubes, and many more. DLOs are characterized by an elongated cylindrical shape lacking relevant features. Therefore, their detection via Computer Vision (CV) is challenging, and the application of robotics aiming at their autonomous manipulation is heavily affected [1], [2], [3]. In fact, the cabling and wire routing processes, which constitute a huge part of industrial assembly tasks, are still largely performed by humans in the automotive and aerospace sectors [4], [5]. Consequently, there is a need to propose new perception methodologies based on CV to bridge this gap [2], [3], [6], [7]. Existing data-driven approaches, in real-world applications, are still massively affected by the quality and size of the datasets. More precisely, in the DLOs domain, there are two main needs as identified by [8]. First, how to generate a real-world dataset for DLOs. Second, investigate if a mix of real and synthetic data is impacting data-driven segmentation learning. Therefore, our contributions are twofold. First, a novel weakly supervised method to collect real-world pixel-wise annotated DLOs samples starting from imprecise user input given by a VR tracker (Fig. 1) is proposed. Second, a thorough analysis of segmentation improvements given by various levels of mix between synthetic and real datasets for DLOs is provided.

To present our contributions the remainder of this letter is organized as follows. Section II provides an overview of the current literature available concerning the perception of DLOs and existing labeling approaches. Sec III presents the methodologies employed to obtain the synthetic samples and the real ones, source code is made available[1]. Section IV outlines the experimental methodologies employed whereas Section V provides a in-depth discussion of the results. Section VI gives the concluding remarks and future research directions.

II. Related Works

A. Segmentation of DLOs

In the past, the problem of DLOs identification has been solved in simple settings like in [1], [9] and [10] where the segmentation is performed with color-based thresholds or controlled backgrounds. Even simpler approaches, common in the past, reside in the application of "filters" to highlight tubular structures in images, as the case of the *Frangi* filter [11], the *Ridge* filter [12] and the ELSD [13] algorithm. Both the *Frangi* and *Ridge* filters are based on the Hessian matrix, while ELSD

[1] https://github.com/lar-unibo/DLO-WSL

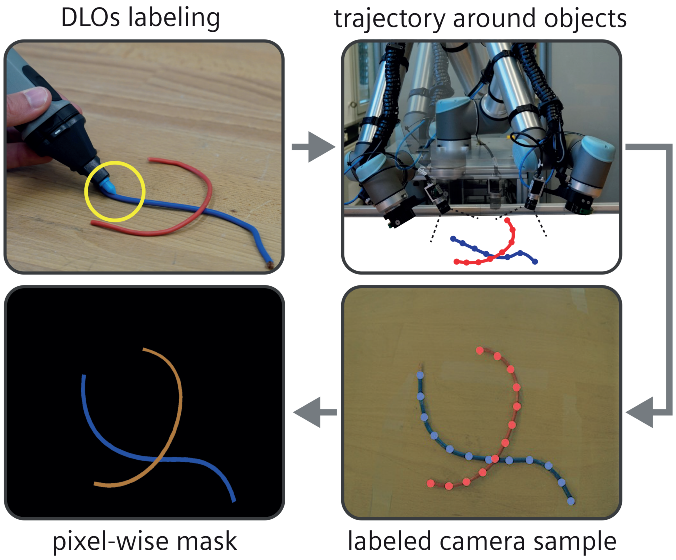


Fig. 1. VR tracker-based weakly supervised dataset generation.

is an algorithm developed to detect line segments and elliptical arcs. Many false positives are detected in case of complex backgrounds [6] when employing these approaches.

The advances of deep learning approaches have brought significant improvements in the context of DLO segmentation. For instance, the semantic segmentation of wires and cables via learning-based methods has been attempted in [2] where a dataset consisting of electric wires obtained with a chroma-key approach is made publicly available. However, for subsequent automatic handling of DLOs, in some cases it is necessary to distinguish the instances, e.g. if cables are on top of each other. Therefore, a more advanced method is *Ariadne*+ [7]. It achieves the instance segmentation of DLOs empowering a segmentation Deep Convolutional Neural Network (DCNN) [14] in its framework. Recently, *FASTDLO* [3] has been proposed improving the performances of *Ariadne*+ both in terms of speed and accuracy.

In the context of general-purpose learning-based instance segmentation methods [15], [16], [17], [18], the major problem in their application to DLOs resides in the lack of publicly available high-quality datasets and, consequently, the difficulty in annotating a large set of images. Compared to [2], our approach is capable of performing both a semantic labeling and an instance-level one. In addition, the captured sample is entirely real and not obtained by mixing a real DLO with a post-processed background image, as the case in [2].

B. Image Labeling

The efficient generation of ground truth segmentation masks, in general, is explored in many studies. The research area is primarily motivated by the need for large data variability in training datasets to achieve performance saturation [19]. Furthermore, supervised methods, i.e., those trained on manually annotated data, currently best perform in public benchmarks [20]. However, the manual labeling process can become burdensome with large datasets. Therefore, to make the image labeling process more efficient, a broad range of methods come into consideration.

