Towards Sim-to-Real Industrial Parts Classification with Synthetic Dataset

Xiaomeng Zhu[1,3,*], Talha Bilal[1], Pär Mårtensson[1], Lars Hanson[2], Mårten Björkman[3], Atsuto Maki[3]
[1]Scania CV AB, [2]University of Skövde, [3]KTH Royal Institute of Technology
{xiaomeng.zhu, talha.bilal, par.martensson}@scania.com,
{xiazhu, celle, atsuto}@kth.se, lars.hanson@his.se

Abstract

This paper is about effectively utilizing synthetic data for training deep neural networks for industrial parts classification, in particular, by taking into account the domain gap against real-world images. To this end, we introduce a synthetic dataset that may serve as a preliminary testbed for the Sim-to-Real challenge; it contains 17 objects of six industrial use cases, including isolated and assembled parts. A few subsets of objects exhibit large similarities in shape and albedo for reflecting challenging cases of industrial parts. All the sample images come with and without random backgrounds and post-processing for evaluating the importance of domain randomization. We call it Synthetic Industrial Parts dataset (SIP-17). We study the usefulness of SIP-17 through benchmarking the performance of five state-of-the-art deep network models, supervised and self-supervised, trained only on the synthetic data while testing them on real data. By analyzing the results, we deduce some insights on the feasibility and challenges of using synthetic data for industrial parts classification and for further developing larger-scale synthetic datasets. Our dataset [†] *and code* [‡] *are publicly available.*

1. Introduction

Efficient and reliable automatic parts classification is critical for various industrial operations and handling processes, such as sorted storing, part feeding, and quality inspection. With the increasing variability of products and required flexibility of processes and material flow, the importance of it has further escalated [15]. Deep learning-based classification algorithms, with their robustness, can be a possible solution for industrial parts classification. However, training these algorithms generally requires a large amount of annotated data, which can be time-consuming and label-expensive to obtain in many real-world industry scenarios.

Synthetic data may present a viable solution to overcome this challenge. In the manufacturing industry, Computer-Aided Design (CAD) models are commonly used to create detailed virtual representations of physical objects for planning and simulating the manufacturing process [30]. Accordingly, synthetic data generated from CAD models can be useful to tackle the challenge of limited real-world data [4, 6, 33]. However, a major issue in it is the domain gap between CAD data and real data, as they are derived from different distributions.

Numerous deep learning studies have focused on addressing the challenge of domain shift from simulated to real images, and a majority of them evaluate their models on the benchmark Sim-to-Real dataset, such as the Visual Domain Adaptation Dataset (VisDa) [20]. However, since the current benchmark datasets often consist of general objects such as animals, furniture, and street view, they may not adequately model the characteristics of industrial parts [18]. In particular, the industrial environment often involves parts with subcategories differences or alignment variations that may not typically be captured by those datasets. As a result, the methods that perform well on the existing Sim-to-Real datasets may not generalize effectively to industrial scenarios.

Therefore, in this study, we introduce a Synthetic Industrial Parts dataset (SIP-17) which contains 17 objects representing six industrial use cases of parts sorting and quality inspection. The dataset comprises both isolated and assembled parts, some of which exhibit significant similarities or albedo, reflecting the challenges encountered in real-world industrial parts classification scenarios. As such, this dataset may serve as a preliminary testbed for Sim-to-Real industrial parts classification research. Testing new models on this dataset may also provide insights into the robustness of the model in solving various Sim-to-Real industrial parts classification use cases.

The dataset focuses on the Sim-to-Real challenge, where only synthetic data is used for the training and validation

[*] Corresponding author
[†] Dataset: https://www.kaggle.com/datasets/mandymm/synthetic-industrial-parts-dataset-sip-17
[‡] Code: https://github.com/BilalTalha/SIP-17

(source) domains, while real images are used for the test (target) domain. Specifically, the dataset includes 66K labeled synthetic images for training and validation, and 566 unlabeled real images for testing. Unlike previous Sim-to-Real object identification datasets that are often benchmarked with domain adaptation models requiring real data for training [20, 35], we train our dataset only on synthetic data to enhance its practical value for industrial applications. By doing so, the manufacturers may bypass the need for manual data collection and annotation, and potentially develop parts classification models for quality inspection or parts sorting stations before the physical production of the parts.

Regarding the aforementioned domain gap, domain randomization can be a possible technique for addressing it. Tobin et al. [29] introduced the concept of domain randomization which involves randomizing various aspects of the training data, such as camera positions, lighting conditions, object positions, and textures, to simulate a wide range of possible scenarios. The goal is to narrow the Sim-to-Real gap by generating synthetic data with sufficient variation allowing the model to perceive real-world data just as another variation [29]. In this study, we generated the SIP-17 dataset following the domain randomization technique. To assess the impact of domain randomization, we generated the synthetic data with random backgrounds and random post-processing (Syn_R) and without them (Syn_O).

We evaluated state-of-the-art classification models on both Syn_R and Syn_O to establish benchmarks for our dataset. We selected a range of classification models with varying design principles, including Convolutional Neural Networks (CNNs), a Vision Transformer (VIT) [7], and a self-supervised learning network. The results demonstrated varying levels of performance while training on data from different use cases, providing insights into the feasibility and challenges of utilizing synthetic data for industrial parts classification. It may also indicate some direction for the development of a larger-scale synthetic dataset in the future.

