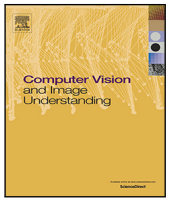
Computer Vision and Image Understanding 240 (2024) 103907

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Indoor Synthetic Data Generation: A Systematic Review

Hannah Schieber [a,c,*], Kubilay Can Demir [b], Constantin Kleinbeck [a,c], Seung Hee Yang [b], Daniel Roth [a]

[a] Technical University of Munich, School of Medicine and Health, Department Clinical Medicine, Klinikum rechts der Isar, Orthopedics and Sports Orthopedics, Munich, Germany
[b] Speech and Language Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[c] Human-Centered Computing and Extended Reality, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

ARTICLE INFO

Communicated by Juergen Gall

MSC:
68U05
68U20
68T40
68T45

Keywords:
Synthetic data generation
Indoor synthetic data
Domain randomization

ABSTRACT

Objective: Deep learning-based object recognition, 6D pose estimation, and semantic scene understanding require a large amount of training data to achieve generalization. Time-consuming annotation processes, privacy, and security aspects lead to a scarcity of real-world datasets. To overcome this lack of data, synthetic data generation has been proposed, including multiple facets in the area of domain randomization to extend the data distribution. The objective of this review is to identify methods applied for synthetic data generation aiming to improve 6D pose estimation, object recognition, and semantic scene understanding in indoor scenarios. We further review methods used to extend the data distribution and discuss best practices to bridge the gap between synthetic and real-world data.
Methods: We adhered to the guidelines of the systematic PRISMA technique. Three databases, IEEE Xplore, Springer Link, and ACM, and an additional manual search were conducted. In total, we identified 241 studies and included 34 in our systematic review.
Conclusion: In summary, synthetic data generation has been performed using crop-out methods, graphic APIs, 3D modeling or authoring tools, or game engine-based methods. To extend the data distribution, varying scene parameters, i.e., lighting conditions or textures and the use of distracting objects in the scene are promising.

1. Introduction

One fundamental challenge that convolutional neural networks (CNNs) face is related to their generalizability, which refers to their ability to perform well on unseen data beyond the training set. This issue arises due to the tendency of CNNs to either overfit or underfit when confronted with publicly available benchmarks (Koch et al., 2021). To train well-generalized CNNs, a broad set of data has to be available. Annotating real-world data is one way to address this challenge. However, this is a time consuming and error-prone process. To address these problems, synthetic datasets are generated by combining 3D models of individual objects and scenes using specific rendering pipelines, see Fig. 1. These rendering pipelines can generate a variety of input data and ground truth labels including, for example, semantic or instance-level annotations (Beery et al., 2020). Moreover, synthesized data can compensate for missing real-world datasets (Ros et al., 2016; Alhaija et al., 2018; Nikolenko et al., 2021; Richter et al., 2016; Johnson-Roberson et al., 2016; Richter et al., 2017; Kleinbeck et al., 2022), and has been used in autonomous driving, robotics, and augmented reality.

Semantic scene understanding plays a vital role in many computer vision tasks. For a comprehensive use of CNNs, pixel-wise annotated training data is crucial to construct robust applications. To address and automate this time consuming annotation process, synthetic virtual worlds have been constructed using game engine-based methods (Cabon et al., 2020; Dosovitskiy et al., 2017; Gaidon et al., 2016; Ros et al., 2016).

In addition to semantic scene understanding, object detection and 6D position estimation are essential for grasping objects with robots. Synthesized data generation using, e.g., the Unreal engine (Jalal et al., 2019; Tremblay et al., 2018b), enables easier creation of object detection and 6D pose ground truth data than using real-world images and manual annotation processes (Jalal et al., 2019). Synthetic data is widely used to train the models in robotic scenarios, as exploration of the real-world environment is constrained by ongoing manufacturing processes (Nikolenko et al., 2021). Furthermore, real-world environments are often cost-intensive and not accessible to the user (Bousmalis et al., 2017).

* Corresponding author at: Technical University of Munich, School of Medicine and Health, Department Clinical Medicine, Klinikum rechts der Isar, Orthopedics and Sports Orthopedics, Munich, Germany.
E-mail address: hannah.schieber@tum.de (H. Schieber).

https://doi.org/10.1016/j.cviu.2023.103907
Received 31 July 2023; Received in revised form 24 November 2023; Accepted 24 December 2023
Available online 2 January 2024

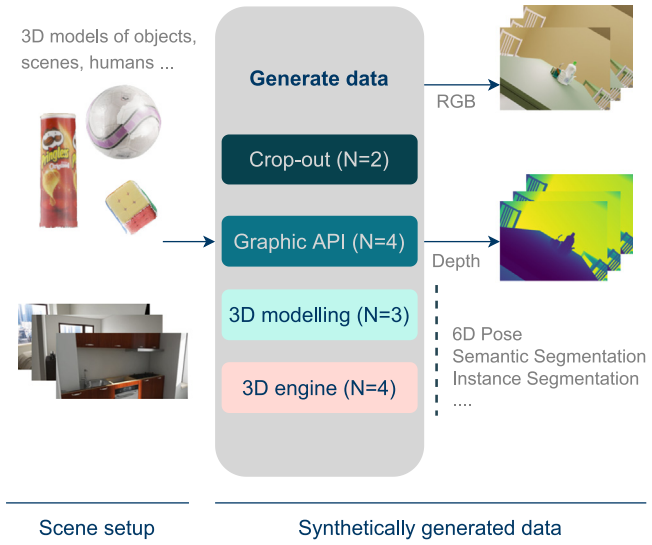


Fig. 1. Our literature review identified four major directions used to generate synthetic data. All directions consider objects and scenes/background images (left) and use then one of the crop-out-based, graphic application programming interface (API)-based, 3D modeling-based or 3D engine-based methods. These methods allow the generation of rich ground truth data. The N indicates the number of identified methods in this sub-category.

Moreover, real-world annotation processes often face limitations in accuracy and scalability (Tremblay et al., 2018b). Considering accuracy, real-world labels can be error-prone due to camera shifts, manual annotation errors, or misclassifications. This can lead to not well-generalized models or misclassifications (Northcutt et al., 2021). Synthesized data enables the creation of error-free labeling after an initial setup step. The 3D models used in the scene can be labeled a priori or in the setup step. Afterward, the 3D models provide accurate object boundaries enabling, for example, accurate bounding boxes or segmentation masks. In terms of scalability, synthetic approaches can simply generate additional data as long as 3D models of the objects and scenes are provided.

Although synthetic data is promising, the sim-to-real gap arises. This gap denotes the variation between synthetic and real-world images. To bridge this gap, domain adaption (Csurka, 2017; Wilson and Cook, 2020; Figueira and Vaz, 2022) and domain randomization (Alghonaim and Johns, 2021; Tobin et al., 2017) methods have been proposed. Domain adaption focuses on adapting an existing model or dataset to an unseen target domain while domain randomization randomizes the input image to extend the data distribution. Other reviews and surveys focused on domain adaption (Csurka, 2017; Wilson and Cook, 2020; Figueira and Vaz, 2022), domain randomization (Alghonaim and Johns, 2021; Zhao et al., 2020) and synthetic data generation in a broader scope (Paulin and Ivasic-Kos, 2023). Zhao et al. (2020) focused on domain randomization for reinforcement learning, and Alghonaim and Johns (2021) benchmarked domain randomization methods for simple objects like a cube.

One common method for domain adaption is the use of generative adversarial networks (GANs) (Goodfellow et al., 2014). The use of GAN for synthetic data generation is deeply analyzed by Figueira and Vaz (2022). Domain adaption in an unsupervised manner is investigated by Wilson and Cook (2020).

To the best of our knowledge, the generation of synthetic indoor datasets for object detection, 6D pose estimation, and semantic scene understanding as well as their randomization approaches to bridge the gap between simulated and real data have not yet been investigated. Synthetic data for indoor scenes is an interesting aspect as robotic grasping or 6D pose estimation are specifically used indoors. The generation of synthetic data for outdoor environments is often limited to autonomous driving.

1.1. Contribution

We contribute a systematic literature review of indoor synthetic data generation approaches to generate synthetic datasets for object detection, 6D pose estimation, and semantic scene understanding. Our review focuses on indoor scene settings and investigates methods of domain randomization to bridge the sim-to-real gap in this scenario. Our systematic literature review:

- identifies reusable approaches for synthetic dataset generation,
- researches synthetic datasets for object detection, 6D pose estimation, and semantic scene understanding, and
- determines how the sim-to-real gap is addressed.

2. Methodology

We followed the PRISMA (Liberati et al., 2009) method to provide a transparent and reliable state-of-the-art review. PRISMA is a method that can be applied to report systematic reviews and meta-analyses. It provides a checklist of items that are considered essential for transparent systematic review, i.e. one item is to define the inclusion and exclusion criteria.

For the keyword definition and study collection in the field of synthetic data generation, we considered the following research questions for this review:

- Q_1: *What are the common methods for indoor synthetic dataset generation?*
- Q_2: *Which synthetic datasets are mainly used for object detection, 6D pose estimation and scene understanding?*
- Q_3: *Which challenges does synthetic data generation face?*
- Q_4: *How are these challenges for synthetic data addressed?*

2.1. Information collection

In search for previous scientific works on the topic, IEEE Xplore, ACM, Springer Link datasets, and an additional manual search were considered. To answer the given research questions (Q_1–Q_4), we identified the following keywords: *{synthetic data, synthetic datasets, photorealistic images, photorealistic image generation, indoor datasets, indoor dataset generation}, {sim-to-real, sim-to-real transfer, domain randomization, domain adaption}, {object detection, object detection, semantic segmentation, 6D pose estimation}*.

We combined these keywords for the database search. All peer-reviewed English publications from 2000–2023 were considered. To avoid missing out important research which does not explicitly mention these keywords, we expanded our search with an additional free search. Using the free search we found some additional approaches and approaches already included in our reading list acquired via the previously conducted database search. However, some non-peer reviewed works that are available as e.g., tech-reports were found which are frequently cited but not available in proceedings or journals. Non-peer-reviewed work was only considered for highly cited works. The database search has been conducted from April 15th, 2023–June 25th, 2023. After an initial title screening and removal of duplicates, 52 studies remained. For the final review, 34 studies are taken into account after full-text reading. The procedure is depicted in Fig. 2. The search, title screening, and removal of duplicates have been performed by a single reviewer. A second reviewer validated the full-text reading and inclusion/exclusion criteria. All disagreements were resolved by an agreement after a discussion or by consulting other reviewers' opinions.