To minimize the need for pixel-wise annotated training data, the development of weakly supervised training methods is one of the most active research fields. Here, all kinds of weak supervision sources are explored to generate segmentation masks, including image labels [21], point clicks [22], bounding boxes [23], or scribbles [24]. Alternatively, saliency detection can be employed [25], but it cannot differentiate among objects' instances. Furthermore, the performances of models trained in these ways are still significantly worse than that of models trained with fully annotated masks [20]. To further correct faulty detections from weakly annotated data, interactive segmentation is introduced. In [26] an iterative method to generate a pixel-wise mask based on an initial human-annotated bounding box and improved with corrective clicks on the predicted mask is proposed. Another industrial weakly supervised labeling tool is introduced in [27] where the image is first oversegmentated in superpixels and the user must label only a few superpixels per class. Then, based on a superpixel similarity metric, an algorithm annotates the remaining superpixels. Another approach is an automatic adjustment of the given weak noisy labels which are corrected based on specific knowledge about their generation process [28]. In the work of [29] this approach is applied by using gradient guidance to automatically correct manual annotations of edge positions. A different approach, however, is proposed with Reviving Iterative Training with Mask Guidance for Interactive Segmentation (RITM) in [30]. In this work, the authors show that using pre-trained networks to infer object masks after the user labels a few points can diminish the labeling effort. However, despite these good results, all these methods apply to single images. Therefore, they are difficult to scale to big datasets while limiting user label points.

A widely adopted strategy for efficient high-volume labeled data generation is to produce synthetically rendered images. With this approach, the generation of a virtually unlimited number of synthetic and photo-realistic images encompassing various types of objects with different sizes, shapes, compositions and 3D distributions is enabled. Thereby, the respective synthetic ground truth segmentation mask is automatically derived for each generated image reaching a segmentation accuracy comparable to human-made labeling. This method is already applied to the problem of three-dimensional pose estimation of DLOs tips by [31] but is also used in related research areas [32], [33], [34], [35], [36], [37]. However, simulated data can cause a domain gap which refers to a loss in model performance due to a difference from training data to test data [38].

Therefore, in this work, we present the generation of both instance-wise labeled synthetic and real-world data for big datasets, and we study the effects of this mixed training.

III. Methodology

A. Problem Statement

As described in the introduction, the goal of this research is to provide a methodology to generate a mix of real-world and synthetic data and show the impact given by mixing the two obtained datasets. The mathematical definition related to this problem is also known as domain adaptation and it is already well-documented in [35]. Therefore, this letter focuses on the challenge for DLOs. First, the generation of synthetic and real-world datasets are described in Section III-B and Section III-C. Second, to study the impact given by mixing two datasets, we employ the End-to-End (e2e) training approach. This approach is selected as long as the synthetic dataset is similar to the real-world one [39]. Our problem is formulated as training on a dataset mixture obtained by different ratios of real-world and synthetic data, specifically:

$$F(x) = w_r P_r(x) + w_s P_s(x) \tag{1}$$

where, w_r and w_s are the ratios between real-world and synthetic datasets with $w_r = 1 - w_s$, $P_r(x)$ and $P_s(x)$ representing the distributions of the real-world and synthetic dataset respectively, and $F(x)$ is the resulting distribution. Therefore, during training, samples extracted from $F(x)$ are used to optimize the learning models. In our research we address how different w_r and w_s impact the segmentation performances, this is described in Section III-D.

B. Synthetic Dataset Sampling

By modeling DLOs as spline curves and by employing a rendering engine it is possible to sample photo-realistic synthetic images of DLOs closely matching real ones. A generic DLO shape can be represented in the Cartesian space by a 3rd-order spline basis as a function of a free coordinate u representing the position along the cable starting from an endpoint ($u = 0$) to the opposite end ($u = L$) being L the length of the DLO. That is:

$$q(u) = \sum_{i=1}^{n} b_i(u) q_i \tag{2}$$

where $q(u) = [x(u)\ \ y(u)\ \ z(u)]^T$ is the vector of Cartesian coordinates of each point along the DLO, $b_i(u)$ is the i-th elements of the spline polynomial basis used to represent the DLO shape and q_i are n properly selected coefficients, usually called *control points*, used to interpolate the DLO shape through the $b_i(u)$ function basis. Therefore, the set of control points is constructed with an iterative propagation method. Being p_t the last element of the ordered set of already generated control points, the new point p_{t+1} is defined as $p_{t+1} = p_t + s\,d$ where s is the propagation step randomly selected between two boundary values and d describes a random direction vector pointing forward with respect to the existing sequence of points. The z component of the point is clamped between boundary values to make the obtained curve follow a real DLO shape more. Therefore, a random sequence of points is generated and then interpolated by a spline curve as defined in (2).

Concerning the rendering of the synthetic images, a novel pipeline making use of Blender™ is exploited [40]. A mesh object is created from the generated spline-based DLO model by specifying the DLO thickness, color and stripes. These attributes can be also made random to increase the variance of the dataset. Therefore, a random texture is chosen as the background along with random lighting settings. In this way, different combinations of shadows can be simulated. All these expedients are needed to enhance the generalization capabilities of the data-driven approaches. Along with the image, the rendering pipeline allows the creation of label data (i.e., mask image). As an example, in Fig. 2 some generated images with the proposed procedure and corresponding labels are shown.

C. Real-World Dataset Sampling

To enable the labeling of DLOs in a real environment through user input, several elements are required. In particular, at first, the dataset is recorded as outlined in Section III-C1. Then, instances of DLOs are labeled via a VR tracker as described in Section III-C2. Finally, the generated labels are corrected to account for possible calibration and user input errors as illustrated in Section III-C3. We denote this labeling method as *DLO-WSL*.

