DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

# Improving Visual Object Detection Using Synthetic Self-Training

## Verbesserung der Objekterkennung durch Selbst-Training mit synthetischen Bildern

***Christopher Mayershofer***
***Adrian Fischer***
***Johannes Fottner***

*Chair of Materials Handling, Material Flow, Logistics*
*Department of Mechanical Engineering*
*Technical University of Munich*

**The current era of supervised learning requires a large corpus of application-specific training data with ground-truth annotations. The creation of large annotated datasets however is a costly endeavor. Moreover, the availability of a large annotated set of training data cannot be guaranteed in certain domains. Self-training attempts to overcome these problems by using a set of labeled data and a potentially infinite pool of unlabeled data to train a model in a semi-supervised manner. Self-training however only works if the annotated data is sufficient for training a strong teacher model, which depending on the application domain, is not always feasible. In this work, we propose and formulate a simple extension to the self-training paradigm and refer to it as Synthetic Self-Training (SST). SST incorporates synthetically generated images into conventional self-training in order to reduce the aforementioned problems, therefore improving model performance. Specifically, we address the problem of object detection in a logistics environment and are able to improve the state-of-the-art detection performance on the LOCO dataset by 12% $mAP_{0.5}$.**

*[Keywords: Synthetic Self-Training, Self-Training, Synthetic Data, Object Detection, Logistics]*

**Die gegenwärtige Praxis des überwachten Lernens erfordert einen umfangreichen Korpus annotierter Trainingsdaten. Die Erstellung großer annotierter Datensätze ist jedoch ein kostspieliges Unterfangen. Darüber hinaus variiert die Verfügbarkeit eines großen annotierten Trainingsdatensatzes über unterschiedliche Anwendungsbereiche. Selbst-Training versucht, diese Probleme zu überwinden, indem eine Kombination aus annotierten Daten und nicht-annotierten Daten verwendet wird, um das Modell zu trainieren. Selbst-Training bedarf jedoch einer ausreichenden Menge annotierter Trainingsdaten, um ein starkes Lehrermodell zu trainieren. Diese Arbeit stellt das Synthetische Selbst-Training (SST) vor, eine Erweiterung des konventionellen Selbst-Trainings. SST löst zuvor genannte Problemedurch Einbeziehen synthetisch erzeugter Daten in den Trainingsprozess. Diese Arbeit formuliert SST im Bereich der Visuellen Objekterkennung und zeigt empirische dessen Vorteile. Konkret ermöglicht SST eine Verbesserung der Erkennungsgenauigkeit logistik-spezifischer Objekte im LOCO Benchmark um 12% $mAP_{0.5}$.**

*[Schlüsselwörter: Synthetisches Selbst-Training, Selbst-Training, Synthetische Daten, Objekterkennung, Logistik]*

## 1 INTRODUCTION

Intelligent robots are increasingly demanded to interact with agents and their environment. To enable this physical interaction robots initially need to process raw sensor data (e.g., image or point cloud data) and extract high-level, semantic information (e.g. object information). This process is referred to as perception [1]. Object detection is one such perception capability typically assigned to autonomous agents to smartly perform actions. For example, a transport robot may be able to give priority to typically faster-moving manual forklift drivers, if it is able to be aware of its surroundings, therefore increasing the overall material flow efficiency. Object detection describes the process of locating the position of an object within a data frame while also assigning a semantic label to it [2]. Since AlexNet's [3] winning entry in the ImageNet Large Scale Visual Recognition Challenge [4], deep learning is increasingly applied as a function to map the raw sensor data input to high-level semantic information. Those models are also reaching state-of-the-art performance when detecting objects. In the current era of supervised learning however, a large set of training data is required in order to achieve a reasonable detection performance. Depending on the application domain, this requirement cannot be fulfilled due to a set of factors including safety regulations, privacy concerns or competitive behavior.

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

The Logistics Objects in Context (LOCO) dataset [5] tackles this problem for logistics applications. LOCO is the first publicly available scene understanding dataset for logistics applications depicting logistics objects in real logistics settings (e.g. warehouse, distribution center). The data allows training an object detection model, able to detect logistics-specific objects in their realistic environment as presented in [5]. Yet, state-of-the-art supervised training methods and architectures are not capable of reaching the same performance level on the LOCO benchmark as compared to other common detection benchmarks. In particular, experiments [5] show an average detection performance gap over three different detection models of 27.5 % $mAP_{0.5}$ compared to the COCO [6] detection benchmark, research's de-facto object detection standard. We hypothesize, this is on the one hand, due to the different dataset statistics and on the other hand due to the lesser extent of annotated training data available, as LOCO only provides a limited set of manually annotated training data. While the trivial solution of simply annotating more data is assumed to increase performance, in this work we focus on making use of not-annotated data in a semi-supervised learning fashion to boost detection performance.

Semi-supervised learning describes a learning paradigm that lies in between supervised learning and unsupervised learning, making use of both, a set of labelled data and a set of unlabelled data. This reduces labelling cost and even more so enables making use of a potentially infinite pool of unlabelled data. Self-Training is a semi-supervised learning method that has shown great progress in the image classification and object detection domain [7] [8] [9] [10]. Self-Training deploys a student-teacher-framework in a three step process: Initially a teacher is trained in a supervised fashion using the labelled data. Next, the teacher is used to infer labels (also known as pseudo labels [11]) for the unlabelled data. Finally, a student is trained in a supervised manner on a combination of the labelled and pseudo-labelled data. While this concept is feasible for domains with a large enough labelled pool of training data (e.g. image classification [8] or detection of common objects [12]), self-training was shown to be sensitive to bad teachers [13]. This implicitly hinders the use of powerful self-training approaches within smaller, domain-specific application scenarios.