2. Related Work

2.1. Sim-to-Real Dataset

Numerous datasets have been developed for Sim-to-Real tasks in the past. The Linemod [12] and Linemod-Occluded [2] datasets, for example, are widely used in 6D pose estimation in robotics. They include synthetic and real images of 15 general objects with varying textures, shapes, camera poses, lighting conditions, occlusions, and more. These datasets serve as the benchmarks for Sim-to-Real object localization and pose estimation tasks.

For Sim-to-Real classification tasks, the Visual Domain Adaptation Classification Dataset (VisDa-C) [20] is a benchmark dataset that comprises both synthetic and real images of 12 objects. The synthetic images are generated from 3D models rendered from various angles and lighting conditions, while the real images are sourced from the Microsoft COCO dataset [16] and the YouTube Bounding Box dataset [23].

For Sim-to-Real segmentation tasks, multiple datasets have been introduced, particularly in the context of 2D and 3D multi-object tracking or autonomous guidance. These datasets often comprise synthetic images that are rendered from video games like GTA5 or different virtual urban environments, as well as real-world data obtained from a moving vehicle in urban settings or GPS, including RGB images, stereo images, and lidar data. Some examples of these datasets include the Domain Adaptation Segmentation Dataset (VisDa-S) [20], the GTA5 dataset [25], the Virtual KITTI [9], and KITTI [10] datasets.

While many Sim-to-Real datasets feature general objects such as toys, animals, furniture, and street view, there are few options available for studying industrial objects. One of the datasets is the T-less dataset [13], which includes 30 industrial objects with uniform textures and colors, such as bearings, U-brackets, metal boxes, and knives. Additionally, the Dataset of Industrial Metal Objects [5] offers real-world and synthetic multi-view RGB images of six objects, including cylinders, blocks, and shafts, placed on three different types of carriers: pallets, bins, and cardboard. However, these datasets are both designed for pose estimation, so they may not be well-suited for the challenge of cross-domain classification, as their test objects are limited to uniform or fixed textures and colors.

2.2. Domain Randomization in Sim-to-Real

Sadeghi and Levine [26] showed that quadcopters could be trained to fly indoors using only synthetic images, and Peng et al. [19] demonstrated the possibility of training object classifiers using 3D CAD models with random textures and backgrounds. Building on these ideas, Tobin et al. [29] proposed the concept of domain randomization to address the reality gap by generating synthetic data with sufficient variations to enable the network to view real-world data as just another variation. Subsequent research [21, 31, 34] applied the domain randomization strategy to the GTA5 and Virtual KITTI datasets, training CNN-based object detection or segmentation models such as Faster-RCNN only on the synthetic data and achieving promising results while evaluating real-world data.

For industrial parts identification, some works have utilized physics-based rendering and domain randomization to generate synthetic training data for various industrial parts, as demonstrated in two studies [8, 14]. The synthetic data is generated with randomized backgrounds, textures, post-processing, and other factors. Ablation studies are performed to analyze the impact of these factors on Sim-to-

Real object detection tasks using object detection models such as Yolov4 [1] and Faster R-CNN [24]. In these studies, Eversberg and Lambrecht [8] focus on identifying three types of turbine blades, while Horváth et al. [14] generated a dataset containing ten different objects, including bracket, pipe clamp, and handle. However, it is worth noting that the dataset only consists of isolated objects, which may not reflect the alignment differences that are frequently encountered in industrial assembly quality inspection use cases. In addition, the evaluation of the dataset using CNNs may not include a Sim-to-Real classification benchmark utilizing advanced models such as VIT.

3. The SIP-17 Dataset

The SIP-17 dataset comprises 17 industrial objects that are representative of six use cases. The first four use cases consist of isolated parts that require classification. The last two involve assembled parts, where the objects consist of two or more parts that are assembled to each other, requiring inspection to ensure whether the parts are correctly aligned.

Use case 1: cabin assembly quality inspection Use Case 1 includes five objects: Airgun, Electricity12, Hammer, Hook, and Plug, which are assembled in the cabin of a truck. These objects from the same assembly station could share large similarities in albedo. By accurately classifying them, we can verify that the correct part has been assembled in the cabin.

Use cases 2 to 4: logistic picking inspection Use case 2 comprises three objects: Fork1, Fork2, and Fork3; use case 3 includes four objects: CouplingHalf, Gear1, Gear2, and Pinion; and use case 4 consists of three objects: Cross, Pin1, and Pin2. These objects can be found in various logistic picking stations, and their classification is vital in ensuring that the operators have picked the correct parts for delivery. The parts in the same station are likely to belong to a closely related product family, which could share large similarities in shape and albedo.

Use case 5: wheel assembly quality inspection Use case 5 involves inspecting whether a wheel has been correctly assembled with a screw. As the wheel can be assembled inside-out, leading to four possible categories during assembly: front side of the wheel with a screw (FwS), front side of the wheel without a screw (FwoS), back side of the wheel with a screw (BwS), and back side of the wheel without a screw (BwoS).