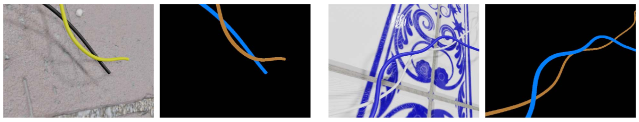


Fig. 2. Synthetically generated RGB samples and labels.

1) Recording of Images With a Robot: To create a dataset via *DLO-WSL*, images along with the position of the camera in the world coordinate system must be known. Therefore, a 2D RGB camera is mounted on the flange of a robotic arm in an eye-in-hand configuration as shown from Fig. 1 for a two-fold benefit. First, the recording of several images is achieved in a matter of seconds. Second, the position of the camera is known as long mechanically connected to the robot. Nonetheless, to achieve these benefits, the camera position had to be known with respect to the mounting screws and the robot trajectory had to fit the application requirements (e.g., visible object in the camera FOV). To solve these two issues, the following strategies are included. On one hand, the transformation from the camera frame to the mounting position is obtained by the iterative camera calibration routine proposed by [41]. On the other hand, a robot trajectory with an ellipsoidal shape is integrated into the robot control to ensure that the images taken by an inward-facing camera had always the object at the center of the trajectory. This trajectory is calculated using (3) (where x, y, z are the trajectory points, a, b, c are the ellipsoid parameters, θ is the zenith angle, ϕ is the azimuth angle, and x_0, y_0, z_0 are the coordinates of the initial position).

$$\begin{cases} x = a \sin\theta \sin\phi + x_0 \\ y = -b \sin\theta \sin\theta + y_0 \\ z = c \cos\theta + z_0 \end{cases} \tag{3}$$

2) VR Tracker Labeling: A methodology involving a sensor tracked in space is selected to label instances of DLOs. Therefore, an input methodology similar to [42] is preferred. However, a different tracking technology to both avoid the need for an additional camera to obtain the position of the tracked sensor and to provide an industrial solution has been chosen. More precisely, the Tracepen™ virtual reality (VR) pen is selected. This VR pen works on the basis of reflective photodiode sensors which, by receiving and mirroring an infrared signal, enable the calculation of the sensor's pose from the emitting station. Due to this working principle, the coordinates of the VR pen are expressed in reference to the emitting station (P_i^t) [43]. However, to obtain DLOs instance labels for the images, such positions had to be transformed in camera pixel coordinates (P_i^c). To solve this issue, homogeneous transformations between the emitting station and the camera position had to be considered as shown in Fig. 3. Hence, the approach of [44] is used and the transformation between the emitting station and the robot (${}^tT_r$) is calculated to obtain VR tracker points in the robot coordinate frame (P_i^r).

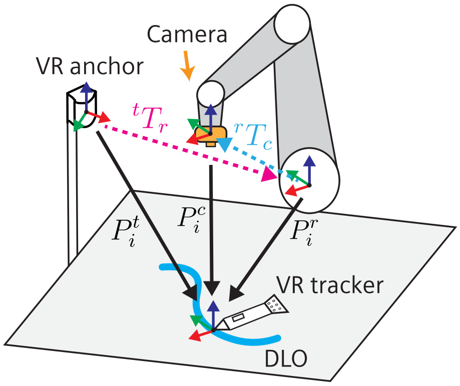


Fig. 3. Transformations involved in the labeling procedure using the VR tracker. The transformations named $^{t}T_r$, $^{r}T_c$ are needed to transform the labeled point from the VR coordinates (P_i^t) to points in the robot coordinates (P_i^r) and in the camera coordinates (P_i^c).

Afterward, such 3D points can be projected in the 2D images to generate training data. This is achieved through (4) (where u, v are pixel coordinates, w' is the scaling factor, K is the intrinsic camera matrix obtained via camera calibration, $^{r}T_c$ is the transformation from robot to camera, $[x, y, z]$ is the 3D point which needs to be projected, and $u = u'/w'$ and $v = v'/w'$).

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = K\, ^{r}T_c \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4}$$

Thus, with this approach, images of real-world scenarios with different camera positions are taken using a single labeled input. This procedure is repeated for each DLO instance to label.

3) Weakly Supervised Semi-Automatic Labeling: Unlike synthetic labeling, which is inherently error-free, during the labeling of real-world DLOs performed by a human operator, some level of error is expected. In particular, the major sources of errors are due to inaccuracies in 1) the calibration of the annotation tool and/or eye-in-hand camera; 2) the labeling performed by the human operator. The presence of errors is more evident and severe especially on very thin DLOs.

To overcome these problems, a fine-tuning step is applied after the human input to each labeled DLO instance, its main stages are shown in Fig. 4. First, the labeling points of one DLO instance are smoothed employing an approximating spline curve in the 2D pixel space, similar to the definition of (2), see Fig. 4(c).

