Investigating the Impact of Domain Randomization on Synthetic-to-Real Deep Learning Performance

Arati Patil , Pritam Chaudhari , Ajinkya Shingne , Rohit Nalandwar , Ashish Pawar

Dr. D. Y. Patil Institute of Master of Computer Applications and Management , Akurdi.

ABSTRACT

Deep-learning vision models need large labelled datasets, and collecting and annotating real images is slow and expensive. Synthetic data from simulators and rendering pipelines offers a scalable alternative, but models trained only on synthetic images often lose accuracy on real ones, a problem known as the synthetic-to-real domain gap. Domain randomization (DR) attacks the gap by varying rendering parameters such as lighting, textures, backgrounds, camera viewpoint, and object pose so widely that the real world looks like one more variation of the training data. This paper is a critical narrative review of how DR affects synthetic-to-real performance, mainly in object detection. It compares ten studies on their synthetic-data approach, randomization factors, architectures, metrics, findings, and limitations.

Across these studies, models trained only on randomized synthetic data reached useful real-world accuracy in some constrained settings, and synthetic pretraining followed by a small amount of real-data fine-tuning often beat real-only training, although the margin can be small when real data is plentiful. Textures, materials, lighting, post-processing, and scene context are each reported as important in individual studies, but the rankings disagree, and differences in datasets, detectors, and metrics prevent a universal ordering of randomization factors. One study also shows that near-perfect synthetic validation scores can sit alongside noticeably lower real-world scores, so synthetic validation alone should not be trusted as a proxy for deployment performance. The review points to gaps in standardized evaluation, controlled factor-level ablation, and reproducibility.

Keywords: domain randomization, synthetic data, sim-to-real transfer, deep learning, computer vision, object detection, domain adaptation

1. Introduction

Deep learning has become the default approach in computer vision, and object detection is no exception. Its weakness is appetite: detectors need large, carefully annotated datasets, and collecting real images across different lighting conditions, viewpoints, and environments is expensive, slow, and error-prone [1]. Synthetic data rendered by graphics engines is an appealing answer, because it is cheap to scale and comes with exact annotations [1].

The difficulty is that a detector trained only on synthetic images often performs worse on real ones. This mismatch is known as the synthetic-to-real (or sim-to-real) domain gap [1]. Domain randomization (DR) takes a deliberately counterintuitive route to closing it. Rather than chasing photorealism, DR randomizes textures, lighting, backgrounds, and other rendering parameters so widely that real images look to the network like one more variation [2]. Tobin et al. [2] demonstrated the idea for robotic object

localization, and Tremblay et al. [3] extended it to bounding-box detection of cars. DR has since been used in industrial inspection, robotic manipulation, and driving scenes.

Several reviews already cover neighbouring ground: synthetic data in general [1], indoor synthetic data generation [4], and learning from randomized simulations for robot control [5]. Westerski and Fong [6] surveyed DR for object detection specifically, comparing published solutions against criteria of their own and reporting outcomes that range from moderate losses to clear gains relative to real-data training. What remains hard to see is how much of the reported evidence is actually comparable. Individual studies differ in simulators, object classes, detectors, baselines, and metrics, and relatively few isolate what each randomization factor contributes.

This review therefore takes a narrower, more evaluation-centred view. It extracts what ten studies with real-world test data report about their randomization factors, baselines, and metrics, and asks how far those reports support common claims about which factors matter and how well synthetic-only training works. Its main contribution is to make the limits of cross-study comparison explicit, separating claims with solid empirical backing from those that mostly reflect differences in how the studies were evaluated.

2. Problem Statement, Objectives, and Research Questions

2.1 Problem statement

DR is widely used to narrow the synthetic-to-real gap, yet published approaches differ in simulators, target objects, baseline models, and metrics, and few studies isolate the contribution of individual randomization factors. As a result it is unclear which factors drive generalization, whether switching on every available factor is always best, and whether performance on synthetic validation data says anything reliable about real-world success.

