applied sciences

MDPI

Article

CNN Training Using 3D Virtual Models for Assisted Assembly with Mixed Reality and Collaborative Robots

Kamil Židek, Ján Piteľ *, Michal Balog, Alexander Hošovský, Vratislav Hladký, Peter Lazorík, Angelina Iakovets and Jakub Demčák

Department of Industrial Engineering and Informatics, Faculty of Manufacturing Technologies with a Seat in Presov, Technical University of Kosice, Bayerova 1, 08001 Presov, Slovakia; kamil.zidek@tuke.sk (K.Ž.); michal.balog@tuke.sk (M.B.); alexander.hosovsky@tuke.sk (A.H.); vratislav.hladky@tuke.sk (V.H.); peter.lazorik@tuke.sk (P.L.); angelina.iakovets@tuke.sk (A.I.); jakub.demcak@tuke.sk (J.D.)
* Correspondence: jan.pitel@tuke.sk; Tel.: +421-90-524-1605 or +421-55-602-6455

Abstract: The assisted assembly of customized products supported by collaborative robots combined with mixed reality devices is the current trend in the Industry 4.0 concept. This article introduces an experimental work cell with the implementation of the assisted assembly process for customized cam switches as a case study. The research is aimed to design a methodology for this complex task with full digitalization and transformation data to digital twin models from all vision systems. Recognition of position and orientation of assembled parts during manual assembly are marked and checked by convolutional neural network (CNN) model. Training of CNN was based on a new approach using virtual training samples with single shot detection and instance segmentation. The trained CNN model was transferred to an embedded artificial processing unit with a high-resolution camera sensor. The embedded device redistributes data with parts detected position and orientation into mixed reality devices and collaborative robot. This approach to assisted assembly using mixed reality, collaborative robot, vision systems, and CNN models can significantly decrease assembly and training time in real production.

Keywords: assisted assembly; mixed reality; collaborative robot; digital twin; convolutional neural networks

Citation: Židek, K.; Piteľ, J.; Balog, M.; Hošovský, A.; Hladký, V.; Lazorík, P.; Iakovets, A.; Demčák, J. CNN Training Using 3D Virtual Models for Assisted Assembly with Mixed Reality and Collaborative Robots. *Appl. Sci.* **2021**, *11*, 4269. https://doi.org/10.3390/app11094269

Academic Editor: Pavol Božek

Received: 13 April 2021
Accepted: 6 May 2021
Published: 8 May 2021

1. Introduction and Related Works

Collaborative robots and their implementation in the assisted assembly process is an important part of the Industry 4.0 concept. They can work in the same workspace as human workers and perform basic manipulations or simple monotonous assembly tasks. This area is open to new research, methodology development and definition of basic requirements, because real applications in production processes are currently still limited.

The main advantage of using collaborative robots in the assembly process is a minimal transport delay of assembly parts between manual and automated operation. Other benefits are, for example, an integrated vision system for additional inspection of manual operation, the possibility to provide the interface for digital data collection from sensors and communication with external cloud platforms.

Appropriate human-robot cooperation can significantly improve assembly time, but both must have exactly defined methods of communication between them. For example, the collaborative robot can check the success of a worker assembly operation by the integrated camera and the worker can get information about this status by mixed reality devices. The mixed reality device also can shortcut the time of staff training. Configuration principles of a collaborative robot in an assembly task were introduced in [1], a framework to implement collaborative robots in the manual assembly in [2] and a human-robot collaboration framework for improving ergonomics in [3]. The important condition for the assisted assembly process is the synchronization of augmented (AR), virtual (VR), or mixed

(MR) reality devices with the digital twin for full digitalization of the used technology. A nice review of virtual, mixed, and augmented reality for immersive systems research is presented in [4]. Some other research results of the mixed assembly process between human and collaborative robots are described in [5–7]. An AR-based worker support system for human-robot collaboration using AR libraries was proposed in [8] and an anchoring support system using the AR toolkit was developed in [9]. A novel approach for end-user-oriented no-code process modeling in IoT domains using mixed reality technology is presented in [10] and a holistic analysis towards understanding consumer perceptions of virtual reality devices in the post-adoption phase in [11]. Technologies of virtual, augmented and mixed reality have an important role also in educational and training purposes as it was, for example, stated in [12,13].

Our research in the field of a digital twin implementation into assembly processes started with the digitalization of the experimental manufacturing assembly system described in [14]. A digital twin can visualize the real status of a manufacturing system as a 3D simulation model with real-time actualization, so there is a need to have an appropriate simulation model, for example a generic simulation model of the flexible manufacturing system developed in [15]. An automatic generation of a simulation-based digital twin of an industrial process plant is described in [16]. The basic overview about the usability of a digital twin can be found in [17], the possibility of improving the efficiency of the production by the implementation of digital twins is presented in [18]. Some learning experiences after establishing digital twins are described in [19] and using a digital twin to enhance the integration of ergonomics in the workplace design in [20]. Recommendations for future research and practice in the use of digital twins in the field of collaborative robotics are given in [21]. Also, this brief overview shows that a digital twin is an important element in the implementation of an assisted assembly into the production process.

The next condition of successful implementation of an assisted assembly into the production process is synchronization with the recognition system mainly based on vision devices with some SMART features like object detection, assembly parts identification and localization with their actual orientation in a workspace. Based on the obtained knowledge from the research on diagnostics of errors at the component surface by vision recognition systems using machine learning algorithms [22] we have started to use a convolutional neural network (CNN) for the recognition of standardized industrial parts (hexagon screw/nut and circular hole assembly elements) trained by real image input [23]. CNN with deep learning can work reliably for parts recognition, but the problem is manual preparation of input data for the learning process, because a very large quantity of input samples need to be prepared, usually several hundred for one part. The solution is to replace real image samples with 3D virtual models [24], but an important task is an automated sample generation from 3D models, which can be simplified using a web interface [25]. Besides automated input data preparation for CNN training, automated image analysis is also important. Some principles of this process were described in [26]. An interesting case study on the recognition of mark images using deep CNN was published in [27]. A methodology for synthesizing novel 3D image classifiers by generative enhancement of existing non-generative classifiers was proposed and verified in [28]. A viewpoint estimation in images using CNNs trained with rendered 3D model views was published in [29] and an accurate and fast CNN-based 6DoF object pose estimation using synthetic training in [30].

The sections of this article are structured in the following manner: following the introduction and related works in this Section a methodology of CNN training using 3D virtual models is introduced in Section 2. Section 3 describes the experimental assisted assembly work cell and the assembled product, in Section 4 the principles of the 3D virtual model preparation and 2D sample generation for CNN training are presented. Section 5 contains results and discussion, including implementation of parts recognition into the collaborative work cell. Finally, Section 6 presents a summary of the article along with some ideas for future work.

The main novelty and the innovation contribution of the article is a complex methodology for CNN training by virtual 3D models and design of a communication framework for assisted assembly devices like collaborative robot and mixed reality device.