Within this paper we show that conventional self-training on the LOCO dataset (i.e. a medium-sized, domain-specific dataset) is affected by the bad teacher problem. Therefore, this paper proposes and studies the concept of Synthetic Self-Training (SST). SST's intuition is simple: In order to improve the teacher's initial performance in application domains lacking large pools of annotated data, synthetically generated data can help to improve teacher performance. This simple concept can extend self-training approaches and make them useable for smaller, domain-specific datasets. In particular, within this paper we empirically show a 12% $mAP_{0.5}$ performance increase for logistics specific object detection over the previous state-of-the-art.

## 2 Related Work

This work proposes synthetic self-training which leverages advances of three strong concepts in deep learning research: *consistency regularization*, p*seudo-labeling* and *synthetic data generation*. Consistency regularization builds on the idea that a model's prediction is supposed to be consistent when inputting a perturbed version of the image **[14]**. Intuitively (and also theoretically **[15]**) this is possible because the perturbations change an image's style but its semantic content is kept constant. Consistency regularization therefore is an important building block for self-training as used, e.g. in **[16] [12] [17] [18]**. Pseudo-labelling on the other hand describes the concept of using a trained model to generate annotations for unlabelled data **[19]**. In practice, a confidence threshold over the models prediction confidence is used to reduce erroneous labels **[11]**. Synthetic data generation describes the process of computationally generating annotated data. This can be done by leveraging, e.g. computer graphics **[20]** or generative models **[21]**. The performance difference as introduced when changing from the synthetic domain to a natural one is referred to as Sim-2-real gap.

Before Self-training ideas were applied on object detection, they were studied on image classification tasks. For image classification, reference [16] proposes noisy student training and shows that incorporating noise both, on an image (i.e. strong augmentations) and model (i.e. dropout, stochastic depth) level boosts classification performance. Moreover, they show that noisy student training significantly improves model robustness when exposed to noised data. Instead of comparing "hard" annotations, reference [22] proposes to directly compare the latent space between teacher and augmented student as a training signal. The teacher is improved over time using a moving average. The breakthroughs in self-training on image classification tasks are also being applied to the problem of object detection. Reference [18] propose consistency regularization and confident pseudo labels for their self-training approach FixMatch. Reference [23] presents a self-training technique for object detection in the pre-deep-learning era. They use a sliding-window based approach in conjunction with a cascaded Bayesian classifier for detection [24] [25]. The detector is initially trained on weakly-labelled data using Expectation Maximization, a technique for estimating model parameters given noisy data. After inferring pseudo-labels using the trained model on the unlabelled data, the predictions get sieved using a set of selection metrics, resulting in a set of pseudo-labelled data with confident predictions only. The data then gets incrementally added to the initial training set and is used for training a student model. The advent of deep learning then allowed using more capable detectors to be used within the self-training paradigm.

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

After initial breakthroughs of Self-Training in the image classification domain, it was also applied to the object detection domain: Jeong et al. [26] introduce a consistency-based self-training approach called CSD. Similar to Noisy Student training, they propose using augmentations as a way for regularizing the model via input consistency, forcing the network to infer consistent predictions. Specifically, they use the Flip operation as augmentation only and perform experiments on the MS COCO and VOC dataset using both, two-stage (R-FCN) and single-stage detectors (SSD). Sohn et. al [27] then propose the STAC framework that employs strong augmentations in a noisy student fashion [16] and adds an unsupervised loss for regularization. Specifically, they use a two-stage detector (Faster R-CNN), train it on the labelled data (a portion of MS COCO) and pseudo-label the unlabelled data, applying a simple confidence threshold. The student is then trained on the mixed set of labelled and pseudo-labelled data. To regularize the overall loss, they employ a weighted average over labelled and pseudo-labelled data. Similarly, Zoph et al. [12] provide an empirical study on pre-training and self-training. They use a similar Self-Training framework as described in STAC. However, they use rather large single-stage detectors (SpineNet) and normalize the loss of labelled and unlabelled data for regularization purposes. Finally, the unbiased teacher framework by Liu et al. [17] even beats the STAC framework. They propose a mutually-beneficial student-teacher paradigm that initially trains a strong teacher which is consecutively used for pseudo labeling. Afterwards, the function parameters are copied to the student which continues training on the pseudo-labelled data under heavy data augmentation. After convergence, the student knowledge is incorporated into the teacher using an exponential moving average. Furthermore, in order to overcome the class-bias of background-classes in semi-supervised learning, they propose using Focal Loss [28] [28]. Yet, none of the presented approaches leverages the potential of synthetically generated images.

Other data modalities and tasks however have shown success when leveraging synthetic data for self-training. Reference [29] uses artificially generated data by a convex combination of patterns in a self-supervised training regime on a simulated, low dimensional toy dataset. To boost natural language processing, reference [30] leverages synthetically generated question-answering (QA) pairs in a self-training fashion. Here a pretrained model is used to generate synthetic QA pairs, which in turn is pseudo-labelled and incorporated into the training set. Moreover, reference [31] proposes the use of synthetic data for self-supervised training in a neural machine translation context. Reference [32] demonstrates the use of synthetic data for self-training on the task of pose estimation. As no prior work incorporated synthetic data into the self-training paradigm for object detection, the following section formalizes Synthetic Self-Training, a self-training approach for visual object detection that builds on consistency regularization, pseudo-labelling and incorporating synthetic data.

