Towards Fully-Synthetic Training for Industrial Applications

Christopher Mayershofer
Chair of Materials Handling, Material Flow, Logistics
Technical University of Munich
Garching, Germany
christopher.mayershofer@tum.de

Tao Ge
Chair of Materials Handling, Material Flow, Logistics
Technical University of Munich
Garching, Germany
tao.ge@tum.de

Johannes Fottner
Chair of Materials Handling, Material Flow, Logistics
Technical University of Munich
Garching, Germany
j.fottner@tum.de

***Abstract*—This paper proposes a scalable approach for synthetic image generation of industrial objects leveraging Blender for image rendering. In addition to common components in synthetic image generation research, three novel features are presented: First, we model relations between target objects and randomly apply those during scene generation (Object Relation Modelling (ORM)). Second, we extend the idea of distractors and create Object-alike Distractors (OAD), resembling the textural appearance (i.e. material and size) of target objects. And third, we propose a Mixed-lighting Illumination (MLI), combining global and local light sources to automatically create a diverse illumination of the scene. In addition to the image generation approach we create an industry-centered dataset for evaluation purposes. Experiments show, that our approach enables fully synthetic training of object detectors for industrial use-cases. Moreover, an ablation study provides evidence on the performance boost in object detection when using our novel features.**

***Keywords*— Object detection, Synthetic data, Domain randomization**

I. Introduction

In recent years, machine learning methods have gained increasing attention. Particularly, supervised learning using deep neural networks solves previously insoluble problems. These developments are especially apparent in the field of computer vision: convolutional neural networks (CNNs) enable object detection [1]–[4] and segmentation [5], [6], as well as pose [7], [8] and depth estimation [9], [10]. A decisive factor for the success of these networks is the existence of large amounts of annotated image data.

While the global research community has already published many diverse datasets, there are still use cases for which no or insufficient data is available. In the public sector in particular, there are large datasets available that deal with autonomous driving [11], [12], common objects [13], [14], or famous landmarks [15], to only name a few. On the other hand, datasets within the industrial domain are rarely found, as few images are published. Accordingly, for many industrial applications, realistic datasets must first be collected and annotated before applying them to specific use-cases. Collecting and annotating a dataset is a very money and time consuming endeavour [16].

This work was supported by the BMW AG (BMW Group).

Fig. 1. Prediction results of a fully-synthetic trained object detector. The detection model is trained on *synthetic image data only* using the proposed image generation approach. To meet industrial requirements, we implement novel features to represent specific object relations, to suppress false detections and to model complex industrial lighting conditions while maintaining a maximum level of scalability.

The time required for annotating images depends on the specific application. For an image-level classification task, for example, annotation can be done rather quickly, as it means that each image to be used for training needs to be assigned to a specific class. The more complex the task, the greater the annotation effort; for example, within object detection, in addition to the classification of an object, its position in the image plays a decisive role. Depending on the desired level of detail, the localization can be done using bounding boxes (low level of detail resulting in low annotation effort) or segmentation masks (high level of detail resulting in high annotation effort).

To solve this challenge in a long-lasting manner, different methods were proposed to synthetically create image data in order to train deep neural networks. In contrast to generating training datasets using natural images, artificial images can be created automatically with very precise annotations (i.e. bounding box, per-pixel depth, object pose, object segmentation, etc.) at negligible cost. However, the existing domain

gap between synthetic and natural images makes the neural network trained on synthetic images perform poorly on natural images [17]. Various methods have been introduced in an attempt to close this gap. Yet, there is no universal approach known to resolve this issue. Therefore, within the scope of this paper we have investigated synthetic image generation for industrial applications.

In particular we present a scalable, Blender-based image generation approach, that enables fully-synthetic training of object detectors used in industrial applications (see Fig. I).

The scientific contributions of this paper can be summarized as follows:

1) **Scalable synthetic image generation approach.** We present a scalable image generation approach adopting well-working methods and augmenting them with novel features such as Object-alike Distractors (OADs), Object Relation Modelling (ORM) and a Mixed-lighting Illumination (MLI).
2) **Industry-centered evaluation dataset.** We present a natural image dataset enabling the evaluation of synthetic image generation approaches for industrial objects. Our dataset contains realistic images covering a single industrial object (small load carrier) in a close-to-industry environment facing multiple industry-relevant challenges (e.g. different lighting conditions, multiple objects, object relations).
3) **Extensive ablation study.** We evaluate our approach on the previously mentioned dataset providing realistic performance feedback. Furthermore, we provide insights into the different aspects of our approach by ablating features and showcasing their performance boost.

