Advanced Engineering Informatics 62 (2024) 102699

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

A systematic review and evaluation of synthetic simulated data generation strategies for deep learning applications in construction

Liqun Xu [a], Hexu Liu [b], Bo Xiao [c], Xiaowei Luo [d], DharmarajVeeramani [e], Zhenhua Zhu [f,*]

[a] Department of Civil and Environmental Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, WI 53706, USA
[b] Department of Civil and Construction Engineering, Western Michigan University, Kalamazoo, MI, USA
[c] Department of Civil, Environmental, and Geospatial Engineering, Michigan Technological University, Houghton, MI 49931, USA
[d] Department of Architecture and Civil Engineering, City University of Hong Kong, Hong Kong, China
[e] Department of Industrial and Systems Engineering, University of Wisconsin-Madison, 1513 University Avenue, Madison, WI 53706, USA
[f] Mortenson Company Assistant Professor Department of Civil and Environmental Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, WI 53706, USA

ARTICLE INFO

Keywords:
Synthetic simulated data (SSD)
SSD generation strategies
Guideline
Construction automation
Deep learning

ABSTRACT

The integration of deep learning (DL) into construction applications holds substantial potential for enhancing construction automation and intelligence. However, successful implementation of DL necessitates the acquisition of substantial data for training. The acquisition process can be error-prone, time-consuming, and impractical. For this reason, synthetic simulated data (SSD) has emerged as a promising alternative. While various strategies have been developed to generate such data, a systematic review and evaluation are lacking to aid researchers and professionals in selecting appropriate strategies for their applications. To fill this gap, this paper conducts a comprehensive literature review related to SSD generation and applications, and develops a guideline for strategy selection. Two hundred and eight articles are identified from the academic database Web of Science by using PRISMA. After thoroughly analyzing the literature, seven SSD generation strategies are identified and evaluated across six metrics. Based on the performance of each strategy, a guideline is synthesized as a decision tree. Users only need to follow the steps and answer the questions in the decision tree, and then they will get the recommended SSD generation strategy. We demonstrate the guideline's effectiveness by comparing its recommendations with the strategies chosen by researchers in existing DL construction applications and achieve a matching rate of 82%.

1. Introduction

The construction industry faces complex challenges, such as cost and time overruns, health and safety concerns, reduced productivity, and labor shortages [1,2]. The construction industry is among the least digitized industries in the world [3,4]. Integrating deep learning (DL) in construction automation is a promising development. It can revolutionize the construction industry and improve construction safety, quality, and productivity [5,6]. Potential applications of DL in construction include, but are not limited to, surveillance [7], progress monitoring [8], safety management [9], quality inspections [10], resource utilization management [11], and construction robotics [12]. For example, DL models detect the features of workers (e.g., hat shape, clothing color, etc.) in construction scene videos to determine whether they are wearing personal protective equipments (PPEs) [13]. If it is detected that workers not meeting PPE regulations, they will be marked so that subsequent reminders can be sent to them.

However, DL models require a substantial amount of high-quality labeled data for the training of the models to produce desirable results [14,15], and these data are difficult to obtain [16,17]. Traditionally, training data are collected from real-world sources and labeled manually. This approach is costly [18,19], labor-intensive [19], and time-consuming [20]. When collecting data from the real world and label them manually, the labeling results are also prone to errors and often inconsistent [21] due to the lack of labelers who have expertise in both construction and DL domains. Data collection may also pose a risk of confidentiality breaches [22,23], particularly when sensitive information is involved. Additionally, data availability is often restricted in the

* Corresponding author.
E-mail addresses: lxu322@wisc.edu (L. Xu), hexu.liu@wmich.edu (H. Liu), boxiao@mtu.edu (B. Xiao), xiaowluo@cityu.edu.hk (X. Luo), raj.veeramani@wisc.edu (DharmarajVeeramani), zzhu286@wisc.edu (Z. Zhu).

https://doi.org/10.1016/j.aei.2024.102699
Received 21 February 2024; Received in revised form 1 July 2024; Accepted 5 July 2024
Available online 11 July 2024
1474-0346/

real world [24], making it difficult to collect sufficient training examples of rare events. This predicament is particularly pronounced within the construction industry, where a lack of established standard schema and practices for data sharing severely constrains access to real-world construction data [25].

The use of synthetic simulated data (SSD) has emerged as a promising solution to the aforementioned challenges [26]. A simulation is an imitative representation of a system that could exist in the real-world [27]. Synthetic data generated using simulation methods offers inherent flexibility in replicating complex real-world scenarios, facilitating controlled experiments and algorithm validation [28]. Furthermore, SSD is relatively easy to generate, limitless, pre-annotated, and less expensive [29–32]. It enables the generation of data that may be infeasible or impractical to attain in reality [19,23,33,34]. Therefore, it has the potential to circumvent ethical concerns and practical issues, such as privacy and security [35].

Despite the advantages of SSD, the existing research lacks a systematic evaluation of SSD generation strategies. This gap leaves potential users unclear about the performance of various strategies, making it difficult to select the most suitable strategies for specific applications [36]. To address this gap, this study aims to identify and evaluate SSD generation strategies through an exhaustive survey. Further, based on the survey results, this study aims to establish a guideline for selecting appropriate SSD generation strategies. To achieve these objectives, we reviewed a total of 208 articles about SSD generation and applications retrieved from the Web of Science. Seven SSD generation strategies were identified within these articles. Additionally, six metrics were utilized to evaluate these strategies, and their performance was compiled into a table. A guideline for selecting the most suitable SSD generation strategy was developed based on the performance of the SSD generation strategies. To demonstrate the effectiveness of our proposed guideline, a comparative analysis was conducted by comparing the SSD generation strategies employed in the existing literature with those recommended by our guideline. We randomly selected eleven SSD application cases from the literature. In each application case, its requirements and assets availability were identified, based on which we can obtain the recommended SSD generation strategy using the decision tree proposed in our guideline. Then, we analyzed the SSD generation method in the application case and figured out its strategy adopted by the authors. If that strategy is the same as our recommended one, a match was noted. Otherwise, it was identified as a not match. In total, we got 9 out of 11 matches which is approximately 82%. Finally, for the literature that adopts inconsistent SSD generation strategies, we discuss the performance of the strategies used in the literature and the strategies recommended by the guidelines under the evaluation metrics.

2. Methodology

To achieve a comprehensive synthesis of existing research on SSD generation and applications, this study conducts a systematic review. Within this review, we will outline the various SSD generation strategies available, identify the metrics used to evaluate these strategies, assess their performance, and ultimately develop a guideline for selecting appropriate SSD generation strategy.

2.1. Literature retrieval strategy

The PRISMA [37] was used for literature retravel to ensure the comprehensiveness and relevance of the literature. In addition, the snowballing strategy [38] was employed as a supplementary method to ensure the literature's comprehensiveness. This two-pronged approach to literature retrieval aimed to minimize the risk of omitting pertinent studies. In the literature retrieval process, it is essential to establish several elements of the search strategy to ensure transparency and reproducibility of the review. These include 1) selecting suitable search engines to access scientific databases, 2) implementing specific search limits to restrict the number of retrieved results, and 3) establishing exclusion criteria to include only the pertinent publications in the final analysis. Fig. 1 depicts the PRISMA [37] process used to find published articles on the topic and how we decided to include them in our study.

In the identification stage, the Web of Science [39] was chosen as the primary search database due to its extensive coverage of scholarly journals and conference proceedings across a variety of academic fields [40]. To facilitate the search process, the search string was formulated utilizing Boolean operators: existing publications related to SSD were retrieved by searching "Title/Abstract/Keywords" with the string "TS= ((("synthetic image") OR ("synthetic data") OR ("synthetic dataset") OR ("synthetic sample") OR ("image synthesis") OR ("data synthesis") OR ("data generation") OR ("data augmentation")) AND ("simulation" OR "simulator" OR "simulating" OR "physical model"))". The latest studies on the subject were identified by selecting publications from the past six years. The search was conducted in January 2024 and a total of 1809 articles were obtained.

