Accelerating Indoor Construction Progress Monitoring with Synthetic Data-Powered Deep Learning

Mathis Baubriaud[1,2], Stéphane Derrode[1], René Chalon[1] and Kevin Kernn[2]

[1]Centrale Lyon, CNRS, Claude Bernard Lyon 1, INSA Lyon, Lumière Lyon 2, LIRIS, UMR5205, 69130 Ecully, France
[2]SPIE Building Solutions, F-69320 Feyzin, France

(mathis.baubriaud, stephane.derrode, rene.chalon)@ec-lyon.fr, kevin.kernn@spie.com

Abstract -

In recent years, there has been a growing interest in automated indoor construction progress monitoring (ICPM) to maximize precision and reduce human intervention. Computer vision approaches, especially based on deep learning (DL) methods, have shown great potential in this task. However, training DL models require large-scale datasets, which are often costly and laborious to obtain, specifically for indoor construction environments. This study proposes an automated approach to generate real-world-like synthetic data of indoor construction by combining building information modeling (BIM) and a photorealistic graphics engine. The approach was validated by efficiently producing annotated synthetic datasets of mechanical, electrical, and plumbing components from various BIM models. A state-of-the-art instance segmentation network was trained using those datasets alongside real manually annotated data and transfer learning methods to assess the results. Preliminary experiments using an on-site augmented reality device demonstrate the promising efficiency of DL for ICPM.

Keywords -

Building Information Modeling; Computer Vision; Deep Learning; Progress Monitoring; Indoor Construction; Augmented Reality.

1 Introduction

Progress monitoring of construction sites is essential because it gives managers the information needed to act quickly and wisely. Ineffective progress monitoring leads to a loss of control, causing time and cost overruns. The manual data entry required by conventional progress monitoring techniques is laborious, time-consuming and prone to human error [1]. Inspections of interior work — for example, mechanical, electrical, and plumbing (MEP) installation — can be even more difficult for inspectors due to the level of detail and interdependence of tasks [2]. Therefore, it becomes desirable to automate these tasks.

The use of computer vision (CV) has been studied on a wide range of construction applications, such as site safety inspection [3], localization, navigation [4], and 3D reconstruction [5, 6], among others. Several studies on vision-based construction progress monitoring have also been reported [7]. Deep learning (DL) techniques excel in numerous CV tasks and is increasingly used in this field [8]. Although DL models are capable of powerful feature representation, they rely heavily on large-scale, high-quality training datasets. Three methods can be found commonly in the literature to acquire such datasets (1) using pre-existing datasets, (2) using web crawling techniques, and (3) by capturing the data manually [9].

Regarding (1), the CV community has produced a number of publicly available datasets, including ImageNet [10], S3DIS [11], MS COCO [12], and ADE20K [13]. Although a handful of datasets such as MOCS (moving objects in construction sites) [14] and CIS (construction instance segmentation) [15], targets the construction domain, they mainly focus on outdoor environments, leaving indoor environments underexplored.

The second and third data collection techniques present a unique set of challenges that hinder the creation of large-scale, high-quality real-world datasets. On one hand, data collection through web crawling requires manual review for quality, privacy regulations, intellectual property rights, and consistency. On the other hand, capturing data in the real world involves tedious work, access to construction sites, and specialized expertise [16]. The data annotation step, arguably the most time-consuming task due to the dataset's large scale, follows the data collection process. Crowdsourcing services like Amazon SageMaker and Google Cloud Vertex AI can significantly reduce the manual labor involved, but they remain costly and prone to error [17].

Because the acquisition of real-world datasets is a challenging and resource-intensive endeavor, researchers have also explored the generation of synthetic data from controllable and computable virtual environments as a cost-effective and efficient alternative [18]. In the realm of synthetic image data generation in construction, there have been a number of virtual environments constructed using 3D modeling and computer graphics software such as Revit, Blender, Unreal and Unity [19, 20, 21]. DL models trained on synthetic or mixed datasets have been shown to

outperform or achieve comparable performance to models trained solely on real images [22].