2. Methodology of Deep Learning Implementation into the Assisted Assembly Process

A methodology of CNN training using 3D virtual models for deep learning implementation into the assisted assembly process is based on an automated generation of input sample data for learning without any monotonous manual work. All tasks, such as an object detection position, background and material change can be automated by the scripting language. This methodology can be divided into eight steps:

(1) Creation of 3D virtual models from the experimental assembly by any 3D design software or point cloud creation by laser scanning technology with conversion to some standard 3D format (OBJ, FBX, STL, IGES, etc.);
(2) Import 3D models into the software with cinematic rendering and some simulation of dynamics;
(3) Algorithms design of an automatic data queue of parts positioning, rotating, and camera setup by parts size;
(4) Rendering two sets of images: the first for CNN teaching and the second for an automated annotation algorithm;
(5) Creating of XML file for single shot detection and JSON format for instance segmentation;
(6) Automated ratio sorting to training and testing samples and moving to separate folder;
(7) Training of convolutional neural network for parts classification and localization (using single shot detection and instance segmentation);
(8) Transformation of CNN models into some type of embedded devices for inference of the trained model and results distribution of the detected position data to assisted assembly systems: a collaborative robot internal Cartesian system and mixed reality device anchoring system;

The detected objects are placed on the floor and they can be rotated only around one axis with a chosen increment from 20° to 360° by step from 1 to 18. The translation and rotation of virtual parts in scene are computed by standard translation and rotation matrixes for placement in 3D environment [24].

The detected object can be too small, for example as nuts or washers, so it is necessary to get magnification to change the field of view for the camera, according to Equation (1):

$$\text{if } \frac{FOV_{H,V}}{D_{X,Y}} > 3; \text{ then } H = Mf_L, \tag{1}$$

where $FOV_{H,V}$ is field of view (horizontal or vertical), $D_{X,Y}$ is object dimension in X or Y axis, H is distance to object [mm], M is magnification setup to 0.5x or 0.25x and f_L is focal length [mm].

The number of generated 2D images from all imported parts is counted by simple Equation (2):

$$N = pfn_\alpha, \tag{2}$$

where p is the number of imported parts, f is the number of used floors and n_α is the number of rotations for every part (in the range from 1 to 18).

Figure 1 presents a diagram of the proposed methodology of automatic data preparation for CNN automated training using experimental values, evaluation, and execution in the embedded device.

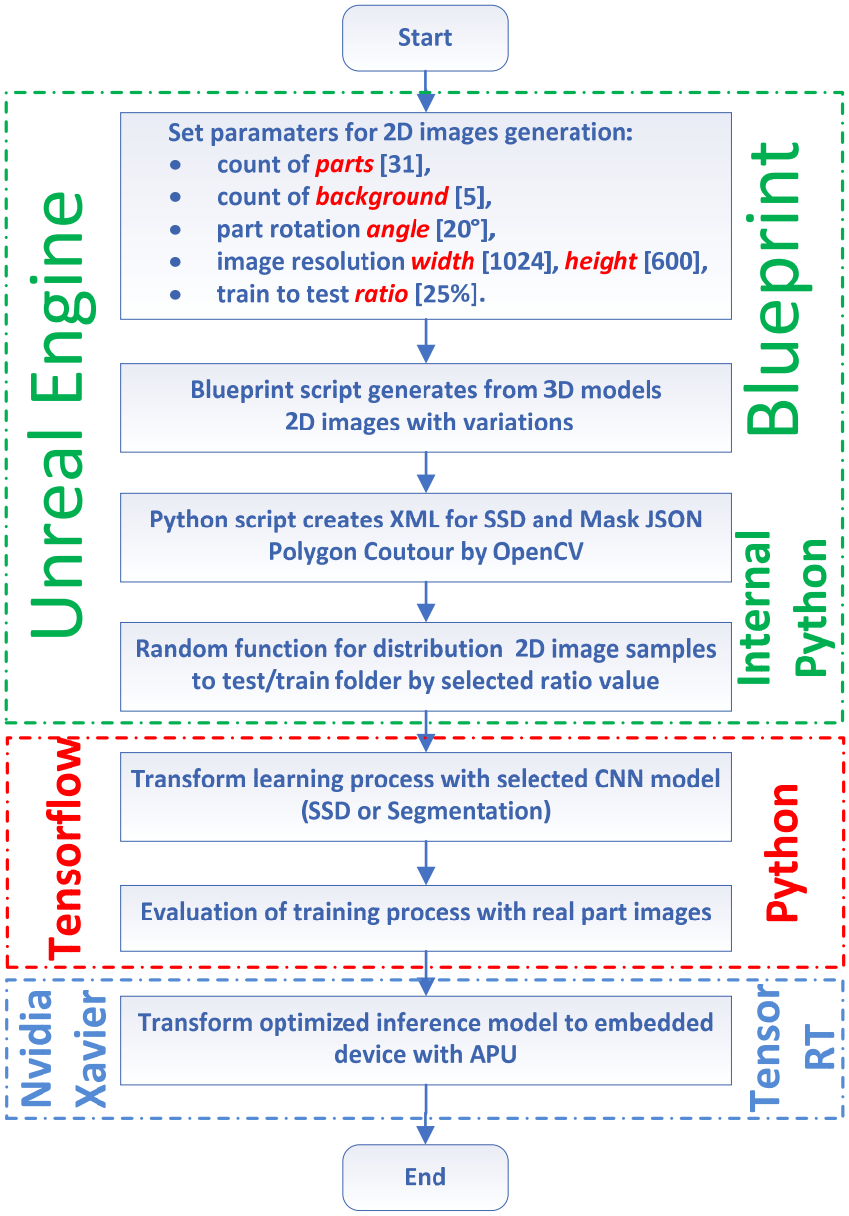


Figure 1. The simplified algorithm for samples generation from 3D virtual models of assembly parts.

3. Experimental Platform

The research has been provided in the SmartTechLab for Industry 4.0 at the Faculty of Manufacturing Technologies of Technical University of Kosice. There is installed an experimental SMART manufacturing system established primarily for research purposes, but also for collaboration with companies and teaching purposes. An important part of this system is a work cell for assisted assembly with incorporated technologies for parts recognition, mixed reality and collaborative robotics (Figures 2 and 3).