In the screening stage, articles were excluded using the following criteria to ensure relevance. (1) Publications did not pertain to the field of engineering and computer science. (2) Publications did not utilize simulation as the primary method for generating synthetic data. The screening process consists of two steps. First, publications in other fields were eliminated based on subject categories. Second, we examined the titles, keywords, and abstracts of the remaining publications to exclude those that did not use simulation methods to generate synthetic data. After these two steps of screening, 1513 articles were eliminated and 296 publications were retained for further analysis.

In the Eligibility stage, we took the articles remaining after the title, keyword, and abstract screening and read their full texts. 106 articles were excluded because they did not include content related to SSD generation and 190 articles left, which were saved in a marked list in Web of Science.

In the including stage, to minimize the risk of omitting pertinent studies, we conducted two types of iterations on the initial 190 articles, namely backward and forward snowballing. Since some articles obtained through the snowballing strategy cannot be retrieved on the Web of Science, Google Scholar [41] was also used at this stage. During the snowballing process, we went through two loops in total and obtained 18 articles. In the backward snowballing process of the first loop, all the cited articles by the initial 190 articles were saved in a marked list of Web of Science. Subsequently, in the forward snowballing process of the first loop, all articles that cite these 190 articles were also saved in the marked list. Then the articles that are duplicates of the 190 articles previously obtained were discard from the marker list. The remaining articles in the marked list were screened, leaving 18 articles. Similarly, the same process was used for these 18 articles in the second loop, but no new articles were found. In the end, 18 articles were obtained using the snowballing method, and a total of 208 articles, including 190 previously obtained articles, were included in this study. Note: All articles identified in this paper include content related to SSD generation and application, but due to the page limit, only some of them are presented in this paper.

2.2. Literature analysis and synthesis

After obtaining the 208 literatures, a comprehensive analysis of 208 pertinent scholarly literatures is conducted from four aspects, including the the application of SSD across various domains, the techniques used for SSD generation, the asset sources leveraged for SSD generation, and the evaluation of generated SSD. Each identified aspect was meticulously cataloged, culminating in the assembly of a tabulated database that serves as the foundational infrastructure of this review.

For the applications of SSD, we conducted a dual-track investigation. Initially, we investigated its deployment within the construction sector, followed by an expansive review across other domains, such autonomous driving and robotics. The reason why we study the application of

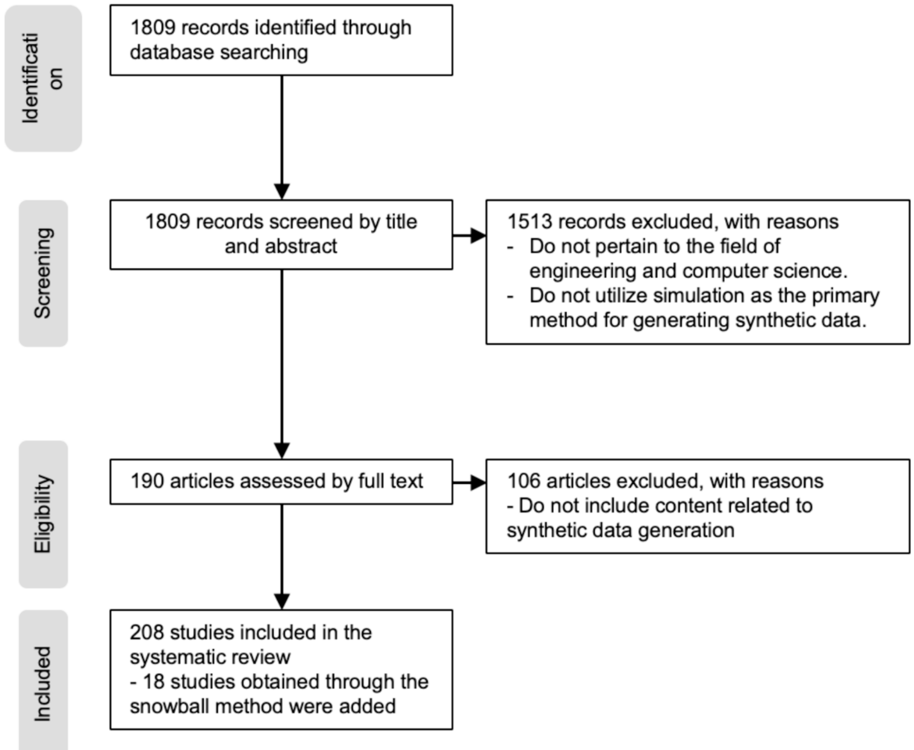


Fig. 1. PRISMA flow diagram of the literature search process for this study.

SSD in other fields is that its application in these fields precedes its application in the construction field and is characterized by the diversity and large number of use cases. We hope that insights drawn from these mature applications could potentially catalyze and shape the nascent adoption of SSD within construction. A summary of these applications is systematically presented in Section 3.1.

Progressing to the techniques and asset source used for SSD generation, our analysis is guided by an integral premise that generating SSD requires not only SSD generation techniques but also corresponding assets. Our literature survey found a total of 3 sources of assets and 3 SSD generation techniques, which combined to a total of seven different SSD generation strategies. At the same time, we summarize six evaluation metrics based on the evaluation of the generated SSD in the formulated tabulate database. Table 2 encapsulates a comparative analysis of the seven strategies, detailing their performance across the six metrics and delineating the technologies and asset sources employed by each.

Finally, since each strategy shows advantages under different evaluation metrics, it became imperative to devise a heuristic tool to aid practitioners in selecting the appropriate SSD generation strategy based on their application requirements. To this end, we opted for the construction of a decision tree. This decision tree is interactive, soliciting user input at successive decision nodes, each predicated on the evaluation metrics previously delineated. The resultant decision-making framework, which facilitates tailored strategy selection based on user responses, is explicated in Section 4.1.

Table 1
Evaluation metrics of SSD generation strategies.

Evaluation metric	Definition
Texture realism	Texture encompasses various characteristics and patterns found on the surfaces of objects. It can be categorized into three types: (a) color texture refers to various patterns and features on the surface of objects, such as marble walls; (b) geometry texture pertains to the micro-level geometry of surface textures such as uneven details on rock surfaces; (c) process texture involves diverse regular or irregular dynamic changes observed in natural phenomena, such as water and clouds [80]. Texture realism refers to the degree to which the texture properties of an object in an image accurately represent the texture properties of the real-world object being captured.
Relationship realism	Relationships refer to the spatial and visual arrangement of objects within an image or scene. They involve the relative positions, sizes, occlusions, and visual hierarchy of objects [81]. Relational realism refers to the degree to which the relationships between objects in an image accurately represent the relationships among the real-world objects being captured.
Motion realism	Motion is the movement of objects over time in a sequence of images or videos [78]. Motion realism is defined as the degree to which the movement of objects in a virtual environment accurately resembles the motion observed in the real world. It involves capturing and portraying the natural dynamics, timing, speed, and fluidity of movement [78,79].
Data variability	This encompasses the range of diverse values and patterns observable within a dataset, including variations in color, shape, texture, size, orientation, and other visual attributes [59].
Assets required	It refers to the source of assets needed to implement this strategy, including real, virtual and a mixture of the two.
Level of complexity	It refers to the relative rating of the complexity of implementing this strategy, which is divided into three levels: high, medium and low.

3. Review results

3.1. SSD applications

3.1.1. SSD applications in construction

SSD has been utilized in a variety of construction applications related to safety, productivity, and occupational ergonomics. Lee et al. [42] recognized two challenges in computer vision-based construction workers safety monitoring: the difficulty in obtaining a sufficient number of unsafe behavior images for a specific accident case, and the fact that the quality of annotation may be affected by the construction experience and capabilities of the annotator. To address these two

Table 2
Evaluation of generation strategies.