## 3 SYNTHETIC SELF-TRAINING

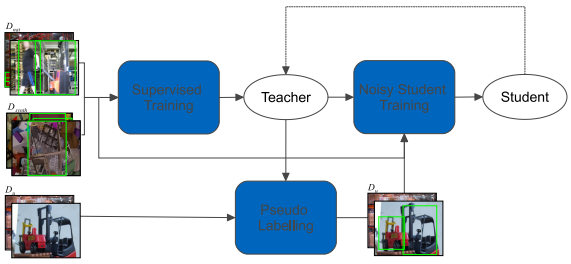



*Figure 1. Synthetic Self-Training leverages the student-teacher framework for semi-supervised learning. Initially, a teacher is trained in a supervised fashion on a combination of synthetic and natural data. The teacher is then used to generate pseudo-labels on the unlabelled data. Finally, a student is trained in a noisy student fashion on the combination of synthetic, natural and pseudo-labelled data.*

Self-training addresses the problem of training a model with a small set of manually annotated, natural data $D_{s,nat} = \{x^i_{s,nat}, y^i_{s,nat}\}_{i=1}^{N_{s,nat}}$ and a potentially infinite set of unlabelled data $D_u = \{x^i_u\}_{i=1}^{N_u}$. In contrast to known self-training approaches as studied in research (where large datasets, e.g. COCO are used as $D_{s,nat}$), many domain- and application-specific settings lack such a well-balanced, natural annotated dataset. Our goal is to reduce this dependency by leveraging synthetically generated images $D_{s,synth} = \{x^i_{s,synth}, y^i_{s,synth}\}_{i=1}^{N_{s,synth}}$ to train a confident teacher.

Therefore, we introduce Synthetic Self-Training (SST), that allows overcoming this problem in the well-established student-teacher framework. The student-teacher-framework starts by training a teacher model on a combination of natural and synthetic images in a supervised fashion. This teacher is then used to infer labels for the unlabelled data. Afterwards, a student network is trained on a combination of the labelled (both, natural and synthetic) and pseudo-labelled data. By replacing the teacher with the student, we can iterate the process until convergence. Synthetic Self-Training provides a performance boost by explicitly incorporating synthetically generated training data effectively extending self-training. It builds on three major advances in representation learning research, namely consistency regularization using strong augmentations, confident pseudo-label generation, and synthetic data generation as a means to provide labelled data for skewed or underrepresented classes. Figure 1 gives an overview of our approach.

The teacher's performance of generating highly confident annotations during inference is of utmost importance for any self-training schema [13]. Therefore, the initial focus is set on training a confident teacher. In order to overcome the issue of biased, missing or underrepresented annotations for domain-specific applications, we propose

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

using synthetically generated data. While fully-synthetic training is deemed the Holy Grail of AI, it lacks detection performance when compared to training on synthetic and natural images combined. We therefore settle for combining natural and synthetic images for training our teacher, both sets being ground-truth annotated. The teacher $\theta_{teacher}$ is then trained in a supervised fashion, minimizing the detection loss L over the training set $T$,

$$T = D_{s,nat} \cup D_{s,synth}$$

with

$$L_{set}(\theta_{teacher}) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{l}_{det}\left(f_{\theta_{teacher}}(x_i), y_i\right)$$

Where $i$ is an index denoting an item within the dataset set, $x_i$ is the image at the index and $y_i$ the corresponding ground-truth annotation. The detection loss $\mathrm{l}_{det}$ then measures the distance between the teacher's prediction $y_i^* = f_{\theta_{teacher}}(x_i)$ and its ground-truth annotation $y_i$, consisting of an architecture-dependent, weighted combination of classification loss and regression loss.

Next, we set our focus on generating confident pseudo labels. Therefore, the trained teacher $f_{\theta_{teacher}}$ is used in a pure inference fashion (i.e. no backward-pass computation), predicting a pseudo-label $y_i^{\sim}$ for each image $x_i$ within the unlabelled dataset $D_u$. Note, that the model inference includes all postprocessing steps, such as non-maximum suppression to remove redundant predictions, similar to [27]. We found prediction aggregation strategies, such as test-time augmentation[1] or model ensembles[2] to be useful for further boosting detection performance during inference, while still keeping the approach scalable and flexible (i.e. no architecture dependent changes). Finally, a threshold $\tau$ is used to remove erroneous labels based on the models prediction confidence [7] [18].

The student model is then trained in a noisy student fashion [16]. The model is initialized using the teacher's weights and trained to minimize the synthetic self-training loss $\mathrm{L}_{SST}$ over the training set combining natural, synthetic and pseudo-labelled data

$$L_{SST}(\theta_{student}) = L_{nat}(\theta_{student}) + L_{synth}(\theta_{student}) + L_{pseudo}(\theta_{student})$$

with

$$L_{pseudo}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{l}_{det}\left(f_{\theta}\left(augment(x_i)\right), y_i^{\sim}\right)$$

where $L_{pseudo}$ denotes the consistency-based loss using strong data augmentations over the pseudo-labels $y_i^{\sim}$. After convergence of the student, it itself can be used as a teacher for additional iterations by starting off with step two.