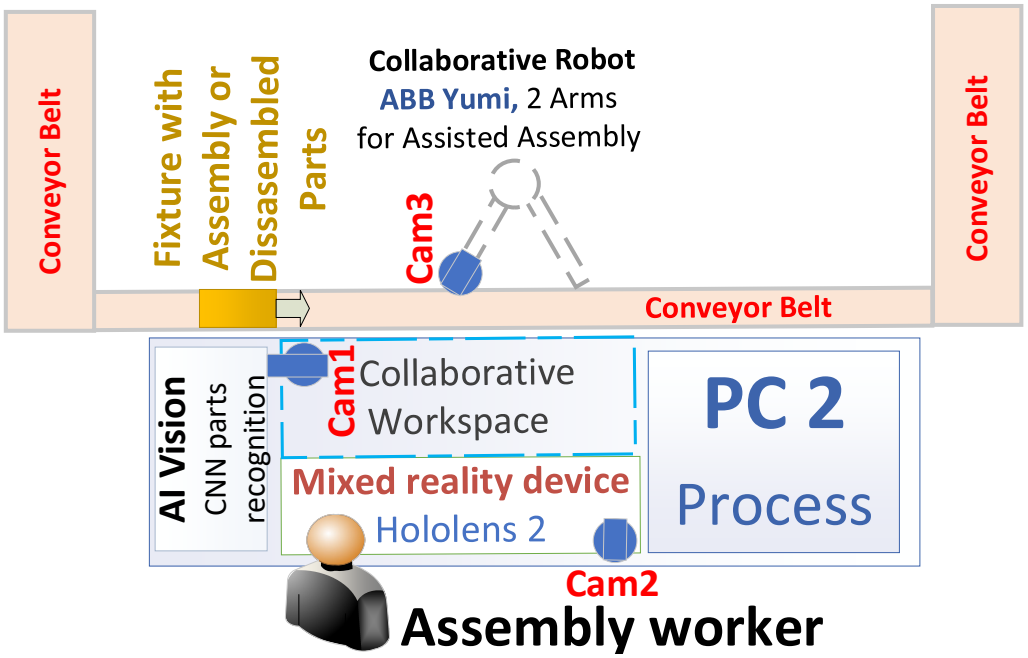
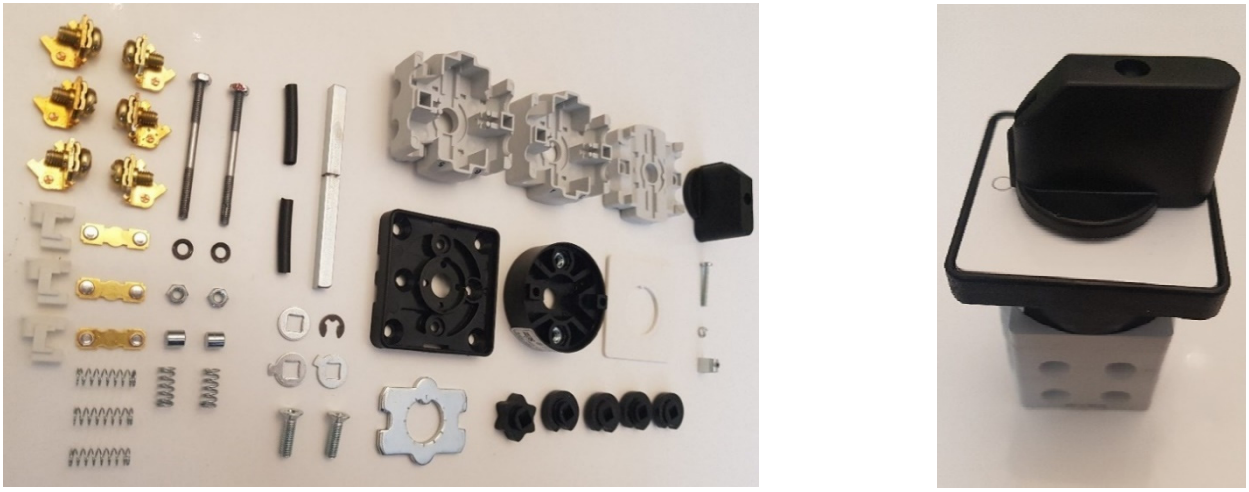


Figure 2. A scheme of the experimental assisted assembly work cell with CNN processing unit, mixed reality device and collaborative robot.

Figure 3. The experimental SMART manufacturing system; red frame: the workplace with an assisted assembly work cell with collaborative robot ABB Yumi and Microsoft Hololens 2 mixed reality device.

The point of interest for experimental assembly is a cam switch consisting of 31 parts made from different materials: plastic, rubber, stainless steel, and brass. The disassembled parts are shown in Figure 4a, and the assembled product is shown in Figure 4b.



(**a**) (**b**)

Figure 4. The experimental product used in research: (**a**) The disassembled product parts for identification by CNN; (**b**) The assembled cam switch.

4. Input Data Preparation for CNN Training

CNN can work reliably for assembly parts recognition, but a problem is the preparation of input data for their training. A very large quantity of input samples need to be prepared, usually several hundred for one assembly part, because it has to be captured with different angular/translation variations and also with different backgrounds and materials. The replacement of real images of assembly parts with their 3D virtual models can significantly accelerate this process. Applying virtual models is also a trend of the Industry 4.0 concept and they can represent the real production process or product. Such virtual models digitally replicate all aspects of real products and they are called digital twins. They consist of 3D models of parts grouped into assemblies with the possibility of data synchronization with a real product, so the first step in the methodology of deep learning implementation into the assisted assembly process is a digital twin creation of the assembled product, in our case a cam switch (Figure 5). This digital twin will serve also as an input model into a mixed reality device for staff training.

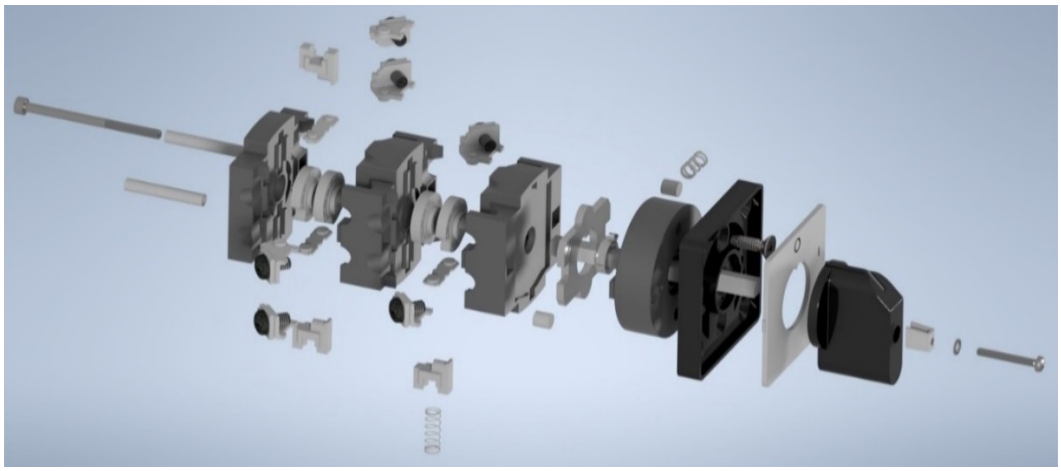


Figure 5. An exploded view of a cam switch digital twin used in the assisted assembly process for implementation into mixed reality devices.

4.1. An Automated 2D Images Generation by the Unreal Engine

In the previous research [24] there was combined Blender 3D software with Python scripting language for an automated generation of the CNN training set, but the Blender rendering engine does not provide cinematic quality of the generated samples. This disadvantage significantly decreases the classification precision of trained convolutional networks by about 20 to 30%. The new approach is based on cinematic rendering from the Unreal Engine combined with Blueprint and Python scripting language. The next new feature is a dynamic collision used for realistic shadow rendering placed under generated samples on a different surface. An example of part position in the 3D inside of Unreal Engine editor setup before dynamic simulation (left) and the part with shadow after simulation (right) is shown in Figure 6.

(**a**) (**b**)

Figure 6. An example project in the Unreal Engine: (**a**) 3D virtual model of a screw in the Unreal Engine editor; (**b**) 3D after dynamic simulation with cinematics texture and shadow.

The basic algorithm is coded in the Blueprint Scripting Language, an example of subprogram for 3D virtual part rotation in Z axis is shown in Figure 7.