Strategy	Texture realism	Relationship realism	Motion realism	Data variability	Assets required	Level of complexity
Strategy-1	None is real	High	Low	High	Virtual	High
Strategy-2	None is real	Medium	Medium	High	Virtual	Medium
Strategy-3	Part is real	High	High	Medium	Hybrid	High
Strategy-4	Part is real	Medium	Medium	Medium	Hybrid	Medium
Strategy-5	Part is real	Low	Low	Medium	Hybrid	Low
Strategy-6	All is real	Low	Low	Low	Real	High
Strategy-7	All is real	Low	Low	Low	Real	Low

challenges, the authors used game engine Unity generate images, which were then used to train a construction worker safety monitoring model. Experimental results show that the method proposed by the author can well solve the above challenges and achieve higher detection accuracy. In response to the lack of public data sets for safety equipment detection, Benedetto et al. [43] generated photo-realistic synthetic safety equipment image dataset. A DL model trained using this dataset was able to identify whether workers were using personal safety equipment correctly during hazardous work activities. Schuster et al. [44] identified that the typical operating environment of an excavator is complex, unstructured, and continuously evolving. In response to these challenges when developing autonomous excavator, the authors proposed a training and testing loop that provides a way to generate synthetic camera and Lidar data using 3D simulated environments. The generated data can be automatically modified to support continuous development and testing of autonomous systems. Rather than identifying challenges unique to the built environment, Assadzadeh et al. [45] proposed a synthetic data generation method based on domain randomization in response to the difficulty of real-world data collection. [45]generated data were used for training excavator pose estimation model. To address the challenge of developing an extensive training image dataset, the study conducted by Kim et al. [46] encompassed the construction of a synthetic training image dataset. The generated data was subsequently utilized to facilitate accurate estimation of workers' 3D pose.

Furthermore, SSD has received interest in both construction progress monitoring and quality inspection. Ramimian et al. [47] noted that in large-scale construction projects, monitoring the implementation of each component of buildings and updating this information in BIM models can be highly labor-intensive and susceptible to errors. In order to address the above challenge, the authors successfully proposed a framework and a proof of concept prototype for on-demand automated simulation of construction projects. In the prototype, synthetic images generated by BIM are used to train the building elements identification model. Tang et al. [48] identified challenges when generating synthetic point clouds from BIM models, which included requiring extensive manual intervention and losing the unique characteristics of laser scans in real-world environments. To overcome the above challenges, the authors proposed a fully automatic method to generate synthetic point clouds from as-built BIM models. After that, these synthetic point clouds were used to train indoor scene understanding model, and experimental results showed that models trained using synthesized point clouds can achieve significant 5% −10% improvement. In view of the lack of images of dam cracks, Xu et al. [49] introduced a method in which real concrete cracks were superimposed onto dam images, generating a synthetic crack detection dataset. This dataset was subsequently employed to train models for crack detection, segmentation, and quantification, and the results showed significant improvements in model accuracy.

3.1.2. SSD applications in other fields

SSD has gained considerable attention in the field of image/video recognition, encompassing classification, object detection, and segmentation. Hwang et al. [50] developed action simulation platform that can generate synthetic elders' daily activities including RGB videos, skeleton trajectories for human action recognition. De Melo et al. [51] employed synthetic human gesture videos to train a gesture recognition model that was subsequently utilized for robot control. In their work, Li et al. [52] generated realistic traffic flows for vehicles and pedestrians in images, subsequently leveraging this data to train the Mask R-CNN [53] model for vehicle and pedestrian detection. To address the scarcity of high-quality data for planetary exploration missions, Müller et al. [54] developed a simulator specifically tailored for unstructured outdoor environments. The effectiveness of this simulator was demonstrated by evaluating the generated images using a segmentation algorithm. Inan et al. [55] used the CARLA [56] to generate a synthetic point cloud dataset for training road-objects segmentation model.

There has been a recent surge of interest in generating SSD for robotics and autonomous driving, such as uncrewed aerial vehicle (UAV) navigation and robot manipulation. Stein and Roy [57] employed synthetic image series to train a quadcopter to navigate through complex, cluttered environments. In their study, Leão et al. [58] utilized simulation to construct an environment specifically designed for bin picking, thereby enabling the generation of synthetic point clouds. These point clouds were then utilized to train a tube perception algorithm tailored for bin-picking tasks.

3.2. Review and evaluation of SSD generation strategies

3.2.1. SSD generation strategies

Advancements in computer graphics tools, including Unity, Unreal, and Gazebo, along with the growing accessibility of 3D assets, have facilitated the synthesis of simulated data. Within the computer graphics pipeline, the system receives input consisting of the 3D scene information (e.g., spatial points defining a vehicle), details regarding the materials and lighting characteristics (e.g., vehicle color and light sources), and rendering parameters (e.g., rasterization or raytracing algorithms), and subsequently generates a 3D visualization of the scene [30]. As the pipeline has information about the scene details, it can automatically generate accurate ground-truth information, including bounding boxes for objects of interest, depth information, and scene segmentation masks [59]. The utilization of advanced physics engines (e.g., Pybullet) enables the enhancement of motion realism in the generated output [60,61].

Cut-Paste is a simple and widely used method in which images of objects are superimposed onto a background to create imagery [62]. This is often accomplished by superimposing virtual objects [21,63] on real backgrounds, ensuring seamless integration through techniques such as surface alignment and lighting adjustments. Extending this approach to fusing real entities on real backgrounds [49,61] introduces the additional challenge of extracting the real entities from their original backgrounds [30].

Finally, there is great potential for integrating computer graphics tools and data-driven methods to generate SSD. All assets are manually created in the computer graphics pipeline[44]. Data-driven methods enable the extraction of patterns and features from data and the integration of the acquired information (e.g., speed and acceleration) into computer graphics pipelines [30].

The assets used for generating SSD can come from virtual assets (e.g., human-create 3D models), real assets (e.g., real-world images), or a hybrid of them [64–66]. Considering the sources of assets and the techniques employed for their combination, we identify seven SSD generation strategies in existing studies as shown in Fig. 2. Among these strategies, Strategy-2 stands out as the prevailing one, being adopted by a significant majority of 79% of the reviewed articles. The identified SSD generation strategies include:

Strategy-1 "Virtual assets by integrated data-driven approaches and computer graphics pipelines" [57,67]. Strategy-1 uses both data-driven approach and computer graphics pipeline to combine virtual assets to generate SSD.

Strategy-2 "Virtual assets by computer graphics pipelines" [64,68]. Strategy-2 exclusively utilizes computer graphics pipelines to combine virtual assets for SSD generation.

Strategy-3 "Virtual & real assets by integrated data-driven approaches and computer graphics pipelines" [65,69]. Strategy-3 employs both data-driven methods and computer graphics pipelines to combine virtual and real assets to generate SSD.

Strategy-4 "Virtual & real assets by computer graphics pipelines" [61,70]. Strategy-4 exclusively uses computer graphics pipelines to combine virtual and real assets for SSD generation.

Strategy-5 "Virtual & real assets by cut-paste" [21,63]. Strategy-5 uses the cut-paste method to combine virtual assets and real assets to generate SSD.

Strategy-6 "Real assets by integrated data-driven approaches and computer graphics pipelines" [65,69]. Strategy-6 uses both data-driven approach and computer graphics pipeline to combine real assets to generate SSD.

Strategy-7 "Real assets by cut-paste" [49,61]. Strategy-7 uses the cut-paste method to combine real assets together to generate SSD.

3.2.2. Evaluation of SSD generation strategies

Several studies have evaluated SSD generation strategies from different aspects. Chen et al. [66] found that images generated by other techniques cannot simulate realistic motion compared to computer graphics pipelines. Two studies [52,71] found that images generated using virtual assets lack texture realism. Beery et al. [61] evaluated the realism of relationships between objects in images generated by Cut-Paste and computer graphic pipelines, and found that relationships generated by computer graphics pipelines were more realistic. They also found that data generated using virtual assets has greater variability than real assets. Rudorfer et al. [72] found that Cut-Paste method is simpler than computer graphic pipelines.

As shown in Table 1, we summarized the above studies and identified six evaluation metrics to assess the performance of SSD generation strategies. The performance of each strategy is summarized as below.