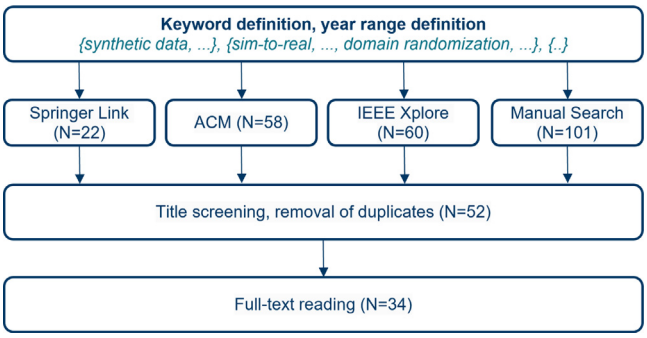


Fig. 2. Literature review process. The keywords in the first box are used in the database search via Springer Link, ACM, and IEEE XPLORE. Additionally, a manual search is conducted using the same year range and keywords. After the removal of duplicates and a review phase, 34 studies are included after full-text reading.

2.2. Inclusion and exclusion criteria

The main objective of this systematic review is to identify approaches that enable synthetic data generation for object detection, 6D pose estimation, and semantic scene understanding in indoor scenarios. Fig. 3 denotes these approaches, alongside chronological relation to domain adaption and domain randomization. A broader overview of more facets of synthetic data generation including humans, autonomous driving, robotic simulations, and flight simulations can be found in Nikolenko et al. (2021).

As inclusion criteria for synthetic dataset generation, we defined the generation of indoor datasets or objects placed in indoor environments. However, some of the work can also be applied to outdoor scenarios like autonomous driving. In autonomous driving, synthetic datasets are already studied extensively (Cabon et al., 2020; Dosovitskiy et al., 2017; Gaidon et al., 2016; Ros et al., 2016; Santara et al., 2021) and some of the reviewed approaches (Borkman et al., 2021) can also be applied towards such use cases. Moreover, synthetic data generation is prominent in many areas of machine learning and computer vision. Another area for example is person re-identification (Xiang et al., 2023, 2022). However, our review focused on the generation of indoor datasets or objects placed in indoor environments. In robotics, predefined setups exist (i.e. Kolve et al. (2019), Shen et al. (2021), Xiang et al. (2020)), providing e.g., semantic segmentation masks. We exclude such approaches as they feature predefined scenarios and objects. Further, they are limited in their customizability. We especially included approaches that are customizable and adaptable for use case-specific scenarios. For domain adaption and randomization, we further extended our inclusion criteria to provide the most usable approaches and knowledge. The knowledge from these approaches can be leveraged when creating an own synthetic dataset.

2.2.1. Study selection

The literature search, removal of duplicates, and title screening are done by a single reviewer. While the full-text reading, inclusion and exclusion criteria are validated by a second reviewer. In the case of disagreements a third reviewer is included and the selection is discussed and resolved. The selected studies are grouped by main topic (synthetic data generation method, synthetic datasets, domain adaption, and domain randomization), publication year, and within their category by their used method.

3. Synthetic dataset generation methods

3.1. Definition and categorization

Based on our literature review, we have identified four overarching categories utilized in synthetic data generation: crop-out, graphic API, 3D modeling (3D authoring), and 3D game engine-based methods. These categories are aligned with specific creation methods, often associated with distinct software frameworks or toolkits. This classification framework provides users with the flexibility to select their preferred method based on individual preferences regarding software or toolkit usage. However, as depicted in Table 1, it is evident that certain approaches offer more ground truth options than others.

Crop-out: Crop-out-based methods are methods combining two individual images or more together to one overall image representation (Georgakis et al., 2017; Sagues-Tanco et al., 2020).

Graphic API: Graphic API-based methods rely on specific graphic APIs, such as OpenGL, (Hinterstoisser et al., 2018; Mercier et al., 2019) as their core framework for synthetic data generation. Additionally, these methods might also incorporate individual physics engines as part of their approach (Morrical et al., 2021).

3D modeling: 3D modeling-based methods leverage modeling tools equipped with integrated physics engines, such as Blender, as foundational platforms for their operations (Greff et al., 2022; Denninger et al., 2020).

3D game engine: The utilization of 3D game engine-based methods involves repurposing engines initially designed for game development to facilitate data generation. Engines such as Unreal and Unity offer a multitude of options within their scene settings, providing extensive flexibility that proves highly advantageous for synthetic data generation purposes (Qiu and Yuille, 2016; Martinez-Gonzalez et al., 2021; To et al., 2018; Borkman et al., 2021).

Initial approaches were mainly crop-out or graphic API-based, and often focused on a small set of ground truth options (Georgakis et al., 2017; Sagues-Tanco et al., 2020; Hinterstoisser et al., 2018; Hodaň et al., 2019; Mercier et al., 2019), see Table 1.

More recent approaches address diverse task settings and are mostly based on a 3D modeling tool (Denninger et al., 2020; Greff et al., 2022) or a game engine (Borkman et al., 2021; Martinez-Gonzalez et al., 2021; Qiu and Yuille, 2016; To et al., 2018), see Fig. 3. Besides these methods, Ge et al. (2022) utilized Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) to generate synthetic training images on-demand for the object detection task. NeRF replaces classical API-based rendering like rasterization or ray-tracing with neural network-based rendering. NeRFs can create novel views from unseen viewpoints by only using a few images for training instead of the need for a full 3D scene setup including complex meshes. However, NeRFs or other network-based methods are often complex to train and still need some initial input data. This data is currently either real-world data which limits the adaptability of scenes and the level of randomization or is generated using e.g., 3D modeling-based methods (Ge et al., 2022) like BlenderProc (Ge et al., 2022; Denninger et al., 2020).

However, crop-out, graphic API-, 3D modeling- (or 3D authoring), and game engine-based methods are still dominating the field. Example renderings of reusable methods can be found in Fig. 4. The used methods provide the possibility to add and control the parameters of the scene. Some approaches allow more parameter manipulation or more complex scene loadings compared to others.

Comparing the reviewed methods to CNN-based data generation, the reviewed methods allow the control of all scene parameters (light, camera pose etc.). Although CNNs can be monitored during the training, at the beginning of the training the success of the outcome is uncertain. CNN based methods are usually more frequently used for domain adaption instead (Wilson and Cook, 2020; Figueira and Vaz, 2022).

3.2. Crop-out methods

Crop-out-based methods use real images from objects of interest. From the original image of this object, only a crop-out of the instance is used without any noise or remaining border information. This crop-out is then added to a background image (Georgakis et al., 2017; Sagues-Tanco et al., 2020).

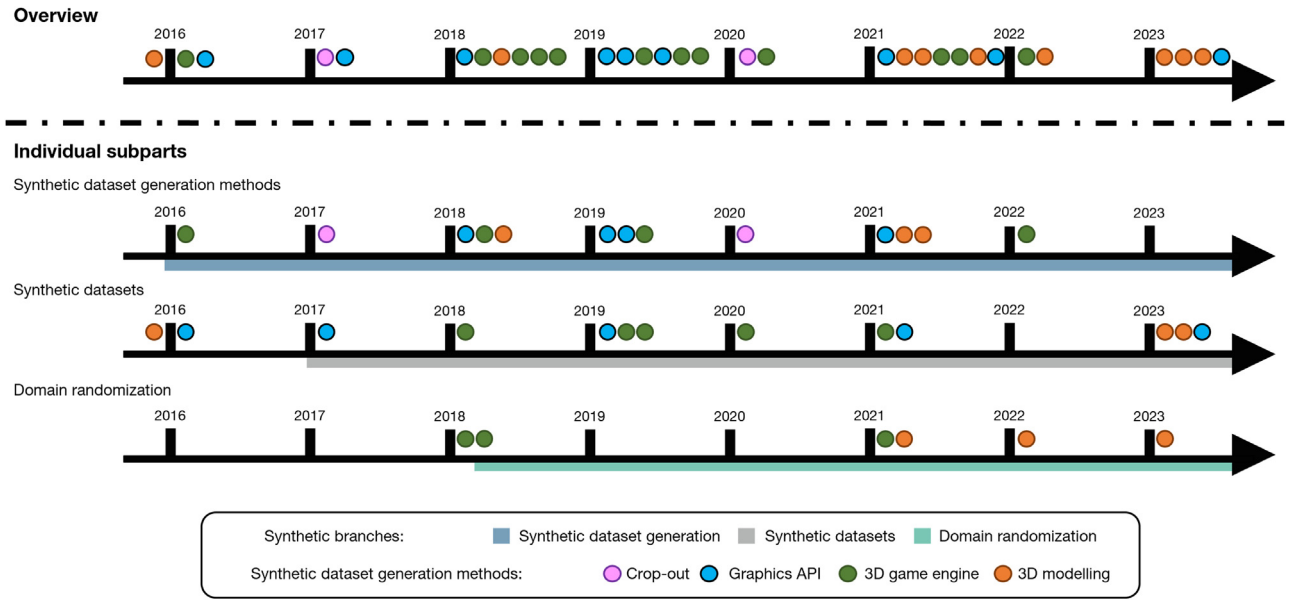


Fig. 3. Timeline of synthetic dataset generation. We provide a generalized overview (top) and show the individual subcategories of the review (bottom). We highlight the identified methods namely crop-out-based, GraphicsAPI-based, 3D game engine-based, and 3D modeling-based methods in each subcategory of the review.

Table 1
Summary of synthetic dataset generation approaches. The ✓ denotes if this ground truth options is available. A (✓) denotes that it is available with limitations. The approaches are grouped by crop-out-based, graphic API-based, 3D modeling-based, and 3D game engine-based methods. Within each category the approaches are listed by year.