Use case 6: engine assembly quality inspection Use case 6 involves inspecting the Oring that is assembled on the Power Take Off. It includes three categories: correct assembly of the Orings (Oon), offside assembly of the Top Oring (Ooff), and missing Top Oring (noO).

In summary, we selected four use cases with 15 isolated parts from the assembly and logistic stations. These parts varied in appearance, with some being similar while others differed. Additionally, we included two use cases with two assembled parts. Each category of assembled parts shared numerous similarities as they contained the same objects but differed in alignment details. We chose these 17 objects from six use cases as they may represent the challenges encountered in real-world industrial parts classification scenarios.

3.1. Dataset Acquisition

To evaluate the effectiveness of domain randomization, we generated two synthetic datasets, one with random backgrounds and post-processing (Syn_R) and one without those (Syn_O).

The Syn_O dataset was created by rendering 3D CAD models from varying camera angles and under diverse lighting conditions. Each object was randomly rotated, scaled, and translated to generate variations. We used camera angles of 360 degrees and six distinct light directions for each model. To ensure that the entire object was captured, the camera was automatically positioned, and random light intensities were used during rendering. We followed the rendering parameters described in work [22].

The Syn_R dataset was generated using a similar process as the Syn_O dataset, with the additional step of introducing random colors to the lighting. Besides, we incorporated randomly selected backgrounds from the Unsplash dataset [32], along with various post-processing techniques such as random color tints, blurs, and noise [22].

During the generation of the dataset, the randomization process was only applied to the virtual camera and environment. As for the CAD models, we applied a single-color texture that approximated the real objects to give them a reasonably realistic appearance. Nevertheless, to maintain the Sim-to-Real domain gap, other parameters affecting surface appearance, such as metallic, specular, and roughness, were held constant across all CAD models.

In total, we generated 33K images for both Syn_R and Syn_O datasets. For each category in each use case, we generated 1200 synthetic images for training and 300 synthetic images for validation. For testing, we captured 566 real images from various industrial scenarios. The number of images per category is outlined in Tab. 1. Some samples of Syn_R, Syn_O, and real images for each category are present in Fig. 1.

4. Evaluation

4.1. Experimental Setup

Our evaluation of the SIP-17 dataset as a benchmark involves testing five classification models that are currently considered state-of-the-art. These models, which represent different design principles, are widely known for their effectiveness and relatively simple implementation. The five

