Synthetic Object Recognition Dataset for Industries

Chafic Abou Akar[†‡], Jimmy Tekli[†§], Daniel Jess[†§], Mario Khoury[†], Marc Kamradt[†] and Michael Guthe[¶]
[†]BMW Group, Munich, Germany
Email: {chafic.abou-akar, jimmy.tekli, daniel.jess, mario.khoury, marc.kamradt}@bmw.de
[‡]Univ. Bourgogne France-Comté, Besançon, France
[¶]University of Bayreuth, Bayreuth, Germany
Email: michael.guthe@uni-bayreuth.de
[§]Authors contributed equally

***Abstract*—Smart robots in factories highly depend on Computer Vision (CV) tasks, e.g. object detection and recognition, to perceive their surroundings and react accordingly. These CV tasks can be performed after training deep learning (DL) models on large annotated datasets. In an industrial setting, acquiring and annotating such datasets is challenging because it is time-consuming, prone to human error, and limited by several privacy and security regulations. In this study, we propose a synthetic industrial dataset for object detection purposes created using NVIDIA Omniverse. The dataset consists of 8 industrial assets in 32 scenarios and 200,000 photo-realistic rendered images that are annotated with accurate bounding boxes. For evaluation purposes, multiple object detectors were trained with synthetic data to infer on real images captured inside a factory. Accuracy values higher than 50% and up to 100% were reported for most of the considered assets.**

I. Introduction

Nowadays, smart robots in factories perform repetitive and well-defined tasks while human workers supervise the whole pipeline and guarantee a flexible industrial process [1]. To successfully accomplish these tasks, Computer Vision (CV) allows these robots to perceive and semantically understand their surroundings, e.g. locating and identifying specific objects in a scene [2].

Early on, CV tasks were achievable by a 2-step process: (i) applying a hand-crafted feature extraction algorithm [3] followed by (ii) training a traditional machine learning model on the extracted features [4]. Throughout the last two decades, Deep Learning (DL) techniques surpassed the traditional 2-step process in terms of accuracy and inference time [5].

Although these DL models learn to extract relevant features from input images in an end-to-end manner, training them requires capturing, storing and annotating large amounts of images in comparison to traditional ML approaches [6]. Moreover, the acquisition of large image datasets, mainly in industrial settings (e.g., factories) is becoming more challenging and critical [7] due to the following reasons:

- Excessive human effort is needed for manual image capture, image preprocessing (e.g. cropping, filtering out noisy images) and image annotation (e.g. bounding boxes and pixel-wise segmentation).
- As stated by Ayle et al. [8], these tasks and more specifically image annotation, are highly prone to human error and subjectivity: "*For instance, target objects might be wrongly labeled when a human annotator is biased to label one side of the target object, when it comes to overlapped and close objects or to similar but different scale assets*".
- Capturing images inside factories can be difficult due to limited access for security (e.g. innovation and high security areas), privacy [9] (e.g. human workers along the assembly lines) or safety reasons (e.g. painting or assembly lines, etc).

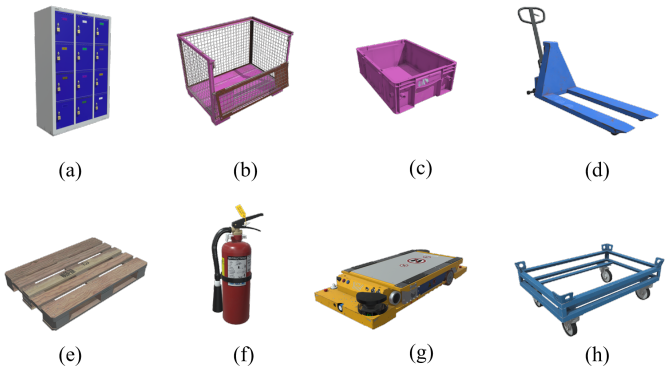

Fig. 1. Assets: (a) Cabinet (b) Stillage (c) KLT (Kleinladungsträger) Box (d) Jack (e) Pallet (f) Fire Extinguisher (g) Smart Transport Robot STR (e) Dolly