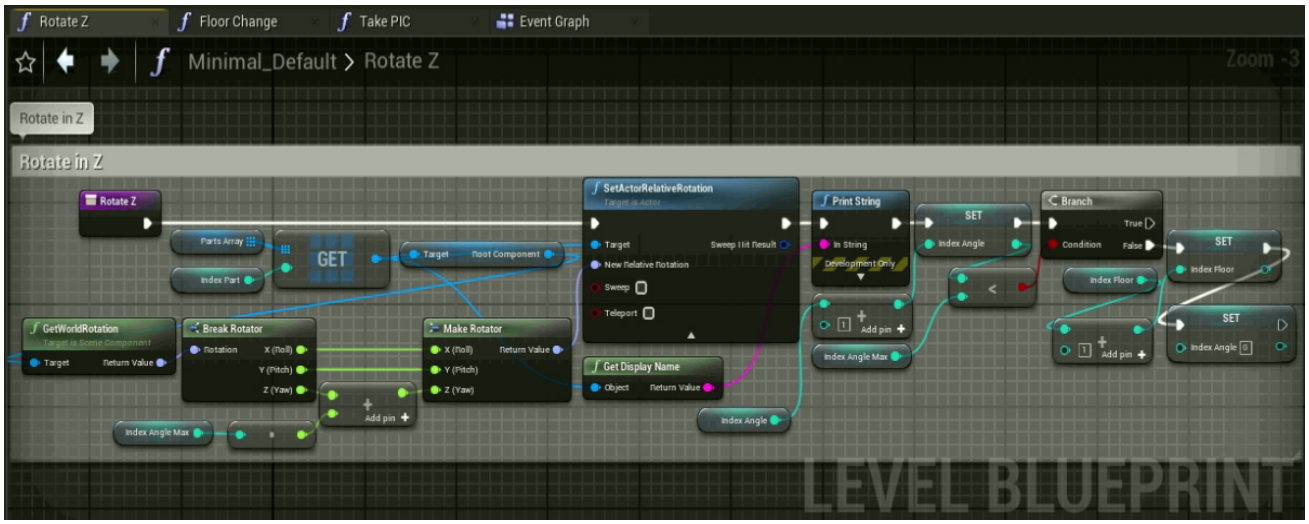


Figure 7. An example of a visual script for automated part rotation in Z-axis using the algorithm coded in the Blueprint Scripting Language.

The initial parameters for 2D sample generation are rotation angle, type of CNN model which provides basic image resolution, number of generated backgrounds as floors, and annotation file type. Basic parameters selection in the Unreal HUD menu before automated generation start is shown in Figure 8. Full assembly of the cam switch consists

of 31 different parts, the basic setup uses 5 different floor textures and the angle of rotation can be set up from 20° to 360° in Z-axis.

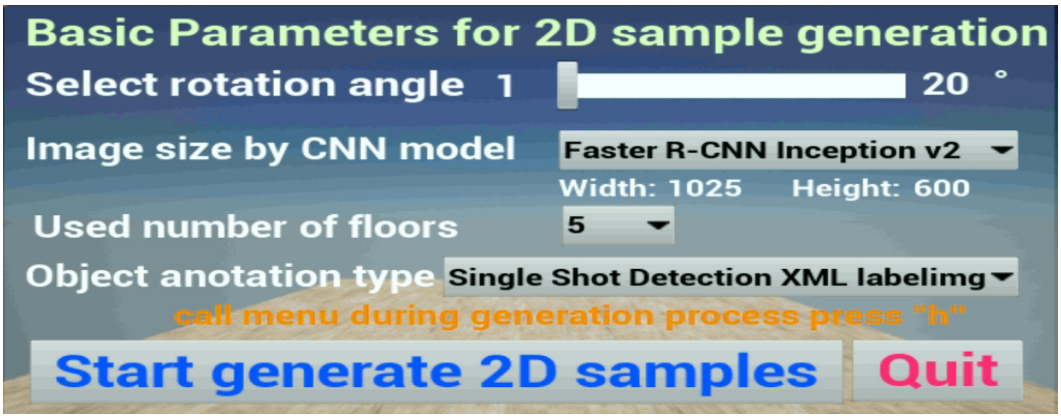


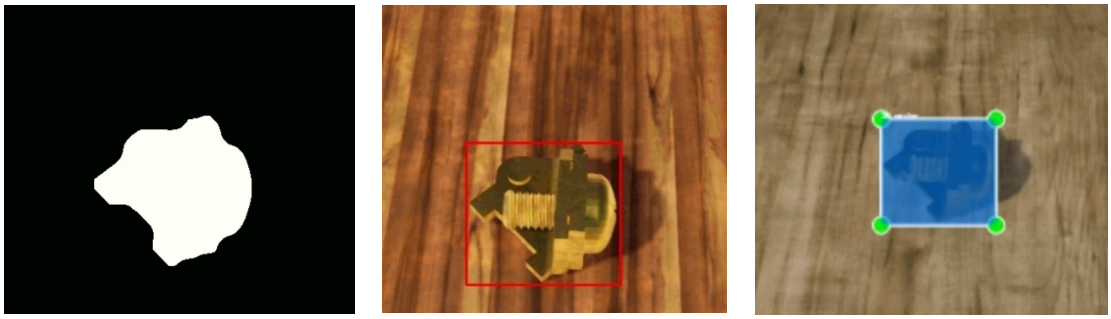
Figure 8. The Unreal Engine application setup of basic parameters for automated generation of 2D samples training and testing set.

4.2. An Automated Annotation by the OpenCV Algorithms

The input condition for automated annotation is a black background and binary threshold to get clear object edges. Two basic annotation methods were selected for generated 2D images from 3D virtual models:

- Single Shot Detection (SSD) annotation by the basic unrotated bounding box in XML format for LabelImg;
- Instance segmentation annotation by a polygon with variable approximation in JSON format for LabelMe.

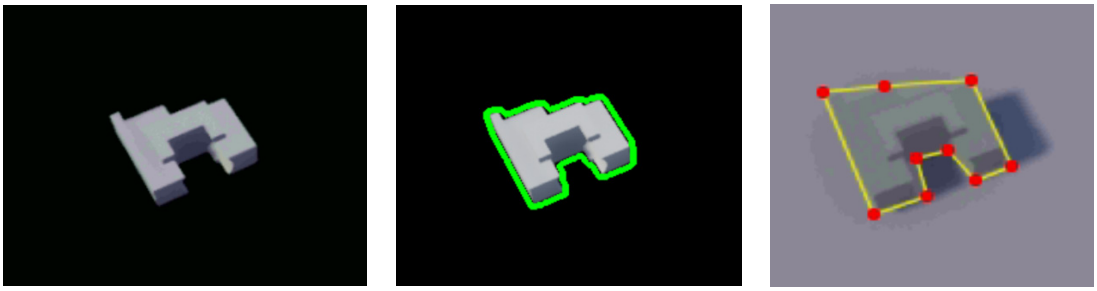
The process of automated SSD annotation and evaluation is shown in Figure 9. The resolution of generated images can be changed exactly for the used CNN model. The first tested model was Faster R-CNN with Inception v2 with default resolution $600 \times 1024 \times 3$. Test samples are separated randomly from the train set for every generated part by a default value of 25%.



