applied sciences

MDPI

Article

Hybrid Quality Inspection for the Automotive Industry: Replacing the Paper-Based Conformity List through Semi-Supervised Object Detection and Simulated Data

Isabel Rio-Torto [1,*,†], Ana Teresa Campaniço [2,*,†], Pedro Pinho [1], Vitor Filipe [2,3] and Luís F. Teixeira [1,3]

[1] Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal; up201605166@fe.up.pt (P.P.); luisft@fe.up.pt (L.F.T.)
[2] School of Science and Technology, University of Trás-os-Montes and Alto Douro, 5000-801 Vila Real, Portugal; vfilipe@utad.pt
[3] INESC TEC—INESC Technology and Science, 4200-465 Porto, Portugal
* Correspondence: icrto@fe.up.pt (I.R.-T.); acampanico@utad.pt (A.T.C.)
† These authors contributed equally to this work.

Abstract: The still prevalent use of paper conformity lists in the automotive industry has a serious negative impact on the performance of quality control inspectors. We propose instead a hybrid quality inspection system, where we combine automated detection with human feedback, to increase worker performance by reducing mental and physical fatigue, and the adaptability and responsiveness of the assembly line to change. The system integrates the hierarchical automatic detection of the non-conforming vehicle parts and information visualization on a wearable device to present the results to the factory worker and obtain human confirmation. Besides designing a novel 3D vehicle generator to create a digital representation of the non conformity list and to collect automatically annotated training data, we apply and aggregate in a novel way state-of-the-art domain adaptation and pseudo labeling methods to our real application scenario, in order to bridge the gap between the labeled data generated by the vehicle generator and the real unlabeled data collected on the factory floor. This methodology allows us to obtain, without any manual annotation of the real dataset, an example-based F1 score of 0.565 in an unconstrained scenario and 0.601 in a fixed camera setup (improvements of 11 and 14.6 percentage points, respectively, over a baseline trained with purely simulated data). Feedback obtained from factory workers highlighted the usefulness of the proposed solution, and showed that a truly hybrid assembly line, where machine and human work in symbiosis, increases both efficiency and accuracy in automotive quality control.

Keywords: automotive industry; hybrid assembly lines; Industry 4.0; information visualization; quality inspection; semi-supervised cross domain object detection; simulated data

Citation: Rio-Torto, I.; Campaniço, A.T.; Pinho, P.; Filipe, V.; Teixeira, L.F. Hybrid Quality Inspection for the Automotive Industry: Replacing the Paper-Based Conformity List through Semi-Supervised Object Detection and Simulated Data. *Appl. Sci.* **2022**, *12*, 5687. https://doi.org/10.3390/app12115687

Academic Editors: Nuno Carlos Sousa Rodrigues and Paulo Manuel Almeida Costa

Received: 16 April 2022
Accepted: 1 June 2022
Published: 3 June 2022

1. Introduction

While automation is commonplace in the automotive industry, many processes still rely on the manual approach [1–3]. This is particularly true in quality control, as with the growing trend in vehicle customization to drive product and production complexity to higher levels, a much tighter control is required to ensure the same high-quality and safety standards are met [1,4]. However, the traditional non-conformities detection method relies on workers to visually compare the assembled vehicle with a mental image of the items on a paper list (Figure 1), in real time. This means that the increase in complexity leads to higher memorization workload, and consequently excessive mental fatigue and other known sources of human error [1,3–5].

Despite the growing adoption of smart factory strategies to increase cost-efficiency and productivity [6], the use of this paper conformity list is still quite customary [1]. Because it depends on highly trained human observers, this method is quite tolerant towards environmental and minor product variations. However, it relies heavily on the worker's

memorization and recall abilities, which must be performed accurately, quickly and reliably in every single inspection, despite any model updates that add and/or remove vehicle components. This entails major drawbacks: it induces worker ocular and mental fatigue, inattentiveness, differences in proficiency levels, costs associated with constant worker training, etc. [4]. Another major issue with the effectiveness of the paper solution is its inherent inflexible design. The information is not individualized according to the different workstations, it does not signal previously detected non-conformities, be it on a particular vehicle or the most commonly detected in a given model, it can not exchange information with the factory's network, nor can it be updated in real time, among other limitations [1,5]. In addition, in our specific context, for a non-conformity report to be submitted the worker must physically walk to access a terminal rather than do so wirelessly, which is another major inefficiency and source of physical fatigue.

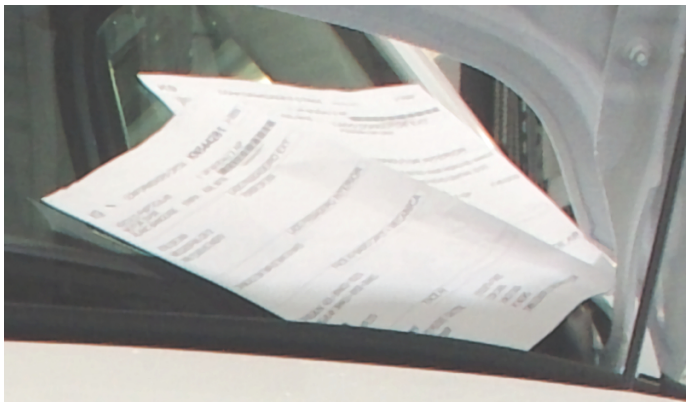

Figure 1. Paper conformity list.

Thus, we propose a hybrid quality inspection approach, tailored for the assembly line of a Portuguese car manufacturer, and the replacement of the paper conformity list with a digital counterpart. We combine the automatic detection of the assembled vehicle parts and the identification of existing non-conformities with an information visualization module on a wearable device to present the results to the factory worker. The workers can then validate the flagged non-conformities, produce the quality control report and provide feedback to the detection system, all without having to leave their posts and without causing mental fatigue, which in turn increases the workers' performance. Together with the design of this novel hybrid quality inspection system, we developed a 3D vehicle generator to integrate the visualization module and use it to generate data for the detection stage, therefore eliminating altogether the annotation cost of the proposed solution. To compensate for the domain gap between simulated and real images collected on the assembly line, we resort to state-of-the-art domain adaptation methods [7,8] and pseudo labeling (PL) [9], joining these techniques in a novel way to tailor them to our real application scenario. Finally, we explore the potential advantages of a fixed camera setup, instead of the current generic unconstrained approach. Therefore, our main contributions are:

- a hybrid end-to-end system for automatic quality control in the automotive industry;
- replacement of the paper based conformity list by a digital alternative;
- application of state-of-the-art domain adaptation and pseudo labeling techniques in a novel joint approach to tackle semi-supervised cross domain object detection for quality control in the automotive industry;
- improvement of a previously proposed [10] multi-purpose 3D vehicle generator.

The rest of this manuscript is organized as follows: Section 2 provides and in-depth review of the state-of-the-art on quality inspection for the automotive industry. Section 3 introduces the materials and methods, starting with the overall system architecture, and afterwards diving deeper into each of its modules. Section 4 is related to the validation of

the proposed solution and the discussion of the obtained results and Section 5 summarizes the main conclusions and points to future improvements and research directions.

2. State-of-the-Art

The recent advancements in machine learning algorithms and the increase in computational power, as well as the smart manufacturing strategies brought by Industry 4.0, have led to the development of computer vision based solutions to address the multiple facets of automotive production. These can range anywhere from the replacement of expensive sensors with cameras to visual inspection in manufacturing and assembly, along many other fields outside the production industry [11]. It is such a profitable field, with such a high demand for cutting-edge solutions, that commercial companies develop several hardware, software and services specifically to answer the demanding conditions and needs of the industrial sector.

The quality control performed through visual inspection is an extremely important part of the production process. For example, in fault detection, machine and deep learning algorithms are used to visually identify the occurrence of quality issues, or other structural flaws in part manufacture and/or assembly. Detection of such faults helps not only with the correction of the error as soon as it occurs, but its classification can diagnose its exact cause, thus locating the machine failure that provoked it [12]. Some examples include the classification of car seat backrests through the combination of speeded up robust features (SURF) and a convolutional neural network (CNN) by Sun et al. [13], and the inspection of defects such as cracks, tears or unwanted inclusions, and other elements, via region based CNN (R-CNN) with a modified stochastic gradient descent with momentum (SGDM) by Kuric et al. [14]. As an example of fault detection done on welding and solder joints, Pei and Chen [15] suggest an inspection on door panels through a machine learning algorithm combined with template matching (based on the Canny edge detector and sequential similarity detection) for the former and the Hough transform and image segmentation for the latter.

The classification of surface defects, on the other hand, focuses on the inspection of fabrics, glass, painted surfaces, etc. to detect any scratches, dents and other minor imperfections that impact the aesthetic quality of the vehicle, and thus can seriously impact the brand's image and consumer opinion. This task can be particularly challenging, as many of the defects can be quite small, faint and irregular in shape, hidden in the highly reflective surfaces and by the uneven lighting conditions [16]. Examples range from the detection of tiny defects on paint surfaces with a combination of a deep convolutional neural network (DCNN) and YOLOv3 by Chang et al. [17], to the detection of multiple manufacturing defects on wheel hubs (such as scratches, indentations or oil pollution) through the use of a Faster R-CNN by Sun et al. [18].

However, despite the constant strides in the field towards automation, workers still play an important role in the production process. Therefore, there is a strong investment in the creation of a multitude of tools to help reduce their workload and increase their productivity [19]. Many are simple maintenance software tools that provide workers with information on that specific vehicle model, often in the form of augmented reality (AR), or with instruction manuals on the parts and procedures. Some examples are the AR smart devices developed by Volkswagen, Bosch, Hyundai and Ferrari [19,20], or the research done by Lima et al. [21] on the development of an AR markerless tracking system of multiple car parts, as the base for training and maintenance solutions. Another form of inspection software are devices developed to support workers in quality control tasks. These replace the previous paper description of the vehicle with a more flexible, responsive and efficient solution that can communicate directly with the smart factory [1,5]. Ford, for example, has developed a smartphone app that serves as a wrist-worn Portable Quality Assurance Device [22]. Other information visualization research examples are the creation of a smart visualization framework for door assembly quality control by Gewohn et al. [1], and of a cognitive assistant AR head-mounted display (HMD) for vehicle defect detection by Chouchene et al. [20].