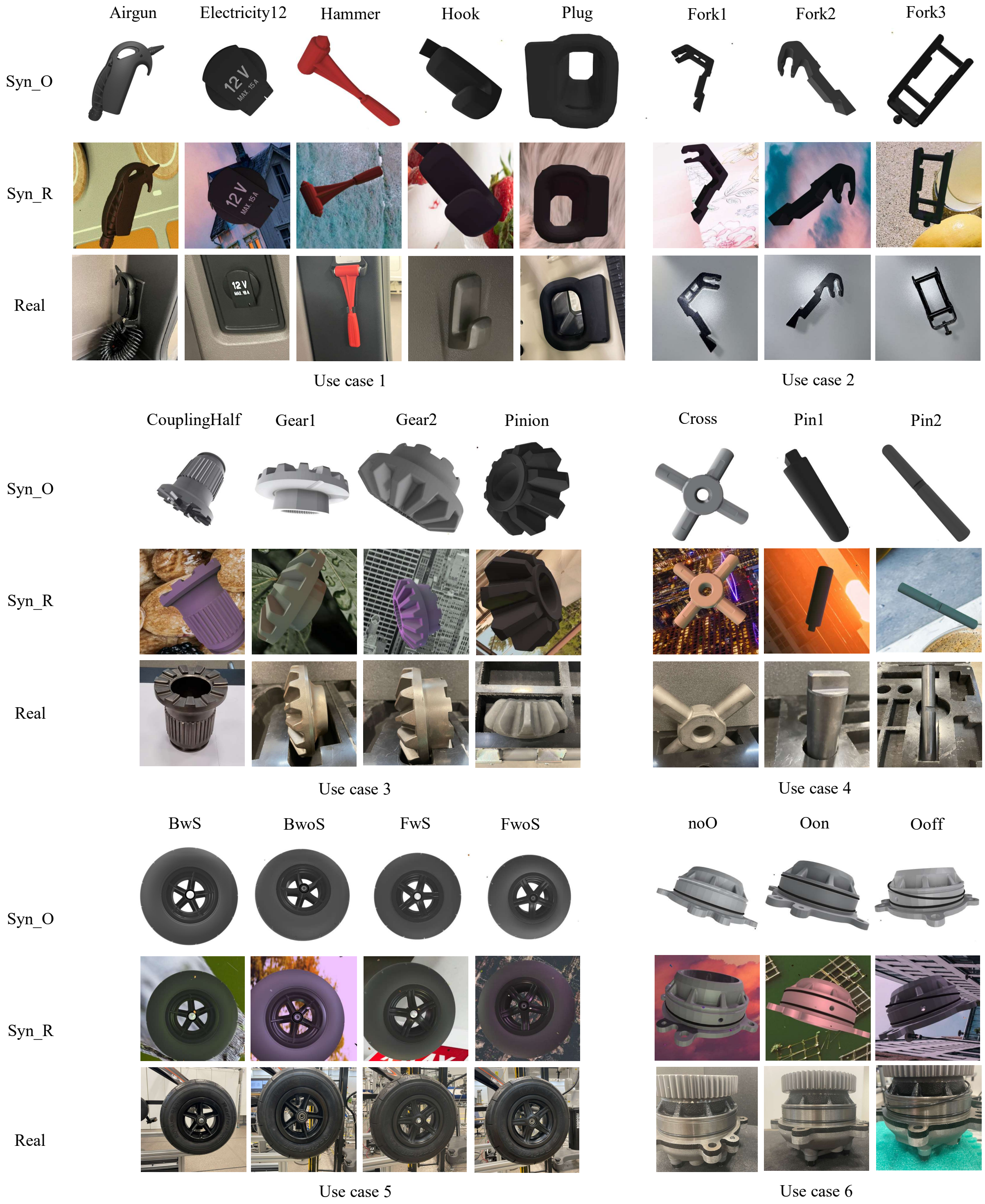


Figure 1. Sample images from the SIP-17 dataset, showcasing three categories: Syn_O, synthetic images without random backgrounds and post-processing; Syn_R, synthetic images with random backgrounds and post-processing; and Real, images captured from cameras in real industrial scenarios. Use cases 1-4 require the classification of isolated industrial parts, while use cases 5 and 6 require the classification of assembled parts.

Use cases	Categories	Train (Syn_R/ Syn_O)	Valid (Syn_R/ Syn_O)	Test (real images)
Use case 1	Airgun	1200	300	39
	Electricity12	1200	300	44
	Hammer	1200	300	34
	Hook	1200	300	40
	Plug	1200	300	53
Use case 2	Fork1	1200	300	32
	Fork2	1200	300	30
	Fork3	1200	300	30
Use case 3	CouplingHalf	1200	300	33
	Gear1	1200	300	34
	Gear2	1200	300	38
	Pinion	1200	300	44
Use case 4	Cross	1200	300	40
	Pin1	1200	300	39
	Pin2	1200	300	36
Use case 5	Back_wheel_with_screw (BwS)	1200	300	32
	Back_wheel_without_screw (BwoS)	1200	300	32
	Front_wheel_with_screw (FwS)	1200	300	32
	Front_wheel_without_screw (FwoS)	1200	300	32
Use case 6	Orings_on (Oon)	1200	300	42
	TopOring_off (Ooff)	1200	300	42
	no_TopOring (noO)	1200	300	42

Table 1. Number of images per category in the SIP-17 dataset.

models under evaluation are:

ResNet [11]: ResNet is a deep CNN architecture that introduced the concept of residual connections. Its simplicity and effectiveness have made it a popular baseline for image classification and benchmarking new methods. It is also frequently used as a classifier for various Sim-to-Real object identification tasks. In our experiments, we employed ResNet with 152 layers (ResNet152).

EfficientNet [27]: EfficientNet is a family of CNN architectures that achieve state-of-the-art performance while being computationally efficient. In our experiments, we utilized the EfficientNet B7 model.

ConvNext [17]: ConvNext is a recent CNN architecture that introduced a split-attention mechanism to enhance the ability of the network to aggregate features. It ”modernized” a standard ResNet toward the design of a Vision Transformer and achieved state-of-the-art classification results in CNNs on several benchmarks. In our experiments, we employed the ConvNext base model.

Vision Transformer (VIT) [7]: VIT is based on the transformer architecture used in natural language processing. Unlike traditional CNNs, it employs a self-attention mechanism that processes image patches directly, effectively capturing global dependencies and relationships between different parts of the input image. It has achieved state-of-the-art performance on several image recognition benchmarks. In our experiments, we utilized the VIT model with a base configuration and a patch size of 16 (vit_b_16).

DINO [3]: DINO is a self-supervised contrastive learning approach that improves feature representation for image classification tasks. It utilizes a teacher network to generate representations of an image and trains a student network to predict similarities between pairs of images in order to learn more meaningful and transferable features. Self-supervised learning methods have shown some effectiveness in Sim-to-Real tasks by learning features that are more transferable across domains. For example, Tian et al. [28] have proposed a self-supervised approach using contrastive learning to learn domain-independent features. In light of this, we aim to evaluate a self-supervised contrastive learning model DINO on our dataset. In our experiments, we employed the DINO model with VIT as its backbone and utilized a base configuration with a patch size of 16 (Dino_vitbase16).