Synthetic image generation [10] can address the above challenges while acquiring a large image dataset with the desirable properties, e.g., multi-modality captures[1] for various scene conditions. Using a renderer such as Blender, Unreal, Unity or NVIDIA Omniverse, it is feasible to control each aspect that affects the image: lighting, camera position, assets distribution and animation, noise, etc. Additionally, renderers automatically generate various accurate annotations based on the selected modalities [11]. However, one main challenge arises with synthetic image generation: the "reality gap", i.e. the difference between real and synthetic images [12], [13]. The goal is to minimize this reality gap as it plays a major role when training a DL model on synthetic images and using it to infer on real ones.

Several synthetic datasets were proposed such as SYNTHIA [14] for rural environments and autonomous driving, NVIDIA's FAT [15] and SIDOD [16] for detecting household assets included in the Yale-CMU-Berkeley (YCB) household asset lists [17], etc. On the other hand, these datasets are not designed for industrial applications.

In our work, we generate a synthetic industrial dataset for object detection in a production environment. We use NVIDIA Omniverse to render our images using the latest NVIDIA RTX GPUs for extensive light bounce calculations, hence more photorealistic renderings. Following the Universal Scene Description (USD) pipeline [18], all scenes

[1]Imaging modality refers to the method in which an image is captured/generated. Each modal presents a different type of information e.g. bounding box image, depth, segmentation, etc.

are constructed in collaboration with different 3D experts. Our dataset contains 200,000 synthetic images captured and annotated with 8 assets (see Figure 1) and more than 1 million bounding boxes in 32 different scenarios. Furthermore, to assess the usability of our synthetically generated dataset, we first trained several DL-based single-class object detection models by randomly selecting 3000 synthetic images per asset for the training set. We then evaluated the DL models by inferring over 300 real images captured in different industrial settings. As a result, we achieve accuracies over 50% and up to 100% for certain assets. To the best of our knowledge, we are the first to generate, evaluate and publish a large synthetic dataset with 200,000 images with 8 assets and 32 scenarios dedicated to the industrial field. The remainder of this paper is organized as follows. In Section II, we discuss some of the largest online datasets. The acquisition process is detailed in Section III, while in Section IV, we present our generated dataset composition. In Section V, we evaluate the dataset to test its usability in several CV object detection and recognition applications. We discuss possible future work in Section VI.

II. Related Work

In this section, we compare the latest real and synthetic image datasets: Sections II-A and II-C are highlighting popular image datasets used in multiple fields. Sections II-B and II-D are dedicated to specific industrial CV use cases.

A. Popular CV Real Image Datasets

CIFAR-100 [19] and MS COCO [20] are prominent CV datasets. They cover more than 90 asset classes of daily life facilities, various fauna and flora, and transportation means. Despite their big size, they do not focus on industrial settings making them unsuitable for industrial-specific applications. Still, they are advantageous in transfer learning to share common feature knowledge, and to enable industrial object detection tasks with smaller datasets. ImageNet [7] is another popular image database following the WordNet[2] hierarchy with over 100,000 synsets[3], and an average of 1,000 images per synset. Altogether, it provides tens of millions of well-sorted images. As stated by Beyer *et al.* [21], the dataset is not scalable anymore since it was human-annotated over the years resulting in different class labels for the same object class. In addition, a single label is assigned to each image instead of multiple labels. Furthermore, ImageNet does not comply with the recent data privacy and security policies [22] such as not revealing human faces. To fix all glitches, ImageNet creators are reproducing the dataset by running multiple scripts e.g. obscuring existing information. This reproduction requires more supervision and quality control to maintain the overall usability and efficiency of the previous dataset version [23]–[25].

