Journal of *Imaging*

MDPI

Article

Combining Synthetic Images and Deep Active Learning: Data-Efficient Training of an Industrial Object Detection Model

Leon Eversberg * and Jens Lambrecht

Industry Grade Networks and Clouds, Faculty IV Electrical Engineering and Computer Science, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany; lambrecht@tu-berlin.de
* Correspondence: leon.eversberg@tu-berlin.de

Abstract: Generating synthetic data is a promising solution to the challenge of limited training data for industrial deep learning applications. However, training on synthetic data and testing on real-world data creates a sim-to-real domain gap. Research has shown that the combination of synthetic and real images leads to better results than those that are generated using only one source of data. In this work, the generation of synthetic training images via physics-based rendering is combined with deep active learning for an industrial object detection task to iteratively improve model performance over time. Our experimental results show that synthetic images improve model performance, especially at the beginning of the model's life cycle with limited training data. Furthermore, our implemented hybrid query strategy selects diverse and informative new training images in each active learning cycle, which outperforms random sampling. In conclusion, this work presents a workflow to train and iteratively improve object detection models with a small number of real-world images, leading to data-efficient and cost-effective computer vision models.

Keywords: active learning; computer vision; data efficiency; deep active learning; deep learning; image synthesis; industrial application; object detection; synthetic images; turbine blade

Citation: Eversberg, L.; Lambrecht, J. Combining Synthetic Images and Deep Active Learning: Data-Efficient Training of an Industrial Object Detection Model. *J. Imaging* **2024**, *10*, 16. https://doi.org/10.3390/jimaging10010016

Academic Editor: Guanghui Wang

Received: 28 November 2023
Revised: 29 December 2023
Accepted: 4 January 2024
Published: 6 January 2024

1. Introduction

Deep learning has become a key technology for solving real-world industrial problems using artificial intelligence. However, deep learning models often require large-scale datasets to achieve adequate performance. Limited data remains one of the major challenges for industrial applications of deep learning [1]. As a solution for computer vision tasks, synthetic images can be generated and used as training data. Generating synthetic images has many advantages compared to collecting and manually annotating real-world images. Synthetic images are fast and cheap to generate. They can be used to balance out real-world dataset biases [2]. Furthermore, they can be used in situations where there are privacy concerns surrounding the usage of real-world images [3]. Additionally, they have pixel-perfect annotations without the potential for human error [4].

However, using synthetic images to train computer vision models and then testing them on real-world images creates a domain gap that continues to be a challenge in this field of research [5]. Research has shown that the combination of synthetic and real images outperforms the use of a single data source [6–11]. But how can real-world training images be efficiently selected for combination with the generated synthetic images? In this work, we propose to solve this problem with strategies from the field of active learning (AL). AL uses the current machine learning model to efficiently select data for the next iteration of training.

This paper builds upon our previous work to generate training images via physics-based rendering for industrial object detection (OD) tasks [11] and makes the following new contributions:

- A workflow is presented to efficiently train industrial object detection models by automatically generating synthetic training images based on 3D models and then using deep active learning to iteratively improve the model with reduced annotation cost.
- Different deep active learning query strategies are investigated on a collected industrial dataset for a real-world object detection use case.
- Multiple deep active learning cycles are compared to a single cycle with an equivalent amount of manually labeled training images.

The remainder of this paper is structured as follows: Section 2 provides a summary of prior work on synthetic images and deep active learning for object detection tasks. In Section 3, the methodology of this paper is presented. Our results for synthetic versus real images and different deep active learning (DAL) query strategies are presented in Section 4. Lastly, Section 5 outlines the limitations of our study and summarizes our primary findings.

2. Related Works

2.1. Using Synthetic Images to Train Computer Vision Models

Generating synthetic training data is a promising solution to the data-hungry nature of modern deep learning models. However, training models on a source domain of synthetic images and testing them on a target domain of real images leads to a domain gap, which remains one of the biggest challenges in this field [12]. In order to overcome the domain gap, different approaches have been used. A simple strategy is to copy objects from real images and then paste them onto random background images to create new images [13,14]. For industrial applications, available 3D models can be used to train object detection models [15]. Domain randomization is an approach where training images are randomized to such an extent that the trained model is supposed to see real images as just another variation of the synthetic training data [5,16,17]. The concept of photorealism is another approach, where the goal is to create highly realistic images using physics-based rendering [8,18,19]. Physics-based rendering uses the ray-tracing algorithm to follow the path of light rays through the virtual scene as they bounce off objects in the scene [20]. Domain adaptation is a third approach to bridging the domain gap. This technique attempts to make the source domain and the target domain as similar as possible through image transformations. Synthetic images can be transformed closer to the target domain using generative adversarial networks [21–23]. Alternatively, image filters can be used to transform both source and target images to an intermediate domain [24,25].

2.2. Deep Active Learning

AL is a subfield of machine learning that attempts to maximize the performance of a machine learning model with the least amount of annotated data. The key idea behind AL is that the model selects the data from which it learns [26]. In traditional AL, most algorithms query only one sample at a time, which is inefficient for modern deep learning. Therefore, DAL uses a batch-based query strategy to select the k most useful samples from a large unlabeled pool of data U for annotation to reduce labeling cost while maintaining performance [27]. To select optimal query samples, unlabeled data are fed into the model to generate features. Given these features, a query strategy attempts to find an optimal batch of samples. The selected k samples are annotated by the oracle, e.g., a human annotator, and are then added to the labeled training set L. Given the updated labeled training set, a new model can be trained. This DAL cycle is depicted in Figure 1. The first iteration of the DAL cycle requires an initial model to be trained on the initial labeled training set L_0.

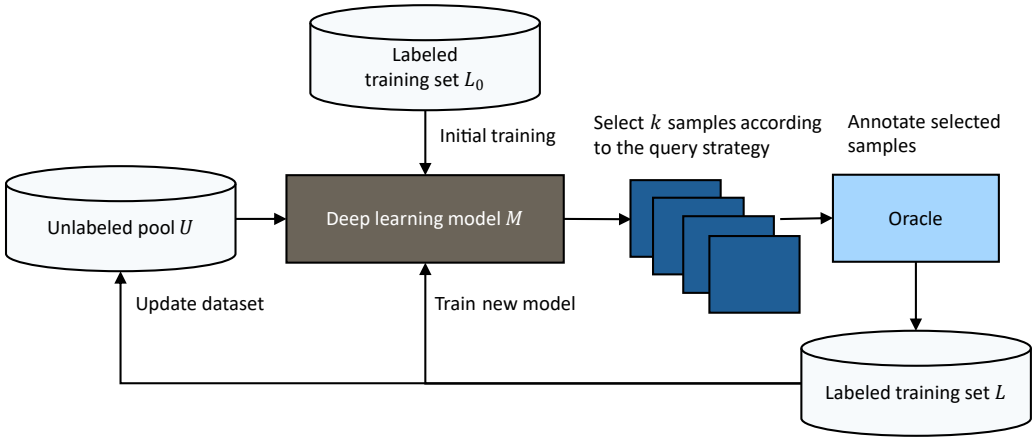

Figure 1. Deep active learning cycle. The large unlabeled pool U is used as input for the current deep learning model. Based on the extracted features, a query strategy selects a batch of k optimal samples for annotation, which can then be used in the next training iteration. Figure based on [27].

