Bridging the Reality Gap in Industrial Synthetic Datasets with Aging Texture Simulation

1st Joe Khalil*
Université de Franche-Comté
BMW Group
Munich, Germany
joe.khalil@bmw.de

2nd Jimmy Tekli
BMW Group
Munich, Germany
jimmy.tekli@bmwgroup.com

3rd Angelo Yaghi
BMW Group
Munich, Germany
angelo.yaghi@bmw.de

4th Marc Kamradt
BMW Group
Munich, Germany
marc.kamradt@bmw.de

5th Najib Metni
Denali AI
London, United Kingdom
nmetni@denaliai.com

6th Raphaël Couturier
Université de Franche-Comté
CNRS, institut FEMTO-ST
F-90000 Belfort, France
raphael.couturier@univ-fcomte.fr

***Abstract*—As industries increasingly rely on computer vision applications, the demand for robust object detection models grows. Synthetic datasets offer a promising solution by providing diverse and accurate data scenarios, enhancing the performance of these models. In this study, we tackle one aspect of the ongoing challenge to bridge the reality gap between synthetic and real images in industrial applications. We incorporated textures that accurately replicate signs of usage on industrial assets. Furthermore, we conducted a comprehensive experiment featuring three distinct datasets from the same simulated environment to evaluate the impact of texture variations on model performance. Our approach introduces a combination of aging texture layers to six industrial assets for object detection. To assess performance, we trained three Deep Learning architectures with these datasets and evaluated them by inferring on real images captured in industrial settings. The results of our experiments demonstrate the effectiveness of the aging textures in real-world scenarios, affirming the success of our proposed approach in enhancing object detection models for industrial applications.**

***Index Terms*—Synthetic Data, Computer Vision, Industry 4.0, Object Detection, and Reality Gap**

I. Introduction

Deep learning (DL) revolutionized various fields of computer vision, and one prominent application is object detection. Object detection plays a vital role in numerous industrial applications, including quality checks throughout assembly lines [12], autonomous systems [8], robotics [14], and others. The effectiveness and accuracy of these DL models rely heavily on the availability of large-scale, diverse, and accurately annotated images [4]. Nevertheless, capturing such images in an industrial setting poses several challenges [2], [4], such as (i) limited access to highly secure areas, (ii) strict safety and privacy regulations [28], [29], and (iii) potential operational disruptions. In summary, the data collection process is difficult, time-consuming, and costly.

To address these challenges, synthetic datasets emerged as a promising solution [18]. By leveraging computer graphics techniques and machine learning algorithms, synthetic image generation addresses the above challenges by producing labeled images with accurate annotations and well-defined properties, e.g. spatial configurations, lighting, textures, camera perspectives, object interactions, and environmental conditions. In recent years, numerous synthetic large-scale datasets with diverse object categories were proposed in an attempt to overcome the limitations of real-world data collection in industrial settings, such as the Synthetic Object Recognition Dataset for Industries [2] and many others [6], [25], [26], [31].

One of the main challenges of the synthetic image generation process is bridging the reality gap. The reality gap represents the difference between carefully controlled simulated environments and the unpredictable nature of real-world settings. For instance, challenges in accurately simulating lighting conditions, capturing the complexities of dynamic environments, and replicating diverse material variations, such as wear, aging, and surface irregularities, all contribute to the observable reality gap between synthetic and real-world data.

In this study, we tackle one aspect of the ongoing challenge to bridge the gap between simulated and real images in industrial applications. We focus on incorporating textures that accurately replicate signs of usage on industrial assets. Furthermore, we study the effect of these textures by training several DL models on synthetic datasets with different texture variations and evaluating them by inferring on real images. We consider six industrial assets with different materials: metallic, plastic, and wood. For each asset, depending on its material, we consider a combination of the following aging texture layers: rust, scratch, dirt, and color change. Our results demonstrate that adding aging textures to the assets enhances the performance of the object detection DL models.

