



HAL
open science

A Cobot Image Dataset for the Creation of Digital Twins in Industrial Robot Applications

Cristina Rosaria Monsone, Adam Csapo

► **To cite this version:**

Cristina Rosaria Monsone, Adam Csapo. A Cobot Image Dataset for the Creation of Digital Twins in Industrial Robot Applications. 12th IEEE International Conference on Cognitive Infocommunications: CogInfoCom 2021, Sep 2021, Online, Hungary. pp. 847-852. hal-04423367

HAL Id: hal-04423367

<https://hal.science/hal-04423367v1>

Submitted on 29 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC0 - Public Domain Dedication 4.0 International License

A Cobot Image Dataset for the Creation of Digital Twins in Industrial Robot Applications

Cristina R. Monsone*, mm B. Csapo*

*Szechenyi Istvan University, Gyor, Hungary

monsone.cristina@sze.hu, csapo.adam@sze.hu

Abstract—This paper focuses on how to generate digital twins (DTs) starting from image or video inputs through the use of modern AI methods. To address the problem of lack of public datasets, this paper reports on the creation of 2 new image datasets featuring 244 and 471 images, respectively, of collaborative robots (Cobots), annotated based on the 3 categories of Cobots, joints and links. Based on metrics obtained through the logging of training runs and the real-time visualization of network parameters, we demonstrate the impact of the number of images had on the learning capacity of the network. Pre-trained results have been saved and made available for future inference as well as transfer learning.

Index Terms—digital twins, cobots, industrial image datasets, convnets

I. INTRODUCTION

The process of digitization is driving a revolution, among others, in a variety of industrial sectors, by providing interactive methods and tools that reflect real-time processes to support professionals throughout the life cycle of a project. In a practical sense, this process is accompanied by a progressive shift from the use of traditional information vehicles, such as documents and spreadsheets to the use of interactive digital models as real-time virtualizations of a project [1].

The range of possibilities within digitization is broad and often dependent on the application scenario at hand. However, in general within the landscape of modern digitization, the concept of digital twins (DTs) has gained increasing ground in recent years across the board, including in manufacturing, construction, health care, aerospace, transportation and other application areas [2, 3, 4]. At the same time, even more widespread adoption of DTs is hindered by the expertise required for their deployment. This aim of this research is to bridge this gap by helping to enable the use of a hybrid (semi-automated) method of digital twin generation based on image or video-based datasets and deep learning techniques. To address the problem of a lack of publicly available datasets for industrial collaborative robotics applications, this paper reports the creation and evaluation of 2 public datasets with images of collaborative robots (Cobots).

The paper is structured as follows. In Section II, a summary is given of use case scenarios for and standardization efforts behind DTs. This is followed by a section on the general design process required for the creation of DTs (Section III). The two Cobot datasets we have created are described in Section IV. Finally, experimental results are provided in Section V.

II. USE CASES OF AND STANDARDIZATION EFFORTS BEHIND DIGITAL TWINS

According to the earliest definition of digital twins, formulated from the perspective of the aerospace industry by Glassgen and Stargel, a DT is “*an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin*” [5]. From a more generic point of view, a digital twin can be any virtual model that is capable of changing its appearance and / or behavior on the basis of external inputs representing events that have an impact on the process or application at hand. This translates into data and information assimilated by the DT and sent to it from different sources, including both events generated by manual (user) actions as well as data from specific sensors (e.g. pressure, temperature, humidity, seismic and so on). As a result of the generality of this framework, DTs are an increasingly hot topic in many application fields.

DTs can play a fundamental role as tools for the simulation of a large variety of scenarios. They can facilitate training activities on the use of expensive or otherwise difficult to access assets, or for situations that are potentially high risk and occur rarely, like natural disasters or pandemic situations [6, 7]. Further, the scarcity of opportunities to access an asset or situation can also present itself in the form of lack of available (historical) data, in which case DTs can also be of paramount importance.