Query strategies can be classified into the following three categories: uncertainty-based query strategies, diversity-based query strategies, and hybrid strategies that combine uncertainty and diversity [28]. Uncertainty-based query strategies, such as least confidence, margin sampling, and entropy, select samples that are difficult to predict by the current model [29]. Diversity-based strategies select batches of unlabeled data samples that are representative of the unlabeled pool. This includes clustering algorithms such as the well-known KMeans algorithm [28] and selecting data samples from a small core set that tries to represent the full dataset distribution [30]. Lastly, hybrid strategies attempt to select samples that balance diversity and uncertainty. Example algorithms include BADGE [31], Exploitation–Exploration [32], and DBAL [33]. Zhan et al. [28] implemented 17 different query strategies for DAL and compared them across 7 datasets for image classification. They found unsatisfactory results for diversity-based strategies compared to uncertainty-based strategies and hybrid strategies. Based on their evaluation, they recommend trying uncertainty-based query strategies first for new tasks.

2.3. Deep Active Learning for Object Detection

While AL is traditionally used for classification tasks, the DAL cycle can also be used on OD tasks to reduce annotation costs. Because OD models can produce multiple detections per image, an aggregation method has to be used in order to compute a single score per image as input to the query strategy [34]. Brust et al. [35] trained a YOLO OD model [36] on the PASCAL VOC 2012 dataset [37] with DAL using margin sampling as an uncertainty-based query strategy. In their experimental evaluation, they compared the aggregation methods sum, maximum, and average to aggregate the uncertainty scores from multiple bounding box detections. They concluded that, overall, the sum was the best aggregation method for their data. Haussmann et al. [38] also compared different query strategies on a large-scale OD dataset including cars, pedestrians, bicycles, traffic signs, and traffic lights. As a model, they used a one-stage object detector based on a UNet [39]. They found that uncertainty-based query strategies and diversity-based strategies both performed better than random sampling. Furthermore, they found that letting the query strategy choose from a combined dataset consisting of the unlabeled pool U and the labeled set L outperforms U alone while reducing labeling costs.

As described in Section 2.2, before running the first DAL iteration, an initial model has to be trained. Usually, the initial model is trained by randomly selecting a first batch of samples as L_0 [28,35,38]. However, randomly sampling a small training set can lead to low initial model performance. Furthermore, randomly sampling a large initial training set increases the annotation cost, which is contrary to the goal of DAL. Therefore, in this work,

we propose to train the initial model using synthetically generated images that include automatically generated annotations.

2.4. Combining Deep Active Learning with Synthetic Images

Peng et al. [40] combined synthetic images with DAL in surgical instrument segmentation. For each DAL cycle, they query the most informative training images according to the uncertainty-based query strategy Bayesian active learning by disagreement (BALD) [41] and then manually label them. Next, they generate additional synthetic images via copy-and-paste based on the selected images. The authors conclude that combining synthetic images with deep active learning for image segmentation results in improved performance, especially with limited labeled data. Similarly, query strategies are used in [42,43] to select a limited amount of relevant synthetic images to improve the available real training dataset. Wang et al. [44] combined AL and synthetic images for weakly-supervised OD. They generated synthetic training images via copy-and-paste from a few manually annotated images to train an initial base model. The synthetic images are used in the initial iteration, and weakly labeled images are used in subsequent iterations to train a teacher–student OD model.

Our proposed method uses available industrial 3D models to automatically generate training images via physics-based rendering for an initial OD model. During deployment, large amounts of unlabeled images can be collected. Given an unlabeled pool of images, DAL is used to efficiently fine-tune the next model iteration on a small number of manually labeled images.

3. Materials and Methods

The overall methodology of our approach is summarized in Figure 2. First, a synthetic training dataset L_0^S is automatically generated according to Section 3.1, based on a given 3D model. With these synthetic images, an initial model M_0^S is trained which can then be used for the first DAL cycle with a collected pool of unlabeled real images U (Section 3.2). The model chooses k real training images according to the DAL query strategy from Section 3.4. These images are labeled and added to the labeled training set L. Given the previous model and the selected training images, a new model is fine-tuned according to Section 3.3 and the DAL cycle can be repeated in the next iteration t.

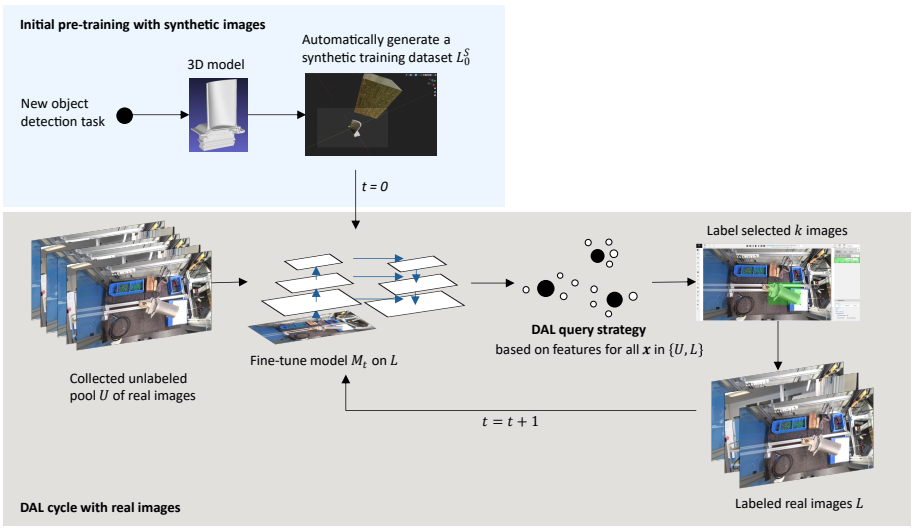

Figure 2. Proposed workflow to train and improve a data-efficient OD model throughout its life cycle.