B. Industrial Real Image Datasets

Apostolopoulos *et al.* [26] employed 6 different industrial datasets to test their proposed modified version of the Virtual Geometry Group (MVVGG19) network for classification [27]. They focused on detecting defective casting [28], metal sheet surfaces [29], magnetic tiles [30], and solar module cells [31]. Additionally, two datasets focus on detecting concrete bridge decks, walls and pavements [32], and 28 industrial 3D objects with different surface, symmetry, complexity, flatness, details, compactness and size, arranged in 800 scenes e.g. cylinder, star, box, engine cover, car rim, etc [33]. However, these datasets are non-generic since they consider specific industrial use cases. Furthermore, they contain fewer images than our dataset and have either low resolution or grayscale images. In contrast, Luo *et al.* proposed a general benchmark image dataset for industrial tool (ITD) [34] to detect different shaped tools that are easily found in every industry. Mechanical engineers hand-labeled 11,000 images in 8 categories. Still, some of the categories are widely general e.g. the protection tools category contains safety goggles, weld eye protectors, and glove objects. Despite the different scenario scenes, ITD focused on small size assets found on a tool shelf or a workshop table only. According to the authors, the dataset is a subject for future enhancement since the mean average precisions vary between 64.37% and 78.20%. In another study, Mayershofer *et al.* considered logistics-specific objects such as pallets, small load loaders, stillages, pallet trucks, and forklifts. The Logistics Objects in Context (LOCO) dataset [35] contains 39,101 images where 5,593 are annotated with 151,428 bounding boxes. Out of these 151,428 bounding boxes, 120,000 refer to only one object class, the pallet. Furthermore, many neural network models are employed to pixelate employees' faces and guarantee their privacy. Even though different physical locations are considered, they remain practically similar as they all consist of warehouses with high pallet density. In addition, using multiple hardware and recording methods resulted in some blurry images that are subject of extra data cleaning and additional image post-processing.

C. Popular Synthetic Image Datasets

The SYNTHIA dataset [14] contains over 213,400 synthetic images (1280×980 px) in a Unity virtual metropolis, comprising both random snapshots and video sequences. 9400 multi-viewpoints are used and there are 13 frequent object classes in driving scenarios while changing seasons, weather, and lighting conditions. Frames feature semantic segmentation annotations at the pixel level as well as depth images. The authors have found that the segmentation result substantially improves when SYNTHIA is combined with real image data [14]. However, bounding box annotations are not included, making SYNTHIA not out of the box useable for object detection use cases, and only dedicated for outdoor, cityscape and driving scenarios. SIDOD [16] comprises 144,000 pairs of stereo images, combining 18 camera views from 3 photorealistic virtual scenarios with up to 10 out of 21 randomly selected household items from the YCB 77 daily life model set [17]. SIDOD images are generated using the NVIDIA Deep Learning Data Synthesizer (NDDS), which is built on top of the Unreal Engine. It renderes images with a higher frame rate then many others, including various features such as: depth, stereo, 3D pose, full rotation, occlusion, extreme lighting, segmentation, bounding box coordinates, and flying distractors. It is intended for object detection, pose estimation, and tracking applications use cases. Compared to its predecessor (i.e., the Falling Things[4] dataset [15]), SIDOD's main scene assets and distractors are

[2]Lexical database of semantic relations between words.

[3]Group of semantically equivalent data elements.

[4]Falling Things (FAT) dataset is a synthetic household image dataset with 21 YCB assets dedicated for 3D object detection and pose estimation.

randomized at each frame instead of capturing a random YCB asset during its falling animation in a static virtual environment/background. However, SIDOD only considers assets found in households such as a bowl and a tomato soup can, making it unsuitable for industrial settings.