II. Related Work

Generating artificial training data is gaining popularity in computer vision research. Reference [18] created training images by cutting images of target objects from other datasets and pasting them on background images. Although their method ensures patch-level realism, it requires plenty of real images with considerable human efforts in segmenting out objects. Others are using computer graphics render engines to generate synthetic training images, relying on an elaborately, manually-created and close-to-real-world scene [19], [20]. Even though synthetically generated images can appear photo-real to us humans, deep neural networks still show problems when being transferred from the simulation to the real world. The domain gap between the simulated training images and the real world experiments might be due to the fact that most render engines are built to leverage the human perception system in order to efficiently create images that appear to be photo-real [21].

Domain adaptation (DA). One way to overcome this gap is to adapt one area to another. The basic idea of DA is to transfer a model trained in the source domain to the target domain [22]. Among all different DA methods, semi-synthetic training is a simple and effective method [23]. Training the neural network on a large synthetic dataset first and fine-tuning it on limited data from the target domain afterwards boosts the performance of neural networks [16], [24], [25]. Furthermore, generative adversarial networks can be applied to achieve domain adaptation [26]. Differently, [27] focused on image translation of synthetic images. They created a generator to translate both synthetic and real images to canonical images. The application only needed to handle canonical images and never got in contact with 'raw' synthetic images. Although DA can improve neural network performance, the deficiencies are apparent. Firstly, it is inevitable to use data from the target domain. However, one of the main reasons to generate synthetic training data in the first place is to avoid the usage (and need) of real training data. Secondly, domain adaptation may improve the performance of the neural network on images from the target domain. However, when it is tested on other domains (even the source domain), its performance degrades significantly [25].

Domain randomization (DR). A second approach towards overcoming the sim2real gap is by applying DR. The general idea behind DR is that by randomizing parameters in the source domain (i.e. simulation), the target domain (i.e. real world) appears to the neural network as just another variation of the source domain [21], [28], [29]. In practice, different parameters of objects, background, camera and lights are randomized in the synthetic image generation process [24], [30]. Besides, [29], [31] imported random distractors to create random occlusion and prevent the neural network from detecting distractors in the real world. Reference [20] proposed structured domain randomization which takes the context into consideration resulting in high recall performance. Moving from color images to depth images, [28] proposes synthetic depth data randomization to generate depth images for training. Although their neural network trained only on synthetic depth images outperforms the detector trained on real data, the method is limited to depth images. Instead of generating images using modeled scenes or images as background, [21] randomly fills the background with plenty of objects in order to prevent the neural network from learning a certain pattern in the background. Their experiments demonstrate the effect of a randomized background generation.

Guided domain randomization (GDR). As an improvement to general DR, many guided DR methods were proposed. Reference [32] developed active domain randomization, which searches for the most informative environment variations by measuring the discrepancies between the randomized and reference environments. Environments with high informativeness can then increase the difficulties of training in order to improve the performance. Although active domain randomization achieves better results compared to general DR methods, the correlation between discrepancy of environments and training difficulty is still unknown. Similarly, [33] proposed the automatic domain randomization approach to increase the difficulty during the training process. This is achieved by automatically and gradually expanding the distribution over environments, helping to improve prediction accuracy but also significantly

increasing the process duration.

In summary, different approaches for generating synthetic data are known. Each approach proposes novel features, which in turn are being utilized in the next iteration of synthetic image generation approaches. Since the industrial application of synthetic image data has hardly been researched so far, this paper presents our approach towards synthetic image generation for industrial applications.

III. Synthetic Image Generation of Industrial Objects

We propose a scalable image generation approach for industrial objects using computer graphics. Specifically, we render images from automatically generated 3d scenes. Hereby, we adopt well-working methods and augment them with novel features such as *Object-alike Distractors (OADs)*, *Object Relation Modelling (ORM)* and a *Mixed-lighting Illumination (MLI)*. Rendering is based on the open-source 3D creation suite Blender. Our image generation approach is visualized in Fig. 2 and can be divided into three process steps, namely background creation, foreground creation and rendering. All process steps are subject to domain randomization in order to reduce the resulting domain gap. Building on top of 3D modeled objects, we are able to automatically generate annotated training data for different computer vision tasks with varying complexity. The following section describes each process step in detail.