(**a**) (**b**) (**c**)

Figure 9. An autogeneration of the part localization by OpenCV for SSD: (**a**) binary threshold; (**b**) contour detection with bounding box generation; (**c**) LabelImg check of XML data generation.

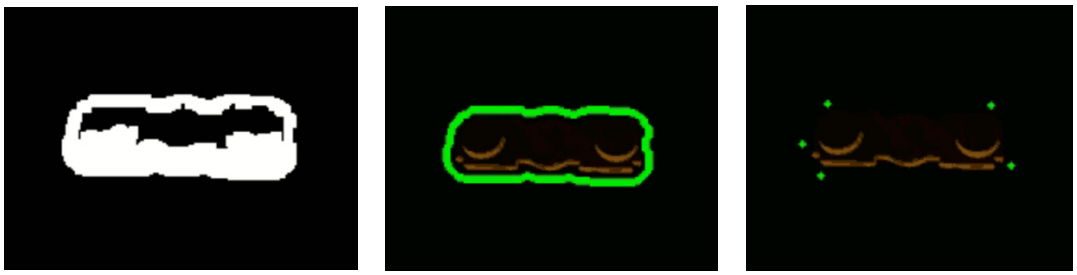
Segmentation needs much more precise thresholding like single shot detection. The closing algorithm was used to get a precise contour of the object. An example of thresholding with object contour closing is shown in Figure 10.



(**a**) (**b**) (**c**)

Figure 10. An autogeneration of the part localization by OpenCV for Segmentation: (**a**) the input image with black background from the Unreal Engine; (**b**) the result of OpenCV algorithm for precise contour generation; (**c**) an example of manual contour selection in VGG Image Annotator.

XML format for SSD is accepted as standard, but instance segmentation has many formats, COCO JSON, CSV, LabelMe JSON, RLE, etc. The simplest JSON structure has LabelMe format with polygon shape, which can be easily implemented to the automated process of contour annotation by OpenCV. An automated contour detection by OpenCV can provide a better contour as a manual process and it can significantly improve CNN instance segmentation after the training process, as can be seen in Figure 11.



(**a**) (**b**) (**c**)

Figure 11. OpenCV detected contour representation: (**a**) binarized image; (**b**) all detected points; (**c**) reduced number of points optimized by Douglas-Peucker algorithm.

4.3. The Generated Training and Testing Sample Set

An example of results from an automated sample generation with XML annotation converted to CSV files is shown in Figure 12.

To start the CNN model training process is only necessary to copy folder *train*, *test*, and two cumulated annotations to *CSV* files to the TensorFlow folder and create TF_record files for training and testing.

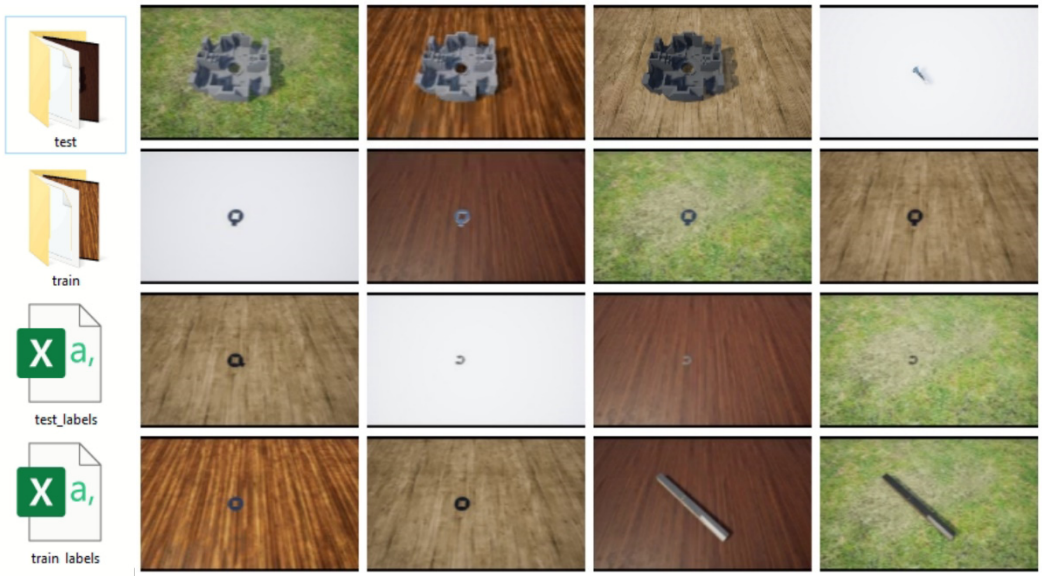


Figure 12. An example of the autogenerated training/tests set with XML bounding box annotation.

All necessary information to identify the sample is encoded in the sample image name:
00007 3**plastic_button** pic or 00007 3**plastic_button** picCV**.png**
where:
00007—number of the generated image [image 7]
3—part identification in the assembly [part 3]
plastic_button—part name
pic—image type
CV—OpenCV generated binary image with black background
.png/.xml/.json—the type of file necessary for TF record algorithm

5. Experimental Results and Implementation into the Assembly Process

An initial experiment of the cam switch parts recognition was executed using a small set of training samples (five per part, 155 samples altogether) with the different floor. Considering this small teaching set the obtained results are acceptable (see Table 1). The training process for Inception V2 is shown in Figure 13, where the unit on the X-axis is the number of cycles and the unit on the Y-axis is mAP.

Table 1. The acquired results by autogenerated training samples based on virtual 3D models.

Pretrained CNN Model	Testing Set	Number of Samples	Classification Results [%]	Training Time [h]
Faster RCNN Inception V2 SSD [1]	virtual	155	67–86	1.35
Mask RCNN Resnet101 [1]	virtual	155	91–95	5.20

[1] Both CNN models were used with a pretrained COCO dataset.

CNN models with single shot detection can be retrained very fast by transfer learning with accepted results within less than 2 h of training without dedicated GPU (for example the used pretrained Faster RCNN Inception V2 SSD reached the required accuracy within 1.35 h). CNN models with segmentation for the same input samples need 4–5 times more time for successful training (for example the used pretrained Mask RCNN Resnet101 reached required accuracy up to 5.20 h). But in contrast to SSD where a model is saved after unpredictable numbers of iteration, the training of models with Segmentation is stored after each epoch. The results of recognition of other images (not training set) of some parts for both tested CNN models (SSD and instance segmentation) are shown in Figure 14.

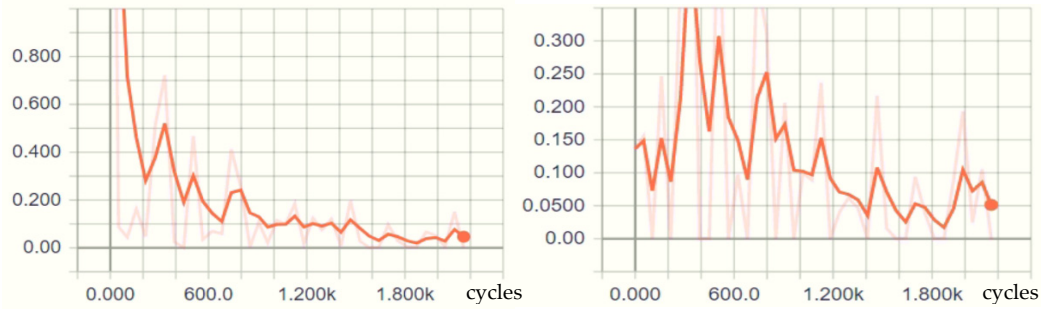


Figure 13. Training graphs for classification (**left**) and localization (**right**).