While these solutions definitely help the workers deal with the ever growing amount of product customization and complexity, not all of them take full advantage of what both humans and machines can offer when working in unison. Hybrid production lines are a more cost-efficient solution than the traditional model, as they combine automation's ability of processing and analyzing large volumes of data in a fast, repetitive, and precise way with the human's higher adaptability and cognitive flexibility. This results in much more responsive and adaptable production lines in the face of sudden change and updates, while requiring less adjustments to the systems already set in place [1,4,23]. However, proper cooperation cannot be achieved if the means of communication are not adequately designed. More often than not the industry focuses more on functionality and reliability, completely neglecting more accessibility and usability related factors that make interfaces easier to read and faster and more intuitive to navigate. This leads to losses in human performance due to a lack of understanding of the worker's needs [2,24,25], a serious issue in a fast paced work environment with zero tolerance for error.

3. Materials and Methods

3.1. System Architecture

The quality control context our solution aims to specifically improve is the final workstation of the vehicle assembly line, where non-conformities are detected by a human worker circling around the vehicle to inspect it. This factory in particular assembles the multi-brand k9 van model, which is customized according to each customer order, meaning there is no set pattern on which model variant the worker will inspect on any given day, outside of some being far more common than others. The worker is expected to compare the assembled vehicle with the paper list, but it was reported that many instead only consult it when faced with the less common models. The reason given was workers must perform parallel tasks that require their hands to be free, so short-term memorization becomes preferable to constant paper consultation. However, this exacerbates the aforementioned problem of mental fatigue. As for the inspection submission process in the observed factory, the worker has to abandon his post and walk to a terminal at the end of the line to manually input the report into the database, which is an added source of physical fatigue.

Therefore, to achieve our goal of addressing these issues via the replacement of the paper conformity list with a digital counterpart, our hybrid quality control support tool must [10,26,27]:

- Describe the conforming vehicle in the factory's database, along with other relevant metrics;
- Access the information via scanning of the barcode placed on the assembled vehicle;
- Use an automated detection system to perform the initial quality inspection of the vehicle;
- Use a wearable device (e.g., Smartglasses) to display the vehicle's conformity list alongside the results of the non-conformity detection, both in a visual format;
- Allow the worker to confirm or cancel each of the results, or submit undetected non-conformities to the final inspection report;
- Allow the worker to attach a photo and/or text report to each individual non-conformity;
- Update the database information with the results of the automated detection and the final inspection report submitted wirelessly by the worker;
- Use the submitted feedback to refine the automated detection system and improve its performance.

The final design of the quality control support tool is divided into two sections: the automated detection, and the human interaction modules (Figure 2). The automated detection of non-conformities is performed at the beginning of the workstation's line, where a set of mounted cameras are installed to capture the vehicle from multiple, fixed angles as it enters, in order to reduce the system's error. The module begins by detecting the vehicle's barcode sticker and use it to fetch the respective information from the factory's database. It then takes the list of parts that form the fully conforming vehicle and compares

them to the detected ones in the assembled vehicle. The result of the comparison serves as that vehicle's non-conformity list, which is stored in the database.

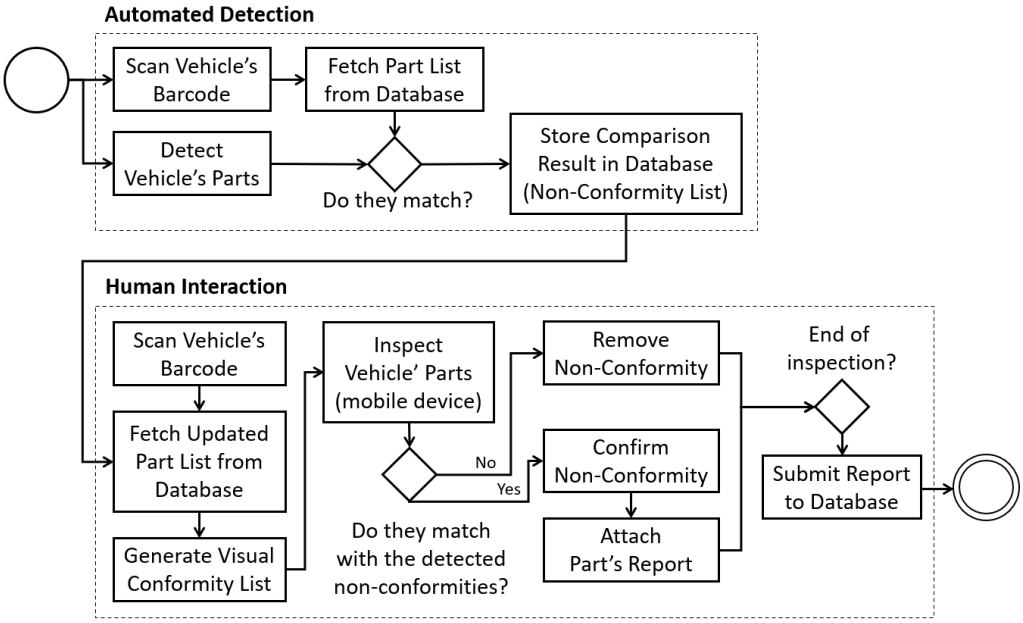

Figure 2. System architecture of the final quality control support tool, that combines automated non-conformity detection with human feedback.

The human interaction (or information visualization) module, on the other hand, involves the visualization of the digital information on a wearable device, which the workers use to perform the inspection on each vehicle. When the worker uses the device's camera to scan the barcode, the module accesses the factory's database and fetches the updated part list. That information is used to generate a visual representation of the conformity list of the parts specific to that quality control post, with all the non-conforming parts flagged by the automated detection highlighted in red. The worker circles the assembled vehicle to perform the inspection on each side, comparing the observed parts with the ones presented on the wearable device. The worker uses the interface to confirm if the non-conformity results of the automated detection highlighted on the device are indeed correct, or removes the incorrect ones. For each detected non-conformity they attach the respective individual report in the form of photos and/or descriptive text of the assembled part. Once the inspection is completed all the information is submitted wirelessly to the database before the worker moves on to the next vehicle.

This hybrid approach allows us to take advantage of what both sides have to offer to mitigate the types of errors produced by each [1,3,4,28]. On one hand, the automated detection system, paired with a visual representation of the parts, reduces memorization over reliance and fatigue induced oversight in the workers. On the other hand, the workers' expertise and versatility is leveraged to provide feedback to the automated detection and correct its mistakes, in a continuous refinement loop.

3.2. Vehicle Generator

Alongside the quality control support tool, we developed a 3D vehicle generator of the multi-brand k9 van model (kSim9). kSim9 is a multi-purpose simulator, capable of converting the vehicle's digital information into a full 3D visual representation of the correctly assembled vehicle, as well as any other possible combinations. This fully autonomous module was developed as a time and cost saving tool, as it can be used as a training simulator, both for workers [29] and detection algorithms based on simulated data [10], among many other applications. By not depending on still images to form its database, all parts and components can be viewed from any angle, in any lighting condition,

making the updating process of both the vehicle models and detection system less time and resource consuming.

To develop the kSim9, 3891 photos of commercial and family models of Citroën and Peugeot vehicles were taken at the Peugeot Société Anonyme (PSA) Mangualde final quality control workstation. From those, 259 distinct components were identified, with only 43 of them being identified as variable (Table 1). These 43 variable components resulted in 53 classes, considering that we distinguish left and right variants of the same object. For a detailed description of the resulting class taxonomy and their corresponding objects, please refer to Table A1 in Appendix A.

Table 1. List of all variable component types according to their planes of location.

Location	Front	Back	Sides	Top
Components	Fog lights Fog light embellishers Grille rim Bumper bar	Top light Window Window hinges Wiper Door handle Side lights Brand logo Model logo Bumper Bumper bar	Side panel Front door pillar Mirror XTR logo Door handles Back door Back door window Window panel Rear window Rail Tire covers	Roof bars

All elements were modeled in Blender 2.8 (https://www.blender.org, accessed on 5 September 2020) and imported to Unity3D (https://unity.com, accessed on 15 November 2021) to build the validation tool used by the factory management and expert observers to confirm the reliability rate of the generated vehicles prior to the development of any software that would use the kSim9 as one of its components. The generator was deemed as an accurate visual representation of the conformity list sample in use, with a reliability rate of 100% [29].

This generator is crucial for providing training data to the automated detection module. Since we do not have ground-truth annotations for the real images collected on the shop floor, kSim9 provides us with an unlimited source of data, with the flexibility to explore several camera angles and lighting conditions, achieving as much variability as desired. More importantly, this simulated data are automatically annotated with object detection bounding boxes after some simple processing, as detailed in [10], which eliminates the annotation process altogether.

Previous works [10,30] already made use of this tool to generate a simulated dataset. We now introduce an updated version that includes refinements of the vehicles and their customization options, as well as a curation of the codes given to each vehicle part. Other improvements encompass more realistic colors and textures (first two columns of Figure 6). This new version includes 40 frames per each of the different 49 car configurations for training and 10 for validation, resulting in a total of 1960 and 490 images in each set, respectively, and a total of 53 macro categories and 150 content-based image retrieval (CBIR) codes (for an explanation of the difference between the two please refer to Section 3.3.2). The previous dataset [10] had around 17 times more images, but preliminary experiments showed that with this smaller counterpart the training time decreased significantly and the performance of the system on simulated images was not significantly hindered and even improved on real images (further details in Section 4.1). Another difference between both datasets is related to the dimensions of the images, where instead of images of 512×512 pixels in size, we generate images with the same resolution as the real ones, i.e., 1280×720 pixels, in an effort to further approximate the two domains.

