

Automating Dynamic Scaffolding Construction Progress Tracking Using Deep Learning-based Semantic Image Segmentation

Natthapol Saovana¹, Chavanont Khosakitchalert^{2,*}

¹ Department of Construction Design and Management, Faculty of Industrial Technology and Management, King Mongkut's University of Technology North Bangkok, Thailand

² Architectural Intelligence (A.I.) Research Group, Department of Architecture, Faculty of Architecture, Chulalongkorn University, Thailand

* Corresponding e-mail: Chavanont.K@chula.ac.th

Received 2025-08-12; Revised 2025-11-14; Accepted 2026-01-05

ABSTRACT

Scaffolding plays a critical role in construction by providing safe access to elevated work areas, yet it is often overlooked in progress monitoring and omitted from reports and building information models (BIM) due to its temporary and dynamic nature. This study presents a deep learning-based semantic segmentation system for tracking scaffolding installation progress, distinguishing between scaffolding with and without safety sheets. A custom dataset of annotated site images was used to train the model, and performance was evaluated on both validation and test sets. The system achieved real-time processing speeds of 33.81 Frame Per Second (FPS) (validation) and 31.28 FPS (test), with mean Average Precision scores of 0.496 and 0.480, respectively. Class-specific results showed consistently higher accuracy for scaffolding with safety sheets, with peak Intersection over Union (IoU) values exceeding 93% in a case study time point. Two multi-time point construction case studies demonstrated the system's robustness across varying site conditions. An automatic mask modification algorithm was applied to address missed detections, improving IoU by up to 3.10% in challenging scenarios. The calculated progress, based on segmented masks and predefined building boundaries, was compared with groundtruth measurements, confirming the system's capability for quantitative progress tracking. Results indicate that scaffolding with safety sheets is more reliably detected, while detection of scaffolding without sheets remains more challenging. The proposed method offers a practical tool for reducing inspection workload, improving safety compliance, and enabling more comprehensive progress tracking in construction projects, particularly in scenarios where scaffolding installation is a major operational activity.

Keywords: construction progress tracking, scaffolding, deep learning, image segmentation, computer vision

INTRODUCTION

Scaffolding is a temporary structure used to support construction at height, consisting mainly of steel rods and toe boards assembled into frames designed to withstand their own weight and that of workers as axial forces (Błazik-Borowa et al., 2023). Safety guidelines recommend covering scaffolding with safety sheets to prevent falls and contain dust and debris (Choo et al., 2023). Despite its importance to safety and productivity, scaffolding is often overlooked because it does not directly contribute to project financial statements (Kim et al., 2018b). Large sites may require significant amounts of scaffolding, leading to indirect costs of up to 15% of total construction costs (Hou et al., 2017) or even more in scaffolding-intensive projects (Yin & Caldas, 2022). Its dynamic nature, changing within days or weeks, and reliance on manual tracking methods such as visual inspections contribute to its omission from reports (Khosakitchalert & Yabuki, 2023) and building information modelling (BIM) model (Khosakitchalert & Seghier, 2025; Sacks et al., 2018) due to infrequent updates by responsible parties (Kim et al., 2018a).

Advances in artificial intelligence (AI) have enabled applications across research and practice, with deep learning emerging as a leading technique capable of learning from large datasets through multi-layer neuron-like networks (Goodfellow et al., 2016; Tochaiwat & Seniwong, 2025). In monitoring and inspection, deep learning can process images or videos to identify meaningful pixels within input data. Prior works include crack detection on concrete using an encoder–decoder FCN (Dung & Anh, 2019), bridge component segmentation from point clouds via PointNet (Kim & Kim, 2020; Garcia-Garcia et al., 2016), worker detection with SOC-YOLO in digital images (Park et al., 2023), and progress tracking of indoor construction using Mask R-CNN (Ekanayake et al., 2024; He et al., 2017). Although deep learning shows potential for scaffolding progress monitoring, a literature review found no related studies.

This study develops a deep learning system that quantifies scaffolding installation progress as a percentage, unlike current practice, which relies on manual inspection and records only whether work is finished or unfinished. This continuous

percentage enables users to report partial completion and track changes over time while distinguishing between scaffolding with and without safety sheets. The originality of this study lies in its integration of deep learning-based semantic segmentation with quantitative progress estimation, allowing automated, class-level tracking of scaffolding installation, a capability not addressed in previous research. A custom segmentation model was trained on site images and evaluated through two multi-time point case studies. An automatic mask modification algorithm was also assessed to fill voids and remove small outliers. The system's calculated progress is then compared with ground-truth values from annotated building and scaffolding boundaries. The primary beneficiaries are contractors, who can use these percentages to track completion and manage planning and costs, thereby reducing manual inspection workload. Beyond immediate use in progress tracking, the approach provides a foundation for researchers to extend scaffolding detection to other aspects, such as safety compliance and risk monitoring on site.

LITERATURE REVIEW

Construction Progress Monitoring Techniques and the Role of Scaffolding

Construction progress tracking has been widely studied, with many approaches leveraging computer vision, photogrammetry, and BIM-based automation to improve accuracy and efficiency. Traditional manual inspections are increasingly replaced by automated systems that provide real-time, objective data collection and analysis, supported by advances in digital imagery and deep learning (Paoletti et al., 2019). In civil engineering, deep learning models can detect and segment construction elements from images, enabling continuous on-site monitoring (Hamledari et al., 2017). Daily site image collection, already common in many projects, can be repurposed for such monitoring without additional workload (Guan et al., 2020).