Then, an approach based on a CNN is applied for computing a correction offset for each labeled point. Given the source image I, the vertically oriented crop extracted around the i-th labeled point and having a size of $s \times s$ pixels is denoted as $\hat{I}_i$. With vertically oriented, we denote the condition with the DLO having an almost vertical shape in $\hat{I}_i$, see input crop in Fig. 5. Thus, the CNN-based network $H(\cdot)$ performs the following operation:

$$h = H(\hat{I}_i)$$

being $h \in \mathbb{R}^s$ the vector approximating the location of the DLO in the image along the horizontal axis, see the output in Fig. 5. In other words, h describes the probability of each image column, i.e. column 0 to column $s - 1$, of corresponding to the center-line of the DLO in $\hat{I}_i$. Finally, the maximum of h is obtained as:

$$k = \text{argmax}(h) \;:\; \dot{h}_k = 0$$

being $\dot{h}_k$ the derivative of h evaluated at point k. Hence, the correcting offset δ for the crop $\hat{I}_i$ is computed as $\delta = k - s/2$, being s the crop size fixed to be 96 in the following.

The CNN structure is composed of a feature extractor, i.e. ResNet-18 [45], and two Fully Connected linear layers, i.e. $FC_{512,256}$ and $FC_{256,96}$. Binary cross-entropy is used as loss function during the training stage to optimize the network weights.

The dataset for the training of this network is obtained from the synthetic samples of Section III-B. Thus, crops of 96 × 96 pixels are randomly extracted from the synthetic images, given the available spline description, employing a fictitious offset to simulate the user error. Hence, the offset is converted with a Gaussian distribution centered at the offset value and with a variance of 8 pixels. In total, 40000 crops are used for the optimization with the typical 90-10 split in training and validation sets. The network is optimized for 50 epochs, employing a batch size of 128 and a learning rate of 5×10^{-5}. Adam is selected as optimizer with the final network weights chosen based on the validation loss. Having corrected the points, Fig. 4(d), the knowledge of the DLO thickness is exploited to construct a polygon that precisely follows the contour of the targeted DLO in each image plane. Thereafter, from the polygon, an instance mask can be easily drawn as shown in Fig. 4(e).

TABLE I
FACTORIAL DESIGN FOR THE ANALYSIS OF DATASET MIXTURES

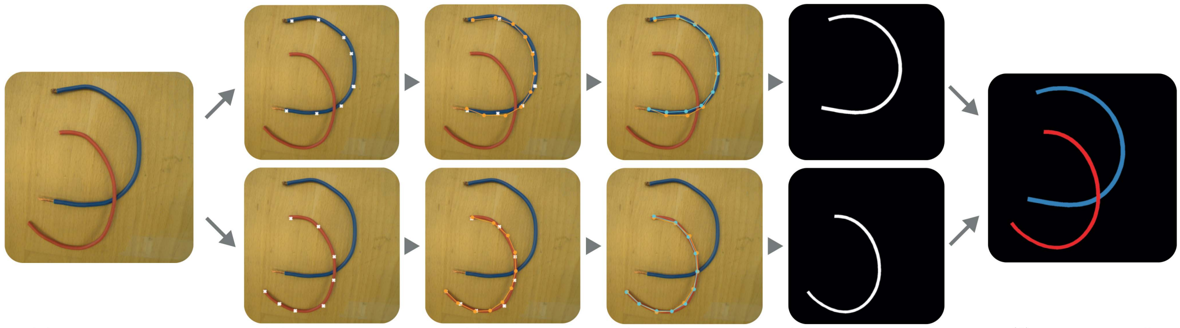
Synthetic	$w_r = 0$	0.05	0.11	0.20	0.33
4	0	-	-	$1^\dagger$	$2^\dagger$
8	0	-	$1^\dagger$	$2^\dagger$	-
16	0*	1*	2*	-	-

The table shows the size of real-world dataset given the synthetic dataset and w_r. Dataset sizes are in thousands. With† are denoted the configurations where U^2PL is trained also in a semi-supervised fashion. With * are denoted the configurations used only by DeepLabV3+ and YOLACT.

D. Training on a Dataset Mixture

For the manipulation of DLOs, the availability of a segmentation mask is fundamental [3]. Therefore, we focus on the impact given by w_s and w_r in semantic and instance segmentation tasks. With the former, our goal is to discern between the DLO class and other objects in the image by assigning a pixel-wise label. With the latter, we aim to recognize all the DLO instances in the images by assigning a unique pixel-wise label for each DLO. For the investigation of w_s and w_r, a full factorial experiment with the datasets as factors on three levels with three runs has been conceived. Specifically, the experiments with the respective w_r are shown in Table I.

For semantic segmentation task, *DeepLabV3+* [14] and U^2PL [46] are used as networks for the tests. The former is a standard fully supervised approach and allows a direct comparison with [2]. The latter is a recently proposed semi-supervised method where, in our case, the real-world dataset is used both labeled and unlabeled. In this way, we can evaluate the benefits of labeling real-world data as opposed to a semi-supervised



(a) input image (b) VR input (c) spline (d) correction (e) instance mask (f) output mask

Fig. 4. Labeling flow performed for each captured image. On the sampled image (a) the input points of the VR tracker captured for each instance are projected (b) and smoothed with a spline curve (c). Thereafter, the CNN-based correction procedure is executed (d), the instance masks generated (e), and the output mask finalized (f). Colors meaning: input tracker points as white crosses, smoothed points in orange and corrected points in cyan.