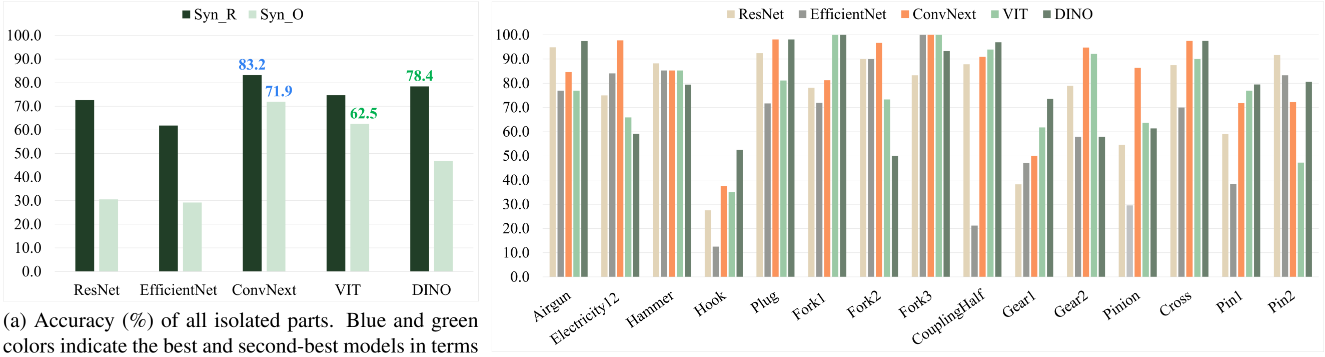
We chose the models with a comparable amount of parameters. For the self-supervised learning model DINO, we used the VIT pre-trained on ImageNet as its backbone and trained the linear classifier on our dataset for 25 epochs. For supervised learning models, we trained the models pre-trained on ImageNet for 25 epochs. To thoroughly evaluate the models performance, we conducted two types of experiments: (1) training the models with the 15 isolated parts and (2) training the models with objects per use case. All the models were trained with Syn_R and Syn_O datasets and tested on real images.

4.2. Experimental Results and Discussion

All the experiments have been repeated three times to obtain an average top-1 classification accuracy. The results of training on 15 isolated parts are presented in Fig. 2, while the results of training on each individual use case are summarized in Fig. 3. To highlight the best and second-best models in terms of total accuracy trained with Syn_R and Syn_O, we use blue and green colors, respectively, in Fig. 2 (a) as well as Fig. 3 (a) and (b).

Domain randomization comparison: By comparing Fig. 2 (a), Fig. 3 (a) and (b), it is evident that the models trained on Syn_R outperformed those trained on Syn_O when trained on both 15 isolated parts and individual use cases. These findings suggest the significance of domain randomization in Sim-to-Real tasks, highlighting the benefits of training models with diverse synthetic data to improve their resilience to real-world variations. As models trained on Syn_R achieved better overall performance, our analysis will mainly focus on the results obtained from this dataset. We present the class-wise performance of the models trained with Syn_R in Fig. 2 (b) and Fig. 3 (c).

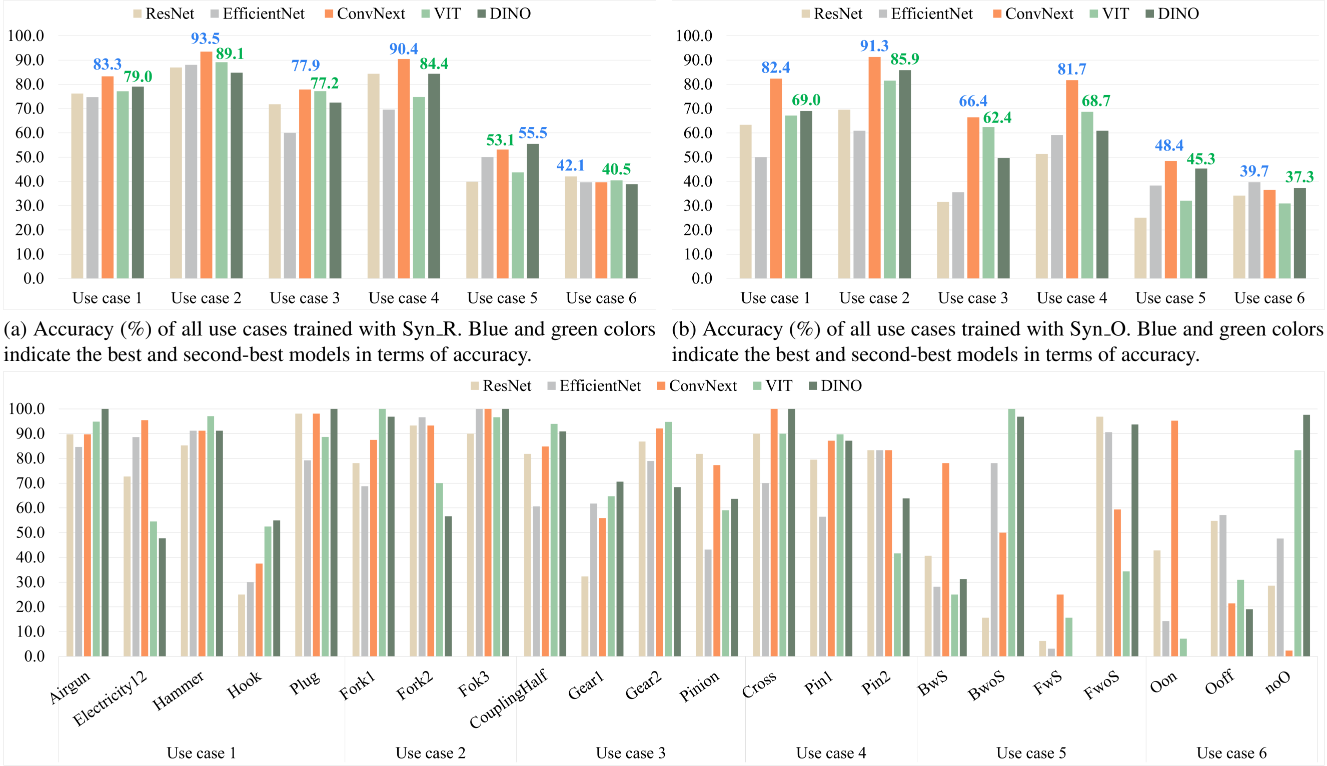
Model performance comparison: As shown in Fig. 2, training on 15 isolated parts with Syn_R yielded the highest total accuracy of 83.2% for the ConvNext model, followed by DINO (78.4%) and VIT (74.7%). These results are consistent with those in Fig. 3, where ConvNext demonstrated the best performance in use cases 1 to 4 and the second-best performance in use case 5, followed by DINO and VIT. Notably, the ConvNext model exhibited the smallest performance difference between training on Syn_O and Syn_R,