Applications of deep learning in construction include segmenting unhealthy bridge components with InceptionV3 (Hüthwohl & Brilakis, 2018), detecting bridge elements (Saovana et al., 2021) with You Only Look At Coefficients (YOLACT) (Bolya et al., 2019), worker detection (Park et al., 2023) with SOC-YOLO (Redmon et al., 2015), and tracking indoor construction progress using Mask R-CNN (Ekanayake et al., 2024). Recent work has also integrated DinoV2 (Oquab et al., 2023) and the Segment Anything Model (Kirillov et al., 2023) for pipe and duct segmentation (Wang et al., 2024), and applied OpDenseNet201 for high-accuracy pavement crack detection (Matarneh et al., 2024).

Despite these advancements, scaffolding is often treated as visual occlusion rather than a component of progress. In photogrammetry and point cloud workflows, it is categorized as static clutter and removed during data filtering (Kavaliauskas et al., 2022). In 4D Augmented Reality-based monitoring, scaffolding is considered an obstruction to image capture and as-built/as-planned comparisons (Mani et al., 2009). Theoretical frameworks on automated progress monitoring also identify it as a complicating factor due to its widespread, immovable nature (Qureshi et al., 2022). This consistent treatment reflects a broader tendency to view scaffolding as a nuisance rather than an element worthy of dedicated tracking.

Scaffolding as an Outlier in Computer Vision Applications

Scaffolding-induced occlusion is a common challenge in computer vision (CV) and deep learning-based approaches. Several studies have developed methods to visually remove scaffolding from images to improve object detection and scene interpretation. Techniques such as semantic segmentation and image inpainting have been used to detect and erase scaffolding, restoring underlying visual information for structural tracking (Ding et al., 2025). In automated façade construction monitoring, scaffolding similarly obstructs brick detection and counting on building exteriors (Hui et al., 2014), leading to its removal to minimize

interference and reinforcing its exclusion from progress tracking.

Even in advanced modular construction monitoring, such as Prefabricated Prefinished Volumetric Construction (PPVC), scaffolding is regarded as an undesirable obstruction (Chua & Cheah, 2024). It occludes critical features like windows needed for installation assessment, prompting the use of inference-based algorithms to rectify occlusions rather than treating scaffolding as a progress indicator (Chua & Cheah, 2024). Likewise, real-time monitoring systems often filter or compensate for scaffolding as a temporary element during data processing (Omara et al., 2018).

Construction Scaffolding and its Overlooked Role in Progress Tracking

Scaffolding is temporary yet essential, providing safe access to elevated workspaces and supporting labor and materials. Its dynamic nature requires repeated assembly, adjustment, and dismantling throughout a project (Bec et al., 2023). Despite its operational importance and role in indicating active work zones, scaffolding is often omitted from BIM models and progress reports (Chern et al., 2024; Kim et al., 2018b).

This exclusion stems partly from the limitations of traditional inspections, which rely on manual assessments prone to subjectivity, inefficiency, and reporting inaccuracies (Golparvar-Fard et al., 2015). While some research has acknowledged scaffolding as a rough progress indicator (Xu et al., 2018), most treat it as visual noise to be removed prior to analysis. Consequently, a significant gap remains in developing monitoring methods that recognize scaffolding as a meaningful component on site rather than interference.

Deep Learning Applications in Scaffolding Detection

Recent studies have applied deep learning to scaffolding detection, focusing mainly on safety compliance and component-level inspection. Such models can automate inspection tasks and

improve on-site safety. Kim et al. (2022) used a robotic platform to collect point cloud data of scaffolding without safety sheets, applying RandLA-Net (Hu et al., 2019) for segmentation and enhancing accuracy through network modifications. Lin et al. (2023) employed Mask R-CNN to detect scaffolding panels, though dataset limitations reduced precision. Dzung et al. (2024) developed the Scaffolding Assembly Deficiency Detection System (SADDS) using YOLOv5 and augmented reality to identify missing components, achieving high detection accuracy but without assessing overall scaffolding progress.

While these studies demonstrate the potential of deep learning for scaffolding-related tasks, few address installation progress as an indicator of site development. Chern et al. (2024) monitored workers on scaffolding using Shallow Cascaded Buffered Intersection over Union (Shallow C-BIoU) (Yang et al., 2023), but detection performance was hindered by structural complexity. These findings reveal an opportunity to reframe scaffolding as a progress marker rather than an obstruction, enhancing monitoring in dynamic construction environments.

Summary and Research Direction

While deep learning has advanced the monitoring of various construction elements, its application to scaffolding remains limited, with most studies treating it as an occlusion to be removed rather than a subject of analysis. Yet, scaffolding plays a critical role in construction logistics, safety, and task sequencing, offering potential as a proxy for progress tracking.

This research addresses this gap by proposing scaffolding as a dynamic construction element whose installation progress can be quantitatively measured using deep learning-based semantic segmentation. By reframing scaffolding from a visual obstruction to a measurable indicator, this study supports a more comprehensive and representative approach to construction progress monitoring.

METHODOLOGY

Overview

This section presents the methodology for automation of scaffolding installation and safety sheet application tracking. The system employs deep learning-based segmentation to differentiate between *Scaffolding with Safety Sheets* and *Scaffolding without Safety Sheets*, then calculates progress based on building area and completed work. The workflow comprises eight stages: (1) input data, (2) scaffolding segmentation, (3) segmentation modification, (4) working area generation, (5) class masks overlay, (6) area calculation, (7) progress calculation, and (8) visualization. The research methodology of this study is shown in Figure 1.