## 4 THE DATASET

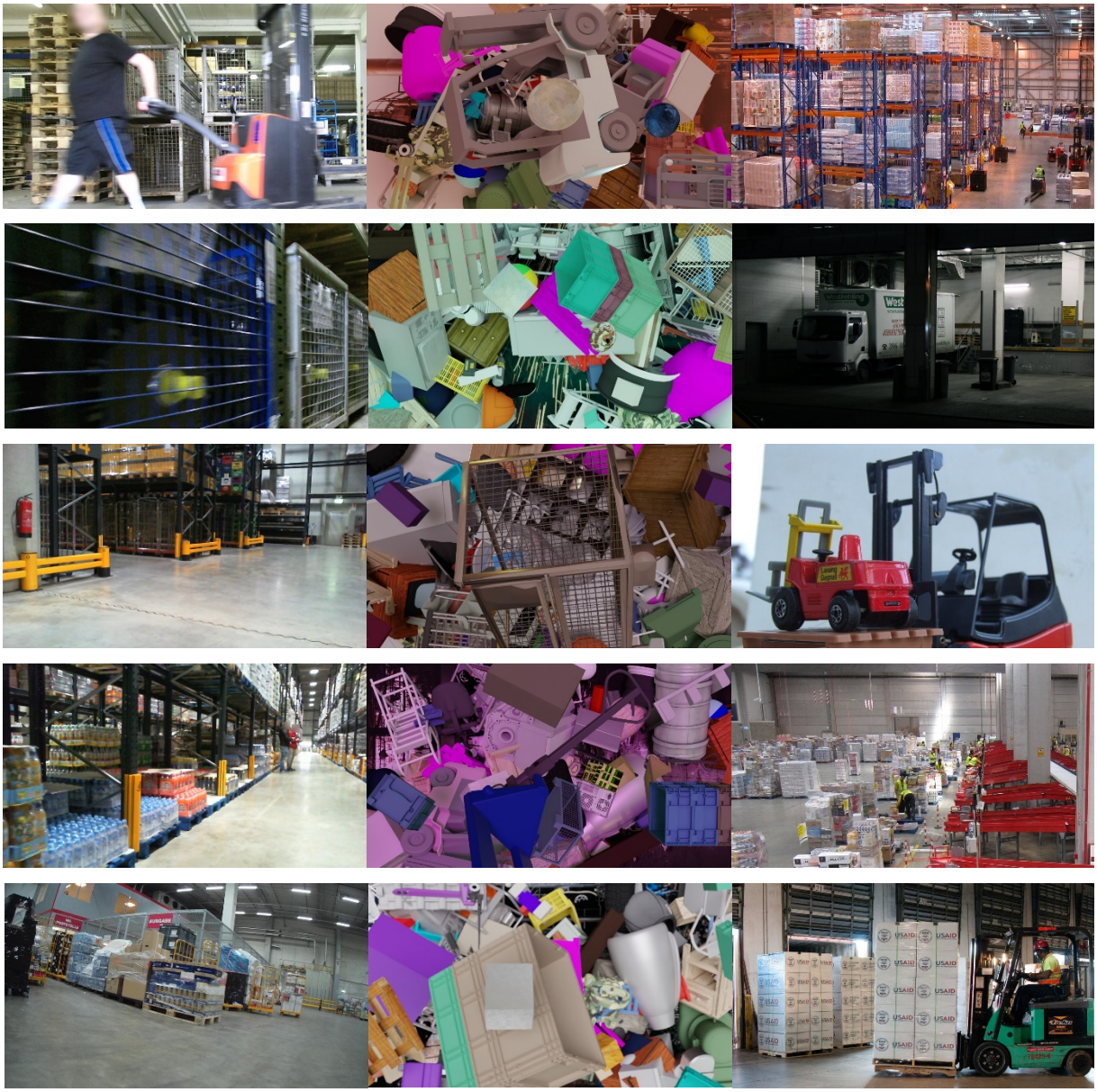
We empirically evaluated SST within a logistics setting. For our experiments we relied on image data from three different sources, namely captured images from natural logistics environments (i.e. $D_{s,nat}$), automatically created synthetic images (i.e. $D_{s,synth}$), and publicly available images from the web (i.e. $D_u$). For images captured in a natural logistics environments, we relied on the LOCO dataset [5]. The LOCO dataset provides 2,820 and 2,277 labelled training and validation images respectively as well as 16,470 unlabelled training images. Furthermore, we used a domain-randomization[3] based synthetic image generation pipeline as introduced in [20] for generating artificial training data. Using this pipeline, 7,315 synthetic images were generated, covering pallets, pallet trucks, forklifts, stillages and small load carriers while also providing their bounding box annotation. Finally, we used domain specific image crawling to download unlabelled images from the web, further extending the training corpus. In total, 459,828 images were captured from the web. We use those images and create five distinguishable dataset components, namely *TRAIN*, *VAL*, *SYNTH*, *TRAIN (UNLABELLED)*, and *WEB (UNLABELLED)*. The *TRAIN* and *VAL* components represent the manually-annotated image data captured in real logistics scenes, the gold standard as introduced in [5]. *TRAIN (UNLABELLED)* refers to the not-sampled and therefore not annotated images from the data captured within the LOCO dataset. The *SYNTH* component refers to the synthetic images and annotations as briefly introduced before. Finally, the *WEB (UNLABELLED)* component refers to the images crawled from the web. Figure 2 provides sample images of the data used for our experiments.

[1] Test-time augmentation aggregates multiple predictions over a transformed (i.e. augmented) version of the input [34].

[2] Model ensembles aggregate multiple predictions of different models on the same input [35] [36].

[3] Domain randomization describes a method to reduce the performance variation when transferring a synthetically trained neural network into the real world. The intuition before domain randomization is, that by randomizing parameters in the source domain (i.e. simulation), the target domain (i.e. the real world) appears to the network as just another variation of the source domain [37] [38].

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288



[4] due to its pareto-optimality when considering both, inference speed and detection accuracy. We use the largest model size available for our experiments allowing to train a well performing teacher. Each model within the following experiments was trained on images of size 640x640 for 30 epochs using a batch size of 16, unless otherwise noted. A cosine-annealing learning rate decay schedule was used. All models are evaluated on LOCO's validation set *VAL*. The trainings were conducted on an Nvidia V100 GPU.