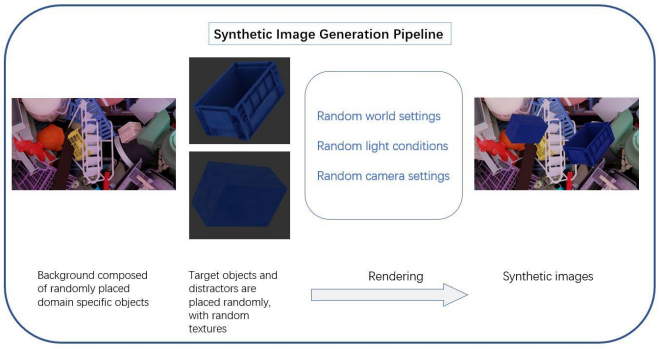


Fig. 2. Synthetic image generation approach for industrial objects. Using 3D modeled objects, we automatically generate a 3D scene and render images from it. Wherever possible we are using domain randomization to decrease the domain gap.

A. Background Generation

First, the background of the 3D scene is created automatically. We mostly adopt the background generation process described in [21]. In a nutshell, [21] is forming the background by using a multitude of 3D objects and randomly positioning them in the background plane.

As we found that using many 3D objects for background generation increases the computational effort and reducing the amount of 3D objects resulted in spots without any object, we additionally load and place a random image in the background plane. The voids among these objects are then filled by the loaded image. This simple measure provides a balance between the level of clutter and the computational cost for image generation.

B. Foreground Generation

Second, we automatically create the foreground consisting of randomized target object(s) and distractor(s) within the cameras view space. Fig. 3 visualizes an automatically generated, pyramid-like 3D scene in Blender.

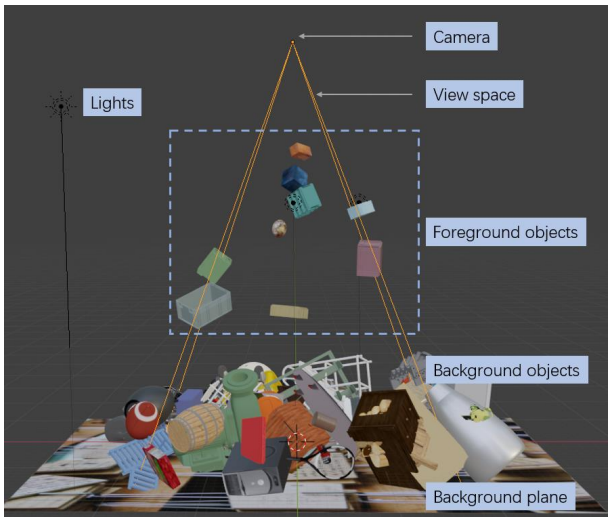


Fig. 3. Automatically generated 3D scene. Our approach creates the background using randomly placed background objects and a background plane, places target objects and distractors within the pyramid-like camera view space and creates the Mixed-lighting Illumination. Next, the 3D scene is transformed to the image space using one of Blenders' render engines.

Similar to known approaches, we randomize the number, type, location, rotation and material of target objects. In addition to that, we propose *Object Relation Modelling (ORM)*, a novel feature applying predefined relations between target objects when placing them in 3D space. In order to do so, relation files are created manually by recording the relative translation and rotation of one target object to another before the image generation. This spatial relation can later be randomly applied during foreground generation. If an intersection is detected when applying a relation, the related object will be deleted. Since ORM is based on coordinate transformation, it is applicable to describe any relation amongst objects. ORM intentionally increases the probability of certain spatial relations which could hardly be achieved by chance within the synthetic dataset. As expected, our experiments show that this increases detection performance when facing those relations in natural images.

In the fashion of [29] and [31], we place distractors in our scenes. Distractors are random geometries in the forground creating occlusions and 'distracting' the detector to be trained. The distractors are automatically created and placed in the scene with random geometry, position, orientation, scale and material. Additionally, we extend the idea of general distractors and implement so called *Object-alike Distractors (OADs)*.

Again, OADs are randomly generated basic geometries, but in contrast to standard distractors they share the same size and material as the target object, causing the detector to focus on structural features of the target objects geometry rather than much simpler textural cues of a certain material.

C. Lighting and Camera

In addition to the previously mentioned steps, lighting as well as camera settings are subject to DR. The lighting condition within training data significantly affects the performance of neural networks [21], [31], [34]. In contrast to other approaches that create a single domain-randomized light source we propose a *Mixed-lighting Illumination (MLI)* that divides illumination into a global and a local component. Global lighting illuminates the entire scene whereas local lighting creates specific highlights on random positions within the cameras view space.