Input Data

The primary input to the proposed system is an original construction-site image in which scaffolding is clearly visible. Each image must represent a real construction scenario and is used to demonstrate how the proposed methodology operates under practical site conditions.

Scaffolding Segmentation

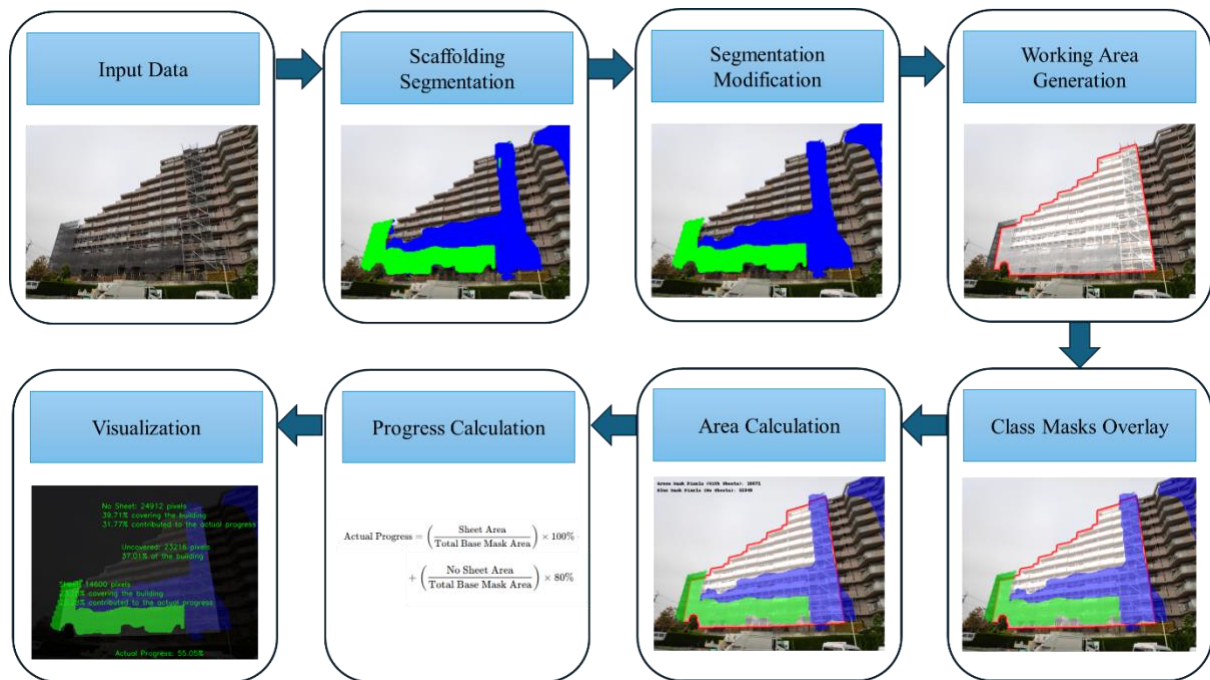
The output of this process is an image segmented into two classes: *Scaffolding with Safety Sheets* and *Scaffolding without Safety Sheets*. To achieve such segmentation, a deep learning network must first be trained through three main steps: (1) dataset creation, (2) network architecture development, and (3) segmentation performance evaluation.

Dataset Creation

A dataset of real construction site images was compiled to train a deep learning model for segmenting scaffolding with and without safety sheets. Images were captured under varied real-world conditions to enhance model generalization, including different lighting, weather, and camera angles. To reflect diverse scaffolding practices and safety standards, data