2.2 Research objectives

1. To describe how recent deep-learning studies use DR to generate synthetic training data.
2. To compare the randomization factors, model architectures, datasets, and evaluation metrics those studies employ.
3. To synthesize the benefits and limitations reported by the original authors, separating well-supported findings from tentative interpretations.
4. To identify methodological gaps in the evidence and suggest directions for future research.

2.3 Research questions

1. How does domain randomization affect generalization from synthetic training images to real-world images?
2. Which randomization factors have been studied, and what are their isolated and combined effects?
3. What do existing studies show about the reliability of synthetic validation performance as a predictor of real-world test performance?
4. How does synthetic-only training compare empirically with synthetic pretraining followed by real-data fine-tuning?
5. What limitations, methodological weaknesses, and research gaps remain in the published DR literature?

3. Theoretical Foundations

3.1 Synthetic data generation

Synthetic image data is produced by rendering engines and procedural pipelines that place 3D objects in virtual scenes and output RGB images together with automatically generated ground truth, such as depth maps and bounding boxes. Simulators such as AI2-THOR [7] and procedural pipelines such as BlenderProc2 [8] provide controllable visual environments for training vision systems, and CAD-based tools can generate and label randomized synthetic images for recognition tasks in manufacturing [9].

3.2 Domain shift and the synthetic-to-real gap

Domain shift occurs when the joint distribution of inputs and labels in the training (source) domain differs from that in the deployment (target) domain. In this review the synthetic-to-real gap G is measured indirectly, as the difference between a performance metric M on held-out synthetic data and the same metric on real data:

$$G = M_{syn} - M_{real} \tag{1}$$

For higher-is-better metrics, a positive G means the model does worse on real images. It is a convenient signed performance difference, not a complete measure of distribution shift, and it bundles together the effects of rendering fidelity, sensor noise, material properties, and scene context [1]. Two cautions apply. First, G depends on which real dataset is chosen: Prakash et al. [10] found that a detector trained on one real driving dataset (BDD100K) scored far below one trained on same-domain KITTI images when both were tested on KITTI, so gaps between real datasets can be large too. Second, because the synthetic validation set usually comes from the same generator as the training set, M_{syn} is often optimistic.

3.3 Domain randomization, domain adaptation, and domain generalization

In DR, a renderer R produces an image x from a scene s and nuisance parameters θ drawn from a broad, often uniform distribution:

$$x = R(s, \theta), \quad \theta \sim p(\theta) \tag{2}$$

Randomizing θ pushes the model toward features that stay stable across those nuisance factors. Structured domain randomization (SDR) narrows p(θ) to contextually plausible arrangements, for example by placing cars on roads rather than floating in the sky [10].

DR works at the data-generation stage, so it is often combined with real data later in the pipeline, for instance by fine-tuning on a small real set [3]. Domain adaptation (DA), by contrast, aligns source and target feature distributions during training and usually needs unlabelled target-domain images [11]. Domain generalization (DG) aims to learn representations that work on unseen domains with no target data at all, and DR is commonly viewed as a data-centric route to DG. One further point matters for what follows: some studies stretch the term DR to include aggressive training-time data augmentation, which blurs the line between randomizing the renderer and augmenting the images [12].

3.4 Object detection evaluation metrics

Detectors such as YOLO [13] and its successors predict class labels and bounding boxes. A predicted box B_p is compared with a ground-truth box B_g using intersection over union (IoU):

$$IoU = |B_p \cap B_g| \,/\, |B_p \cup B_g| \tag{3}$$

Performance is summarized with precision (P), recall (R), and average precision (AP). Mean average precision (mAP) averages AP over C classes:

$$P = TP / (TP + FP), \quad R = TP / (TP + FN), \quad mAP = (1/C) \Sigma_{c=1...C} AP_c \tag{4}$$

The metric mAP@0.5 counts a detection as correct when IoU ≥ 0.5, which tolerates loose localization. The metric mAP@0.5:0.95 averages over thresholds from 0.5 to 0.95 in steps of 0.05 and penalizes imprecise boxes far more heavily. This matters later, because the reviewed studies do not share one threshold: Tremblay et al. [3] evaluate AP at IoU 0.5, Prakash et al. [10] at 0.7, and most industrial studies report mAP@0.5 only.