D. Industrial Synthetic Image Datasets

In addition to its real captured image dataset, T-LESS [36] contains 10,000 3D synthetic images for 6D pose estimation of textureless 30 rigid objects from 20 different scenes. Each object is represented by two 3D models: the first one is created using CAD and the second one is semi-automatically reconstructed from RGB-D images using fast fusion [37], a volumetric 3D reconstruction system. T-LESS synthetic images vary from simple to complex scenes with multiple instances and a high amount of occlusions and clutter. Despite the benefits of photogrammetry [38] and 3D scanning in providing highly realistic meshes, the whole scene is exported as one single mesh without annotations. As a solution, each asset component must be scanned independently and merged to re-construct the product. Otherwise, time-consuming post-processing is needed. Mayershofer *et al.* [39] suggested a scalable automated pipeline for synthetic image generation using Blender. The pipeline has 3 main phases: first, a background is composed of many random 3D objects and a background image to fill the void gap between the objects. Second, target objects and object-alike distractors are randomly placed within the camera view. Finally, lighting and camera are randomly set up. As a result, automatic annotation, segmentation, and depth images are exported. The authors evaluated their pipeline by detecting differently sized KLT boxes in real images. They demonstrated that real image-based detectors outperform their proposed synthetic image-based detectors. Last but not least, the generation pipeline highly depends on full randomization, hence additional work is needed to minimize the domain gap between synthetic and real images.

III. Data Acquisition Approach

A realistic rendering of a synthetic scene depends on three attributes: geometry, material, and lighting. The lighting affects the material which is assigned to the 3D model. In addition, the quality of the image highly depends on the assets' quality and scene complexity. As a solution, we use NVIDIA Omniverse as it satisfies the aforementioned conditions.

A. Universal Scene Description

Our synthetic image dataset generation is inspired by Pixar Animation Studios' core graphics and rendering pipeline [18]. A cinematic rendering is possible using the Universal Scene Description (USD) framework. Each team working on a specific area is only responsible for delivering, updating, maintaining, and exchanging their complex assets or materials. These assets are then assembled to construct a parent scene similar to the real environment.

B. 3D Mesh Modeling

USD is an open-source standard with wide industry adoption. It is supported by companies such as Autodesk, Apple, Blender, NVIDIA, etc. In parallel, NVIDIA Omniverse also enables live collaborations between different applications e.g. 3DsMax, Unreal, etc., that supports exporting in the USD format. Therefore, as an advantage, it is not necessary to re-model existing 3D assets e.g. whole factories, machines and robots that have already been modeled in one of the aforementioned digital content creation (DCC) software. Yet, for the missing asset models, a 3D modeling team captured industrial assets from different points of view and re-modeled them using Blender.

Fig. 2. Scenes content generation based on real industrial scenarios (a) real captures (b) synthetic replication in NVIDIA Omniverse

C. Material Design

Inspired by the real asset surfaces and textures, the 3D modeling team optimized manual photorealistic parametric (or procedural) materials using Substance 3D Suite. As a definition, a material layer on top of an industrial mesh allows the user to identify what the object is made of. Compared to the traditional UV unwrap materials, a parametric material is better - like vector graphics are better than a raster image - because, it ensures flexibility, modularity and reusability while recording the procedural creation process instead of saving the final texture file [40]. We distinguish between two types of materials:

1) Fully procedural material: it does not require a lot of resources, e.g. its file size is usually limited to only a few kilobytes and it generates an 8K resolution image. Only a few parameters are required: color, roughness, metallicity, patterns, and surface relief.
2) Scan-based material: It provides the highest accuracy and realism, yet less creative flexibility is possible and more resources are needed because texture maps are captured by material scanners.

Furthermore, our 3D modeling team adopted the Image to Material AI tools provided by Substance Alchemist to generate real-life inspired materials.

D. Lighting and Rendering

For a higher image fidelity, we use path-tracing [41] instead of more simple global illumination algorithms because factories are lit by many fluorescent tube lights and there is much indirect lighting due to normally bright walls. This diffuse lighting contributes a lot to the typical appearance of (real) images captured in factories or warehouses. More computation power (e.g. GPUs) is needed for more complex scenes. As Omniverse supports multi-GPU systems we have used two 24 GB RTX 3090 GPUs.