Figure 1

Overview of the Proposed Methodology

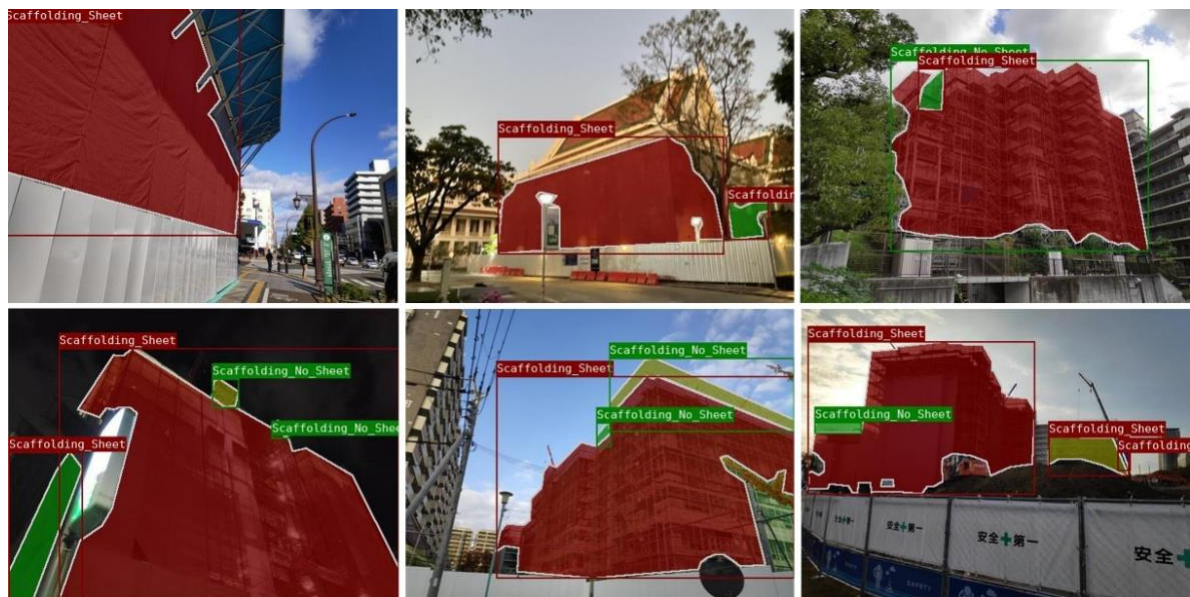


were collected from multiple cities of Japan and Thailand, using digital cameras, smartphones, and computer tablets. Conditions ranged from sunrise to dusk and from sunny to cloudy, providing a broad spectrum of contrasts and visibility challenges. After quality control and selection, the dataset contained 2,500 original images. Example images are shown in Figure 2.

The dataset was split into 1,860 training images, 465 validation images, and 175 test images, with each subset used exclusively for its designated phase to prevent data leakage. Images were then resized to 512×324 pixels to reduce features, decrease noise, and accelerate training (Gupta & Gupta, 2019). Annotation was performed in LabelMe (Russell et al., 2008) with two classes: *Scaffolding with Safety Sheets* (Scaffolding_Sheets) and *Scaffolding without Safety Sheets* (Scaffolding_No_Sheets). This two-class division was selected because these stages are both visually distinct and operationally sequential, representing the major transition observable during scaffolding installation. *Scaffolding without safety sheets* represents the steel frame erection stage, while *scaffolding with*

safety sheets represents the fully completed installation. Further subdivision of the steel erection process was considered but not adopted, as intermediate phases lack consistent visual indicators for accurate labeling and would introduce annotation ambiguity.

Manual polygonal segmentation excluded occlusions such as trees or fences to reduce labeling noise (Saovana et al., 2020). To increase dataset diversity, augmentation produced four additional variants per image through horizontal flipping, rotation ($\pm 15^\circ$), and brightness/contrast adjustments ($\pm 20\%$). Figure 3 shows labeled examples from Figure 2, where mask colors indicate object areas, and labels above boxes specify the class. Scaffolding was detected as a whole rather than by individual components, as site supervisors noted that tracking every piece is overly labor-intensive; assessing overall area coverage is sufficient for progress evaluation. The dataset used in this study is available at the following link: https://drive.google.com/drive/u/1/folders/1rxS08jGLmVAKuwybR3Z_texHQAlpFGk.

Figure 2*Example Images From the Dataset of This Study***Figure 3***Labels of the Example Images in Figure 2***Deep Learning Architecture**

The proposed deep learning architecture consists of five sequential stages designed to perform efficient instance segmentation as shown in Figure 4. An input image of a complex construction scene is first processed by a backbone network to extract high-level feature maps capturing structural and contextual information. These maps are fed into two parallel

branches: class prediction, which detects object instances and assigns categorical labels (e.g., scaffolding with or without safety sheets), and semantic segmentation, which generates class-agnostic prototype masks representing generalized spatial patterns. In the mask coefficient stage, each detected instance receives unique coefficients to combine the prototype masks into instance-specific masks.

The final stage applies these coefficients to produce refined, pixel-level segmentations. The training was conducted for 300 epochs on a desktop equipped with an NVIDIA GeForce RTX 3070 GPU, 32GB of RAM, and an Intel Core i7-12700 CPU.

Segmentation Performance Evaluation

The segmentation results were primarily evaluated using the Intersection over Union (IoU) at mAP@50, which served as the main indicator for assessing subsequent processes. In addition, several quantitative metrics, including mean Average Precision (mAP), Recall, F1 Score, and Frames Per Second (FPS), were employed to provide a comprehensive assessment of the model's accuracy and real-time applicability.

- *Intersection over Union (IoU)*

IoU is a standard evaluation metric in image segmentation that measures the overlap between the predicted mask and the ground-truth mask. It is calculated as the ratio of the area of intersection to the area of union between the two masks. A higher IoU value indicates a closer match between the predicted segmentation and the ground truth, reflecting better model accuracy.

- *Mean Average Precision (mAP)*

mAP is a widely used segmentation metric, with higher values indicating better accuracy.

Following COCO detection standards (Lin et al., 2014), mAP@[0.5:0.95] was computed across 10 IoU thresholds (0.5–0.95, step 0.05), offering a balanced assessment from lenient to strict matching. For each threshold, Average Precision (AP) was calculated as the area under the Precision–Recall curve, and mAP was obtained by averaging AP across thresholds for each class (Scaffolding with Safety Sheets and Scaffolding without Safety Sheets).

- *Recall*

Recall measures the proportion of true positives among all actual positives, indicating the model's ability to detect relevant scaffolding. A high score suggests effective detection, even with occasional false positives.