Global illumination consists of on the one hand passive illumination from the set environment texture and on the other hand a global light source that illuminates the entire scene. This global light source is placed randomly on a projected hemisphere. In contrast, local light sources are only placed within the pyramid-like camera view space. The environment texture as well as the global and local light source properties (i.e. location, color, energy, size) are all subject to domain randomization. Fig 4 showcases the variance in illumination; (a) shows a brightly illuminated scene with a soft greenish-turquoise hue, whereas (b) is a much darker lit scene, that puts the focus on the yellow sphere-shaped geometry due to local spotlights automatically generated by MLI.

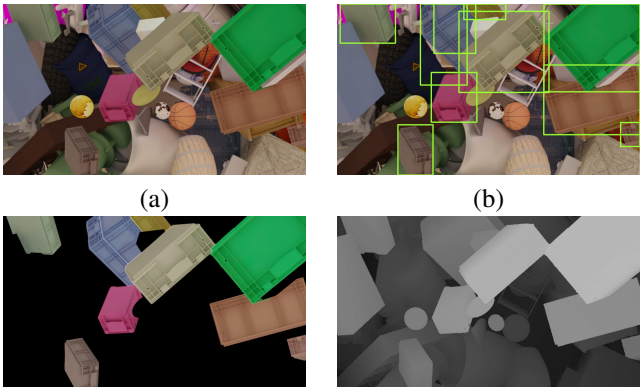
(a) (b)

Fig. 4. Mixed-lighting illumination. Two automatically generated sample images using the mixed-lighting illumination approach. Image (a) consists of a bright scene with greenish-turquise hue, whereas (b) is a rather dark scene with a high contrast created due to the local light sources.

Also the cameras properties are subject to domain randomization. In contrast to other approaches, we only randomize the camera z-position to vary the distance from camera to object. Furthermore, we randomize the focus distance, f-stop and aperture blades in order to generate images with random out-of-focus blur.

D. Annotations

One of the main advantages of synthetic image generation is that creating annotations can be performed automatically. In addition to rendering images, we also generate a variety of annotations useful for different computer vision tasks. Our system currently generates ground truth data in the form of bounding boxes, segmentation masks as well as depth maps (as illustrated in Fig. 5).



(a) (b) (c) (d)

Fig. 5. Ground-truth generation. For each rendered image (a), our system automatically generates bounding boxes (b), panoptic segmentation masks (c) as well as depth maps (d).

IV. Experiments

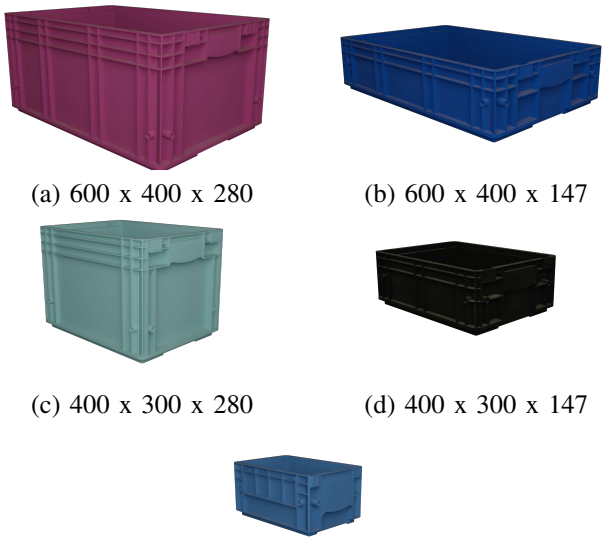
We now present experiments of an object detection model trained only on synthetically generated images using the proposed approach. Furthermore, an ablation study was conducted and provides detailed insights into the design principles for reducing the domain gap featured in this novel approach.

In order to do so, we chose a single object class, namely small load carriers (SLCs), because they are widely used in industry, they are standardised and are also subject to automated material flow handling by robots in the future. We believe that synthetic image generation techniques for use in logistics could allow such applications. In sum, SLCs are used for transporting materials and are standardized by the German Association of the Automotive Industry (VDA). Furthermore SLCs are available in different sizes as illustrated in Fig. 6 which can be consolidated and stacked fitting onto a pallet (1.200 mm x 800 mm). In the context of our experiments, we consider load carrier types VDA RL-KLT 3147, 4147, 4280, 6147 and 6280.

