Make it till You Fake It: Construction-Centric Computational Framework for Simultaneous Image Synthetization and Multimodal Labeling.

Ali Tohidifar[1], Daeho Kim[2], and SangHyun Lee[3]

[1]PhD Candidate, Civil and Mineral Engineering, University of Toronto,
35 St. George St., Toronto, ON M5S1A4, CA, E-mail: ali.tohidifar@mail.utoronto.ca.
[2]Assistant Professor, Civil and Mineral Engineering, University of Toronto,
35 St. George St., Toronto, ON M5S1A4, CA, E-mail: civdaeho.kim@mail.utoronto.ca (corresponding author).
[3]Professor, Civil and Environmental Engineering, University of Michigan,
2350 Hayward St., Ann Arbor, MI 48109-2125, U.S., E-mail: shdpm@umich.edu.

Abstract

We introduce BlendCon, a fully automated framework capable of simultaneously synthesizing and labeling construction imagery data. This framework simulates a construction site by orchestrating 3D mobile objects against a 3D background and produces multimodal labels for target entities. The effectiveness of the synthetic data in training object detection models was thoroughly validated. For the task of construction worker detection, a YOLOv7 model trained with synthetic data nearly matched the performance of a model trained with real data, achieving 71% AP@0.5-0.95 compared to 75% for the model trained with real data. Moreover, the model trained with synthetic data surpassed its real data counterpart in scenarios requiring stricter IoU thresholds, particularly above 85%. Acquiring a sufficient number and diverse range of imagery data has been a primary challenge in construction studies addressing automation and digitization with deep neural networks. BlendCon can significantly contribute to addressing this data scarcity challenge.

Keywords: Visual Artificial Intelligence, Deep Neural Network, DNN Training, Image Synthetization, Automated Labeling, Computer-Generated Imagery, 2D Worker Detection

1. Background and Motivation: Unlocking the Potential of Visual AI in Construction

The advent of visual artificial intelligence (visual AI), powered by deep neural networks (DNNs), has opened up new horizons of digitization and automation in various sectors. Take for example the following cross-domain applications built upon visual AIs which are already in our daily lives: (i) autonomous driving [1]; (ii) medical image diagnosis [2]; (iii) face recognition [3]; and (iv) a variety of smartphone camera functions including 3D scanning [4] and navigation assist [5]. These examples demonstrate the broad applicability of visual AI across different target domains, showcasing its transformative potential in digitization and automation. Engineering-based sectors are at the forefront of these revolutionary changes—the fourth industrial revolution (Industry 4.0)—and the construction sector is no exception.

The construction sector, while still in the early stages of transformation, is gradually embracing visual AI-driven robotic solutions. Leading equipment manufacturers, like Komatsu [6] and Built Robotics [7], are retrofitting their equipment with autonomous kits enabled by visual AI. Concurrently, robotics firms have developed their own visual AI solutions for autonomous navigation, unveiling several mobile robot platforms including Spot by Boston Dynamics [8] and Husky Observer by Clearpath Robotics [9]. In tandem with this industry momentum, academic research into applications for visual AI in construction has been expanding rapidly. A growing number of studies are exploring potential applications of visual AI, with a notable emphasis on the areas of progress monitoring [10,11], safety monitoring [12], quality control [13], and the cutting-edge concept of live digital twining [14]. As the construction sector ventures further into visual AI, robotic solutions, and Industry 4.0 practices, a transformative shift in the construction industry is anticipated.

The future of visual AI appears promising, as there have been significant advancements in both software and hardware. Since 2016 we have witnessed a notable evolution in DNN architectures and the development of more varied DNN training algorithms. Concurrently, computing hardware (e.g., graphical processing units (GPUs)) have continued to increase in power while becoming more accessible and affordable. Cloud-based solutions are now just a click away and offer increasing value for money. Last but not least, more diverse benchmark datasets with new modalities tailored for emerging vision tasks have been increasingly available online. The results of these advances have been continuous growth in the capabilities of visual AI, with this rapid pace of development anticipated to continue into the foreseeable future.

Nevertheless, the pace of research into visual AI for the construction sector lags behind that of other major sectors, with an ever-widening gap. While the computer science domain continually introduces new (or deeper) DNN architectures, novel training algorithms, innovative training platforms, and new data modalities, adoption and implementation has been slow in construction research. This study addresses the primary reason behind this slow progress: the absence of a sufficient number and diverse range of DNN training data specifically crafted for construction-centric applications.

2. The Bottleneck: How Data Scarcity Affects DNN Excellence

At the core of visual AI lies supervised DNN models, replete with millions of learnable parameters. Success in training these DNN models depends heavily on both the quantity and diversity of training data. Simply put, the lesser the training data, the lower the accuracy [15], and the lesser the diversity of training data, the lower the generalizability [15]. These are the axioms of supervised learning. Without sufficient and high-quality data, even the most intricate hyper-parameter tuning, including architecture modification, falls short, never reaching optimal performance. Ensuring a sufficient number of diversified training images and labels must therefore be the first priority in the optimization of DNN models for construction visual AI.

The scarcity of DNN training data has been a significant hurdle for many construction studies. The volume and diversity of training data typically used in previous construction studies were insufficient to fully saturate a DNN architecture, and therefore insufficient to support valid conclusions. The magnitude of the gap in the size of construction DNN training sets becomes evident when we compare the volume of training images used in computer science studies with those used in construction studies. In the former, most DNN studies leverage shared benchmark training datasets featuring a minimum of 1,000,000 images [16]. In contrast, previous construction studies typically used a small set of proprietary construction data, with most of these sets containing fewer than 15,000 training images, with some much smaller than this [17,18]. Expecting a DNN model trained on such a limited data set to achieve a level of performance comparable to a DNN model trained on a multimillion-image dataset is, from a theoretical standpoint, untenable.

Given that every construction project is both unique and ever-changing, developing effective DNN models requires training with large-scale datasets that capture a wide spectrum of backgrounds, objects, materials, equipment, and workers at various scales, viewpoints, and illumination conditions [19]. Despite its critical importance to DNN optimization, assembling a sufficiently large set of construction scene training images has so far proved logistically impossible. Manual data collection and labeling are prohibitively costly and time-consuming (e.g., segmentation of a single image on Google AI Platform costs approximately $2.72 [20]). Also, there are cases where labeling data necessitates deploying physical sensors (e.g., motion capture sensors for 3D pose labeling), which is not feasible on real construction sites. Last but not least, the sharing of construction site images is not allowed in many cases due to issues of confidentiality.

Although many construction studies have invested a significant portion of their research funds and resources into manual data collection and labeling, these individual datasets are small and biased. Low-quality individual datasets pose three challenges. They: (1) result in overfitted DNN models of low accuracy and scalability since they fail to balance between the complexity of DNN architecture and the amount and diversity of training images [21]; (2) miss the opportunity to develop or apply deeper DNN architectures since such an attempt would make the balancing even harder and cause worse overfitting [22]; and (3) do not allow for competitive benchmarking against other research outcomes, since there is no standardization in training, validation, and test datasets, making fair comparison between competing architectures, training algorithms, and other hyper-parameter settings unfeasible [21].

While data scarcity is a common issue in academia, it persists in an industrial context as well. Most AI datasets that are made publicly available are free to use in academic circles. However, their commercial use is increasingly restricted. Consider several examples of representative benchmark datasets for visual AI research: ImageNet [23], Celeb A [24], and H3.6M [25]. None of the benchmark datasets are available for commercial use. As AI training data are increasingly recognized as value-added assets, the competition to secure proprietary training datasets has already begun among big tech companies. With access to commercially available benchmark datasets increasingly narrowed, tech startups focusing on construction AI development will be running up against barriers caused by data scarcity.

Overall, the significance of data to the training of construction AI is undeniable, and the current limitations of available training data are equally undeniable. Recognizing this roadblock to progress, an increasing number of recent studies highlight the potential value of synthetic datasets as replacement for datasets of real construction environments. This was the focus of our research.

3. Research Question: Can Synthetic Data Be the Answer to the Data Scarcity Problem?

Unlike human vision, DNNs interpret an image as a set of numbers in three channels, meaning that the images for DNN training do not necessarily need to be real, as long as they can visually characterize realistic scene contexts [26]. This perspective opens a novel approach to preparing DNN training images—the use of computational image synthetization and automated labeling. The holy grail of this approach is to automatically generate and label non-real but real-looking construction scene images using a process that does not require any manual inputs or site visits. The large volume datasets created through this automated solution could be deployed to retrain current DNN models to reach higher levels of performance, while allowing for the exploration of even deeper DNN architectures. Automated labeling of synthetic images will save construction studies significant time and resources, and there will be no legal issues in sharing this non-real data, allowing such datasets to be used for benchmarking, competition and collaboration. Currently, this innovative approach to addressing the serious gap in construction training data remains underexplored.

Recent research in the computer science domain has increasingly deployed synthetic data in training and optimizing DNN performance. According to a study conducted by Gartner [27], it is expected that by 2024, 60% of all data used in AI development will be synthetic rather than real. This trend is already evidenced in tech industry stalwarts like Tesla and Facebook, who are intensively harnessing proprietary synthetic data. Tesla, a pioneer in using synthetic data, is leveraging it to surmount the intricate challenges of autonomous driving [28]. Meanwhile, Facebook’s acquisition of AI. Reverie, an early player in synthetic data, signals escalating interest from big tech in this domain [29]. These cases evidence the transformative potential of synthetic data in AI development across various sectors.

This technique capitalizes on computational power to synthetize new data from analogous sources. One of the most well-known studies in this domain, known as ‘Flying Chairs’ research [30], generated synthetic images using 3D models of chairs in a virtual world. These synthetic images proved to be effective in boosting DNN training. The process of generating synthetic images includes the automatic and simultaneous labeling of each image, allowing for the creation of virtually unlimited datasets in a seamless process [31], as exemplified by the SYNTHIA dataset [32]. Another recent benchmark study validated the effectiveness of synthetic data on DNN training [33] by using the engine of the ‘GTA’ video game to generate a large dataset for multi-person human detection and tracking. This experiment proved that when synthetic data is crafted with precision, it can effectively replace real data for vision tasks including pedestrian detection, re-identification, segmentation, and tracking [33]. This growing body of research supports the proposition that synthetic data, provided it is sufficiently realistic, can serve as a viable substitute for real-world images in training DNNs [33].

Despite demonstrated potential, synthetic image generation and labeling for DNN training has not been fully addressed in construction settings. While several construction studies have generated preliminary insights, this research area is still in its early developmental stages. Previous attempts to use building information modeling (BIM) for data generation in construction have omitted key elements driving productivity, including and especially human workers. Challenges also arose from an inability to address multimodal labeling and the poor level of reality of hastily crafted synthetic images. This study aims to bridge these gaps.

The following outlines the three specific research gaps that were identified and addressed in our research:

- Worker-centric dynamic simulation: Field workers, the key players of a construction project, are the key elements of productivity and thereby the major target to monitor. The most critical research gap that the authors identified at the outset of this study was the lack of a computational tool that can simulate virtual construction workers with realistic motions. Recent studies, such as those by Acharya [34], Ma et al. [35], Hong et al. [36], and Ying et al. [37], leveraged virtual data that building information models generate. However, those studies were required to scope down to static structural components as the platforms used did not allow for modelling of the dynamic mobility inherent in construction resources. Recent endeavors by Soltani et al. [38], Kim and Kim [39], and Mahmood et al. [40], have used 3D modeling tools to synthetize virtual construction equipment data from varied viewpoints. While these efforts addressed construction equipment, the dynamic presence of field workers with realistic construction motions remains largely unexplored. Only a few studies, for example, Neuhausen et al. [31], have explored the inclusion of human workers, but in limited settings involving limited diversity in background textures, workers' motions, and avatar designs, combined with confined imaging conditions. The previous literature highlights this evident research gap and calls attention to the need for a reproducible method to simulate and label realistic construction workers in motions [31].
- Multimodal labeling: The integration of diverse sensory data, such as coupling depth maps with RGB images [41] or merging semantic segmentation masks with other imagery data [42], has been recognized as a promising approach for enriching DNN training processes [43]. Despite this, construction studies have encountered challenges in capitalizing on these

advancements, primarily due to a dearth of suitable multimodal datasets. For multimodal data to effectively augment DNN training, all modalities must be collected at the same viewpoint, not to mention that an extensive amount of data is required [44]. Manually compiling such datasets presents significant hurdles, particularly due to the dynamic nature of construction environments resulting in viewpoint or timeframe inconsistencies in collected data. This challenge highlights the need for innovative solutions to multimodal labeling under dynamic imaging conditions.

- The reality gap: The level of reality (or fidelity) of synthetically generated images is paramount in DNN training. The "reality gap" is a term that refers to the divergence between the distributions of real and synthetic images as a result of their disparate origins [45]. Earlier construction research that attempted to use synthetic images in DNN training overlayed 3D synthetic avatars onto a 2D real background image [46,47], which often compromised the fidelity of the resulting outcome. This approach faced challenges in convincingly placing avatars within 2D backgrounds, a task that can be more suitably addressed in a 3D environment (refer to **Error! Reference source not found.**). Additionally, the resulting discrepancy in lighting conditions between the avatar and the background was evident. Avatars rendered under unique lighting conditions contrasted with their backgrounds, inadvertently providing DNNs with misleading cues during training—a visual cue absent during real-world inference. When present, such discrepancies ultimately diminish the effectiveness of DNN models trained with synthetic images in real-world applications, underscoring the need for high fidelity approaches to synthetic image generation.