- Texture realism: Virtual assets often lack the richness and authenticity of real-world data [52,66]. Therefore, all textures of data generated using virtual assets are not real, all textures of data generated using real assets are real, and some textures of data generated using hybrid assets are real.
- Relationship realism: Computer graphics pipelines can provide realistic relationship realism and the Cut-Paste method cannot guarantee the same level of fidelity [61]. Combining data-driven approaches with computer graphics pipelines can yield high relationship realism by integrating relationship-related patterns extracted from real-world data, such as the distribution of cars on roads [52,66]. Hence, when it comes to the relationship realism of the generated data, the Cut-Paste method yields the lowest level of relationship realism, while both data-driven approaches and computer graphics pipelines produce the highest level. The relationship realism of data generated through computer graphics pipelines falls somewhere in between.
- Motion realism: Similar to relationship realism, computer graphics pipelines can provide realistic motion realism, while the Cut-Paste method may fall short in ensuring this level of fidelity. Combining data-driven approaches with computer graphics pipelines can achieve high motion realism by incorporating motion-related patterns derived from real-world data [66]. Therefore, the data produced through Cut-Paste exhibits the lowest level of motion realism, while the data generated through both data-driven approaches and computer graphics pipelines achieves the highest level of motion realism. Data generated solely through computer graphics pipelines falls somewhere in between.
- Data variability: The data generated using virtual assets offers higher data variability due to the ability to control visual features such as color, shape, texture, size, and orientation [61]. So, in theory, the more virtual assets one uses, the greater the diversity of data that can be generated. If we were to rank the diversity of generated data from highest to lowest, it would be as follows: the strategy utilizing virtual assets, the strategy employing hybrid assets, and finally, the strategy relying solely on real assets.
- Assets required: Utilizing virtual assets for SSD generation necessitates the collection of virtual assets. Similarly, it becomes inevitable to collect real assets when employing them for SSD generation. Therefore, when recommending a generation strategy, it is crucial to contemplate the accessibility of required assets for users. This is why we considered the required assets when categorizing the SSD generation strategies.
- Level of complexity: The complexity of the Cut-Paste method is lower relative to computer graphics pipelines, whereas the combination of data-driven approaches and computer graphics pipelines proves to be more intricate [72].

Table 2 summarizes our assessment of the seven SSD generation strategies in terms of the six evaluation criteria. Notably, the relationships (high, medium, low) in the table are qualitative, describing the

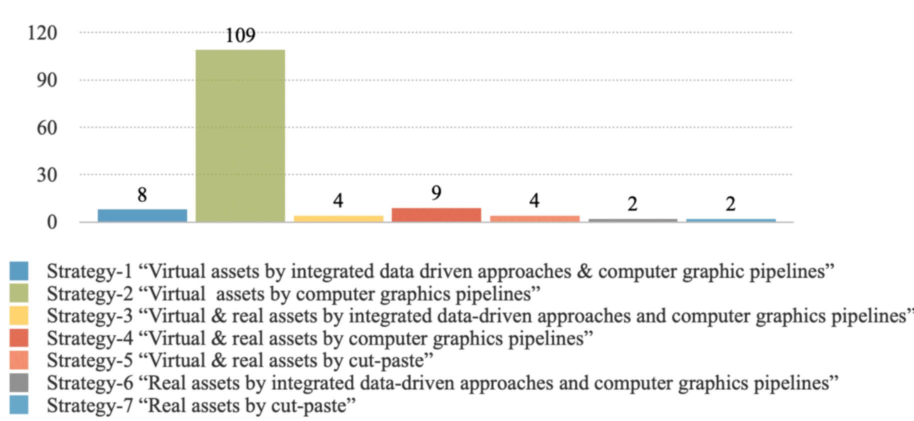
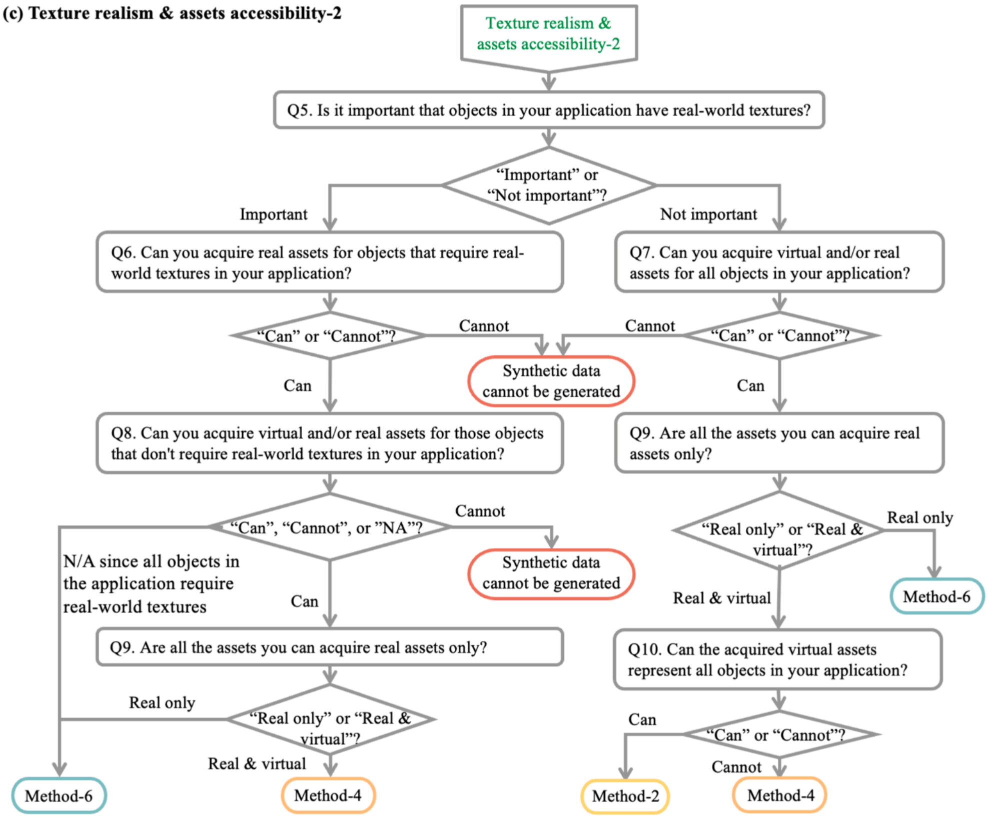


Fig. 2. Number of articles reviewed- categorized by generation strategies.

Fig. 3. The decision tree for recommending synthetic data generation strategies. Fig. 3. (Continued from the previous page) The decision tree for recommending synthetic data generation strategies. Fig. 3. (Continued from the previous page) The decision tree for recommending synthetic data generation strategies. Fig. 3. (Continued from the previous page) The decision tree for recommending synthetic data generation strategies.

(c) Texture realism & assets accessibility-2



(d) Texture realism & assets accessibility-3

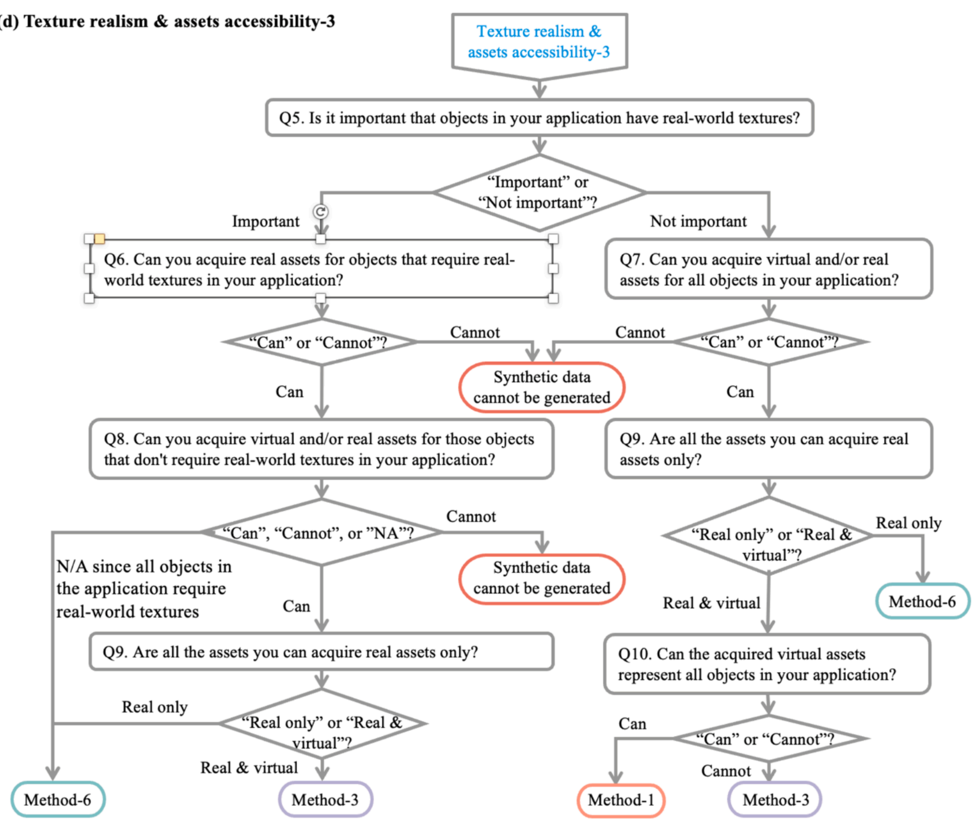


Fig. 3. (*continued*).

relative ranking of one strategy compared to others. It can be found that none of the strategies outperformed other strategies under all metrics.