[4] https://github.com/ultralytics/yolov5

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

### 5.1 TRAINING A CAPABLE TEACHER

First, detection performance for different weight initialization strategies as well as different training data variations were studied. The training data was varied in that either *TRAIN* was used (Data-1) or a combination of *TRAIN* and *SYNTH* (Data-2). A fully synthetic training turned out not to be performant ($mAP_{0.5} < 1\%$) and was therefore not pursued further. In addition, different model initialization strategies were investigated. Variant one (Init-1) initializes the weights randomly, variant two (Init-2) uses publicly available weights pre-trained on the COCO dataset, variants three and four use weights pre-trained on the *SYNTH* component. Here, variants three and four again differ in the initialization of their weights: Variant three (Init-3) uses randomly initialized weights, while variant four (Init-4) was initialized with COCO-trained weights. All models were trained with the same hyperparameters. Experiments initialized with pre-trained weights ran for 30 epochs, whereas those initialized with random weights ran for 200 epochs to ensure convergence. Figure 2 shows the results of the experiments. Performance results of these experiments are documented in Table 1.

*Table 1. Performance results of trained teachers using different combinations of training data, initialization strategies and hyperparameter configurations (Higher is better)*

| Name | Hyperparameters | Model | Data* | Init** | $mAP_{0.5}$ | $mAP_{0.5:0.95}$ |
|---|---|---|---|---|---|---|
| $TEACHER_1$ | Standard | Yolov5x | Data-1 | Init-1 | 27.01% | 10.74% |
| $TEACHER_2$ | Standard | Yolov5x | Data-1 | Init-2 | 40.86% | 19.32% |
| $TEACHER_3$ | Standard | Yolov5x | Data-1 | Init-3 | 29.20% | 10.75% |
| $TEACHER_4$ | Standard | Yolov5x | Data-1 | Init-4 | 39.82% | 18.00% |
| $TEACHER_5$ | Standard | Yolov5x | Data-2 | Init-1 | 38.34% | 17.63% |
| $TEACHER_6$ | Standard | Yolov5x | Data-2 | Init-2 | 43.32% | **21.65%** |
| $TEACHER_7$ | Standard | Yolov5x | Data-2 | Init-3 | 34.05% | 14.35% |
| $TEACHER_8$ | Standard | Yolov5x | Data-2 | Init-4 | **44.61%** | 21.64% |
| $TEACHER_{OPT}$ | Optimized | Yolov5x | Data-2 | Init-2 | **45.04%** | **22.17%** |

* Data-1: TRAIN only, Data-2: SYNTH and TRAIN

** Init-1: Random, Init-2: COCO pretrained, Init-3: SYNTH pretrained, Init-4: COCO + SYNTH pretrained

From these initial investigations, the following observations can be made. Overall, the addition of synthetic data provides an increase in performance of about six percentage points $mAP_{0.5}$ and about four percentage points $mAP_{0.5:0.95}$ on average. In addition to the training data, the initialization strategy also played a significant role: for example, randomly initialized weights (i.e., Init-1 and Init-3) performed between seven and ten percentage points worse than task-agnostic weights (e.g., pre-trained on COCO). Finally, the impact of using synthetic data for pretraining does not result in significant improvements compared to simply mixing synthetic and natural data (i.e., Data-2). In particular, comparing $TEACHER_2$ with $TEACHER_4$ and $TEACHER_8$ shows a mere difference of approximately one percentage point.

After determining the best training data and initialization strategy, the second step focused on optimizing the training hyperparameters using hyperparameter evolution. In total, 300 generations were evolved in a training time of 14 days on the NVIDIA V100 GPU resulting in a set of optimized hyperparameters. Using the optimized hyperparameters, a teacher model could ultimately be trained to convergence using the $TEACHER_8$ scenario. The $TEACHER_{OPT}$ model shows a slight performance improvement in contrast to the non-optimized model by 0.43 percentage points $mAP_{0.5}$ and 0.53 percentage points $mAP_{0.5:0.95}$.

In summary, two conclusions can be drawn in order to train a capable teacher: (1) using task-agnostic weight initialization provides a significant performance gain. This corresponds with the findings in [12]. And (2) while neither fully-synthetic training nor using the synthetic data for pretraining does provide a performance gain, simply mixing the same synthetic data with natural training data has a net positive effect on detection performance.

### 5.2 SELF-TRAINING VS. SYNTHETIC SELF-TRAINING

After training a well-performing teacher, we now study the use of synthetic self-training and compare it to conventional self-training. For this purpose, we started by training two student generations in a self-training manner *without* adding synthetic data. $STUDENT_1$ was trained on components *TRAIN* and *TRAIN (UNLABELLED)* under the supervision of $TEACHER_4$. Then, $STUDENT_2$ was trained under the supervision of $STUDENT_1$ on the *TRAIN*, *TRAIN (UNLABELLED)* and *WEB (UNLABELLED)* data. For comparison purposes, three student generations were subsequently trained using the proposed synthetic self-training approach. $STUDENT_{1,\,SYNTH}$ was trained on the components *TRAIN*, *TRAIN (UNLABELLED)*, and *SYNTH* under the supervision of $TEACHER_{OPT}$. Subsequently, this model was again used for supervision of the second student generation $STUDENT_{2,\,SYNTH}$, trained on the entire dataset. Finally, $STUDENT_{3,\,SYNTH}$ was trained on the same components under supervision of an ensemble of the previous students. In all settings, test-time augmentation was used to improve pseudo-labelling performance. Only pseudo-labels with a confidence greater than 0.7 were used as labels. All models were trained with the same hyperparameter configurations. Training these models is compute intensive: The inference for one image takes 0.085 seconds, resulting in a pseudo-labeling time for *TRAIN (UNLABELLED)* of 0.8 hours and for *WEB (UNLABELLED)* of almost 12.4 hours. In addition, training a model on the *TRAIN* and *TRAIN (UNLABELLED)* component took approximately six hours, and adding *WEB (UNLABELLED)* increased training time (for 30 epochs) to approximately 4.3 days. The temporal influence of the *SYNTH* component is negligible due to its comparatively small size. Table 2 shows the model performance on the LOCO validation set.