(a) Accuracy (%) of all isolated parts. Blue and green colors indicate the best and second-best models in terms of accuracy while training with Syn_R or Syn_O.

(b) Class-wise accuracy (%) of all isolated parts while trained with Syn_R.

Figure 2. Results of all isolated parts.



(a) Accuracy (%) of all use cases trained with Syn_R. Blue and green colors indicate the best and second-best models in terms of accuracy.

(b) Accuracy (%) of all use cases trained with Syn_O. Blue and green colors indicate the best and second-best models in terms of accuracy.

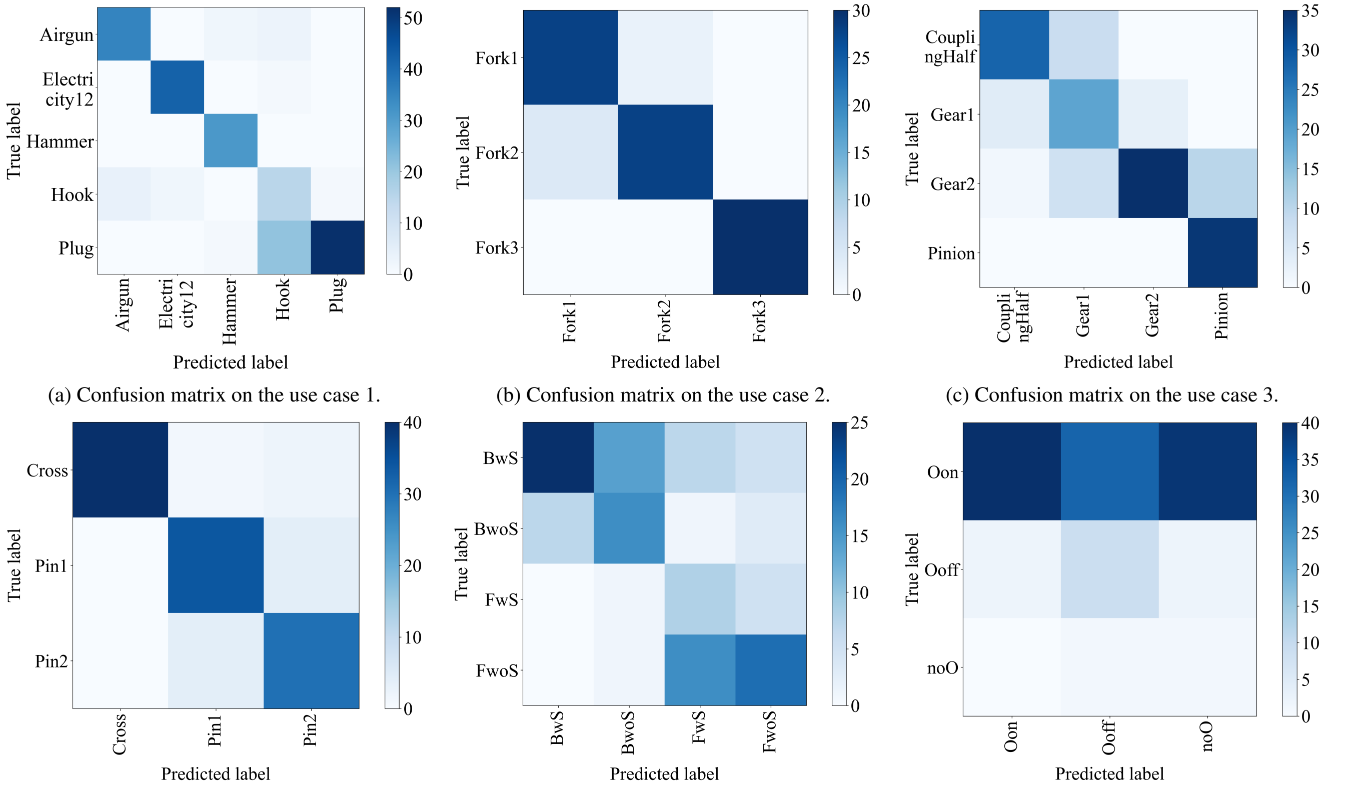
(c) Class-wise accuracy (%) of all use cases trained with Syn_R.