3.1. Generating a Synthetic Training Dataset

The open-source 3D creation software Blender is a popular tool amongst many researchers to generate synthetic training images for computer vision tasks, e.g., [19,45–47].

Blender utilizes a path tracing rendering engine called Cycles for producing physically-based renders and can be automated using its Python API.

As described in more detail in our previous work [11], Blender v2.93 is used to automatically generate synthetic training images for a turbine blade detection task. In [11], various strategies for generating images were compared, including different lighting, background, object texture, additional foreground objects, and bounding box computation. Based on these results, a virtual camera is created for each scene and one of the three turbine blade models shown in Figure 3 is added with a randomized position. For the turbine blade models, a realistic-looking material texture is sampled from a pre-defined set of texture images that are either gray or dark blue. Furthermore, up to three distractor objects are added with a randomly selected material texture from a pool of texture images. For each virtual scene, a high dynamic range image is randomly sampled for image-based lighting. After rendering the scene, a random image from the COCO dataset [48] is added to the image background. Thus, we generate an automatically annotated synthetic training dataset consisting of 5000 different images for our generic turbine blade detection task. As an example, a Blender scene and the resulting annotated image are shown in Figure 4. Our code for generating synthetic training data based on 3D models is publicly available on GitHub (https://github.com/ignc-research/blender-gen, accessed on 28 December 2023).

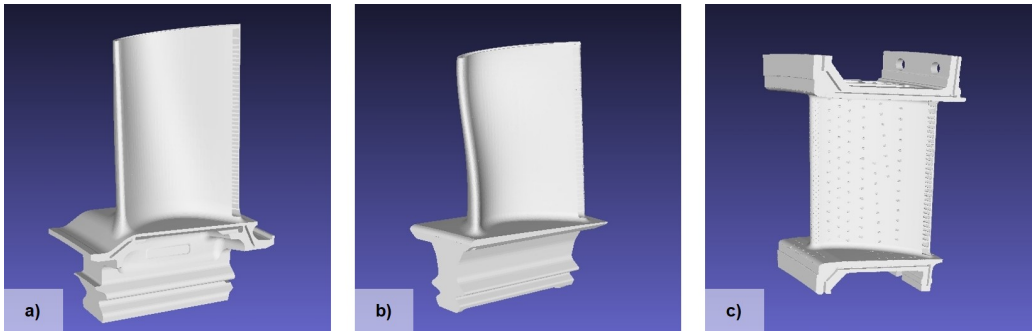

Figure 3. Three different industrial turbine blade models were used to generate synthetic training images. (**a**) Turbine blade 3D model 1. (**b**) Turbine blade 3D model 2. (**c**) Guide vane 3D model.

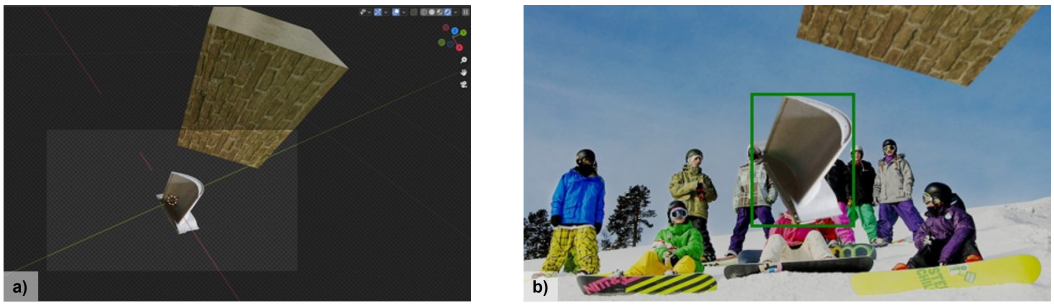

Figure 4. Synthetic data generation example. (**a**) Blender scene with a turbine blade and an additional distractor object. The box shows the camera view. (**b**) Generated image with bounding box annotation in green.

3.2. Real Dataset of Our Industrial Object Detection Use Case

We collected 1300 images in 1080P quality from two Microsoft Azure Kinect cameras on an industrial workbench from our previous work [49] over several days. The images were collected from two different camera angles. Each image contains a minimum of one and a maximum of three turbine blades. Example images are depicted in Figure 5. Tools and additional objects on the workbench create a moderate amount of clutter. We randomly split the collected data into a pool of 1000 training images and 300 validation images.

Figure 5. Annotated example images from the collected dataset. (**a**) Top view with three turbine blades on the table. (**b**) Side view with a clamped turbine blade. (**c**) Top view with a turbine blade in hand.

3.3. Object Detection Model Training Details

For our object detection model, we used the Faster R-CNN [50] implementation from MMDetection [51], which uses a feature pyramid network [52] based on a ResNet-50 backbone [53] and is pre-trained on the Microsoft COCO dataset [48]. We trained all our models with stochastic gradient descent with an input image size of 640×360, a batch size of 4, a learning rate of 0.00001, a momentum factor of 0.9, and a L^2 weight decay factor of 0.0001 [54]. To increase data efficiency, we use data augmentation during training. We used the library Albumentations [55] for online data augmentation, where we randomly performed flipping, color jitter, Gaussian noise, Gaussian blur, shifting, and scaling on training images. Augmenting training images is particularly useful when fine-tuning the model with small query batches of real images.

We trained all our models on an Nvidia GeForce RTX 3090 GPU until the average precision (AP) metric converged on the validation set. The AP metric is widely used to evaluate the performance of an object detection model. It computes the area under the precision-recall curve for a given threshold T and ranges from zero to one. Specifically, we use COCO's AP@[0.5:0.95], which uses 10 different thresholds $T = [0.5, 0.55, \ldots, 0.95]$ regarding the bounding box intersection over union and averages them into one single metric. A mathematical definition of AP@[0.5:0.95] can be found in [56].

3.4. Deep Active Learning Pipeline

Based on the comparative survey of DAL query strategies from Zhan et al. [28], we implemented an uncertainty-based query strategy and a hybrid query strategy. For our experiments, a pre-trained model is needed to complete one DAL cycle. For experiments with real images only, a publicly available Faster R-CNN base model M_0^R pre-trained on the COCO dataset was used. For experiments with synthetic images as described in Section 3.1, the COCO base model was fine-tuned on a labeled training set L_0 of 5000 synthetic images for 85 epochs, resulting in an average precision of AP@[0.5:0.95] = 0.555 for the synthetic base model M_0^S.