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

*Table 2. Synthetic Self-Training achieves higher detection performance compared to conventional Self-Training approach*

| | | Training data | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Supervision | train | trainun | webun | synth | $mAP_{0.5}$ | $mAP_{0.5:0.95}$ |
| **Self-Training** | | | | | | | |
| $TEACHER_4$ | - | ✓ | x | x | x | 40.86% | 19.32% |
| $STUDENT_1$ | $TEACHER_4$ | ✓ | ✓ | x | x | 43.39% | 20.91% |
| $STUDENT_2$ | $STUDENT_1$ | ✓ | ✓ | ✓ | x | 46.41% | 23.15% |
| **Synthetic Self-Training** | | | | | | | |
| $TEACHER_{OPT}$ | - | ✓ | x | x | ✓ | 45.04% | 22.17% |
| $STUDENT_{1,SYNTH}$ | $TEACHER_{OPT}$ | ✓ | ✓ | x | ✓ | 45.36% | 23.16% |
| $STUDENT_{2,SYNTH}$ | $STUDENT_{1,SYNTH}$ | ✓ | ✓ | ✓ | ✓ | 51.75% | 27.52% |
| $STUDENT_{3,SYNTH}$ | ($STUDENT_{1,SYNTH}$, $STUDENT_{2,SYNTH}$) | ✓ | ✓ | ✓ | ✓ | 52.24% | 28.25% |

From these experiments, the following observations can be derived. In general, self-training on the unannotated data of the LOCO dataset allows for an improvement in detection performance. Using the *WEB (UNLABELLED)* and *TRAIN (UNLABELLED)* components over two self-training iterations provides a significant performance improvement of 5.55% $mAP_{0.5}$ and 3.83% $mAP_{0.5:0.95}$. Using the same non-annotated data, synthetic self-training outperforms conventional self-training. Specifically, there is a 6.71% increase in $mAP_{0.5}$ and a 5.35% increase in $mAP_{0.5:0.95}$. Interestingly, the synthetic data thus helps *not only* in training a teacher, but additionally allows for an increase in self-training performance with the addition of non-annotated data. A third iteration by a teacher ensemble again slightly increases these values. A closer look at the precision-recall curves (see Figure 3) for individual object classes reveal the benefits of SST: Due to the underrepresentation of forklifts and pallet trucks in the TRAIN component, models trained without synthetic images perform comparatively poor on aforementioned classes. Synthetic Self-Training tackles this problem by allowing to adjust the data distribution towards underrepresented classes. Specifically, within these experiments, using synthetic data improves detection performance of forklifts and pallet trucks by 33%.

Summing up, there are three conclusions to be drawn: (1) Synthetic Self-Training (i.e., incorporating synthetic data into the training process) helps for training a capable teacher and student. (2) Large-scale, domain-specific image data crawled from the web can be used to increase detection performance. (3) Synthetic Self-Training is a simple, scalable, flexible and cost-effective approach to train object detectors in domain-specific applications without requiring a large corpus of annotated data.

### 5.3 KNOWLEDGE COMPRESSION

While within the previous experiments only detection performance was compared, applications might require faster inference speed. Thus, within the final set of experiments we studied the use of generated confident pseudo-labels for knowledge compression. Therefore, this final set of experiments is devoted to study the trade-off between detection performance and inference speed of the Yolov5 detector and the LOCO dataset. These smaller models were trained in a noisy student fashion [7] supervised by the ensemble of $STUDENT_{1,SYNTH}$ and $STUDENT_{2,SYNTH}$ as described before. The results (as indicated in Table 3) show, that different performance-/speed-settings can be achieved.

*Table 3. Performance of knowledge-compressed detectors using Synthetic Self-Training on the LOCO dataset*

| Name | Model | Image size | $mAP_{0.5}$ | $mAP_{0.5:0.95}$ | $t_{inf,V100*}$ |
|---|---|---|---|---|---|
| $mAP_{0.5}$ | $mAP_{0.5:0.95}$ | $t_{inf,V100*}$ | | | |
| $STUDENT_{SMALL}$ | Yolov5s | 640 | 42.7% | 21.3% | 15.5 ms |
| $STUDENT_{MEDIUM}$ | Yolov5m | 640 | 46.3% | 23.5% | 19 ms |
| $STUDENT_{3,SYNTH}$ | Yolov5x | 640 | **53.1%** | **28.5%** | 28.4 ms |

* measured on an Nvidia V100, batch size = 1, precision FP32, incl. NMS

## 6 CONCLUSION

For deep learning to have a broad impact on domain-specific applications, it is important to study data-centric approaches allowing to reduce the amount of manual annotations required while still boosting detection performance. This paper contributes towards this goal by introducing, formalizing and empirically validating the use of synthetically generated data in a self-training fashion referred to as Synthetic Self-Training (SST). With the help of SST, we were able to improve detection performance on the LOCO benchmark by 12% $mAP_{0.5}$ compared to the baseline trained in a supervised fashion. Furthermore, we demonstrate a significant performance boost of approximately 6% $mAP_{0.5}$ compared to conventional self-training on the LOCO benchmark. In addition, we also demonstrate the use of synthetic self-training for knowledge compression allowing to increase inference speed while keeping detection performance constant.