One emerging source of data over the past decade comes from the building information modeling (BIM). BIM is a set of interacting policies, processes and technologies aimed at managing the essential data of a construction site, in digital form, throughout its life cycle [23]. BIM models have become an appropriate data source for generating synthetic images of indoor building scenes due to the inclusion of accurate geometry and sometimes appearance information (*i.e.*, material and texture). BIM authoring tools can provide virtual cameras and several studies have leveraged their capabilities to generate synthetic data [24, 25]. However, the potential of BIM models combined with computer graphics software has not been fully explored and there is no universal approach to generate data automatically.

This paper proposes a procedural approach to generate synthetic datasets from BIM models to address the data collection and availability drawbacks mentioned above. To test this approach, three synthetic datasets were generated from three different BIM models (*i.e.* Figure 1) and used to train and test a state-of-the-art instance segmentation model. Lastly, a preliminary approach is presented that compares the prediction of the trained neural network model with the BIM model view through an augmented reality (AR) device.

Figure 1. Snapshots of three BIM projects imported in the graphic engine.

This article is structured as follows: Section 2 delves into the technical aspects of synthetic image generation and annotation. Section 3 presents an experimental procedure to evaluate the framework performance, followed by a concise experimental study showcasing its practical application. Finally, Section 4 summarizes the findings and outline directions for future research.

2 Methodology

The objective of this study is to develop an automated approach that generates photorealistic synthetic RGB images of indoor building scenes with comprehensive annotations using BIM and a computer graphics engine. Figure 2 illustrates the workflow of the approach, each step is further described in the following.

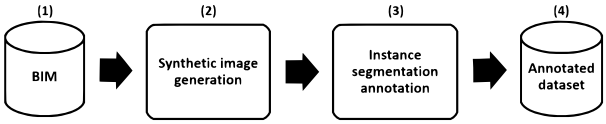

Figure 2. Framework to generate annotated photorealistic images from a BIM model.

2.1 Synthetic image generation

Since the study focuses on indoor construction sites, the first step is to acquire BIM models of such projects. A BIM model for a building is typically created by a team of professionals from various trades, each contributing their expertise to the development of the model. These trades can be broadly categorized into the following groups: architecture, structural engineering, MEP and interior design. The MEP trade, representing the construction phase with the most dynamic and complex changes throughout a project's life cycle, is a critical area for progress monitoring automation. Therefore, we focus on generating a synthetic dataset in this field.

NVIDIA Isaac Sim is a robotics simulation toolkit for the NVIDIA Omniverse platform that provides researchers and practitioners with the tools and workflows they need to create robust, physically accurate simulations and synthetic datasets. With the use of a Revit plugin, one can directly import construction projects into Isaac Sim, making it a viable tool for synthetic dataset generation using BIM. For each physical object, the following information is obtained: object class, instance ID, and triangular mesh. However, the projects lack materials and textures, which we need to add through Isaac Sim to create a photorealistic environment.

The application of true-to-life textures to each object class within the BIM environment can be achieved through the utilization of the application programming interface (API) and the Omniverse material library. By leveraging the unique IDs associated with each object instance, it is possible to efficiently apply a variety of textures to specific object classes, such as pipes, for enhanced realism. For

example, red and blue metal textures can be applied to represent warm and cold currents, respectively, while black foam textures can be used to depict insulation. Additionally, the ability to create or import custom texture packs from external platforms, along with the incorporation of 3D models of specific elements, such as fan coils, further enhances the realism of the BIM environment. Moreover, the accurate representation of lighting, including both indoor and natural illumination, is crucial for achieving a realistic virtual representation. Artificial lighting should be meticulously tailored to match the lighting fixtures specified within the BIM model, while additional light sources may need to be strategically placed to compensate for the absence or non-functional state of lighting fixtures at the actual construction site. Various rendering parameters can be adjusted to optimize the depiction of light behavior, its interaction with objects, their reflectivity, colors, and transparency/absorption of materials. To achieve the highest level of fidelity, we use the interactive path tracing mode instead of approximation methods that prioritize performance while sacrificing accuracy.