4. Research Objectives

In this study, we identified two research objectives: (i) developing an end-to-end fully automated computational framework that can simultaneously synthetize and label multimodal construction images and (ii) conducting DNN training experiments to validate the effectiveness of resulting synthetic data on DNN training and final performance. We first developed the computational framework from scratch, incorporating extensive source coding, and named it BlendConstruction—BlendCon, for brevity. BlendCon offers novel functionality, including the following:

- Worker-centric simulation under dynamic imaging conditions: BlendCon stands out by seamlessly integrating dynamic elements into the synthetizing process, including and especially simulations of construction workers with realistic motions under varying camera viewpoints. This feature is expected to maximize the level of diversity of resulting datasets. While the current version of BlendCon is capable of simulating mobile construction equipment along with workers in motions, the focus of this study was tuned to worker-centric simulation and validation, as this was more challenging and addressed a clear research gap.
- Multimodal labeling: In response to the challenge of the paucity of multimodal datasets for construction objects and scenes, BlendCon has the capacity to generate multiple types of labels, encompassing: (1) 2D and 3D bounding boxes of workers; (2) 2D and 3D poses (i.e., 2D/3D coordinates of each keypoint) of workers; (3) semantic segmentation masks of workers; and (4) a depth map. At each iteration of simulation, BlendCon automatically and mathematically generates precise labels simultaneously with image generation and saves those along with synthetized RGB images.

- Mitigated reality gap: To mitigate the reality gap identified in prior studies, BlendCon elevates the quality of resulting images by incorporating 3D backgrounds and orchestrating 3D simulation. BlendCon's simulation algorithm ensures realistic avatar placements (e.g., on the floor, not floating in air; see Figure 1) and harmonizes lighting conditions, mitigating the discrepancies between avatar and background light reflections, thereby providing high-fidelity outcomes.

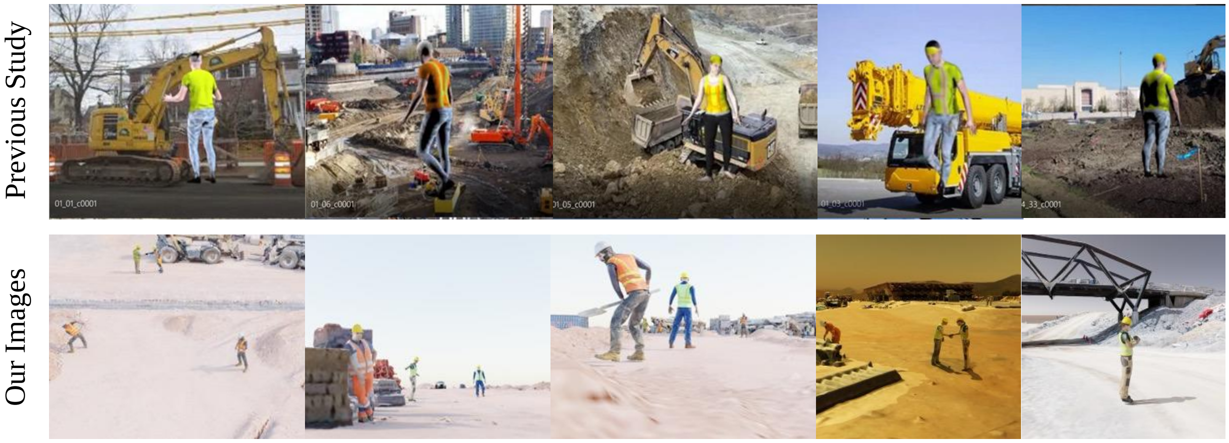


Figure 1. Samples of synthetic construction images: prior study [46] vs. ours

Following the software development, we conducted a series of DNN training experiments to determine the effectiveness of the resulting synthetic data, looking at the fundamental questions of trainability and scalability.

- Trainability (qualitative verification): How do DNNs react to the synthetic data during training? The aim was to confirm whether the synthetic data generated by BlendCon could train a model from scratch at an equivalent level to real data. We conducted visual verification on training patterns to affirm the trainability of the synthetic data that we generated on our own via BlendCon. This phase can be considered a preliminary step of the main experiment—the scalability validation.

- Scalability (quantitative validation): How scalable will a DNN trained on synthetic data be to real-world scenarios? The aim was to test the performance of a DNN model trained with synthetic data, comparing the performance against DNNs trained with real-world data. Multiple DNN models were trained with real data for use as comparison using varying quantities of training samples, to allow for more in-depth and varied comparative analyses. We employed quantitative evaluation metrics for direct comparison.

The body of this paper provides details on the technical aspects of BlendCon (Section 5 below), and presents the validation procedure and the results achieved as measured by the trainability and the scalability of the synthetic data-trained model (Section 6). Finally, our key insights (Section 7) and conclusion (Section 8) are presented.

5. BlendCon: The Construction-Centric Worker-Focused Computational Framework for Simultaneous Image Synthetization and Multimodal Labeling

BlendCon is built upon the Blender graphic simulation engine [48] (Figure 2), with its operation structured in two primary stages: (i) image synthetization; and (ii) multimodal labeling.

- Stage #1 - image synthetization: BlendCon initiates a synthetization process, loading a meticulously designed realistic horizon (e.g., a sky model; Figure 2). Subsequently, a pre-processed construction scene (e.g., a 3D point cloud), which is randomly chosen from the back-end data archive, is encapsulated inside of the horizon. Following this, a designated number of animated avatars with motions are positioned within predefined locations within the scene. The camera parameters, including camera distance, focal length, and resolution, along with lighting variables, such as time of day, atmospheric conditions, and the sun's angle, are then meticulously configured. Following these configurations, BlendCon sets and simulates a virtual construction scene, which is in turn used to render synthetic image sequences, generating the first major research outcome.

- Stage #2 - multimodal labeling: Having a computational and mathematical simulation as the base, BlendCon retrieves and tracks precise details of all elements and hyper-parameter values activated during the simulation. This retrieved data serves as the foundation for subsequent mathematical processing that generates multimodal labels. This stage manages the conversion of raw scene element data (e.g., camera's extrinsic and intrinsic parameters and kinematic information of animated avatars) into a series of ground truths.

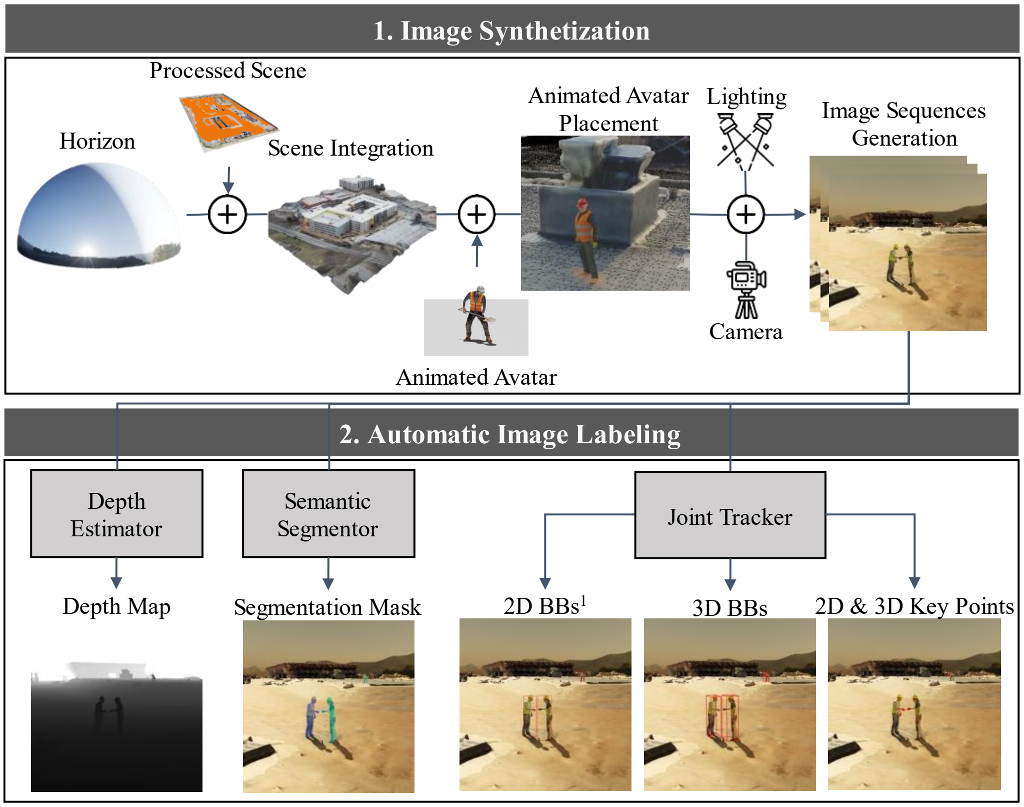


Figure 2. Overview of BlendCon

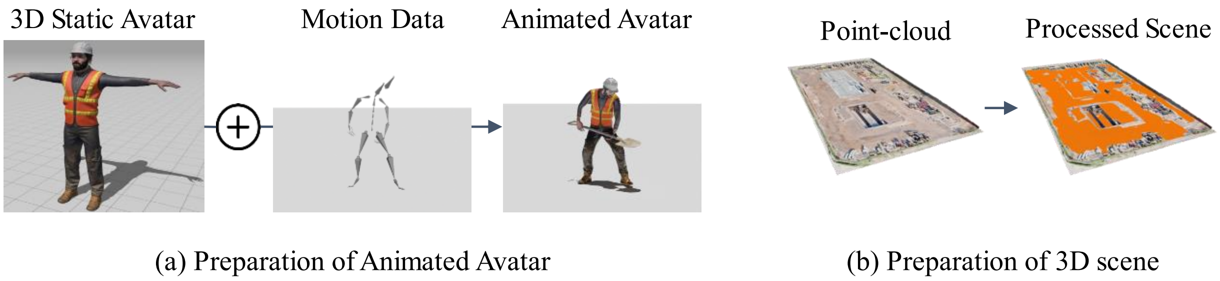
Note: BB(s) stands for bounding box(es).

The following describes the specific stages BlendCon uses to synthetize and label construction scene images.

5.1. Stage #0: Preparation of Back-End Data Archive

Prior to the two main stages, there is Stage #0, which involves archiving back-end data (e.g., pre-processed 3D point clouds and animated avatars). BlendCon completes simulations by blending these data; therefore, the diversity of simulations BlendCon can produce is directly influenced by the diversity of the back-end data. This subsection offers potential users a set of guidelines for preparing the necessary back-end data, and details the extent and nature of data the authors prepared for this study.

The primary inputs to BlendCon are animated 3D construction worker avatars (foreground) and 3D construction scenes (background). These inputs require preprocessing to facilitate the full automation of BlendCon. Figure 3 illustrates an overview of the data preparation process, which is described in detail in the rest of this subsection.



(a) Preparation of Animated Avatar (b) Preparation of 3D scene

Figure 3. Overview of BlendCon's input data pre-processing

5.1.1. Preparation of 3D Construction Avatars Animated with Motions

We followed a three-step procedure as illustrated in Figure 3(a).

- Step #1, collecting 3D static avatars: We collected pre-designed static avatars from online sources. These avatars are meticulously designed by expert artists, yet static without any motions embodied. To boost the diversity, we expanded our selection to varied body shapes, heights, weights, and the range of skin tones, as illustrated in Figure 4. In this study, we collected 37 avatars: 18 avatars resemble construction workers and the remaining 19 avatars wear non-construction clothing. We included this mix not only to mirror the typical attire found on construction sites but also to add richness to the diversity of our archived data.

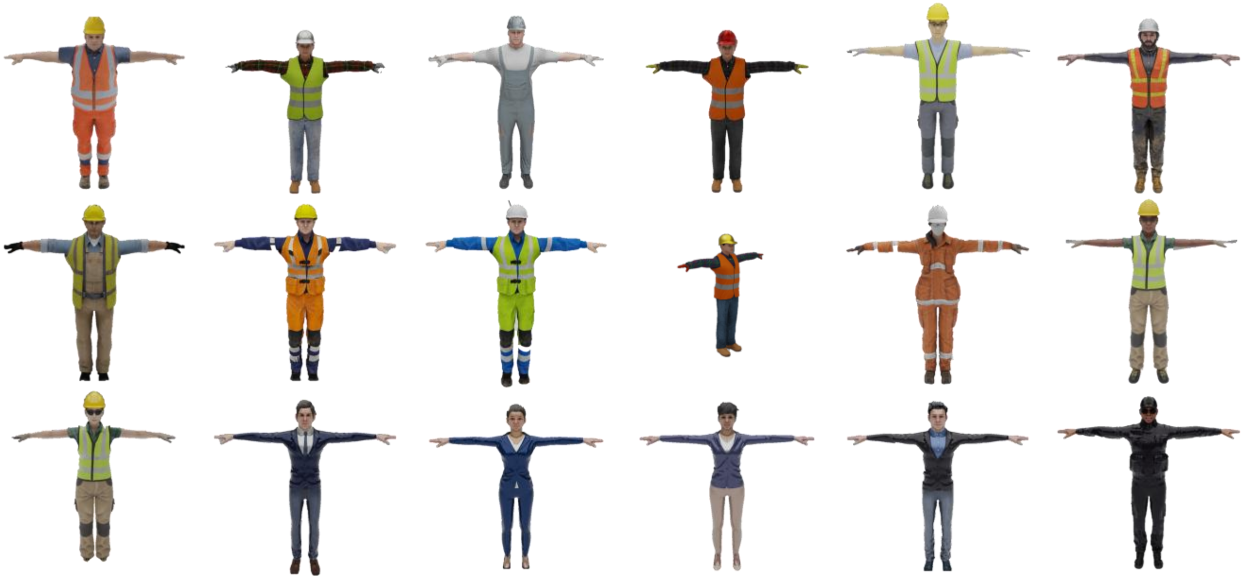


Figure 4. Sample of avatars archived.

- Step #2, collecting construction motion data: Motion is fundamental to spatio-temporal simulations of mobile objects such as construction workers. To collect our own motion capture (MoCap) data, we utilized a wearable motion capture (MoCap) system (Smartsuit Pro, Rokoko [49]), specifically emphasizing actions commonly observed at construction sites (Table 1). We collected 15 types of construction MoCap data, each resembling typical construction activities, including bricklaying, material handling and carrying, shoveling, hammering, drilling, cutting, and more. Each MoCap data is collected at 100 frames per second (FPS).