The remainder of this paper is structured as follows: First, we present in Section II an overview of existing literature related to bridging the reality gap in synthetic data. In Section III, we detail the design of the aging textures, providing an in-depth understanding of the proposed approach. Section IV outlines the experimental setup, providing a detailed exploration of the methodologies utilized. Section V presents our results, showcasing the performance of object detection models on real images across diverse texture variations. Finally, we conclude and discuss future research directions in Section VI.

II. Related Work

The field of synthetic data has gained considerable attention in recent years due to its potential to overcome the challenges associated with the limited availability of real-world data in various computer vision applications. Several techniques, including domain randomization (DR), physically based rendering (PBR), and domain adaptation, have emerged as prominent approaches. DR involves altering input data during machine learning training to improve model generalization by introducing random variations. PBR is a graphics technique that simulates realistic lighting and material interactions to generate synthetic images. Domain adaptation, on the other hand, focuses on transferring knowledge across different domains to mitigate domain gaps. In this section, we discuss the DR, PBR, and domain adaptation techniques that are used to reduce the reality gap between real-world and synthetic images.

A. Domain Randomization - DR

While simulation environments provide a low-cost and safe way to train robots, they often fail to capture the diversity and complexity of real-world scenarios, leading to poor generalization when learned models are deployed in the real world. Tobin et al. [30] propose a technique called "Domain Randomization" which addresses the challenge of transferring deep neural networks trained in simulation to the real world. The authors worked on randomizing the simulation environment by introducing variations in lighting conditions, random object textures, camera positions, and other parameters. Tremblay, J. et al. [31] used DR for a car detection task. They generated images by rendering 3D models of cars on top of random backgrounds while randomizing the lighting, camera viewpoints, and textures of random flying distractors, as well as the objects of interest. They conclude that synthetic data, combined with DR, is useful for training deep neural networks for object detection applications.

The idea of structured domain randomization (SDR) was introduced by Prakash et al. [23] in the context of a car detection task. SDR involves randomizing objects' appearance, such as textures, lighting, and positioning, while preserving the scene's underlying structure, which enables the neural network to utilize context when detecting objects. The authors argue that by including SDR in synthetic data generation, the model will learn the relationship between pedestrians and sidewalks, cars and roads, etc.

In comparison to these approaches, our study incorporates SDR to enhance synthetic data generation for object detection. We focus on maintaining realistic relationships between industrial assets within the simulated environment to reflect real-world industrial arrangements. Unlike existing methods that often rely on limited or single-asset industrial datasets, our study distinguishes itself with a comprehensive dataset covering diverse industrial assets commonly found in an industrial setting.

B. Physics-Based Rendering - PBR

PBR is a technique that simulates how light interacts with materials in the real world using physically accurate models of light and material properties. Eversberg and Lambrecht [5] present a PBR image generation pipeline for training deep learning object detection models for industrial objects such as blade turbines. They evaluated different approaches for lighting, background images, object texture, and additional foreground objects to reduce the reality gap between real and synthetic images. They found that image-based lighting with high dynamic range images (HDRIs) using background images from a large-scale dataset such as COCO [13] and randomizing the 3D object's texture while keeping it realistic, results in good performance.

Hodaň et al. [11] used the path tracing render engine Arnold [7] to create realistic 3D models of indoor scenes and objects with realistic materials, using physics simulation to generate object poses. They achieved a 24% improvement in mean average precision compared to the baseline of Hinterstoisser et al. [10], who used OpenGL rendering [35]. They also found that context is important for object detection performance, with up to a 16% enhancement when objects were placed in a scene that realistically modeled the test data setup.

Wong et al. [34] focus on using synthetic data to train deep learning models for object detection in industrial environments, where obtaining real-world data can be challenging. The authors describe a method for generating synthetic data using photogrammetry-based rendering techniques, which involves using a 3D scanner to capture real-world objects and generate 3D models out of the captured images. They used the software Blender to create synthetic images by randomizing lighting intensity, number of lights, camera position, and random background images from the SUN dataset [36]. The authors evaluated their method on a set of real-world objects and showed that the models trained on synthetic data are more effective than using image augmentation with real data, where a limited number of real training images are available.