4. SSD generation guideline and validation

From evaluation results in Table 2, we can find that each strategy shows advantages under different evaluation metrics. Therefore, it is necessary to establish a guideline to help users choose the strategies suitable for their applications.

4.1. SSD generation guideline

The SSD generation guideline is modeled as a decision tree comprising steps and decision points. Since each strategy exhibits comparable performance in terms of motion realism and relationship realism, users are queried about their requirements for these two metrics concurrently as shown in Fig. 3 (a). This offers the advantage of reducing the number of branches within the decision tree. Subsequently, users are queried regarding their specifications concerning texture realism, followed by inquiries regarding the availability of the required assets as shown in Fig. 3 (b), (c), (d).

It is worth noting that we introduced post-processing in the last step of the decision tree to identify the optimal SSD generation strategy. After completing the aforementioned steps, multiple alternative strategies that satisfy the requirements may emerge, and we need to select the most suitable option among them. Firstly, these alternative strategies will be sorted based on their data variability, with the highest variability being prioritized. If multiple strategies are equally viable, they will be further sorted based on their complexity level, with less complex strategies receiving higher priority. Ultimately, the strategy that ranks first will be recommended.

Specifically, the guideline involves ten questions, centered around the specified application's demands for data quality and the availability of necessary assets. As shown in Fig. 3 (a), the first two questions (Q1 and Q2) inquire whether the motion of objects and the relationships between objects hold significance for the target application. If neither holds significance, the progression proceeds directly to the first branch. Conversely, if motion is deemed important, Q3 prompts the user to determine the existence of corresponding motion models. Alternatively, if the relationship between objects is identified as crucial, Q4 is reached, wherein the user is asked to ascertain the presence of a corresponding relationship model. The user's responses to Q3 and Q4 determine the subsequent course of action. If the user lacks corresponding motion or relationship models, the progression proceeds to the third branch; otherwise, it will enter the second branch.

Subsequently, within branches 1, 2, and 3, the questions posed within each branch are the same. For instance, as shown in Fig. 3 (b), the user is queried regarding the significance of real-world textures for the target application in Q5. An affirmative response leads to Q6, where the availability of real assets for objects requiring real-world textures is examined. Conversely, a negative response in Q5 leads to Q7, where the user is asked about the availability of corresponding assets for all objects in the application. If the answer to either Q6 or Q7 is "Cannot", SSD cannot be generated due to the unavailability of required assets. However, if the answer to Q6 is affirmative, the user is queried in Q8 about the ability to obtain assets for objects not requiring real-world textures. If this question is not applicable, that is, all objects require real textures, then the final recommended strategy is Strategy-7. Conversely, if the response is negative, SSD generation remains infeasible due to the unavailability of requisite assets. A positive response in Q8 leads to Q9, where the user is asked if the assets they can obtain are solely real assets. If the answer is "Real only", Strategy-7 is the ultimate recommendation. Otherwise, Strategy-5 is recommended. In the scenario where the response to Q7 is "Can", progression to Q9 is initiated. A response of "Real only" in Q9 results in the final recommended strategy being Strategy-7. Conversely, if the answer is "Real & virtual", it directs the user to Q10, where the user is asked if the acquired assets can represent all objects in the application. If the answer is "Can", Strategy-2 is recommended; otherwise, Strategy-5 is suggested.

4.2. A guideline use case

To explain how the decision tree works for SSD generation strategies recommendation, a use case of the guideline in concrete surface defect detection is presented. Initially, we set up the background and requirements of the concrete surface defect detection. It requires the use of digital images to detect surface crack defects on concrete dam. After clarifying the application background and requirements, the using of the decision tree in the guideline commences. This begins with responses to Q1 and Q2. Given that defect detection does not require consideration of object motion, object motion is not important in defect detection. For Q2, the defect detection does not involve exploring the relationship between defects, so the answer to this question is not important. The decision tree then evaluates the responses to Q1 and Q2. With both answers being in the negative, the process advances to "Texture realism & assets accessibility-1".

After entering the "Texture realism & assets accessibility-1", the first decision point is Q5. Defect detection relies on discerning textural discrepancies between defective zones and the intact substrate. Given the inherent challenge in simulating authentic textures of concrete defects and their corresponding intact backgrounds, it becomes paramount to employ real-world textures for both defect and background. Advancing to Q6, since I can obtain a public data set of concrete surface defects and images of a concrete dam on the Internet, that is, obtain the real-world textures of the concrete defects and background, the answer to Q6 is "Can". So, the decision tree directs us to Q8. Since I can create a 3D model of dam, the response to Q8 is "Can". This leads to Q9. Obviously based on my previous answers to Q6 and Q8, the answer to Q9 is "Real & virtual". Ultimately, the decision tree culminates in the endorsement of Strategy-5 as the optimal strategy for this context, as per the delineated decision-making process.

4.3. Guideline testing and discussion

In order to ascertain the efficacy of the proposed guideline in the realm of construction, we conducted a comparative analysis, specifically by comparing the methodologies employed in the literature and those recommended by the guideline for targeted applications. Among the 208 articles reviewed, 36 pertain to SSD generation in construction. Following the 7:3 rule of thumb for DL data splitting, we randomly selected 11 articles to serve as the test set. It is worth noting that, to ensure the reliability and fairness of the test results, none of the articles in the test set were used during the development of the guideline. We then tested the proposed guideline on these 11 articles. The test process encompasses a wide range of DL applications, including tracking of construction workers (1 case), equipment recognition (1 case), pose estimation (3 cases), construction process monitoring (2 cases), infrastructure scene understanding (1 case), PPE detection (1 case), and defect detection (2 cases). The analysis result revealed that nine of the eleven articles adopted the same strategies as those suggested by our guideline (with a matching rate of 82%), while the remaining two [49,73] used different strategies, as listed in Table 3.

In order to demonstrate the robustness of the decision tree in the guideline, we calculated the sensitivity and specificity of the recommended strategies in the cases. The results are summarized in Table 4. In the table, a positive instance refers to the case where the strategy is recommended by the guideline. A negative instance refers to refers to the case where the strategy is not recommended by the guideline. True positive (TP) refers to the positive instances which are actually used by the cases. True negative (TN) refers to the negative instances which are not actually used by the cases. False positive (FP) refers to the positive instances which are not actually used by the cases. False negative (FN)

Table 3
List of the publications generating synthetic data in the realm of construction.