4. Related Work and Literature Review

4.1 Foundational domain-randomization research

Tobin et al. [2] showed that a robotic object localizer could be trained purely on non-realistic, randomly textured simulated images and still reach roughly 1.5 cm localization error on real objects, without seeing a single real image. Tremblay et al. [3] carried the idea over to bounding-box detection of cars on KITTI. Their synthetic scenes added randomized lighting and a new ingredient, “flying distractors” (random geometric shapes that clutter the scene), and after fine-tuning on real KITTI images the detector beat one trained on real data alone.

4.2 Textures, materials, lighting, and viewpoint

Randomizing textures is the common baseline in DR pipelines [2][3][14][15], and several studies went on to ablate individual factors. In the one-factor-at-a-time ablation of Tremblay et al. [3], removing random textures from the target objects lowered AP from 73.7 to 69.0, fixing the lights lowered it to 67.6, and removing flying distractors cost about 1.1%. Borrego et al. [14] and Dehban et al. [15] isolated the contribution of components such as texture patterns and parametric shapes for object-category detection. In an industrial setting, Zhu et al. [16] identified material properties, rendering methods, post-processing, and distractors as the factors that mattered most. These findings overlap but do not agree on a ranking, a point taken up in Section 7.2.

4.3 Backgrounds, distractors, and structured randomization

Complex backgrounds and flying distractors push a model to rely on the object itself rather than its surroundings [3][17]. Hinterstoisser et al. [17] rendered objects over densely randomized 3D background models and used a pose curriculum so that every object was seen under all poses, training detectors for 64 retail objects that compared favourably with models trained on real data.

Fully random placement has a cost, though. Prakash et al. [10] point out that plain DR needs a great deal of data and struggles with small, occluded objects because it ignores context. Their SDR places objects according to scene structure and outperformed DR at every KITTI difficulty level. On the hardest level, AP at IoU 0.7 was 52.5 for SDR against 23.9 for DR, and 45.6 for a detector trained on real BDD100K images, yet still well below the 88.8 reached by a detector trained on real KITTI images.

4.4 Industrial and robotic object detection

Industrial scenes tend to be visually constrained, which suits DR. Using YOLOv4, Horváth et al. [18] reached 86.32% mAP@50 with no real training images and 97.38% with a single real image, on a manually annotated set of 190 real images. Zhu et al. [16] report YOLOv8 models trained only on

synthetic data reaching 96.4% mAP@50 on a public robotics dataset. Rawal et al. [19] approach the problem differently: for production environments with complex assemblies they argue that generic randomization may not be enough, and they compare basic generation procedures and their combinations, reporting improvements of up to 15% on real production images. Martínez-Arellano and Buck [9] describe a Blender- and CAD-based tool built specifically to study which randomized factors matter in manufacturing, with ablation studies left for future work. Newer preprints keep pushing headline numbers up on public robotics benchmarks; one reports 99.1% mAP@50 [20], which is a reminder that single-benchmark comparisons date quickly.

4.5 Related reviews and sim-to-real work beyond detection

Nikolenko [1] gives a broad overview of synthetic data, Schieber et al. [4] review indoor synthetic data generation using the PRISMA method, and Muratore et al. [5] review learning from randomized simulations in robot control. Vanherle et al. [21] ran a controlled study on a dataset with real images and synthetic copies in which particular aspects are randomized, which let them analyse what kinds of variation help. In control tasks, Peng et al. [22] and OpenAI [23] used dynamics randomization for sim-to-real transfer, and Sadeghi and Levine [24] trained a quadcopter to fly indoors using synthetic images alone. On the theory side, Chen et al. [25] analysed when DR can be expected to succeed. This review confines itself to visual object detection with real-world test data.

5. Review Methodology

This is a critical narrative review. Differences in datasets, baselines, and metrics rule out a quantitative meta-analysis for now, so a structured qualitative comparison is the most defensible approach.