- Step #3, avatar-motion mapping: This process involves retargeting MoCap data to static avatars using a specialized algorithm. Built on Rokoko's retargeting function [50], this method matches each bone location of the avatars with the corresponding location in the motion data. A series of normalizations are applied to both the motion data and avatars, ensuring a compatible naming convention between them. Once normalized, each joint movement in the avatar is determined by its corresponding joint in the motion data for each frame. As a result, we successfully archived a total of 555 animated construction worker avatars (37 static avatars ×15 MoCap data = 555 animated avatars).

Table 1. List of all animation collected and used for this study.

Motion Type	Duration (unit = seconds)
Brick Laying	19.5
Carrying Wheelbarrow	13.7
Checking Design Sheet	16.1
Curing Concrete	17.5
Cutting Plate	15.9
Digging	20
Drilling	16.5
Hammering – Standing	17.7
Hammering – Bending Knees	18.1
Hammering – Knees on Ground	17.2
Loading Wheelbarrow	9.4
Looking Around	5.9
Shaking Hand	1.9
Walking	15.5
Leaning on Wall	27.2

5.1.2. Preparation of 3D Construction Scenes

In BlendCon, a 3D scene serves as a digital background where animated avatars are deployed. This background can simply be a 3D point cloud; however, to facilitate the initial stages of the simulation process it needs to be pre-processed. First, these digital arenas need to be scaled to real-world dimensions. Second, a ‘work-zone’ needs to be defined within each 3D point cloud to determine plausible areas where animated avatars can stand up and navigate through (e.g., the orange shaded areas in Figure 5). We collected 47 instances of 3D point clouds each of which captures a unique construction site. When pre-processing of all these instances was completed we had realized 32,473,716 m^2 of work-zone. Figure 5 visualizes some samples of the scene dataset and their corresponding work zones, highlighted in orange.

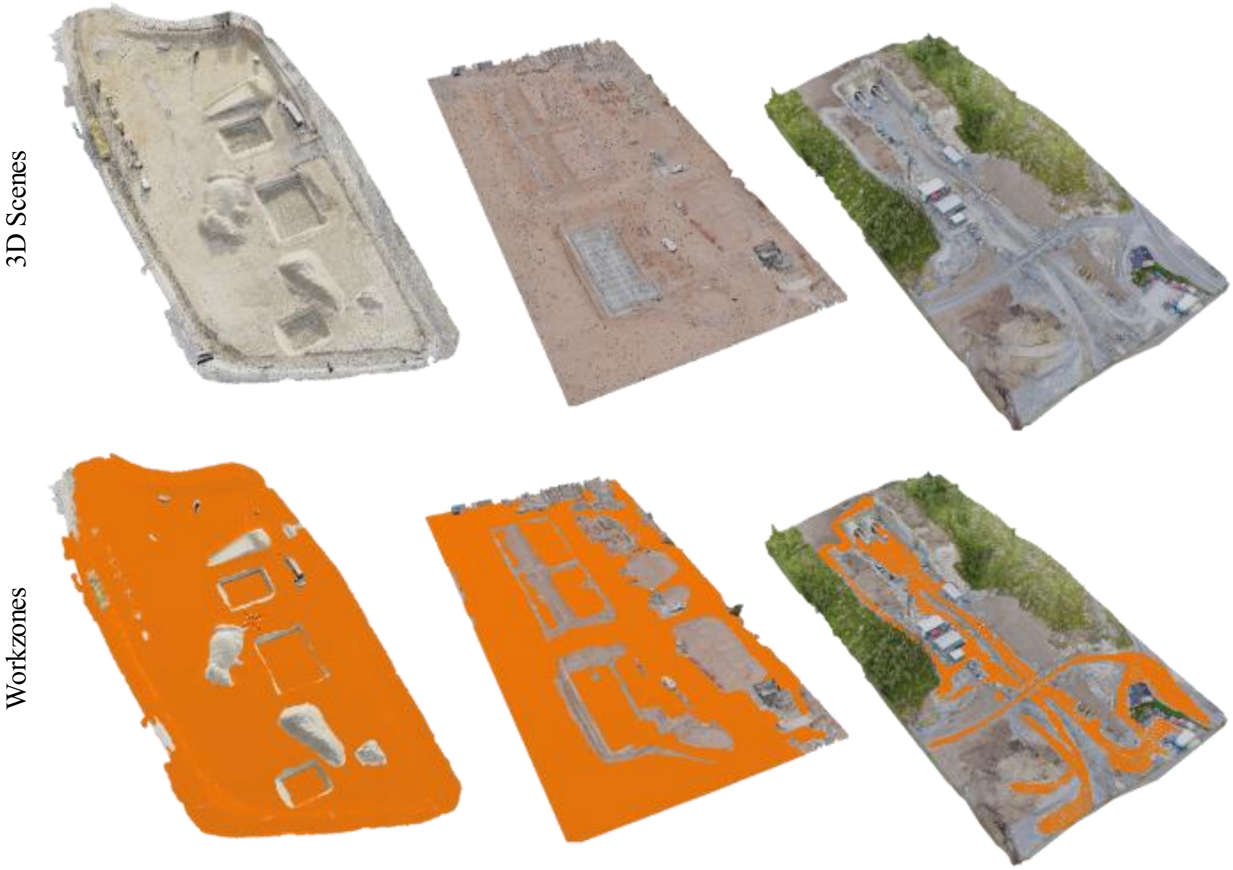


Figure 5. Samples of 3D scenes with their corresponding "work zones" designed for BlendCon

5.2. Stage #1: Image Synthetization

Image synthetization, the first main stage of BlendCon, consists of four major steps: (i) setting up an arena; (ii) positioning a camera; (iii) setting up lighting conditions; and (iv) rendering.

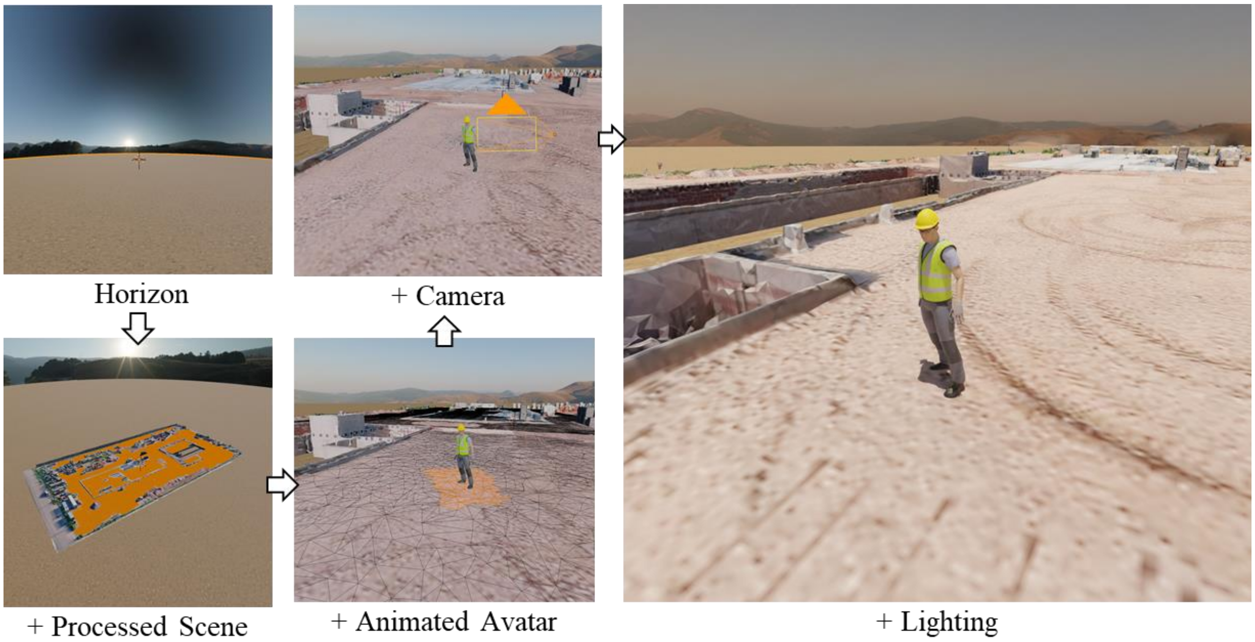


Figure 6. Procedural stage set up.

5.2.1. Step #1: Setting up an Arena

The data synthetization process begins with setting up a realistic arena. First, a real-looking horizon is added to an empty stage (Figure 6), noting that this horizon functions as a boundary. A pre-designed horizon is used, consisting of a vast half-sphere with a high-dynamic-range image mapped onto it. The half-sphere provides a visual representation of the atmosphere and horizon simultaneously. Next, a processed scene is picked randomly from our back-end data archive (Figure 6). This scene already matches to the real-world scene scale and comes with a pre-defined work-zone, setting the simulation arena for the animated construction avatars. Subsequently, a user-specified number of avatars are randomly selected from the back-end data archive and placed within the predefined work zone (Figure 6). BlendCon allows users to determine the number of avatars to be placed in a simulation.

5.2.2. Step #2: Positioning the Camera

Once global scene setup is complete a camera is placed in the scene (Figure 6). The camera's primary role is to project 3D points on the global coordinate system back to a local 2D image plane (i.e., pixel coordinate system) through extrinsic and intrinsic transformations. This section explains how a camera is set up before rendering, with an emphasis on defining the appropriate extrinsic and intrinsic camera parameters, as these settings are essential to achieving a high degree of realism.

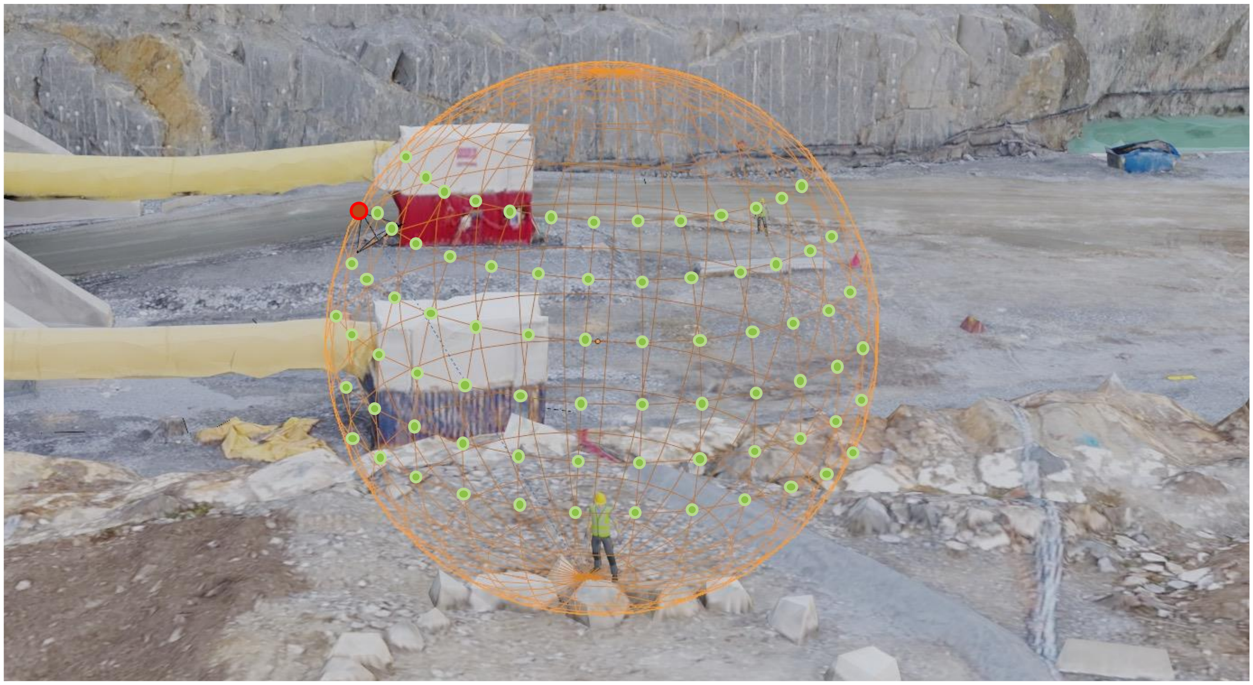


Figure 7. Camera positioning on a user-defined sphere around an animated avatar in BlendCon.

Note: On the sphere, green circles show some of the available positions for camera locations, while red circle shows the randomly selected camera location.

To ensure the presence of at least one worker in the camera's view, a virtual camera is set up around a primary avatar who is chosen at random. Functionally, this is achieved by positioning a camera on a sphere with a radius specified by a user, centered on the target avatar (see **Error!**

Reference source not found.). Our heuristic experiments and observations led us to define a realistic range for the radius of this virtual sphere. We selected the radius from a normal distribution, with an average of 6 meters and a variance of 18 meters.

BlendCon also allows users to define intrinsic camera parameters, namely its focal length and resolution. Through heuristic experiments, we determined that a focal length spanning from 32 to 70 millimeters yields realistic images suitable for our research. The choice of resolution is largely influenced by computational capacity; in our study, we opted for an image resolution of 1920 by 1920 pixels. Table 2 summarizes the camera parameters.

Table 2. Overview of default setting of BlendCon for camera parameters

Camera Parameter	Type	Values (Default; User-adjustable)
Location	Extrinsic	Localized around target avatar with distance with $N(\mu = 6,\ \sigma^2 = 18)$
Orientation	Extrinsic	Set to track the head of the primary target avatar
Focal Length	Intrinsic	Uniform range between 32 to 70 millimeters
Resolution	Intrinsic	1920 by 1920

Note: N represents a normal (Gaussian) distribution.

As a final step we added an occlusion measurement function to camera positioning in BlendCon. This is essential to ensure a clear or partially unobstructed view of the target avatar. By calculating the occlusion percentage of the target avatar from the camera's perspective, this function helps BlendCon place the camera with an optimal view of the scene, ensuring the target avatar remains unblocked by any scene meshes. This function is also beneficial for users who intentionally want to simulate occlusion data, although it was not addressed in this study.