For the sake of consistency and comparability with previous works [10,30], we maintain the real test set of 10 videos of 10 different vehicles collected on the real assembly line,

resulting in 100 test images. Note that we do not have bounding box annotations for these images; we have, for each video/car, a list of its constituent parts.

3.3. Automated Detection

3.3.1. Related Work

Deep learning-based solutions require large amounts of labeled data, which is time- and resource-wise costly to obtain. This is aggravated in object detection, where each object needs to be localized and classified. Using synthetic data from simulators is a viable alternative; although developing a 3D model is challenging and time consuming, it is a one-time investment that can produce an unlimited amount of automatically labeled data [10,31] and the resulting model can be used for visualization.

Notwithstanding, there is an inherent domain gap between simulated and real data, that can be bridged by several strategies [32], including image-to-image translation. An unpaired translation model converts one domain into the other, reducing distribution shift such that the detector performs better on the target data [33–35]. However, the detector never sees images from the target domain, given that they are usually unlabeled. Pseudo labeling is able to leverage the unlabeled instances by fine-tuning the detector on highly-confident predictions, progressively improving them [32,36].

Inoue et al. [37] combine both paradigms to perform cross domain object detection. However, traditional pseudo labeling was not designed for object detection, so Liu et al. [9] propose an unbiased mean teacher to tackle the inherent foreground/background imbalance of these tasks. After training on labeled source data, a teacher model generates pseudo labels for the unlabeled target domain, which are used as supervision for a student model, that, in turn, updates the teacher with what it has learned via Exponential Moving Average (EMA). The process is repeated so both evolve jointly and the pseudo labels given to the student are continuously being improved. Thus, the teacher can be regarded as a temporal ensemble of the student at different time steps.

The teacher-student learning paradigm suffers from bias towards the source domain, so Deng et al. [38] convert target images using a Cycle Generative Adversarial Network (CycleGAN) [7] to give as inputs to the teacher, while the student receives original target images, to perform knowledge distillation. The student is also trained with both original and target-like source images, mitigating its bias towards the source domain. Although powerful, this approach requires translation in both directions, implying having either a cyclic model or training two translation models. Therefore, we hypothesize that combining the approach from [37] with an object detection-specific pseudo labeling method such as [9] might be more suitable for our particular application.

Despite the plethora of works on semi-supervised cross domain object detection using simulated data, this is, to the best of our knowledge, the first work to incorporate all these concepts and methods to automatic quality control in the automotive industry.

3.3.2. Baseline

The automated inspection of the assembled vehicle is performed by an ad hoc hierarchical deep learning-based architecture introduced in [10] and improved upon in [30]. It is represented at the top of Figure 3: in a first stage, Detectron2 [39] performs object detection, considering macro categories, i.e., the location (front, back, left and right) and part (e.g., mirror, light, etc.) codes. Each detected object is cropped and given to the CBIR module of its detected category. In this second stage, a feature extraction network (ResNeXt-50 32x4d [40] pretrained on ImageNet) obtains a latent representation of the cropped object, which is compared via Euclidean distance to the representations of all the training set objects of that same macro class. Then, the part codes of the top-3 instances with the lowest distances are voted to obtain the predicted code of the object. This code now includes the brand, model and material information for each object (e.g., the light of a commercial Peugeot). After all images of a vehicle are processed, a list of the detected parts is generated for comparison with the corresponding production list.

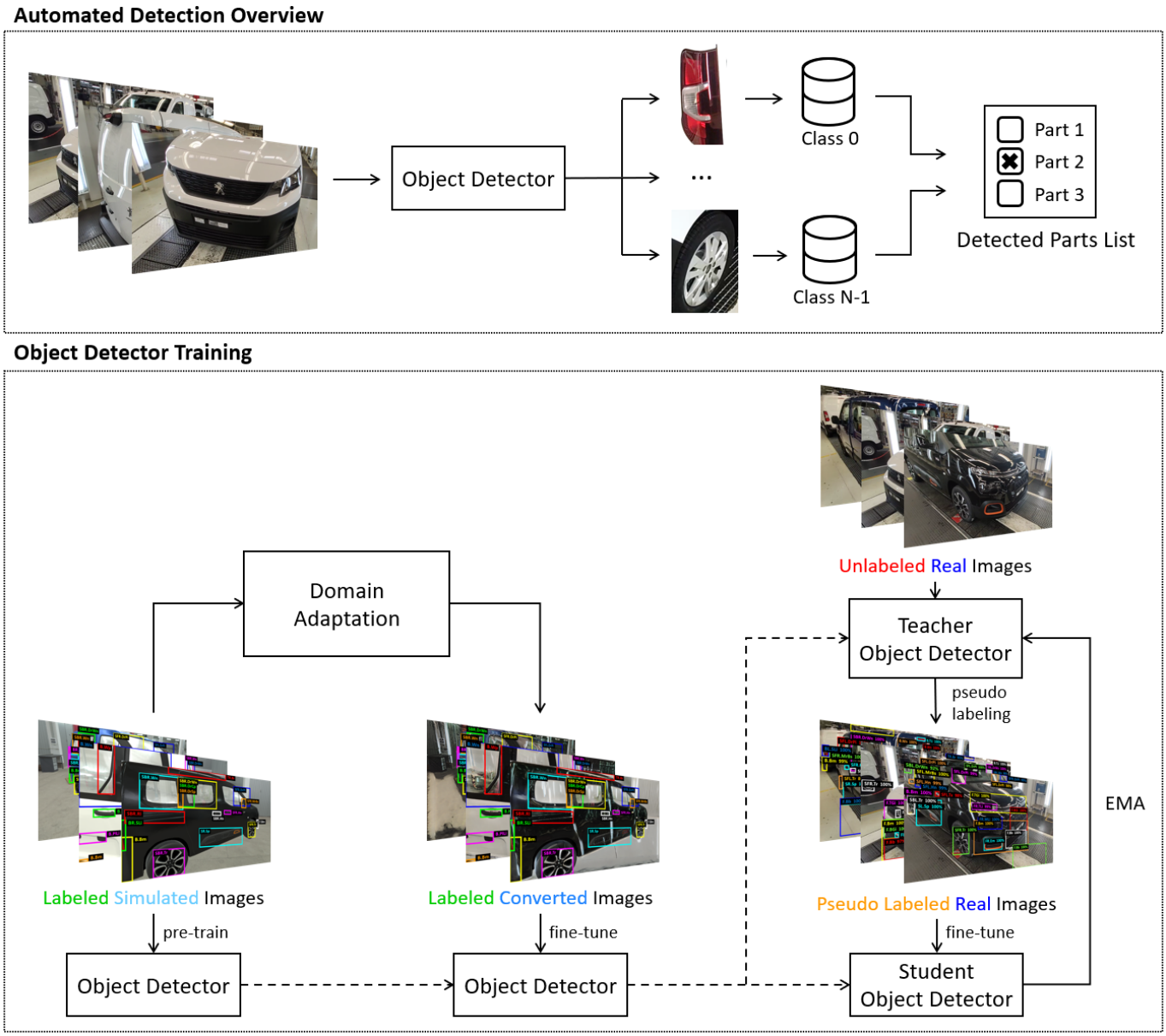

Figure 3. Overall architecture of the hierarchical automated detection module (**top**) and its iterative training process (**bottom**). First, an object detection neural network identifies vehicle parts (e.g., left rear view mirror) and each object is forwarded to the corresponding retrieval sub-module, responsible for outputting the final code for each part (e.g., left rear view mirror of a commercial Peugeot). The training process can be viewed as a semi-supervised object detection pipeline with 3 main stages: pretraining on labeled simulated images which are later converted into more realistic versions, which, in turn, are used to fine-tune the detector. Training is completed by combining both labeled converted images with pseudo labeled real images, in a teacher-student framework with Exponential Moving Average (EMA) updates of the teacher. The dashed arrows represent the weight transfer between the models obtained at each stage.

The hierarchical nature of the system is based on the premise that new vehicle components are far less likely to be introduced than new versions of existing components. As such, this two-stage approach allows increased scalability and versatility since introducing a new version of a given part (e.g., another version of a tire cover) only implies adding (unlabeled) examples to the corresponding CBIR database and avoids retraining the whole object detector from scratch.

3.3.3. Improvements

The baseline was trained solely on simulated images, the only ones that had bounding box annotations. Despite the realness of these simulated images, there is still a domain gap between the simulated and real data distributions, as can be perceived by looking at the second and last columns of Figure 6.

Previous work [30] tackled this issue by training a CycleGAN [7] with an added semantic consistency loss. More recently, the CycleGAN authors proposed Contrastive Unpaired Translation (CUT) [8], which outperforms CycleGAN while being faster and more lightweight, since the conversion is now unidirectional instead of bidirectional. CUT achieves this by employing contrastive learning at the patch level: a generated patch is encouraged to map to a similar point in the learned latent space as its corresponding input patch, while being further apart from other (random) patches. Similarly to what had been done in [30], we also introduced DiffAugment (Differentiable Augmentation) [41] in CUT.

Hyperparameter selection was done empirically taking into account the suggestions offered in the original CycleGAN and CUT papers and, to compare both domain adaptation approaches, both models were trained for 100 epochs with an initial learning rate of 2×10^{-4} and a batch size of 1. The images were resized to 256 $\times$ 256 pixels by center cropping after resizing the longest edge, and DiffAugment involved color, translation and cutout transformations. For the source domain we used the 1960 images from the simulated train set and for the target domain we used 2285 images sampled from videos collected on the real assembly line. For more details on the training hyperparameters of CycleGAN please refer to [30]. With CUT, after preliminary experiments with lower values, we set the PatchNCE loss scaling hyperparameter to 5 to obtain more conservative translations.