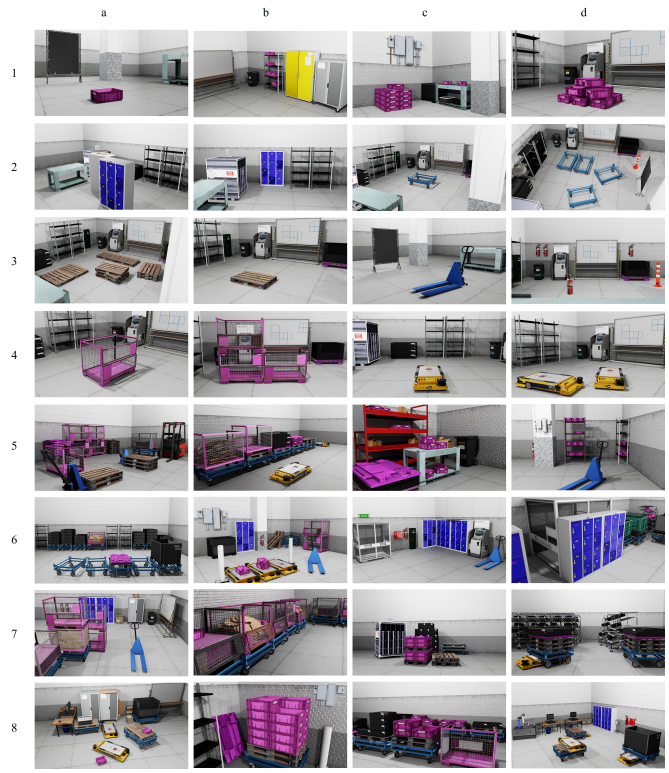

Fig. 3. Synthetic images of different industrial scenarios

E. Scene Content Generation

We add assets to our synthetically generated scenes by emulating their setup and usage within a factory or warehouse. Some scenes contain either a single asset (see Figure 3. 1-4) or several ones (see. Figure 3. 5-8). Each asset might also have either one instance (see Figure 3. a1, b2, c2, b3, c3, a4) or multiple ones (see Figure 3. b1, c1, d1) in a scene. Additional assets (e.g., electrical boxes, industrial racks, whiteboards, office desks, etc.) were added to the surroundings to ensure the scene's randomness while capturing images from different angles. Please refer to Section IV for additional details.

F. Data Cleaning

One of the main advantages of synthetic image generation is the ability to automatically acquire accurate annotations. On the other hand, these annotations are sometimes over-accurate to the pixel level where each image pixel is annotated. As a result, small bounding boxes for far or mostly hidden objects are generated. We noticed that these cases might negatively affect the training accuracy of the object detection models. Therefore, we cleaned our dataset by removing the labels of barely visible assets and preserving the clearly distinguishable ones: we removed bounding boxes whose sizes are below a certain threshold.
Furthermore, generating thousands of synthetic images, especially in a small random space, might lead to similarities. Hence, we implemented an algorithm that (i) hashes each image into an 8x8 monochrome thumbnail, (ii) measures the similarity between two hashes based on the Manhattan distance and (iii) removes the images whose distance is below a predefined threshold.

IV. Dataset Statistics

The dataset contains 200,000 images, automatically captured in 32 different scenes as presented in Figure 3. The first 16 scenes contain single asset scenarios, with a possibility of having multiple instances per image. The rest of the scenes are associated with real-alike factory representations with multiple assets and multiple instances. For each image the camera position and rotation are randomized. We render the images in 720p resolution. In addition, we apply transformation domain randomization e.g. x and y axis position and z axis rotation for some of the annotated assets.

Overall, our 200,000 synthetically generated images have 1,315,642 instances where each image contains on average 2 assets and 7 instances. As seen in Figure 4.a, the most common assets are pallets (21.25%), dollies (19.50%), and KLT boxes (16.01%) since they act as containers or holders in a factory. Numerous KLT boxes can be found in a single scene, due to their small size and use, hence the high number of instances in Figure 4.b. As for the other assets (e.g., STR in Figure 3.a7), each one is available in around 8.65% of the complete dataset. In other words, the industrial assets are not distributed in an equal manner due to their different sizes and usage. Furthermore, 45.12% of the dataset consists of single asset captures. However, it is possible to have multiple instances of the same asset in the same image (see Table I). Yet, less than 6.29% of the images include more than 5 assets.

