Generating Synthetic Training Data for Assembly Processes

Johannes Dümmel, Valentin Kostik, Jan Oellerich

Generating Synthetic Training Data for Assembly Processes

Johannes Dümmel[1][0000−0001−7941−0958], Valentin Kostik[1] and Jan Oellerich[1]

[1]Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme, Gotthard-Franz-Str. 8, 76131 Karlsruhe, Germany
johannes.duemmel@kit.edu
https://www.ifl.kit.edu/

Abstract. Current assembly assistance systems use different methods for object detection. Deep learning methods occur, but are not elaborated in depth. For those methods, great amounts of individual training data are essential. The use of 3D data to generate synthetic training data is obvious, since this data is usually available for assembly processes. However, to guide through the entire assembly process not only the individual parts are to be detected, but also all intermediate steps. We present a system that uses the assembly sequence and the STEP file of the assembly as input to automatically generate synthetic training data as input for a convolutional neural network to identify the entire assembly process. By means of experimental validation it can be demonstrated, that domain randomization improves the results and that the developed system outperforms state of the art synthetic training data.

Keywords: Object Detection · Synthetic Training Data · Domain Randomization · Assembly Assistance Systems · Assembly Sequence.

1 Introduction

Assembly assistance systems (AAS) are key enablers for fully dynamic cross-company production networks [7]. They assist in manual assembly by guiding the user step by step through the process. AAS display the individual steps and control their correct execution. Therefor they need to detect all objects used in an assembly as accurately as all the steps of the assembly. Current research activities considering AAS mainly focus on the architecture of the whole system [31]. However, the different types of object detection still require a certain amount of manual effort [1].

In the field of machine vision convolutional neural networks (CNNs) efficiently accomplish the task of object detection [2, 27, 32]. Their main advantage compared to traditional object detection algorithms is the ability, to automatically extract relevant features of objects from a sufficient amount of training images. For this CNNs depend on a great number of annotated training data. For the development and testing of the algorithms, developers draw on large databases of annotated images [6, 10, 18, 19]. In industrial applications, however,

CNNs must detect individual parts while annotating large numbers of individual images manually is a time consuming process [30]. This results in the motivation to accelerate the process by automation.

In [8] a system is presented that can be used to reduce the time of creating and labeling training images without available 3D data of the object significantly. Here, the object is moved manually under a depth camera to define the objects position while at the same time RGB images of the respective view are created. This process is only applicable for small objects that can be hold in the hand and still requires manual effort. In industrial environments 3D data is available for almost all objects, most often in the form of STEP files [15]. There are several approaches using 3D data to generate synthetic training images to replace the real ones partially or completely [22, 23, 25].

In this paper we reduce the manual effort of generating training data to detect assemblies by combining synthetic training data from 3D files with the underlying assembly sequence. We automatically generate training images for the individual assembly parts just as for the assembly steps of an entire assembly. Finally, the performance of synthetic images with random backgrounds in comparison to the performance of images with partially application oriented (AO) backgrounds is experimentally evaluated.

2 Related Work

Current AAS use different technologies to identify the individual parts for the assembly steps. Pick-by-light is one of the simpler technologies, as it only marks the corresponding container of the individual parts [16]. This is done by virtually mapping the position of the container [9] or by 1D or 2D barcodes [13, 28].

Another possibility of identifying individual parts is the use of image processing algorithms. This involves the use of depth image data [9, 17] and RGB data [24]. In current AAS the data for recognizing the objects and the assembly steps are created manually.

Thamm et al. propose in [31] the use of the CNN YOLO [26] for object detection in their AAS. Here, the usage as well as the training methods remain still unexplained. Židek et al. already use a CNN for object recognition in an AAS [33]. There are also approaches to train a CNN with individual objects using synthetic data from CAD files to detect assembly parts [34].

The generation of synthetic training data for CNNs to recognize individual objects becomes popular in the field of computer vision [22, 23, 25]. Within this research area, there are two distinct approaches. Domain Adaptation (DA) aims to generate photorealistic training images [4,11,12] which requires a realistic synthetic environment. Furthermore objects will age in industrial environments and therefor wear out or pollute. Recognizing these objects using the DA approach is time consuming due to mapping all possible circumstances.