Existing approaches fall short of capturing the level of realism needed for our industrial scenarios. Our emphasis lies on accurately modeling 3D assets while incorporating different signs of damage to them. Moreover, our approach utilizes a path-tracing render engine, ensuring the generation of high-quality images that simulate realistic material properties and accurately represent the surrounding environment. This attention to detail distinguishes our study, offering a more accurate representation of real industrial scenarios.

C. Domain Adaptation

Domain adaptation has attracted significant interest in recent years due to its potential to close the gap between synthetic and real domains. Different approaches have been proposed, such as learning invariant features between the synthetic and real domains, retraining the model in the real domain, and learning a mapping between the two domains [1], [27], [32], [37].

In this work, we explore the use of synthetic images to train object detection models for inference over real images captured in industrial settings without performing domain adaptation.

III. Aging Textures Design

In this section, we describe the process of selecting industrial assets for our study, followed by the creation of aging texture layers and the defining of aging levels. We used Adobe Substance 3D Designer [3] to design realistic aging textures to simulate wear and tear on various materials.

A. Selecting Industrial Assets

In this work, we consider six industrial assets commonly found in industrial settings. As shown in Fig. 1, these assets are stillage, dolly, jack, pallet, Folding-KLT (Kleinladungsträger) box, and safety cone. The selection of assets is based on their importance and prevalence in industrial environments. Moreover, this selection allowed us to assess the performance of object detection models on a diverse set of objects.

Fig. 1. Comprehensive overview of the experimental assets.

B. Aging Texture Layers

Real-world industrial assets naturally undergo an aging process, making them vulnerable to different types of damage. Continuous usage, challenging environmental conditions, and lack of maintenance all play a role in the gradual degradation of materials [22]. This aging process can take on various forms that affect the robustness and appearance of industrial assets, as seen in Fig. 2.

In this section, we elaborate on our methodology to design the aging textures. We classify the materials under investigation as metallic, wood, and plastic, each featuring a tailored set of aging texture layers, as depicted in Fig. 3. Crafted based on real-world observations from the BMW Munich factory and warehouses, these layers aim to capture the distinct ways in which aging is observed in industrial settings.

Fig. 2. Real-world samples of signs of usage on industrial assets.

1) Metallic material (Dolly, Stillage, and Jack): For metallic objects, we consider three layers: rust, scratch, and dirt.

- **Rust layer:** The rust layer simulates the natural oxidation process that occurs on metallic surfaces over time and gives the object a weathered appearance.
- **Scratch layer:** The scratch layer introduces imperfections caused by wear and tear, representing scratches and the peeling of the paint. This layer is essential for simulating the visual impact of long-term usage on industrial assets.
- **Dirt layer:** The dirt layer replicates the accumulation of dirt on the surface of metallic objects, making them appear as they would in actual industrial environments where dust and dirt tend to settle on surfaces.

2) Wooden material (Pallet): For wooden objects, we consider two layers: dirt and color change.

- **Dirt layer:** Similar to metallic objects, wooden surfaces accumulate dirt and debris over time. The dirt layer is applied to mimic this accumulation across different regions of the object.
- **Color change layer:** The color change layer captures the natural aging of wood, causing it to darken or change hues over time due to exposure to environmental factors such as sunlight and moisture.

3) Plastic material (F-KLT Box and Safety Cone): For plastic objects, we consider two layers: scratch and dirt.

- **Scratch layer:** Similar to metallic objects, plastic surfaces can also experience scratches and paint peeling. The scratch layer introduces these imperfections, emphasizing the impact of wear on the plastic surfaces.
- **Dirt layer:** The dirt layer represents the accumulation of dirt particles on the plastic surface. It is applied to create a natural and realistic appearance.