Category and method	Authors	Year	Inst. seg.	Sem. seg.	2D Obj. det.	3D Obj. det.	6D Pose	Depth	Keypoints	Normals	DR	Physics	Ray tracing	Source (Link)	API
Crop-out-based															
–	Georgakis et al. (2017)	2017	–	–	✓	–	–	–	–	–	–	(✓)	–	ProjectPage	Objective-C
–	Sagues-Tanco et al. (2020)	2020	✓	✓	✓	–	–	–	–	–	(✓)	–	–	–	Python
Graphic API-based															
–	Hinterstoisser et al. (2018)	2018	–	–	✓	–	–	–	–	–	(✓)	–	–	–	OpenGL/C
–	Mercier et al. (2019)	2019	–	–	–	–	✓	✓	–	–	(✓)	–	–	–	OpenGL/Python
–	Hodaň et al. (2019)	2019	–	✓	✓	–	✓	–	–	–	✓	✓	✓	–	Autodesk Maya/Arnold
NViSII	Morrical et al. (2021)	2021	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	GitHub	Python/NVIDIA OptiX
3D Modeling-based															
–	Talukdar et al. (2018)	2018	–	–	✓	–	–	–	–	–	–	✓	–	–	Python/Blender
BlenderProc	Denninger et al. (2020)	2019	✓	✓	✓	(✓)	✓	✓	–	✓	✓	✓	✓	GitHub	Python/Blender
Kubric	Greff et al. (2022)	2022	✓	✓	(✓)	✓	✓	✓	–	✓	(✓)	✓	✓	GitHub	Python/Blender
3D Game engine-based															
UnrealCV	Qiu and Yuille (2016)	2016	✓	✓	✓	–	–	✓	–	✓	(✓)	✓	–	ProjectPage	C++/Python
NDDS	To et al. (2018)	2018	✓	✓	✓	✓	✓	✓	✓	–	✓	–	–	GitHub	C++
Unity Perception	Borkman et al. (2021)	2021	✓	✓	✓	✓	–	✓	✓	-✓	✓	✓	✓	GitHub	C#
UnrealRox+	Martinez-Gonzalez et al. (2021)	2021	✓	✓	✓	✓	–	✓	–	✓	✓	–	✓	GitHub	Python

Georgakis et al. (2017) apply this method in a household setting. The process of placing the objects of interest is automated and based on the depth image of the background scene. After placing the object, it is scaled according to its location. Sagues-Tanco et al. (2020) present a similar approach using kitchen objects and scenes. They present a fully automatic approach where the objects are placed at random positions in scenes and a manual approach where the objects are placed at meaningful positions by hand.

Summary and discussion. Although these approaches lead to successful results in object detection, they are limited by available images and different perspectives from the objects of interest. Moreover, they are limited by available ground truth options. For example, using a crop-out-based method for normal generation is simply not possible. In terms of scalability, an image from the object of interest has to be available. Furthermore, different viewpoints from the object of interest are only supported when multiple images of different viewpoints are available. In addition to image-based crop-out methods, computer-aided design (CAD) models used in 3D modeling-based, Graphic API-based or 3D game engine-based methods ensure the availability of multiple viewpoints at one object.

3.3. Graphic API-based methods

Graphic API-based methods use, e.g., OpenGL and either render full complex scenes (Hodaň et al., 2019) or use the 3D models from objects of interest and place them on images from already available

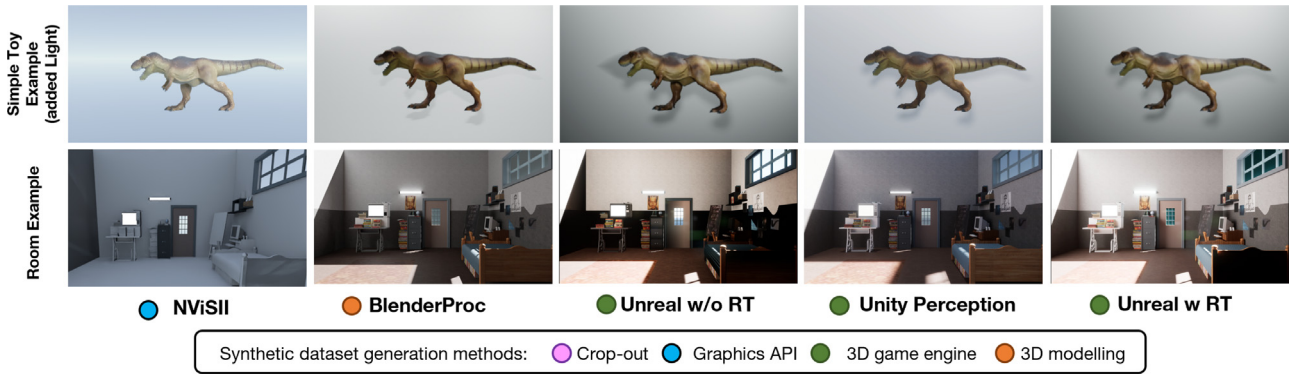


Fig. 4. Example renderings of the reviewed methods. As objects for rendering we used objects from Objaverse (Deitke et al., 2023). The camera pose is set to the same position for all renderings, however, not all approaches allow modifying all camera parameters to enable exactly the same view. For approaches using Unreal we show it without ray tracing (w/o RT) and with ray tracing (w RT). Note that for NVISII, additional adaption of material loading is necessary.

datasets (Hinterstoisser et al., 2018; Mercier et al., 2019). The 3D models are 3D meshes from the objects of interest. Often the 3D meshes are referred to as CAD models. However, all of the approaches use meshes and not original CAD data.

Hinterstoisser et al. (2018) compared the use of real and synthetic images using Faster R-CNN (Ren et al., 2015). To render the images they used OpenGL and placed the objects at random on background images. For more variety in their data, the background image which contains a cluttered scene is randomly selected. For training their CNN they tested several configurations with different backbones or freezing of layers, as well as domain randomization methods. Their test on real data showed that a CNN trained purely on synthetic data can achieve a mean average precision (mAP) over 60%, close to their model trained on real data at around 70%. However, the generated images are not very realistic in terms of context information.

In contrast to this approach, Hodaň et al. (2019) generate images considering context information and aim to target object detection and 6D pose estimation. They generated low and high-quality images using physically based rendering (PBR) and images with high-quality objects rendered out of context. The 3D objects are rendered in six 3D scenes using realistic material and lighting conditions. To initialize the object pose, the rule-based FLARE (Gal et al., 2014) system was applied. Physics were simulated with NVIDIA PhysX to arrange the objects in plausible locations. The camera parameters were randomized with a focus on one random object in the scene. In their experiments with Faster R-CNN, low and high-quality PBR images achieved a significantly better mAP compared to the approach of Hinterstoisser et al. (2018). Their results demonstrate the importance of context information. High-quality PBR images compared to low-quality images only improved the results if the scene contains complex lighting and materials.

For 6D pose estimation, the SIXD toolkit (Hodaň et al., 2017) was modified (Mercier et al., 2019) to render many objects at once and capture various viewpoints. SIXD already provides the generation of RGB-D images. To add background images, they used images from an existing dataset (Song and Xiao, 2015). In their experiment, they used synthetic and weakly labeled images. Their results show that their CNN trained on a combination of these images performs equally to, e.g., PoseCNN (Xiang et al., 2018) while requiring less and only sparse annotated real-world annotated data.

A more reusable method is introduced by Morrical et al. (2021) with NVIDIA Scene Imaging Interface (NVISII). NVISII enables the generation of images using logic from existing synthetic dataset generation approaches (Hinterstoisser et al., 2019; Tobin et al., 2017; Tremblay et al., 2018b). It further allows the manipulation of materials, textures, light, volumetric data, and backgrounds. As denoted in Table 1, it is the only Graphics API approach with available source code for reproducibility. NVISII also provides an export option to the NVIDIA Deep Learning Data Synthesizer (NDDS) (To et al., 2018) format, which is usable in the game engine Unreal. For their experiments, they followed previous approaches (Hinterstoisser et al., 2019; Tobin et al., 2017; Tremblay et al., 2018b) and generated three datasets of 6k images each. They observed that following these previous approaches and combining them improves the performance significantly compared to using only a single generation method.

Summary and discussion. These graphic API-based approaches show that synthetic data can achieve decent results on real-world data. The approach of Hinterstoisser et al. (2018) was methodology wise still close to crop-out-based methods. Therefore, the use of 3D context could only be used in a limited fashion. Hodaň et al. (2019) utilized more of the 3D world and demonstrated the importance of context information compared to randomly placed objects. Considering the success of synthetic data one important aspect is domain randomization. Morrical et al. (2021) utilized one common method which is the combination of randomized and realistic data. This randomization method can lead to a higher accuracy on real-world data.

3.4. 3D modeling-based methods

The 3D modeling tool Blender provides many aspects like animation or texture loading. It is a Python API that enables to access the scene context and scripts everything which is also possible in the user interface. Thus the 3D modeling tool allows one to generate ground truths for diverse tasks.

Talukdar et al. (2018) used Blender and an object dictionary of household objects to investigate object detection. Their approach automatically controlled light conditions and the camera pose. For training their CNN, they applied various data augmentation options such as Gaussian noise, Gaussian blur, or random-crop. Their experiments showed that transfer learning can be applied by using synthetic data for training and real-world images for testing.

More diverse ground truth options are provided by Denninger et al. (2020) with BlenderProc. The provided scripts are extendable and customizable. BlenderProc enables loading objects from external sources or existing datasets, the exchange of objects, adaption of light, camera position, materials, and object placement. Possible randomization methods are random 3D background objects with random texture, occlusion, and various lighting conditions. In comparison to previous works e.g., Hinterstoisser et al. (2018), it requires more time to generate images due to the focus on realism (Denninger et al., 2020) and, in turn, higher computational costs. One advantage of BlenderProc is that it provides outputs in common formats such as COCO (Lin et al., 2014)-, BOP (Hodaň et al., 2020)- or the SUN CG (Song et al., 2017) dataset format. To generate 6D pose, ground truth data BlenderProc is frequently used (Hodaň et al., 2020). Additionally, SynthDet (Jhang et al., 2020) leverages this approach for an end-to-end object detection model.

Table 2
Synthetic datasets for 3D objects which provide object recognition, 6D pose estimation or semantic segmentation ground truth data.