- *F1 Score*

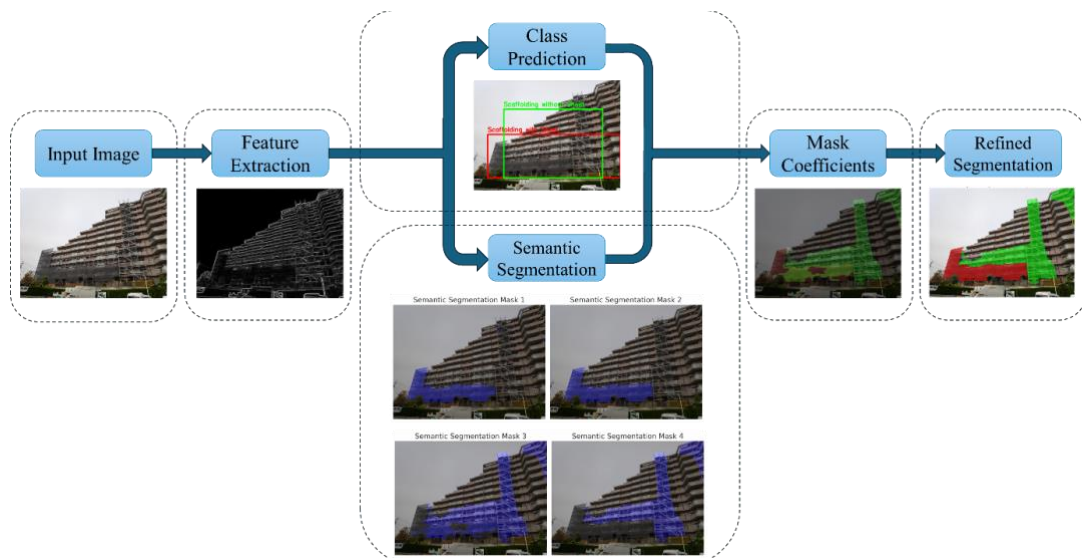
The F1 Score is the harmonic mean of Precision and Recall, balancing the model's ability to be both accurate and comprehensive. In the COCO framework, it was computed manually using Eq. (1):

$$F1 = \sum_{i=1}^n \left(2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \right) \quad (1)$$

where $n = 10$ IoU thresholds, $Precision_i$ is the proportion of true positives among positive predictions, and $Recall_i$ is the proportion of true positives among all actual positives at each threshold.

Figure 4

Deep Learning Architecture in This Study



▪ *Frame Per Second (FPS)*

FPS measures real-time performance by calculating how many frames the system can process per second. Approximately 32 FPS is considered efficient for real-time monitoring (Peng et al., 2020).

Segmentation Modification

Deep learning-based segmentation alone may produce errors that lead to low-accuracy masks. Therefore, an automatic mask modification algorithm is introduced to enhance the segmentation results. This algorithm improves segmentation quality by filling voids within scaffolding regions and removing small, irrelevant outliers. It uses functions from NumPy (Harris et al., 2020), OpenCV (Bradski, 2000), SciPy (Virtanen et al., 2020), and Scikit-image (van der Walt et al., 2014). This postprocessing step enhances progress monitoring accuracy, as scaffolding typically forms continuous, compact regions without internal gaps. While small voids or scattered masks often indicate segmentation

errors, the algorithm is optional, as such patterns may also result from torn safety sheets or scaffolding under construction. The pseudocode is provided in **Algorithm 1**.

The `binary_fill_holes` function from SciPy (Virtanen et al., 2020) is a morphological image processing method that fills voids in a binary image (pixel values of 0 or 1). A hole is defined as a background region (0) completely surrounded by foreground pixels (1). The function identifies such enclosed regions and converts them to foreground, closing gaps in the binary mask. An example of this operation is illustrated in Figure 5.

The `remove_small_objects` function from scikit-image removes connected components smaller than a user-defined size threshold F in Algorithm 1. It identifies all connected foreground regions and filters out those below F , treating them as noise. F can be set by minimum area or as a percentage of total image area; in this study, it was set to 10% to remove small, incomplete segmentation masks. An example is shown in Figure 6.

Algorithm 1

Mask Modification Algorithm Proposed by This Study

Input Masks from segmentation $M = \{M_i \mid i = 1, 2, 3, \dots, n\}$; Minimum pixel or percentage of masks not to be filtered out F

Output Modified masks

1. Load M
2. Import Numpy, OpenCV, Scipy, and Skimage
3. For M_i in M :
4. Change masks to binary
5. Utilize the `binary_fill_holes` function from Scipy
6. Utilize the `remove_small_objects` function from Skimage with F
7. Save the modify mask M_i

Figure 5

Example of a Mask Before and After Processing With the Binary_Fill_Holes Function



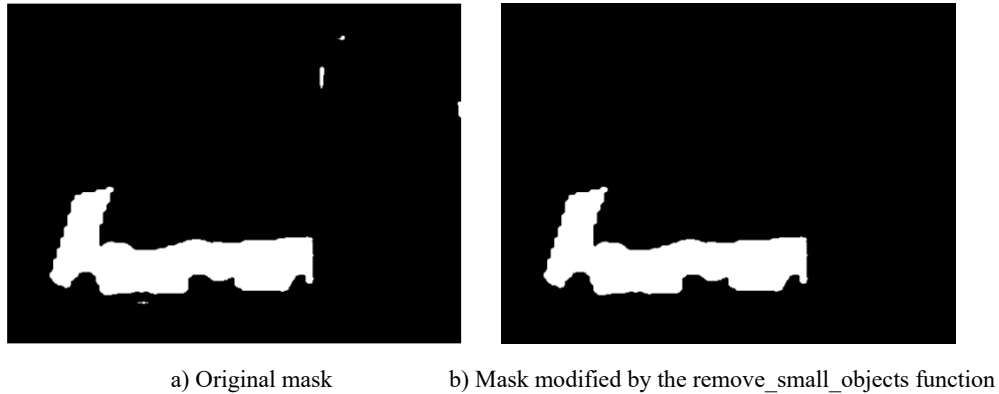
a) Original mask



b) Mask modified by the `binary_fill_holes` function

Figure 6

Example of a Mask Before and After Processing With the Remove_Small_Objects Function



Working Area Generation

The working area, or base mask, is generated from building boundary data in the JSON file, representing the total area to be covered by scaffolding and ensuring that only valid regions are included in progress calculations. For stationary image capture (e.g., CCTV or tripod), a single boundary can theoretically be reused across different time points. However, when the scene changes, such as due to camera repositioning, object movement, or environmental variation, a new base mask must be defined to maintain alignment and accuracy. Figure 7 shows an example from one of the testing samples in this study, with the red line marking the target scaffolding coverage area.