Search strategy. Searches covered IEEE Xplore, ACM Digital Library, Scopus, and arXiv for publications available by October 2026, using combinations of “domain randomization”, “synthetic data”, “sim2real”, and “object detection”.

Study selection. Ten studies were selected purposively for the comparative core: [2], [3], [10], [12], [18], [16], [14], [17], [15], and [19]. A study qualified if it (i) applied or evaluated domain randomization or closely related randomized synthetic-data generation, (ii) tested on real-world images, and (iii) reported quantitative results. Two of the ten sit at the edge of this scope. Tobin et al. address object localization rather than detection but are included as the foundation of the field, and Rawal et al. emphasize generation procedures and domain knowledge alongside randomization. Borrego et al. and Dehban et al. share authors and, judging from the authors' public code, a common pipeline, so they should not be read as fully independent evidence. The remaining papers (reviews, robot-control studies, and theory) are cited for background only, and theoretical work without a real-world test set was excluded from the core. The search was not run as a PRISMA-style systematic search and the records screened and excluded were not counted, so the selection should be read as illustrative rather than exhaustive.

Data extraction. For each of the ten studies, the target application, synthetic-generation tools, explicit DR factors, neural architectures, evaluation metrics, baselines, and author-reported findings were extracted iteratively. All numbers come from the original publications as reported; none were reproduced.

6. Comparative Analysis

This section compares the ten core studies along four lines: applications and architectures, the randomization factors implemented, the reported outcomes, and the methodological limitations. Table 1 gives an overview, and the metrics in it should not be compared across rows.

6.1 Target applications and architectures

The studies span a range of applications. Tobin et al. [2] localized objects for robotic manipulation with a VGG-16-based network. Tremblay et al. [3] and Prakash et al. [10] both addressed car detection on KITTI: Tremblay et al. compared Faster R-CNN, R-FCN, and SSD, with Faster R-CNN performing best, while Prakash et al. used Faster R-CNN. Hinterstoisser et al. [17] worked on 64 retail objects, Rawal et al. [19] on parts and assemblies in a production setting, and Borrego et al. [14] and Dehban et al. [15], from the same group, on object-category detection, the latter explicitly with parametric shapes and synthetic textures. More recent studies mostly use YOLO variants: Horváth et al. [18] used YOLOv4, Zhu et al. [16] used YOLOv8 on manufacturing and robotics data, and Torquato Niño and Gardi [12] used YOLOv11 on a single object, a soup can, in a Kaggle competition.

Table 1. Overview of the ten core studies (values as reported by the original authors; metrics are not directly comparable)

Study	Task and real test data	Randomization emphasis	Headline result as reported
Tobin et al. [2]	Object localization for robotic manipulation; real objects on a physical setup	Non-realistic random textures, lighting, and camera position; no real training images	About 1.5 cm localization error on real objects (synthetic only)
Tremblay et al. [3]	Car detection on KITTI (500-image test subset for the main comparison)	Random textures, lighting, camera pose, and backgrounds; flying distractors	Synthetic-only comparable to Virtual KITTI; after fine-tuning on 6,000 real images, 98.5 AP@0.5, 2.1% above real-only
Borrego et al. [14]	Object-category detection with a small labelled real set (about 200 images)	Non-photorealistic randomization with component ablation	“25%” mAP improvement over a fine-tuning baseline (relative or absolute not stated in the abstract)
Prakash et al. [10]	Car detection on KITTI (1,500-image subset for comparisons)	Context-aware (structured) randomization of scene layout, lighting, and materials	KITTI Hard, AP@0.7: 52.5 (SDR) vs 23.9 (DR); SDR plus real data above real-only
Hinterstoisser et al. [17]	Detection of 64 retail objects on a challenging real evaluation set	Dense randomized 3D backgrounds; pose curriculum	Detectors compared favourably with real-data training at far lower annotation cost
Dehban et al. [15]	Multi-category detection; parametric shapes and synthetic textures	Component-wise study of DR	DR reported to help when combined with at least a small domain-specific real set
Horváth et al. [18]	Industrial objects; 190 manually annotated real images	Automated DR data generation; YOLOv4	mAP50: 86.32% zero-shot, 97.38% with one real image
Rawal et al. [19]	Parts and assemblies in a production environment	Basic generation procedures and their combinations (photorealistic rendering)	Up to 15% improvement from combining procedures (relative or absolute not stated in the abstract)
Zhu et al. [16]	Manufacturing parts; a public robotics dataset and three industrial use cases	Materials, background, illumination, camera, post-processing; distractors	mAP@50 (synthetic only): 96.4% public dataset; 94.1%, 99.5%, 95.3% on three use cases
Torquato Niño and Gardi [12]	Single-object (soup can) detection; 159 real test images	Varied synthetic scenes plus heavy training-time augmentation	mAP@50: 0.995 synthetic validation vs 0.877–0.952 manual real test, 0.885–0.910 hidden competition test