5.2.3. Step #3: Setting the Lighting Conditions

Lastly, lighting conditions need to be set up before the final rendering of the camera-ready simulation (Figure 6). To this end, we employed a sky rendering model proposed by Nishita et al. [51], implemented in Blender as a ‘sky texture’ [52]. This model facilitates the simulation of sky lighting with several parameters including sun size, intensity, elevation, rotation, altitude, density of air molecules, dust and water droplets, and ozone molecules (Table 3). Combined, these parameters enable users to simulate the full spectrum of lighting conditions.

Table 3. Overview of lighting-related parameters in BlendCon

Lighting Parameter	Value
Sun Size	0.545
Sun Intensity	0.1
Sun Elevation	$\frac{\pi}{2} \times N(\mu = 0.5,\ \sigma^2 = 0.16)$
Altitude	0
Density of Air Molecules	$10 \times W(\alpha = 1, \beta = 1.1)$
Density of Dust and Water Droplets	Clear day atmosphere
Density Of Ozone Molecules	Clear day atmosphere

Note: N represents a Normal (Gaussian), and W represents a Weibull distribution.

In this research we aimed for photo-realism. To this end, we utilized the default values for sun size, altitude, dust density, water droplet density, and ozone molecule density as suggested by Nishita et al. [51]. For the reminder of the parameters, we used values determined through our heuristic experiments. A sun intensity of 0.1 best suited our day-time realistic simulation. Sun elevation, which represents the sun’s angular displacement from the horizon in degrees, was observed to provide realistic daytime lighting when values were sourced from a Gaussian distribution with a mean of 0.5 and a variance of 0.16. The last hyperparameter was the density of air molecules which is critical in capturing the reddish hues of the sky at dawn and dusk [51]. For this study, the ideal density was determined experimentally by multiplying ten with a value sourced from a Weibull distribution, which had a shape parameter of one and a scale parameter of 1.1.

5.2.4. Step #4: Rendering

Once the arena (i.e., scene stage), players (i.e., animated construction worker avatars), a camera, and imaging conditions have been set up, BlendCon starts synthetizing images. Rendering of a scene is enabled by a render engine called Cycles [53]. This rendering engine uses NVIDIA-Optix [54] to construct images from the camera's viewpoint. Each image pixel is constructed by sampling ray traces. These rays propagate within the scene, reflecting off materials until they either encounter a light source or hit a set bounce limit [55]. The resulting light capture is translated into pixel values in the image. BlendCon uses an adaptive sampling algorithm [53] to automatically sample ray traces of lights for each pixel to produce images with a controlled level of noise threshold. As a hyperparameter, we set the noise threshold to 0.01 in this study. The noise threshold can be any value between 1 to 0.0001. Tweaking this threshold can speed up the rendering process at the expense of image quality, or conversely, enhance image quality at the cost of rendering time.

Another key hyperparameter is the frame sampling rate. Images are generated in sequences, drawing from the animations of the avatars. Users can determine this sampling rate. In our study we selected a rate of 10 frames per second. With the motion capture data used in BlendCon captured at 100 frames per second (FPS), our research extracted one frame for every 10 frames of the animation sequence.

5.3. Stage #2: Multimodal Labeling

Multimodal labeling in computer vision refers to the process of annotating images with multiple types of information or labels that come from different modalities. These annotations can include spatial coordinates of body joints, known as 3D [56] and 2D [57] keypoints, which are necessary to understand body positions and movements. Similarly, object detection relies on 3D [41,58,59] and 2D [60] bounding boxes to define the parameters of objects in three-dimensional and/or two-dimensional space. For tasks that necessitate depth perception, depth maps [61] provide the distance between objects and the camera. And finally, semantic segmentation masks [62] are utilized to uniquely identify and classify each object in a scene, enhancing the model's ability to understand complex visual inputs. Figure 8 illustrates the different types of labels generated by BlendCon, showcasing how each modality provides distinct yet complementary information for comprehensive scene understanding.

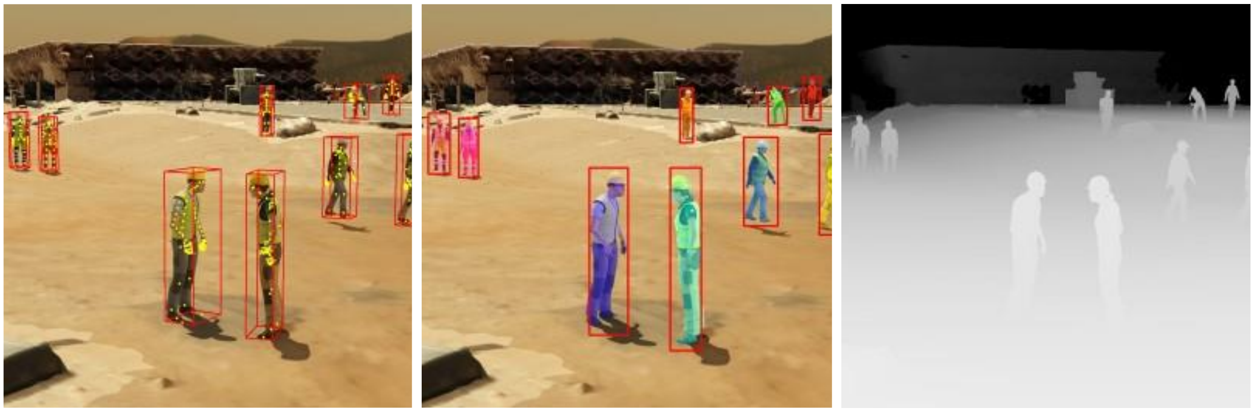


Figure 8. Illustration of BlendCon's multimodal annotation capabilities.

Note: (Left) Visualization of 3D bounding boxes and keypoints, represented by yellow dots, indicating precise joint locations; (Middle) The segmentation mask delineates distinct object boundaries alongside 2D bounding boxes, highlighting object extents; (Right) Depth map translating distance information into grayscale values, reflecting the spatial depth of each element within the scene.

To ensure accuracy and consistency in multimodal labeling, BlendCon employs three distinct coordinate systems. First, there is Blender’s global coordinate system, the natural reference frame for scenes or objects within Blender. Second, the camera’s coordinate system originates on the image plane, with the Z-axis extending outwards, perpendicular to this plane. This system represents the camera's viewpoint, with world coordinates transformed to camera view coordinates through rotation and translation defined by the camera's extrinsic parameters in Blender. Lastly, the pixel coordinate system projects the 3D world into 2D, as perceived by the camera, with coordinates in pixels and the origin at the top left corner of the image.

Crucially, the labels generated by BlendCon are both calibrated and synchronized. Calibration ensures that spatial information is consistently matched across different modalities, providing an accurate representation of objects in the scene. Synchronization guarantees that these modalities are aligned in time, maintaining temporal consistency and ensuring that dynamic events are accurately captured and represented.

Table 4 offers a comprehensive overview of the labels supported by the current version of BlendCon. This table not only delineates each label type but also specifies their corresponding coordinate systems and the functions responsible for generating these labels.

Table 4. Comprehensive summary of BlendCon's labeling capabilities

Label Type	Coordinate System	Function Involved	Data Type
3D Keypoint	Camera Coordinates	Joint Tracking	[x, y, z]
2D Keypoint	Pixel Coordinates	Joint Tracking	[x, y]
3D Bounding Box	Camera Coordinates	Joint Tracking	[x, y, z]
3D Bounding Box	Pixel Coordinates	Joint Tracking	[x, y]
2D Bounding Box	Pixel Coordinates	Joint Tracking	[x, y]
Depth Map	Pixel Coordinates	Depth Estimator	Greyscale Image
Semantic Segmentation	Pixel Coordinates	Semantic Segmentation	Greyscale Image

The labeling process of BlendCon consists of three distinct functions: the (i) joint tracking function; (ii) depth estimation function; and (iii) semantic segmentation function. These functions play an essential role in the accurate depiction of the scene, and the rest of this subsection will delve into the technical details of each function.

5.3.1. Joint Tracking Function

The joint tracking function of BlendCon is designed to capture the kinematic details of 3D animated worker avatars by extracting precise 2D and 3D labels from their bone structures, which are analogous to human skeletal systems. This process is depicted in Figure 9.

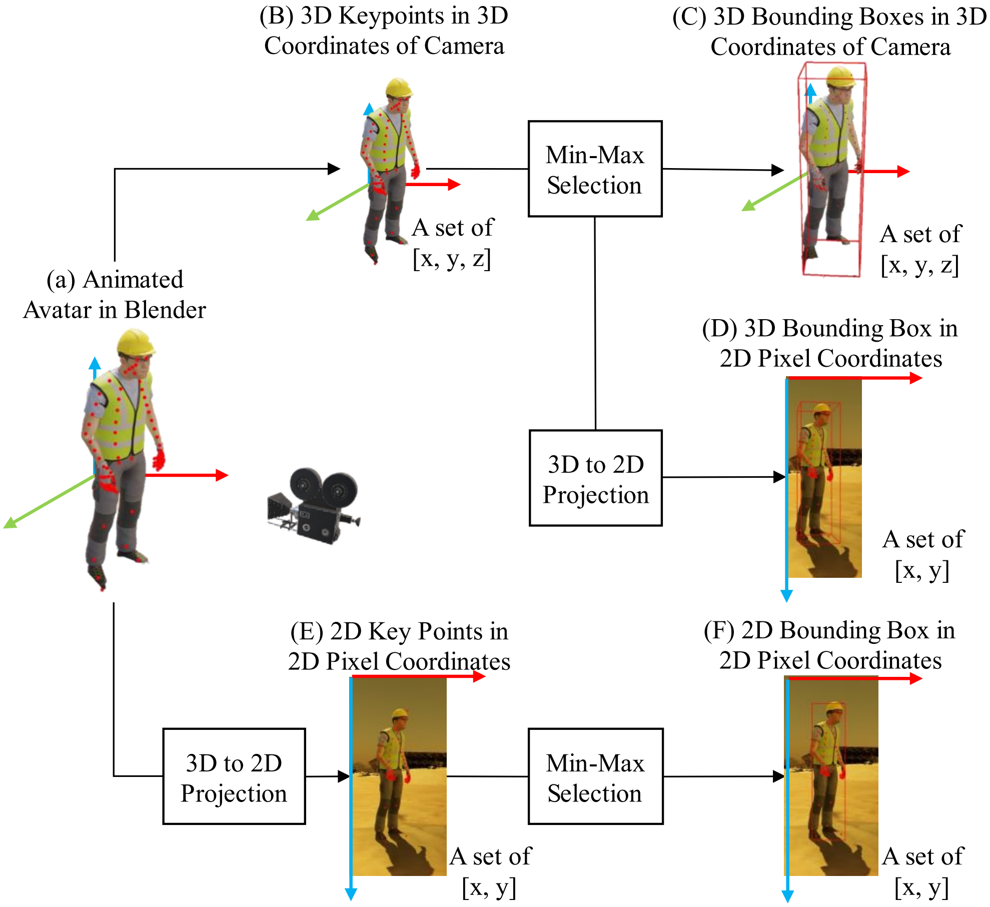

(B) 3D Keypoints in 3D Coordinates of Camera
(C) 3D Bounding Boxes in 3D Coordinates of Camera
Min-Max Selection
A set of [x, y, z]
A set of [x, y, z]
(a) Animated Avatar in Blender
(D) 3D Bounding Box in 2D Pixel Coordinates
3D to 2D Projection
A set of [x, y]
(E) 2D Key Points in 2D Pixel Coordinates
(F) 2D Bounding Box in 2D Pixel Coordinates
3D to 2D Projection
Min-Max Selection
A set of [x, y]
A set of [x, y]

Figure 9. Joint tracking function's data pipeline

Note: Just like real humans, each worker in BlendCon is designed with a bone structure. The joint tracking function leverages these bones and joints to extract labels.

The joint tracking function begins with the positioning of each of the avatar’s joints in the 3D space of Blender’s global coordinate system (see Figure 9 (a)). The function then translates these 3D coordinates from Blender’s global coordinate system into the camera's coordinate system (see Figure 9 (b)). This step generates the labels needed to train 3D pose estimation models. These coordinates are expressed as x, y, and z values in the camera’s coordinate system.

Next, the function employs these coordinates to outline the 3D bounding boxes around the workers. To determine the precise dimensions of the 3D bounding boxes, we calculated the extremities of the keypoints in the camera's coordinate frame (see Figure 9 (c)), which is essential when predicting 3D bounding boxes in 3D coordinates [59].

To meet the requirements of DNN models requiring pixel-level prediction [41,58], the joint tracking function also transforms these 3D bounding boxes defined in the camera coordinate system into 2D projections within the pixel coordinate system. This is achieved by computing a transformation matrix from the camera's intrinsic and extrinsic parameters, projecting the 3D information onto a 2D pixel grid (see Figure 9 (d)).

The function then proceeds to extract 2D keypoints, projecting the 3D joint locations onto the pixel coordinates, resulting in a set of 2D coordinates that serve as training labels for models focused on 2D human pose estimation (see Figure 9 (e)). Subsequently, it applies a min-max selection algorithm to these 2D keypoints to deduce the 2D bounding boxes (see Figure 9 (f)).

5.3.2. Depth Estimation Function

The depth estimation function in BlendCon translates the distances of scene objects from the camera lens into grayscale pixel values, a crucial step for creating depth maps. This transformation is accomplished using Blender's "map range node," which acts as a converter, mapping the distance of an object from the camera to a pixel value on a scale from 0 to 255. This scale corresponds to the grayscale spectrum, allowing for the visualization of distance through shades of gray. Our empirical tests have shown that setting the maximum input range to 100 meters results in the most accurate depth maps, as it optimizes the representation of distances up to that point. Distances beyond 100 meters are assigned the maximum grayscale value, appearing as white in the depth map. By applying this mapping technique, BlendCon can render detailed depth maps that accurately reflect the spatial layout of a scene. It should be highlighted that because the depth map is processed using the same viewpoint as the corresponding RGB image they can be used as a set, i.e., as a 4-channel training source for both 2D and 3D computer vision tasks.