Class Masks Overlay

In this step, segmentation masks for both scaffolding classes are overlaid onto the base mask to identify installed scaffolding, either with or without safety sheets, within the defined working area. This ensures that only regions inside the building boundary are analyzed, forming the basis for area and progress calculations. An example of this overlay is shown in Figure 8, where the green region represents *Scaffolding with Safety Sheets* and the blue region represents *Scaffolding without Safety Sheets*.

Area Calculation

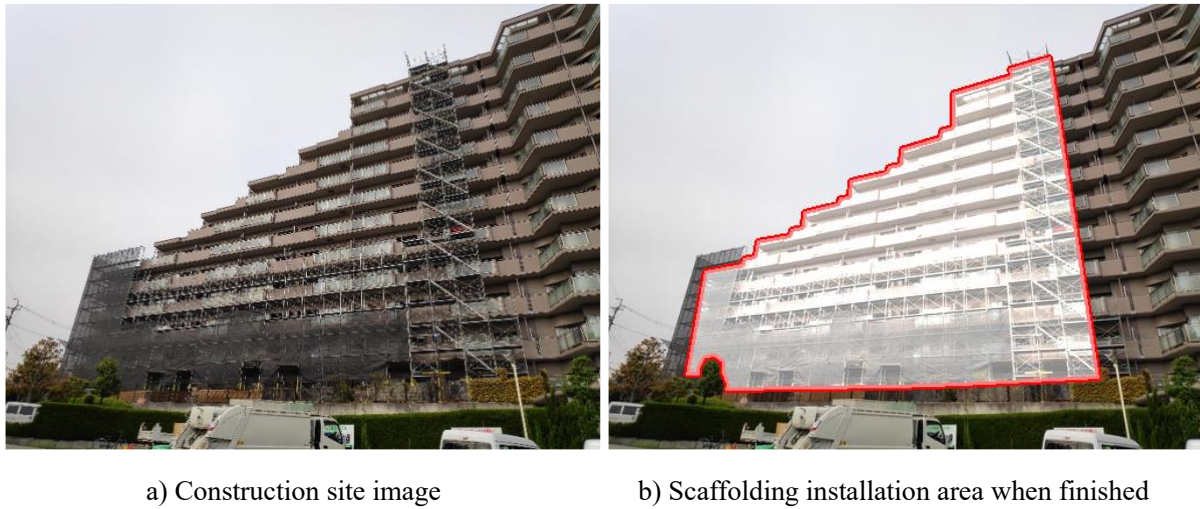
The system calculates the pixel area of each class within the base mask, with all measurements expressed in pixels.

Progress Calculation

Interviews with site supervisors revealed that scaffolding installation progress is typically recorded only as 0% or 100% upon full completion. For more detailed tracking, this study introduces a weighting factor W_{NS} to represent the completion ratio of scaffolding without safety sheets. Based on practical input, W_{NS} was assigned a value of 0.8 (or 80%) in this case study, corresponding to the stage where the steel structure is fully installed but safety sheets have not yet been attached. Progress is calculated from the segmented masks and the base mask using Eq. (2):

$$\text{Actual Progress} = \left(\frac{SS \cap BM}{BM} \right) \times 100\% + \left(\frac{NS \cap BM}{BM} \right) \times W_{NS} \times 100\% \quad (2)$$

where $SS \cap BM$ is the intersection area between the Scaffolding with Safety Sheets class and the base mask; $NS \cap BM$ is the intersection area between the Scaffolding without Safety Sheets class and the base mask; BM represents the total base mask area; and W_{NS} is the weighting coefficient for the partially completed scaffolding class. The value of W_{NS} can be adjusted for other projects according to site-specific assessment criteria or supervisor input.

Figure 7*Example of Base Mask Creation***Figure 8***Example of Segmentation Mask Overlay Showing the Identified Scaffolding Regions Within the Defined Building Boundary*

Note. The green region represents Scaffolding with Safety Sheets, and the blue region represents Scaffolding without Safety Sheets.

Visualization

The system outputs an annotated image of the original input, categorizing the working area as No Sheet, Sheet, or Uncovered. An example of the output from the proposed system is shown in

Figure 9. Progress is displayed both overall and by class contribution, integrating visual overlays with numerical results to support precise tracking. This combined representation enhances usability for site supervisors and project managers, enabling informed decision-making based on actual installation progress.

Figure 9

Example of Result Visualization From the Proposed System



RESULTS

This section presents the evaluation of the various stages involved in the scaffolding progress tracking system developed in this study. The performance metrics for training, segmentation, mask modification, and actual progress tracking are analyzed to assess the accuracy and effectiveness of the proposed approach. Each subsection provides a detailed breakdown of the respective performance stage, offering insights into the model's capability to accurately segment scaffolding, refine segmentation masks, and track progress over time. The objective is to demonstrate that the system performs reliably across different conditions and construction phases, ensuring its suitability for practical deployment in real-world projects.