6.2 Implementation of randomization factors

The scope of DR runs from basic texture swapping to elaborate procedural pipelines. Tobin et al. [2] and Tremblay et al. [3] set the common recipe of non-realistic textures, random lighting, and random camera viewpoints, and Tremblay et al. added flying distractors to clutter and occlude the scene. Borrego et al. [14] and Dehban et al. [15] focused their ablations on texture patterns and parametric shapes. Prakash et al. [10] broke from uniform sampling with context-aware object placement. Hinterstoisser et al. [17] paired a pose curriculum with dense 3D backgrounds. The most recent industrial studies use multi-factor pipelines that vary material properties, backgrounds, illumination, camera settings, and post-processing together [16], while Rawal et al. [19] compare generation procedures for parts and assemblies. Torquato Niño and Gardi [12] lean heavily on training-time augmentation as their form of randomization.

6.3 Reported empirical outcomes

All ten studies report that synthetic data helps, but under very different conditions and metrics, so the figures below are best read one study at a time.

Synthetic-only (zero-shot) performance.

- Tobin et al. [2] reached about 1.5 cm localization error on a physical robot setup without real training images.
- Tremblay et al. [3] found that a detector trained on randomized data alone was comparable to one trained on the much more realistic Virtual KITTI set. With Faster R-CNN, the best architecture, it scored 78.1 AP against 79.7 for Virtual KITTI; with R-FCN and SSD the randomized data did better.
- Prakash et al. [10] showed that SDR-only training beat real data from a different domain (BDD100K) on KITTI, although it remained well below real same-domain data.
- Horváth et al. [18] reported 86.32% mAP@50 on industrial objects. Zhu et al. [16] reported 96.4% mAP@50 on a public robotics dataset and 94.1%, 99.5%, and 95.3% on three industrial use cases.
- Torquato Niño and Gardi [12] reached 0.98–0.995 mAP@50 on synthetic validation sets, but 0.877–0.952 on a manually labelled real test set and 0.885–0.910 on the competition's hidden test set.

Mixed training (synthetic pretraining plus real-data fine-tuning).

- Tremblay et al. [3] fine-tuned on real KITTI images and reached 98.5 AP with all 6,000 real images, which they report as 2.1% better than real data alone.
- Prakash et al. [10] found SDR plus real KITTI data consistently above real KITTI alone, with the improvement especially pronounced when only a few labelled real images were available.
- Horváth et al. [18] raised mAP@50 from 86.32% to 97.38% using a single real image (one-shot transfer).
- Borrego et al. [14] reported a 25% mAP improvement over a fine-tuning baseline when only about 200 labelled images were available.

Rawal et al. [19] report something slightly different: up to 15% improvement from combining synthetic generation procedures, which concerns how the synthetic data is built rather than how it is mixed with real data.

6.4 Methodological limitations and threats to validity

Ambiguous effect sizes. The abstracts of Borrego et al. [14] and Rawal et al. [19] do not say whether their reported gains of 25% and up to 15% are relative improvements or absolute percentage points, which hides the true effect size.