3.4.1. Uncertainty-Based Query Strategy

Considering the results from Brust et al. [35], we chose maximum margin sampling with the sum aggregation method as our uncertainty-based query strategy. In maximum margin sampling, an informativeness score s_{margin} for a detected object x_d is calculated according to Equation (1), where $P(\hat{y}_1|x_d)$ is the predicted probability of the class with the highest confidence and $P(\hat{y}_2|x_d)$ is the predicted probability of the second most confident class.

$$s_{margin}(x_d) = 1 - [P(\hat{y}_1|x_d) - P(\hat{y}_2|x_d)] \tag{1}$$

Because an image x can contain D detections, an aggregation method is required to combine multiple detections into one score. The sum aggregation method $a_{sum}(x)$ simply computes the sum over all detections in an image according to Equation (2).

$$a_{sum}(x) = \sum_{d \in D} s_{margin}(x_d) \tag{2}$$

If the OD model returns zero detections for an image, then $a_{sum}(x)$ is set to zero. Intuitively, the uncertainty-based query strategy described in Algorithm 1 will select samples x with multiple uncertain detections per image.

Algorithm 1 Maximum margin sampling

Input: Unlabeled pool of images U, empty labeled training set L, query batch size k, pre-trained model M_0^S
Output: Fine-tuned model M

$t = 1$
loop
  Obtain informativeness score $a_{sum}(x)$ for every image $x \in \{U, L\}$
  if an image x has no detections **then**
    Set $a_{sum}(x) = 0$
  end if
  Select and label top k images with the highest scores, add them to L
  Fine-tune object detection model M_t on labeled training set L
  $t = t + 1$
end loop

3.4.2. Hybrid Query Strategy

As a hybrid query strategy, we chose the diverse mini-batch active learning (DBAL) algorithm from Zhdanov [33]. As described in Algorithm 2, DBAL first filters out training images with a low informativeness score by using a pre-filter factor β. To this end, the top βk images are selected for further processing. In our experiments, $\beta = 2$ was used. Then, k diverse samples are selected from the remaining βk images with weighted KMeans++ clustering [57], where the weights are represented by the maximum margin informativeness scores. By selecting the image closest to each of the k clusters, the selected training images are expected to be more diverse.

In order to perform clustering, feature vectors that represent the training images x are required. We use the last feature map P_2 of size $(256, 90, 160)$ from the feature pyramid network model M_0^S [52] and perform global average pooling to convert the feature map to a one-dimensional feature vector of size 256. These feature vectors are then used for weighted KMeans++ clustering.

Algorithm 2 DBAL

Input: Unlabeled pool of images U, empty labeled training set L, query batch size k, pre-filter factor β, pre-trained model M_0^S
Output: Fine-tuned model M

$t = 1$
loop
  Obtain informativeness score $a_{sum}(x)$ for every image $x \in \{U, L\}$
  if an image x has no detections **then**
    Set $a_{sum}(x) = 0$
  end if
  Pre-filter to top βk informative images
  Cluster βk images to k clusters with weighted KMeans++
  Select and label k images closest to the cluster centers, add them to L
  Fine-tune the object detection model M_t on labeled training set L
  $t = t + 1$
end loop

4. Results

Using the described methodology from Section 3, we trained multiple OD models by combining synthetic data and DAL. As training data, we used either only real training

images (R) or we used the synthetically pre-trained model M_0^S and then fine-tuned it on real images (S+R). For DAL query strategies, we implemented the two described algorithms from Sections 3.4.1 and 3.4.2. Additionally, we implemented a random sampling strategy as a baseline, which shuffles the unlabeled pool of images and then selects a batch of k training images randomly. We ran each random strategy three times using different random seeds.

4.1. Combining Synthetic Images and Deep Active Learning for One DAL Cycle

First, we ran experiments for Algorithms 1 and 2, and random sampling for one DAL cycle with different query batch sizes k. Results for different DAL query strategies are shown in Figure 6. All numerical results can be found in the Appendix A in Table A1.

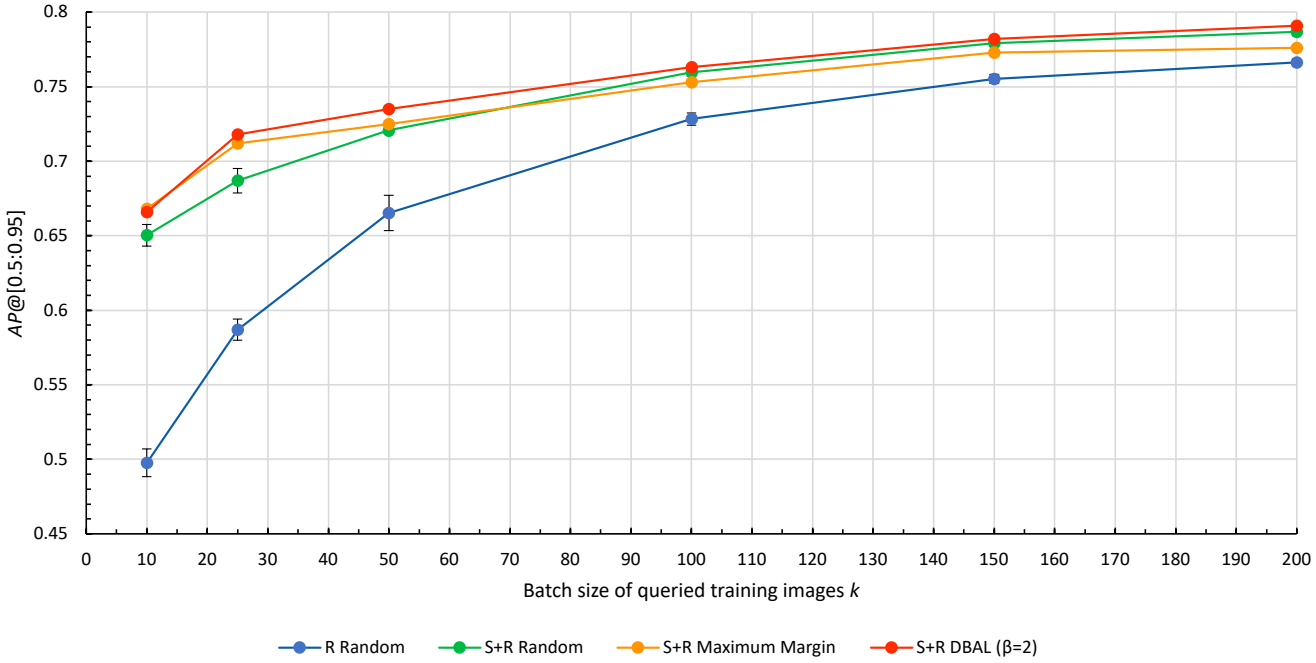

Figure 6. Results for the first DAL cycle with different query strategies. R Random: Baseline model using random sampling and only real images. S+R Random: Synthetic base model fine-tuned on real images with random sampling. S+R Maximum Margin: Synthetic base model fine-tuned on real images with Algorithm 1. S+R DBAL: Synthetic base model fine-tuned on real images with Algorithm 2.