The camera sensor simulation provides granular control over parameters like lens properties, aperture, shutter, clipping, and fisheye distortion, replicating real-world camera behavior. Key parameters include focal length, field of view, output resolution, and focus distances. It is important to note that digital cameras may produce radial and tangential distortions due to manufacturing imperfections in their lenses. The virtual camera supports various distortion models, with popular options including rational polynomial, brown conrady, and fisheye. Post-processing enhancements, including exposure adjustment, tone mapping, color grading, color correction, depth of field, motion blur, bloom, and others, further refine the simulated camera feed. We target a low-end camera simulation mimicking a laptop webcam for embedding compatibility.

After configuring the virtual camera, the next step is to determine its possible positions within each scene. This involves defining routes that mimic the movement of a worker inspecting the construction site. The API provides a tool for manually creating these routes. While grid-based viewpoints could be used, manually drawn routes provide a more natural representation of an inspector's movement.

2.2 Instance segmentation annotation

We use the semantics schema API to associate semantic data to each object class in our dataset. The synthetic data recorder options are RGB, depth, semantic and instance segmentation, 2D tight and loose bounding box. For this dataset, we only select RGB and instance segmentation data, capturing one image per half second along the predefined route.

So far, valid photorealistic synthetic images have been generated. We now need to translate the semantic data to a DL model annotation format. We chose to use the Darknet text format compatible with the YOLOv8 instance segmentation model, one of the latest iteration of the popular YOLO model [26] released in January 2023 by Ultralytics. The architecture of YOLOv8 combines a convolutional neural network (CNN) backbone with a self-attention mechanism to achieve high accuracy and speed for object detection. This CNN model's high-speed inference makes it ideal for real-time applications on edge devices. Additionally, a segmentation head is added to predict the binary segmentation masks for each object detected.

Each line in a Darknet text file corresponds to a single object annotation. We create a Python script to attribute to each pair of RGB and semantic image a text file containing each object's class IDs, bounding box coordinates, and the mask coordinates, determined using the Ramer–Douglas–Peucker (RDP) algorithm. The RDP algorithm is a curve-fitting algorithm that simplifies the contours of an object by reducing the number of points that describe it while preserving its shape. By configuring the spatial resolution, minimum surface, and the number of points describing a contour, we can eliminate objects that are too small.

3 Experimental study

3.1 The generated synthetic dataset

We perform a series of experiments to evaluate the proposed approach. All experiments are run on a laptop equipped with an Intel Core i7-10750H CPU, 32 GB of RAM, and a NVIDIA Quadro RTX 3000 GPU. SPIE Building Solutions, a subsidiary of SPIE, provides access to three construction projects for the experiment (see Figure 1). The first project is an eight-story tower that will be used as an office, the second is an extension to a scientific university laboratory, and the third is a completely new site for a business school, making them an excellent sample of tertiary construction projects. Prior to importing the geometry into Isaac Sim, filters are applied to the BIM model in Revit to eliminate all interior design elements and architectural components that obstruct the MEP components. A route is created within each building, traversing each floor and alternating the viewpoints.

The dataset generated in this study is named MEP-SEG. By capturing one image every half a second of the created route, we collected 8,751 samples from the BIM projects, with a rendering time of approximately 9 hours. The distribution of assets for each of the 13 classes is shown in Table 1. Examples of the generated samples are depicted in Figure 3. As shown, some classes are severely under-represented compared to others. This

Table 1. Distribution of assets among the 13 classes in the generated dataset.

Class	No. of assets
Wall	90801
Pipe	44998
Floor	44266
Circular duct	34973
Rectangular duct	26227
Framework	11627
Air vent	8585
Pole	5131
Fan coil	4286
Radiant panel	3031
Ceiling	2431
Pipe accessory	1449
Climatic equipment	1309

will have to be taken into account when using this dataset. The entierety of the data is publicly available at : https://datasets.liris.cnrs.fr/mep-seg-dataset-synthetic-images-generated-building-information-modeling-bim-v1.