Name	Authors	Year	Sem. seg.	2D Obj. det.	6D Pose	Category	Tool	Objects	Source (Link)
Semantic scene understanding									
–	Papon and Schoeler (2015)	2015	✓	–	(✓)	3D modeling	Blender/ BlenSor	CAD	–
SceneNet	Handa et al. (2016a)	2016	✓	–	–	Graphic API	OpenGL	CAD	GitHub
SceneNet RGB-D	McCormac et al. (2017)	2017	✓	–	–	Graphic API	OpenGL	CAD	ProjectPage
THEODORE	Scheck et al. (2020)	2020	✓	–	–	3D Game engine	Unity	–	ProjectPage
Hypersim	Roberts et al. (2021)	2021	✓	✓	✓	Graphic API	V-Ray, Python	3D assets	GitHub
GeoSynth	Pugh et al. (2023)	2023	✓	(✓)	–	3D modeling	Blender	3D assets	GitHub
Object detection and *6D pose estimation*									
FAT	Tremblay et al. (2018b)	2018	✓	✓	✓	3D Game engine	NDDS	YCB	ProjectPage
WISDOM-Sim	Danielczuk et al. (2019)	2019	✓	✓	–	Graphic API	OpenGL	CAD	ProjectPage
SIDOD	Jalal et al. (2019)	2019	✓	✓	✓	3D Game engine	NDDS	YCB	ProjectPage
NOCS	Wang et al. (2019)	2019	–	✓	✓	3D Game engine	Unity	ShapeNetCore	ProjectPage
Synthetic HOPE	Shi et al. (2021)	2021	✓	–	✓	3D Game engine	NDDS/NViSII	daily objects	GitHub
UOAIS-SIM	Back et al. (2022)	2023	✓	(✓)	–	3D modeling	Blender	3D assets	GitHub
SynTable	Ng et al. (2023)	2023	✓	(✓)	–	Graphic API	Isaac Sim	CAD/YCB	–

Similar to Blenderproc, Greff et al. (2022) present a Python package named Kubric leveraging Python, PyBullet, and Blender to load assets from an external source and generates high-quality ground truth data. It is further extendable for custom use cases. Kubric focuses on scalability during rendering to enable efficient data generation. It can be scaled from a local workflow up to running large jobs.

Summary and discussion. The summarized 3D modeling tools BlenderProc and Kubric provide diverse ground truth and randomization options. 3D modeling tools using an object dictionary (Talukdar et al., 2018) need to be extended individually per dataset. In addition to the previous graphic API-based approaches except for NVISII, BlenderProc, and Kubric addresses diverse tasks and provide a broader set for customization. BlenderProc and Kubric have the advantage that datasets can be generated directly with known formats.

3.5. Game engine-based methods

Due to the rise of available high-performant game engines, such as Unity or Unreal Engine, it is rational that these were applied to create synthetic data. Borkman et al. (2021) introduced Unity Perception which provides a comparable amount of ground truth options to BlenderProc or Kubric, see Table 1. It further provides key points which could be used for human-pose estimation, supports a variety of domain randomization methods, and is customizable. To evaluate their approach, they generated a synthetic and a real-world groceries dataset. Faster R-CNN purely on synthetic images underperformed compared to real-world images. In contrast to the pure real-world baseline, the combination of 400,000 synthetic and 760 real-world images performed best and improved mAP by almost 20%. The combination of 100,000 synthetic and real images performed only 2.2% worse compared to the use of 400,000 synthetic images.

Qiu and Yuille (2016) presented the Unreal Engine-based UnrealCV enabling the manipulation of 3D scenes and includes options to interface with common machine learning frameworks like Caffe. A Python API is provided which adds the option to generate ground truth data, for example, for semantic segmentation tasks.

NVIDIA Deep Learning Data Synthesizer (NDDS) (To et al., 2018) is another 3D game engine-based approach, integratable in to Unreal. NDDS enables the generation of ground truth data for semantic segmentation, depth images, object poses, bounding boxes, and custom options. Domain randomization is possible via different lighting conditions, varying the camera position or textures of objects. In this approach, rasterization is applied instead of ray tracing, while other approaches support ray tracing.

Other approaches using Unreal are UnrealROX (Martinez-Gonzalez et al., 2019) and UnrealROX+ (Martinez-Gonzalez et al., 2021). UnrealROX was developed to generate synthetic data for virtual reality scenes while UnrealROX+ focuses on generating ground truth data for diverse tasks. Besides the ground truth options denoted in Table 1, UnrealRox+ is able to generate hand-joint estimation ground truth data. It further provides a Python API to directly integrate it into the deep learning process, for example, for reinforcement learning.

Summary and discussion. When using 3D game engines Unity Perception, NDDS-based on Unreal and UnrealRox+ provide the most ground truth options directly. Although they almost provide the same options, Unity Perception provides the most under the game engine-based approaches.

3.6. Summary

In summary, crop-out-based, graphic API-based, 3D modeling, and game engine-based approaches showed promising potential and sophisticated proof of principles. Fig. 3 outlines that 3D modeling and 3D game engine-based approaches are on the rise. Potentially due to the fact that they are easily extendable and already provide a stack of ground truth options, see Table 1. NVISII represents the latest approach of a graphic API-based method from 2021. Crop-out-based methods are used less frequently. They also provide the least ground truth options and flexibility. All methods could result in specific datasets. NDDS (Jalal et al., 2019; Shi et al., 2021; Tremblay et al., 2018b), NVISII (Shi et al., 2021) and BlenderProc (Denninger et al., 2020; Hodaň et al., 2020) led to publicly available datasets.

4. Synthetic datasets

The reviewed synthetic data approaches enable a use-case-specific dataset generation. In addition, manifold synthetic datasets for object detection, 6D pose estimation and scene understanding exist or have been generated using the presented approaches (Tremblay et al., 2018b; Jalal et al., 2019; Shi et al., 2021). The objects or scenes in these datasets are often either CAD models or real-world 3D reconstructions, see Table 2.

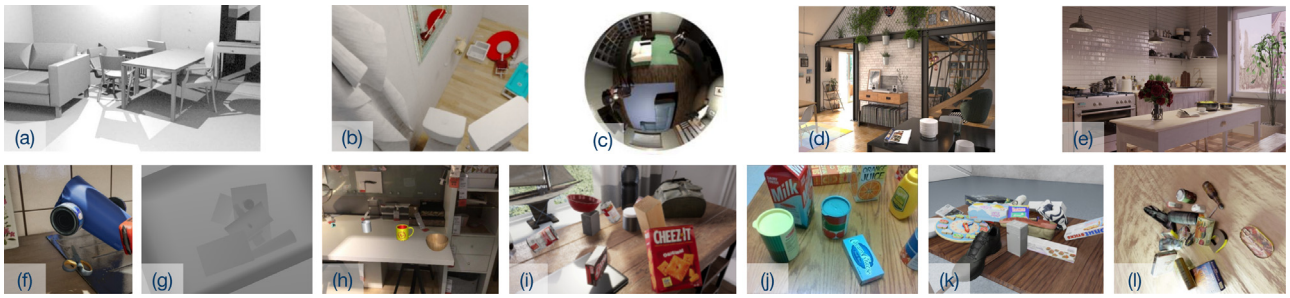


Fig. 5. Example images of the reviewed indoor semantic scene understanding (top), object detection and pose estimation datasets (bottom). (a) Papon and Schoeler (2015), (b) SceneNet/SceneNet RGB-D, (c) THEODORE, (d) Hypersim, (e) GeoSynth, (f) FAT, (g) WISDOM-Sim (Danielczuk et al., 2019), (h) NOCS, (i) SIDOD, (j) HOPE, (k) SynTable and (l) UOAIS-SIM.

4.1. Categorization

The reviewed datasets can be categorized into datasets that target semantic scene understanding and datasets that target object detection/6D pose estimation.

Scene Understanding: We define scene semantic understanding datasets as datasets focusing on providing pixel-wise ground truth values for semantic or instance segmentation.

Object detection/6D pose estimation: Object detection/6D pose estimation datasets focus on individual objects instead of the full-scene. The datasets can contain semantic segmentation ground truth as well, as the bounding boxes for object detection can be derived from segmentation masks.

4.2. Semantic scene understanding

To generate indoor synthetic datasets different methods have been utilized. For example, game engine-based approaches (Scheck et al., 2020) as well as graphic API-based methods (Papon and Schoeler, 2015; Handa et al., 2016a; McCormac et al., 2017), see Table 2. Papon and Schoeler (2015) combine random placement and object cues to add object models to virtual rooms. With the context cues, the aim is the realistic placement of the objects. For shadow effects, as depicted in Fig. 5, the position of the light is randomized.

Other approaches follow the placement of existing 3D models in scenes. SceneNet (Handa et al., 2016a) is an indoor RGB-D dataset. It is generated fully automatic using a graphic API. The featured objects were from existing datasets (Chang et al., 2015; Wu et al., 2015) and placed using object co-occurrence statistics. Their position was optimized using simulated annealing (Handa et al., 2016b). Comparing the data of SceneNet and real-world data led to the decision to add noise to the depth maps. Their experiment demonstrated that pretraining on 10k synthetic data can improve the performance on real-world datasets. In subsequent work, SceneNet RGB-D (McCormac et al., 2017) provides per-pixel annotations for videos instead of key-frames, photorealistic RGB textures, and more than 15k configuration possibilities. Besides that difference, 5M images were annotated and trajectory ground truth is provided.

While previous approaches followed the standard pinhole camera, THEODOR (Scheck et al., 2020) is captured with a fish-eye camera. THEODOR is a semantic segmentation fish-eye indoor dataset containing 16 classes in 100,000 images. The dataset contains top-down views of rooms including human avatars. THEODOR is captured with 3D game engine, Unity, and domain randomization is applied with varying cameras, light positions and varying textures. Depending on the task and in the end used real-data using different camera models (pinhole, fish-eye) can be helpful.

Focusing on photorealism, Hypersim (Roberts et al., 2021) provides scenes for holistic indoor scene understanding. This dataset relies on publicly available 3D assets (Evermotion, 2023) and the scene were created by professional artists. In total, the dataset features 77,400 images of 461 indoor scenes. For all scenes, the geometry and lighting information as well as semantic instance per-pixel annotations are available.