TABLE I
PERCENTAGE OF DIFFERENT ASSET OCCURRENCE IN A SINGLE CAPTURE

Occurrence	1	2	3	4
Percentage	45.12%	14.72%	23.14%	10.72%
Occurrence	5	6	7	8
Percentage	4.25%	0.43%	0.53%	1.08%

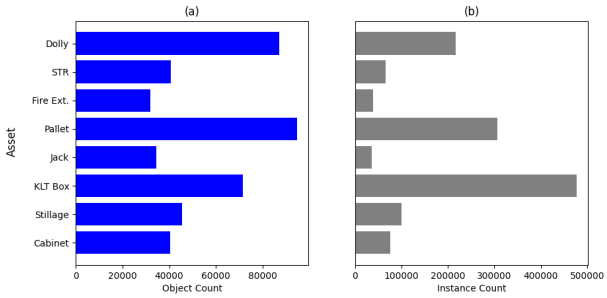

Fig. 4. Dataset (a) object and (b) instance statistics

V. Evaluation Results

As mentioned before, we synthetically generated our dataset to train DL models and perform object detection tasks on real images. In the following section, we consider single-class object detection models.

A. Training & Test sets

For each asset i.e., for each object detection model, we randomly selected 3000 images from our synthetic dataset as training set. For the test set, we captured 300 real images in a factory and annotated them manually. These manually annotated bounding boxes are considered the ground-truth (GT) labels throughout our evaluation.

B. Architectures & Implementation details

We evaluate transfer learning based on the following architectures for object detection tasks: FRCNN Resnet-50, FRCNN Resnet-101 [42], [43], SSD Inception and SSD Mobilenet [44]. All pre-trained models' weights are based on COCO [45]. We conducted our experiments[5] on Tesla V100-SMX2-16GB GPUs.

C. Evaluation metric

The IoU is a widely used metric for object detection tasks. It is calculated by dividing the overlapped area by its union between the GT and the predicted bounding-box. An IoU threshold value determines if a prediction is TP or FP. For instance, if $IoU(GT, Pred) > IoU_{threshold}$ then the prediction is considered a TP, otherwise it is a FP [8]. In the following, we consider an $IoU_{threshold}$ value of 0.5 and report the Average Precision (AP) metric, i.e. AP@0.5.

D. Evaluation of the DL models on real industrial images

As mentioned before, we evaluate the DL models by inferring over real images captured inside a factory. We achieve an accuracy of 100% for the jack and the fire extinguisher using the FRCNN Resnet-101 architecture (see Table II). We noticed that FRCNN Resnet-50 is the best to detect stillages, STRs, dollies and pallets with AP@0.5 of 69.90%, 89.93%, and 48.60%, 46.98% respectively.

As for the other assets, we reached lower accuracy when trying to detect a cabinet (36.39%) or a KLT box (22.17%). As seen in Figure 2, these assets are not characterized by a unique texture, shape, dimension or location and they are usually either stacked or sided next to each other in factories. Our dataset maintains the same 3D model in all images in contrary to the evaluation datasets that include a larger shape variation of our 8 assets. Therefore, the DL models did not perform well when a group of cabinets, or a pack of KLT boxes were homogeneously grouped without any clear separation.