Using synthetic training images for model pre-training always outperformed using only real images. In fact, the difference between using synthetic images and not using synthetic images is much greater than the difference between the different query strategies. The results show that the importance of synthetic images increases as the number of labeled training images decreases. For $k = 10$, the model pre-trained on a synthetic dataset (S+R Random) increased the AP@[0.5:0.95] by 30.5% compared to the baseline model trained only on real images (R Random).

The hybrid query strategy DBAL has a higher AP than the random query strategy for all batch sizes k and shows overall the best performance. The chart shows that DAL query strategies are most useful with a small number of training images selected from a bigger pool of unlabeled data. The largest improvement over random sampling is at $k = 25$, where S+R DBAL increased the AP by 4.5% in comparison to S+R Random. In other words, using 25 real training images with S+R DBAL yielded equivalent AP results to randomly selecting about 50 training images. For large batch sizes with $k \geqslant 100$, neither DAL query strategy yielded a meaningful improvement in model performance over random sampling in the first DAL cycle. As k approaches the total number of images in U, all query strategies must converge eventually. As shown by the standard error, selecting training images randomly

yields varying AP values due to dependence on the random seed. Therefore, employing DAL minimizes the chance of selecting an unfavorable random seed.

Figure 7 shows the top five selected images from the unlabeled pool U by the initial model $M_{0,S}$ according to the different query strategies in the first DAL cycle. As expected from Equations (1) and (2), maximum margin sampling and DBAL both select images from the unlabeled pool U with many false positive detections with high uncertainty.

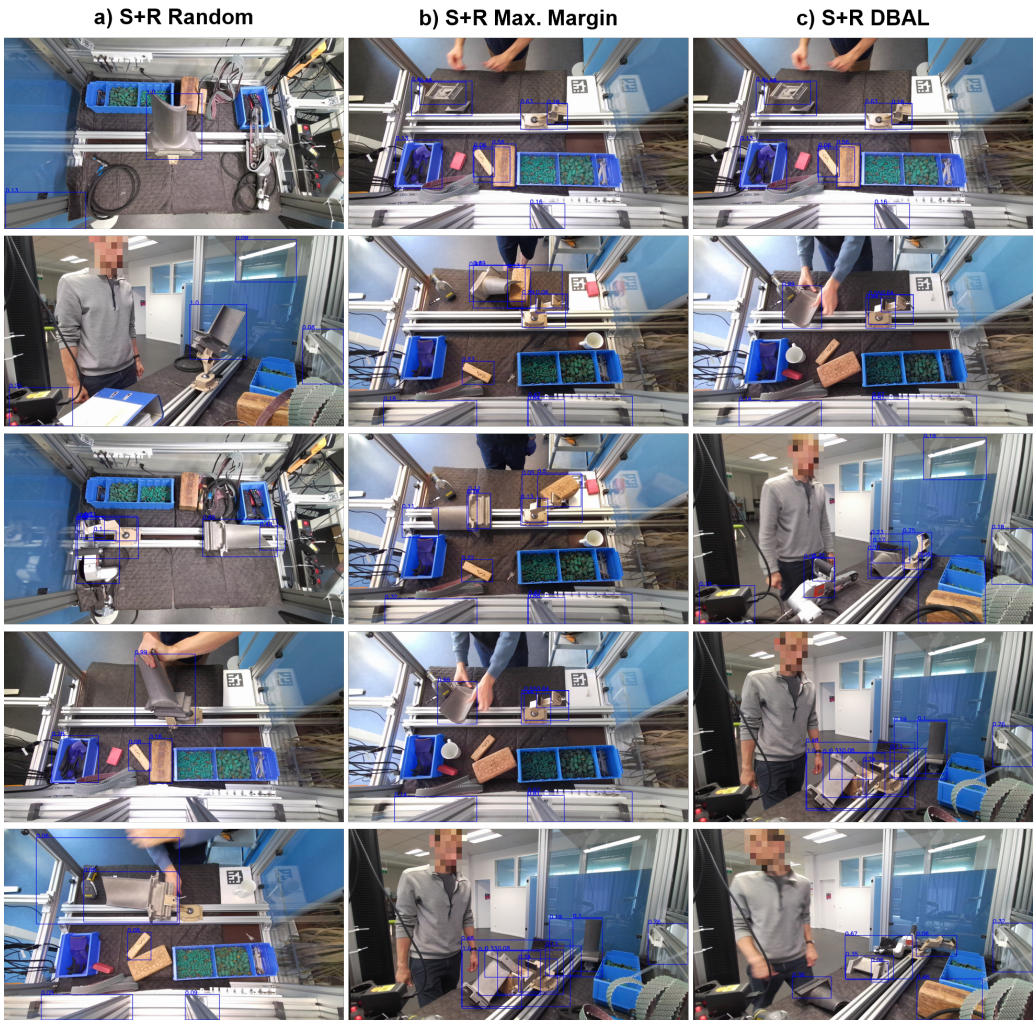

Figure 7. Top five training images for the initial model M_0^S from the unlabeled pool U according to the different query strategies. Bounding box predictions are displayed in blue, including the turbine blade class confidence value. Best viewed with zoom. (**a**) Top five training images according to S+R Random. (**b**) Top five training images according to S+R Maximum Margin. (**c**) Top five training images according to S+R DBAL.

4.2. Multiple Deep Active Learning Cycles

Based on our findings in Section 4.1, we opted for DBAL as our query strategy with a fixed batch size of $k = 25$. Starting with the synthetic base model M_0^S, the model was iteratively fine-tuned for eight DAL cycles according to Algorithm 2. At each cycle, the labeled training set L was extended by the 25 selected samples $x \in \{U, L\}$, based on the feature vectors from the previously trained model. Results for DBAL with up to $t = 8$ DAL cycles are compared to the previous charts in Figure 8 for a single cycle. Numerical results can be found in the Appendix A in Table A1.