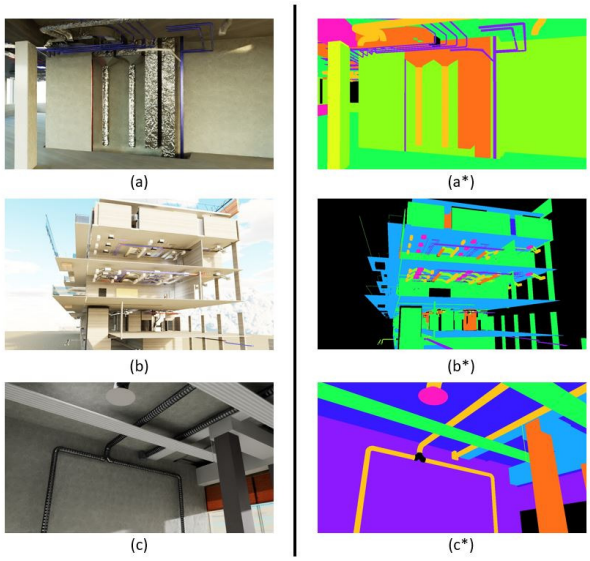

Figure 3. Three synthetic images generated on the left column and their corresponding semantic segmentation image on the right column.

3.2 Evaluation procedure

3.2.1 Evaluation of synthetic pre-trained models on small real datasets

For this initial evaluation, we compare transfer learning (TL) using two different pre-trained models of the same YOLOv8 architecture. The first model is pre-trained on the COCO image dataset [12], the second model on our MEP-SEG dataset. The objective is to determine whether using a synthetic dataset in the targeted domain can yield better performance compared to a generic but more comprehensive public dataset. We focused on detecting only three classes: circular ducts, rectangular ducts, and pipes. These objects are among the most challenging to detect in images of the MEP domain due to their various shapes, textures, and sizes. We extracted these three classes from our MEP-SEG dataset, resulting in a duct and pipe dataset named D&P-SEG. A blank YOLOv8 neural network was trained using the API of YOLOv8. All available images are distributed 70% for training, 20% for validation and 10% for testing. The training strategy included a batch size of 6, learning momentum of 0.937, weight decay of 0.0005, learning rate (LR) of 0.01, width and height of 640 for 1000 epochs. Training took approximately 8 hours on an RTX 3000 GPU.

Real-world images were collected from inside five construction sites: the three used for the MEP-SEG dataset and two new projects: a residential building for Olympic athletes and a new corporate headquarters. Two distinct devices were used for data collection: a smartphone and the Microsoft HoloLens 2 AR glasses. A total of 217 images were acquired and manually labeled. Then, two datasets of varying dimensions are created: (1) the first small (S) dataset included 45 images for training and 19 images for validation. (2) The second medium (M) dataset contained 131 images for training and 43 images for validation. Both pre-trained models are fine-tuned on those real-world datasets with the same strategy, adding a patience of 50 for early stopping. The two models obtained their lowest validation loss in around 200 epochs, for a training time of 20 minutes. For testing, 40 additional images are selected. Table 2 summarizes the test results.

Table 2. Performances on COCO vs. synthetic TL on small (S) and medium (M) real datasets.

Metrics	COCO TL		Synthetic TL	
S dataset	box	mask	box	mask
Precision (%)	43	51	66	64
Recall (%)	43	34	47	46
mAP50 (%)	42	38	53	49
mAP50_95 (%)	26	19	37	30
M dataset	box	mask	box	mask
Precision (%)	52	52	69	63
Recall (%)	45	42	40	38
mAP50 (%)	43	41	47	43
mAP50_95 (%)	29	24	30	24

The performance metrics used on both the predicted boxes and masks are precision, recall, mean average precision at an intersection-over-union (IoU) threshold of 0.5 (mAP50), and mean average precision at IoU thresholds ranging from 0.5 to 0.95 (mAP50_95). As we can see on the S dataset, the pre-trained model on our synthetic data outperforms the one pre-trained on COCO in every metric.

Additionally, the gap between box and mask on precision and recall is lower using the synthetic TL method, suggesting that the model better detects the edges of objects. Looking at the results on the M dataset, we observe a less significant difference between the two methods, indicating that the more real data available, the less relevant is using a synthetic dataset for a pre-trained model. In conclusion, using a synthetic pre-trained model showed promising results for transferring knowledge onto a small real dataset.