C. Aging Levels

As shown in Fig. 3, we combined the layers for each material (metallic, wood, and plastic). For instance, we merged the rust, scratch and dirt layers textures for metallic objects into a single texture. Additionally, we established ten intensity levels for each layer, depicting the transition from clean (new) to used (old) objects, as illustrated in Fig. 4. This process resulted in a total of 10 combined textures for each asset.

Fig. 3. Layered texture composition and combined textures across material types.

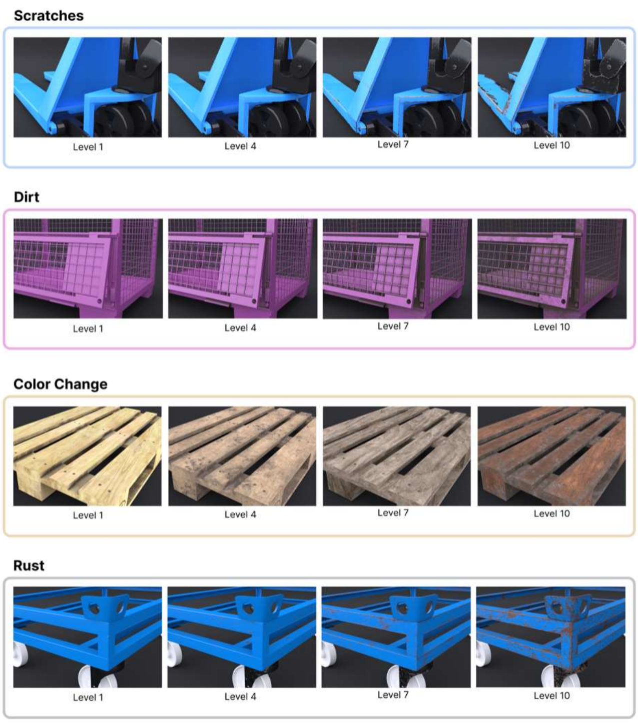

Fig. 4. Examples of individual layers at different levels, from clean (new - Level 1) to used (old - Level 10) textures.

IV. Experimental Setup

In this section, we outline our methodology, including datasets generation, randomizations, datasets statistics, and the training and evaluation procedures for assessing object detection model performance. We used NVIDIA Omniverse [21] and its Isaac Sim application [19] to create simulated industrial environments. In addition to its simulation capabilities, Omniverse Isaac Sim includes tools for annotating and labeling the generated images, making it suitable for training DL models for object detection tasks. We designed the scene with accurate geometry, lighting, and camera setups, ensuring consistency across the generated images.

A. Creating Texture-based Datasets

In this work, we consider and adapt three different texture variations to the previously mentioned assets (c.f. Section III-A):

1) **Clean textures:** The textures applied to the assets were static, new, and remained consistent throughout the images (one texture per asset). This was established as the baseline and comprised images of assets without any aging or visual deterioration textures. Furthermore, we grouped all images containing assets with clean textures into one dataset called Clean Textured Dataset **(CTD)**.
2) **Aging textures:** As mentioned in Section III, we consider ten combined aging textures with different levels of signs of usage, to achieve a high level of realism, mimicking the aged appearance of assets commonly found in practical industrial scenarios. Furthermore, we grouped all images containing assets with aging textures into one dataset called Aging Textured Dataset **(ATD)**.
3) **Random textures:** We consider over 150 random Material Definition Language (MDL) materials from NVIDIA's vMaterials Library [20], such as brick, wood, plastic, stone, and others. By incorporating these textures, we assessed the models' ability to handle unconventional surface appearances, evaluating their generalization beyond typical industrial materials. Finally, we grouped all images featuring assets with randomly varied textures into one dataset called Random Textured Dataset **(RTD)**.