Small or narrow test sets. Real test sets are often small: Horváth et al. [18] evaluated on 190 real images and Torquato Niño and Gardi [12] on 159. Tremblay et al. [3] ran their main comparison on 500 KITTI images and only on objects taller than 40 pixels with little truncation. Several studies also concentrate on a single class, such as cars [3][10] or a soup can [12], which limits what can be said about multi-class generalization in unconstrained scenes.

Inconsistent metrics and baselines. AP at IoU 0.5 [3], AP at IoU 0.7 [10], and mAP@50 [18][16][12] are not interchangeable, and the real-data baselines differ from study to study. Even the competition scoring in [12] used a method the authors say was not public, and their own manual labels ranked the configurations differently from the organizers' scores.

Overlap and blurred definitions. Borrego et al. [14] and Dehban et al. [15] come from the same group and appear to share a pipeline, so they are less independent than two separate studies would be. Torquato Niño and Gardi [12] treat aggressive training-time augmentation as a form of domain randomization, which makes it hard to credit gains to the rendering engine alone.

7. Critical Discussion

7.1 Efficacy of domain randomization and real-data fine-tuning

Several studies report high real-world detection accuracy in constrained industrial and robotic settings [18][16]. Those numbers depend on the object set, the real test data, the detector, and the training configuration, and they should not be extrapolated to open-world scenes. In more open settings such as driving, synthetic pretraining followed by real fine-tuning gave the best results [3][10].

How large the benefit is seems to depend on how much real data exists. With all 6,000 real KITTI images the advantage reported by Tremblay et al. [3] was about 2.1%, whereas Prakash et al. [10] describe the gain as especially pronounced with few real images, and the one-shot and roughly 200-image results in [18] and [14] point the same way. In practice, DR looks most valuable exactly where real labels are scarce, which is also where it is most often applied.

7.2 Isolating the effect of randomization factors

Per-study ablations do exist [3][10][16][14][15], but they change one factor at a time, run on different datasets and detectors, and do not agree on a ranking. Tremblay et al. [3] found that lighting and textures mattered most and flying distractors least. Prakash et al. [10] note that lighting was the most important parameter in that earlier DR study, but for SDR they found random saturation had the largest effect, with scene context and variety also important. Zhu et al. [16] name material properties, rendering methods, post-processing, and distractors, and do not single out lighting. Distractors are a minor factor in one study [3] and an important one in another [16]. Viewpoint is similarly unsettled: in the soup-can competition, removing distant camera views looked helpful on visual inspection but the full set with those views scored best on the hidden test [12].

What the literature lacks, then, is not ablation as such but factorial designs that estimate interactions, shared benchmarks with fixed baselines, and cost-aware reporting. Without them it is impossible to say

whether every factor is needed or whether some only add rendering cost. Textures, materials, lighting, post-processing, and scene context are each plausible candidates, but this review does not find evidence for a universal ranking.

7.3 How reliable is synthetic validation?

The clearest evidence comes from Torquato Niño and Gardi [12]. In their cross-validation, synthetic mAP@50 stayed at 0.98–0.99 for every augmentation configuration, and all four final models in their summary table scored 0.995 on the synthetic validation set. On the manually labelled real test set the same four models scored between 0.877 and 0.952, and on the competition's hidden test set between 0.885 and 0.910. Because the synthetic score was identical across models while the real scores differed, it could not tell the configurations apart, whereas real test data could. The authors themselves describe synthetic validation metrics as poor predictors of real-world performance.

Two qualifications keep this finding in proportion. It comes from a single study on a single object class, and the authors report no correlation analysis, so it is a documented cautionary case rather than a general law. And even the real-data estimates were not stable: the authors' manual labels and the organizers' scoring ranked some configurations differently, for example 0.877 against 0.905 for the standard split and 0.952 against 0.910 for the expanded dataset. A plausible explanation is that synthetic validation data comes from the same generator as the training data and so measures fit to the generator's distribution rather than robustness to real sensor noise and lighting, but this remains an interpretation. The practical lesson is to evaluate sim-to-real models on independent, manually labelled real data rather than on synthetic validation scores alone.