5.3.3. Semantic Segmentation Function

Semantic segmentation is a process in computer vision where each pixel in an image is classified into distinct categories, delineating different objects and elements within the scene. As currently realized within BlendCon, it specifically involves classifying each pixel as part of an individual avatar, the sky, or the background (i.e., any object except avatars and sky in the scene), thus distinctly separating the avatars from other elements. To achieve this, the semantic segmentation function capitalizes on Blender's object index rendering feature. This feature enabled us to assign distinct indices to each avatar individually, while the sky and the background were indexed separately. These indices were then converted into pixel values within the grayscale spectrum. The resulting image, composed of these object-specific indices, enables the precise segmentation of each object based on their unique pixel values.

5.4. BlendCon: A Scalable Solution for Diverse Use Cases

The simplicity and flexibility inherent in BlendCon's design contributes significantly to its scalability, enabling it to effectively meet the demands of diverse use cases, whether in research or industrial settings. To confirm the usability of BlendCon in an academic context, we installed and ran the framework to generate a small dataset comprising 25,000 images. For this purpose, we utilized three NVIDIA GPUs (RTX3080s) on our local machines, and the framework was successful in generating the required dataset for our specific use case. The framework demonstrated similar success in an industrial context at a considerably larger scale. Our industry partner, utilized BlendCon to generate 985,660 images using 260 virtual machines on Amazon AWS service within 2 days. This successful large-scale application was a significant milestone in confirming both the usability and flexibility of BlendCon in the hands of third-party users, as well as the ability of the platform to successfully scale.

6. Validation of the Effectiveness of Synthetic Construction Data: Trainability and Scalability

To assess the impact of synthetic construction data on DNN training we conducted two experiments, with one focusing on trainability and the other scalability. Below is an outline of our experimental design:

- Assigned computer vision task and selected architecture: 2D object detection is central to modern computer vision pipelines. In addition to being a core function, 2D object detection serves as a precursor for advanced tasks like 3D object detection and 2D/3D pose estimation. Our validation experiments targeted the 2D detection of construction workers, employing the YOLOv7 architecture [60]. YOLOv7 is particularly favored among construction researchers due to its accessibility and immediate applicability, alongside its accuracy and computational efficiency.

- Application scenario: While designing the experiments, we decided to focus on a practical scenario where an autonomous ground vehicle (AGV) requires 2D worker detection DNN. Basically, we intended to develop 2D worker detection DNNs for the AGV using two distinct datasets: one consisting solely of synthetic data, and the other using exclusively real data. This approach was intended to demonstrate the genuine value of synthetic data in the development of DNNs for real-world commercial applications. For the purposes of this research, the objective of the trained DNNs was set to reliably detect construction workers within a 20-meter range of the AGV.
- Training approach: To achieve a pure comparison between synthetic and real data, we meticulously trained the YOLOv7 models from scratch across all experiments. In addition, our heuristic observations led us to set our training duration to 800 epochs, which would typically be considered quite long. We noticed that terminating the training at earlier epochs often resulted in the most recent model being the best, suggesting that the saturation point had not been reached. Therefore, to ensure comprehensive learning and to accurately gauge the model's potential we extended the training to 800 epochs. This ensured the models could fully assimilate the dataset, crucial for an unadulterated evaluation of synthetic versus real data training outcomes.
- Evaluation metric: We narrowed our focus to the Average Precision (AP), which has been widely employed in computer vision studies. The AP combines the meaning of both precision and recall, offsetting potential bias that individual metrics could yield. It computes the average of precision values for varying recall levels across multiple thresholds. This comprehensive measure reflects the accuracy and completeness of the detections, making it suitable for evaluating scenarios demanding high recall (detecting as

many objects as possible) and high precision (ensuring these detections are accurate). While adopting the AP, another consideration was the extent of Intersection over Union (IoU). While AP at the IoU of 0.5 (AP@0.5) is a common benchmark — balancing leniency and strictness — it does not suffice, especially in evaluating industry-grade AIs. In our experiments, we chose to assess APs from 0.5 to 0.95 IoUs to capture a nuanced understanding of the model's performance across various localization precision requirements. AP@0.5-0.95 (i.e., the average of the APs across the IoU of 0.5-0.95) was also adopted for a single-shot comparison.

6.1. Synthetic Data Generation and Real Data Collection

Creating or obtaining suitable datasets is a fundamental prerequisite for any machine learning experiment. Our experiments used BlendCon to generate the synthetic dataset, and our team assembled the real dataset through careful curation and combining two public benchmark datasets.

- Generating the synthetic dataset: Using BlendCon with the hyperparameter settings described previously in this paper, we rendered 1,578 sets of construction simulations, resulting in a synthetic dataset with 24,008 images which in turn contained 40,322 bounding boxes. Figure 10 provides samples of the generated synthetic data.

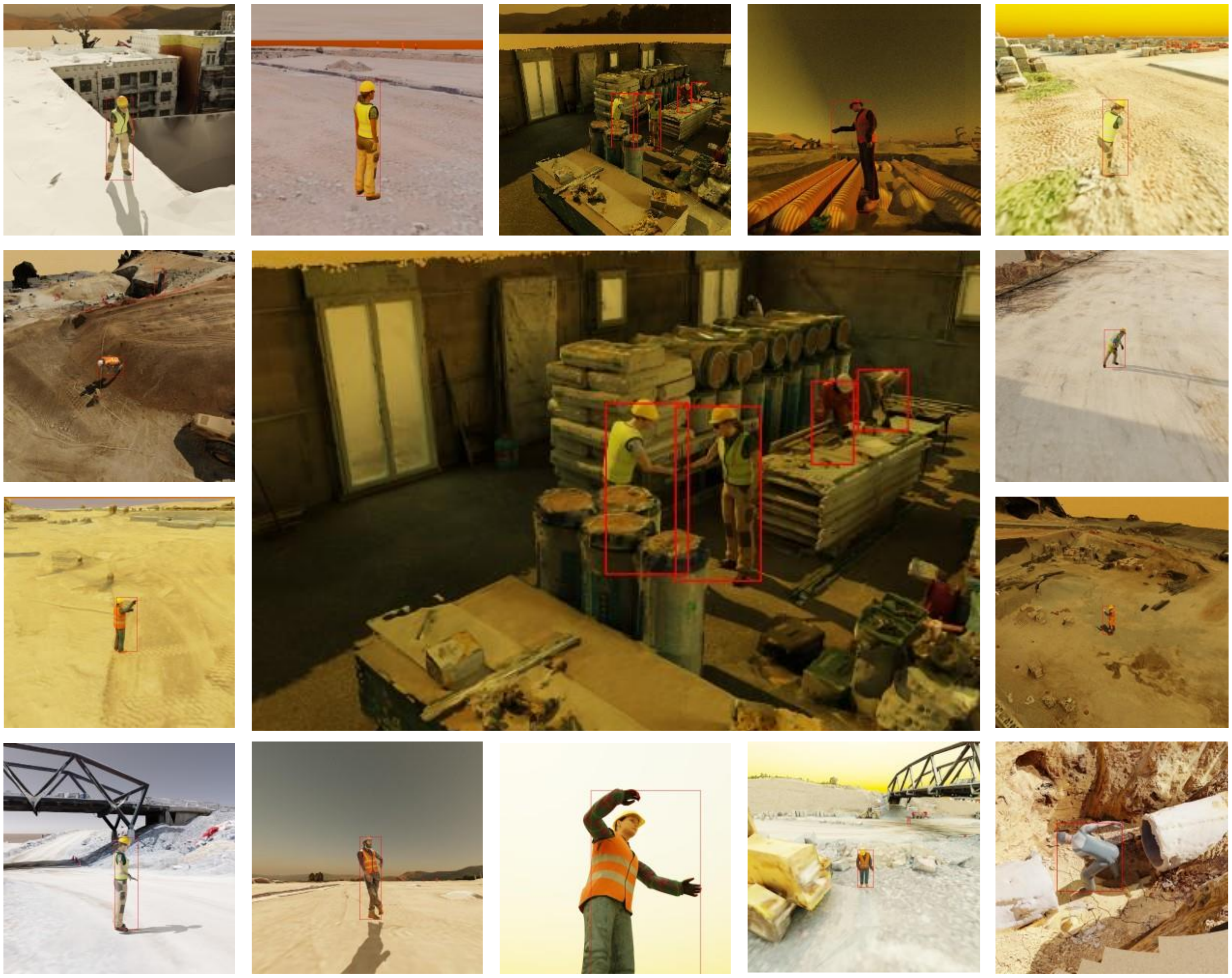


Figure 10. Samples of synthetic images with 2D bounding boxes

- Assembling the real dataset: We utilized two public benchmark datasets as our starting point: (i) Moving Objects in Construction Sites (MOCS) [17]; and (ii) the Large-Scale Small Object Detection Dataset (SODA) [18]. Both datasets are large-scale and tailored for 2D object detection. We selectively filtered these datasets, retaining images featuring human workers along with their corresponding labels, as dictated by our focus on worker detection. By combining images from MOCS and SODA we prepared a diversified and balanced real-image dataset, containing a total of 22,773 images with 74,026 bounding boxes. This integrative approach was deliberately chosen to mitigate potential biases that could arise from relying exclusively on a single dataset.

- Filtering process: Both synthetic and real datasets underwent a series of filters equally. First, filters were applied based on bounding box sizes to suit our use case: worker detection within 20 meters. Validation and test subsets excluded bounding boxes smaller than 9% of the image size, while training subsets removed those smaller than 4%. This approach was based on heuristic observations: workers within 20 meters typically have bounding boxes over 172 pixels, about 9% of a 1920x1920 image. Extending training sets to include images with bounding boxes down to 4% of image size was found to enhance model learning. Exclusively for real dataset, additional filters were applied to sort out night-time shots, blurred, cropped, watermarked images, and aerial drone shots.
- Dataset splitting: In splitting the synthetic dataset between the training, validation, and test sets, we took precautions to prevent any data leakage caused by scene contexts. The data was split based on video names, not individual images, ensuring that during training, the model would not be exposed to familiar scene contexts present in the test or validation sets. As a result, the synthetic training set had 35,708 bounding boxes; the validation set contained 3,689, and the test set had 2,917 bounding boxes. We also split the real dataset based on bounding box counts, sampling images for the test and validation sets until we reached a near thousand bounding boxes for each. The remaining pool of images constituted the training set.

The number of images and bounding boxes in each of the synthetic and real datasets is provided in Table 5.

Table 5. Number of samples in synthetic and real datasets

Dataset Name	Data Type	Training	Validation	Test
Synthetic	Bounding Box	35,708	3,689	925
	Image	20,997	2,336	675
Real	Bounding Box	74,026	998	925
	Image	19,862	418	391

The distribution of bounding boxes in the real and synthetic datasets is demonstrated in Figure 11. In this figure, each bounding box is represented as a single dot. The horizontal axis illustrates the ratio of the bounding box width to the image width, while the vertical axis shows the ratio of the bounding box height to the image height. A small empty square at the bottom right of the graph is evident due to the data filtering process. Additionally, kernel density estimates are plotted on the top and right marginal axes, which demonstrate a close proximity in the peak of the bounding boxes in both datasets.

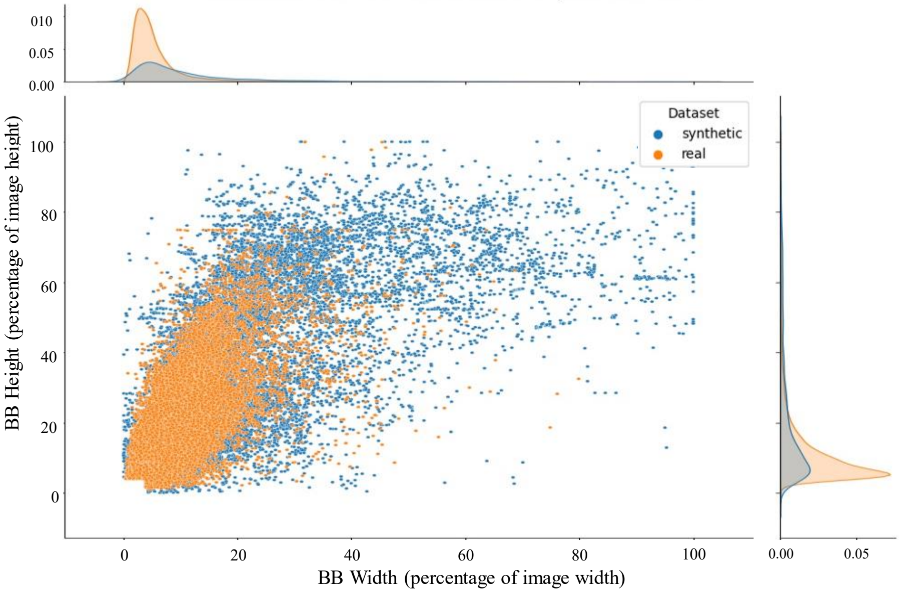


Figure 11. Distribution of bounding box size ratio in real and synthetic datasets

Note: Each bounding box is represented as a single dot. Also, kernel density estimates are plotted on the top and right marginal axes.

6.2. Validation of Trainability

A YOLOv7 model was trained with and validated on synthetic data. For comparison, another model was trained with and validated on real data. Our goal was to visually examine the training loss history for both models to identify any similarities or differences in training patterns between synthetic and real data. Figure 12 shows training and validation losses for both instances. YOLO’s loss function comprises box loss, object loss, and class loss [60]. Class loss is a measure of how accurately the model can identify the correct category of an object in multi-class detection tasks. However, given our dataset's focus on a single class, class loss becomes irrelevant in our context. Box loss measures the accuracy in predicting target object bounding boxes, while object loss evaluates the model’s ability in identifying target object presence within these boxes. The total loss linearly combines box, object, and class losses [60].