Figure 3. Results of all use cases.

indicating its potential robustness for cross-domain classification, possibly due to its split-attention mechanism. On the other hand, the self-supervised learning model DINO achieved the second-best average performance, which may suggest the effectiveness of contrastive learning strategies for cross-domain classification tasks.

The results also suggest that DINO outperformed ConvNext in some categories, such as Hook and Gear1. Therefore, combing the strength of ConvNext and DINO may potentially lead to further improvement in model performance. For instance, we could use a contrastive learning strategy on ConvNext or add a supervised loss to DINO to create supervised contrastive learning models with attention mechanisms, potentially enhancing their ability to capture subcategory details in cross-domain classification.

High-performing categories: The highest accuracy achieved in our experiments was by the ConvNext model in use cases 2 and 3 while trained with Syn_R, with accuracies of 93.5% and 90.4%, respectively. These findings indicate the potential of utilizing synthetic data in parts classification



(a) Confusion matrix on the use case 1.

(b) Confusion matrix on the use case 2.

(c) Confusion matrix on the use case 3.

(d) Confusion matrix on the use case 4.

(e) Confusion matrix on the use case 5.

(f) Confusion matrix on the use case 6.

Figure 4. Confusion matrices on different use cases with the ConvNext model.

and suggest that training models only on synthetic images with domain randomization may yield promising results.

Low-performing categories We carried out further analysis to identify which categories may have contributed to the low performance in other use cases. When analyzing the results of isolated parts classification from use cases 1 to 4, a comparison between Fig. 2 (b) and Fig. 3 (c) reveals that the class-wise results obtained from training on individual use cases exhibit a similar trend to those obtained from training on all 15 parts. These findings suggest that, in industrial parts classification, models are likely to confuse objects that share the same manufacturing process and stations. In addition, both Fig. 2 (b) and Fig. 3 (c) reveal that the categories Hook and Gear1 received the lowest accuracy across most models in isolated parts classifications.

As for the results of assembled parts classification from use cases 5 and 6, Fig. 3 (a) indicates that they performed considerably worse than the isolated part classification. All models received an accuracy of around 50% or lower in use cases 5 and approximately 40% in use cases 6.

To further explore the issue, confusion matrices were generated for all use cases of the best model, ConvNext, as shown in Fig. 4. Upon analyzing these matrices, it became apparent that certain subcategories exhibited high confusion rates. Specifically, in the isolated parts classification, Hook was frequently confused with Plug, while Gear1 was often mistaken for CouplingHalf or Gear2. In the assembled parts classification of use case 5, Back_wheel_with_srew (BwS) exhibits high confusion with Back_wheel_without_srew (BwoS), and Front_wheel_with_screw (FwS) was frequently misclassified as Front_wheel_without_screw (FwoS). Moreover, in use case 6, all categories were confused and misclassified as Orings_on (Oon). These low-performance results indicate that the models failed to capture the semantic representation that distinguishes these categories.

To summarize the issue, we divided these categories into two groups. The first group comprises objects with similar albedos and simple shapes, such as Hooks and Plugs. As depicted in Fig. 1 use case 1, Hooks are commonly assembled on a black surface in real images. Since the color of the Hook is also black, and it has a small size and simple shape, the model may focus on the albedo of the entire image rather than the specific features of the Hook, resulting in confusion with other objects that share similar albedo characteristics with Hook, such as Plugs.

The second category includes objects that share partial similarities, such as Gear 1, CouplingHalf, and Gear 2, as well as all assembled objects. As depicted in Fig. 1 use cases 3, classes such as Gear 1, CouplingHalf, and Gear 2 exhibit a similarity of around 50%. Therefore, the ConvNext model could potentially misclassify Gear 1 as either CouplingHalf or Gear 2. Additionally, the degree of similarity among assembled objects may be even higher, depending on the size difference between the objects being assembled together. For example, as demonstrated in use cases 5 and 6 in Fig. 1, the screw and Orings only constitute a small portion of the Wheel and Power Take Off, respectively. This could potentially bias the model because larger objects may have more easily identifiable features and significant impacts on the model than smaller objects. This could explain the poor results obtained from the confusion matrix in Fig. 4 (e) and (d), where the model struggled to capture the alignment relationship between the assembled parts and misclassified them into a single category.

The low-performance categories in our dataset may indicate the limitations and challenges associated with Sim-to-Real industrial parts classification. They offer us opportunities to specialize in addressing the most challenging use cases. Furthermore, the presence of high-performance and low-performance categories suggests that our dataset includes use cases in various levels of difficulty and complexity, making it a potential benchmark for evaluating future Sim-to-Real classification models.

5. Limitation and Future Work

Given that Sim-to-Real industrial parts classification presents challenges for categories sharing the same albedo and those that are partially the same, our next step is to further develop the dataset to address these challenges.

One possible approach is to randomize the albedo on synthetic images. Adding more variations of color, texture, and material to the CAD models may generate synthetic images with more variations, enabling the network to perceive real-world albedo as just another variation. Moreover, to improve the classification of assembled parts, we could apply different random albedos to each object that is assembled with others. This could potentially allow the models to identify each assembled object and learn their alignment relationships.