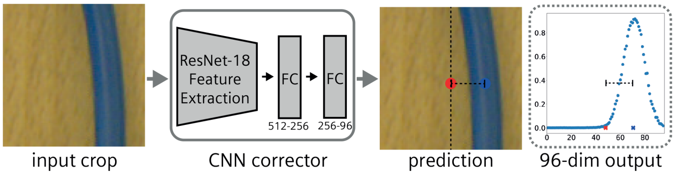
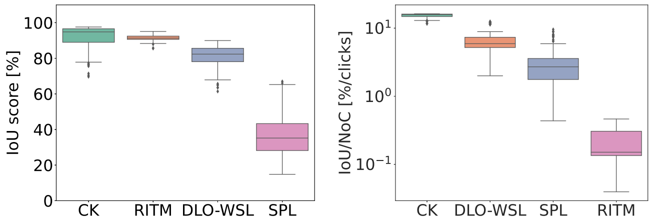


Fig. 5. CNN corrector schema. The input crop of 96 × 96 pixels is forwarded to the network outputting a 96-dim vector approximating the DLO center horizontally. Using this method, the correction offset for the user input is computed.



(a) segmentation (b) segmentation and number of clicks

Fig. 6. Evaluation of the performances in terms of semantic segmentation and number of clicks for the labeling approaches. The segmentation reports the outcomes with chroma-key (*CK*), *RITM* [30], our approach (*DLO-WSL*) and raw data from the user fitted in a spline without the CNN correction (*SPL*).

approach. *DeepLabV3+* is trained with a ResNet-101 [45] backbone with a batch size of 10, output stride of 16, separable convolutions, a polynomial learning rate of 10^{-5} with power 0.95, and Adam optimizer. Instead, U^2PL is trained with a ResNet-50 [45] backbone with a batch size of 2, a polynomial learning rate of 5×10^{-5} with power 0.9, and stochastic gradient descent (SGD) as optimizer.

For instance segmentation task, *YOLACT* [15] is selected and trained with a ResNet-101 backbone with a batch size of 6, a step learning rate of 10^{-3} with factor 0.1 at iterations 30 K and 60 K, and SGD as optimizer.

All networks are optimized for 50 epochs with the final weights selected based on the validation loss. As augmentation scheme, the following is employed: channel shuffling; hue, saturation and value randomization; flipping; perspective distortions; random cropping; random brightness and contrast.

With reference to Table I, for *DeepLabV3+* and *YOLACT*, the training mixtures consist of a total of 9 configurations. For U^2PL, due to computation burden, the synthetic 16 K configuration is avoided thus resulting in a total of 10 configurations accounting also the semi-supervised condition.

IV. Experiments

In this section, the results of two sets of experiments are reported to investigate the impact of our method on both the user experience and the labeling performances. More precisely, with these experiments we aim to answer the following:

RQ1: What is the perceived usability, required effort and labeling accuracy when using *DLO-WSL*?

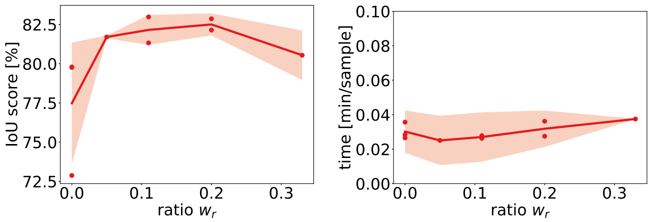
RQ2: Does the inclusion of real-world data in the synthetic dataset, through *DLO-WSL*, help the training of data-driven CV segmentation algorithms for DLOs? If yes, in which way?

A. DLO-WSL *Perceived Usability and Performances*

To evaluate the impact of labeling and find data for *RQ1*, a user test with a balanced randomized order of three subsequent interactions was envisioned. To the best of the authors' knowledge, no labeling method exists for big datasets which requires labeling for only one image with uneven backgrounds. Therefore, the comparison is done against the *chroma-key* technique for its adequacy to generate multi-image datasets with single human intervention in even backgrounds [47], and *RITM* due to its good performance in state-of-the-art single image weakly-supervised labeling [30]. For these comparisons, the users were requested to label 10 images with the three different methods. At the end of each interaction, usability and workload were measured through the System Usability Scale (SUS) [48] and the NASA-TLX [49]. Additionally, the number of clicks (NoC) to complete the labeling task was also recorded.

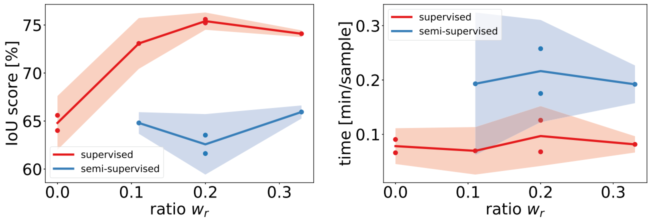
A total of 13 users, not experienced with labeling techniques, age mean (M) = 32.70 yrs, standard deviation (SD) = 9.23, participated in the study. All of them performed the test correctly and no data were discarded.

Concerning the usability, the score for *chroma-key* is M = 60.38%, SD = 21.00, for *DLO-WSL* is M=69.61%, SD=16.26, and for *RITM* is M=82.30%, SD=9.54. A Mann-Whitney-U-Test is applied for statistical difference as long the normality



(a) IoU score (b) time

Fig. 7. Evaluation of w_r on *DeepLabV3+*. To the left is the change of average IoU score with different w_r. To the right is the change of training time over dataset size.