The second main approach generating synthetic training data is Domain Randomization (DR) [14, 21, 29]. The idea is to make reality appear as just another synthetic modification of the training images. Random lighting, backgrounds and

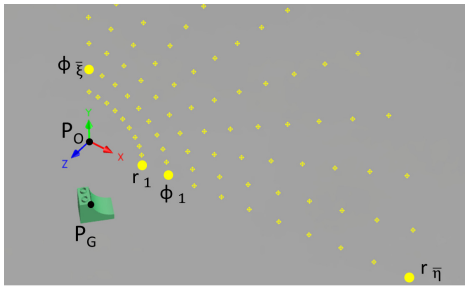

Fig. 1: Camera movement in quarter-circle orbits in the xy-plane around an object

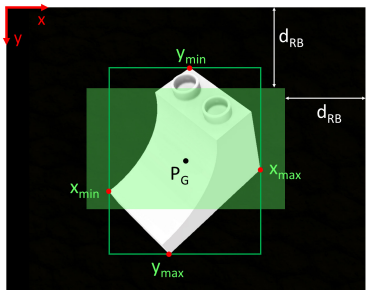

Fig. 2: Random position of the object in an image respecting the constraining distance d_{RB}

filters are applied to the initial images. DR is more robust than DA regarding detection under varying circumstances and can be implemented with less manual effort. For this reason our method is using DR.

3 Methodology

In the following section we describe the steps for generating synthetic training images and their corresponding bounding boxes. After that we demonstrate our approach for generating the training data for an entire assembly.

3.1 Synthetic Images

In the further course of the work, the CAD software Autodesk Inventor 2020 is used to read STEP files. They are widely spread in industrial environment and can be opened with any CAD software. In terms of image generating, we use one camera which moves in quarter-circle orbits in the xy-plane around an object. Hereby, the camera always aims at the origin P_{O}, see Fig. 1. To move the camera around an object and determine its position, we define the following sets

$$\mathbf{\Phi} = [\phi_1, \ldots, \phi_\xi, \ldots, \phi_{\bar{\xi}}]$$
$$\mathbf{R} = [r_1, \ldots, r_\eta, \ldots, r_{\bar{\eta}}]$$

and obtain the corresponding Cartesian coordinates

$$x_{\xi\eta} = r_\eta \cdot \cos\left(\phi_\xi \frac{\pi}{180°}\right), \quad \phi_\xi \in \mathbf{\Phi}, r_\eta \in \mathbf{R} \tag{1}$$
$$y_{\xi\eta} = r_\eta \cdot \sin\left(\phi_\xi \frac{\pi}{180°}\right), \quad \phi_\xi \in \mathbf{\Phi}, r_\eta \in \mathbf{R} \tag{2}$$

where $\mathbf{\Phi}$ describes the angular range between the x- and the y-axis with $\phi_1 \geq 1°$ and $\phi_{\bar{\xi}} \leq 90°$. This enables the adjustment of the training images depending on

(a) Random backgrounds from COCO (b) AO backgrounds

Fig. 3: Synthetic training images with random (a) and AO backgrounds (b)

the use case. The radius of the quarter-circle orbit between origin in the center and camera is defined by the set $\mathbf{R}$. The minimum value r_1 is determined so that the whole object fits in the image. We verify this by finding the minimal surrounding cuboid of the object fitting its biggest extension in the image. Consequently r_1 is variable depending on the size of the object. By defining the increment of the angular range $\mathbf{\Phi}$ by ξ and the increment of the radius range $\mathbf{R}$ by η we calculate the number of images. Consequently the camera position is defined by $P_{\mathrm{C}}(x_{\xi\eta}, y_{\xi\eta})$ for the current step $\xi\eta$.