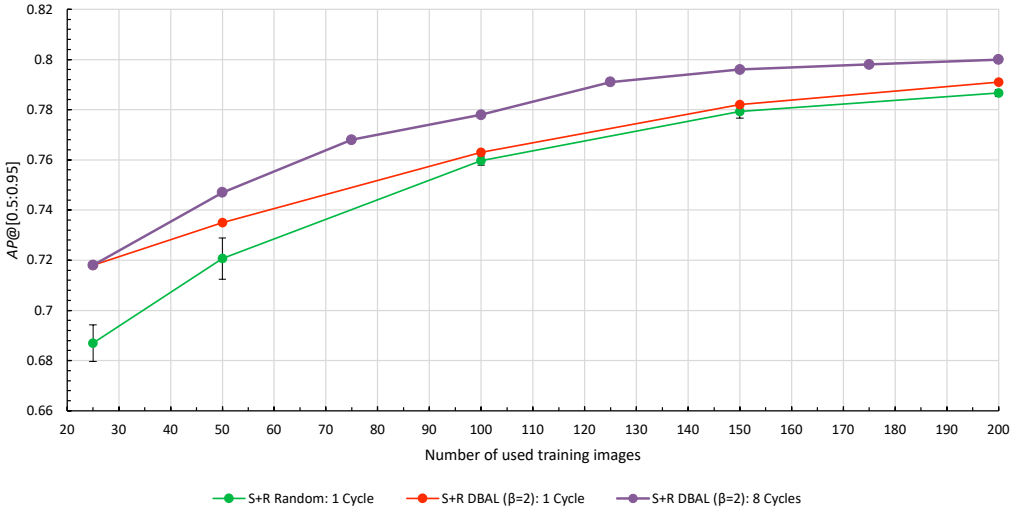

Figure 8. Results for one DAL cycle with varying batch sizes k compared to eight DAL cycles with a fixed batch size of $k = 25$.

The results presented in Figure 8 show that running DBAL for multiple DAL cycles yields better OD performance compared to running only a single cycle with an equivalent number of training images. For instance, a single cycle of DBAL with 150 labeled images performed the same as running four cycles of DBAL with 25 new images each time, which requires a maximum amount of 100 labeled images. Qualitative results on validation images are depicted in Figure 9 which shows the iterative learning of the model over the course of multiple DBAL cycles. False positive detections are reduced and the confidence values of turbine blade detections increase with each new cycle.

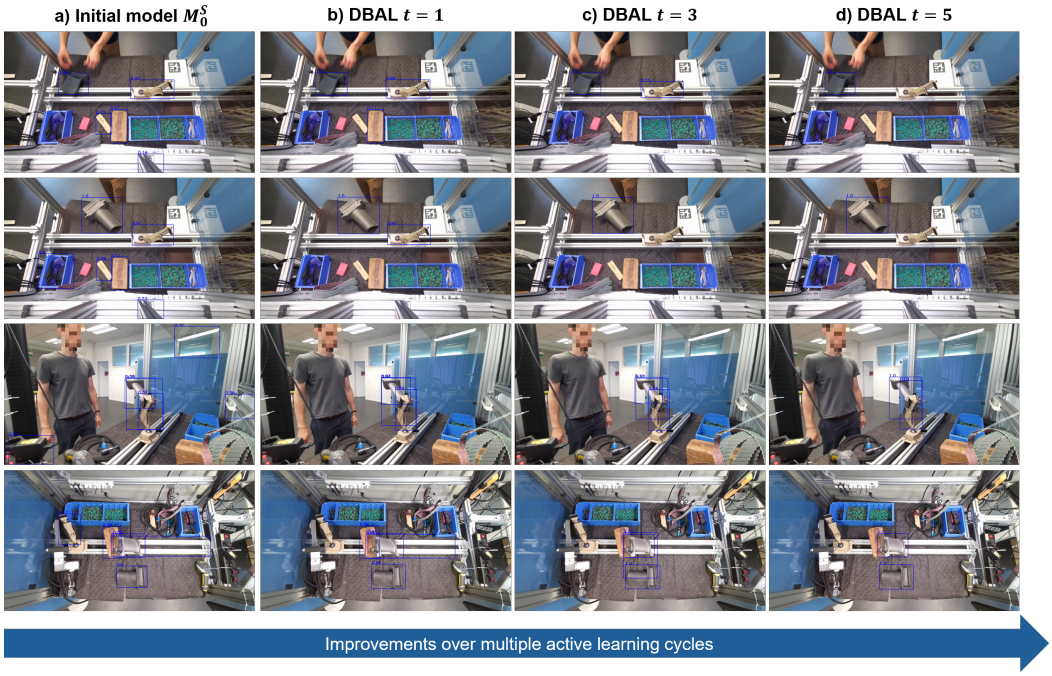

Figure 9. Qualitative results from S+R DBAL on validation images throughout multiple active learning cycles. Bounding box predictions are displayed in blue, including the turbine blade class confidence value. Best viewed with zoom. (**a**) Results from the initial model trained only on synthetic images. (**b**) Results after one cycle with real images. (**c**) Results after three cycles with real images. (**d**) Results after five cycles with real images.

5. Discussion and Conclusions

To summarize, this work combined the generation of synthetic training images with DAL in order to train industrial OD models with minimal manual annotations. The base model is initially trained on automatically generated synthetic images and subsequently fine-tuned in each DAL cycle with real images. The synthetic base model enables early deployment, while unlabeled real training images can be collected over time. To ensure data efficiency, the DAL query strategy selects a limited batch of images for training from a larger pool of unlabeled images. On our turbine blade detection dataset, we found that using synthetic images for pre-training improved model performance, especially when the number of real training images was small. Additionally, the hybrid query strategy DBAL outperformed uncertainty-based maximum margin sampling and random sampling for small batch sizes. Furthermore, running multiple DAL cycles with a small batch size performed better than running only one cycle with an equivalent number of training images. Utilizing DAL can either increase model performance with the same amount of data, or provide the same performance with fewer data compared to randomly selecting training images. Additionally, employing DAL minimizes the risk of selecting an unfavorable batch of training images by chance.

Our findings are limited by our specific industrial use case of a turbine blade detection model. However, the presented methodology is not restricted to turbine blades and can be applied to any object. In future work, we plan to apply our approach to new industrial applications and datasets. For both of our implemented DAL query strategies, we used maximum margin as an informativeness score combined with the sum aggregation method. Choosing an alternative informativeness score and aggregation method could lead to different results. For our experiments with multiple DAL cycles in Section 4.2, we did not change the unlabeled pool of images U. However, during real-world deployment of an OD model, it is possible to collect new images over time. A steady increase in U will provide the DAL query strategy with a larger selection of images to choose from.