(a) IoU score (b) time

Fig. 8. Evaluation of w_r on U^2PL with the comparison between supervised (red) and semi-supervised (blue) training.

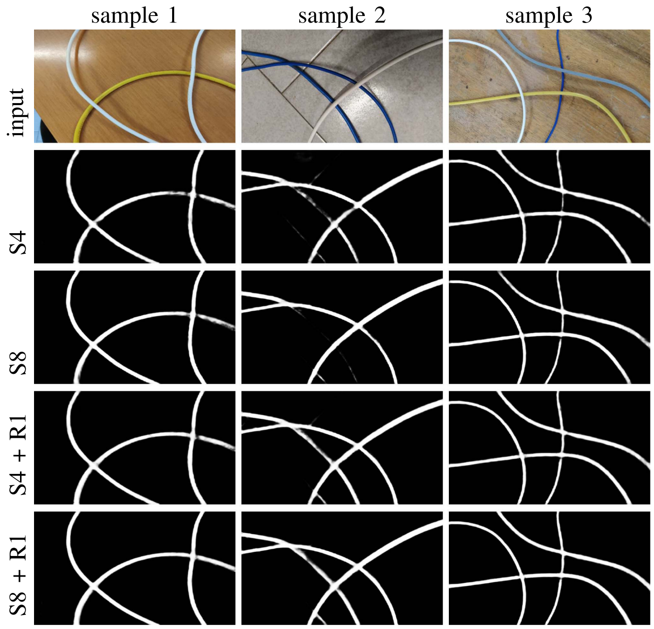


Fig. 9. Qualitative *test set* samples and predictions of *DeepLabV3+* when trained on 4 K synthetic images (S4), 8 K synthetic images (S8), a mix of 4 K synthetic and 1 K real (S4 + R1, $w_r = 0.20$) and a mix of 8 K synthetic and 1 K real (S8 + R1, $w_r = 0.11$).

pre-condition did not hold true $p > 0.05$ (CI=95%). The test reported $p > 0.05$ (CI=95%) when comparing *DLO-WSL* with the other methods. Therefore, it is possible to conclude that the usability of *DLO-WSL* is good and comparable to *chroma-key* and *RITM*.

Regarding perceived effort, the score for *chroma-key* is M=30.51%, SD=15.08, for *DLO-WSL* is M=29.74%, SD=12.96, and for *RITM* is M=22.31%, SD=13.12. A Mann-Whitney-U-Test is applied for statistical difference as long the normality pre-condition did not hold true $p > 0.05$ (CI=95%). The test reported $p > 0.05$ (CI=95%), therefore it is possible to conclude that the workload perceived in using *DLO-WSL* is comparable to the other methods.

Finally, to examine the performances of labeling, the average Intersection over Union (IoU) and IoU over the average Number of Clicks (NoC) for the dataset of 10 images are used. The results are shown in Fig. 6. To further analyze the differences across the distributions, a pair-wise Mann-Whitney-U-Test is conducted due to invalidity of the homogeneity of variance pre-condition. The test reported $p < 0.05$ (CI=95%) when comparing *DLO-WSL* with the other methods. More precisely, IoU scores were M=91.68%, SD=6.56, M=91.32%, SD=1.52, M=81.05%, SD=6.22 and M=36.88%, SD=12.45 for *chroma-key*, *RITM*, *DLO-WSL* and *spline* respectively. IoU/NoC scores were M=15.31%/clicks, SD=1.09, M=6.54%, SD=2.78, M=3.16%/clicks, SD=2.05 and M=0.21%/clicks, SD=0.13 for *chroma-key*, *DLO-WSL*, *spline* and *RITM* respectively. Thus, *DLO-WSL* obtains a good average IoU while minimizing the number of clicks for uneven backgrounds.

B. *Mixing Synthetic and Real Samples for Training*

The baseline deep learning models, datasets mixtures and training pipelines outlined in Sec.III-D are employed to find data regarding *RQ2*. As real-world *test set*, a set of 135 manually labeled real images of electrical wires with varying diameters is used from [3]. The IoU is employed as metric, with the mask M corresponding respectively to the binary mask for the semantic segmentation task and to the instance mask where each DLO instance is denoted by a unique color for the instance segmentation task. Additionally, in the latter case, the IoU score is the average score across the instances composing one image.

V. Discussions

A. DLO-WSL *Perceived Usability and Performances*

As shown from the results, the usability and the workload of *DLO-WSL* are comparable with the state-of-the-art *chroma-key* and RITM approaches. However, *DLO-WSL* enables users to label more images with good average IoU while requiring fewer clicks in uneven backgrounds as shown by the IoU/NOC metric, thus lowering the overall effort and allowing us to reply positively to *RQ1*.