After training the Detectron2 on simulated images and fine-tuning on their domain converted versions, inspired by [37,38], we employed pseudo labeling to be able to fine-tune the model on real unlabeled images. We focused on the work of Liu et al. [9], because, besides the advantages stated in Section 3.3.1, the authors made their PyTorch code publicly available (https://github.com/facebookresearch/unbiased-teacher, accessed on 8 January 2022) and it is directly built upon Detectron2. It is worth noting that in this process the labeled instances are still being used together with the pseudo labeled examples. The authors also include a confidence threshold to filter predicted bounding boxes with low confidence and prevent confirmation bias or error accumulation. In our experiments we set this threshold to 0.7 and the unsupervised loss weight to 2.

In summary, the whole object detection training process can be divided into 3 phases, as depicted at the bottom of Figure 3: training on labeled simulated images, fine-tuning on their domain adapted versions, and fine-tuning on the latter together with pseudo labeled real images.

For all phases we resized the images to 640 $\times$ 360 pixels, used a learning rate of 0.01, a batch size of 4 (In the pseudo labeling stage, each batch included 4 labeled simulated images and 4 pseudo labeled real images.), a confidence threshold on the Detectron2's predictions of 0.5 and 0.1 for the Non-Maximum Suppression (NMS) threshold. Similarly to what had been proposed in [30], global NMS was applied in every experiment to reduce the number of bounding boxes in the same location, keeping only the most confident ones. Training was conducted for 58.8k, 29.4k and 20k iterations for each phase, respectively.

All developed code was implemented in Python 3.8, more specifically in PyTorch [42]. We adapted the original code from the Detectron2 (https://github.com/facebookresearch/detectron2, accessed on 23 November 2021), CycleGAN (https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix, accessed 5 December 2021), CUT (https://github.com/taesungp/contrastive-unpaired-translation, accessed on 17 December 2021) and Unbiased-Teacher https://github.com/facebookresearch/unbiased-teacher, accessed on 8 January 2022) repositories. In terms of computational frameworks, CycleGAN and CUT were trained on an NVIDIA GeForce RTX 2080Ti graphics card with 11 GB of memory and an Intel i7-9700K processor, taking approximately 21 and 11 h, respectively. All Detectron2 models were trained on an NVIDIA GeForce RTX 3080 graphics card with 10 GB of memory

and an AMD Ryzen 7 3700X processor. Regarding training times, the three training phases took approximately 7, 3 and a half and 5 h, respectively.

We further improved the performance when inferring on real images by domain adapting the retrieval databases. The rationale is that performing the search of a real object in databases whose images are closer to this real domain will refine the retrieval stage.

3.3.4. Fixed Camera Setup

We also hypothesized that training with images collected by fixed cameras, as opposed to the current generalist scenario that simulates real data collection via a mobile device, would further improve the detection performance in the context of an automotive assembly line. In this fixed camera setup (FCS), besides keeping the image capturing conditions constant, it would be possible to have one object detector per camera and obtain more specialized and accurate models.

In the current solution the detector needs to be able to detect 53 classes. We estimate that in a FCS each detector would have to detect a minimum of 12 and a maximum of 26 classes (This estimate was reached by considering that we have collected 10 samples/views/perspectives from the videos of each test car and by computing the average number of object classes visible in each view.). This would not only increase the overall detection performance, but the detection speed as well, due to the possibility of parallelization. To test the viability of this option, the following steps were followed:

- each image in the real test was labeled with its visible vehicle view(s) and their corresponding objects
- any left/right discrepancies were corrected, by switching the incorrectly detected sides to match the one present in the corresponding view(s) (this was only applied to images with one visible side of the vehicle, not to images where both left and right sides are visible)
- the detected objects that did not belong to the given view(s) were eliminated

Note that these situations would never occur in a FCS, because the corresponding detector would only have been trained on objects specific to that view, meaning it could not detect an object from the right side of the car if trained with images from the left view.

Since the adopted class taxonomy separates front/back and left/right objects, there are no repeated classes (e.g., although there are 4 car rims, they each have their own Detectron2 macro class), which means that multiple detections of the same object stem from having 10 views of the same vehicle and not from more than one object with the same class. Therefore, we can use any of the detections (or combinations of them such as majority voting, for example) to produce the final CBIR code for a given object. Taking this into account, we post process (PP) the final list of predicted vehicle parts by the code with the minimum ratio between the CBIR distance (This corresponds to the Euclidean distance between the query object and its closest training data neighbor of the same macro class.) and its Detectron2's confidence score. This way we are able to choose the detections with the best trade-off between both Detectron2 and CBIR modules.

3.4. Information Visualization

While the automated detection subsystem aims to eliminate human errors caused by inattentiveness or fatigue, by performing a preliminary inspection of the vehicle's parts and providing a list of the non-conforming elements for the workers to validate, the information visualization module aims to convert those results into a more meaningful format to aid the workers. By providing a visual representation of each part that composes the tailored conformity list with the detection results highlighted, the workers can interact with a wearable device through an intuitive interface that streamlines the inspection tasks. The goal of the final quality control support tool is to increase the worker's speed and efficiency, while also reducing fatigue through the removal of the mental strain caused by constant memorization and recall. This requires an interface that is effective and efficient at presenting the information.

In order to determine which of the multiple interface designs best met the workers needs, a preliminary study was performed on the information visualization of the conformity list and highlighting of the non-conforming parts (Figure 4). Each paper prototype was evaluated on their effectiveness at quickly and easily conveying information and ease of interaction, until 5 design choices were selected for full development:

- Part buttons with the visual example and name label;
- Part buttons show/hide according to the side of the vehicle being viewed;
- 3D visual representation of the fully conforming vehicle;
- Buttons' font and 3D parts' color changes to signal non-conformities;
- Text and photo icons on each part button to signal which individual report was attached to the non-conformity.

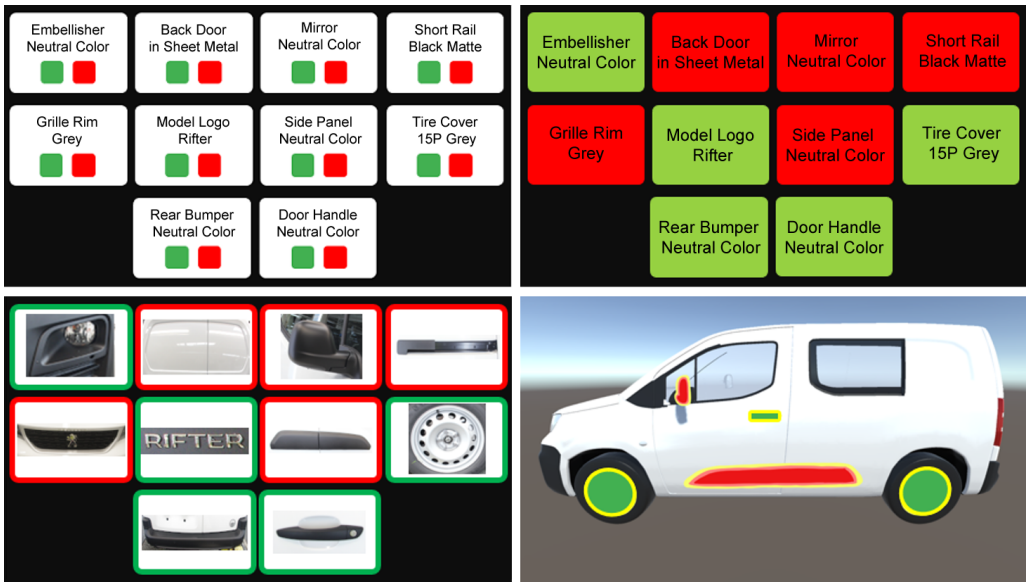

Figure 4. Examples of different studies of the information visualization of the conformity list, with particular focus on the highlighting of the non-conforming parts.

This final interface study (Figure 5) was presented to 2 factory quality control experts, prior to its full implementation on the factory floor, to confirm the internal validity of the information presented (e.g., incorrect part names) and to determine the usability and usefulness of the current design. This was done by having them perform their inspection tasks with the digital replacement of the paper conformity list and ask a Technology Acceptance Model (TAM) on the Task-Technology Fit Model (TIF) questionnaire [2]. The integration with the automated detection results was not performed at this stage, as that information was not relevant for the test.

The test is composed of 2 applications, an assembly line simulator that represents the vehicles to be inspected, and the support tool prototype that simulates the mobile device the workers will perform the inspection with. Both were developed in Unity3D.

The goal of the assembly line simulator is to provide a controlled test environment where all the vehicle configurations and respective non-conformities are known ahead of time. Thus, it avoids dependence on a list of real vehicles, ordered by the clients, that might not contain enough diversity for this test's purposes. The simulator uses the previously developed kSim9 [10,27] to generate the list of 6 conforming kSim9 vehicles (3 Citroën and 3 Peugeot, 3 commercial and 3 private models) and adds the predetermined non-conformities (tire covers, brand and model logos) to some of them. It also generates the vehicle's barcode with the ZXing plugin [43]. This application is meant to be viewed on a PC monitor, so the inspectors can circle the vehicle via the arrow keys or navigation buttons and click on the list of parts to zoom in to inspect in more detail.