While the camera moves on a defined path, we select one random orientation of the object for each image. To obtain realistic orientations of the objects we use the software Unity 2019 to drop the objects 45 times and extract the resulting Euler angles. We select one random Euler angle combination to define the orientation of the object and rotate the object around the y-axis by a random angle. The position of the object in the image is defined by the position of the center of gravity P_{G}. We start with $P_{\mathrm{G}} = P_{\mathrm{O}}$ and move P_{G} to a random position in each image limited by d_{RB} which is defined as the largest distance between P_{G} and the minimal surrounding cuboid as shown in Fig. 2. This ensures the whole object being completely visible in the image.

The set of background images consists of 44 AO images and 5 000 random images from the COCO validation dataset [19] (see Fig. 3). For each training image we choose a random background image from the set. In addition we change the lightning of the image by choosing randomly one of the 23 available lightning styles from Inventor. We create two images for each $P_{\mathrm{C}}(x_{\xi\eta}, y_{\xi\eta})$: one RGB training image and one contour image to define the bounding box around the object shown in Fig. 2. After generating the images we apply multiple filters from the ImageFilter module of pillow [5] to each image.

3.2 Data for entire assembly

Let an assembly $A = [\mathbf{P}, \mathrm{C}]$ consist of a total amount of parts $\mathbf{P} = [p_1, \ldots, p_k]$ and a certain amount of connecting parts $\mathbf{C} = [c_1, \ldots, c_l]$ such as bolts or shims. In this context, we define that the entire assembly A can be decomposed into single subassemblies where S_0 represents the smallest reasonable subassembly which consists of parts $\mathbf{p}_0 \subseteq \mathbf{P}$ and connection parts $\mathbf{c}_0 \subseteq \mathbf{C}$. The following greater subassembly S_1 in turn contains S_0 as well as a number of parts $\mathbf{p}_1 \subseteq \mathbf{P}$ and connection parts $\mathbf{c}_1 \subseteq \mathbf{C}$. Considering now that A can be represented by n

subassemblies and that each subassembly is related to the corresponding assembly step, we obtain the following recursive relation for a certain subassembly

$$S_i = (\mathbf{p}_i \subseteq \mathbf{P}, \mathbf{c}_i \subseteq \mathbf{C}, S_{i-1}), \quad i = 1, \ldots, n \quad n \in \mathbb{N}\backslash\{0\}. \tag{3}$$

In order to generate synthetic training data for A we first import the corresponding STEP file of the entire assembly. Then, each individual part, i.e. each element of $\mathbf{P}$, is extracted and saved as a separate file. The same applies for the subassemblies S_i. Here, the parts and connection parts which are not included are suppressed. Analogous to the parts, each subassembly is saved as a separate file as well. Finally, we generate synthetic training data based on each of the saved files.

4 Experiments and Validation

In this section, the performance of the developed method is evaluated by means of experimental validation. After describing the results we additionally determine the accuracy of our method by training the CNN CenterNet [32] with synthetic data and evaluating the resulting precision.

4.1 Design of Experiments

Two assemblies are used for the experiments: the Duplo™ and the charger assembly. Both assemblies with all individual parts and all subassemblies are shown in Fig. 4. The Duplo™ assembly possesses form-fit pins and recesses as connections and thus no connection parts. We always connect two components in this assembly, two parts in the first step and in every further step one part to the previous subassembly. With $k = 6$ this results in eleven objects for which synthetic training data must be created.