B. Performing Environment Randomizations

To evaluate the effect of texture variations in different scenarios, we employ several DR techniques consistently across our three datasets, as seen in Fig. 5, to ensure that any observed detection performance differences are only related to texture variation, enabling a fair and accurate comparison. To create diverse and complex training scenarios, we consider the following randomizations:

- Wall and ground texture randomization: Over 150 textures from NVIDIA's vMaterials Library were applied to the simulated environment's walls and ground and varied randomly.
- Lighting randomization: The orientation and intensity of lighting sources were randomly adjusted, enhancing the model's adaptability to diverse brightness levels and shadows [38].
- State randomization: The state of specific assets, such as stillages being either closed or opened, empty or full, was randomized. This variation ensures that the model is capable of recognizing and responding to different states of objects within the environment.
- Visibility randomization: This involves the dynamic alteration of objects' visibility within the simulated environment. This technique introduces scenarios where certain objects might appear, disappear, or change their visibility status during the generation of training data.
- Placement randomization: The positions and orientations of the objects within the environment were randomized. This variation forces the model to learn spatial relationships and object interactions from different perspectives [23].
- Camera randomization: Camera position, perspective, and look-at target were randomized to introduce variability in the visual perspectives presented during training, enhancing model adaptability.

C. Datasets Statistics

In this section, we explore the count of annotations for each asset in the synthetic training datasets and real evaluation dataset, offering insights into the distribution and prevalence of industrial assets.

1) Synthetic training datasets: As mentioned before, to maintain a fair and consistent comparison across all three datasets, we controlled various factors, including randomizations and camera positioning. This approach ensures that the number of images remains consistent, with 24,000 images generated with their corresponding labels for each of the three datasets.

2) Real evaluation dataset: For the evaluation phase, 550 real images were collected from diverse sources, including the BMW Munich factory, warehouses, toolshops, and online open-source images. This diverse selection ensures a variety of environments for a comprehensive evaluation. Furthermore, these real images were hand-annotated to provide reliable and accurate ground truth using the following tool [15].

In Fig. 6, we count the number of annotations per asset across the training and evaluation datasets. It illustrates varying occurrences of industrial assets, intentionally reflecting their real-world prevalence, which is crucial for robust training and evaluation of the DL models.

D. Training and Evaluation

For the training phase, we selected several state-of-the-art models to thoroughly assess the three datasets. We employed Faster RCNN models with 101 and 152 backbones [24], known for their robust object detection capabilities. We also performed transfer learning with pre-trained weights based on the COCO dataset [9] using the publicly available training tools [16], [17]. In addition, we selected the YOLOv7 model [33] and trained it from scratch.

It is important to note that for each model, we performed the training process three times with identical hyperparameters and a training-validation ratio of 80-20%, and reported the average accuracy, ensuring robustness and reliability in the evaluation of object detection performance. All the trainings were conducted on an NVIDIA DGX station with four Tesla V100 32 GB GPUs.

V. Experimental Results

In this section, we compare the evaluation results of the different synthetic training datasets discussed above. In Table I, we provide the evaluation performance with the mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5), highlighting the significant differences between the datasets and the effect of texture variation on the object detection models. Taking the Faster RCNN 101 model as an example, the ATD dataset exhibited the highest mAP of 77.72%, surpassing the performance of CTD and RTD with mAP values of 72.51% and 68.16%, respectively. Similar results are observed across the Faster RCNN 152 model, with ATD consistently achieving a superior mAP value of 79.05%. Moreover, the YOLOv7 model recorded 77.32% mAP for ATD, compared to 67.72% for CTD and 65.95% for RTD. These results emphasize the effectiveness of the Aging Textured Dataset in enhancing object detection model performance compared to the other textures (clean and random c.f. Section IV-A).