Testing Samples

Two real-world construction sites were selected to evaluate the proposed scaffolding progress tracking system under different environmental and structural conditions. Each case consisted of multiple images captured during various time points of scaffolding installation and safety sheet application. The selected samples represent distinct scenarios, one featuring a straightforward

layout with some occlusion and another involving a more complex configuration with overlapping classes. These contrasting setups allow for comprehensive validation of the system's segmentation robustness, mask modification reliability, and progress calculation accuracy under both ideal and challenging site conditions.

Sample 1

This case featured a straightforward scaffolding layout with distinct installation and safety sheets. A large tree was present to assess occlusion effects. Images were captured orthogonally to the scaffolding, following Dzung et al. (2024), who recommend perpendicular capture for optimal quality. Figure 10 and Figure 11 show images from both captured time points.

Sample 2

This case involved a more complex scaffolding configuration, with both classes present in all three time points. Images were captured from an oblique angle to assess the impact of viewing perspective on progress tracking. Figure 12 illustrates the three time points of this sample, showing variations in scaffolding installation and safety sheet coverage over time. The site served as a secondary validation of the model's segmentation and progress tracking accuracy.

Figure 10

First Time Point of Scaffolding Installation for Sample 1



Figure 11

Second Time Point of Scaffolding Installation for Sample 1



Figure 12

Three Time Points of Scaffolding Installation for Sample 2



Training Results

The change in mask loss over the training process is illustrated in Figure 13. At the beginning of training (epoch 0), the mask loss is relatively high, at approximately 0.9, which is expected since the model starts with random weights and no prior knowledge of the dataset. During the first 10 epochs, a substantial reduction in mask loss is observed, indicating rapid learning and significant improvement in segmentation performance. This steep decline demonstrates the effectiveness of the early-stage training process. Periodic spikes in mask loss occur throughout the training period, caused by learning rate adjustments, such as scheduled decay or other optimization strategies, that momentarily increase the loss before the model adapts. Despite these fluctuations, the overall trend remains downward, showing the model's capacity to recover from temporary setbacks and

continue improving. By epoch 300, the mask loss stabilizes at a low value of 0.01251, suggesting that the model has effectively converged and is consistently producing accurate segmentation predictions.

The system's performance was assessed using both validation and test datasets, with the results summarized in Table 1. During validation, the system achieved an average processing speed of 33.81 FPS, while the test dataset yielded 31.28 FPS, confirming its capability for real-time operation. For the validation dataset, the overall mean Average Precision (mAP), Recall, and F1 score were 0.496, 0.571, and 0.531, respectively. On the test dataset, the system achieved an overall mAP of 0.480, a Recall of 0.547, and an F1 score of 0.511. These results indicate consistent performance across datasets, with only minor variation between validation and test phases.

Figure 13
The Loss of the Training Over Time

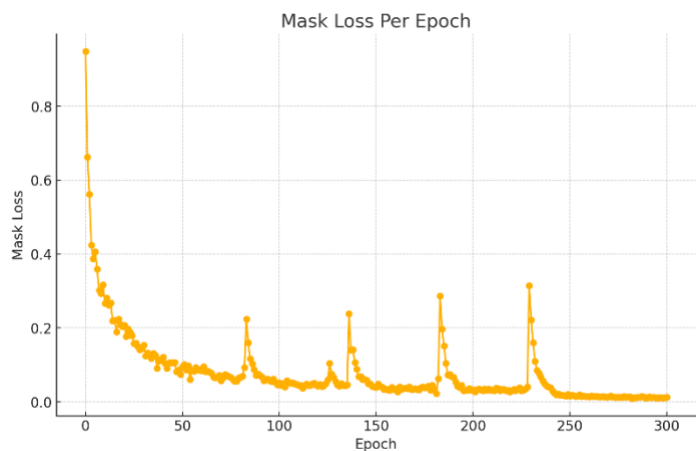


Table 1
Performance Metrics of the Scaffolding Segmentation System on Validation and Test Datasets

Dataset	FPS	Classes	mAP	Recall	F1
Validation	33.81	Overall	0.496	0.571	0.531
		Scaffolding with Safety Sheets	0.527	0.593	0.558
		Scaffolding without Safety Sheets	0.464	0.549	0.503
Test	31.28	Overall	0.480	0.547	0.511
		Scaffolding with Safety Sheets	0.552	0.610	0.580
		Scaffolding without Safety Sheets	0.407	0.483	0.442

In terms of class-specific performance, the segmentation of scaffolding with safety sheets consistently outperformed that of scaffolding without safety sheets across both datasets. During validation, the mAP, Recall, and F1 scores for scaffolding with safety sheets were 0.527, 0.593, and 0.558, respectively, while scaffolding without safety sheets achieved scores of 0.464, 0.549, and 0.503. On the test set, the segmentation of scaffolding with safety sheets yielded higher scores, with an mAP of 0.552, a Recall of 0.610, and an F1 score of 0.580. In contrast, the segmentation of scaffolding without safety sheets showed lower performance, with an mAP of 0.407, a Recall of 0.483, and an F1 score of 0.442.

These results indicate that the system performs more accurately in detecting scaffolding covered by safety sheets, while the segmentation of scaffolding without safety sheets is more challenging, leading to slightly lower performance metrics.

Segmentation Results

The segmentation results in this section evaluate the system’s ability to accurately distinguish between scaffolding with and without safety sheets across multiple case studies.