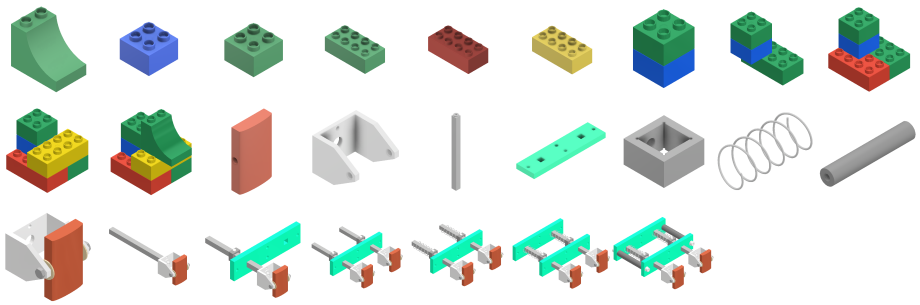

Fig. 4: Objects used in the experiments

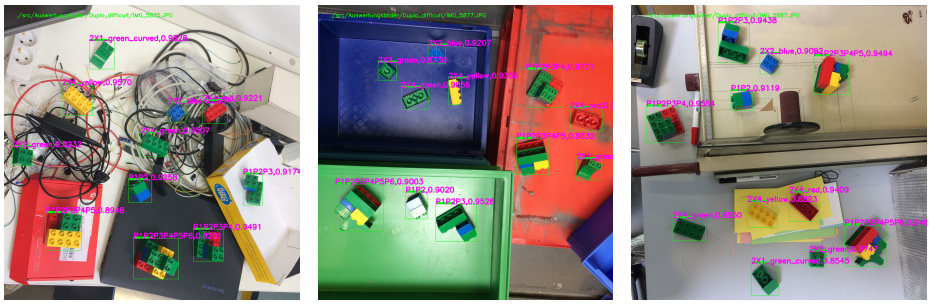

Fig. 5: Results of different test images with varying clutter in the background

The charger assembly has different connection parts like bolts, shims and nuts. Each step consists of two parts or of one part and one subassembly and a varying amount of connection parts. With $k = 7$ and $l = 7$ this results in 14 objects we need to create synthetic training data for. We do not create training data for the connection parts because they are generally delivered in larger quantities and supplied to the assembly workers in these quantities. Therefor the AAS does not need to detect connection parts. We use CenterNet to create one weight file for each assembly. The learning rate is set to $1.5625 \cdot 10^{-5}$. In epoch 90 and 120 we reduce the learning rate by the factor 10 and train 150 epochs.

Our evaluation dataset contains 900 real RGB images captured with the iPhone 5s camera at a resolution of 2448×2448. Here, backgrounds and cluttering in the dataset vary (see Fig. 5). Each object is displayed exactly 100 times on the images while the lightning conditions and the orientation vary as well. Here, all images are labelled manually using LabelImg [20].

4.2 Results

We train five times and vary the parameters for the synthetic training data, see Fig. 6. For the evaluation of our method we use the mean average precision (mAP) at 50 % intersection over union (IoU) between the ground truth and the detected labels evaluated with a tool from Cartucho et al. [3].

Fig. 6 shows the results including standard deviation as error bars. We trained both assemblies separately with 50 % AO background images and compared the performance of applying no filters to applying filters to the images. The mAP of the Duplo™ assembly improved by 0.03. The small difference can result from the small variance of the parts of the assembly in size and properties. In order to prove that filtering and thus DR actually provide better results, we compare the charger assembly trained without filters and 50 % AO background images and trained with filters and 50 % AO background images. Filtering improves the mAP by 0.27. Comparing all objects (see Fig. 4) with and without filter results in an increase of the mAP by 0.16. In another experiment, we verify how the results change by using only random background images. We compare the charger assembly trained with 50 % AO and with exclusively random background

images. Exclusively random background images result in the declining of the mAP by 0.10. Within the assemblies the standard deviation is declining with increasing randomization.