3.2.2 Evaluation of augmented datasets for MEP object detection

For the second evaluation, we constructed two sets of training datasets, one comprising 142 real images and the other 500 synthetic images. Secondly, we prepared a real test dataset containing 50 real images. This time, we focused on five classes: duct (encompassing circular and rectangular ducts), pipe, radiant panels, fan coil, and diffuser. The real training images were carefully selected from the MEP-SEG dataset to ensure they were the most representative. Thirdly, to explore the effectiveness of augmenting synthetic data with a small amount of real data, we prepared two additional training datasets:

- A mixed training dataset with 550 images, containing 90% synthetic images and 10% real images.
- A mixed training dataset with 600 images, containing 80% synthetic images and 20% real images.

In each training dataset, 80% of the images were used for training, and the remaining 20% were used for validation. The YOLOv8 model pretrained on the COCO image dataset served as the foundation and the same training strategy as in the previous evaluation was employed. The evaluation on the 50 real test images of the precision metric across all training datasets is provided in Table 3.

Table 3. Performance of the YOLOv8 model trained on different datasets.

Training dataset	Precision (%)	
	box	mask
Real	77	75
Synthetic	30	29
Synthetic +10% real	71	69
Synthetic +20% real	80	79

The purely synthetic dataset exhibited inferior performance compared to the same dataset enhanced with 10% real images, which itself performed less effectively than that augmented with 20% real images. To achieve performance that is comparable or even superior to that of the real image dataset, it appears that at least 20% of real images are needed. This indicates that augmenting a dataset of real images with synthetic images holds promise for effectiveness and efficiency.

3.3 Preliminary validation of effectiveness.

This section presents a preliminary work towards the automation of progress monitoring in the construction industry using the Hololens 2 AR glasses. The work involves the use of an YOLOv8 model, pre-trained on our MEP-SEG dataset and fine-tuned on a real dataset to accurately detect and segment diverse MEP objects within images. The NEXT-BIM application serves as a foundation for this work, which already enables the visualization of BIM models using AR glasses as well as a semi-automatic alignment of the user view. Post-alignment, the BIM view is superimposed onto the real-world view in real-time, enabling a seamless comparison between the two. A prototype tool for visual progress monitoring has been developed on top of NEXT-BIM's application. The tool is capable of assessing the MEP work progress in a real on-site scenario.

Figure 4 provides an example of the method used to assess MEP work progress.

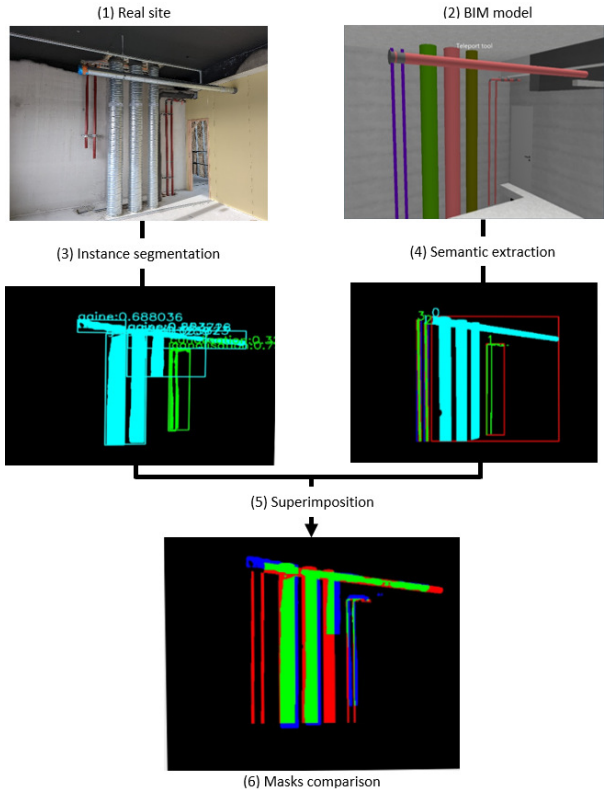

Figure 4. Captured versus BIM comparison.