A. Industry-centered Evaluation Dataset

In order to investigate the different characteristics of our method in a systematic and controlled way an evaluation dataset with natural images was created (see Fig. 7). The evaluation dataset was recorded at the chair's research facility, resembling a realistic industrial environment. Hereby special emphasis was put on various aspects of capturing known influences in the industrial environment. The evaluation dataset contains images under different lighting conditions, with varying number of objects, different object sizes, different distances to the target objects as well as logistically specific states of the SLCs such as loaded as well as stacked load carriers. In total, the dataset consists of 1460 manually annotated images, which can be divided into seven categories:

- single-object images
- multi-object images
- images with small object instances ($<$1 % image area)



(a) 600 x 400 x 280 (b) 600 x 400 x 147

(c) 400 x 300 x 280 (d) 400 x 300 x 147

(e) 300 x 200 x 147

Fig. 6. Small load carriers and their respective size (length x width x height) in millimeters. VDA RL-KLT 6280 (a), 6147 (b), 4280 (c), 4147 (d) and 3147 (e) are standardized by the Association of the Automotive Industry (VDA) and can be found in logistics throughout different sectors.

- images with medium object instances (between 1-10 % image area)
- images with large object instances (>10 % image area)
- loaded SLCs
- stacked SLCs

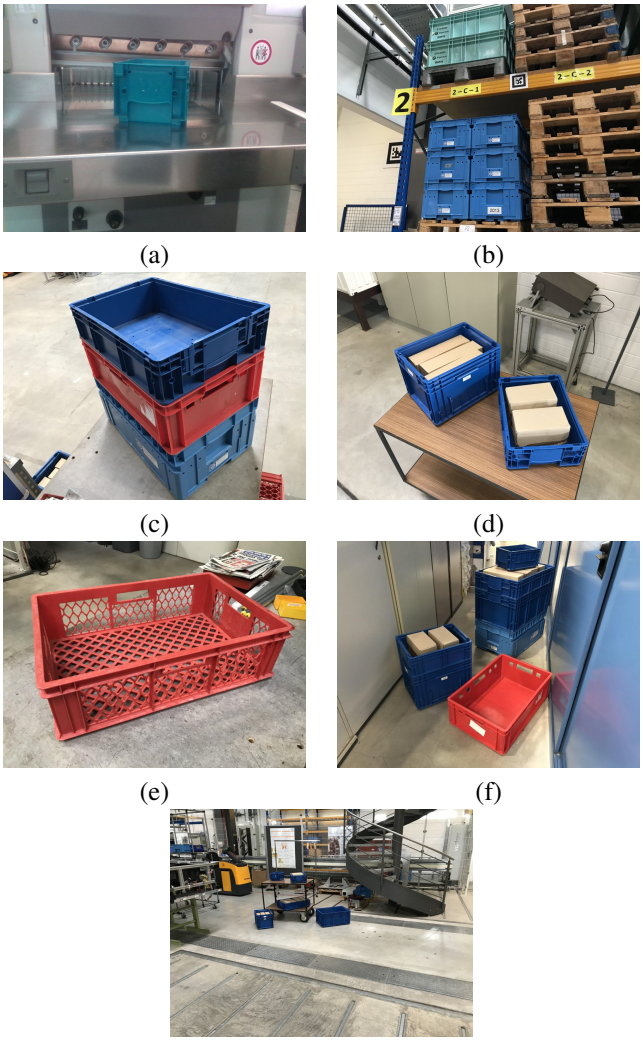
B. Model Training

We evaluate our method by training a deep neural network object detector. The darknet framework and in particular the YOLOv3 model [35] was chosen for object detection. The model is trained exclusively on synthetic data and evaluated on natural test images. For training a pretrained feature extractor is used. From a network structure perspective, only the number of convolutional filters is adjusted to accommodate the single object category. Other parameters as well as the augmentation strategy remain unchanged. The model was trained on a Nvidia V100 GPU.

C. Evaluation and Ablation Study

Finally we describe the evaluation process and ablate different design choices within our image generation approach. Unless otherwise noted, the training was performed as described in section IV-B. We present the precision as well as recall metric for each of the models at an intersection over union (IoU) of 0.5.

Render engines. Firstly, we compared the effects of images generated using Cycles and EEVEE, two different render engines within Blender. Cycles renders images by tracing back light paths and accumulating them, causing it to be slow, but rather physically correct. EEVEE in contrast is a game-engine and creates images by projecting images from



(a) (b) (c) (d) (e) (f) (g)

Fig. 7. Industry-centered evaluation dataset. In order to evaluate our approach, we captured and annotated a dataset containing images with single (a), multiple (b), stacked (c) and loaded (d) small load carriers. Furthermore, the images can be distinguished by the relative size of the SLC in an image (large (e), medium (f) and small (g)). Note that the classification is not mutually exclusive.