Publication	Construction application	Description	Data types	Generation strategy	
				Adopted	Recommended
[74]	Construction workers tracking	Tracking workers from a bird's view on a construction site	Video	Strategy-2	Strategy-2
[45]	Equipment pose estimation	Keypoints-based excavator pose estimation in construction site	Image	Strategy-2	Strategy-2
[11]	Equipment recognition	Recognizing excavators with different poses on the construction site	Image	Strategy-5	Strategy-5
[75]	Equipment pose estimation	Excavator operation activity estimation in construction site	Video	Strategy-4	Strategy-4
[46]	Workers pose estimation	Keypoints-based workers' pose estimation in construction site	Video	Strategy-4	Strategy-4
[47]	Construction progress monitoring	Monitoring the implementation of every single part of the buildings	Image	Strategy-2	Strategy-2
[76]	Infrastructure scene understanding	Recognizing every single part of buildings	Image	Strategy-2	Strategy-2
[77]	Construction progress monitoring	Recognizing every single part of buildings	Point cloud	Strategy-2	Strategy-2
[43]	PPE detection	Recognizing the correct use of PPE during at-risk work activities	Image	Strategy-2	Strategy-2
[49]	Defect detection	Dam crack detection, segmentation, and quantification	Image	Strategy-4,7	Strategy-5
[73]	Defect detection	Recognizing rusted area, crack size, and deformation size of scaffolding part	Image & point cloud	Strategy-2	Strategy-5

Table 4
Guideline validation results.

Strategy	TP	TN	FP	FN	Sensitivity	Specificity
Strategy-2	6	4	0	1	85.71%	100.00%
Strategy-4	2	8	0	1	66.67%	100.00%
Strategy-5	1	8	2	0	100.00%	80.00%

refers to the negative instances which are actually used by the cases. It can be found that the decision tree has low sensitivity when determining whether to recommend Strategy-2 and Strategy-4, and low specificity when judging whether to recommend Strategy-5. This is due to the mismatches between the recommended strategies and those actually used in the two defect detection cases. However, for these two cases, our assessment (as discussed below) indicate that the strategies recommended by our guideline are more suitable than the ones used in those cases.

In one case, Kim et al. [73] utilized Strategy-2 to generate SSD for training a DL model to identify rusted areas, crack sizes, and deformation sizes in scaffolding. However, this strategy exhibits limitations regarding the variability of scaffold crack defects in the generated data, as it exclusively permits the generation of elliptical-shaped cracks, which do not accurately represent the real-world cracks on scaffolding. Additionally, the generated data lacks interference textures in the real world, such as cement stains, as the other parts of the scaffolding remain perfect except for the defects. These two constraints can adversely impact the performance of the model trained on such data. Since real-world scaffolding cracks exhibit diverse irregular shapes, and interference textures are easily mistaken as defects.

In contrast, Strategy-5, recommended by our guideline, can effectively overcome these limitations. In Strategy-5, images of scaffolding defects and interference elements, such as cement stains from the real world, are integrated into various positions within the scaffolding 3D model. On one hand, this strategy utilizes real assets to obtain diverse real defects and interference backgrounds, thereby enhancing texture realism and variability. On the other hand, Strategy-5 uses virtual assets to acquire images of various types of scaffolding from multiple perspectives, thus enriching data variability. Hence, we believe that the strategy recommended by our guideline is better suited for this application than the one used in the literature.

In another case, Xu et al. [49] employed Strategy-4 and Strategy-7 to generate images of cracks in a concrete dam, which was used to train models for crack detection and segmentation. However, the authors did not evaluate and compare the data generated by these two strategies. Notably, Strategy-7 potentially reduces data variability since it exclusively employs real assets. In contrast, Strategy-4 introduces virtual assets beyond dams, such as the sky, thereby potentially enhancing data variability. Nevertheless, Strategy-4 is more complicated compared to Strategy-5 as illustrated in Table 2.

Strategy-5 is recommended for this application by our guideline. It is important to note that dam defect detection does not entail the detection of moving objects. Additionally, the relationship between dams and defects is relatively straightforward, allowing a simple Cut-Paste approach to sufficiently simulate this relationship. As a result, while Strategy-4 may improve motion realism and relationship realism, it is not crucial for dam defect detection. Consequently, both Strategy-4 and Strategy-5 can adequately meet the application's requirements, with Strategy-4 being the more complex of the two. Taking all factors into account, the recommended strategy for this case is Strategy-5.

5. Summary and conclusions

SSD has emerged as a potential solution to address the data collection and labeling challenges for DL-based industry applications. This paper provides an extensive literature review of the current applications of SSD in construction and other fields, as well as the strategies employed for SSD generation. Subsequently, we have summarized SSD generation strategies considering both the techniques used and the required assets and identified seven SSD generation strategies in existing studies. Furthermore, we identified six evaluation metrics of SSD generation strategies and summarized the performance of each strategy under these evaluation metrics. We found that none of the strategies outperformed other strategies under all metrics. We then formulated a guideline for selecting the suitable strategy to generate SSD for the intended application based on the performance of each strategy. Users only need to answer the questions in the guideline to obtain the recommended SSD generation strategy for their target application. To demonstrate the efficacy of the proposed guideline, we compared the strategies employed in eleven cases with those recommended by our guideline. Among these cases, nine align with the strategies recommended by our guideline. Further examination of the other two cases revealed that the strategies recommended by our guideline were, in fact, better suited than the ones used in the cases. These findings show that the developed guideline can be effective to help users select suitable SSD generation strategies for DL applications in construction.

This research contributes to the body of knowledge by summarizing existing SSD generation strategies, evaluation metrics, and their performance under these evaluation metrics. Furthermore, this paper synthesizes these insights into practical guidance for SSD users. The proposed guideline takes into account data quality, necessary assets, strategy complexity, and application requirements, making it applicable in practice. It has the potential to facilitate the utilization of SSD within the construction industry and additionally advances the progression of construction automation. However, the proposed guideline is limited to strategies for generating visual data, such as images and videos, and is not applicable to generating auditory content, text, drawings, or other forms of data. Another limitation is that the validation section of the guideline only qualitatively compares the recommended and adopted strategies when they do not match, without conducting quantitative comparisons.

CRediT authorship contribution statement

Liqun Xu: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Hexu Liu:** Writing – review & editing, Formal analysis. **Bo Xiao:** Writing – review & editing, Formal analysis. **Xiaowei Luo:** Writing – review & editing, Formal analysis. **DharmarajVeeramani:** Writing - review & editing, Formal analysis. **Zhenhua Zhu:** Writing – review & editing, Supervision, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This paper is based in part upon the work supported by the Wisconsin Alumni Research Foundation (WARF) under Project No. AAM3225. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of WARF.