The process starts with the capture of the real-world scene (1) using the integrated webcam. Subsequently, the corresponding view within the BIM model (2) is extracted based on the webcam sensor's coordinates and orientation. The captured on-site image is run through our instance segmentation model (3), resulting in a prediction

mask that categorizes and labels each pixel with a unique color corresponding to its respective class. For instance, ventilation ducts are represented in cyan, while piping is depicted in green. In parallel, semantic extraction is applied to the BIM model image (4) to generate the ground truth mask. This involves transforming each material component within the view frustum to its corresponding class, followed by color filtering and shape estimation. The two generated segmentation masks, namely the prediction mask and the ground truth mask, are then superimposed (5), with adjustments made to ensure alignment. Finally, a comparison of the superimposed masks yields the resulting discrepancy mask (6), where red represents the ground truth, blue represents the prediction, and green represents the overlapping pixels.

To ensure the robustness of our AR-based progress monitoring system, a rigorous validation methodology will be employed. Success criteria for detection, segmentation, alignment, and processing speed will be defined first. The next step involves using the Hololens glasses to collect diverse pairs of real-world images and ground truth segmentation masks, deliberately targeting challenging edge cases. Iterative evaluation metrics will be utilized to pinpoint weaknesses, which will guide the refinement of our model and data. The final stage involves designing a real-life progress assessment scenario, where the efficiency and user experience of our method will be compared to traditional alternatives.

This innovative approach, enabled by AR technology, will empowers inspectors to visualize and interact with the BIM model in real time, enhancing their understanding of the physical environment and enlighten informed decision-making. This automatic segmentation makes it easier to assess the presence or absence of MEP objects, calculate the possible difference between the as-planned and as-built, and facilitate ICPM in the end.

4 Conclusion and future work

We presented a promising solution towards the automation of indoor construction progress monitoring (ICPM) using synthetic data and deep learning (DL). We proposed a procedural building information modeling (BIM) based synthetic image generation approach to address the data scarcity of real-world data, particularly for indoor construction environments. Our experimental study showcases the successful implementation of the proposed approach in three real-world construction projects, and the preliminary validation of effectiveness using augmented reality (AR) glasses further demonstrates the potential practical applications of the developed tool.

However, our current method has limitations related to the differences between the real and synthetic worlds. Building scenes represented by BIM models are often cleaner or more well organized than real-world scenes, where random objects can be found, and the locations of movable objects can be arbitrary. Also, it relies heavily on the quality of the BIM models, which frequently contain labeling errors, geometric clashes, or an absence of object geometry.

Future work will address these limitations to make the captured images more scene-realistic. Furthermore, we now focus on exploiting instance segmentation models for ICPM. We investigate comparison methods to compute more information between as-planned and as-built segmentation masks. Additionally, to improve the model performance, we consider running it through a live video feed instead of a still image and adding temporal and tracking methods to reduce the imprecision of the model.

5 Acknowledgements

The research leading to these results is part of a thesis co-supervised between École Centrale Lyon and SPIE Building Solutions. NEXT-BIM is thanked for its support in implementing the solution in their working environment and the use of their development platform.

References

[1] Teizer and Jochen. Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure construction sites. *Advanced Engineering Informatics*, 29(2):225–238, 2015. doi:10.1016/j.aei.2015.03.006.

[2] Koo B. and Fischer M. Feasibility Study of 4D CAD in Commercial Construction. *Journal of Construction Engineering and Management*, 126(4):251–260, 2000. doi:10.1061/(ASCE)0733-9364(2000)126:4(251).

[3] Tang S., Roberts D., and Golparvar-Fard M. Human-object interaction recognition for automatic construction site safety inspection. *Automation in Construction*, 120:103356, 2020. doi:10.1016/j.autcon.2020.103356.

[4] Asadi K., Ramshankar H., Noghabaei M., and Han K. Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction. *J. Comput. Civ. Eng.*, 33 (5):04019031, 2019. doi:10.1061/(ASCE)CP.1943-5487.0000847.

[5] Fathi H., Dai F., and Lourakis M. Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges. *Advanced Engineering Informatics*, 29 (2):149–161, 2015. doi:10.1016/j.aei.2015.01.012.