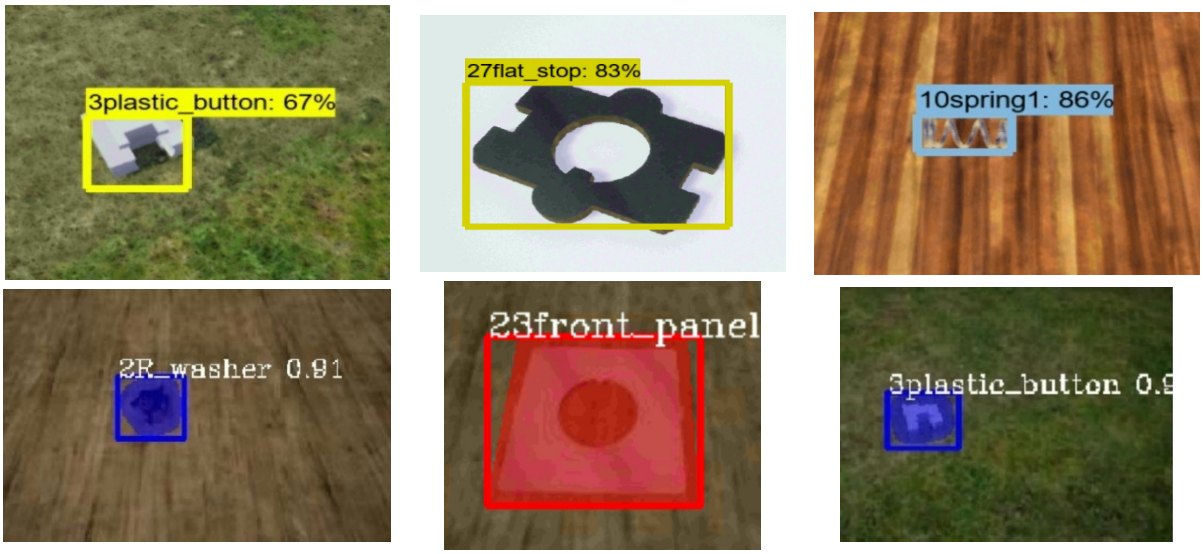


Figure 14. Inference experiments with the SSD trained model (**top**) and instance segmentation (**bottom**).

The inference time experiments with trained CNN models (SSD and Mask) have been performed on many different platforms. The delay results presented in Tables 2 and 3 for inference time is average value from a multiple test in loop for recognition of 40 sample images. The obtained results for CNN model SSD Inception V2 and TensorFlow 1 are in Table 2, for CNN Segmentation model Resnet101 and TensorFlow 2 with Pixelib in Table 3.

Table 2. The acquired inference results for the SSD CNN model.

Testing Environment	Platform Type	Platform Parameters	Delay [ms]
Google Colab	CPU	Intel Xeon 2.30GHz Gen. 6	2726.18
Google Colab	GPU	Tesla P4	358.37
Google Colab	TPU	Cloud TPU	2474.20
NVIDIA APU	Xavier AGX	Saved model native	885.18
NVIDIA APU	AGX	FP32 Tensor RT	445.43
NVIDIA APU	AGX	FP16 Tensor RT	339.41

Table 3. The acquired inference results for the Mask CNN model.

Testing Environment	Platform Type	Platform Parameters	Delay [ms]
NVIDIA APU	Xavier AGX	Maximum power 30W	690.71
Conda GPU	GPU	Nvidia RTX 2060	166.00
Conda CPU	CPU	Intel i9-10900KF 3,7 Ghz	701.12
WSL2 CPU	CPU	i7-9750H 2,7 Ghz	1522.88
WSL2 GPU	AGX	NVIDIA GTX 1660 Ti DirectML	5807.61

The FP16 SSD Inception V2 CNN model can reach about 3 FPS, which is an acceptable parts identification delay for checking worker assembly tasks and collaborative robot assembly status. The experiment with the Mask RCNN segmentation model reached in AGX device about 700 ms delay, which is acceptable in comparison to Desktop PC with high-performance CPU.

The mixed reality device is based on ARM64 architecture which does provide enough power for the execution of the trained inference model. The new approach is to stream video data to NVIDIA Xavier APU which runs an inference model and sends only extracted data: bounding box, contour polygon, and a result of classification as feedback.

The collaborative work cell contains a SMART vision system consisting of three cameras. The primary camera is connected to NVIDIA Xavier AGX (Figure 15a), where is uploaded trained CNN model. The second camera is integrated into mixed reality devices (Figure 15b) and the third camera is integrated into the right hand of the collaborative robot (Figure 15c). The principle of parts detection can be described in these steps:

(1) Get images from the mixed reality device and collaborative robot hand;
(2) Send images to the front of data;
(3) Get images by NVIDIA Xavier embedded device;
(4) Inference images by selected CNN model and acquire part position data;
(5) Send bounding box (contour) and classification value by TCP communication to both devices (mixed reality and collaborative robot).

(**a**) (**b**) (**c**)

Figure 15. SMART vision system of the collaborative work cell: (**a**) CNN processing unit NVIDIA Xavier AGX; (**b**) Mixed reality device camera; (**c**) Collaborative robot ABB Yumi Vision system integrated into the right hand.

The implementation principle of parts recognition into the collaborative work cell is shown in Figure 16.

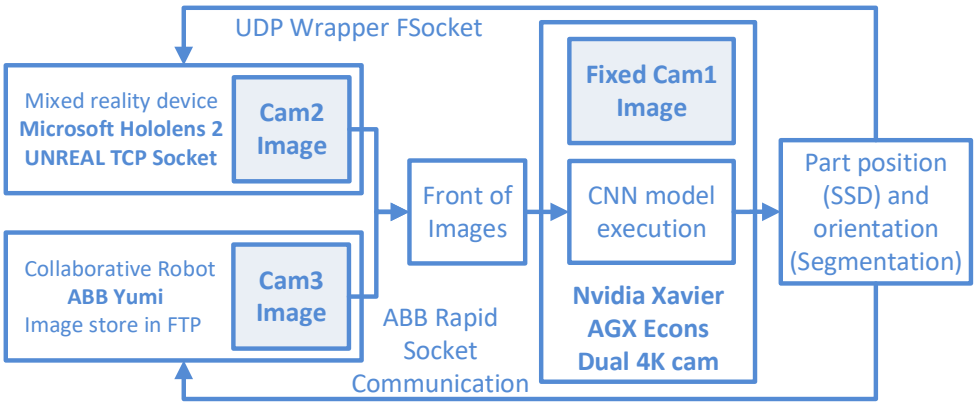


Figure 16. The principle of an experimental assisted assembly work cell with CNN processing unit, mixed reality device, and collaborative robot.