TABLE II
DIFFERENT AP@0.5 OBJECT DETECTION MODELS

Asset	FRCNN		SSD	
	Resnet-50	Resnet-101	Inception	Mobilenet
Cabinet	18.84%	**36.39%**	33.85%	13.25%
Stillage	**69.90%**	68.55%	44.14%	31.86%
KLT Box	21.1%	**22.17%**	14.05%	2.19%
Jack	**100.00%**	**100.00%**	77.78%	58.31%
Pallet	36.87%	**46.98%**	24.69%	17.37%
Fire Ext.	80.65%	**100.00%**	50.00%	71.78%
STR	**89.83%**	86.25%	75.13%	42.72%
Dolly	**48.60%**	30.83%	31.31%	23.06%

VI. CONCLUSIONS

In this paper, we presented a dataset with 200,000 synthetic industrial images for 8 commonly available assets in 32 different industrial scenarios. The dataset is divided into single and multiple asset scenarios. Camera and main target position randomizations are used to increase image diversity. Furthermore, the computational power of NVIDIA RTX GPUs enabled the use of path tracing to improve the render fidelity. Afterward, we assessed our synthetically generated dataset by training DL models for object detection purposes and inferring on real captured industrial images. whereas FRCNN Resnet-50 has successfully detected stillages, STRs, and dollies with over 50% accuracy. Cabinets, pallets and boxes are considered complex and hard-to-detect assets as they can be stacked or placed next to each other. Nevertheless, recognition accuracies are promising and subject for future improvements. For complete access, please contact the authors.

Future works may study the potential effects of domain randomization more carefully to further generalize the dataset, for instance randomizing the camera imaging sensor parameters, light spectral power, assets' textures and materials, assets' shape variations, as well as light interaction with different industrial translucent materials, etc. To support this amount of variations, an automated pipeline is required. Additionally, employing generative networks might decrease the reality gap [12], [13] between real and synthetic images in terms of image content and image quality. Furthermore, comparing our models to real-based trained models could explore this gap even further. Last but not least, increasing the size of the real test dataset is necessary when considering additional object classes.

[5]For training purposes, we implemented the publicly available code on: https://github.com/BMW-InnovationLab/BMW-TensorFlow-Training-GUI

ACKNOWLEDGMENT

We wish to express our gratitude to Dylan SHEPPARD and his designers' team at the QUT Academy for their technical support. We would also like to thank our families and all BMW TechOffice Munich employees, friends, and interns for their encouragement.

REFERENCES

[1] M. Dianatfar, J. Latokartano, and M. Lanz, "Task balancing between human and robot in mid-heavy assembly tasks," *Procedia CIRP*, vol. 81, pp. 157–161, 2019.

[2] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object detection and localization using local and global features," in *Toward category-level object recognition*. Springer, 2006, pp. 382–400.

[3] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1901–1907, 2015.

[4] S. Paisitkriangkrai, J. Sherrah, P. Janney, V.-D. Hengel *et al.*, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–43.

[5] S. Mittal and S. Vaishay, "A survey of techniques for optimizing deep learning on gpus," *Journal of Systems Architecture*, vol. 99, p. 101635, 2019.

[6] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[8] M. Ayle, J. Tekli, J. El-Zini, B. El-Asmar, and M. Awad, "Bar—a reinforcement learning agent for bounding-box automated refinement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, 2020, pp. 2561–2568.

[9] J. Tekli, B. al Bouna, R. Couturier, G. Tekli, Z. al Zein, and M. Kamradt, "A framework for evaluating image obfuscation under deep learning-assisted privacy attacks," in *17th International Conference on Privacy, Security and Trust, PST 2019, Fredericton, NB, Canada, August 26-28, 2019*. IEEE, 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1109/PST47121.2019.8949040

[10] G. A. NVidia, "What Is Synthetic Data?" https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/, 2021, [Online; Accessed January 26, 2022].

[11] Unity, "Materials, Shaders & Textures," https://docs.unity3d.com/560/Documentation/Manual/Shaders.html, 2021, [Online; Accessed January 26, 2022].

[12] 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," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[13] W. Chen, Z. Yu, S. De Mello, S. Liu, J. M. Alvarez, Z. Wang, and A. Anandkumar, "Contrastive syn-to-real generalization," *arXiv preprint arXiv:2104.02290*, 2021.