Reducing the amount of supervision required for deep learning will increase the impact of deep learning in various real-world applications. In this respect, Self-Training is a flexible and scalable paradigm to incorporate unlabelled data. To make it even more useful, future work should consider unifying different self-training approaches into an application-driven, self-training framework accompanied by an open-source implementation. This allows combining and testing different self-training features which are currently being introduced in research, including slightly different hyperparameters (e.g., for confidence thresholding or loss regularization), extensions (e.g., Exponential-Moving-Average or Synthetic Self-Training) and model initialization (e.g., task-agnostic vs. randomized) to only name a few. Similar to other contributions that combine implementations of different deep learning models, this framework should define common interfaces, allow the usage of different approaches and parameters and study the impact of novel self-training approaches.

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

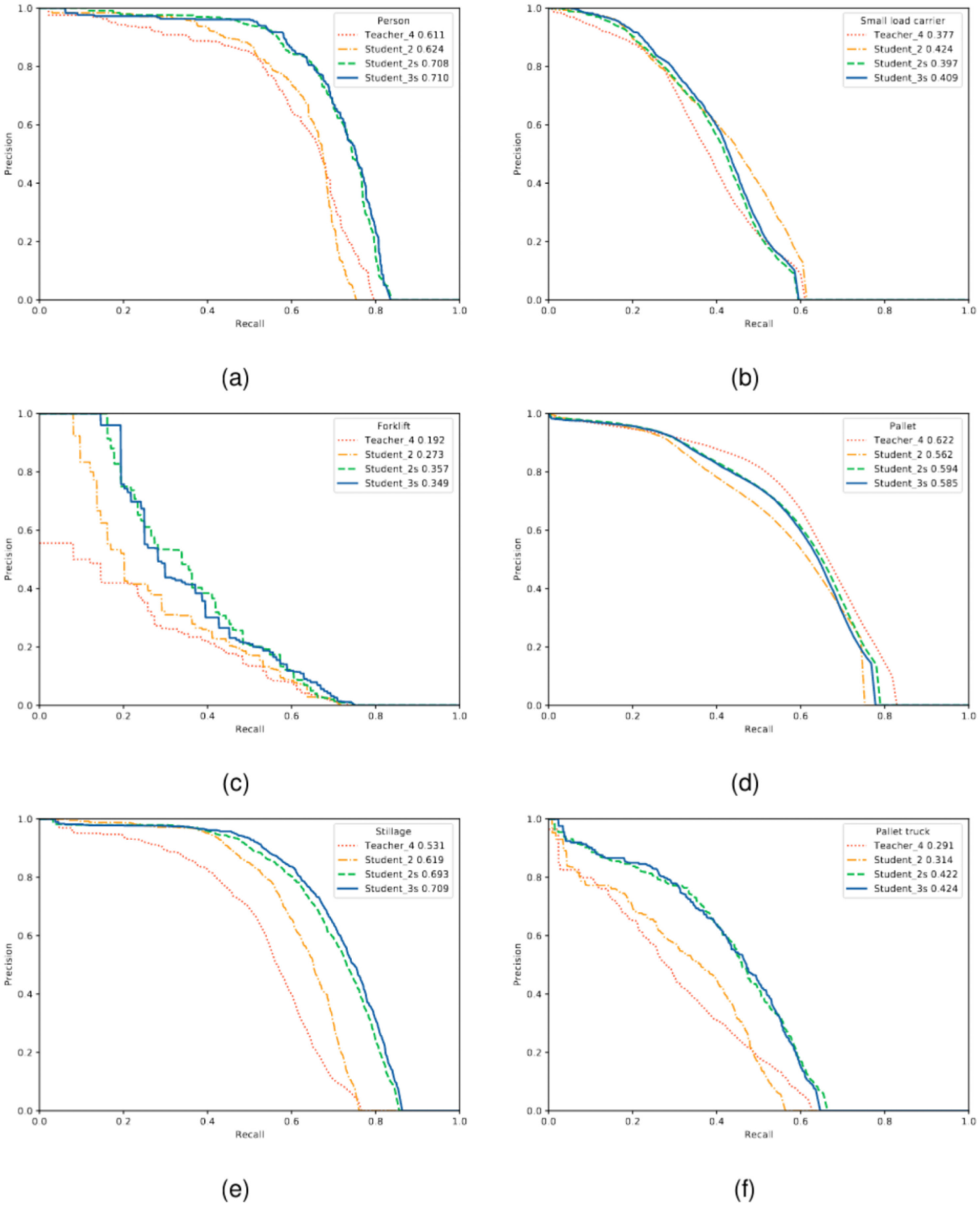



*Figure 3. Precision-recall curves on the LOCO validation set using the baseline model ($TEACHER_4$), the second student generation using conventional self-training ($STUDENT_2$) and the second and third student generation when applying SST ($STUDENT_{2,SYNTH}$ and $STUDENT_{3,SYNTH}$). Each subplot corresponds to a single class, namely person (a), small load carrier (b), forklift (c), pallet (d), stillage (e) and pallet truck (f).*

DOI: 10.2195/lj_Proc_mayershofer_en_202112_01
URN: urn:nbn:de:0009-14-54288

## LITERATURE

[1] C. Mayershofer und J. Fottner, „Towards an Artificial Perception Framework for Autonomous Robots in Logistics," in *Stuttgart Conference on Automotive Production 2020 (SCAP)*, 2020.

[2] C. Mayershofer, A. Hammami und J. Fottner, „Multi-Class Object Detection Using 2D Poses," in *19th IEEE International Conference on Machine Learning and Applications (ICMLA20)*, 2020.

[3] A. Krizhevsky, I. Sutskever und G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, 2012.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg und L. Fei-Fei, „ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV),* pp. 211-252, 2015.