TABLE I
PERFORMANCE EVALUATION WITH mAP@0.5 ACROSS THE THREE DATASETS AND MODELS USED (HIGHEST mAP VALUES IN BOLD)

Dataset	F-RCNN 101	F-RCNN 152	YOLOv7
CTD	72.51%	75.09%	67.72%
ATD	**77.72%**	**79.05%**	**77.32%**
RTD	68.16%	68.42%	65.95%

After showing that ATD has better overall performance, we now investigate each of the assets as shown in Table II. This gives us a better idea of how different textures affect the accuracy of detecting certain objects. The AP@0.5 for each asset in all three datasets is presented across different

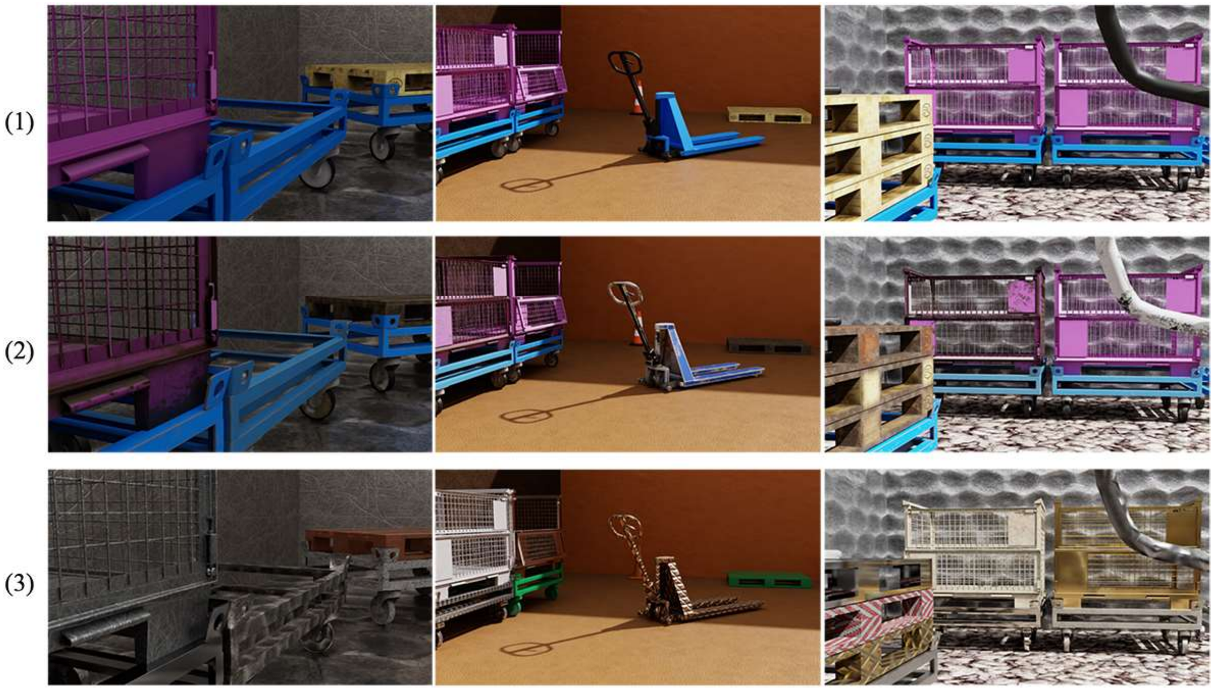

Fig. 5. Comparative dataset samples: (1) CTD, (2) ATD, and (3) RTD, with consistent perspectives and randomizations, highlighting texture variations on industrial assets.