DTs can also serve as tools for guidance of user behavior in complex situations, such as when navigating through an unfamiliar building (such as a hospital [7]), or when trying to reduce energy consumption or perform maintenance interventions in a real estate, industrial or smart city context [8, 9]. In some cases, DTs can even be collaborative, enabling users from different locations to interact with the same digital twin from different locations [10].

Depending on the use case, then, DTs can be widely different interactive models tailored to the needs of those who will work with them [11]. At the same time, it is no coincidence that there is strong interest surrounding DTs from the point of view of standardization, by organizations that take care of the development of digital standards for the market. In this regard, the activities carried out by ISO (International Organization for Standardization) are especially relevant, given

in particular the work carried out by the technical committee ISO / TC 184 Automation systems and integration, which for the manufacturing sector includes specific activities precisely on the topic of digital twins (ISO / TC 184 / AHG 2)[12] .

III. DEVELOPMENT PROCESS BEHIND DIGITAL TWINS

As highlighted in [10], DTs that have practical value are often data driven, therefore the creation of DTs involves challenges such as finding the most useful information within large quantities of data; integrating the filtered information into a presentable, actionable form; being able to respond to incoming data in real time; and being able to predict future events based on the past. At the same time, modern DTs also have a strongly visual element to them: in order to engage users in the most effective ways possible, 2D web technologies and increasingly 3D applications (both VR and AR) are being developed [13, 14, 15, 16].

Therefore, the first step of creating DTs is generally to use a CAD or other 3D modeling tool – or alternatively a 3D scanning approach – to virtualize the physical entity or entities being represented [17, 18]. This is followed by the development of a suitable data processing pipeline, the development of simulation capabilities (i.e. affordances of the virtualized copy to be driven by user interactions and data), development of control affordances for the physical entity or entities being modeled (so that lessons learned from the DT can be used to control their physical counterpart), and the establishment of real-time secure connections between the DT and the physical entity [10].

Clearly, the development of DTs requires know-how from many domains, without even considering the target domain of the DT application itself. Most small- and medium-sized enterprises do not have the know-how or resources to engage in graphical modeling or DT-oriented data modeling activities. Therefore, it will be essential to develop high-level tools that facilitate the creation and customization of DTs in an application agnostic way.

IV. CREATION OF A DATASET FOR IMAGE-BASED COBOT STRUCTURE RECOGNITION

The development of AI software based on artificial learning requires the creation of data sets dedicated to training and system validation. Collecting and preparing data in sufficient quantities for these purposes is a difficult task. It can take sufficient domain-specific knowledge and a long time for the data scientist to be able to select the most relevant information. From a CogInfoCom perspective, human and artificial capabilities are both essential in achieving the desired result [19, 20].

The creation of training datasets has great value in the field of data science, because it is closely correlated with the ability to solve problems through AI software, by helping to guide the deep learning neural network in extracting the useful features from the training set, allowing it to generalize on unknown data. Further, a well-designed dataset can contribute to the

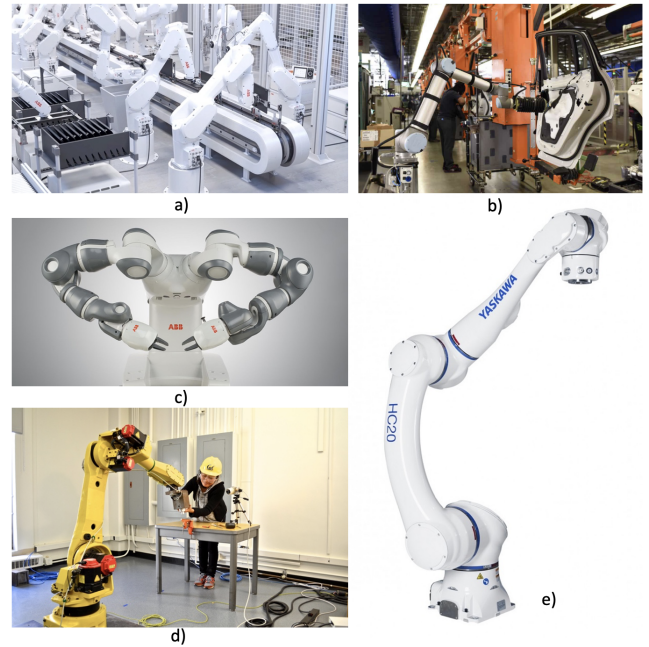


Fig. 1. Sample images used in our datasets. The images represent a variety of configurations, with different Cobot joint configurations and with or without human presence.

development of new applications that are tangentially related to the original one, through so-called transfer learning.