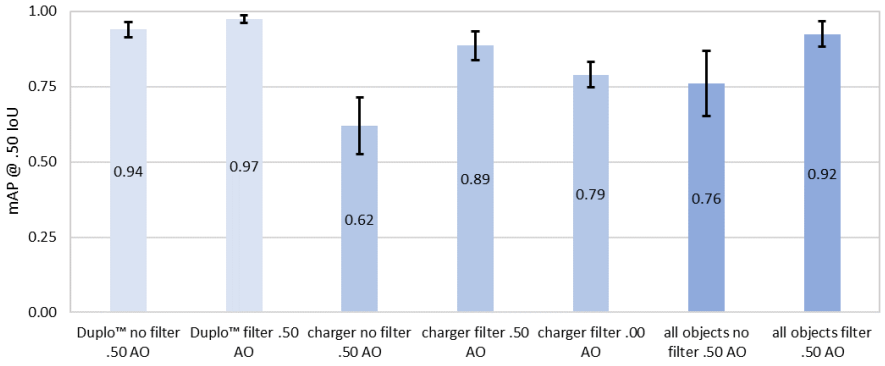

Fig. 6: mAP at .50 IoU for both assemblies and varied parameters

4.3 Discussion

Filtering images results for both assemblies in an increase of mAP. This increase is more visible with the charger assembly due to the higher variance and complexity of the involved assembly parts. The decreasing standard deviation within an assembly with increasing randomization leads to more robust systems. All objects are detected with a similar AP. Removing manual work completely for creating a new dataset by using exclusively random background images decreases the mAP. This could result from the COCO dataset containing everyday object categories in their natural environment. The missing industrial environment leads to a decrease in mAP due to CenterNet confusing industrial backgrounds as industrial objects. Adding the AO dataset increases the variance of backgrounds and leads to improved results for our test images. Whether the variance within the dataset is also sufficient in other industrial environments must be verified in further studies. The mAP for all 25 objects with best performing parameters is 0.92 and outperforms state of the art synthetic training data reaching an mAP @ .50 IoU of 0.89 [14] despite industrial objects being harder to detect due to less features.

5 Conclusion and Future Work

In this paper we describe a system to automatically generate training images from STEP files for an entire assembly. We experimentally demonstrate, that our system outperforms existing approaches of training with synthetic data. Experiments also show, that randomizing training images leads to less variation in AP within an assembly. Random background images from COCO however do not have sufficient variance. Our AO dataset with industrial backgrounds solves this problem. Further studies should verify whether the variance in our dataset is sufficient and expand it if necessary.

The goal of our future work is to gather the sequence directly from STEP files via simulation. Combining our system with this simulation would lead to the automatic generation of training images for object detection for an entire assembly with just the STEP file of the assembly as input data. This can be offered to companies as a service for AAS but also for the automatic assembly with robots or for remote services enabling customers to assemble or repair complex products without support from the supplier. Using object detection with synthetic training data as a service companies must be willing to share sensitive data. Further research should consider this problem by using encryption techniques or by allowing the software to be used locally on the customers hardware.

Acknowledgements This research has been funded by the German Federal Ministry of Education and Research (BMBF) under the program "Innovationen für die Produktion, Dienstleistung und Arbeit von morgen" and is supervised by Projektträger Karlsruhe (PTKA). The authors wish to acknowledge the funding agency and all the DPNB project partners for their contribution.

References

1. Bertram, P., Birtel, M., Quint, F., Ruskowski, M.: Intelligent manual working station through assistive systems. IFAC-PapersOnLine **51**(11), 170–175 (2018). https://doi.org/https://doi.org/10.1016/j.ifacol.2018.08.253, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection (2020), https://arxiv.org/pdf/2004.10934
3. Cartucho, J., Ventura, R., Veloso, M.: Robust object recognition through symbiotic deep learning in mobile robots. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2336–2341 (2018)
4. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
5. Clark, A.: Pillow (pil fork) documentation (2015), https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf, accessed: 15.03.2021
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)