Similarly, Pugh et al. (2023) introduce GeoSynth a highly photorealistic indoor dataset on 18 195 scenes generated with Blender. It covers several ground truths like semantic segmentation, instance segmentation, depth, surface normals, lighting, reflection, environment map, etc. As 3D assets, assets from Evermotion (Evermotion, 2023), similar to ones used in Hypersim are used and assets from the IKEA catalog.

Summary and discussion. Many deep learning architectures scale by more data (Smith et al., 2023; Kang et al., 2023), datasets like SceneNet RGB-D are beneficial.

Findings from these datasets can be applied to generate new ones, e.g., the post-processing method to add noise to the depth map. Existing 3D modeling-based approaches like BlenderProc also enable adding noise to the generated depth already at the generation time. In addition, the noise factor can be randomized, to reduce the overfitting on one specific depth map type. By randomizing the noise factor, it is assumed that changing the input cameras in the real-world is more bearable.

Comparing the datasets for scene understanding, it appears, that the more recent datasets became more photorealistic. This is attributed to the assets used in the scene.

4.3. Object detection and 6D pose estimation datasets

falling things (FAT) (Tremblay et al., 2018b) features 21 YCB objects in 61k frames split into two parts. One part with single objects while the other part contains mixed objects. The objects are rendered in a photorealistic manner using physics and placed in a virtual environment. For the camera pose, azimuth angle, elevation angle, and distance are selected at random from a predefined interval.

Similarly, SIDOD (Jalal et al., 2019) contains 21 YCB, placed in three realistic scenes. From these scenes, 18 varying viewpoints are used to capture 144k frames. For domain randomization, random lighting, camera poses as well as a random placement of the individual objects were chosen. The objects follow their gravity to place them on the ground. Additionally, the flying distractors i.e., cubes or cylinders, are placed randomly in the scene.

The Normalized Object Coordinate Space (NOCS) dataset (Wang et al., 2019), includes real and 400k synthetic images. The synthetic scenes contain real background images with tabletop scenes and synthetic objects from ShapeNetCore (Chang et al., 2015). The objects are rendered in a table-top manner, see Fig. 5. As ground truth, RGB, depth, 6D pose, NOCS maps and ground truth masks of the objects are available.

Synthetic HOPE (Shi et al., 2021) contains 28 scans of daily objects, scanned with an EinScan-SE 3D Scanner. For each object, 60,000 images were generated using different levels of occlusion and five varying light settings for domain randomization (Tremblay et al., 2018a). For training, 3D object models are available to generate data.

WISDOM-SIM (Danielczuk et al., 2019) contains 50,000 synthetic depth images generated with OpenGL using objects from Thingiverse. The synthetic images are split into 40,000 training images with 1280 objects and 10,000 test images with 320 objects. To sample the synthetic images a set of foreground and background objects is used. While the camera pose and foreground object poses vary, the background objects are fixed.

Using the 3D modeling tool BlenderProc, Back et al. (2022) generate UOAIS-Sim. The dataset contains of 50,000 RGB-D images of 1000 cluttered scenes. As objects they use 3D models from *The Kit Object Database* (Kasper et al., 2012), *Bigbird* (Singh et al., 2014) and from *the BOP challenge* (Hodaň et al., 2020). The dataset mainly features household objects like a bottle or a cereal box. As scene setup planes or bins where selected and the objects where placed at random on the randomly textured surface.

Another approach builds upon the NVIDIA Omniverse platform using Isaac Sim Replicator Composer (Ng et al., 2023). SynTable (Ng et al., 2023) is a table top dataset with instance segmentation masks. The object covered in the dataset are 1075 CAD models from the Google Scanned Objects dataset (Downs et al., 2022) and the BOP benchmark (Hodaň et al., 2020).

Summary and discussion. The datasets mainly focus on household objects. While some address domain randomization (Tremblay et al., 2018b; Jalal et al., 2019). Others, focus on table-top scenes (Back et al., 2022) where the sim-to-real transfer can be questionable.

4.4. Summary

Generating synthetic object detection/6D pose estimation datasets is often validated using the YCB benchmark or other datasets containing household objects (Rennie et al., 2016; Kaskman et al., 2019; Shi et al., 2021; Jung et al., 2022). To ensure the transfer from the household domain to other domains such as industry or medical contexts, a benchmark with specular, small, and very similar-looking objects would be challenging. Recent findings prove the challenge of transparent or specular objects within 6D pose estimation (Gao et al., 2021; Wang et al., 2022). Industry tasks are already considered (Hodan et al., 2017; Drost et al., 2017). However, their data creation process is not fully synthetic and a time consuming capturing process is used instead of CAD models, 3D meshes and pure synthetic data.

5. Addressing the challenges of synthetic datasets

Synthetic data have the potential to compensate for missing real-world datasets. However, CNNs trained solely on synthetic data often exhibit suboptimal performance in real-world scenarios. This discrepancy is attributed to the so-called sim-to-real gap, which arises due to differences between modeled scenes and real-world conditions. Domain randomization (Alghonaim and Johns, 2021) or domain adaptation (Csurka, 2017; Wilson and Cook, 2020; Figueira and Vaz, 2022) are employed to address this gap.

5.1. Categorization

Domain adaption: Domain adaption tackles the challenge of adapting a network trained on a source domain to work on an unseen target domain. This is in general used to adapt domains for deep learning models and is not limited to synthetic data (Wilson and Cook, 2020).

Domain randomization: Domain randomization randomizes scenes and extends the data distribution intending to enhance a better generalization of CNNs. More detailed information on domain adaption can be found in Csurka (2017), Wilson and Cook (2020), Figueira and Vaz (2022), while we only provide an outline of domain adaption approaches.

5.2. Domain adaption

In the generation of synthetic datasets, domain adaptation involves amalgamating real-world images with synthetic objects. Typically, the GAN architecture (Goodfellow et al., 2014) is commonly applied for this purpose (Csurka, 2017; Ho et al., 2021; Wilson and Cook, 2020; Figueira and Vaz, 2022). For instance, RetinaGAN (Ho et al., 2021) utilizes CycleGAN (Zhu et al., 2017) to bi-directionally map real-world and synthetic images. The dataset utilized for this purpose comprises a mixture of both real-world and synthetic images. To generate synthesized images, a physics engine is employed to realistically position the objects.

Within the medical domain, GANs are utilized to augment available training data. For instance, Han et al. (2018) trained a GAN specifically to synthesize magnetic resonance images. Moreover, alongside the classical GAN framework that operates solely on images, text-to-image models have emerged as an extension.

In addition to GANs, diffusion models are promising for data generation. For example, Saharia et al. (2022) introduced *Imagen*, a text-to-image diffusion model. *Imagen* leverages large transformer language models capable of understanding text, coupled with diffusion models that generate photorealistic images. Although these approaches show impressive results in learning synthesized data or even to exploit text-to-image, they are often used in a more artistic manner than towards a specific use-case creation of synthetic data.

Besides GAN-based or diffusion model-based approaches, Kar et al. (2019) introduce a graph convolutional network (GCN)-based approach denoted as Meta-Sim. Meta-Sim utilizes probabilistic scene grammar to generate valid and diverse 3D worlds. The scene grammar provides a scene graph, which is fed into Meta-Sim. This GCN builds upon two matrices to capture top-down and bottom-up information. The output is a transformed scene graph. They optimize the GCN based on a maximum threshold between the real-world and synthesized scene graph.

Summary and discussion. For CNN based domain adaption, potential challenges may be that CNNs learn the underlying distribution of data. Therefore, CNN-based generated datasets could suffer from bias in data distribution, while previously introduced rendering approaches can easily overcome this drawback by providing an equal data distribution.

5.3. Domain randomization

Domain randomization in principle extends the basic approaches for synthetic data generation by expanding and varying the number of parameters to be manipulated, see Fig. 6. As categorized in Table 3, it can be categorized by purely data-based methods and by methods using domain randomization along training a CNN.

5.3.1. Categorization

Data-based methods: Purely data-based methods are cateogrized by the fact that domain randomization takes place during data generation, i.e. before the training process,

Training-based methods: Training-based methods can be identified by the fact that the randomization is adjusted during the training or by the actual training progress of a CNN.

5.3.2. Data-based methods

Tobin et al. (2017) hypothesized that CNNs exhibit enhanced generalization when confronted with highly randomized data. To test this hypothesis, they trained an adapted VGG-16 model using basic geometric shapes. Randomization was introduced through the inclusion of distractor objects, variations in object positions and textures in the environment, and adjustments to the camera's position and field of view. Their study demonstrated that augmenting the level of randomization led to a decrease in detection errors associated with the objects.

Fig. 6. Examples of domain randomization generated with BlenderProc (Denninger et al., 2020). Up from a starting scene setup *(a)*, the scene is modified and randomized using different camera and object poses *(c, f)*, material and textures *(d, e)* and randomization of the surrounding *(b, g, h)*. As distractors *(h)* additional textured objects from the YCB benchmark and geometric objects are used.

Table 3
Summary of methods used for domain randomization. The ✓ denotes if this domain randomization method is available in this approach. A (✓) denotes varying backgrounds but with the limitation to 2D scenes. The - notes either no information provided or not used in the method. We group the approaches by year. Additionally, we provide information about the randomization of background, object poses, textures, material, camera pose, and the use of distractors. Ani et al. (2021) especially investigated the influence of textures, therefore, they only randomized textures.