A. Cobot Database

The dataset introduced in this paper is based on a collection of images of collaborative robots (Cobots). To understand the suitability of our approach towards the creation of the dataset, we created two versions of the dataset: one with only 106 raw images, and a second one that consists of 211 raw images before data augmentation. Figure 1 shows some examples of the images included in the datasets.

A second, perhaps even more important challenge involved the creation of suitable labels for the samples. With a view towards applying the various YOLO architectures for both object recognition and object localization [21, 22], we used the LabelImg app [23] to annotate the location of Cobots, joints and links in each sample image. Annotations were saved as XML files in PASCAL VOC format, the format used by ImageNet. The three object types are shown on Figure 2.

Our general assumption in our selection of labels was that the resulting classifier would perform better with just 3 categories, while the categories of cobot, joint and link would provide sufficient information for the Cobots to be reproducible in as a DT. Further details on the two datasets are provided in Section IV-C and shown in Figures 3 and 4.

B. YOLOv5 in Pytorch

The YOLOv5 algorithm is natively available in PyTorch, which makes the loading of data and experimentation with different hyperparameters straightforward [22]. The architecture

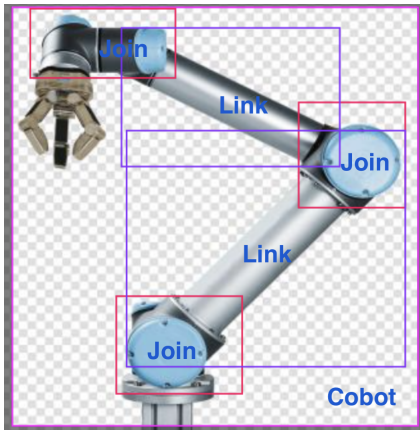


Fig. 2. Labels created for classification represent the location and identity of cobots, joints and links.

comes pretrained on the COCO dataset. Although the model architecture of YOLOv5 remains close to YOLOv4, it has a smaller size (27 MB instead of 244 MB) and presents further advantages in speed, accuracy and performance alike.

The YOLO family belongs to single shot object detectors; unlike the RCNN family it does not have a separate network for Regional Proposals (RPN) and depends on anchors at various scales [24]. The YOLO architecture can be divided into three parts: the backbone, neck and head as follows:

- 1) The backbone is a convolutional neural network that aggregates and forms image features at different granularities;
- 2) The neck is a series of layers to mix and combine image features to pass them forward to prediction;
- 3) The head consumes features from the neck and takes box and class prediction steps

C. Data Augmentation

In order to obtain larger datasets and improve the models that could be trained using the datasets, we performed data augmentation as well, thus expanding our datasets to include 244 (based on 106 raw images) and 471 (based on 211 raw images) samples.

Note that as described in Section IV-D, the YOLOv5 architecture also automatically augments the training data using scaling, color space adjustments, as well as a mosaic-like tiling of multiple training images. Therefore, we applied a different set of augmentation methods as described below.