As a next step, we would like to train and iteratively improve multiple OD models using the developed workflow over a longer period of time on the shop floor. Future work should incorporate best practices from the machine learning operations (MLOps) paradigm [58] to automatically train and test new models and to ensure that each model update performs better than the previous model. Automatic triggering of a new DAL cycle could be initiated through continuous model monitoring. For instance, this could occur when a specific amount of new data in U are collected, a certain time period has passed, a dataset shift is detected [59], or model performance declines on key metrics.

Author Contributions: Conceptualization, J.L. and L.E.; methodology, L.E.; software, L.E.; data curation, L.E.; writing—original draft preparation, L.E.; writing—review and editing, J.L.; visualization, L.E.; supervision, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is part of the project MRO 2.0—Maintenance, Repair and Overhaul and was supported in part by the European Regional Development Fund (ERDF) under grant number ProFIT-10167454. We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The industrial turbine blade data are not publicly available due to protection of intellectual property.

Acknowledgments: We would like to thank our MRO 2.0 project partners Siemens Energy, Gestalt Robotics, and Fraunhofer Institute for Production Systems and Design Technology.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AL	active learning
AP	average precision
DAL	deep active learning
DBAL	diverse mini-batch active learning
MLOps	machine learning operations
OD	object detection

Appendix A

Numerical results from the experiments from Section 4 are documented in Table A1.

Table A1. Numerical results for all experiments. R: Only real training images were used; S+R: Synthetic base model fine-tuned with real training images.

Strategy	Number of Real Training Images	*AP*@[0.5:0.95] Random Seed 1	*AP*@[0.5:0.95] Random Seed 2	*AP*@[0.5:0.95] Random Seed 3	(Average) *AP*@[0.5:0.95]
R Random *	10	0.479	0.507	0.507	0.498
	25	0.578	0.582	0.601	0.587
	50	0.686	0.665	0.645	0.665
	100	0.733	0.732	0.720	0.728
	150	0.754	0.761	0.751	0.755
	200	0.765	0.771	0.763	0.766
S+R Random *	0				0.555
	10	0.636	0.655	0.660	0.650
	25	0.671	0.692	0.698	0.687
	50	0.724	0.718	0.720	0.721
	100	0.757	0.765	0.757	0.760
	150	0.778	0.782	0.778	0.779
	200	0.782	0.788	0.790	0.787
S+R Max. Margin	0				0.555
	10				0.668
	25				0.712
	50				0.725
	100				0.753
	150				0.773
	200				0.776
S+R DBAL	0				0.555
	10				0.666
	25				0.718
	50				0.735
	100				0.763
	150				0.782
	200				0.791
S+R DBAL (8 cycles)	0				0.555
	25				0.718
	50				0.747
	75				0.768
	100				0.778
	125				0.791
	150				0.796
	175				0.798
	200				0.800

* Random sampling strategies were repeated with three different random seeds.

References

1. Gupta, C.; Farahat, A. Deep Learning for Industrial AI: Challenges, New Methods and Best Practices. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 6–10 July 2020. [CrossRef]
2. Torralba, A.; Efros, A.A. Unbiased look at dataset bias. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado Springs, CO, USA, 20–25 June 2011; IEEE: New York, NY, USA 2011. [CrossRef]
3. Coyner, A.S.; Chen, J.S.; Chang, K.; Singh, P.; Ostmo, S.; Chan, R.V.P.; Chiang, M.F.; Kalpathy-Cramer, J.; Campbell, J.P. Synthetic Medical Images for Robust, Privacy-Preserving Training of Artificial Intelligence: Application to Retinopathy of Prematurity Diagnosis. *Ophthalmol. Sci.* **2022**, *2*, 100126. [CrossRef] [PubMed]
4. Northcutt, C.; Athalye, A.; Mueller, J. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 (NeurIPS Datasets and Benchmarks 2021), Virtual, 6–14 December 2021; Vanschoren, J., Yeung, S., Eds.; Curran: Red Hook, NY, USA 2021; Volume 1.
5. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: New York, NY, USA 2017. [CrossRef]
6. Lambrecht, J.; Kästner, L. Towards the Usage of Synthetic Data for Marker-Less Pose Estimation of Articulated Robots in RGB Images. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; IEEE: New York, NY, USA 2019. [CrossRef]
7. Nowruzi, F.E.; Kapoor, P.; Kolhatkar, D.; Hassanat, F.A.; Laganiere, R.; Rebut, J. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *arXiv* **2019**, arXiv:1907.07061. [CrossRef]
8. Movshovitz-Attias, Y.; Kanade, T.; Sheikh, Y. How Useful Is Photo-Realistic Rendering for Visual Learning? In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2016; pp. 202–217. [CrossRef]
9. de Melo, C.M.; Rothrock, B.; Gurram, P.; Ulutan, O.; Manjunath, B. Vision-Based Gesture Recognition in Human-Robot Teams Using Synthetic Data. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Virtual, 24 October 2020–24 January 2021; pp. 10278–10284. [CrossRef]
10. Yang, X.; Fan, X.; Wang, J.; Lee, K. Image Translation Based Synthetic Data Generation for Industrial Object Detection and Pose Estimation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7201–7208. [CrossRef]
11. Eversberg, L.; Lambrecht, J. Generating Images with Physics-Based Rendering for an Industrial Object Detection Task: Realism versus Domain Randomization. *Sensors* **2021**, *21*, 7901. [CrossRef] [PubMed]
12. Schraml, D. Physically based synthetic image generation for machine learning: A review of pertinent literature. In Proceedings of the Photonics and Education in Measurement Science 2019, Jena, Germany 17–19 September 2019; Proc. SPIE: Bellingham, WA USA, 2019; Volume 11144. [CrossRef]
13. Georgakis, G.; Mousavian, A.; Berg, A.; Kosecka, J. Synthesizing Training Data for Object Detection in Indoor Scenes. In Proceedings of the Robotics: Science and Systems XIII. Robotics: Science and Systems Foundation, Cambridge, MA, USA, 12–16 July 2017. [CrossRef]
14. Dwibedi, D.; Misra, I.; Hebert, M. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017. [CrossRef]
15. Gorschlüter, F.; Rojtberg, P.; Pöllabauer, T. A Survey of 6D Object Detection Based on 3D Models for Industrial Applications. *J. Imaging* **2022**, *8*, 53. [CrossRef] [PubMed]
16. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: New York, NY, USA, 2018. [CrossRef]
17. Prakash, A.; Boochoon, S.; Brophy, M.; Acuna, D.; Cameracci, E.; State, G.; Shapira, O.; Birchfield, S. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019; IEEE: New York, NY, USA, 2019. [CrossRef]
18. Hodan, T.; Vineet, V.; Gal, R.; Shalev, E.; Hanzelka, J.; Connell, T.; Urbina, P.; Sinha, S.N.; Guenter, B. Photorealistic Image Synthesis for Object Instance Detection. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; IEEE: New York, NY, USA 2019. [CrossRef]
19. Jabbar, A.; Farrawell, L.; Fountain, J.; Chalup, S.K. Training Deep Neural Networks for Detecting Drinking Glasses Using Synthetic Images. In *Neural Information Processing*; Springer International Publishing: Cham, Switzerland, 2017; pp. 354–363. [CrossRef]
20. Pharr, M.; Jakob, W.; Humphreys, G. *Physically Based Rendering: From Theory to Implementation*, 3rd ed.; Morgan Kaufmann: Burlington, Massachusetts, USA, 2016.
21. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning From Simulated and Unsupervised Images Through Adversarial Training. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2242–2251. [CrossRef]