the 3D space to a 2d plane. As these projections do not take into consideration lighting and tracing lightrays, these images look less realistic, but are also generated faster. Fig. 8 presents the test results of YOLOv3 trained on 2.500 images generated by Cycles and EEVEE respectively. Trained on the same amount of images, it can be concluded that the detector trained on images rendered by Cycles outperforms the one that was trained on EEVEE-rendered images. Due to the fact that EEVEE ignores the physical realism, it can render images much faster than Cycles. In our experiments, the time for Cycles to generate 2.500 images is equal to the time for EEVEE to generate 5.700 images using devices with same computational capabilities. Therefore, we further investigated the effect of 2.500 Cycles images and 5.700 EEVEE images. It can be seen that increasing the size of the training set improves

detection performance. In conclusion, it can be summarized, that the information content per synthetic image generated using Cycles is higher. In contrast, EEVEE is able to generate more images in the same amount of time, but these images contain less information useful for training the network. We chose to continue our experiments with Cycles as a render engine (denoted as standard in following figures).

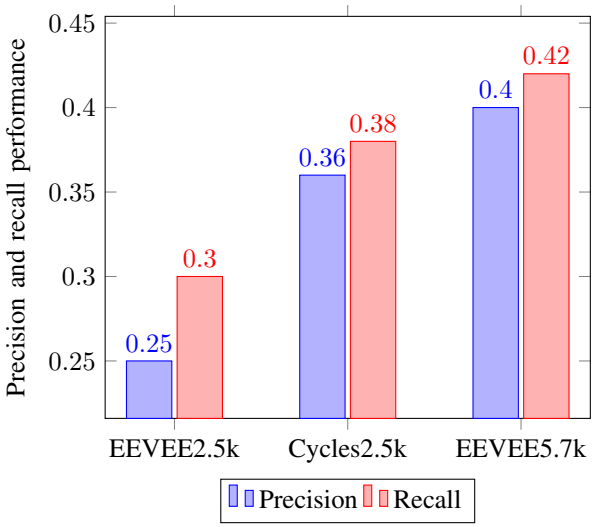


Fig. 8. Ablation study on render engines. Detection performance of a model fully trained on synthetic images rendered by Cycles and EEVEE, two different render engines within Blender.

Background objects and distractors. In the next experiment we evaluated the importance of background objects and distractors in our approach. To do so, two additional image batches were created, one without background objects (but the background plane), and another one without distractors. Fig. 9 illustrates our findings and compares them to the performance of the standard model with background objects and distractor. It can be observed that precision and recall decrease slightly after removing the background objects. This suggests that the background objects are not as effective as in experiments presented by [21]. Besides, this experiment clearly shows the importance of using distractors as the detection performance plummets when removing them.

Object relation modelling. In order to study the effects of ORM in our approach, we used the standard image batch with a single modelled relation and generated two additional image batches; one batch without modelled relations (i.e. objects were placed randomly), and one batch with multiple different relations. Again, the trained detectors were tested on natural images, many of which contain SLC stacks and loaded SLCs. Fig. 10 suggests, that increasing the number of applied relations in synthetic training images improves the performance of the detector. Furthermore, we tested these detectors on two different subsets of our dataset. One subset containing only images with SLC stacks ("stacks" subset) and another subset containing images of loaded SLCs ("loaded" subset). The results are shown in Fig. 11 and indicate a similar

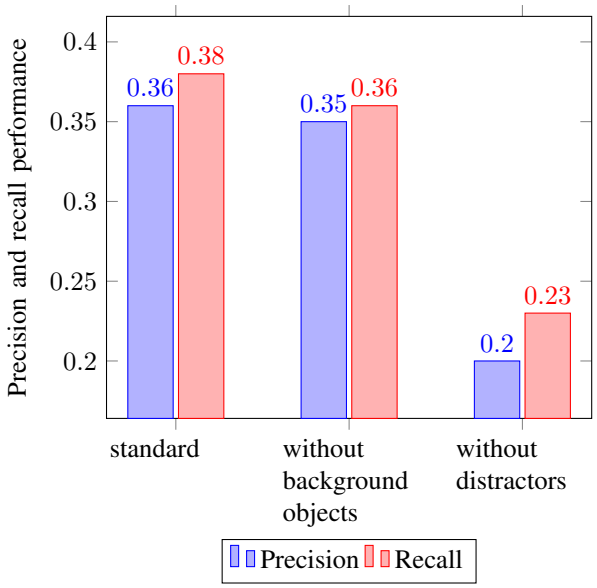


Fig. 9. Ablation study on background objects and distractors. Detection performance of a model fully trained on synthetic images, synthetic images without background objects and synthetic images without distractors.

tendency as shown before. Furthermore, this suggests that ORM has similar effects on both subsets.