To execute the pre-processing steps, we used Roboflow as follows:

- We first resized all images by downscaling them to a size of 416-by-416 pixels. We scaled annotations accordingly.
- We then stripped all images of their EXIF data in order to remove orientation information [25]
- Finally, we augmented the datasets to hopefully increase the generalization capability of the model, by increasing the diversity of training samples. The following approaches were used:

	Dataset 471	Dataset 244
Number of raw images	211	106
Annotations	1042	228
Number of images after data augmentation	471	244
Images for train	408	212
Images for validation	42	21
Images for test	21	11

Fig. 3. Datasets parameters

	Class balance-labels		
	Join	Link	Cobot
Dataset 471	428	328	286
Dataset 244	155	55	18

Fig. 4. Class balance

- manipulation of exposure of images
- manipulation of Gaussian blur with a kernel of up to 2 pixels

Following augmentation, we obtained the following number of representatives for each category:

- Dataset471 class balance: 408 for Joint, 328 for Link. and 286 for Cobot
- Dataset244 class balance: 155 for Joint, 55 for Link. and 18 for Cobot

Regarding Dataset471, Figures 3 and 4 show various parameters of and summary information on the class balance. Figures 5 and 6 provide further information on the distribution and correlation of various label parameters with the dataset.

D. Automatic Data Augmentation YOLOv5

With each training batch, YOLOv5s passes training data through a data loader, which augments data online. The data loader makes three kinds of augmentations: scaling, color space adjustments, and mosaic augmentation. The most novel of these being mosaic data augmentation, which combines four images into four tiles of random ratio.

Mosaic augmentation is especially useful helping the model learn to address the well-known “small object problem”.

Some examples of ground truth data are shown in Figure 7.

V. EXPERIMENTAL RESULTS

Following the training procedure, the performance has been evaluated by looking at the validation metrics and considering a new video with a Cobot working in real time. In particular, weights & biases were used for real-time visualization and cloud logging was performed of training runs.

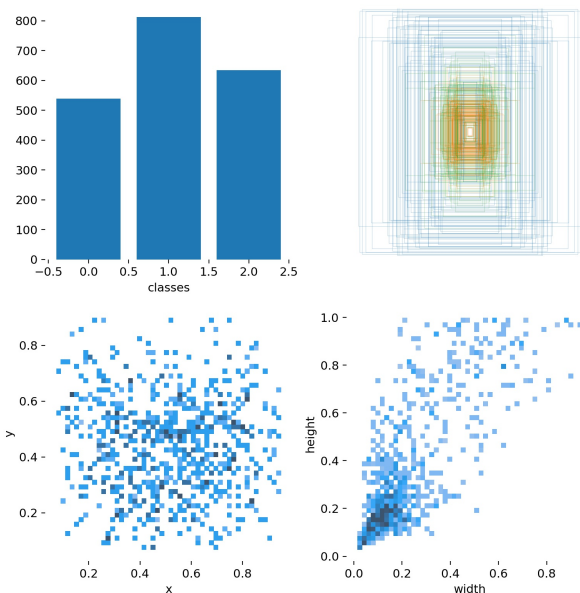


Fig. 5. Labels distribution for Dataset471

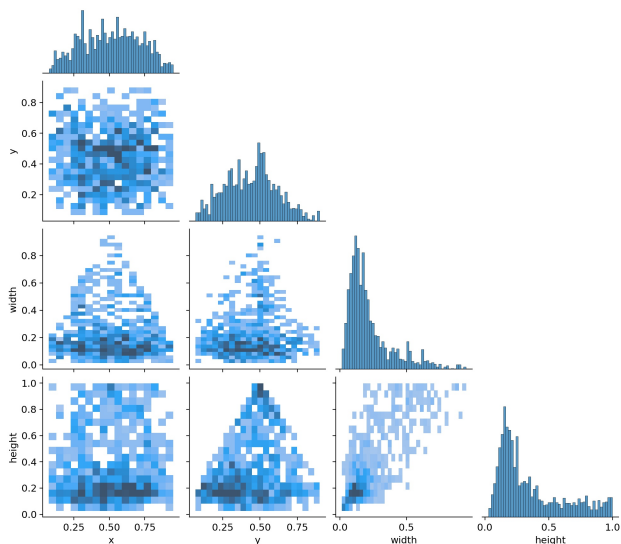


Fig. 6. Labels correlogram for Dataset471

As mentioned at the beginning, we considered 2 datasets. For each of the datasets, the metrics are reported in Figures 8 and 9. Each figure shows the relevant bounding box location, confidence and classification related losses.