Author	Year	Method	Tool	Background	Object pose	Textures	Material	Camera pose	Light	Distractors
Data-based										
Tobin et al. (2017)	2017	–	–	✓	✓	✓	–	✓	–	✓
Tremblay et al. (2018a)	2018	3D Game engine	Unreal Plugin	–	✓	✓	–	.	✓	✓
Tremblay et al. (2018c)	2018	3D Game engine	NDDS	✓	✓	✓	–	–	–	✓
Ani et al. (2021)	2021	–	–	–	–	✓	–	–	–	–
Alghonaim and Johns (2021)	2021	3D Game engine	Unity	✓	–	✓	–	✓	✓	–
Training-based										
Heindl et al. (2021)	2021	3D modeling	Blender	–	✓	–	–	✓	–	✓
Hagelskjær and Buch (2022)	2022	3D modeling	BlenderProc, Python	(✓)	–	(✓)	–	–		
Zakharov et al. (2022)	2022	–	Neural DR	–	–	–	✓	–	✓	–
Horváth et al. (2023)	2023	3D modeling	PyBullet	(✓)	✓	✓	–	✓	–	✓

The use of distracting objects and randomization is further investigated by Tremblay et al. (2018a). They randomly placed objects of interest in 3D scenes. To only learn the object of interest from a scene, further objects are added as 'flying distractors'. For all objects in the scene, random textures are applied, and the position and number of light sources as well as the camera position are randomized. Their experiments showed that the flying distractors increases the network performance and varying light sources had a positive impact on the reported mAP.

Plausible physical rendering and domain randomization are useful as well. Tremblay et al. (2018c) apply various distractors, overlaid textures, varying backgrounds, different object positions and noise. In their photorealistic setup 21 YCB objects follow their gravity. Their CNN trained with highly randomized and photorealistic images outperformed the real-world only approach.

Summary and discussion. While Tobin et al. (2017) and Tremblay et al. (2018c) use various randomization methods, Ani et al. (2021) investigated the randomization of textures and ranked them based on Wasserstein (Dimitrakopoulos et al., 2020) and Fréchet (Heusel et al., 2017) distance metrics. They demonstrated the effectiveness of ranking textures to succeed in localization tasks based on synthetic data. Alghonaim and Johns (2021) benchmarked the use of background, textures and distractors. They observed that adding varying textures and background led to a robust model in unseen environments with new textures and distractors.

5.3.3. Training-based methods

The direct integration of domain randomization into CNN training is approached with BlendTorch (Heindl et al., 2021). It combines photorealistic image generation using Blender with deep learning in PyTorch. Similarly, Hagelskjær and Buch (2022) integrate the domain randomization in the training process. As a randomization method, they augment the data using Gaussian XYZ noise, Gaussian normal vector noise, Gaussian RGB noise, Gaussian RGB shift, rotation, and flattening. To improve the training process they first use no domain randomization in the first epochs and afterwards add domain randomization during the training process.

Horváth et al. (2023) introduce another work using PyBullet for image generation, focusing not on highly realistic but on a well-randomizable setup. Several parameters are configurable like the number of distracting objects and textures.

Moreover, Zakharov et al. (2022) implemented a neural domain randomization based on a ray tracer approximater. As input, a G-buffer (XYZ, normals) is used and the neural approximator produces different materials and light conditions in a simple geometric setup. The outputs of the approximater are combined using a tone mapper.

Summary and discussion. For 6D pose estimation synthetic data generation is often used for pretraining the CNN. However, synthesized and randomized data can often not reach real-world performance in that task (Tremblay et al., 2018c; Tobin et al., 2017; Tremblay et al., 2018a). Therefore, investigating domain randomization even more and adapt it for example during training (Heindl et al., 2021; Hagelskjær and Buch, 2022; Zakharov et al., 2022) showed that it is indeed possible to reach real-world performance with domain randomized data.

Table 4
The previous graphic API-, 3D modeling- and 3D game engine-based approaches enable various domain randomization methods. The ✓ denotes if this domain randomization is available in this approach. A (✓) denotes that the domain randomization method is available with limitations.

Method	Author	Background	Object pose	Textures	Material	Camera pose	Light	Distractors
Crop-out-based								
–	Georgakis et al. (2017)	(✓)	✓	–	–	(✓)	–	–
–	Sagues-Tanco et al. (2020)	(✓)	✓	–	–	(✓)	–	–
Graphic API-based								
–	Hinterstoisser et al. (2018)	(✓)	✓	–	–	–	–	(✓)
–	Mercier et al. (2019)	(✓)	✓	–	–	–	–	–
–	Hodaň et al. (2019)	(✓)	✓	–	–	–	✓	✓
NViSII	Morrical et al. (2021)	✓	✓	✓	✓	✓	✓	✓
3D Modeling-based								
–	Talukdar et al. (2018)	✓	–	–	–	✓	✓	–
BlenderProc	Denninger et al. (2020)	(✓)	✓	✓	✓	✓	✓	✓
Kubric	Greff et al. (2022)	✓	✓	✓	✓	✓	✓	✓
3D Game engine-based								
UnrealCV	Qiu and Yuille (2016)	✓	✓	(✓)	(✓)	✓	✓	✓
NDDS	To et al. (2018)	✓	✓	✓	–	✓	✓	✓
Unity Perception	Borkman et al. (2021)	✓	✓	✓	✓	✓	✓	✓
UnrealRox+	Martinez-Gonzalez et al. (2021)	✓	✓	✓	–	✓	–	✓

5.4. Domain randomization in synthetic dataset generation methods

We further investigated how domain randomization is addressed in the reviewed crop-out, graphic API, 3D modeling, and game engine-based approaches, see Table 4. Some approaches directly enable the exchange and adaption of backgrounds and photorealism. Crop-out-based methods (Georgakis et al., 2017; Sagues-Tanco et al., 2020) enable randomization but with a limited level as the setup in detail is a 2D image where crop-outs are placed on background images. 3D Game engine-based methods and 3D modeling-based methods enable the most levels of randomization without additional adaptions. Especially when loading existing scenes and adapting them. Although NVISII also allows the manipulation of all parameters, the APIs of the 3D game-engine and 3D modeling methods allow the manipulation in a greater extend.

Summary and discussion. Especially crop-out-based methods limit the level of randomization. Different viewpoints on the object are only enabled if different crop-outs of the same image are available. Other graphic API-based, 3D modeling-based or game engine-based approaches enable a richer option of variations like changing the light conditions, textures or materials of the objects (Morrical et al., 2021; Talukdar et al., 2018; Denninger et al., 2020; Qiu and Yuille, 2016; To et al., 2018; Borkman et al., 2021; Martinez-Gonzalez et al., 2021). Unity Perception especially provides a randomization framework to apply various randomization methods.

5.5. Summary

In summary, various domain randomization methods have been proposed. Distractors and varying textures proved to be successful to minimize the sim-to-real gap (Tobin et al., 2017; Tremblay et al., 2018c) and can even reach real-world performance for specific tasks (Tremblay et al., 2018c). Utilizing distance metrics and ranking strategies or directly integrating it into the learning process seem even more promising (Ani et al., 2021; Heindl et al., 2021). Based on the evaluated methods it can be concluded that domain randomization is a valuable strategy for synthetic dataset generation improving the generalization of CNNs in the real world.

6. General discussion

In summary, we reviewed approaches to generate synthetic datasets based on systematic literature research and investigated publicly available synthetic datasets for semantic scene understanding, object detection, and 6D pose estimation. To overcome the sim-to-real gap, we reviewed domain adaption and especially domain randomization approaches.

Research Q_1 — What are common methods for indoor synthetic dataset generation? We identified four common approaches: (i) approaches that are building directly upon real crop-outs of objects, (ii) graphic API-based approaches (Hinterstoisser et al., 2018; Hodaň et al., 2019; Mercier et al., 2019; Morrical et al., 2021), (iii) approaches integrating 3D modeling tools (Talukdar et al., 2018; Denninger et al., 2020), and (iv) 3D game engine-based approaches (Qiu and Yuille, 2016; To et al., 2018; Martinez-Gonzalez et al., 2021). Graphic API-based approaches showed that synthetic data generation can perform equally well compared to real-world datasets (Hinterstoisser et al., 2018). Hodaň et al. (2019) outlined again the importance of context information for the results of CNNs which is also present for synthetic data. Although, these approaches showed promising results, more recent 3D modeling and game engine-based methods enable multifaceted ground truth options, domain randomization options, and adaption of scenes in general. While contextual information holds high importance for CNNs (Hodaň et al., 2019; Lin et al., 2017), Morrical et al. (2021) demonstrated that integrating realistic and randomized scenes further enhances performance. This effect has also been observed in other experiments employing highly realistic and strongly randomized images (Tremblay et al., 2018c). The adaptability of 3D modeling and game engine-based approaches enables the generation of both realistic and highly randomized images. Particularly, Unity Perception emphasizes domain randomization. In terms of synthetic data generation methods, the use of GANs (Han et al., 2018) or diffusion models (Saharia et al., 2022) is also mentionable. However, especially diffusion models are often used more in an artistic way and contain a lot of effort until a synthetic dataset can be made available. They contain the effort of training a CNN and potentially a language model. Graphic API-based, 3D modeling-based, and 3D game engine-based approaches instead can be leveraged directly as soon as 3D models are available. Furthermore, they enable the manipulation of various scene parameters (Xiang et al., 2022; Denninger et al., 2020; Borkman et al., 2021) like light conditions, textures or camera intrinsic and extrinsic parameters and everything can be controlled in a more automated fashion than during the training of a CNN.

Research Q_2 — Which synthetic datasets are mainly used for object detection, 6D pose estimation, and scene understanding? For semantic scene understanding e.g., SceneNet and SceneNet RGB-D were successful and helped improving results on real-world datasets. For object detection the YCB benchmark is mainly used to generate synthetic datasets (Tremblay et al., 2018b; Jalal et al., 2019). Purchasable household objects are also featured within synthetic HOPE (Shi et al.,

2021). These benchmarks make it easy to reproduce datasets and test the results in the real-world with varying setups. Through publicly available benchmarks a comparison of novel deep learning approaches is enabled. Even though synthetic data is often generated for a specific individual use, benchmarks are helpful for comparability and real-world tests. Looking ahead, there is a need to delve into industry-specific or medically-oriented benchmarks. Particularly, intricate tools and reflective surfaces could represent novel and challenging benchmarks, necessitating enhancements in rendering parameters–such as materials and reflections–within generation approaches.