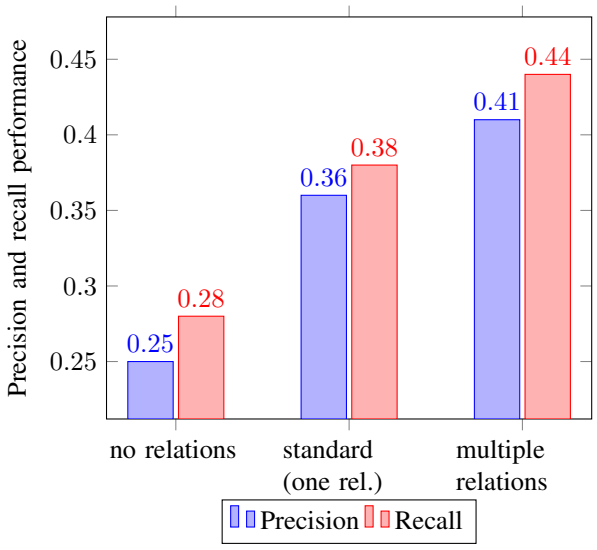


Fig. 10. Ablation study on Object Relation Modelling (ORM). Detection performance of a model fully trained on synthetic images with one modelled relation (standard), synthetic images without ORM and synthetic images with multiple modelled relations.

Object size. Finally, we performed experiments to investigate how the detector trained on synthetic images generated by our method performs on real test images with respect to objects of different size. The results presented in Fig. 12 show that the performance drops prominently as the object size decreases. When detecting large objects, the precision reaches up to 0.81. However, for the challenging detection of small objects, it drops to only 0.11.

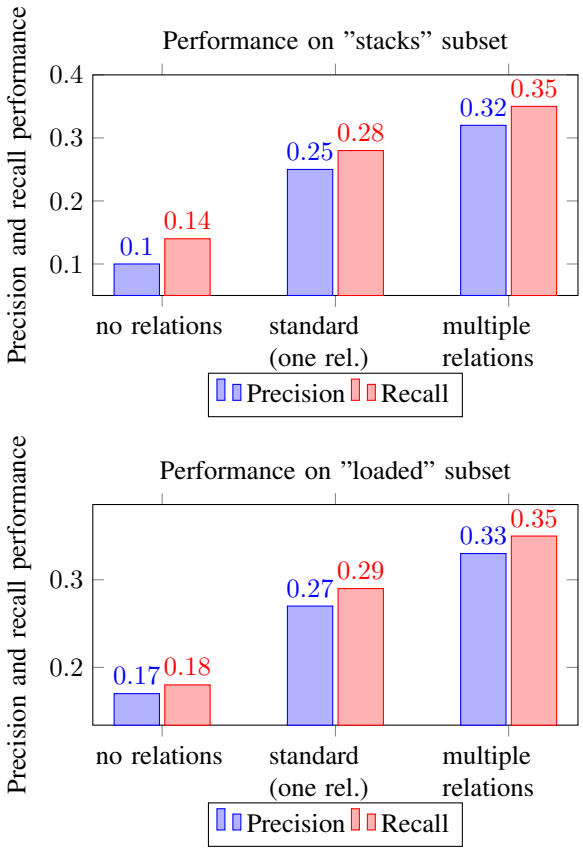


Fig. 11. Ablation study on Object Relation Modelling (ORM) focussing on “stacks” (top) and “loaded” (bottom) subset containing images where understanding the relation between objects is necessary. For each subset, we present the detection performance of a model fully trained on synthetic images with one modelled relation (standard), synthetic images without ORM and synthetic images with multiple modelled relations.

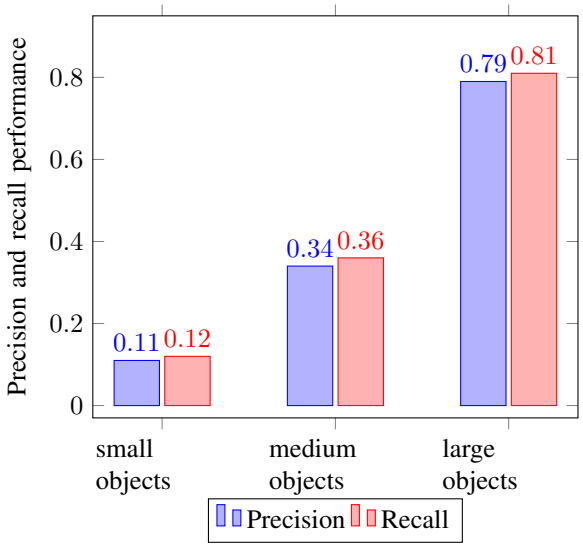


Fig. 12. Ablation study on object size. Detection performance of a model fully trained on synthetic images with one relation (standard) and synthetic images with more relations tested on subsets of our dataset containing large, medium and small objects.