Figures 10 and 11 both pertain to Dataset471 and show the training and validation errors for location (box), classification (cls) and confidence (objectness). Similarly, Figures 12 and V show the precision, recall, precision-recall and F1 curves, as well as the confusion matrix for the whole Dataset471.

Generally, precision-recall curves are used in binary classi-

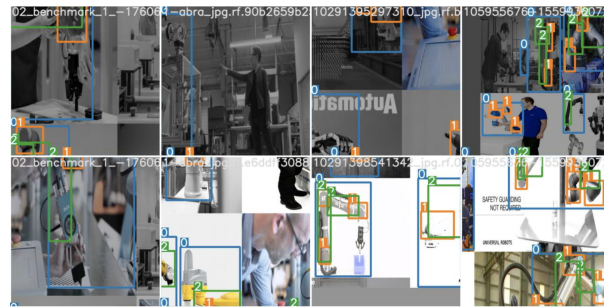


Fig. 7. Ground truth augmented data for Dataset471

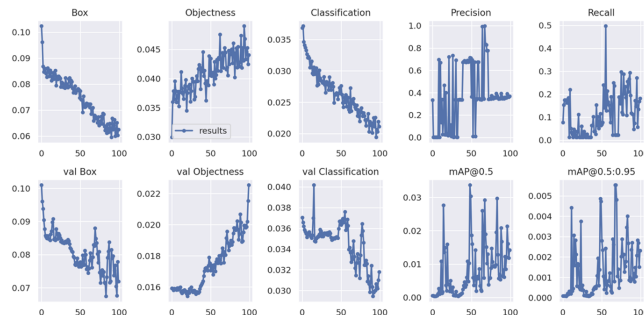


Fig. 8. Metrics for Dataset244

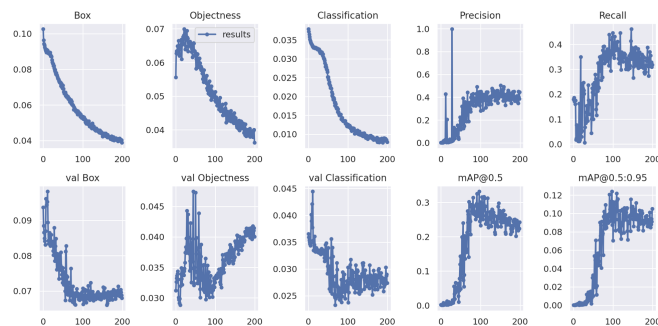


Fig. 9. Metrics for Dataset471.

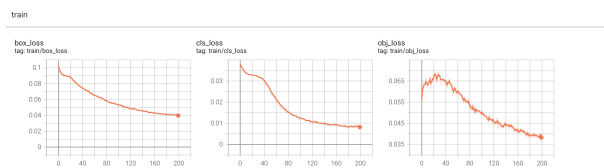


Fig. 10. Metrics train

fication, however, they can be extended for multi-class cases as well, in which case a separate curve can be drawn for each label. The results, especially in Figure 12 show that the dataset and associated model are a good start towards identifying cobots and cobot parts in images, however, there is a tradeoff between precision and recall depending on the decision threshold chosen.

Finally, some examples of performance on test images are

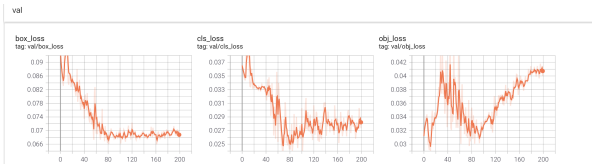


Fig. 11. Metric val

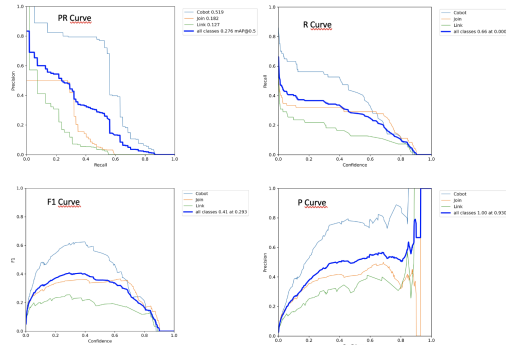


Fig. 12. P, R, PR and F1 Curves.