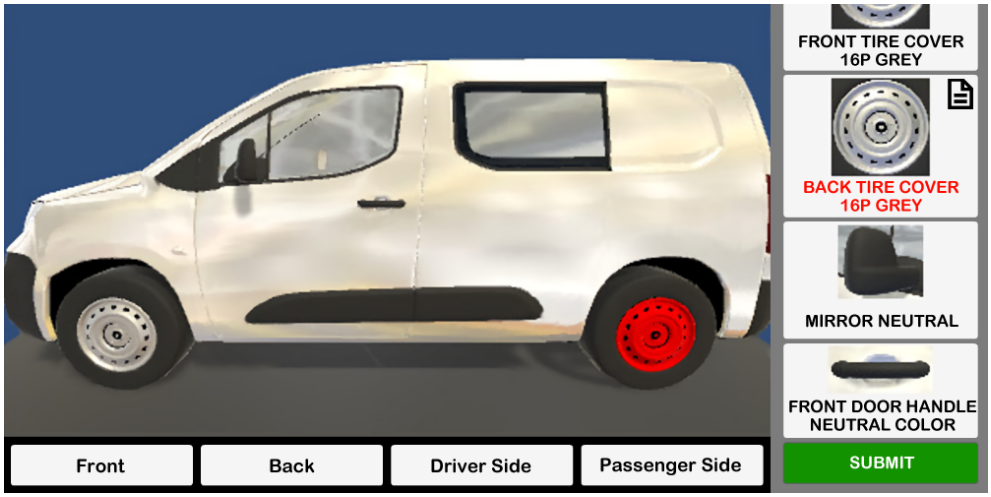

Figure 5. Smartphone design of the conformity list interface of the information visualization prototype. The worker can use the four navigation buttons on the bottom to rotate the 3D representation of the fully conforming vehicle and compare it with its assembled counterpart. The list of scrollable part buttons on the right changes according to which face of the vehicle is being viewed. Any signaled non-conformity changes the color of the respective button and 3D part model to red, and adds a text and/or photo icon to the button as a visual reminder of the type of report the worker performed.

As mentioned, the goal of the interface prototype is to simulate almost all of the support tool's functionalities for the inspectors to test, validate the accuracy of all the information presented and provide important feedback. Its results also inform some of the design decisions on how to best integrate the automated detection in later tests. Once the prototype reads the barcode presented by the simulator with the ZXing plugin [43], it uses the kSim9 to generate both the visual representation of the correctly assembled vehicle and the images for each of the parts buttons. When a button is clicked, the non-conformity is flagged as red by changing both the button's font and the 3D component's color, while a text and/or photo icon shows on the button to signal the type of report submitted in each (Figure 5). An Android smartphone was selected as the stand-in wearable device for its inherent interaction familiarity, quicker prototyping and closer size to wrist mounted devices. Other devices were considered, such as HMD, but several outside factors restricted our choices at the time of this research.

The questionnaire (Table A2 in Appendix A) was performed remotely, due to Covid restrictions, by monitoring each interaction with the software individually and then have each worker rate their subjective impression with a Likert scale from 1 (strongly disagree) to 5 (strongly agree). The questions measured [2]:

- Perceived Useful Scale (6 questions)—how much the individual perceives the use of the software can enhance their work performance;
- Perceived Ease of Use Scale (4 questions)—how much the individual perceives the software to make tasks be easy to perform;
- The Right Data and Right Level of Detail (5 questions)—how well the individual felt the software provided the correct amount of information in the right amount of detail;
- Ease of Software Use (2 questions)—how easy it felt to the individual to learn how to use the software without prior practice;
- Look and Feel (2 questions)—how intuitive and visually appealing the individual found the software to be.

Additionally, 4 open questions were asked regarding the worker's general opinion of the support tool, which features were particularly good, which were bad, and if they considered it a useful tool to their work.

4. Results and Discussion

4.1. Automated Detection

Results from the first 2 training stages can be found in Table 2; these refer to the results on labeled simulated images (before and after domain adaptation) in terms of the mean average precision (mAP) over all intersection over union (IoU) thresholds from 50% until 95% with 5% increments and AP at 50% and 75% IoU. Although the new version of the simulated dataset has around 17 times less examples, the performance of the detector is not severely hindered, having gone from 86.76 [10] to 82.13 mAP, and, more importantly, it improved on the real test dataset, as will be discussed later.

Table 2. Detection results obtained on simulated images, before and after domain adaptation, in terms of mAP and AP at 50% and 75% of IoU. The first column refers to the training process and the second specifies the test data for each scenario; for example, the 3rd row refers to the performance when training on simulated images and testing on CycleGAN converted images, while the 6th row refers to fine-tuning and testing on simulated images converted by CycleGAN. We include the results from [10] for comparison.

Training	Data	Detection		
		mAP	AP@50	AP@75
From scratch	simulated	82.13	93.99	90.38
	CycleGAN	69.23	83.91	78.66
	CUT	42.89	58.43	49.15
Fine-tuning w/CycleGAN	simulated	77.41	90.80	86.71
	CycleGAN	80.53	93.08	89.53
Fine-tuning w/CUT	simulated	75.25	89.45	84.43
	CUT	77.26	92.15	86.76

As expected, the performance of the detector trained on unaltered simulated images decreases when testing with domain converted images (2nd versus 3rd and 4th rows of Table 2) due to the domain gap introduced. When fine-tuning on the adapted data, mAP increases without significantly decreasing on original images; for example, after fine-tuning with CUT converted images, the performance improves from 42.89 to 77.26, while the mAP on the original set only decreases from 82.13 to 75.25. Interestingly, this improvement (and the decrease on original images) is more accentuated with CUT than with CycleGAN. This hints that CUT is able to better approximate the simulated to the real domain, which is also corroborated by the Fréchet Inception Distances (FID) [44] presented in Table 3: after the conversion with CUT, the FID between the resulting distribution and the real domain is lower than with CycleGAN (56.87 vs. 60.04), while being higher for the original domain (58.17 vs. 35.40) (The closer two distributions are, the lower the FID.). These conclusions can also be visually verified in Figure 6, where the progression from the simulated ("Simulated (new)") to the real domain via CycleGAN and CUT is shown, as well as a comparison with the old version of the simulated dataset. In general, both methods tend to darken the original image, make the surface finishes less dull/more metallic, and introduce reflections, but these are more accentuated with CUT (see the 3rd and 6th rows). It is also interesting to note that CUT introduces more texture to the floors (1st and 5th rows), transforms the background pillars into vertical lights (rows 2, 3, 4 and 6), and makes the glass windows more see-through (2nd row).

The results regarding the performance of the system on real unlabeled images, the main focus of this work, can be found in Table 4 and Figure 7. Similarly to previous works [10,30], due to the lack of ground truth bounding boxes for real images, we frame the problem as multi-label multi-class classification, in which for each vehicle we obtain a list of detected parts and compare it to the ground-truth list.

Table 3. Fréchet Inception Distances (FIDs) between the simulated/real domains and the domain adapted versions of the simulated images by CycleGAN and CUT. Lower FIDs mean closer distributions, so CUT achieves better approximation to the real domain.

	FID w/ CycleGAN	FID w/ CUT
Simulated vs. Adapted	35.40	58.17
Real vs. Adapted	60.04	56.87

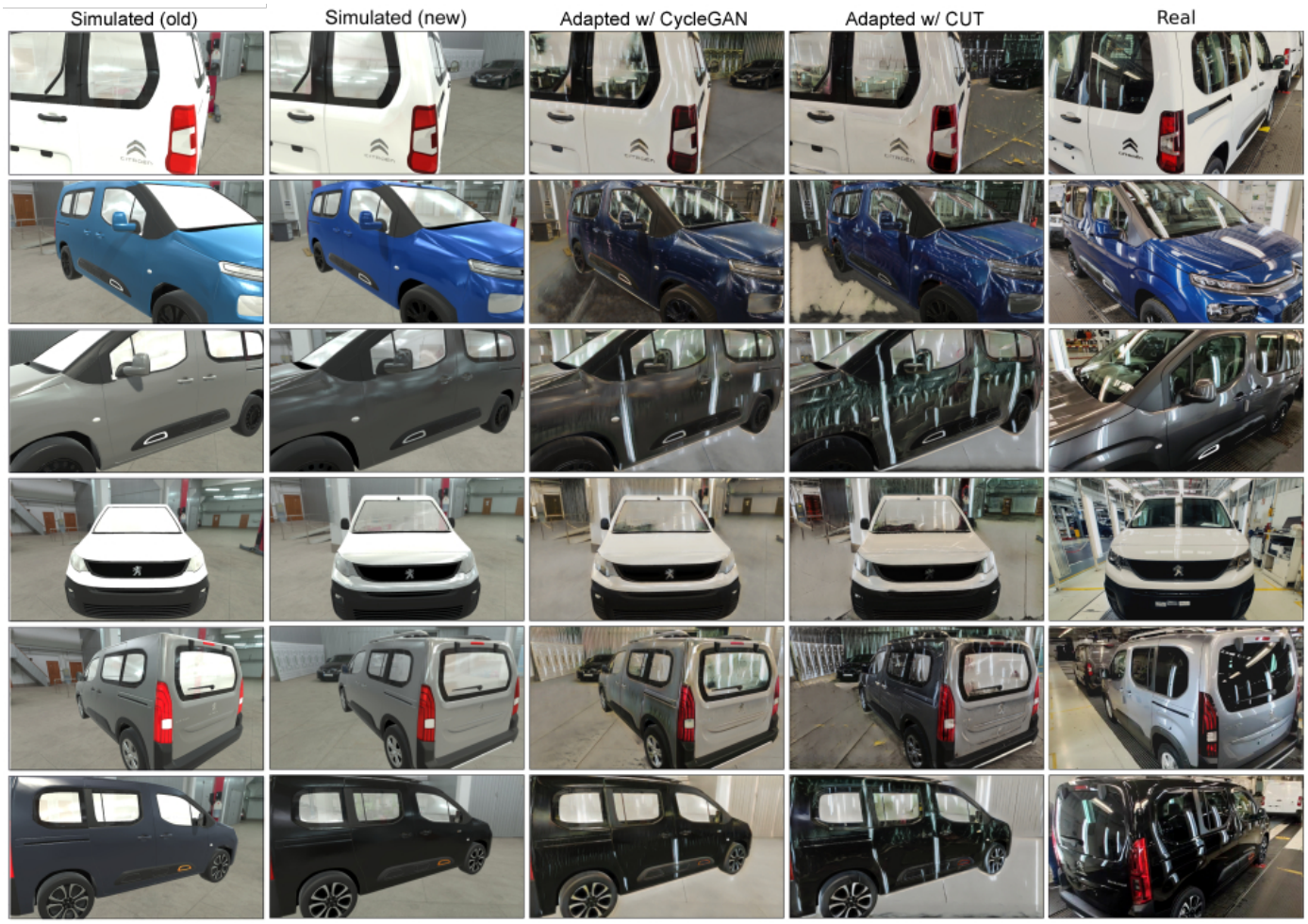

Figure 6. Examples of domain adapted images with CycleGAN and CUT, and their comparison to real images collected on the factory floor. We also include a comparison between the old and new versions of the simulated datasets on the first two columns.