V. Conclusion

In this paper we have presented a scalable approach for synthetic image generation of industrial objects utilizing well-working features and augmenting them with novel components, such as Object-alike Distractors (OAD), Object Relation Modelling (ORM) and a Mixed-lighting Illumination (MLI). Due to missing industrial datasets, we generated and presented a industry-centered dataset for evaluation purposes of synthetic image generation methods. Finally, an extensive ablation study is presented, wrapping up our experiments.

Most importantly, we show, that our approach enables fully synthetic training for object detection in industry. Still, in its current state, it is limited to certain boundaries and the detection performance is worse compared to detectors trained on natural images only. Furthermore, we show that our novel features (Object Relation Modelling, Object-alike Distractors and Mixed-lighting Illumination) are simple, effective and scalable (i.e. are able to be automated) methods to consider when developing synthetic image generation methods. They work by changing the statistics of certain features within the synthetic dataset and ’guiding’ the detection model to focus on these features. All of this, whilst still being scalable, without the need of manually modelling different 3D scenes. Finally, we found that backwards ray-traced rendering increases the information entropy within synthetic datasets, suggesting that physical correctness is important for current convolutional neural networks.

For future research we plan to extend our approach and further analyse Blenders capabilities as a synthetic image generation system. This requires additional experiments with different object classes in diverse industrial applications to prove the generalization and robustness of our method. Furthermore, we will be expanding our tests to other network architectures as well as computer vision tasks in order to investigate the domain gap in different tasks. Finally, the domain gap is still apparant, so future research needs to focus on minimizing it.

Acknowledgement. We thank the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW) for the provision and support of cloud computing infrastructure essential to this publication. All support is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the writers and do not necessarily reflect the views of the BMW Group or the LRZ.

References

[1] S. Ren, K. He, R. B. Girshik, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” 2015.
[2] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” 2016.
[3] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2017.
[4] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” 2019.
[5] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” 2017.

[6] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "FasterSeg: Searching for faster real-time semantic segmentation," 2020.

[7] Z. Su, M. Ye, G. Zhang, L. Dai, and J. Sheng, "Cascade feature aggregation for human pose estimation," 2019.

[8] A. Bulat, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "Toward fast and accurate human pose estimation via soft-gated skip connections," 2020.

[9] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 2016.

[10] J. H. Lee, M. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," 2013.

[12] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," 2019.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," 2010.

[14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollr, "Microsoft coco: Common objects in context," 2014.

[15] H. Noh, A. Araujo, J. Sim, and B. Han, "Image retrieval with deep local features and attention-based keypoints," 2016.

[16] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. Al Hassanat, R. Laganiere, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," 2019.

[17] X. Peng and K. Saenko, "Synthetic to real adaptation with deep generative correlation alignment networks," 2017.

[18] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," 2017.

[19] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" 2017.

[20] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," 2018.

[21] S. Hinterstoisser, O. Pauly, H. Heibel, M. Marek, and M. Bokeloh, "An annotation saved is an annotation earned: Using fully synthetic training

[32] B. Mehta, M. Diaz, G. Florian, C. J. Pal, and L. Paull, "Active domain randomization," 2019.

for object instance an annotation saved is an annotation earned: Using fully synthetic training for object instance detection," 2019.

[22] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," 2018.

[23] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3d pose inference from synthetic images," 2018.

[24] J. Borrego, A. Dehban, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Applying domain randomization to synthetic data for object category detection," 2018.

[25] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," 2018.

[26] G. Yang, H. Xia, M. Ding, and Z. Ding, "Bi-directional generation for unsupervised domain adaptation," 2020.

[27] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim data-efficient robotic grasping via randomized-to-canonical adaptation networks," 2019.

[28] S. Thalhammer, K. Park, T. Patten, M. Vincze, and W. Kropatsch, "Sydd synthetic depth data randomization for object detection using domain-relevant background," 2019.

[29] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," 2017.

[30] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, vol. 126, no. 9, pp. 942–960, 2018.

[31] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," 2018.

[33] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," 2019.

[34] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?" 2016.

[35] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.