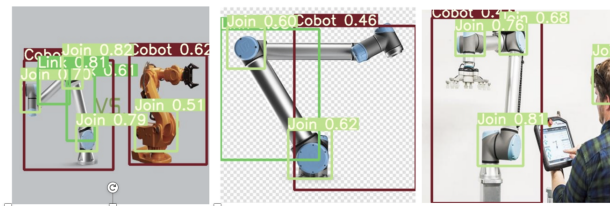
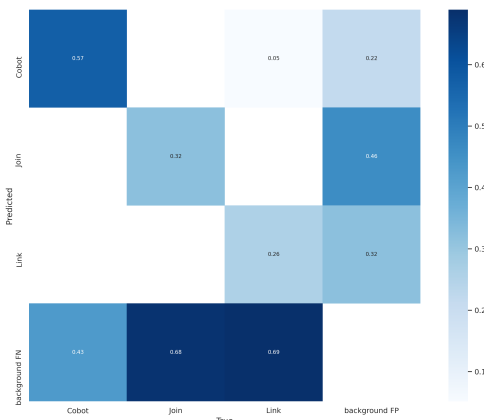


Fig. 13. Examples of performance on held-out test images

shown in Figure 13.

It remains to be seen whether the tradeoff is due to the limited size of the dataset or the inadequacy of the model. At the same time, it seems that the created Dataset471 and trained model can be useful when applied to videos in that a low recall but high precision can help the detection of cobots at least in some frames. This was the case in our experiment with a separate video, as shown on Figure 14.

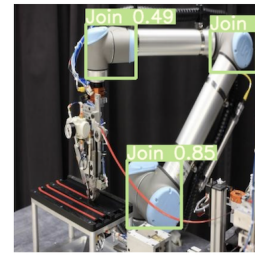


Fig. 14. Results of object detection using the weight of our dataset in a video showing Cobots working in real time.

At the end, the YOLOv5 weights based on our dataset have been saved for future inference¹.

VI. CONCLUSION

This paper focused on how to generate digital twins (DTs) starting from image or video inputs through the use of modern AI methods. To address the problem of lack of public datasets, this paper reported on the creation of 2 new image datasets featuring 244 and 471 images, respectively, of collaborative robots (Cobots), annotated based on the 3 categories of Cobots, joints and links. Based on metrics obtained through the logging of training runs and the real-time visualization of network parameters, we demonstrated the impact of the number of images had on the learning capacity of the network. Pre-trained results have been saved and made available for future inference as well as transfer learning.

REFERENCES

- [1] Fahmi Bellalouna. Case study for an operation-based topology optimization using the digital twin approach. *Procedia CIRP*, 98:342–347, 2021.
- [2] Maryam Farsi, Alireza Daneshkhah, Amin Hosseinian-Far, Hamid Jahankhani, et al. *Digital twin technologies and smart cities*. Springer, 2020.
- [3] Cristina Rosaria Monsone and János Jósvai. Challenges of digital twin system. *Acta Technica Jaurinensis*, 12(3):252–267, 2019.
- [4] Cristina Monsone. Holistic approach to smart factory. In *IFIP International Workshop on Artificial Intelligence for Knowledge Management*, pages 160–176. Springer, 2021.
- [5] Edward Glaessgen and David Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, page 1818, 2012.
- [6] Chao Fan, Cheng Zhang, Alex Yahja, and Ali Mostafavi. Disaster city digital twin: A vision for integrating artificial and human intelligence for disaster manage-

¹<https://github.com/CMon-code/CMon-code-Dataset-and-training-algorithm-for-images-classification-in-real-time-.git>