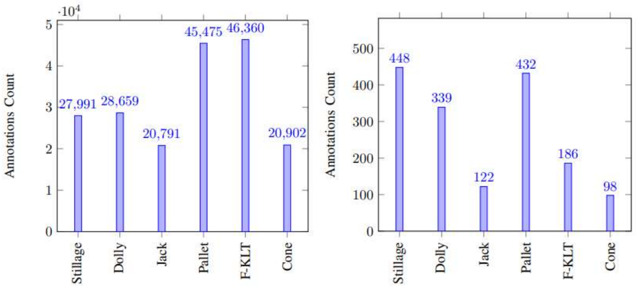

Fig. 6. Annotations count per asset in the three synthetic training datasets (left) and real evaluation dataset (right).

models. Remarkably, the wooden and plastic assets, Pallet, F-KLT Box, and Safety Cone consistently achieved the highest AP scores within the ATD across all models. This highlights the significant influence of incorporating the aging texture layers (c.f. Section III-B), for these materials on the overall performance. In the analysis of the metallic material assets (particularly Dolly, Stillage, and Jack) the performance of the ATD dataset demonstrated some variations compared to CTD and RTD. Depending on the models used, the results were sometimes close across all datasets, with the ATD dataset occasionally surpassing the others such as for the Jack when trained on Faster RCNN 152 and YOLOv7 models. Based on literature and previous studies incorporating DR in an unrealistic manner, RTD achieved the highest accuracy on some assets, like the stillage on the YOLOv7 model. However, this was not the prevailing case for the majority of other assets, where ATD consistently demonstrated the highest accuracy, highlighting the effectiveness of our method.

VI. Conclusion

We investigated in our study the efficiency of synthetic datasets in bridging the reality gap for object detection tasks in industrial settings. We designed and incorporated textures that accurately replicate signs of usage on six industrial assets. Furthermore, we studied the effect of these textures by training several DL models on synthetic datasets with different texture variations and evaluating them by inferring on real images. The evaluation of the Clean, Aging, and Random Textured Datasets demonstrated how aging textures actually help increase the accuracy of DL models. For instance, the Aging Textured Dataset consistently outperformed its counterparts, showcasing the importance of realistic aging layers in enhancing object detection accuracy in industrial scenarios. These findings are promising for diverse industrial applications, encompassing areas such as quality control, inventory management, and automation, where precise object detection is essential.

To gain a more thorough understanding, future work will investigate the impact of generating synthetic datasets with aging textures in realistic simulated environments, such as warehouses or factories. Additionally, future research will delve into a detailed analysis of how each aging layer influences model performance. However, it is important to acknowledge potential limitations, such as the complexity of accurately simulating all aspects of real-world industrial environments and the need for extensive computational resources. Furthermore, expanding the study to include a wider range of industrial

TABLE II
ASSET-SPECIFIC AP@0.5 ACROSS DATASETS AND MODELS (HIGHEST AP VALUES IN BOLD)

Models	Datasets	Assets					
		Stillage	*Dolly*	*Jack*	*Pallet*	*F-KLT Box*	*Safety Cone*
Faster RCNN 101	CTD	63.96%	68.46%	**72.03%**	71.92%	80.33%	78.37%
	ATD	**70.91%**	71.16%	70.32%	**78.85%**	**82.85%**	**92.25%**
	RTD	66.58%	**71.45%**	58.93%	58.96%	70.16%	82.88%
Faster RCNN 152	CTD	**78.91%**	69.76%	67.53%	73.74%	79.16%	81.46%
	ATD	76.20%	68.89%	**72.51%**	**77.26%**	**83.53%**	**95.92%**
	RTD	71.73%	**70.45%**	59.54%	65.09%	77.29%	66.41%
YOLOv7	CTD	67.89%	69.46%	53.55%	62.70%	83.51%	69.46%
	ATD	81.95%	**76.9%**	**54.33%**	**82.98%**	**88.52%**	**79.24%**
	RTD	**86.98%**	75.94%	44.95%	63.28%	86.58%	38.00%

assets and exploring additional layers would further enhance the development of synthetic datasets, leading to improved object detection models for real-world industrial applications.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Emily Boehmer, the talented industrial designer, for her valuable contributions to our study. We extend our appreciation to the QUT Academy, the BMW TechOffice Munich team, and all those who provided insights, guidance, and support throughout the course of this study.