Table 4 contains the label- and example-based metrics of the improvements proposed: the baselines of previous published works (Baseline [10] and CycleGAN [30]), the updated Detectron2 baseline trained on the new simulated dataset, the fine-tuned versions with CycleGAN and CUT converted images, and the remaining improvements upon CUT, including pseudo labeling (CUT + PL), the conversion of the CBIR databases (CUT + PL + CC), the simulation of a fixed camera setup (CUT + PL + CC + FCS), and post processing (CUT + PL + CC + FCS + PP). For label-based precision, recall, F1-score and Matthews Correlation Coefficient (MCC) we report both macro and micro averages. For every metric higher is better, except for the Hamming loss. As stated in Pinho et al. [30], our primary metric of interest is the example-based F1-score because we prefer high detection rates for each vehicle in detriment of high detection rates for each category (i.e., the label-based alternative) and, more importantly, the F1-score not only takes into account both precision and recall, but does not consider true negatives. This is paramount in our context, since one vehicle does not include all possible parts. As such, one model might have higher

accuracies by increasing the number of true negatives at the expense of true positives, which is less reflected in the F1-score. Thus, the last two columns of Table 4 include the F1-score improvements upon the baselines, in percentage and percentage points (pp). Finally, we present the multi-label multi-class metrics for the final 150 CBIR codes and for the 53 Detectron2 macro classes.

Table 4. Multi-label multi-class classification results on the real test set. We present label- and example-based metrics for both stages of the hierarchical process (detection and CBIR). For label-based precision, recall, F1 and MCC we include both macro and micro averages. Our main performance metric is the example-based F1 score, for which we include the improvement relative to the corresponding baselines in percentage and percentage points (pp). For every metric except the Hamming loss, higher is better.

Method	Label-Based									Example-Based						
	Acc.	Prec.		Rec.		F1		MCC		Acc.	Prec.	Rec.	Hamm. (↓)	F1	F1 improv. %	F1 improv. pp
Detectron2 Macro Categories																
Baseline [10]	0.656	0.462	0.647	0.419	0.449	0.394	0.530	**0.198**	0.284	0.357	0.647	0.446	0.344	0.528	-	-
CycleGAN [30]	0.652	0.491	0.594	0.550	0.616	0.455	0.605	0.166	0.294	0.435	0.600	0.617	0.348	0.609	15.34	8.10
New Baseline	0.642	0.800	0.954	0.535	0.581	0.599	0.722	0.093	0.372	0.576	0.958	0.596	0.358	0.735	-	-
CycleGAN	0.692	0.730	**0.965**	0.585	0.640	0.623	0.769	0.107	0.435	0.635	**0.964**	0.653	0.308	0.778	5.850	4.30
CUT	0.753	**0.898**	0.951	0.683	0.729	0.734	0.826	0.134	0.473	0.706	0.950	0.738	0.247	0.830	12.93	9.50
CUT + PL	0.794	0.870	0.914	0.812	0.821	0.808	0.865	0.173	0.450	0.759	0.911	0.824	0.206	0.865	17.69	13.0
CUT + PL + FCS	**0.857**	0.874	0.923	**0.889**	**0.896**	**0.868**	**0.909**	**0.218**	**0.569**	**0.830**	0.919	**0.899**	**0.143**	**0.909**	**23.70**	**17.4**
CBIR Final Codes																
Baseline [10]	0.709	0.295	0.438	0.322	0.359	0.264	0.395	0.154	0.207	0.244	0.426	0.356	0.291	0.388	-	-
CycleGAN [30]	0.671	0.284	0.401	0.428	0.498	0.293	0.444	0.140	0.217	0.289	0.403	0.497	0.329	0.445	14.70	5.70
New Baseline	0.709	0.326	0.484	0.372	0.421	0.305	0.450	0.185	0.255	0.295	0.478	0.435	0.291	0.455	-	-
CycleGAN	0.703	0.305	0.476	0.400	0.468	0.311	0.472	0.176	0.266	0.326	0.479	0.487	0.297	0.483	6.150	2.80
CUT	0.707	0.394	0.484	0.458	0.529	0.371	0.506	0.224	0.298	0.347	0.484	0.545	0.293	0.513	12.75	5.80
CUT + PL	0.692	0.360	0.468	0.536	0.645	0.395	0.543	0.220	0.328	0.374	0.469	0.651	0.308	0.545	19.78	9.00
CUT + PL + CC	0.713	0.403	0.495	0.528	0.635	0.412	0.556	0.264	0.355	0.398	0.503	0.646	0.287	0.565	24.18	11.0
CUT + PL + CC + FCS	0.721	0.424	0.506	**0.565**	**0.680**	**0.445**	0.580	0.291	0.388	0.423	0.513	**0.691**	0.279	0.589	29.45	13.4
CUT + PL + CC + FCS + PP	**0.773**	**0.440**	**0.603**	0.482	0.586	0.425	**0.594**	**0.326**	**0.437**	**0.440**	**0.605**	0.598	**0.227**	**0.601**	**32.09**	**14.6**

The new baseline clearly outperforms its older version, achieving 0.735 (macro) and 0.455 (CBIR) example-based F1-scores compared to the previous 0.528 and 0.388. It even performs better than the previously domain adapted version (CycleGAN [30]). This allows us to conclude that the updated dataset with its improved vehicle colors and textures, although 17 times smaller than the original, is clearly more adequate for our task.

Fine-tuning with CUT converted images leads to a 5.8pp improvement over the new baseline compared to the 2.8pp improvement obtained with CycleGAN, which is consistent with previous observations. Pseudo labeling further boosts performance, resulting in a 9pp improvement, and converting the CBIR databases with CUT improves the F1-score from 0.537 to 0.565, an overall improvement of 11pp.

Detection examples can be found in Figure 7. After fine-tuning with CycleGAN and CUT converted data there is a slight increase in the number of detected instances and a reinforcement of the confidence scores. Pseudo labeling greatly improves the number of detected instances. However, there are still some failure cases, mainly switches between left and right (note that we consider the left side the driver's side) and in particular the image on the 4th row. The detector identifies some objects as belonging to the back of the car, (e.g., door handles identified as plate lights, B.PlLi, on columns 2 and 3). After pseudo labeling they are identified as door handles, but still from the back of the car (B.Hn instead of SFL.Hn). Moreover, there seems to be an excessive amount of detections inside the window. However, these are not completely misplaced: there is a reflection of the rear view mirror (pink SFL.MrBs bounding box) on the left side of the vehicle, the rear view mirror on the right side (blue and orange SFL.MrBs boxes) is visible through the windows, one of the headrests (white SFL.MrBs box) looks similar to the back of a rear view mirror, and the right door pillar (red SFL.DrPi box) is visible through the windows. In fact, side views of the vehicles are currently the most challenging, since these are the most discrepant from

the simulated image (in most simulated images of the side of the car more than 50% of the vehicle is visible, while in the real images the perspective includes only the front or the back halves of the car separately). We wish to correct this in future versions of the simulated data if the collection of real images continues to be done in this mobile scenario. Another alternative is the fixed camera setup described in Section 3.3.4, whose simulation improved the example-based F1-score from 0.565 to 0.589 and 0.601, before and after post processing, respectively. The results can be observed in the last column of Figure 7, where there is a correction of the left/right mismatches (e.g., on the first row the light blue bounding box of the car rim SFR.Tr was switched to SFL.Tr) and a decrease in the number of detected objects, especially on the 4th row. This confirms the viability of the solution, especially considering that we are interested in solving this specific problem for this particular factory environment, as opposed to finding a general approach.

Figure 7. Examples of detection results on real test images with different improvements upon the Baseline, including fine-tuning with CycleGAN and CUT converted images and fine-tuning the latter model with pseudo labeling (PL) of real images. The last column represents the results of the simulation of a fixed camera setup (FCS).

4.2. Information Visualization

While the number of samples collected does not provide enough variation to be statistically significant, the goal of only having 2 experts performing the qualitative evaluation was to validate the design choices by argument from authority, so glaring errors that can negatively impact worker performance, and thus skew the results, could be corrected before proceeding with the test on the factory floor. Only minor text corrections were requested, but the qualitative evaluation of each criteria the questions belong to provided useful insight on the workers' priorities, needs and capabilities that inform future design decisions. One of the questions in the Perceived Ease of Use criteria was excluded due to a semantic misinterpretation resulting in a missing value.

As seen in Table 5, all criteria were rated above 3.50 (1 to 5 Likert scale). Both workers show a great degree of internal consistency, with the reason for Inspector 2 giving a

lower average score being different standards of evaluation, as confirmed by the feedback provided in the open questions. Inspector 1 put particular emphasis on how the support tool would help reduce fatigue, the skill gap between workers and factory costs, while Inspector 2 paid more attention to design flaws and pointed out some improvements (larger screen and button image resolution and removal of the 3D visual representation). Both stated how useful this tool would be to their line of work, particularly at how easy and fast it makes the identification and reporting of non-conformities, and how much the existence of an integrated camera and the notes options facilitate the documentation process.

Table 5. Mean and standard deviations of the quantitatively measurable criteria of the TAM and TIF questionnaire, presented to quality control inspectors for validation of the information visualization module.