7. DPNB: Broker für dynamische produktionsnetzwerke (2021), https://www.dpnb.de/, accessed: 19.03.2021
8. Dümmel, J., Hochstein, M., Glöckle, J., Furmans, K.: Effizientes labeln von artikeln für das einlernen künstlicher neuronaler netze. In: Logistics Journal : Proceedings. Wissenschaftliche Gesellschaft für Technische Logistik (2019). https://doi.org/10.2195/lj_Proc_duemmel_de_201912_01
9. Funk, M., Mayer, S., Schmidt, A.: Using in-situ projection to support cognitively impaired workers at the workplace. In: Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility. p. 185–192. ASSETS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2700648.2809853, https://doi.org/10.1145/2700648.2809853
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
11. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing training data for object detection in indoor scenes (2017), https://arxiv.org/pdf/1702.07836
12. Guo, Y., j. zhang, Cai, J., Jiang, B., Zheng, J.: Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(6), 1294–1307 (2019). https://doi.org/10.1109/TPAMI.2018.2837742
13. Hinrichsen, S., Riediger, D., Unrau, A.: Development of a projection-based assistance system for maintaining injection molding tools. In: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). pp. 1571–1575. IEEE (2017)
14. Hinterstoisser, S., Pauly, O., Heibel, H., Martina, M., Bokeloh, M.: An annotation saved is an annotation earned: Using fully synthetic training for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 2787–2796. IEEE (2019). https://doi.org/10.1109/ICCVW.2019.00340
15. ISO 10303-21:2016: Industrial automation systems and integration — product data representation and exchange — part 21: Implementation methods: Clear text encoding of the exchange structure (2016)
16. König, M., Stadlmaier, M., Rusch, T., Sochor, R., Merkel, L., Braunreuther, S., Schilp, J.: Ma 2 ra-manual assembly augmented reality assistant. In: 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). pp. 501–505. IEEE (2019)
17. Kosch, T., Kettner, R., Funk, M., Schmidt, A.: Motioneap – ein system zur effizienzsteigerung und assistenz bei produktionsprozessen in unternehmen auf basis von bewegungserkennung und projektion (2016)
18. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (2020). https://doi.org/10.1007/s11263-020-01316-z
19. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (2014), http://arxiv.org/pdf/1405.0312v3
20. Lin, T.: Labelimg. Git code (2015), https://github.com/tzutalin/labelImg, accessed: 02.08.2020
21. Mayershofer, C., Ge, T., Fottner, J.: Towards fully-synthetic training for industrial applications. In: 10th International Conference on Logistics, Informatics and Service Sciences (LISS) (2020)

22. Nowruzi, F.E., Kapoor, P., Kolhatkar, D., Hassanat, F.A., Laganiere, R., Rebut, J.: How much real data do we actually need: Analyzing object detection performance using synthetic and real data (2019)
23. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models (2014), https://arxiv.org/pdf/1412.7122
24. Quint, F., Loch, F., Orfgen, M., Zuehlke, D.: A system architecture for assistance in manual tasks. In: The 12th International Conference on Intelligent Environments. pp. 43–52 (2016)
25. Rajpura, P.S., Hegde, R.S., Bojinov, H.: Object detection using deep cnns trained on synthetic images. CoRR **abs/1706.06782** (2017), http://arxiv.org/abs/1706.06782
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks, https://arxiv.org/pdf/1506.01497
28. Rüther, S.: Assistive systems for quality assurance by context-aware user interfaces in health care and production. Ph.D. thesis, Universitätsbibliothek Bielefeld (2014)
29. Sarkar, K., Varanasi, K., Stricker, D.: Trained 3d models for cnn based object recognition. In: Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. pp. 130–137. SCITEPRESS - Science and Technology Publications (2017). https://doi.org/10.5220/0006272901300137
30. Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. vol. 51, pp. 1–8 (2008). https://doi.org/10.1109/CVPRW.2008.4562953
31. Thamm, S., Huebser, L., Adam, T., Hellebrandt, T., Heine, I., Barbalho, S., Velho, S.K., Becker, M., Bagnato, V.S., Schmitt, R.H.: Concept for an augmented intelligence-based quality assurance of assembly tasks in global value networks. Procedia CIRP **97**, 423–428 (2021). https://doi.org/https://doi.org/10.1016/j.procir.2020.05.262, 8th CIRP Conference of Assembly Technology and Systems
32. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points (2019), https://arxiv.org/pdf/1904.07850
33. Židek, K., Hosovsky, A., Piteľ, J., Bednár, S.: Recognition of assembly parts by convolutional neural networks. In: Hloch, S., Klichová, D., Krolczyk, G.M., Chattopadhyaya, S., Ruppenthalová, L. (eds.) Advances in Manufacturing Engineering and Materials, pp. 281–289. Lecture Notes in Mechanical Engineering, Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-319-99353-9_30
34. Židek, K., Lazorík, P., Piteľ, J., Pavlenko, I., Hošovský, A.: Automated training of convolutional networks by virtual 3d models for parts recognition in assembly process. In: Trojanowska, J., Ciszak, O., Machado, J.M., Pavlenko, I. (eds.) Advances in Manufacturing II, pp. 287–297. Lecture Notes in Mechanical Engineering, Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-18715-6_24