REFERENCES

[1] Chafic Abou Akar, Rachelle Abdel Massih, Anthony Yaghi, Joe Khalil, Marc Kamradt, and Abdallah Makhoul. Generative adversarial network applications in industry 4.0: A review. *International Journal of Computer Vision*, pages 1–60, 2024.

[2] Chafic Abou Akar, Jimmy Tekli, Daniel Jess, Mario Khoury, Marc Kamradt, and Michael Guthe. Synthetic object recognition dataset for industries. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 150–155. IEEE, 2022.

[3] Adobe. 3D design software for authoring - Adobe Substance 3D — adobe.com. https://www.adobe.com/products/substance3d-designer.html.

[4] Morgane Ayle, Jimmy Tekli, Julia El-Zini, Boulos El-Asmar, and Mariette Awad. Bar—a reinforcement learning agent for bounding-box automated refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2561–2568, 2020.

[5] Leon Eversberg and Jens Lambrecht. Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization. *Sensors*, 21(23):7901, 2021.

[6] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[7] Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, et al. Arnold: A brute-force production path tracer. *ACM Transactions on Graphics (TOG)*, 37(3):1–12, 2018.

[8] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[11] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. In *2019 IEEE international conference on image processing (ICIP)*, pages 66–70. IEEE, 2019.

[12] Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering*, 149:106773, 2020.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[14] Mauro Martini, Andrea Eirale, Simone Cerrato, and Marcello Chiaberge. Pic4rl-gym: a ros2 modular framework for robots autonomous navigation with deep reinforcement learning. In *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, pages 198–202. IEEE, 2023.

[15] BMW TechOffice Munich. LabelTool lite. https://github.com/BMW-InnovationLab/BMW-Labeltool-Lite.

[16] BMW TechOffice Munich. AI Evaluation GUI. https://github.com/BMW-InnovationLab/SORDI-AI-Evaluation-GUI, 2022.

[17] BMW TechOffice Munich. Tensorflow Training GUI. https://github.com/BMW-InnovationLab/BMW-TensorFlow-Training-GUI, 2022.

[18] Jimmy Nassif, Joe Tekli, and Marc Kamradt. *Synthetic Data: Revolutionizing the Industrial Metaverse*. Springer Nature, 2024.

[19] NVIDIA. Isaac Sim — developer.nvidia.com. https://developer.nvidia.com/isaac-sim.

[20] NVIDIA. NVIDIA Developer. https://developer.nvidia.com/vmaterials.

[21] NVIDIA. NVIDIA Omniverse — nvidia.com. https://www.nvidia.com/en-us/omniverse/.

[22] Luca Pinciroli, Piero Baraldi, and Enrico Zio. Maintenance optimization in industry 4.0. *Reliability Engineering & System Safety*, page 109204, 2023.

[23] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[25] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.

[26] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[27] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[28] Jimmy Tekli, Bechara al Bouna, Raphaël Couturier, Gilbert Tekli, Zeinab al Zein, and Marc Kamradt. A framework for evaluating image

obfuscation under deep learning-assisted privacy attacks. *2019 17th International Conference on Privacy, Security and Trust (PST)*, pages 1–10, 2019.

[29] Jimmy Tekli, Bechara Al Bouna, Gilbert Tekli, and Raphaël Couturier. A framework for evaluating image obfuscation under deep learning-assisted privacy attacks. *Multimedia Tools and Applications*, pages 1–33, 2023.

[30] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter 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)*, pages 23–30. IEEE, 2017.

[31] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.

[32] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[33] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[34] Matthew Z Wong, Kiyohito Kunii, Max Baylis, Wai Hong Ong, Pavel Kroupa, and Swen Koller. Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ Computer Science*, 5:e222, 2019.

[35] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[36] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[37] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[38] Sergey Zakharov, Rareș Ambruș, Vitor Guizilini, Wadim Kehl, and Adrien Gaidon. Photo-realistic neural domain randomization. In *European Conference on Computer Vision*, pages 310–327. Springer, 2022.