Despite these results, *DLO-WSL* showed worse statistically significant performance in the average IoU of the labels, however still better than the *spline* approach. Therefore, some pitfalls might have been in the CNN fine-tuning step proposed in our method (see Section III-C3). To better evaluate its specific performances, a synthetic test-set is generated. This set consisted of 1000 randomly cropped image regions obtained according to Section III-B where a random horizontal shift of the DLO is introduced to simulate user error. Afterward, this set was used to evaluate the CNN corrector by monitoring the offset prediction error computed as $\delta = O_p - O_{gt}$, where O_p is the offset predicted and O_{gt} is the ground truth. Out of this test, the resulting error δ can be approximated by a normal distribution with M=0.05 pixels, SD=1.57. Thus, considering that the test DLOs have diameters in the range of 10-20 pixels wide, one source of error resulting in a non optimal IoU is due to the not perfect matching of the DLOs edges by the CNN fine-tuning. Therefore, future work should address this issue and improve the pixel-wise labeling of the DLOs around the contour of the instances.

B. Mixing Synthetic and Real Samples

In Fig. 7 results of the semantic segmentation task employing *DeepLabV3+* as detailed in Section III-D are shown. From the plot of Fig. 7(a), it is evident that the introduction of real images helps in achieving higher accuracy in general. For instance, employing a total of 5 K samples (4 K synthetic and 1 K real, $w_r = 0.20$) an IoU of M=82.83%, SD=0.54 is obtained compared to M=74.97%, SD=2.78 for the 4 K synthetic only dataset (statistically significant $p < 0.05, CI = 95\%$). Similarly, when employing 1 K real images on the 8 K synthetic dataset ($w_r = 0.11$), an overall higher IoU of M=83.93%, SD=0.17 is reached compared to M=81.60%, SD=0.42 for the 8 K synthetic only dataset (statistically significant $p < 0.05, CI = 95\%$). For comparison, an IoU score of 81.9% is obtained on the same *test set* when employing the dataset from [2] for training. Concerning Fig. 7(b), the training time shows a slight rise as w_r increases, as expected due to the growth of the overall dataset size.

In Fig. 9(a) qualitative analysis of the results from *DeepLabV3+* is performed. Visually, both synthetic and real samples bring important benefits. The first ones permit an accurate segmentation of the DLO thanks to the availability of high-quality error-free ground truth labels. The latter ones allow, during the training stages, to recover the DLOs texture and appearance real information that are difficult to capture in synthetic renderings. Indeed, in Fig. 9, the DLOs with complex textures and lights are better handled once the real samples are employed (e.g., yellow DLO sample 1, blue DLO sample 2). The contribution of increasing the size of the synthetic dataset is beneficial but less significant.

The evaluation of the segmentation and time performances achieved by U^2PL is instead shown in Fig. 8. Focusing on Fig. 8(a) and comparing the scores for the same values of w_r, we can observe how the semi-supervised training brings only very marginal benefits compared to the only synthetic fully-supervised one. Instead, the utilization of labels obtained exploiting *DLO-WSL* for the real samples boosts the IoU performances by about 10% with the highest result for $w_r = 0.20$. The timings of Fig. 8(b) depict how the semi-supervised training approach is much slower due to the need of constructing pseudo-labels at run-time.

Considering this we can then answer to *RQ2* saying that for DLOs trained with a mixture of real and synthetic data an optimal value of $w_r = 0.2$ seems to be beneficial. Additionally, we can conclude that w_r seems to not drastically change the training time if differences between semi-supervised and supervised training are not considered.

A similar analysis but in the context of the instance segmentation task is also addressed employing *YOLACT* as deep learning model, see Section III-D. For instance, we obtained the mean IoU scores of M=31.3% SD=1.07, M=33.2% SD=1.72 and M=34.5% SD=1.73 for the 4 K only synthetic, 4K+1 K real ($w_r = 0.20$) and 4K+2 K real ($w_r = 0.33$) datasets respectively. Although the introduction of real samples, in particular the 2 K split, increases the overall IoU score, statistical significance is not found and future studies are needed concerning the instance segmentation of DLOs.

VI. Conclusion

In this work, a method to create datasets composed of real-world and synthetic data for DLOs perception has been presented. This method encompasses a weakly supervision to label points by an operator via a VR tracker and a CNN-based method to refine the input points and compute pixel-wise labels. Additionally, a synthetic DLOs data generator is proposed to render photo-realistic images of DLOs. The weakly supervised labeling method is gauged in terms of usability and quality of the labels, whereas the generated datasets are evaluated in terms of the impact of mixing real-world and synthetic data. The results show that the VR tracker labeling is as usable as the *chroma-key* and RITM but drastically reduces the number of necessary clicks for uneven backgrounds. Moreover, the introduction of real samples during the training of segmentation models helps in recovering not modeled specific texture and appearance information boosting the performance if the ratio between real-world data and synthetic one is 0.2. Despite these results, we identified several points that can be further studied in future work. In particular, the proposed CNN fine-tuning approach to correct the noisy user input could be improved in terms of annotation precision. In a more general context, the instance segmentation of DLOs via deep learning methods needs further investigation and analysis to understand the pitfalls of current methods and develop new techniques.

Ethics: Ethical review and approval are waived for this study due to anonymized data collection which, in Bavaria, where the study was conducted, does not need approval from an ethical committee (https://ethikkommission.blaek.de/studien/sonstige-studien/antragsunterlagen-ek-primarberatend-15-bo) (accessed on 22 November 2022).

References

[1] A. Keipour, M. Bandari, and S. Schaal, “Deformable one-dimensional object detection for routing and manipulation,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4329–4336, Apr. 2022.