References

[1] S.O. Abioye, L.O. Oyedele, L. Akanbi, A. Ajayi, J.M.D. Delgado, M. Bilal, O. O. Akinade, A. Ahmed, Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges, Journal of Building Engineering 44 (2021) 103299.
[2] J. Yang, Z. Shi, Z. Wu, Vision-based action recognition of construction workers using dense trajectories, Adv. Eng. Inf. 30 (2016) 327–336, https://doi.org/10.1016/J.AEI.2016.04.009.
[3] D. Young, K. Panthi, O. Noor, Challenges involved in adopting BIM on the construction jobsite, EPiC Series in Built, Environ. 2 (2021) 302–310.
[4] F. Barbosa, J. Woetzel, J. Mischke, Reinventing construction: A route of higher productivity, 2017.
[5] A. Darko, A.P.C. Chan, M.A. Adabre, D.J. Edwards, M.R. Hosseini, E.E. Ameyaw, Artificial intelligence in the AEC industry: Scientometric analysis and visualization of research activities, Autom Constr 112 (2020) 103081.
[6] Y. Pan, L. Zhang, Roles of artificial intelligence in construction engineering and management: A critical review and future trends, Autom Constr 122 (2021) 103517.
[7] T. Zeng, J. Wang, B. Cui, X. Wang, D. Wang, Y. Zhang, The equipment detection and localization of large-scale construction jobsite by far-field construction surveillance video based on improving YOLOv3 and grey wolf optimizer improving extreme learning machine, Constr Build Mater 291 (2021) 123268.
[8] B. Ekanayake, J.-K.-W. Wong, A.A.F. Fini, P. Smith, Computer vision-based interior construction progress monitoring: A literature review and future research directions, Autom Constr 127 (2021) 103705.
[9] N.D. Nath, A.H. Behzadan, S.G. Paal, Deep learning for site safety: Real-time detection of personal protective equipment, Autom Constr 112 (2020) 103085.
[10] S. Lee, M. Jeong, C.-S. Cho, J. Park, S. Kwon, Deep Learning-Based PC Member Crack Detection and Quality Inspection Support Technology for the Precise Construction of OSC Projects, Appl. Sci. 12 (2022) 9810.
[11] M.M. Soltani, Z. Zhu, A. Hammad, Automated annotation for visual recognition of construction resources using synthetic images, Autom Constr 62 (2016) 14–23.
[12] L. Huang, Z. Zhu, Z. Zou, To imitate or not to imitate: Boosting reinforcement learning-based construction robotic control for long-horizon tasks using virtual demonstrations, Autom Constr 146 (2023) 104691.
[13] M. Yang, C. Wu, Y. Guo, R. Jiang, F. Zhou, J. Zhang, Z. Yang, Transformer-based deep learning model and video dataset for unsafe action identification in construction projects, Autom Constr 146 (2023) 104703, https://doi.org/10.1016/J.AUTCON.2022.104703.
[14] A. Adadi, A survey on data-efficient algorithms in big data era, J Big Data 8 (2021) 1–54.
[15] Y. Li, Deep reinforcement learning: An overview, ArXiv Preprint ArXiv:1701.07274 (2017).
[16] M. Yang, C. Wu, Y. Guo, Y. He, R. Jiang, J. Jiang, Z. Yang, A teacher–student deep learning strategy for extreme low resolution unsafe action recognition in construction projects, Adv. Eng. Inf. 59 (2024) 102294, https://doi.org/10.1016/J.AEI.2023.102294.
[17] Y. Gong, M. Liu, X. Wang, IndusSynthe: Synthetic data using human-machine intelligence hybrid for enhanced industrial surface defect detection through self-updating with multi-view filtering, Adv. Eng. Inf. 59 (2024) 102253, https://doi.org/10.1016/J.AEI.2023.102253.
[18] A. Hämäläinen, K. Arndt, A. Ghadirzadeh, V. Kyrki, Affordance learning for end-to-end visuomotor robot control, in, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2019 (2019) 1781–1788.
[19] N. Cauli, D. Reforgiato Recupero, Survey on Videos Data Augmentation for Deep Learning Models, Future Internet 14 (2022) 93.
[20] T. Pivo\v{n}ka, L. Přeučil, Stereo Camera Simulation in Blender, in: International Conference on Modelling and Simulation for Autonomous Systems, 2020: pp. 206–216.
[21] J. Lu, F. Richter, M.C. Yip, Pose Estimation for Robot Manipulators via Keypoint Optimization and Sim-to-Real Transfer, IEEE Robot Autom Lett 7 (2022) 4622–4629.
[22] J.H. Ziegeldorf, O.G. Morchon, K. Wehrle, Privacy in the Internet of Things: threats and challenges, Security and Communication Networks 7 (2014) 2728–2742.
[23] M. Abufadda, K. Mansour, A Survey of Synthetic Data Generation for Machine Learning, in: 2021 22nd International Arab Conference on Information Technology (ACIT), 2021: pp. 1–7.
[24] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017 (2017) 23–30.
[25] Iris Belle, The architecture, engineering and construction industry and blockchain technology, Digital Culture (2017) 279–284. https://www.researchgate.net/profile/Iris-Belle/publication/322468019_The_architecture_engineering_and_construction_industry_and_blockchain_technology/links/5a59fe6aa6fdcc3bfb5c0369/The-architecture-engineering-and-construction-industry-and-blockchain-technology.pdf (accessed August 13, 2023).
[26] R. Rao, C. de Melo, H. Krim, Synthetic environments for artificial intelligence (AI) and machine learning (ML) in multi-domain operations, DEVCOM Army Research Laboratory, in, 2021.
[27] Simulation, Https://En.Wikipedia.Org/Wiki/Simulation#cite_note-Definition-3 (n. d.).
[28] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward, Proceedings of the National Academy of Sciences 118 (2021) e1907856118.
[29] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic Data for Text Localisation in Natural Images, in, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016 (2016) 2315–2324, https://doi.org/10.1109/CVPR.2016.254.
[30] C.M. de Melo, A. Torralba, L. Guibas, J. DiCarlo, R. Chellappa, J. Hodgins, Next-generation deep learning based on simulators and synthetic data, Trends Cogn Sci (2021).
[31] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, S. Birchfield, Training deep networks with synthetic data: Bridging the reality gap by domain randomization, in: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 969–977.
[32] J. Papon, M. Schoeler, Semantic Pose Using Deep Networks Trained on Synthetic RGB-D, in, IEEE International Conference on Computer Vision (ICCV) 2015 (2015) 774–782, https://doi.org/10.1109/ICCV.2015.95.
[33] D.P. Kaur, N.P. Singh, B. Banerjee, A review of platforms for simulating embodied agents in 3D virtual environments, Artif Intell Rev (2022) 1–43.
[34] S.I. Nikolenko, Synthetic data for deep learning, Springer, 2021.
[35] T. Kikuchi, T. Fukuda, N. Yabuki, Development of a synthetic dataset generation method for deep learning of real urban landscapes using a 3D model of a non-existing realistic city, Adv. Eng. Inf. 58 (2023) 102154, https://doi.org/10.1016/J.AEI.2023.102154.
[36] J. Fonseca, F. Bacao, Tabular and latent space synthetic data generation: a literature review, J Big Data 10 (2023) 1–37, https://doi.org/10.1186/S40537-023-00792-7/FIGURES/5.
[37] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M.M. Lalu, T. Li, E.W. Loder, E. Mayo-Wilson, S. McDonald, L.A. McGuinness, L.A. Stewart, J. Thomas, A.C. Tricco, V.A. Welch, P. Whiting, D. Moher, The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, BMJ 372 (2021), https://doi.org/10.1136/BMJ.N71.
[38] S. Jalali, C. Wohlin, Systematic literature studies: Database searches vs, in: Backward Snowballing, International Symposium on Empirical Software Engineering and Measurement, 2012, pp. 29–38, https://doi.org/10.1145/2372251.2372257.