Criteria	Inspector 1	Inspector 2
Perceived Usefulness	4.83 ± 0.37	4.50 ± 0.50
Perceived Ease of Use	4.50 ± 0.00	4.00 ± 0.00
Right Data and Level of Detail	5.00 ± 0.00	4.00 ± 0.00
Ease of Software Use	5.00 ± 0.00	4.50 ± 0.50
Look and Feel	5.00 ± 0.00	3.50 ± 0.50

A set of clarification questions were made outside of the questionnaire, which provided additional insight into the answers provided:

- The 3D vehicle representation was considered unnecessary because of their level of expertise no longer requiring a visual aid to know where each part should be located in the vehicle. The label description suffices;
- In the paper version the reporting of a non-conformity requires a lengthy written description of the occurrence, which the support tool partially eliminates just from having each button associated with the respective part;
- In the paper version the photographic and additional written details of the non-conformity are optional, due to the additional hassle of uploading the information in the physical terminal under tight time constraints.

5. Conclusions and Future Research

We proposed replacing the traditional paper-based conformity list, a limited approach still frequently used in quality control in the highly automated automotive industry, with a digital version. With this we tackle two major issues, mental and physical fatigue, since the worker no longer needs to have a detailed mental image of all the items on the lists, nor needs to physically access a terminal at the end of the assembly line to submit the report, which also represents an increased inspection time.

Our fully hybrid quality control system aims to integrate the automated detection of the vehicle parts with an information visualization interface to present the digital conformity list to the worker, allowing them to verify the non-conforming parts and wirelessly submit the final report. One of the core building blocks of both modules is a 3D vehicle generator that is used to generate the vehicle parts for the digital conformity list and to train the automated detection system. The use of simulated data allows us not only to have unlimited amounts of data with sufficient variability regarding environment conditions, but also to automatically annotate the images while avoiding the time and resource consuming process of data collection and manual labeling.

We show that, through incremental improvements upon a hierarchical object detection baseline trained solely using simulated data, which included domain adaptation of simulated images and pseudo labeling of real images, we are able to improve by 11pp the baseline and achieve a final 0.565 example-based F1-score on real images in an unconstrained scenario, without any recourse to annotated real data. Moreover, we simulated a fixed camera setup in which data collection would be performed in a very controlled environment and there would be one object detector for each view of the vehicle, and validated

the viability of this solution, having improved the baseline by 14.6pp (corresponding to 0.601 F1-score).

Furthermore, the internal validation questionnaire conducted with the factory's quality control experts confirmed the extreme usefulness of this solution, especially when compared to the paper one. They found the support tool to be very fast and easy to use, its design simple and intuitive enough for them to figure out how to navigate and report the non-conformities without any prior training. Testing the proposed system on the factory floor, rather than in a simulated environment, would be necessary to confirm how much this solution would improve the worker's inspection performance. However, the experts showed absolute certainty that the support tool would greatly ease the documentation and submission of the reporting process and prove to be a major asset in aiding their quality control inspection tasks.

The lack of bounding box annotations for the real data, coupled with the fact that this work constitutes a preliminary validation before testing on the factory floor, are the main, current limiting factors of the proposed solution. As such, the immediate next step is to obtain the aforementioned fine grained annotations and to perform extensive testing on the factory floor and compare the mobile and fixed camera scenarios. Following the validation of our hypothesis that a more constrained *ad hoc* solution specific to the final deployment scenario is preferable to a generalist approach, it would be interesting to further improve the system by employing prior knowledge ensuring that vehicle parts that do not physically fit a certain chassis would not be detected. This is only possible because, in the proposed Industry 4.0 scenario with a digital conformity list, the system knows, at any given moment, which vehicle version is being assembled and, thus, which parts would (not) fit its chassis.

Depending on the results obtained on the factory floor, we plan on replacing the kSim9 vehicle generator with official Computer Aided Design (CAD) part models, for higher fidelity and better automated detection results. We also intend to expand the part library to include a wider variety of vehicles beyond the k9 van model and adapt the visualization module into a wider variety of mobile devices, that best suit different workers' needs. Other next steps include closing the loop by integrating the workers' validation of the detected conformities and non-conformities to correct the automated detection module in an active learning setup. We also plan on further integrating the worker's expertise by adding to the visualization module the pseudo labeled real images for each button for individual evaluation, rather than only presenting the simulated version of the parts. This way the worker can give the object detector more fine-grained feedback regarding the detection itself and not only about the classification. Thus, this would constitute a truly hybrid assembly line, where the machine helps the worker and vice-versa, and where the overall accuracy and efficiency of the quality control process is improved because it leverages the best of both worlds in true symbiosis.

Author Contributions: Conceptualization, A.T.C., I.R.-T., V.F. and L.F.T.; methodology, A.T.C., I.R.-T., P.P., V.F. and L.F.T.; software, A.T.C., I.R.-T. and P.P.; validation, A.T.C., I.R.-T., P.P., V.F. and L.F.T.; formal analysis, A.T.C. and I.R.-T.; investigation, A.T.C., I.R.-T. and P.P.; resources, V.F. and L.F.T.; data curation, A.T.C. and I.R.-T.; writing—original draft preparation, A.T.C. and I.R.-T.; writing—review and editing, V.F. and L.F.T.; visualization, A.T.C.; supervision, V.F. and L.F.T.; project administration, V.F. and L.F.T.; funding acquisition, V.F. and L.F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Project "INDTECH 4.0—New technologies for smart manufacturing", No. POCI-01-0247-FEDER-026653, financed by the European Regional Development Fund (ERDF), through the COMPETE 2020—Competitiveness and Internationalization Operational Program (POCI).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Ethics Committee of Universidade de Trás-os-Montes e Alto Douro (Ref. Doc16-CE-UTAD-2022, 30 March 2022).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from PSA Mangualde and are available from the authors with the permission of PSA Mangualde.

Acknowledgments: We would like to acknowledge the preliminary work carried out by António Pedro Pereira, without which this project would not have been able to take off. We would also like to thank Bridget M. Bradley for her very useful insights in graphic design and information visualization.

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.

Abbreviations

The following abbreviations are used in this manuscript:

AR	Augmented Reality
CAD	Computer Aided Design
CBIR	Content-Based Image Retrieval
CC	Converted CBIR
CNN	Convolutional Neural Network
CUT	Contrastive Unpaired Translation
DCNN	Deep Convolutional Neural Network
DiffAugment	Differentiable Augmentation
EMA	Exponential Moving Average
FCS	Fixed Camera Setup
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
HMD	Head-Mounted Display
IoU	Intersection over Union
k9	Name of the multi-brand van model
kSim9	Name of the 3D k9 Vehicle Generator
mAP	mean Average Precision
MCC	Matthews Correlation Coefficient
MLP	Multi Layer Perceptron
NMS	Non-Maximum Suppression
PL	Pseudo Labeling
PP	Post Processing
PSA	Peugeot Société Anonyme
R-CNN	Region Based Convolutional Neural Network
SGDM	Stochastic Gradient Descent with Momentum
SURF	Speeded Up Robust Features
TAM	Technology Acceptance Model (TAM)
TIF	Task-Technology Fit Model
YOLOv3	3rd version of the You Only Look Once algorithm

Appendix A

Table A1. Description of every vehicle part code, organized by location. The code is built by combining the location of the object followed by its initials, for example B.Bb is the bumper bar located at the back of the vehicle. The following mapping is used: B—back, F—front, S—side, T—top, M—middle, L—left and R—right. Combinations of the latter can also occur, for example FL—front left or SBR—side back right.

Location	Code	Object Description
Back	B.Bb	Bumper bar
	B.Bm	Bumper
	B.Hn	Door handle
	B.Lg	Brand logo
	B.PlLi	Plate lights
	B.TLi	Top light
	B.Tx	Model logo/text
	B.Wn	Window
	B.WnHg	Window Hanger
	B.Wp	Wiper
	BL.SLi/BR.SLi	Side light
Front	F.Bb	Bumper bar
	F.BGi	Bottom grille
	F.Bm	Bumper
	F.TGi	Top grille
	FL.Em/FR.Em	Fog light embellisher
	FL.FLi/FR.FLi	Fog light
	FL.MLi/FR.MLi	Middle headlight
	FL.TLi/FR.TLi	Top headlight
	FT.GiLg	Grille logo
Back Side	SBL.DrCp/SBR.DrCp	Door curved panel
	SBL.DrSb/SBR.DrSb	Door side bar
	SBL.DrWn/SBR.DrWn	Door window
	SBL.Hn/SBR.Hn	Door handle
	SBL.Rl/SBR.Rl	Rail
	SBL.Tr/SBR.Tr	Tire rim
	SBL.Wn/SBR.Wn	Window
Front Side	SFL.DrPi/SFR.DrPi	Door pillar
	SFL.Hn/SFR.Hn	Door handle
	SFL.MrBs/SFR.MrBs	Rear view mirror
	SFL.Tr/SFR.Tr	Tire rim
	SFL.Tx/SFR.Tx	XTR logo
Side	SL.Sp/SR.Sp	Side panel
Top	TL.Br/TR.Br	Roof bar

Table A2. List of the TAM and TIF questionnaire questions presented to the the factory's quality control inspectors to measure the support tool's usefulness and how well it meets their needs.

Criteria	Questions
Perceived Usefulness	The tool helps me do the inspections faster.
	The tool helps me perform my work in a more systematic way.
	The tool allows me to report the non-conformities more quickly.
	The tool makes the inspection work easier.
	The tool helps me to identify the non-conformities more easily.
	I find the tool useful for the tasks performed during the inspection.

Table A2. *Cont.*

Criteria	Questions
Perceived Ease of Use	The use of the tool is easy to understand. The use of the tool is flexible. It's easy to perform the inspection tasks using the tool. The tool is easy to use.
Right Data and Level of Detail	The information presented by the tool is the one I need to perform the inspections. It's more difficult for me to perform the inspections using the tool because the necessary information is lacking. The tool provides enough textual and visual information. In the case of non-conformities, the tool allows the recording of the necessary details. The tool allows me to read the information and to visualize the parts simply and quickly.
Ease of Software Use	It's easy to learn how to use the tool. The inspection support tool is adequate and easy.
Look and Feel	Using the tool is intuitive. I like the look of the tool.
Open Questions	What's your general impression of the tool? Are there any positive aspects that you want to highlight about the tool? Are there any negative aspects that you want to highlight about the tool? Would you like to use a tool like this to support you in the inspection tasks you perform?