7.4 Summary of evidence by research question

Table 2. What the reviewed evidence supports

Research question	What the evidence supports	Strength of evidence
RQ1: Generalization effect	Detectors trained on randomized synthetic data reach useful real-world accuracy in constrained settings; in open scenes they usually stay below same-domain real training unless fine-tuned.	Moderate: consistent direction, heterogeneous metrics
RQ2: Randomization factors	Textures, materials, lighting, post-processing, and context are each reported as important somewhere; no stable ranking, and interactions are uncharacterized.	Weak to mixed
RQ3: Synthetic vs real validation	One study shows uninformative, near-perfect synthetic validation alongside lower and varying real-world scores.	Limited: one study, one class
RQ4: Synthetic-only vs fine-tuned	Pretraining plus fine-tuning gives the best absolute results; the margin shrinks as real data grows.	Moderate
RQ5: Gaps	Standardized benchmarks, factorial ablations, clearer effect-size reporting, and reproducible pipelines (Section 8).	Not applicable

Strength ratings are the author's judgement, based on the number, independence, and comparability of the supporting studies.

8. Research Gaps and Future Directions

The reviewed evidence points to several concrete directions:

- **Standardized synthetic-to-real benchmarks.** Shared real test sets with fixed baselines, reporting mAP@0.5:0.95 alongside mAP@0.5 so that tight localization is assessed and not just the forgiving 0.5 threshold.

- **Controlled, factorial ablation.** Experiments that toggle randomization variables (for example lighting and texture) both independently and jointly, to estimate effect sizes and interactions, with variability reported across random seeds.
- **Real-data efficiency curves.** Systematic maps of real-world accuracy against the number of real images used in fine-tuning, from zero and one image up to hundreds, to identify minimum requirements for deployment.
- **Larger, independent real test sets.** Manually labelled real test data with uncertainty estimates, and a clear separation between renderer-side randomization and training-time augmentation.
- **Open reproducibility.** Sharing simulator random seeds, rendering scripts, and parameter ranges so that results can be verified across studies.

9. Limitations of This Review

The review is limited by the heterogeneity of its sources. Differences in architectures, object classes, and IoU thresholds preclude a quantitative meta-analysis, and ambiguous reporting, such as not distinguishing absolute percentage points from relative improvements [14][19], limits how deeply studies can be compared. The core set of ten studies was chosen purposively rather than through a counted, PRISMA-style search, all figures are taken from the original papers and were not reproduced, and some details for four of the studies rest on abstracts. Finally, publication bias is likely: failed DR configurations rarely get published, so negative findings are probably under-represented.

10. Conclusion

Domain randomization is a cost-effective way to narrow the synthetic-to-real gap in object detection. By exposing networks to non-photorealistic variation in textures, lighting, backgrounds, and viewpoint during data generation, it can produce detectors that transfer to physical scenes, and in several constrained industrial settings synthetic-only training has delivered strong results. The evidence reviewed here suggests that mixed training, meaning synthetic pretraining followed by a little real fine-tuning, is the more dependable route when scenes are open-ended, although its advantage shrinks as real data becomes plentiful.

What the evidence does not yet support is a ranking of randomization factors or a trustworthy synthetic-only validation protocol. Ablations change one factor at a time, metrics and thresholds differ, test sets are small, and a well-documented case shows near-perfect synthetic validation scores alongside noticeably lower real-world ones. Progress will need shared real test sets, factorial ablations, and transparent rendering pipelines.

Declarations

Author contributions: Pritam Chaudhari , Ajinkya Shingne , Rohit Nalandwar , Ashish Pawar conceptualized the study, conducted the literature search, analyzed the comparative data, and drafted the manuscript. Mrs. Arati Patil provided structural feedback, supervision, and editorial review.

- **Funding:** This research received no external funding.
- **Competing interests:** The authors declare no competing interests.

- **Data availability:** No new data were generated. All information was extracted from the cited publications.
- **Use of AI tools:** AI tools were used for language editing, formatting, and cross-checking citations and reported figures against the source papers. The authors reviewed the results and remain responsible for all content.