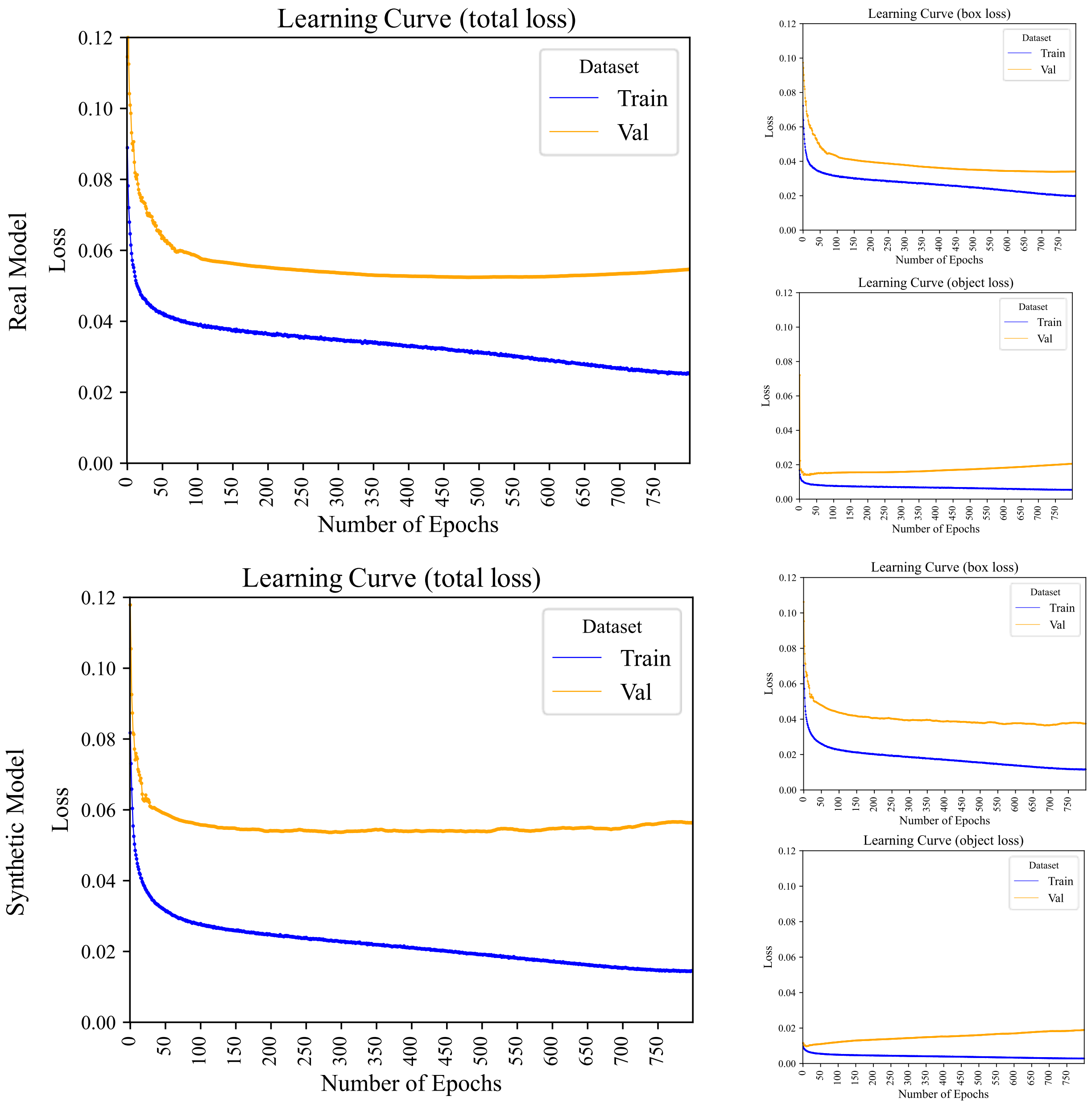


Figure 12. Learning curves of real and synthetic models

Figure 12 provides a comparative analysis of the learning curves for both the synthetic and real YOLOv7 models, which was central to our 'trainability' validation objective. It demonstrates that the synthetic model's learning trajectory mirrors that of the real model, with both exhibiting similar patterns in their loss reduction over epochs.

When examining box loss, both models display a rapid decline in the initial phases of training, indicative of quick adaptation and efficient bounding box prediction. This convergence suggests that both datasets equip the model with sufficient information for identifying object boundaries early in the training process.

In terms of object loss, the early plateau in the synthetic model's learning curve suggests it may not encounter the same dataset complexity as the real data. Here, 'complexity' pertains to the dataset's variety and intricacy. This diversity is critical, as it pressures the model to generalize better to real-world diversity. A less complex dataset may lead to quicker mastery of simpler patterns, as indicated by a steep initial decline in box loss for the synthetic model. However, this could hinder the model's performance in more varied real-world scenarios. Despite these concerns, the parallel trends in object loss for both datasets indicate that the synthetic dataset's simplicity does not significantly impede the model's learning of object detection, suggesting that synthetic data remains a viable training resource.

Furthermore, a consistent decrease in total loss during the early training phase is visible for both models, reinforcing the notion that synthetic data can be as effective as real data for initial learning stages. These observations affirm that training with synthetic data is not significantly different from training with real data in terms of the overall training process and loss patterns. This similarity bolsters our confidence in the efficacy of synthetic data for training DNN models, as it appears to provide a comparable learning experience to real-world data.

6.3. Validation of Scalability

To set the stage for our scalability experiment, we crafted a multi-tiered benchmarking approach. The objective was to analyze the impact of training data volume on model performance and to establish where the synthetic model stands compared to real models trained on datasets of varying size. We segmented the real data pool into smaller, incremental datasets, training multiple YOLOv7 models to create a spectrum ranging from data-scarce to data-rich conditions. We achieved this by sampling a series of datasets, each with different sizes for the training set, but with identical validation and test sets. These training sets were sampled from the pool, ranging from 5,000 to 35,000 samples, in increments of 5,000. This partitioning strategy, detailed in Table 6, enabled us to not only scrutinize the performance gradient across differently trained models but also to pinpoint the synthetic model's relative position in terms of data sufficiency and model robustness.

Table 6. Number of bounding boxes in each training, validation, and test datasets

Dataset Name	Training	Validation	Test
Synthetic	35,708 (synthetic)	3,689 (synthetic)	925 (real)
Real 5K	5004	998	925
Real 10K	10001	998	925
Real 15K	15007	998	925
Real 20K	20005	998	925
Real 25K	25003	998	925
Real 30K	30002	998	925
Real 35K	35017	998	925

Note: It was assumed that no real data are available while training the synthetic model. Thus, no real data were used while training the synthetic model, even for validation.

To provide a fair basis for comparison, we set a limit on our largest real dataset of 35,000 bounding boxes. This decision was taken to prevent the real dataset from surpassing the synthetic dataset. With our seven distinct real datasets at hand, we proceeded to train seven different YOLOv7 models. These models were all trained from scratch, using YOLOv7's default hyperparameters. The AP@0.5-0.95 of these models on the real test dataset are displayed in Figure

13. The result obtained from the synthetic model is displayed as a dashed line. It should be noted that the DNN model trained with synthetic data was also validated with synthetic data, with no real data of any kind used during the development stage, ensuring that the results observed were achieved solely from synthetic data.

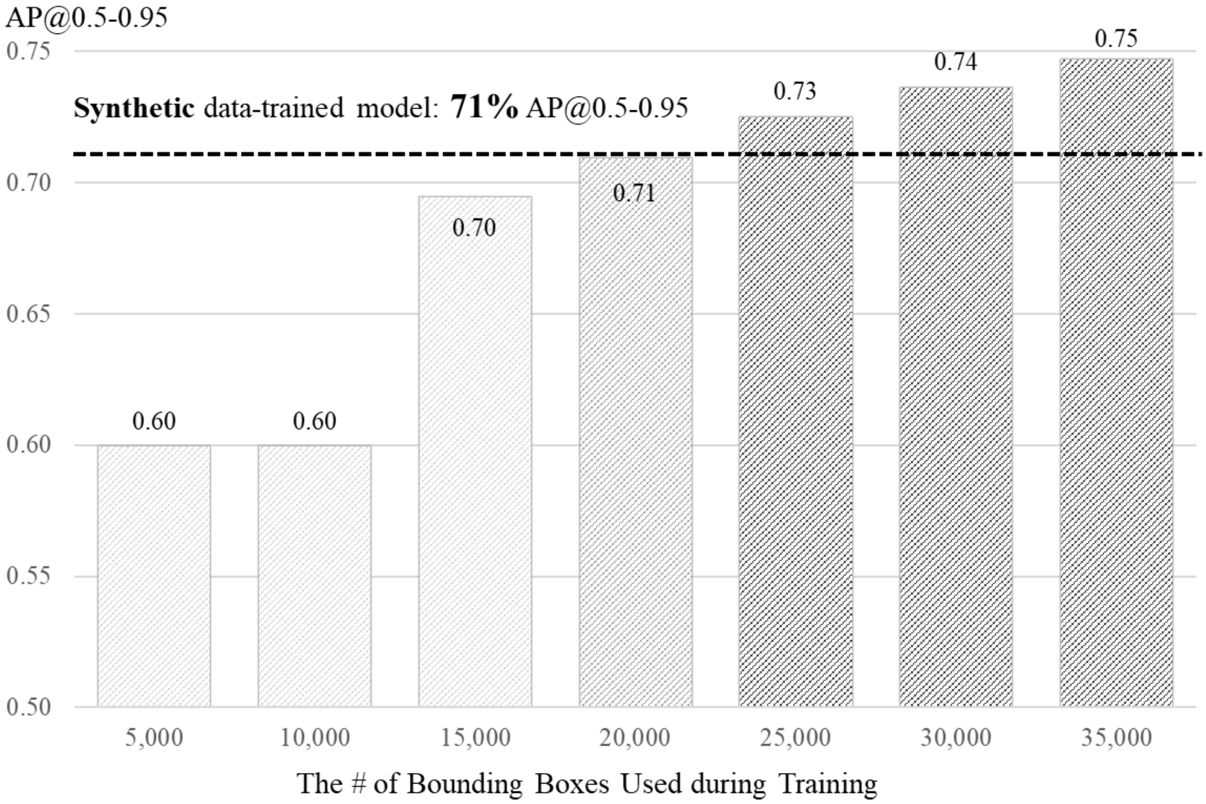


Figure 13. AP@0.5-0.95 of real and synthetic models on real test dataset

Upon analyzing the AP@0.5-0.95, a notable observation emerges. The models trained solely on synthetic data, despite their lack of exposure to real-world scenarios, outperform those trained with limited real data. Further, when the same amount of training samples used (i.e., around 35,000 bounding boxes in this case), the AP@0.5-0.95 difference between the synthetic and real models was merely about 4%. The promising performance of the synthetic model, achieved without any exposure to real images, highlights its potential. It demonstrates the synthetic model's ability to effectively bridge the reality gap and adeptly generalize to the unfamiliar domain of the real world.

Another critical observation is the direct correlation between the volume of training data and the performance of real models. This trend solidifies the idea that the efficacy of DNN models is deeply intertwined with the abundance of data. In scenarios of data scarcity, even the most potent models may falter. It was evident that performance of the models was sub-optimum and stagnant when using training datasets containing 5,000 or 10,000 bounding boxes. However, increasing the size of datasets beyond this threshold realized a significant surge in model efficiency, suggesting a pivotal point in the learning curve.

Upon examination, the performance of a synthetic model trained on a dataset of 35,708 labelled bounding boxes was comparable to a real model trained on a dataset of 20,000 bounding boxes. As the amount of training data increases beyond 20,000 images, real models improve but not dramatically so. This gap in the performance of the synthetic and real DNN models is presumably a manifestation of the reality gap, and it does not significantly widen even with enhanced training data for real models. We posit that this slight disparity can be bridged either by augmenting the diversity or the volume of synthetic data, or by implementing strategies to reconcile domain differences, transitioning smoothly from synthetic to real scenarios.

To interpret these results with more detail, we evaluated the AP at each IoU threshold value, from 0.5 to 0.95. Table 7 presents the APs of all models.

Table 7. Average precision scores at different IoU thresholds.

					IoU Thresholds					
Model Name	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95

Synthetic	0.88	0.86	0.84	0.81	0.79	0.76	0.71	0.63	0.52	0.27
Real 5K	0.91	0.89	0.86	0.82	0.76	0.67	0.56	0.36	0.16	0.02
Real 10K	0.91	0.89	0.86	0.81	0.75	0.66	0.55	0.36	0.16	0.01
Real 15K	0.95	0.94	0.92	0.89	0.84	0.79	0.69	0.55	0.32	0.06
Real 20K	0.96	0.95	0.92	0.89	0.85	0.79	0.71	0.57	0.36	0.09
Real 25K	0.97	0.96	0.93	0.91	0.88	0.82	0.71	0.58	0.38	0.12
Real 30K	0.97	0.96	0.94	0.91	0.88	0.80	0.73	0.60	0.41	0.16
Real 35K	0.97	0.96	0.95	0.92	0.89	0.83	0.74	0.61	0.44	0.16

Table 7 sets the stage for a deeper analysis by comparing the precision of detection at various degrees of overlap between predicted and ground-truth bounding boxes. Figure 14 delineates the AP scores over a range of IoUs, plotting the synthetic model's performance against that of the real models.