Research Q_3 — Which challenges does synthetic data generation face? Research Q_4 — How are these challenges for synthetic data addressed? It has been noted that CNN exclusively trained on synthetic data often exhibit a performance loss when deployed in real-world scenarios (Tremblay et al., 2018a; Morrical et al., 2021). To mitigate this challenge, methods such as domain adaptation or domain randomization can be employed. Domain adaptation, while commonly applied to synthetic data, also finds utility in this context. Specifically, the utilization of GANs has demonstrated success in this domain (Bousmalis et al., 2017; Ho et al., 2021; Nikolenko et al., 2021; Figueira and Vaz, 2022). Nevertheless, the architecture of GANs inherently introduces a level of uncertainty regarding the generated outcomes, requiring extensive time for both training and testing phases. For domain randomization, different approaches have proven that randomizing textures, light conditions, camera orientations, and distractors help the generalization of CNNs trained on synthetic data (Alghonaim and Johns, 2021; Tobin et al., 2017; Tremblay et al., 2018a,c). Mainly the combination of highly randomized and photorealistic images (Tremblay et al., 2018c; Nowruzi et al., 2019; Morrical et al., 2021) improves the results and the use of distractors (Tremblay et al., 2018a). An argument could be made that realistic images bolster the contextual information utilized by CNNs, while randomization aids in highlighting the differences between the 3D models and the real-world environment. In future endeavors, more intricate 3D benchmarks could incorporate further randomization to enhance their efficacy (Deitke et al., 2023). The ranking of textures based on Wasserstein (Dimitrakopoulos et al., 2020) and Fréchet (Heusel et al., 2017) distance metrics improved localization tasks (Ani et al., 2021). Furthermore, varying camera parameters in general are crucial for robust applications. CNNs trained on one camera setup can underperform on another setup (Liu et al., 2020). This is a general issue of CNNs and does not only occur with models trained on synthetic data. Future research shall investigate not only varying extrinsic parameters but also varying intrinsic camera parameters. Another randomization method is the image quality. Blurred images caused by camera motion could be considered. Thereby, it shall be investigated if data augmentation using e.g., Gaussian blur is sufficient or if blurred images caused by camera motion lead to a more robust real-world applicability. The direct integration of domain randomization into the learning process is another promising future direction (Heindl et al., 2021). Domain randomization is also provided by the reviewed synthetic data generation approaches and considered for the generation of existing synthetic datasets (Jalal et al., 2019; Shi et al., 2021). Despite the promise exhibited by domain randomization, certain unresolved challenges persist and must be acknowledged when constructing synthetic datasets. Real-world image patterns may encompass contextual information that is either inadequately modeled or beyond the scope of representation within synthetic data. As synthetic data continues to evolve, future directions should explore whether domain randomization can surmount these limitations and address these drawbacks.

7. Conclusion and further remarks

We can categorize current approaches to synthetic data generation and synthetic datasets into crop-out-based, graphics API-based, 3D modeling-based, or game engine-based methods. Additionally, many of these approaches incorporate domain randomization to varying extents. Domain randomization primarily expands the data distribution during the data generation process. Our analysis indicates that combining highly randomized and photorealistic images yields more realistic real-world results. Furthermore, we observed that incorporating distractors with diverse textures is generally beneficial for effective randomization.

However, the optimal amount of synthetic data and the degree of randomization required to train a well-generalized model is yet to be investigated. While obtaining a generalized model might be achievable with unlimited data generation, the necessity to exhaust all available resources remains uncertain. Future work could go in several directions. Firstly, designing CNN architectures with enhanced robustness to effectively handle variations in data could be a valuable direction. Another direction could be a more benchmarkable synthetic data generation method to categorize the variation of randomization parameters better.

Efforts should focus on ensuring that synthetic data generation methods are quick and efficient to ensure scalability and compatibility with existing deep learning approaches. Overall, the adaptability and customizability of synthetic data approaches exhibit promising potential for individual applications and future advancements.

CRediT authorship contribution statement

Hannah Schieber: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Kubilay Can Demir:** Validation, Writing – review & editing. **Constantin Kleinbeck:** Visualization, Writing – review & editing. **Seung Hee Yang:** Supervision, Writing – review & editing. **Daniel Roth:** Conceptualization, Methodology, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hannah Schieber, Constantin Kleinbeck and Kubilay Can Demir report financial support was provided by a d.hip campus stipend.

Data availability

No data was used for the research described in the article.

Acknowledgments

We gratefully acknowledge funding for this study by d.hip campus and Bundesministerium für Bildung und Forschung (BMBF), Germany with the grant number 16SV8973.

References

Alghonaim, R., Johns, E., 2021. Benchmarking domain randomisation for visual sim-to-real transfer. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 12802–12808.

Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A., Rother, C., 2018. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. Int. J. Comput. Vis. (IJCV).

Ani, M., Basevi, H., Leonardis, A., 2021. Quantifying the use of domain randomization. In: International Conference on Pattern Recognition. pp. 6128–6135.

Back, S., Lee, J., Kim, T., Noh, S., Kang, R., Bak, S., Lee, K., 2022. Unseen object amodal instance segmentation via hierarchical occlusion modeling. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE, pp. 5085–5092.

Beery, S., Liu, Y., Morris, D., Piavis, J., Kapoor, A., Joshi, N., Meister, M., Perona, P., 2020. Synthetic examples improve generalization for rare classes. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 863–873.

Borkman, S., Crespi, A., Dhakad, S., Ganguly, S., Hogins, J., Jhang, Y.-C., Kamalzadeh, M., Li, B., Leal, S., Parisi, P., Romero, C., Smith, W., Thaman, A., Warren, S., Yadav, N., 2021. Unity perception: Generate synthetic data for computer vision.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D., 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Cabon, Y., Murray, N., Humenberger, M., 2020. Virtual KITTI 2. arXiv:2001.10773.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012.

Csurka, G., 2017. A comprehensive survey on domain adaptation for visual applications. In: Domain Adaptation in Computer Vision Applications. pp. 1–35.

Danielczuk, M., Matl, M., Gupta, S., Li, A., Lee, A., Mahler, J., Goldberg, K., 2019. Segmenting unknown 3D objects from real depth images using mask R-CNN trained on synthetic data. In: International Conference on Robotics and Automation (ICRA). pp. 7283–7290.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A., 2023. Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153.

Denninger, M., Sundermeyer, M., Winkelbauer, D., Olefir, D., Hodan, T., Zidan, Y., Elbadrawy, M., Knauer, M., Katam, H., Lodhi, A., 2020. BlenderProc: Reducing the reality gap with photorealistic rendering. In: Robotics: Science and Systems (RSS) Workshops.

Dimitrakopoulos, P., Sfikas, G., Nikou, C., 2020. Wind: Wasserstein inception distance for evaluating generative adversarial network performance. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3182–3186.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. In: Annual Conference on Robot Learning. pp. 1–16.

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V., 2022. Google scanned objects: A high-quality dataset of 3d scanned household items. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE, pp. 2553–2560.

Drost, B., Ulrich, M., Bergmann, P., Hartinger, P., Steger, C., 2017. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Workshops. pp. 2200–2208.

Evermotion, 2023. Evermotion web shop. URL https://evermotion.org/shop.

Figueira, A., Vaz, B., 2022. Survey on synthetic data generation, evaluation methods and GANs. Mathematics 10 (15), 2733.

Gaidon, A., Wang, Q., Cabon, Y., Vig, E., 2016. Virtual worlds as proxy for multi-object tracking analysis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4340–4349.

Gal, R., Shapira, L., Ofek, E., Kohli, P., 2014. FLARE: Fast layout for augmented reality applications. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 207–212.

Gao, D., Li, Y., Ruhkamp, P., Skobleva, I., Wysock, M., Jung, H., Wang, P., Guridi, A., Navab, N., Busam, B., 2021. Polarimetric pose prediction.

Ge, Y., Behl, H., Xu, J., Gunasekar, S., Joshi, N., Song, Y., Wang, X., Itti, L., Vineet, V., 2022. Neural-sim: Learning to generate training data with NeRF.

Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J., 2017. Synthesizing training data for object detection in indoor scenes.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. 27.

Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D.J., Gnanapragasam, D., Golemo, F., Herrmann, C., Kipf, T., Kundu, A., Lagun, D., Laradji, I., Liu, H.-T.D., Meyer, H., Miao, Y., Nowrouzezahrai, D., Oztireli, C., Pot, E., Radwan, N., Rebain, D., Sabour, S., Sajjadi, M.S.M., Sela, M., Sitzmann, V., Stone, A., Sun, D., Vora, S., Wang, Z., Wu, T., Yi, K.M., Zhong, F., Tagliasacchi, A., 2022. Kubric: a scalable dataset generator. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3749–3761.

Hagelskjær, F., Buch, A.G., 2022. ParaPose: Parameter and domain randomization optimization for pose estimation using synthetic data. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6788–6795. http://dx.doi.org/10.1109/IROS47612.2022.9981511.

Han, C., Hayashi, H., Rundo, L., Araki, R., Shimoda, W., Muramatsu, S., Furukawa, Y., Mauri, G., Nakayama, H., 2018. GAN-based synthetic brain MR image generation. In: IEEE International Symposium on Biomedical Imaging (ISBI 2018). pp. 734–738. http://dx.doi.org/10.1109/ISBI.2018.8363678.

Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R., 2016a. Understanding realworld indoor scenes with synthetic data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4077–4085.

Handa, A., Pătrăucean, V., Stent, S., Cipolla, R., 2016b. SceneNet: An annotated model generator for indoor scene understanding. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 5737–5743.

Heindl, C., Brunner, L., Zambal, S., Scharinger, J., 2021. BlendTorch: A real-time, adaptive domain randomization library. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (Eds.), Pattern Recognition. ICPR International Workshops and Challenges. pp. 538–551.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: International Conference on Neural Information Processing Systems. pp. 6629–6640.

Hinterstoisser, S., Lepetit, V., Wohlhart, P., Konolige, K., 2018. On pre-trained image features and synthetic images for deep learning. In: European Conference on Computer Vision (ECCV) Workshops.

Hinterstoisser, S., Pauly, O., Heibel, H., Martina, M., Bokeloh, M., 2019. An annotation saved is an annotation earned: Using fully synthetic training for object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV), Workshops.

Ho, D., Rao, K., Xu, Z., Jang, E., Khansari, M., Bai, Y., 2021. RetinaGAN: An object-aware approach to sim-to-real transfer.

Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X., 2017. T-LESS: An RGB-d dataset for 6D pose estimation of texture-less objects. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 880–888.