[5] C. Mayershofer, D.-M. Holm, B. Molter und J. Fottner, „LOCO: Logistics Objects in Context," in *19th IEEE International Conference on Machine Learning and Applications (ICMLA20)*, 2020.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll{\'a}r und C. L. Zitnick, „Microsoft COCO: Common Objects in Context," in *Computer Vision -- ECCV 2014*, 2014, pp. 740-755.

[7] Q. Xie, E. Hovy und M.-T. d. Luong, „Self-Training With Noisy Student Improves ImageNet Classification," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 10684-10695, 2020.

[8] T. Chen, S. Kornblith, K. Swersky, M. Norouzi und G. E. Hinton, „Big Self-Supervised Models are Strong Semi-Supervised Learners," *ArXiv,* Bd. abs/2006.10029, 2020.

[9] J. Li, C. Xiong und S. Hoi, „CoMatch: Semi-supervised Learning with Contrastive Graph Regularization," *ArXiv,* Bd. abs/2011.11183, 2020.

[10] H. H. Pham, Q. Xie, Z. Dai und Q. V. Le, „Meta Pseudo Labels," *ArXiv,* Bd. abs/2003.10580, 2020.

[11] D. Lee, „Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," 2013.

[12] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk und Q. V. Le, „Rethinking pre-training and self-training," *arXiv preprint arXiv:2006.06882,* 2020.

[13] J. E. Van Engelen und H. H. Hoos, „A survey on semi-supervised learning," *Machine Learning,* Bd. 109, p. 373–440, 2020.

[14] P. Bachman, O. Alsharif und D. Precup, *Learning with Pseudo-Ensembles,* 2014.

[15] J. von Kugelgen, Y. Sharma, L. Gresele, W. Brendel, B. Scholkopf, M. Besserve und F. Locatello, „Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style," 2021.

[16] Q. Xie, E. Hovy, M.-T. Luong und Q. V. Le, „Self-Training With Noisy Student Improves ImageNet Classification," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 10684-10695, 2020.

[17] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira und P. Vajda, „Unbiased teacher for semi-supervised object detection," *arXiv preprint arXiv:2102.09480,* 2021.

[18] K. Sohn, D. Berthelot, C. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang und C. Raffel, „FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence," *ArXiv,* Bd. abs/2001.07685, 2020.

[19] G. J. McLachlan, „Iterative Reclassification Procedure for Constructing an Asymptotically Optimal Rule of Allocation in Discriminant Analysis," *Journal of the American Statistical Association,* Bd. 70, pp. 365-369, 1975.

[20] C. Mayershofer, T. Ge und J. Fottner, „Towards Fully-Synthetic Training for Industrial Applications," in *10th International Conference on Logistics, Informatics and Service Sciences (LISS)*, 2020.

[21] Z. Zheng, L. Zheng und Y. Yang, „Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of*

*the IEEE international conference on computer vision*, 2017, pp. 3754-3762.

[22] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar und others, „Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733,* 2020.

[23] C. Rosenberg, M. Hebert und H. Schneiderman, „Semi-supervised self-training of object detection models," 2005.

[24] H. Schneiderman, „Learning a restricted Bayesian network for object detection," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004.

[25] H. Schneiderman, „Feature-centric evaluation for efficient cascaded object detection," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004.

[26] J. Jeong, S. Lee, J. Kim und N. Kwak, „Consistency-based Semi-supervised Learning for Object detection," in *NeurIPS*, 2019.

[27] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee und T. Pfister, „A simple semi-supervised learning framework for object detection," *arXiv preprint arXiv:2005.04757,* 2020.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He und P. Dollár, „Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[29] M. Pérez-Ortiz, P. Tiňo, R. K. Mantiuk und C. Hervás-Martínez, „Exploiting Synthetically Generated Data with Semi-Supervised Learning for Small and Imbalanced Datasets," in *AAAI*, 2019.

[30] J. Devlin, M.-W. Chang, K. Lee und K. Toutanova, „Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[31] W. Jiao, X. Wang, Z. Tu, S. Shi, M. R. Lyu und I. King, *Self-Training Sampling with Monolingual Data Uncertainty for Neural Machine Translation,* 2021.

[32] J. Mu, W. Qiu, G. D. Hager und A. L. Yuille, „Learning From Synthetic Animals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] U. Bertram, M. Bergmann, P. Hartinger, P. Steger und C. null.

[34] D. Shanmugam, D. Blalock, G. Balakrishnan und J. Guttag, „When and Why Test-Time Augmentation Works," *arXiv preprint arXiv:2011.11156,* 2020.

[35] Á. Casado-García und J. Heras, „Ensemble methods for object detection," in *ECAI 2020*, IOS Press, 2020, p. 2688–2695.

[36] M. A. Ganaie, M. Hu und others, „Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395,* 2021.

[37] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba und P. Abbeel, „Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23-30.

[38] S. Hinterstoisser, O. Pauly, H. Heibel, M. Marek und M. Bokeloh, „An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection," 2019.

---

**Christopher Mayershofer, M.Sc.,** Head of Engineering, Noyes Technologies GmbH. Work done during his time as a Research Associate at the Chair of Materials Handling, Material Flow, Logistics, Department of Mechanical Engineering, Technical University of Munich.

**Adrian Fischer, B.Sc.,** Research Assistant at the Chair of Materials Handling, Material Flow, Logistics, Department of Mechanical Engineering, Technical University of Munich.

**Prof. Dr.-Ing. Johannes Fottner,** Full Professor at the Chair of Materials Handling, Material Flow, Logistics, Department of Mechanical Engineering, Technical University of Munich.

Address: Chair of Materials Handling, Material Flow, Logistics, Department of Mechanical Engineering, Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany,
Phone: +49 089 289-15921, Fax: +49 089 289-15922, E-Mail: christopher.mayershofer@tum.de