Images captured by all vision systems (static dual 4K e-con cameras connected to NVIDIA Xavier AGX and JetPack 4.4, integrated Cognex 7200 camera in ABB Yumi right hand and Microsoft Hololens 2 internal head camera) are shown in Figure 17.

(**a**) (**b**) (**c**)

Figure 17. Images captured in the assisted assembly work cell by: (**a**) Nvidia Xavier AGX e-con camera; (**b**) Mixed reality Microsoft Hololens 2 head camera; (**c**) ABB Yumi Cognex Vision system.

The application for the mixed reality device Hololens 2 is coded in the same software (Unreal Engine) as a sample generation software but does not use Python programming language and is coded only by Blueprint programming language with UXTool library.

The designed calibration principle of all used vision systems to one Cartesian coordinate is shown in Figure 18 and their synchronization is realized in these steps:

- NVIDIA Xavier AGX system with e-con dual 4K camera is static and default zero position is set as fixed X, Y offset in pixels;
- ABB Yumi collaborative robot position is realized by offset from home point to the axis of the left-hand vision system in Rapid programming language;
- The mixed reality device Microsoft Hololens 2 is synchronized by QR code placed on the lower-left corner of the assembly table and rotation measured by integrated MEMS sensors.
- The information about part position adjusted by QR code is shared between all devices as Part identification, where position is X, Y; size W, H, and contour of the detected part is represented by array of points C.

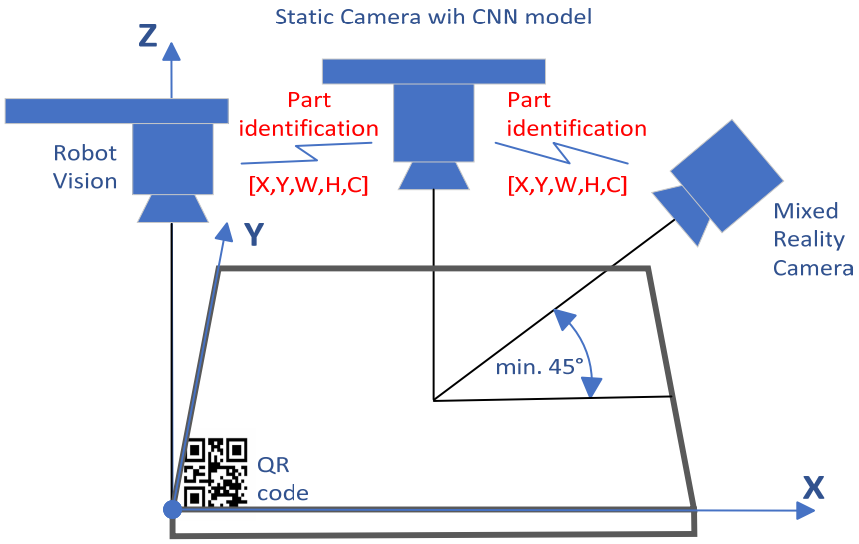


Figure 18. The principle of vision systems synchronization to one Cartesian coordinate.

Current deep learning frameworks provide only image augmentation, which only reduces the number of images that are needed to be prepared. It means, the most monotonous works in deep learning implementation to real application still exist. This is the main reason why is not profitable to use deep learning in small series assembly tasks. On the other hand, an automated generation of training samples from CAD models, which are available before production starts, can help to implement more assisted assembly solutions into practice.

The research field in an automated generation of training samples for CNN models from 3D virtual models has a big potential to expand. Current progress in GPU with real-time raytracing can provide new rendering possibilities to reach cinematic quality in object visualization and fast preparation of virtual samples. An interesting project is Kaolin from NVIDIA, a modular differentiable rendering for applications like high-resolution simulation environments, though it is still only as a library under research.

The early research progress is improving the presented tested solution with parts overlay recognition implemented into the Unreal Engine. An example of the first testing implementation with overlays is shown in Figure 19.

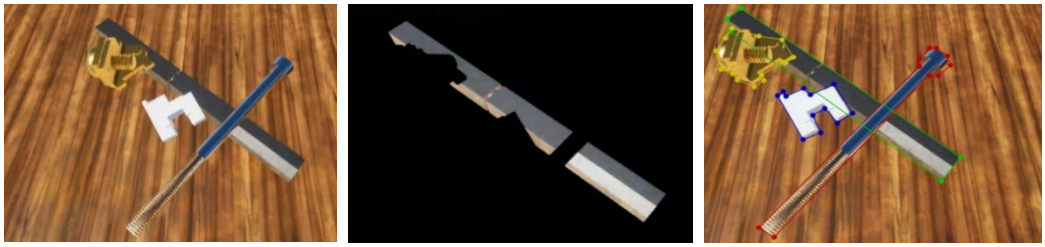


Figure 19. Parts overlay early research implemented into the Unreal Engine.

6. Conclusions

An automated generation of training samples based on 3D virtual models is a new approach in the field of deep learning that can save many hours of manual work. The presented research in the article introduces a methodology of CNN training for deep learning implementation into the assisted assembly process. This methodology was evaluated in an experimental SMART manufacturing system with assisted assembly work cell using cam switch as chosen assembly product from real production where is still used fully manual assembly process [31].

To summarize, those experiments have been performed and these main research results have been acquired in the field of CNN training for parts recognition in the assisted assembly process:

- A Blueprint program for an automated generation of 2D images from 3D virtual models as CNN training set in the Unreal Engine 4 software has been created;
- A Python algorithm with OpenCV library has been implemented for an automated image annotation for single shot detection as XML and instance segmentation as JSON;
- Two separate CNN models (SSD and Mask) have been trained in TensorFlow 2 framework for evaluation of the proposed methodology of deep learning implementation into the assisted assembly process;
- Inference experiments with trained CNN models on different platforms including an embedded APU have been performed and evaluated;
- Parts recognition data transfer among mixed reality devices, a collaborative robot and an embedded APU device has been designed and implemented.

The future works can be divided into more steps because there are further plans mainly in extending the current software:

- To increase rendered image quality to cinematic by real-time raytracing implemented in the new generation of GPUs;
- To implement an automated parts overlay recognition, which can be simply solved using Unreal Newton Physics and automatic switching to black texture for objects overlapping as input for OpenCV annotation;
- Transform current experimental project for free use as a plugin into the Unreal Engine software.