[2] R. Zanella, A. Caporali, K. Tadaka, D. De Gregorio, and G. Palli, “Auto-generated wires dataset for semantic segmentation with domain-independence,” in *Proc. IEEE Int. Conf. Comput., Control Robot.*, 2021, pp. 292–298.

[3] A. Caporali, K. Galassi, R. Zanella, and G. Palli, “FASTDLO: Fast deformable linear objects instance segmentation,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9075–9082, Oct. 2022.

[4] J. Trommnau, J. Kühnle, J. Siegert, R. Inderka, and T. Bauernhansl, “Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends,” *Procedia CIRP*, vol. 81, pp. 387–392, 2019.

[5] J. Guo, J. Zhang, D. Wu, Y. Gai, and K. Chen, “An algorithm based on bidirectional searching and geometric constrained sampling for automatic manipulation planning in aircraft cable assembly,” *J. Manuf. Syst.*, vol. 57, pp. 158–168, 2020.

[6] D. D. Gregorio, G. Palli, and L. D. Stefano, “Let’s take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation,” in *Proc. Asian Conf. Comput. Vis.*, Springer, 2018, pp. 662–677.

[7] A. Caporali, R. Zanella, D. De Gregorio, and G. Palli, “Ariadne+: Deep learning-based augmented framework for the instance segmentation of wires,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8607–8617, Dec. 2022.

[8] H. G. Nguyen, R. Habiboglu, and J. Franke, "Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing," *Procedia CIRP*, vol. 107, pp. 1263–1268, 2022.

[9] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2372–2379, Apr. 2020.

[10] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 479–484.

[11] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 1998, pp. 130–137.

[12] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[13] V. Pătrăucean, P. Gurdjos, and R. G. Von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2012, pp. 572–585.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[15] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9157–9166.

[16] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: Better real-time instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1108–1121, Feb. 2022.

[17] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8573–8581.

[18] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*Springer, 2020, pp. 282–298.

[19] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 843–852.

[20] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 616–625.

[21] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," 2014, *arXiv:1412.7144*.

[22] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "Training object class detectors with click supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6374–6383.

[23] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *Proc. IEEE 8th Int. Conf. Comput. Vis.*, 2001, pp. 105–112.

[24] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-Supervised Convolutional Networks for Semantic Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3159–3167.

[25] H. Zhang, Y. Zeng, H. Lu, L. Zhang, J. Li, and J. Qi, "Learning to detect salient object with multi-source weak supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3577–3589, Jul. 2022.

[26] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11700–11709.

[27] B. Lutz, L. Janisch, D. Kisskalt, and D. Regulin, "Interactive Image Segmentation Using Superpixels and Deep Metric Learning for Tool Condition Monitoring," in *Proc. 16th CIRP Conf. Intell. Computation Manuf. Eng.*, 2022.

[28] G. Zheng, A. H. Awadallah, and S. Dumais, "Meta label correction for learning with weak supervision," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11053–11061.

[29] J. Liu et al., "Automatic label correction for the accurate edge detection of overlapping cervical cells," 2020, *arXiv:2010.01919*.

[30] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving iterative training with mask guidance for interactive segmentation," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 3141–3145.

[31] S. Fröhlig, M. v. F. auf Mayerhofen, M. Meiners, and J. Franke, "Three-dimensional pose estimation of deformable linear object tips based on a low-cost, two-dimensional sensor setup and ai-based evaluation," *Procedia CIRP*, vol. 107, pp. 499–504, 2022.

[32] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," in *Proc. Conf. Robot Learn.*, 2018, pp. 306–316 .

[33] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[34] J. Tremblay, T. To, and S. Birchfield, "Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2038–2041.

[35] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[36] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3752–3761.

[37] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4077–4085.

[38] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko, "Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation," 2018, *arXiv:1806.09755*.

[39] G. Ros, S. Stent, P. F. Alcantarilla, and T. Watanabe, "Training constrained deconvolutional networks for road scene semantic segmentation," 2016, *arXiv:1604.01545*.

[40] M. Denninger et al., "Blenderproc," 2019, *arXiv:1911.01911*.

[41] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.

[42] D. Gregorio, A. Tonioni, G. Palli, and L. Di Stefano, "Semiautomatic labeling for deep learning in robotics," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 2, pp. 611–620, Apr. 2020.

[43] A. K. T. Ng, L. K. Y. Chan, and H. Y. K. Lau, "A low-cost lighthouse-based virtual reality head tracking system," in *Proc. IEEE Int. Conf. 3D Immersion*, 2017, pp. 1–5.

[44] W. Zhang, X. Ma, L. Cui, and Q. Chen, "3 Points Calibration Method of Part Coordinates for Arc Welding Robot," in *Intelligent Robotics and Applications*. Berlin, Heidelberg: Springer2008, pp. 216–224.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[46] Y. Wang et al., "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4248–4257.

[47] T. Nguyen et al., "PennSyn2Real: Training Object Recognition Models Without Human Labeling," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5032–5039, Jul. 2021.

[48] J. Brooke, *SUS - A. Quick and Dirty Usability Scale*. Boca Raton, Florida, USA: CRC Press, 1996.

[49] S. G. Hart and L. E. Staveland, *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*. Amsterdam, The Netherlands: Elsevier, 1988.