References

1. Gewohn, M.; Beyerer, J.; Usländer, T.; Sutschet, G. Smart Information Visualization for First-Time Quality within the Automobile Production Assembly Line. *IFAC-PapersOnLine* **2018**, *51*, 423–428. [CrossRef]
2. Kluge, A.; Termer, A. Human-centered design (HCD) of a fault-finding application for mobile devices and its impact on the reduction of time in fault diagnosis in the manufacturing industry. *Appl. Ergon.* **2017**, *59*, 170–181. [CrossRef] [PubMed]
3. Pfeiffer, S. Robots, Industry 4.0 and humans, or why assembly work is more than routine work. *Societies* **2016**, *6*, 16. [CrossRef]
4. Piero, N.; Schmitt, M. Virtual commissioning of camera-based quality assurance systems for mixed model assembly lines. *Procedia Manuf.* **2017**, *11*, 914–921. [CrossRef]
5. Gewohn, M.; Beyerer, J.; Usländer, T.; Sutschet, G. A quality visualization model for the evaluation and control of quality in vehicle assembly. In Proceedings of the 2018 7th International Conference on Industrial Technology and Management (ICITM), Oxford, UK, 7–9 March 2018; pp. 1–10. [CrossRef]
6. Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [CrossRef]
7. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
8. Park, T.; Efros, A.A.; Zhang, R.; Zhu, J.Y. Contrastive Learning for Unpaired Image-to-Image Translation. In Proceedings of the European Conference on Computer Vision, Virtual, 23–28 August 2020.
9. Liu, Y.C.; Ma, C.Y.; He, Z.; Kuo, C.W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; Vajda, P. Unbiased Teacher for Semi-Supervised Object Detection. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
10. Rio-Torto, I.; Campaniço, A.T.; Pereira, A.; Teixeira, L.F.; Filipe, V. Automatic quality inspection in the automotive industry: a hierarchical approach using simulated data. In Proceedings of the 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), Virtual, 23–26 April 2021; pp. 342–347. [CrossRef]
11. Luckow, A.; Cook, M.; Ashcraft, N.; Weill, E.; Djerekarov, E.; Vorster, B. Deep learning in the automotive industry: Applications and tools. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 3759–3768. [CrossRef]
12. Chauhan, V.; Surgenor, B. Fault detection and classification in automated assembly machines using machine vision. *Int. J. Adv. Manuf. Technol.* **2017**, *90*, 2491–2512. [CrossRef]
13. Sun, S.; Huang, J.; Zhu, J.; Yu, Y.; Zheng, L. Research on Both the Classification and Quality Control Methods of the Car Seat Backrest Based on Machine Vision. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 3106313. [CrossRef]
14. Kuric, I.; Klarák, J.; Bulej, V.; Sága, M.; Kandera, M.; Hajdučík, A.; Tucki, K. Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence. *Appl. Sci.* **2022**, *12*, 864. [CrossRef]
15. Pei, Z.; Chen, L. Welding component identification and solder joint inspection of automobile door panel based on machine vision. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 6558–6563.
16. Zhou, Q.; Chen, R.; Huang, B.; Liu, C.; Yu, J.; Yu, X. An automatic surface defect inspection system for automobiles using machine vision methods. *Sensors* **2019**, *19*, 644. [CrossRef]
17. Chang, F.; Liu, M.; Dong, M.; Duan, Y. A mobile vision inspection system for tiny defect detection on smooth car-body surfaces based on deep ensemble learning. *Meas. Sci. Technol.* **2019**, *30*, 125905. [CrossRef]

18. Sun, X.; Gu, J.; Huang, R.; Zou, R.; Giron Palomares, B. Surface defects recognition of wheel hub based on improved faster R-CNN. *Electronics* **2019**, *8*, 481. [CrossRef]
19. Halim, A.A. Applications of augmented reality for inspection and maintenance process in automotive industry. *J. Fundam. Appl. Sci.* **2018**, *10*, 412–421.
20. Chouchene, A.; Ventura Carvalho, A.; Charrua-Santos, F.; Barhoumi, W. Augmented Reality-Based Framework Supporting Visual Inspection for Automotive Industry. *Appl. Syst. Innov.* **2022**, *5*, 48. [CrossRef]
21. Lima, J.P.; Roberto, R.; Sim oes, F.; Almeida, M.; Figueiredo, L.; Teixeira, J.M.; Teichrieb, V. Markerless tracking system for augmented reality in the automotive industry. *Expert Syst. Appl.* **2017**, *82*, 100–114. [CrossRef]
22. Ford. 2016. Available online: https://media.ford.com/content/fordmedia/fna/us/en/news/2016/02/10/innovative-smartphone-app-saves-ford-factory-workers.html (accessed on 14 December 2021).
23. Rega, A.; Di Marino, C.; Pasquariello, A.; Vitolo, F.; Patalano, S.; Zanella, A.; Lanzotti, A. Collaborative Workplace Design: A Knowledge-Based Approach to Promote Human–Robot Collaboration and Multi-Objective Layout Optimization. *Appl. Sci.* **2021**, *11*, 2147. [CrossRef]
24. Borisov, N.; Weyers, B.; Kluge, A. Designing a human machine interface for quality assurance in car manufacturing: An attempt to address the "functionality versus user experience contradiction" in professional production environments. *Adv. Hum.-Comput. Interact.* **2018**, *2018*, 9502692. [CrossRef]
25. Khamaisi, R.K.; Prati, E.; Peruzzini, M.; Raffaeli, R.; Pellicciari, M. UX in AR-Supported Industrial Human–Robot Collaborative Tasks: A Systematic Review. *Appl. Sci.* **2021**, *11*, 448. [CrossRef]
26. Capela, S.; Silva, R.; Khanal, S.R.; Campaniço, A.T.; Barroso, J.; Filipe, V. Engine Labels Detection for Vehicle Quality Verification in the Assembly Line: A Machine Vision Approach. In *CONTROLO 2020. Lecture Notes in Electrical Engineering*; Gonçalves, J.A., Braz-César, M., Coelho, J.P., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 695, pp. 740–751. [CrossRef]
27. Khanal, S.R.; Amorim, E.V.; Filipe, V. Classification of Car Parts Using Deep Neural Network. In *CONTROLO 2020. Lecture Notes in Electrical Engineering*; Gonçalves, J.A., Braz-César, M., Coelho, J.P., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 695, pp. 582–591. [CrossRef]
28. Mete, S.; Çil, Z.A.; Özceylan, E.; Ağpak, K.; Battaïa, O. An optimisation support for the design of hybrid production lines including assembly and disassembly tasks. *Int. J. Prod. Res.* **2018**, *56*, 7375–7389. [CrossRef]
29. Campaniço, A.T.; Khanal, S.; Paredes, H.; Filipe, V. Worker Support and Training Tools to Aid in Vehicle Quality Inspection for the Automotive Industry. In *Technology and Innovation in Learning, Teaching and Education. TECH-EDU 2020*; Reis, A., Barroso, J., Lopes, J.B., Mikropoulos, T., Fan, C.W., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 432–441. [CrossRef]
30. Pinho, P.; Rio-Torto, I.; Teixeira, L.F. Improving Automatic Quality Inspection in the Automotive Industry by Combining Simulated and Real Data. Advances in Visual Computing. In *ISVC 2021. Lecture Notes in Computer Science*; Bebis, G., Athitsos, V., Yan, T., Lau, M., Li, F., Shi, C., Yuan, X., Mousas, C., Bruder, G., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 278–290. [CrossRef]
31. Nikolenko, S.I. Introduction: The Data Problem. In *Synthetic Data for Deep Learning*; Springer International Publishing: Cham, Switzerland, 2021; pp. 1–17. [CrossRef]
32. Oza, P.; Sindagi, V.A.; VS, V.; Patel, V.M. Unsupervised domain adaptation of object detectors: A survey. *arXiv* **2021**, arXiv:2105.13502.
33. Zhang, D.; Li, J.; Xiong, L.; Lin, L.; Ye, M.; Yang, S. Cycle-Consistent Domain Adaptive Faster RCNN. *IEEE Access* **2019**, *7*, 123903–123911. [CrossRef]
34. Hsu, H.K.; Yao, C.H.; Tsai, Y.H.; Hung, W.C.; Tseng, H.Y.; Singh, M.; Yang, M.H. Progressive Domain Adaptation for Object Detection. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 738–746. [CrossRef]
35. MacKay, C.T.; Moh, T.S. Learning for Free: Object Detectors Trained on Synthetic Data. In Proceedings of the 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, 4–6 January 2021; pp. 1–8. [CrossRef]
36. RoyChowdhury, A.; Chakrabarty, P.; Singh, A.; Jin, S.; Jiang, H.; Cao, L.; Learned-Miller, E. Automatic adaptation of object detectors to new domains using self-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 780–790.
37. Inoue, N.; Furuta, R.; Yamasaki, T.; Aizawa, K. Cross-domain weakly-supervised object detection through progressive domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5001–5009.
38. Deng, J.; Li, W.; Chen, Y.; Duan, L. Unbiased mean teacher for cross-domain object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4091–4101.
39. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 23 November 2021).
40. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv* **2016**, arXiv:1611.05431.
41. Zhao, S.; Liu, Z.; Lin, J.; Zhu, J.Y.; Han, S. Differentiable Augmentation for Data-Efficient GAN Training. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020.

42. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
43. ZXing. 2021. Available online: https://github.com/zxing/zxing (accessed on 15 November 2021).
44. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; NIPS'17; pp. 6629–6640.