Hodaň, T., Michel, F., Sahin, C., Kim, T.-K., Matas, C., 2017. SIXD challenge 2017.

Hodaň, T., Sundermeyer, M., Drost, B., Labbé, Y., Brachmann, E., Michel, F., Rother, C., Matas, J., 2020. BOP challenge 2020 on 6D object localization. In: European Conference on Computer Vision Workshops (ECCVW).

Hodaň, T., Vineet, V., Gal, R., Shalev, E., Hanzelka, J., Connell, T., Urbina, P., Sinha, S., Guenter, B., 2019. Photorealistic image synthesis for object instance detection. In: IEEE International Conference on Image Processing (ICIP).

Horváth, D., Erdős, G., Istenes, Z., Horváth, T., Földi, S., 2023. Object detection using Sim2Real domain randomization for robotic applications. IEEE Trans. Robot. 39 (2), 1225–1243. http://dx.doi.org/10.1109/TRO.2022.3207619.

Jalal, M., Spjut, J., Boudaoud, B., Betke, M., 2019. SIDOD: A synthetic image dataset for 3D object pose recognition with distractors. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Workshops.

Jhang, Y.-C., Palmar, A., Li, B., Dhakad, S., Vishwakarma, S.K., Hogins, J., Crespi, A., Kerr, C., Chockalingam, S., Romero, C., Thaman, A., Ganguly, S., 2020. Training a performant object detection ML model on synthetic data using unity perception tools.

Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R., 2016. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?.

Jung, H., Zhai, G., Wu, S.-C., Ruhkamp, P., Schieber, H., Wang, P., Rizzoli, G., Zhao, H., Meier, S.D., Roth, D., Navab, N., et al., 2022. HouseCat6D–a large-scale multi-modal category level 6D object pose dataset with household objects in realistic scenarios. arXiv preprint arXiv:2212.10428.

Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T., 2023. Scaling up gans for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10124–10134.

Kar, A., Prakash, A., Liu, M.-Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S., 2019. Meta-sim: Learning to generate synthetic datasets. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4551–4560.

Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S., 2019. HomebrewedDB: RGB-d dataset for 6D pose estimation of 3D objects. In: IEEE/CVF International Conference on Computer Vision (ICCV), Workshops. pp. 2767–2776.

Kasper, A., Xue, Z., Dillmann, R., 2012. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. Int. J. Robot. Res. 31 (8), 927–934.

Kleinbeck, C., Schieber, H., Andress, S., Krautz, C., Roth, D., 2022. Artfm: augmented reality visualization of tool functionality manuals in operating rooms. In: 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). IEEE, pp. 736–737.

Koch, B., Denton, E., Hanna, A., Foster, J.G., 2021. Reduced, reused and recycled: The life of a dataset in machine learning research. arXiv preprint arXiv:2112.01716.

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A., 2019. AI2-THOR: An interactive 3D environment for visual AI.

Liberati, A., Altman, D.G., Tetzlaff, J., Mulrow, C., Gøtzsche, P.C., Ioannidis, J.P.A., Clarke, M., Devereaux, P.J., Kleijnen, J., Moher, D., 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. Ann. Int. Med. 151 (4), W65–94.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: European Conference on Computer Vision.

Liu, Z., Lian, T., Farrell, J., Wandell, B.A., 2020. Neural network generalization: The impact of camera parameters. IEEE Access 8, 10443–10454.

Martinez-Gonzalez, P., Oprea, S., Castro-Vargas, J.A., Garcia-Garcia, A., Orts-Escolano, S., Garcia-Rodriguez, J., Vincze, M., 2021. UnrealROX+: An improved tool for acquiring synthetic data from virtual 3D environments. CoRR.

Martinez-Gonzalez, P., Oprea, S., Garcia-Garcia, A., Jover-Alvarez, A., Orts-Escolano, S., Garcia-Rodriguez, J., 2019. UnrealROX: An eXtremely photorealistic virtual reality environment for robotics simulations and synthetic data generation.

McCormac, J., Handa, A., Leutenegger, S., Davison, A.J., 2017. SceneNet RGB-D: 5M photorealistic images of synthetic indoor trajectories with ground truth.

Mercier, J.-P., Mitash, C., Giguère, P., Boularias, A., 2019. Learning object localization and 6D pose estimation from simulation and weakly labeled real images. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3500–3506.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision (ECCV). pp. 405–421.

Morrical, N., Tremblay, J., Lin, Y., Tyree, S., Birchfield, S., Pascucci, V., Wald, I., 2021. NViSII: A scriptable tool for photorealistic image generation.

Ng, Z., Wang, H., Zhang, Z., Hock, F.T.E., au2, M.H.A.J., 2023. SynTable: A synthetic data generation pipeline for unseen object amodal instance segmentation of cluttered tabletop scenes.

Nikolenko, S.I., et al., 2021. Synthetic Data for Deep Learning. Springer.

Northcutt, C.G., Athalye, A., Mueller, J., 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. arXiv preprint arXiv:2103.14749.

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

Papon, J., Schoeler, M., 2015. Semantic pose using deep networks trained on synthetic RGB-D. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 774–782.

Paulin, G., Ivasic-Kos, M., 2023. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. Artif. Intell. Rev. 1–45.

Pugh, B., Chernak, D., Jiddi, S., 2023. GeoSynth: A photorealistic synthetic indoor dataset for scene understanding. IEEE Trans. Visual. Comput. Graph. 29 (5), 2586–2595. http://dx.doi.org/10.1109/TVCG.2023.3247087.

Qiu, W., Yuille, A., 2016. UnrealCV: Connecting computer vision to unreal engine. In: European Conference on Computer Vision Workshops (ECCVW). pp. 909–916.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: International Conference on Neural Information Processing Systems. pp. 91–99.

Rennie, C., Shome, R., Bekris, K.E., De Souza, A.F., 2016. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place. IEEE Robot. Autom. Lett. 1 (2), 1179–1185.

Richter, S.R., Hayder, Z., Koltun, V., 2017. Playing for benchmarks. In: IEEE International Conference on Computer Vision (ICCV). pp. 2213–2222.

Richter, S.R., Vineet, V., Roth, S., Koltun, V., 2016. Playing for data: Ground truth from computer games. In: European Conference on Computer Vision (ECCV). In: LNCS, Vol. 9906, pp. 102–118.

Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M., 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10912–10922.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M., 2016. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3234–3243.

Sagues-Tanco, R., Benages-Pardo, L., López-Nicolás, G., Llorente, S., 2020. Fast synthetic dataset for kitchen object segmentation in deep learning. IEEE Access 8, 220496–220506.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al., 2022. Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487.

Santara, A., Rudra, S., Buridi, S.A., Kaushik, M., Naik, A., Kaul, B., Ravindran, B., 2021. MADRaS: Multi agent driving simulator. J. Artificial Intelligence Res. 70, 1517–1555.

Scheck, T., Seidel, R., Hirtz, G., 2020. Learning from THEODORE: A synthetic omnidirectional top-view indoor dataset for deep transfer learning. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 932–941. http://dx.doi.org/10.1109/WACV45572.2020.9093563.

Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Pérez-D'Arpino, C., Buch, S., Srivastava, S., Tchapmi, L., Tchapmi, M., Vainio, K., Wong, J., Fei-Fei, L., Savarese, S., 2021. iGibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 7520–7527.

Shi, G., Zhu, Y., Tremblay, J., Birchfield, S., Ramos, F., Anandkumar, A., Zhu, Y., 2021. Fast uncertainty quantification for deep object pose estimation.

Singh, A., Sha, J., Narayan, K.S., Achim, T., Abbeel, P., 2014. Bigbird: A large-scale 3d database of object instances. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 509–516.

Smith, S.L., Brock, A., Berrada, L., De, S., 2023. ConvNets match vision transformers at scale. arXiv preprint arXiv:2310.16764.

Song, F.Y.Y.Z.S., Xiao, A.S.J., 2015. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T., 2017. Semantic scene completion from a single depth image. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1746–1754.

Talukdar, J., Biswas, A., Gupta, S., 2018. Data augmentation on synthetic images for transfer learning using deep CNNs. In: International Conference on Signal Processing and Integrated Networks (SPIN). pp. 215–219.

To, T., Tremblay, J., McKay, D., Yamaguchi, Y., Leung, K., Balanon, A., Cheng, J., Hodge, W., Birchfield, S., 2018. NDDS: NVIDIA deep learning dataset synthesizer.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P., 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 23–30.

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S., 2018a. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Workshops. pp. 969–977.

Tremblay, J., To, T., Birchfield, S., 2018b. Falling things: A synthetic dataset for 3D object detection and pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Workshops. pp. 2038–2041.

Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S., 2018c. Deep object pose estimation for semantic robotic grasping of household objects.

Wang, P., Jung, H., Li, Y., Shen, S., Srikanth, R.P., Garattoni, L., Meier, S., Navab, N., Busam, B., 2022. PhoCaL: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21222–21231.

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J., 2019. Normalized object coordinate space for category-level 6d object pose and size estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651.

Wilson, G., Cook, D.J., 2020. A survey of unsupervised deep domain adaptation. ACM Trans. Intell. Syst. Technol..

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shapes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1912–1920.

Xiang, S., Qian, D., Guan, M., Yan, B., Liu, T., Fu, Y., You, G., 2023. Less is more: Learning from synthetic data with fine-grained attributes for person re-identification. ACM Trans. Multimedia Comput., Commun. Appl. 19 (5s), 1–20.

Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H., 2020. SAPIEN: A SimulAted part-based interactive environment. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11097–11107.

Xiang, Y., Schmidt, T., Narayanan, V., Fox, D., 2018. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In: Robotics: Science and Systems (RSS).

Xiang, S., You, G., Li, L., Guan, M., Liu, T., Qian, D., Fu, Y., 2022. Rethinking illumination for person re-identification: A unified view. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4731–4739.

Zakharov, S., Ambruş, R., Guizilini, V., Kehl, W., Gaidon, A., 2022. Photo-realistic neural domain randomization. In: European Conference on Computer Vision. Springer, pp. 310–327.

Zhao, W., Queralta, J.P., Westerlund, T., 2020. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In: IEEE Symposium Series on Computational Intelligence. pp. 737–744.

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2223–2232.