[6] Wang Boyu, Wang Q., Cheng J.C.P., Song C., and Yin C. Vision-assisted BIM reconstruction from 3D LiDAR point clouds for MEP scenes. *Automation in Construction*, 133:103997, 2022. doi:10.1016/j.autcon.2021.103997.

[7] Reja V.K., Varghese K., and Ha Q.P. Computer vision-based construction progress monitoring. *Automation in Construction*, 138:104245, 2022. doi:10.1016/j.autcon.2022.104245.

[8] Pal A. and Hsieh S. Deep-learning-based visual data analytics for smart construction management. *Automation in Construction*, 131:103892, 2021. doi:10.1016/j.autcon.2021.103892.

[9] Wei W., Lu Y., Zhong T., Li P., and Liu B. Integrated vision-based automated progress monitoring of indoor construction using mask region-based convolutional neural networks and BIM. *Automation in Construction*, 140:104327, 2022. doi:10.1016/j.autcon.2022.104327.

[10] Deng J., Dong W., Socher R., Li L., Li K., and Fei-Fei L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. doi:10.1109/CVPR.2009.5206848.

[11] Armeni I., Sener O., Zamir A.R., Jiang H., Brilakis I., Fischer M., and Savarese S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, Las Vegas, NV, USA, 2016. IEEE. doi:10.1109/CVPR.2016.170.

[12] Lin T., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., and Zitnick C.L. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. doi:10.1007/978-3-319-10602-1_48.

[13] Zhou B., Zhao H., Puig X., Fidler S., Barriuso A., and Torralba A. Scene Parsing through ADE20K Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017. doi:10.1109/CVPR.2017.544.

[14] 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*, 122:103482, 2021. doi:10.1016/j.autcon.2020.103482.

[15] Yan X., Zhang H., Wu Y., Lin C., and Liu S. Construction instance segmentation (CIS) dataset for deep learning-based computer vision. *Automation in Construction*, 156:105083, 2023. doi:10.1016/j.autcon.2023.105083.

[16] Soltani M.M., Zhu Z., and Hammad A. Automated annotation for visual recognition of construction resources using synthetic images. *Automation in Construction*, 62:14–23, 2016. doi:10.1016/j.autcon.2015.10.002.

[17] Northcutt C.G., Athalye A., and Mueller J. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. doi:10.48550/arXiv.2103.14749.

[18] Handa A., Patraucean V., Badrinarayanan V., Stent S., and Cipolla R. Understanding Real-World Indoor Scenes with Synthetic Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4077–4085, 2016. doi:10.1109/CVPR.2016.442.

[19] Ha I., Kim H., Park S., and Kim H. Image retrieval using BIM and features from pretrained VGG network for indoor localization. *Building and Environment*, 140:23–31, 2018. doi:10.1016/j.buildenv.2018.05.026.

[20] 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*, 150:245–258, 2019. doi:10.1016/j.isprsjprs.2019.02.020.

[21] Lee J.G., Hwang J., Chi S., and Seo J. Synthetic Image Dataset Development for Vision-Based Construction Equipment Detection. *Journal of Computing in Civil Engineering*, 36(5):04022020, 2022. doi:10.1061/(ASCE)CP.1943-5487.0001035.

[22] Acharya D., Singha Roy S., Khoshelham K., and Winter S. A Recurrent Deep Network for Estimating the Pose of Real Indoor Images from Synthetic Image Sequences. *Sensors*, 20(19):5492, 2020. doi:10.3390/s20195492.

[23] Sacks R., Eastman C., Lee G., and Teicholz P. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*. Wiley, 2018. doi:10.1002/9781119287568.

[24] Hong Y., Park S., Kim H., and Kim H. Synthetic data generation using building information models. *Automation in Construction*, 130:103871, 2021. doi:10.1016/j.autcon.2021.103871.

[25] Ying H., Sacks R., and Degani A. Synthetic image data generation using BIM and computer graphics for building scene understanding. *Automation in Construction*, 154:105016, 2023. doi:10.1016/j.autcon.2023.105016.

[26] Redmon J., Divvala S., Girshick R., and Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi:10.1109/CVPR.2016.91.