22. Sankaranarayanan, S.; Balaji, Y.; Jain, A.; Lim, S.N.; Chellappa, R. Learning From Synthetic Data: Addressing Domain Shift for Semantic Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3752–3761. [CrossRef]
23. Peng, X.; Saenko, K. Synthetic to Real Adaptation with Generative Correlation Alignment Networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: New York, NY, USA, 2018. [CrossRef]
24. Rojtberg, P.; Pollabauer, T.; Kuijper, A. Style-transfer GANs for bridging the domain gap in synthetic pose estimator training. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), Virtual, 14–18 December 2020; IEEE: New York, NY, USA, 2020. [CrossRef]
25. 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]
26. Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; University of Wisconsin: Madison, WI, USA, 2009.
27. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Gupta, B.B.; Chen, X.; Wang, X. A Survey of Deep Active Learning. *ACM Comput. Surv.* **2021**, *54*, 1–40. [CrossRef]
28. Zhan, X.; Wang, Q.; hao Huang, K.; Xiong, H.; Dou, D.; Chan, A.B. A Comparative Survey of Deep Active Learning. *arXiv* **2022**, arXiv:2203.13450. [CrossRef]
29. Wang, D.; Shang, Y. A new active labeling method for deep learning. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; IEEE: New York, NY, USA, 2014. [CrossRef]
30. Sener, O.; Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In Proceedings of the 2018 International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
31. Ash, J.T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; Agarwal, A. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In Proceedings of the 2020 International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
32. Yin, C.; Qian, B.; Cao, S.; Li, X.; Wei, J.; Zheng, Q.; Davidson, I. Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; IEEE: New York, NY, USA, 2017. [CrossRef]
33. Zhdanov, F. Diverse mini-batch Active Learning. *arXiv* **2019**, arXiv:1901.05954. [CrossRef]
34. Li, Y.; Fan, B.; Zhang, W.; Ding, W.; Yin, J. Deep active learning for object detection. *Inf. Sci.* **2021**, *579*, 418–433. [CrossRef]
35. Brust, C.A.; Käding, C.; Denzler, J. Active Learning for Deep Object Detection. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), Prague, Czech Republic, 25–27 February 2019; SciTePress: Setúbal, Portugal, 2019; pp. 181–190. [CrossRef]
36. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
37. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–338. [CrossRef]
38. Haussmann, E.; Fenzi, M.; Chitta, K.; Ivanecky, J.; Xu, H.; Roy, D.; Mittel, A.; Koumchatzky, N.; Farabet, C.; Alvarez, J.M. Scalable Active Learning for Object Detection. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; IEEE: New York, NY, USA, 2020. [CrossRef]
39. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241. [CrossRef]
40. Peng, H.; Lin, S.; King, D.; Su, Y.H.; Bly, R.A.; Moe, K.S.; Hannaford, B. Reducing Annotating Load: Active Learning with Synthetic Images in Surgical Instrument Segmentation. *arXiv* **2021**, arXiv:2108.03534. [CrossRef]
41. Houlsby, N.; Huszár, F.; Ghahramani, Z.; Lengyel, M. Bayesian Active Learning for Classification and Preference Learning. *arXiv* **2011**, arXiv:1112.5745. [CrossRef]
42. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
43. Niemeijer, J.; Mittal, S.; Brox, T. Synthetic Dataset Acquisition for a Specific Target Domain. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Paris, France, 2–6 October 2023; pp. 4055–4064.
44. Wang, Y.; Ilic, V.; Li, J.; Kisačanin, B.; Pavlovic, V. ALWOD: Active Learning for Weakly-Supervised Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 6459–6469.
45. Denninger, M.; Sundermeyer, M.; Winkelbauer, D.; Olefir, D.; Hodan, T.; Zidan, Y.; Elbadrawy, M.; Knauer, M.; Katam, H.; Lodhi, A. BlenderProc: Reducing the Reality Gap with Photorealistic Rendering. In Proceedings of the Robotics: Science and Systems (RSS), Virtual, 12–16 July 2020.
46. Dirr, J.; Gebauer, D.; Yao, J.; Daub, R. Automatic Image Generation Pipeline for Instance Segmentation of Deformable Linear Objects. *Sensors* **2023**, *23*, 3013. [CrossRef] [PubMed]

47. Druskinis, V.; Araya-Martinez, J.M.; Lambrecht, J.; Bøgh, S.; de Figueiredo, R.P. A Hybrid Approach for Accurate 6D Pose Estimation of Textureless Objects From Monocular Images. In Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), Sinaia, Romania, 12–15 September 2023; pp. 1–8. [CrossRef]
48. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
49. Eversberg, L.; Lambrecht, J. Evaluating digital work instructions with augmented reality versus paper-based documents for manual, object-specific repair tasks in a case study with experienced workers. *Int. J. Adv. Manuf. Technol.* **2023**, *127*, 1859–1871. [CrossRef]
50. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28, pp. 91–99.
51. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155. [CrossRef]
52. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
54. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
55. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]
56. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
57. Arthur, D.; Vassilvitskii, S. k-means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
58. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* **2023**, *11*, 31866–31879. [CrossRef]
59. Moreno-Torres, J.G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N.V.; Herrera, F. A unifying view on dataset shift in classification. *Pattern Recognit.* **2012**, *45*, 521–530. [CrossRef]