[39] Clarivate, (n.d.). https://access.clarivate.com/login?app=wos&alternative=true&shibShireURL=https:%2F%2Fwww.webofknowledge.com%2F%3Fauth%3DShibboleth&shibReturnURL=https:%2F%2Fwww.webofknowledge.com%2F&roaming=true (accessed April 11, 2024).
[40] C. Birkle, D.A. Pendlebury, J. Schnell, J. Adams, Web of Science as a data source for research on scientific and scholarly activity, Quantitative Science Studies 1 (2020) 363–376, https://doi.org/10.1162/qss_a_00018.
[41] Google Scholar, (n.d.). https://scholar.google.com.hk/schhp?hl=en&as_sdt=0,5&inst=10434768688823972643&inst=17644838422235682599&inst=5693673 60547434339 (accessed April 11, 2024).
[42] H. Lee, J. Jeon, D. Lee, C. Park, J. Kim, D. Lee, Game engine-driven synthetic data generation for computer vision-based safety monitoring of construction workers, Autom Constr 155 (2023) 105060, https://doi.org/10.1016/J.AUTCON.2023.105060.
[43] M. Di Benedetto, F. Carrara, E. Meloni, G. Amato, F. Falchi, C. Gennaro, Learning accurate personal protective equipment detection from virtual worlds, Multimed Tools Appl 80 (2021) 23241–23253.
[44] A. Schuster, R. Hagmanns, I. Sonji, A. Löcklin, J. Petereit, C. Ebert, M. Weyrich, Synthetic data generation for the continuous development and testing of autonomous construction machinery, At-Automatisierungstechnik 71 (2023) 953–968, https://doi.org/10.1515/AUTO-2023-0026/MACHINEREADABLECITATION/RIS.
[45] A. Assadzadeh, M. Arashpour, I. Brilakis, T. Ngo, E. Konstantinou, Vision-based excavator pose estimation using synthetically generated datasets with domain randomization, Autom Constr 134 (2022) 104089.
[46] J. Kim, D. Kim, J. Shah, S.H. Lee, Synthetic Training Image Dataset for Vision-Based 3D Pose Estimation of Construction Workers, in: Construction Research Congress 2022: Computer Applications, Automation, and Data Analytics - Selected Papers from Construction Research Congress 2022, American Society of Civil Engineers, 2022: pp. 254–262. DOI: 10.1061/9780784483961.027.
[47] F.P. Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, N. Dawood, On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning, Autom Constr 110 (2020) 103012.
[48] S. Tang, H. Huang, Y. Zhang, M. Yao, X. Li, L. Xie, W. Wang, Skeleton-guided generation of synthetic noisy point clouds from as-built BIM to improve indoor scene understanding, Autom Constr 156 (2023) 105076, https://doi.org/10.1016/J.AUTCON.2023.105076.
[49] J. Xu, C. Yuan, J. Gu, J. Liu, J. An, Q. Kong, Innovative synthetic data augmentation for dam crack detection, segmentation, and quantification, Struct Health Monit (2022), 14759217221122318.
[50] H. Hwang, C. Jang, G. Park, J. Cho, I.J. Kim, ElderSim: A Synthetic Data Generation Platform for Human Action Recognition in Eldercare Applications, IEEE Access 11 (2023) 9279–9294, https://doi.org/10.1109/ACCESS.2021.3051842.
[51] C.M. de Melo, B. Rothrock, P. Gurram, O. Ulutan, B.S. Manjunath, Vision-based gesture recognition in human-robot teams using synthetic data, in, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020 (2020) 10278–10284.
[52] W. Li, C.W. Pan, R. Zhang, J.P. Ren, Y.X. Ma, J. Fang, F.L. Yan, Q.C. Geng, X. Y. Huang, H.J. Gong, et al., Augmented autonomous driving simulation using data-driven algorithms, Sci Robot 4 (2019) eaaw0863.
[53] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017: pp. 2961–2969.
[54] M.G. Müller, M. Durner, A. Gawel, W. Stürzl, R. Triebel, R. Siegwart, A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments, in, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2021 (2021) 9765–9772.
[55] B.A. Inan, D. Rondao, N. Aouf, Enhancing LiDAR Point Cloud Segmentation with Synthetic Data, 2023 31st Mediterranean Conference on Control and Automation, MED 2023 (2023) 370–375. DOI: 10.1109/MED59994.2023.10185741.
[56] CARLA Simulator, (n.d.). https://carla.org/ (accessed May 3, 2024).
[57] G.J. Stein, N. Roy, Genesis-rt: Generating synthetic images for training secondary real-world tasks, in, IEEE International Conference on Robotics and Automation (ICRA) 2018 (2018) 7151–7158.
[58] G. Leão, C.M. Costa, A. Sousa, L.P. Reis, G. Veiga, Using Simulation to Evaluate a Tube Perception Algorithm for Bin Picking, Robotics 11 (2022) 46.
[59] M. Angus, M. ElBalkini, S. Khan, A. Harakeh, O. Andrienko, C. Reading, S. Waslander, K. Czarnecki, Unlimited Road-scene Synthetic Annotation (URSA) Dataset, in: In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 985–992, https://doi.org/10.1109/ITSC.2018.8569519.
[60] Y. Long, W. Wei, T. Huang, Y. Wang, Q. Dou, Human-in-the-Loop Embodied Intelligence With Interactive Simulation Environment for Surgical Robot Learning, IEEE Robot Autom Lett 8 (2023) 4441–4448, https://doi.org/10.1109/LRA.2023.3284380.
[61] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, N. Joshi, M. Meister, P. Perona, Synthetic examples improve generalization for rare classes, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020: pp. 863–873.
[62] F. Hossain, T. Dagiuklas, On the benchmarking of ResNet forgery image model using different datasets, in, Human-Centered Cognitive Systems (HCCS) 2022 (2022) 1–7, https://doi.org/10.1109/HCCS55241.2022.10090275.
[63] G. Varol, I. Laptev, C. Schmid, A. Zisserman, Synthetic humans for action recognition from unseen viewpoints, Int J Comput vis 129 (2021) 2264–2287.
[64] S. Ahmad, K. Samarawickrama, E. Rahtu, R. Pieters, Automatic Dataset Generation From CAD for Vision-Based Grasping, in: 2021 20th International Conference on Advanced Robotics (ICAR), 2021: pp. 715–721.
[65] H. Abu Alhaija, S.K. Mustikovela, L. Mescheder, A. Geiger, C. Rother, Augmented reality meets computer vision: Efficient data generation for urban driving scenes, Int J Comput vis 126 (2018) 961–972.
[66] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, R. Urtasun, Geosim: Realistic video simulation via geometry-aware composition for self-driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: pp. 7230–7240.
[67] W. Liu, W. Wang, Y. You, T. Xue, Z. Pan, J. Qi, J. Hu, Robotic picking in dense clutter via domain invariant learning from synthetic dense cluttered rendering, Rob Auton Syst 147 (2022) 103901.
[68] A. Assadzadeh, M. Arashpour, H. Li, R. Hosseini, F. Elghaish, S. Baduge, Excavator 3D pose estimation using deep learning and hybrid datasets, Adv. Eng. Inf. 55 (2023) 101875, https://doi.org/10.1016/J.AEI.2023.101875.
[69] Y. Wang, J. Gu, L. Zhao, Y. Zhang, H. Wang, Target detection based on simulated image domain migration, IEEE Access 8 (2020) 79724–79733.
[70] G. Baulig, T. Gulde, C. Curio, Adapting egocentric visual hand pose estimation towards a robot-controlled exoskeleton, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018: p. 0.
[71] O. Gamal, K. Rameshbabu, M. Imran, H. Roth, Bridging the Reality Gap: Investigation of Deep Convolution Neural Networks Ability to Learn from a Combination of Real and Synthetic Data, Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (2020). https://www.academia.edu/99881941/Bridging_the_Reality_Gap_Investigation_of_Deep_Convolution_Neural_Networks_Ability_to_Learn_from_a_Combination_of_Real_and_Synthetic_Data (accessed February 1, 2024).
[72] M. Rudorfer, L. Neumann, J. Krüger, Towards Learning 3d Object Detection and 6d Pose Estimation from Synthetic Data, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019: pp. 1540–1543.
[73] A. Kim, K. Lee, S. Lee, J. Song, S. Kwon, S. Chung, Synthetic Data and Computer-Vision-Based Automated Quality Inspection System for Reused Scaffolding, Appl. Sci. 12 (2022) 10097.
[74] M. Neuhausen, P. Herbers, M. König, Using synthetic data to improve and evaluate the tracking performance of construction workers on site, Appl. Sci. 10 (2020) 4948.
[75] W. Torres Calderon, D. Roberts, M. Golparvar-Fard, Synthesizing pose sequences from 3D assets for vision-based activity analysis, J. Comput. Civ. Eng. 35 (2021) 4020052.
[76] Y. Hong, S. Park, H. Kim, H. Kim, Synthetic data generation using building information models, Autom Constr 130 (2021) 103871.
[77] S. Hoop, T. Tschickardt, J. Schmitt, Parametrisierte Modellierung für den Einsatz von KI am Beispiel Betonbrückenbau, Bautechnik 99 (2022) 807–816, https://doi.org/10.1002/BATE.202200007.
[78] D. Gries, F.B. Schneider, Computer Vision Algorithms and Applications, accessed June 11, 2023, Computer Vision: Algorithms and Applications (2011) 311–320, http://172.104.50.50/handle/cinec20/233.
[79] Y.-J. Zhang, Motion Analysis, in: Y.-.-J. Zhang (Ed.), Handbook of Image Engineering, Springer Singapore, Singapore, 2021, pp. 1127–1164, https://doi.org/10.1007/978-981-15-5873-3_32.
[80] R. Dong, Q. Chang, S. Ikuno, A deep learning framework for realistic robot motion generation, Neural Comput Appl (2021) 1–14, https://doi.org/10.1007/S00521-021-06192-3/FIGURES/15.
[81] K. Yang, J. Jie, S. Haihui, Study on the virtual natural landscape walkthrough by using Unity 3D, in, IEEE International Symposium on VR Innovation 2011 (2011) 235–238, https://doi.org/10.1109/ISVRI.2011.5759642.