- ment. *International Journal of Information Management*, 56:102049, 2021.
- [7] Qibing Lv, Rong Zhang, Xuemin Sun, Yuqian Lu, and Jinsong Bao. A digital twin-driven human-robot collaborative assembly approach in the wake of covid-19. *Journal of Manufacturing Systems*, 2021.
- [8] Sin Yong Teng, Michal Touš, Wei Dong Leong, Bing Shen How, Hon Loong Lam, and Vítězslav Máša. Recent advances on industrial data-driven energy savings: Digital twins and infrastructures. *Renewable and Sustainable Energy Reviews*, 135:110208, 2021.
- [9] Timo Ruohomäki, Enni Airaksinen, Petteri Huuska, Outi Kesäniemi, Mikko Martikka, and Jarmo Suomisto. Smart city platform enabling digital twin. In *2018 International Conference on Intelligent Systems (IS)*, pages 155–161. IEEE, 2018.
- [10] Fei Tao, Fangyuan Sui, Ang Liu, Qinglin Qi, Meng Zhang, Boyang Song, Zirong Guo, Stephen C-Y Lu, and Andrew YC Nee. Digital twin-driven product design framework. *International Journal of Production Research*, 57(12):3935–3953, 2019.
- [11] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and AYC Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 2019.
- [12] Kilian Bächle and Stefan Gregorzik. Digital twins in industrial applications—requirements to a comprehensive data model. *Journal of Innovation, November*, pages 1–14, 2019.
- [13] Samad ME Sepasgozar. Digital twin and web-based virtual gaming technologies for online education: A case of construction management and engineering. *Applied Sciences*, 10(13):4678, 2020.
- [14] Greyce Schroeder, Charles Steinmetz, Carlos Eduardo Pereira, Ivan Muller, Natanael Garcia, Danubia Espindola, and Ricardo Rodrigues. Visualising the digital twin using web services and augmented reality. In *2016 IEEE 14th international conference on industrial informatics (INDIN)*, pages 522–527. IEEE, 2016.
- [15] Michael E Porter and James E Heppelmann. Why every organization needs an augmented reality strategy. *HBR'S 10 MUST*, 85, 2017.
- [16] Roberto Revetria, Flavio Tonelli, Lorenzo Damiani, Melissa Demartini, Federico Bisio, and Nicola Peruzzo. A real-time mechanical structures monitoring system based on digital twin, iot and augmented reality. In *2019 Spring Simulation Conference (SpringSim)*, pages 1–10. IEEE, 2019.
- [17] Mika Lohtander, Niko Ahonen, Minna Lanz, Juho Ratava, and Jarno Kaakkunen. Micro manufacturing unit and the corresponding 3d-model for the digital twin. *Procedia manufacturing*, 25:55–61, 2018.
- [18] Markus Sommer, Josip Stjepandić, Sebastian Stobrawa, and Moritz von Soden. Automatic generation of digital twin based on scanning and object recognition. In *Transdisciplinary Engineering for Complex Socio-technical Systems*, pages 645–654. IOS Press, 2019.
- [19] Péter Baranyi, Adam Csapo, and Gyula Sallai. *Cognitive Infocommunications (CogInfoCom)*. Springer, 2015.
- [20] Cristina Monsone and Adám B Csapó. Charting the state-of-the-art in the application of convolutional neural networks to quality control in industry 4.0 and smart manufacturing. In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 463–468. IEEE, 2019.
- [21] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [22] Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu, Adam Hogan, et al. ultralytics/yolov5: v3. 1-bug fixes and performance improvements, 2020.
- [23] TzuTa Lin. Labelimg, 2015.
- [24] Glenn Jocher, Alex Stoken, Jirka Borovec, et al. ultralytics/yolov5: v4. 0-nn. silu () activations, weights & biases logging, pytorch hub integration (version v4. 0). zenodo, 2021.
- [25] Inc. Roboflow. Roboflow docs – image transformations. <https://docs.roboflow.com/image-transformations/image-preprocessing>, 2021.