In addition, our SIP-17 dataset, which comprises only 17 objects from six industrial use cases, would only partly explain a comprehensive representation of the diverse range of real-world industrial parts. To address this limitation, we intend to increase the size of our dataset by including more isolated and assembled parts with varying degrees of similarity, based on the insights gained from this study.

6. Conclusion

In this study, we present a Synthetic Industrial Parts dataset (SIP-17) designed for Sim-to-Real industrial parts classification. It contains 17 objects from six industrial use cases, comprising both isolated and assembled parts. We generated synthetic images using domain randomization techniques, resulting in two datasets: Syn_R, with randomized backgrounds and postprocessing, and Syn_O, without them.

To benchmark the dataset, we evaluated it with various state-of-the-art classification models. The models allowed varying levels of performance when training on data from different use cases, with some achieving more than 90% accuracy while some below 50%. These results may reveal some potential and challenges of using synthetic data for industrial parts classification and for further creating larger-scale synthetic datasets. One of the main challenges was raised from the subcategories that share similar albedo or are partially the same.

We wish to encourage researchers to focus on Sim-to-Real classification using only synthetic data for training, with a particular emphasis on addressing the challenges posed by the subcategories. Such research has the potential to bring significant benefits to the manufacturing industry, where parts from the same stations often share similar albedo and shapes. Enabling training without real-world data can in principle eliminate the need for data collection and annotation, thus saving time and resources for manufacturers. We hope our work will serve as a preliminary testbed and benchmark for future Sim-to-Real industrial parts classification research.

Acknowledgement

This work is partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018-05973, as well as by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

We gratefully acknowledge our colleagues at the Production Oskarshamn, Production Zwolle, Transmission Assembly, Engine Assembly, Academy, and Smart Factory Lab Departments at Scania CV AB for providing the CAD models and use cases. We extend our thanks to Pooja Rangarajan and her colleagues at the Volkswagen Virtual Engineering Lab for their support in generating synthetic images from CAD models. In addition, we express our sincere appreciation to Marcos Escudero-Viñolo at Universidad Autónoma de Madrid for his insightful suggestions on benchmarking the dataset.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *13th European Conference on Computer Vision (ECCV)*, pages 536–551. Springer, 2014. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 9650–9660, 2021. 5

[4] Julia Cohen, Carlos F Crispim-Junior, Céline Grange-Faivre, and Laure Tougne. Cad-based learning for egocentric object detection in industrial context. In *15th International Conference on Computer Vision Theory and Applications*, volume 5, pages 644–651, 2020. 1

[5] Peter De Roovere, Steven Moonen, Nick Michiels, et al. Dataset of industrial metal objects. *arXiv preprint arXiv:2208.04052*, 2022. 2

[6] Jonathan Dekhtiar, Alexandre Durupt, Matthieu Bricogne, Benoit Eynard, Harvey Rowson, and Dimitris Kiritsis. Deep learning for big data applications in cad and plm–research review, opportunities and case study. *Computers in Industry*, 100:227–243, 2018. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 5

[8] 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. 2, 3

[9] 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 (CVPR)*, pages 4340–4349, 2016. 2

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[11] 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 (CVPR)*, pages 770–778, 2016. 5

[12] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *11th Asian Conference on Computer Vision (ACCV)*, pages 548–562. Springer, 2013. 2

[13] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 2

[14] Dániel Horváth, Gábor Erdős, Zoltán Istenes, Tomáš Horváth, and Sándor Földi. Object detection using sim2real domain randomization for robotic applications. *IEEE Transactions on Robotics*, 2022. 2, 3

[15] Joerg Krueger, Jan Lehr, Marian Schlueter, and Nils Bischoff. Deep learning for part identification based on inherent features. *CIRP Annals*, 68(1):9–12, 2019. 1

[16] 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 *13th European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 2

[17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 5

[18] Christopher Mayershofer, Dimitrij-Marian Holm, Benjamin Molter, and Johannes Fottner. Loco: Logistics objects in context. In *19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 612–617. IEEE, 2020. 1

[19] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE international conference on computer vision*, pages 1278–1286, 2015. 2

[20] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2021–2026, 2018. 1, 2

[21] 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 *International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019. 2

[22] Pooja Rangarajan, Nikhil Gupta, Müller Andreas, Andre Breitenfeld, Sebastian Schulz, Sheng Ling, Thomas Kammerlocher, and Fabian Baier. Computer-implemented method and system for generating a synthetic training data set for training a machine learning computer vision model, December 2022. European patent application, 21179758.4. 3

[23] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5296–5305, 2017. 2

[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. 3

[25] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *14th European Conference on Computer Vision (ECCV)*, pages 102–118. Springer, 2016. 2

[26] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016. 2

[27] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[28] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *16th European Conference on Computer Vision (ECCV)*, pages 776–794. Springer, 2020. 5

[29] 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 *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 2

[30] Michael Tovey. Drawing and cad in industrial design. *Design Studies*, 10(1):24–39, 1989. 1

[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 (CVPR) workshops*, pages 969–977, 2018. 2

[32] Unsplash. Unsplash, 2003–2023. 3

[33] 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. 1

[34] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2100–2110, 2019. 2

[35] Youshan Zhang. A survey of unsupervised domain adaptation for visual recognition. *arXiv preprint arXiv:2112.06745*, 2021. 2