Based on Figure 14, the synthetic model initially appears to underperform compared to the real models, particularly at lower Intersection over Union (IoU) thresholds. This trend persists until an IoU of 0.6, where the synthetic model reaches the performance level of the real model trained with the smallest data volume. As the IoU threshold increases, the synthetic model shows progressive improvement, surpassing all real models at an IoU of 0.85. This pattern may be attributed to the inherently precise bounding boxes of synthetic data. In contrast, real-world data labeling, often prone to human error, may result in less accurate bounding box definitions. This becomes more noticeable at higher IoU thresholds, where precision is critical. The decline in AP for real models is steeper than for the synthetic model as IoU thresholds rise, underscoring the value of the meticulous and precise labeling that synthetic data offers. It leads to more consistent detection results, even with tightening IoU thresholds. This observation suggests that for tasks requiring high localization accuracy, investing in the quality of data labeling could be as important, if not more so, than merely increasing the quantity of training data—a potentially significant insight for future research.

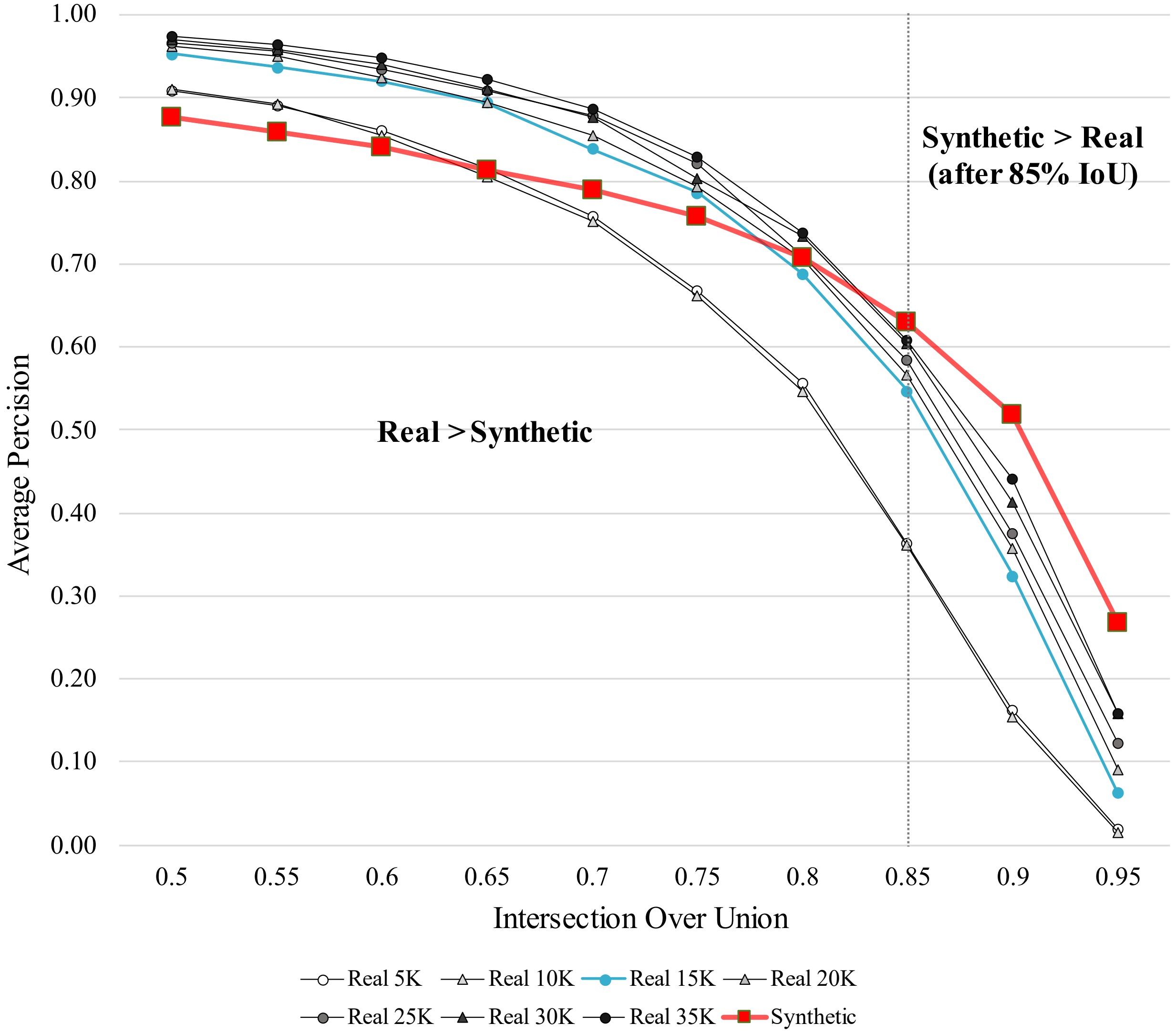


Figure 14. Synthetic vs. Real Model Performance.

7. Discussion: The Impact of Synthetic Data on Construction-Centric DNN

This section explores the potential implications and impacts of using synthetic data in DNN training within the context of the construction industry. Based on the results of our study, we discuss the significant opportunities that the use of BlendCon opens up for researchers and industry professionals. We also highlight a number of unexpected findings, new challenges, and future research directions.

7.1. Unleashing Potential: The Implications of BlendCon in DNN Training

Our study unearthed a significant observation. Synthetic data holds the potential to serve as a primary dataset in DNN training. The parity in performance between models trained on synthetic data and their real-data counterparts, especially when assessed using average precision, is a compelling testament. Such findings underscore the capability of BlendCon-generated synthetic images to effectively replace real images. This positions BlendCon as a uniquely valuable tool in the toolkit of DNN training methodologies.

Moreover, synthetic data proved to be an economical alternative to traditional data collection methods. Our experiments suggest that achieving the DNN performance synthetic data provides would require over 20,000 real images. Synthetic data's ease of generation—merely a few clicks away, particularly with cloud computing—stands in stark contrast to the hefty costs and efforts of collecting and labeling real data. For example, just the manual labeling process for segmentation masks costs around $2.27 per image, costing well over $40,000 for a 20K dataset. Meanwhile, synthetic data, being readily available at the click of a button, eliminates such financial and logistical burdens.

Additionally, employing synthetic data enhances the efficiency of training DNNs for rare but significant scenarios. Tailored synthetic data can be created for unique DNN use-cases, bypassing the challenges of data collection and privacy concerns associated with real construction sites while facilitating training for uncommon but crucial situations. For instance, there might be limited real-world data on specific safety incidents like proximal human-robot collaboration on sites. Sole reliance on this sparse real data isn’t viable. Synthetic data, however, offers cost-effective simulation of such situations, allowing for the development of DNNs adept at managing even highly uncommon and unlikely scenarios.

7.2. Looking Ahead: The Future with BlendCon

In addition to BlendCon's contribution to the training of DNN models for visual AI, this tool opens up new avenues of academic research. Extrapolating from our study's outcomes, we see vast new horizons for synthetic data-led development by both academic researchers and industry professionals. The following spotlights a series of unanticipated insights, emergent challenges, and prospective trajectories in harnessing synthetic data for DNN training endeavors.

7.2.1. BIM-integrated BlendCon: A journey into the future to generate prospective images

One of BlendCon's standout features is its adaptability to diverse input data. BlendCon needs a 3D scene as an arena for the animated avatars. This scene sets the background of the generated images. In this study, we used point clouds as the primary source of the 3D scene. However, the construction sector offers another valuable resource: BIM. BIM files are fully compatible and can be utilized as inputs for BlendCon's 3D scene creation. Tapping into BIM enables the generation of synthetic images with backgrounds that mirror the actual structure to be constructed. Given that building information models are conceived and realized during the early phases of a building's lifecycle, integrating them with BlendCon essentially offers a glimpse into the future, generating imagery from sites yet to be constructed. This forward-thinking image generation allows DNN models to familiarize themselves with forthcoming construction landscapes. In essence, by integrating BIM with BlendCon, we may be able to equip DNN models with a temporal advantage, enabling them to gain acquaintance with future sites.

7.2.2. Beyond Single Modalities: Embracing Multimodal Data Generated by BlendCon

An increasing number of studies indicate that DNN models trained with multimodal data demonstrate improved performance [44]. Multimodal DNNs leverage diverse data inputs—such as RGB, depth, and segmentation indexes—to create more comprehensive data representations, thereby enhancing their capability to perform complex visual tasks. However, a significant hurdle is the dearth of multimodal datasets, which stems from the intricacies involved in both the spatial calibration of data across various sensors and the temporal synchronization of data streams [44]. BlendCon addresses these challenges by efficiently generating and aligning multimodal data for individual scenes, streamlining the process of data collection and labeling. This capability broadens the spectrum of vision tasks that can be addressed. For instance, by integrating multiple modalities into a single training sample—like combining RGB with depth maps for improved 3D perception, or depth with segmentation for advanced spatial recognition—BlendCon paves the way for DNNs to tackle a wider array of applications, from night vision enhancement to complex spatial analyses.

7.2.3. Active Learning with BlendCon: Advancement of DNN Training Paradigms

Beyond its image generation capabilities, BlendCon’s engine offers adaptability. While this study focused on the applicability of the data generated by the platform, it’s also worth noting that the flexibility of the platform itself is a valuable asset. It is a stepping stone towards integrating the platform more directly with the DNN training frameworks. This integration could open the possibility of automated, continuous, and iterative learning for DNN models. We could design a training framework wherein a DNN model identifies its failure cases, maps these failures to the inputs of BlendCon, and generates synthetic images corresponding to that failure. The model could then enrich the existing dataset with synthetic data that is specifically tailored to the observed failure case and retrain the model for continuous improvement. This potential for automated learning and adaptation could revolutionize the way we approach DNN training and significantly enhance model performance over time.

7.3. Investigation of BlendCon's Potential Enhancements

When considering how BlendCon's functionality can be enhanced, two areas stand out: (1) diversity; and (2) level of reality. In synthetic data generation for DNN training, diversity is critical [26]. This encompasses the diversity of avatar poses and shapes, clothing, motions, and background scenes. The more diverse the input to the data generation process, the more inclusive the features in the resulting data, leading to better generalization in real-world applications. However, providing this level of diversity in the data generation process is a challenge. The repository of open-source 3D assets suitable for data generation is not only limited but often comes entangled with licensing complexities that can impede the DNN training workflow. To truly harness the potential of BlendCon in enhancing DNN training, a deliberate investment in developing a high-quality dataset of diversified 3D assets is essential. This involves the accumulation of detailed 3D backgrounds and the creation of dynamic, lifelike animated avatars that closely mimic the appearance and actions of actual construction workers. Our efforts to expand this library of back-end data will be progressive and ongoing.

Realism is the other significant area for improvement. Some studies suggest that the realism of synthetic data plays a vital role in DNN training [63]. The argument posits that increasing the realism of generated images can help DNN models bridge the "reality gap" between synthetic and real domains when deployed in real-world applications. To address the reality gap, recent efforts include advanced generative models that introduce targeted noise into synthetic images to mimic real-world imperfections. With the emergence of controllable diffusion models, these techniques have become more practical, showing promise in increasing the realism of synthetic datasets. However, the challenge of realism extends beyond just the fine details and noise patterns of the images. Enhancing realism would require complex modelling of real-world dynamics, which can be particularly challenging. To enhance realism, the interactions of avatars with their environment need to be modeled in detail. This would include simulating complex scenarios, where multiple entities interact with one another. It would also require the adaptation of worker motions based on their specific working environments. Last but not least, modeling more realistic clothing would be another challenge. In our study, we used fixed meshes attached to the avatars as clothes. To simulate the real world more accurately, we would need to adopt a new approach that allows the modeling of realistic clothing for construction worker avatars.

With these advancements in follow-up studies, BlendCon will have another chance to improve its effectiveness.

8. Conclusion

The construction industry, faced with enduring challenges in productivity and safety, stands to benefit greatly from advancements in DNN-based visual AI. However, the industry’s lack of access to high-quality, diversified data significantly limits this potential. This study addresses this roadblock by presenting BlendCon, a novel computational framework designed to generate synthetic data for DNN training. By harnessing the power of graphic engines, BlendCon can create a realistic virtual replica of a construction site, generating non-real yet realistic images. The framework’s design allows for full control over image properties, lighting conditions, and avatar customizations, enabling a high degree of diversity and flexibility in the synthetic data created. Through this research, we not only introduce a computational (synthetic) data generation solution that eliminates the time-consuming manual data collection process, but also demonstrates the potential of this synthetic data in training scalable, field-applicable DNNs. Our study's experimental findings validate the trainability of models using synthetic datasets and demonstrate that synthetic data can indeed act as a viable substitute for real data, specifically in the task of construction worker detection.