References

[1] S. I. Nikolenko, "Synthetic data for deep learning," arXiv:1909.11512, 2019.

[2] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2017, pp. 23–30.

[3] J. Tremblay et al., "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2018, pp. 969–977.

[4] H. Schieber, K. C. Demir, C. Kleinbeck, S. H. Yang, and D. Roth, "Indoor synthetic data generation: A systematic review," Comput. Vis. Image Underst., vol. 240, Art. no. 103907, Mar. 2024, doi: 10.1016/j.cviu.2023.103907.

[5] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," Front. Robot. AI, vol. 9, Art. no. 799893, Apr. 2022, doi: 10.3389/frobt.2022.799893.

[6] A. Westerski and F. W. Teck, "Synthetic data for object detection with neural networks: State of the art survey of domain randomisation techniques," ACM Trans. Multimedia Comput. Commun. Appl., doi: 10.1145/3637064.

[7] E. Kolve et al., "AI2-THOR: An interactive 3D environment for visual AI," arXiv:1712.05474, 2017.

[8] M. Denninger et al., "BlenderProc2: A procedural pipeline for photorealistic rendering," J. Open Source Softw., vol. 8, no. 82, Art. no. 4901, 2023, doi: 10.21105/joss.04901.

[9] G. Martínez-Arellano and M. G. Buck, "A tool for generating and labelling domain randomised synthetic images for object recognition in manufacturing," presented at the 2nd Eur. Symp. Artif. Intell. Manuf. (ESAIM), Athens, Greece, Oct. 2024.

[10] A. Prakash et al., "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2019, pp. 7249–7255.

[11] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in Proc. 32nd Int. Conf. Mach. Learn. (ICML), 2015, pp. 1180–1189.

[12] L. Torquato Niño and H. A. A. Gardi, "Synthetic-to-real object detection using YOLOv11 and domain randomization strategies," arXiv:2509.15045, 2025.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.

[14] J. Borrego, A. Dehban, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Applying domain randomization to synthetic data for object category detection," arXiv:1807.09834, 2018.

[15] A. Dehban, J. Borrego, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "The impact of domain randomization on object detection: A case study on parametric shapes and synthetic textures," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2019.

[16] X. Zhu et al., "Domain randomization for object detection in manufacturing applications using synthetic data: A comprehensive study," arXiv:2506.07539, 2025.

[17] S. Hinterstoisser, O. Pauly, H. Heibel, M. Marek, and M. Bokeloh, "An annotation saved is an annotation earned: Using fully synthetic training for object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), 2019, pp. 2787–2796, doi: 10.1109/ICCVW.2019.00340.

[18] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, and S. Földi, "Object detection using Sim2Real domain randomization for robotic applications," IEEE Trans. Robot., vol. 39, no. 2, pp. 1225–1243, Apr. 2023, doi: 10.1109/TRO.2022.3207619.

[19] P. Rawal, M. Sompura, and W. Hintze, "Synthetic data generation for bridging Sim2Real gap in a production environment," arXiv:2311.11039, 2023.

[20] J. M. Araya-Martinez et al., "SynthRender and IRIS: Open-source framework and dataset for bidirectional sim-real transfer in industrial object perception," arXiv:2602.21141, 2026.

[21] B. Vanherle, S. Moonen, F. Van Reeth, and N. Michiels, “Analysis of training object detection models with synthetic data,” in Proc. British Mach. Vis. Conf. (BMVC), 2022.

[22] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2018, pp. 3803–3810.

[23] OpenAI: M. Andrychowicz et al., “Learning dexterous in-hand manipulation,” Int. J. Robot. Res., vol. 39, no. 1, pp. 3–20, 2020 (arXiv:1808.00177).

[24] F. Sadeghi and S. Levine, “CAD2RL: Real single-image flight without a single real image,” in Proc. Robot.: Sci. Syst. (RSS), 2017.

[25] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, “Understanding domain randomization for sim-to-real transfer,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2022.