Author Contributions: Conceptualization, K.Ž. and J.P.; methodology, M.B.; software, P.L.; validation, V.H. and A.I.; formal analysis, M.B.; investigation, A.H.; resources, J.P.; data curation, K.Ž.; writing—original draft preparation, K.Ž.; writing—review and editing, J.P.; visualization, J.D.; supervision, A.H.; project administration, K.Ž.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0590 and also by the projects VEGA 1/0700/20, 055TUKE-4/2020 granted by the Ministry of Education, Science, Research and Sport of the Slovak Republic.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The article was written as a result of the successful solving of the Project of the Structural Funds of the EU, ITMS code: 26220220103.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Realyvásquez-Vargas, A.; Arredondo-Soto, K.C.; García-Alcaraz, J.L.; Márquez-Lobato, B.Y.; Cruz-García, J. Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company. *Robot. Comput. Manuf.* **2019**, *57*, 315–328. [CrossRef]
2. Malik, A.A.; Bilberg, A.; Katalinic, B. Framework to Implement Collaborative Robots in Manual Assembly: A Lean Automation Approach. In Proceedings of the 29th International DAAAM Symposium 2018, Zadar, Croatia, 8–11 November 2017; Volume 1, pp. 1151–1160.
3. Kim, W.; Peternel, L.; Lorenzini, M.; Babič, J.; Ajoudani, A. A Human-Robot Collaboration Framework for Improving Ergonomics During Dexterous Operation of Power Tools. *Robot. Comput. Manuf.* **2021**, *68*, 102084. [CrossRef]
4. Liberatore, M.J.; Wagner, W.P. Virtual, mixed, and augmented reality: A systematic review for immersive systems research. *Virtual Real.* **2021**, 1–27. [CrossRef]
5. Khatib, M.; Al Khudir, K.; De Luca, A. Human-robot contactless collaboration with mixed reality interface. *Robot. Comput. Manuf.* **2021**, *67*, 102030. [CrossRef]

6. Akkaladevi, S.C.; Plasch, M.; Maddukuri, S.; Eitzinger, C.; Pichler, A.; Rinner, B. Toward an Interactive Reinforcement Based Learning Framework for Human Robot Collaborative Assembly Processes. *Front. Robot. AI* **2018**, *5*, 126. [CrossRef] [PubMed]
7. Ghadirzadeh, A.; Chen, X.; Yin, W.; Yi, Z.; Bjorkman, M.; Kragic, D. Human-Centered Collaborative Robots With Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 566–571. [CrossRef]
8. Liu, H.; Wang, L. An AR-based Worker Support System for Human-Robot Collaboration. *Procedia Manuf.* **2017**, *11*, 22–30. [CrossRef]
9. Takaseki, R.; Nagashima, R.; Kashima, H.; Okazaki, T. Development of Anchoring Support System Using with AR Toolkit. In Proceedings of the 2015 7th International Conference on Emerging Trends in Engineering & Technology (ICETET), Kobe, Japan, 18–20 November 2015; pp. 123–127.
10. Dehghani, M.; Acikgoz, F.; Mashatan, A.; Lee, S.H. (Mark) A holistic analysis towards understanding consumer perceptions of virtual reality devices in the post-adoption phase. *Behav. Inf. Technol.* **2021**, 1–19. [CrossRef]
11. Seiger, R.; Kühn, R.; Korzetz, M.; Aßmann, U. HoloFlows: Modelling of processes for the Internet of Things in mixed reality. *Softw. Syst. Model.* **2021**, 1–25. [CrossRef]
12. Allcoat, D.; Hatchard, T.; Azmat, F.; Stansfield, K.; Watson, D.; Von Mühlenen, A. Education in the Digital Age: Learning Experience in Virtual and Mixed Realities. *J. Educ. Comput. Res.* **2021**. [CrossRef]
13. Kesim, M.; Ozarslan, Y. Augmented Reality in Education: Current Technologies and the Potential for Education. *Procedia Soc. Behav. Sci.* **2012**, *47*, 297–302. [CrossRef]
14. Židek, K.; Piteľ, J.; Adámek, M.; Lazorík, P.; Hošovský, A. Digital Twin of Experimental Smart Manufacturing Assembly System for Industry 4.0 Concept. *Sustainability* **2020**, *12*, 3658. [CrossRef]
15. Luscinski, S.; Ivanov, V. A simulation study of Industry 4.0 factories based on the ontology on flexibility with using Flexsim®software. *Manag. Prod. Eng. Rev.* **2020**, *11*, 74–83. [CrossRef]
16. Martinez, G.S.; Sierla, S.; Karhela, T.; Vyatkin, V. Automatic Generation of a Simulation-Based Digital Twin of an Industrial Process Plant. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3084–3089.
17. Tomko, M.; Winter, S. Beyond digital twins—A commentary. *Environ. Plan. B Urban Anal. City Sci.* **2019**, *46*, 395–399. [CrossRef]
18. Shubenkova, K.; Valiev, A.; Shepelev, V.; Tsiulin, S.; Reinau, K.H. Possibility of Digital Twins Technology for Improving Efficiency of the Branded Service System. In Proceedings of the 2018 Global Smart Industry Conference (GloSIC), Chelyabinsk, Russian, 13–15 November 2018; pp. 1–7.
19. David, J.; Lobov, A.; Lanz, M. Learning Experiences Involving Digital Twins. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3681–3686.
20. Caputo, F.; Greco, A.; Fera, M.; Macchiaroli, R. Digital twins to enhance the integration of ergonomics in the workplace design. *Int. J. Ind. Ergon.* **2019**, *71*, 20–31. [CrossRef]
21. Malik, A.A.; Brem, A. Digital twins for collaborative robots: A case study in human-robot interaction. *Robot. Comput. Manuf.* **2021**, *68*, 102092. [CrossRef]
22. Židek, K.; Piteľ, J.; Hošovský, A. Machine learning algorithms implementation into embedded systems with web application user interface. In Proceedings of the IEEE 21st International Conference on Intelligent Engineering Systems 2017 (INES 2017); IEEE. 2017; pp. 77–81.
23. Židek, K.; Hosovsky, A.; Piteľ, J.; Bednár, S. Recognition of assembly parts by convolutional neural networks. In *Advances in Manufacturing Engineering and Materials; Lecture Notes in Mechanical Engineering*; Springer: Cham, Germany, 2019; pp. 281–289.
24. Židek, K.; Lazorík, P.; Piteľ, J.; Hošovský, A. An Automated Training of Deep Learning Networks by 3D Virtual Models for Object Recognition. *Symmetry* **2019**, *11*, 496. [CrossRef]
25. Pollák, M.; Baron, P.; Tellišková, M.; Kočiško, M.; Török, J. Design of the web interface to manage automatically generated production documentation. *Tech. Technol. Educ. Manag. TTEM* **2018**, *7*, 703–707.
26. Gopalakrishnan, K. Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review. *Data* **2018**, *3*, 28. [CrossRef]
27. Mao, K.; Lu, D.; E, D.; Tan, Z. A Case Study on Attribute Recognition of Heated Metal Mark Image Using Deep Convolutional Neural Networks. *Sensors* **2018**, *18*, 1871. [CrossRef] [PubMed]
28. Varga, M.; Jadlovský, J.; Jadlovská, S. Generative Enhancement of 3D Image Classifiers. *Appl. Sci.* **2020**, *10*, 7433. [CrossRef]
29. Su, H.; Qi, C.R.; Li, Y.; Guibas, L.J. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7 December 2015; pp. 2686–2694.
30. Su, Y.; Rambach, J.; Pagani, A.; Stricker, D. SynPo-Net—Accurate and Fast CNN-Based 6DoF Object Pose Estimation Using Synthetic Training. *Sensors* **2021**, *21*, 300. [CrossRef] [PubMed]
31. Lazár, I.; Husár, J. Validation of the serviceability of the manufacturing system using simulation. *J. Effic. Responsib. Educ. Sci.* **2012**, *5*, 252–261. [CrossRef]