Acknowledgement

This research was supported financially by Natural Sciences and Engineering Research Council of Canada (NSERC) Awards (#1: Discovery Grant, RGPIN-2022-04429, 'Development and Evaluation of Real-like Synthetic Construction Images to Enhance the Performances of Deep Neural Network for Construction Applications' and #2: Collaborative Research and Development Grants, 530550-2018, 'BIM-Driven Productivity Improvements for the Canadian Construction Industry'). The authors would like to extend special thanks to Dr. Julia Panfield and Dr. Veeru Taleja from VelocityEHS, and Mr. Francis Baek from the University of Michigan, for their cooperation in research.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT4.0 in order to spot and revise grammatical errors and typos in our draft manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Gajjar, H., Sanyal, S., and Shah, M., “A Comprehensive Study on Lane Detecting Autonomous Car Using Computer Vision,” *Expert Systems with Applications*, Vol. 233, 2023, p. 120929. https://doi.org/10.1016/j.eswa.2023.120929

[2] Faes, L., Wagner, S. K., Fu, D. J., Liu, X., Korot, E., Ledsam, J. R., Back, T., Chopra, R., Pontikos, N., Kern, C., Moraes, G., Schmid, M. K., Sim, D., Balaskas, K., Bachmann, L. M., Denniston, A. K., and Keane, P. A., “Automated Deep Learning Design for Medical Image Classification by Health-Care Professionals with No Coding Experience: A Feasibility Study,” *The Lancet Digital Health*, Vol. 1, No. 5, 2019, pp. e232–e242. https://doi.org/10.1016/S2589-7500(19)30108-6

[3] Boutros, F., Struc, V., Fierrez, J., and Damer, N., “Synthetic Data for Face Recognition: Current State and Future Prospects,” *Image and Vision Computing*, Vol. 135, 2023, p. 104688. https://doi.org/10.1016/j.imavis.2023.104688

[4] Karami, A., Rigon, S., Mazzacca, G., Yan, Z., and Remondino, F., “NERFBK: A High-Quality Benchmark for NERF-Based 3D Reconstruction,” presented at the Computer Vision and Pattern Recognition, 2023. https://doi.org/10.48550/arXiv.2306.06300

[5] Kuriakose, B., Shrestha, R., and Sandnes, F. E., “DeepNAVI: A Deep Learning Based Smartphone Navigation Assistant for People with Visual Impairments,” *Expert Systems with Applications*, Vol. 212, 2023, p. 118720. https://doi.org/10.1016/j.eswa.2022.118720

[6] Komatsu, “Intelligent Machine Control | Smart Construction | Komatsu,” Komatsu. Retrieved 12 September 2023. https://www.komatsu.com/en/site-optimization/smart-construction/intelligent-machine-control/

[7] “Exosystem^{TM}. The World’s First Fully Autonomous Upgrade for Heavy…,” Built Robotics. Retrieved 12 September 2023. https://www.builtrobotics.com/technology/exosystem

[8] BostonDynamics, “Spot® - The Agile Mobile Robot,” Boston Dynamics. Retrieved 5 February 2022. https://www.bostondynamics.com/products/spot

[9] “Husky Observer,” Clearpath Robotics. Retrieved 12 September 2023. https://clearpathrobotics.com/husky-observer/

[10] Braun, A., Tuttas, S., Borrmann, A., and Stilla, U., “Improving Progress Monitoring by Fusing Point Clouds, Semantic Data and Computer Vision,” *Automation in Construction*, Vol. 116, 2020, p. 103210. https://doi.org/10.1016/j.autcon.2020.103210

[11] Cheng, M.-Y., Cao, M.-T., and Nuralim, C. K., “Computer Vision-Based Deep Learning for Supervising Excavator Operations and Measuring Real-Time Earthwork Productivity,” *The*

Journal of Supercomputing, Vol. 79, No. 4, 2023, pp. 4468–4492. https://doi.org/10.1007/s11227-022-04803-x

[12] Kim Daeho, Lee SangHyun, and Kamat Vineet R., “Proximity Prediction of Mobile Objects to Prevent Contact-Driven Accidents in Co-Robotic Construction,” *Journal of Computing in Civil Engineering*, Vol. 34, No. 4, 2020, p. 04020022. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000899

[13] Deng, J., Singh, A., Zhou, Y., Lu, Y., and Lee, V. C.-S., “Review on Computer Vision-Based Crack Detection and Quantification Methodologies for Civil Structures,” *Construction and Building Materials*, Vol. 356, 2022, p. 129238. https://doi.org/10.1016/j.conbuildmat.2022.129238

[14] Boje, C., Guerriero, A., Kubicki, S., and Rezgui, Y., “Towards a Semantic Construction Digital Twin: Directions for Future Research,” *Automation in Construction*, Vol. 114, 2020, p. 103179. https://doi.org/10.1016/j.autcon.2020.103179

[15] Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P., “Deep Learning with Limited Numerical Precision,” presented at the International conference on machine learning, 2015.

[16] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “ImageNet Classification with Deep Convolutional Neural Networks,” Vol. 25, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 2012.

[17] Xuehui, A., Li, Z., Zuguang, L., Chengzhi, W., Pengfei, L., and Zhiwei, L., “Dataset and Benchmark for Detecting Moving Objects in Construction Sites,” *Automation in Construction*, Vol. 122, 2021, p. 103482. https://doi.org/10.1016/j.autcon.2020.103482

[18] Duan, R., Deng, H., Tian, M., Deng, Y., and Lin, J., “SODA: A Large-Scale Open Site Object Detection Dataset for Deep Learning in Construction,” *Automation in Construction*, Vol. 142, 2022, p. 104499. https://doi.org/10.1016/j.autcon.2022.104499

[19] Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., and Wang, X., “Computer Vision Techniques in Construction: A Critical Review,” *Archives of Computational Methods in Engineering*, Vol. 28, No. 5, 2021, pp. 3383–3397. https://doi.org/10.1007/s11831-020-09504-3

[20] “Google Data Labeling Service,” Google Cloud. Retrieved 11 July 2023. https://cloud.google.com/ai-platform/data-labeling/pricing

[21] Delgado, J. M. D., and Oyedele, L., “Deep Learning with Small Datasets: Using Autoencoders to Address Limited Datasets in Construction Management,” *Applied Soft Computing*, Vol. 112, 2021, p. 107836. https://doi.org/10.1016/j.asoc.2021.107836

[22] Akinosho, T. D., Oyedele, L. O., Bilal, M., Ajayi, A. O., Delgado, M. D., Akinade, O. O., and Ahmed, A. A., “Deep Learning in the Construction Industry: A Review of Present Status

and Future Innovations," *Journal of Building Engineering*, Vol. 32, 2020, p. 101827. https://doi.org/10.1016/j.jobe.2020.101827

[23] "ImageNet." Retrieved 11 July 2023. https://www.image-net.org/index.php

[24] "CelebA Dataset." Retrieved 11 July 2023. https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[25] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 7, 2014, pp. 1325–1339. https://doi.org/10.1109/TPAMI.2013.248

[26] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S., "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization," presented at the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018. https://doi.org/10.1109/CVPRW.2018.00143

[27] "By 2024, 60% of the Data Used for the De-vel-op-ment of AI and An-a-lyt-ics Projects Will Be Syn-thet-i-cally Gen-er-ated," Andrew White, Jul 24 2021. Retrieved 28 July 2023. https://blogs.gartner.com/andrew_white/2021/07/24/by-2024-60-of-the-data-used-for-the-development-of-ai-and-analytics-projects-will-be-synthetically-generated/

[28] ML2Grow, "Synthetic Data: A Game-Changer for AI," ML2Grow, Aug 03 2022. Retrieved 28 July 2023. https://ml2grow.com/synthetic-data-a-game-changer-for-ai/

[29] Toews, R., "Synthetic Data Is About To Transform Artificial Intelligence," Forbes. Retrieved 28 July 2023. https://www.forbes.com/sites/robtoews/2022/06/12/synthetic-data-is-about-to-transform-artificial-intelligence/

[30] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T., "Flownet: Learning Optical Flow with Convolutional Networks," 2015.

[31] Neuhausen, M., Herbers, P., and König, M., "Using Synthetic Data to Improve and Evaluate the Tracking Performance of Construction Workers on Site," *Applied Sciences*, Vol. 10, No. 14, 2020, p. 4948. https://doi.org/10.3390/app10144948

[32] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M., "The Synthia Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," 2016.

[33] Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., and Cucchiara, R., "MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?," 2021.

[34] Acharya, D., Khoshelham, K., and Winter, S., "BIM-PoseNet: Indoor Camera Localisation Using a 3D Indoor Model and Deep Learning from Synthetic Images," *ISPRS Journal of*

Photogrammetry and Remote Sensing, Vol. 150, 2019, pp. 245–258. https://doi.org/10.1016/j.isprsjprs.2019.02.020

[35] Ma, J. W., Czerniawski, T., and Leite, F., “Semantic Segmentation of Point Clouds of Building Interiors with Deep Learning: Augmenting Training Datasets with Synthetic BIM-Based Point Clouds,” *Automation in Construction*, Vol. 113, 2020, p. 103144. https://doi.org/10.1016/j.autcon.2020.103144

[36] Hong, Y., Park, S., Kim, H., and Kim, H., “Synthetic Data Generation Using Building Information Models,” *Automation in Construction*, Vol. 130, 2021, p. 103871. https://doi.org/10.1016/j.autcon.2021.103871

[37] Ying, H., Sacks, R., and Degani, A., “Synthetic Image Data Generation Using BIM and Computer Graphics for Building Scene Understanding,” *Automation in Construction*, Vol. 154, 2023, p. 105016. https://doi.org/10.1016/j.autcon.2023.105016

[38] Soltani, M., Zhu, Z., and Hammad, A., “Framework for Location Data Fusion and Pose Estimation of Excavators Using Stereo Vision,” *Journal of Computing in Civil Engineering*, Vol. 32, No. 6, 2018, p. 04018045. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000783

[39] Kim, H., and Kim, H., “3D Reconstruction of a Concrete Mixer Truck for Training Object Detectors,” *Automation in Construction*, Vol. 88, 2018, pp. 23–30. https://doi.org/10.1016/j.autcon.2017.12.034

[40] Mahmood, B., Han, S., and Seo, J., “Implementation Experiments on Convolutional Neural Network Training Using Synthetic Images for 3D Pose Estimation of an Excavator on Real Images,” *Automation in Construction*, Vol. 133, 2022, p. 103996. https://doi.org/10.1016/j.autcon.2021.103996

[41] Huang, K.-C., Wu, T.-H., Su, H.-T., and Hsu, W. H., “MonoDTR: Monocular 3D Object Detection with Depth-Aware Transformer,” presented at the Computer Vision and Pattern Recognition, 2022. https://doi.org/10.48550/arXiv.2203.10981

[42] Deng, L., Yang, M., Li, T., He, Y., and Wang, C., “RFBNet: Deep Multimodal Networks with Residual Fusion Blocks for RGB-D Semantic Segmentation,” presented at the Computer Vision and Pattern Recognition, 2019. https://doi.org/10.48550/arXiv.1907.00135

[43] Rahate, A., Walambe, R., Ramanna, S., and Kotecha, K., “Multimodal Co-Learning: Challenges, Applications with Datasets, Recent Advances and Future Directions,” *Information Fusion*, Vol. 81, 2022, pp. 203–239. https://doi.org/10.1016/j.inffus.2021.12.003

[44] Bayoudh, K., Knani, R., Hamdaoui, F., and Mtibaa, A., “A Survey on Deep Multimodal Learning for Computer Vision: Advances, Trends, Applications, and Datasets,” *The Visual Computer*, Vol. 38, No. 8, 2022, pp. 2939–2970. https://doi.org/10.1007/s00371-021-02166-7

[45] Mullick, K., Jain, H., Gupta, S., and Kale, A. A., "Domain Adaptation of Synthetic Driving Datasets for Real-World Autonomous Driving," presented at the Computer Vision and Pattern Recognition, 2023. https://doi.org/10.48550/arXiv.2302.04149

[46] Kim, J., Kim, D., Lee, S., and Chi, S., "Hybrid DNN Training Using Both Synthetic and Real Construction Images to Overcome Training Data Shortage," *Automation in Construction*, Vol. 149, 2023, p. 104771. https://doi.org/10.1016/j.autcon.2023.104771

[47] Hwang, J., Kim, J., and Chi, S., "Site-Optimized Training Image Database Development Using Web-Crawled and Synthetic Images," *Automation in Construction*, Vol. 151, 2023, p. 104886. https://doi.org/10.1016/j.autcon.2023.104886

[48] Blender Foundation, "Blender," 2023. Retrieved 10 April 2022. https://www.blender.org/

[49] Rokoko Electronics, "Smartsuit Pro: Quality Motion Capture in One Simple Suit." Retrieved 25 September 2021. https://www.rokoko.com/products/smartsuit-pro

[50] "Real-Time Motion Capture in Blender with Rokoko's Native Integration." Retrieved 11 July 2023. https://www.rokoko.com/integrations/blender

[51] Nishita, T., Sirai, T., Tadamura, K., and Nakamae, E., "Display of the Earth Taking into Account Atmospheric Scattering," presented at the Proceedings of the 20th annual conference on Computer graphics and interactive techniques, New York, NY, USA, 1993. https://doi.org/10.1145/166117.166140

[52] Blender 3.6 Reference Manual, "Sky Texture Node," Blender 3.6 Reference Manual. https://docs.blender.org/manual/en/latest/render/shader_nodes/textures/sky.html#sky-texture-node

[53] the Blender Foundation, "Cycles Rendering Engine." Retrieved 23 August 2023. https://www.cycles-renderer.org/

[54] Blender Foundation, "Accelerating Cycles Using NVIDIA RTX," Developer Blog. Retrieved 3 September 2023. https://code.blender.org/2019/07/accelerating-cycles-using-nvidia-rtx/

[55] Blender Foundation, "Light Paths — Blender Manual." Retrieved 25 September 2023. https://docs.blender.org/manual/en/latest/render/cycles/render_settings/light_paths.html

[56] Martinez, J., Hossain, R., Romero, J., and Little, J. J., "A Simple Yet Effective Baseline for 3d Human Pose Estimation," presented at the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017. https://doi.org/10.1109/ICCV.2017.288

[57] Chen, H., Feng, R., Wu, S., Xu, H., Zhou, F., and Liu, Z., "2D Human Pose Estimation: A Survey," *Multimedia Systems*, Vol. 29, No. 5, 2023, pp. 3115–3138. https://doi.org/10.1007/s00530-022-01019-0

[58] Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., and Luo, P., "Learning Depth-Guided Convolutions for Monocular 3D Object Detection," presented at the Computer Vision and Pattern Recognition, 2019. https://doi.org/10.48550/arXiv.1912.04799

[59] Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., and Han, S., "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation," presented at the Computer Vision and Pattern Recognition, 2022.

[60] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M., "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," presented at the Computer Vision and Pattern Recognition, 2022.

[61] Laga, H., Jospin, L. V., Boussaid, F., and Bennamoun, M., "A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, No. 4, 2022, pp. 1738–1764. https://doi.org/10.1109/TPAMI.2020.3032602

[62] Hafiz, A. M., and Bhat, G. M., "A Survey on Instance Segmentation: State of the Art," *International Journal of Multimedia Information Retrieval*, Vol. 9, No. 3, 2020, pp. 171–189. https://doi.org/10.1007/s13735-020-00195-x

[63] Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., and Shotton, J., "Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.