[14] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. 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*, 2016, pp. 3234–3243.

[15] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2038–2041.

[16] M. Jalal, J. Spjut, B. Boudaoud, and M. Betke, "Sidod: A synthetic image dataset for 3d object pose recognition with distractors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[17] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[18] M. A. Bolstad, "Large-scale cinematic visualization using universal scene description," in *2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE, 2019, pp. 1–2.

[19] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "The cifar-10 and cifar-100 dataset," https://www.cs.toronto.edu/~kriz/cifar.html, 2009, [Online; Accessed August 25, 2021].

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. v. d. Oord, "Are we done with imagenet?" *arXiv preprint arXiv:2006.07159*, 2020.

[22] J. Whitaker, "The fall of imagenet," https://towardsdatascience.com/the-fall-of-imagenet-5792061e5b8a, March 19, 2021, [Online; Accessed August 25, 2021].

[23] Khari Johnson, "ImageNet creators find blurring faces for privacy has a 'minimal impact on accuracy'," https://venturebeat.com/2021/03/16/imagenet-creators-find-blurring-faces-for-privacy-has-a-minimal-impact-on-accuracy, 2021, [Online; Accessed January 18, 2022].

[24] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky, "Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 547–558.

[25] K. Yang, J. H. Yau, L. Fei-Fei, J. Deng, and O. Russakovsky, "A study of face obfuscation in imagenet," in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 313–25 330.

[26] I. D. Apostolopoulos and M. A. Tzani, "Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with deep learning approach," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2022.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] R. Dabhi, "Casting product image data for quality inspection," https://www.kaggle.com/ravirajsinh45/real-life-industrial-dataset-of-casting-product/activity, 2020, [Online; Accessed August 25, 2021].

[29] X. Lv, F. Duan, J.-j. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, 2020.

[30] Y. Huang, C. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *The Visual Computer*, vol. 36, no. 1, pp. 85–96, 2020.

[31] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Automatic classification of defective photovoltaic module cells in electroluminescence images," *Solar Energy*, vol. 185, pp. 455–468, 2019.

[32] S. Dorafshan, R. J. Thomas, and M. Maguire, "Sdnet2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data in brief*, vol. 21, pp. 1664–1668, 2018.

[33] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing mvtec itodd-a dataset for 3d object recognition in industry," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2200–2208.

[34] C. Luo, L. Yu, E. Yang, H. Zhou, and P. Ren, "A benchmark image dataset for industrial tools," *Pattern Recognition Letters*, vol. 125, pp. 341–348, 2019.

[35] C. Mayershofer, D.-M. Holm, B. Molter, and J. Fottner, "Loco: Logistics objects in context," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 612–617.

[36] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 880–888.

[37] F. Steinbrücker, J. Sturm, and D. Cremers, "Volumetric 3d mapping in real-time on a cpu," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2021–2028.

[38] E. M. Mikhail, J. S. Bethel, and J. C. McGlone, "Introduction to modern photogrammetry," *New York*, vol. 19, 2001.

[39] C. Mayershofer, T. Ge, and J. Fottner, "Towards fully-synthetic training for industrial applications," in *LISS 2020*. Springer, 2021, pp. 765–782.

[40] Adobe, "Parametric Materials: Back to the Source of Substance 3D Assets!" https://substance3d.adobe.com/magazine/parametric-materials-back-to-the-source-of-substance-3d-assets/, June 23, 2021, [Online; Accessed September 11, 2021].

[41] A. Keller, T. Viitanen, C. Barré-Brisebois, C. Schied, and M. McGuire, "Are we done with ray tracing?" in *SIGGRAPH Courses*, 2019, pp. 3–1.

[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[43] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[44] L. Barba-Guaman, J. Eugenio Naranjo, and A. Ortiz, "Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded gpu," *Electronics*, vol. 9, no. 4, p. 589, 2020.

[45] H. Yu, C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, and J. Li, "TensorFlow Model Garden," https://github.com/tensorflow/models, 2020.