Performance is assessed using the IoU metric, enabling a detailed analysis of segmentation accuracy for each class.

Sample 1

The segmentation accuracy for each time point in Sample 1 is summarized in Table 2. Two time points were evaluated, each containing only one scaffolding class. In Time Point 1, the system segmented scaffolding without safety sheets with an IoU of 81.23%. As no scaffolding with safety sheets was present, this class was not evaluated. In Time Point 2, the system achieved an IoU of 93.22% for scaffolding with safety sheets, with no scaffolding without safety sheets present. These results demonstrate the system’s ability to accurately segment scaffolding elements, even when class presence varies between project time points.

The comparison between ground truth and segmented results for both time points is shown in Figure 14 and Figure 15. These images provide a visual representation of the system's performance, highlighting its ability to accurately delineate scaffolding from other elements in the scene. The blue mask represents the scaffolding without safety sheets, while the green mask represents the scaffolding with safety sheets.

Table 2
Segmentation Accuracy of Sample 1

Results	IoU (%)		
	Scaffolding with Safety Sheets	Scaffolding without Safety Sheets	Overall
Time Point 1	-	81.23	81.23
Time Point 2	93.22	-	93.22

Figure 14
Segmented Result of the First Time Point of Sample 1



Figure 15
Segmented Result of the Second Time Point of Sample 1



Sample 2

The segmentation accuracy for each time point in Sample 2 is summarized in Table 3. The performance was evaluated across three distinct time points, focusing on the detection of scaffolding with and without safety sheets.

In Time Point 1, the system achieved an IoU of 85.50% for scaffolding with safety sheets and 68.19% for scaffolding without safety sheets, resulting in an overall segmentation accuracy of 70.82%. As shown in Figure 16, the system effectively captured the scaffolding structures; however, some discrepancies were observed in detecting scaffolding without safety sheets.

In Time Point 2, the segmentation performance improved, with an IoU of 87.58% for scaffolding with sheets and 72.01% for scaffolding without sheets, leading to an overall accuracy of 79.33%. The segmented result in Figure 17 reflects this improvement, particularly in the more comprehensive detection of scaffolding structures, although minor misclassification was noted in the top-right corner of the image.

In Time Point 3, the system achieved a high IoU of 94.45% for scaffolding with safety sheets, reflecting its ability to accurately segment this class. However, no scaffolding without sheets was detected in this time point, resulting in an IoU of 0% for this class. Despite this, the overall segmentation accuracy for Time Point 3 remained high at 82.84%, as shown in Figure 18.

Mask Modification Evaluation

The mask modification algorithm's effectiveness was assessed by comparing IoU values and segmentation quality before and after its application. Designed to fill gaps and remove noise, the algorithm was expected to improve IoU by correcting errors in the initial segmentation, ensuring final masks more accurately represent the true scaffolding layout.

Sample 1 Time Point 1

The mask modification algorithm was applied to the first time point of Sample 1 to improve the accuracy of the initial segmentation. The segmentation accuracy before and after the modification is shown in Table 4. The original segmentation result yielded an IoU of 81.23% for the scaffolding without safety sheets class. After applying the mask modification algorithm, the IoU increased to 84.33%, resulting in a +3.10% improvement. This improvement highlights the algorithm's ability to refine the segmentation by filling in missing regions. Figure 19 illustrates the comparison between the groundtruth, the original segmentation result, and the modified segmentation result. The visual and numerical enhancements demonstrate the effectiveness of the mask modification process in improving segmentation quality, particularly in challenging areas where the initial segmentation may have failed to capture the full extent of the scaffolding.

Table 3
Segmentation Accuracy of Sample 2

Results	IoU (%)		
	<i>Scaffolding with Safety Sheets</i>	<i>Scaffolding without Safety Sheets</i>	<i>Overall</i>
Time Point 1	85.50	68.19	70.82
Time Point 2	87.58	72.01	79.33
Time Point 3	94.45	0.00	82.84

Figure 16
Segmented Result of the First Time Point of Sample 2

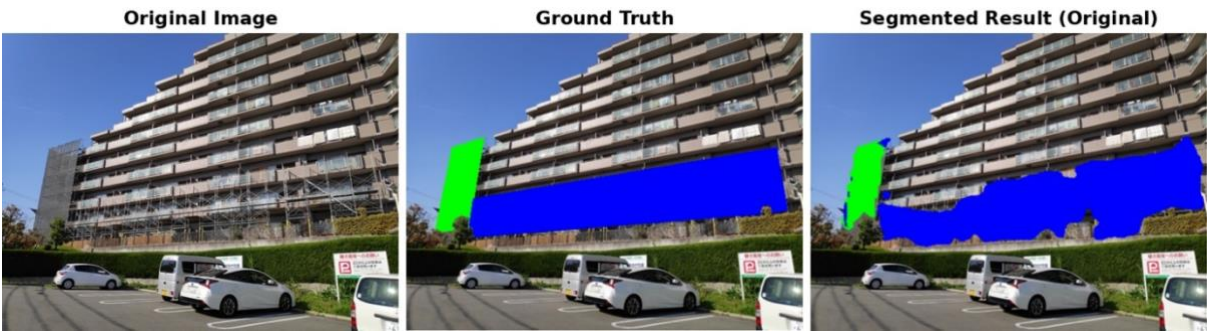


Figure 17
Segmented Result of the Second Time Point of Sample 2



Figure 18
Segmented Result of the Third Time Point of Sample 2

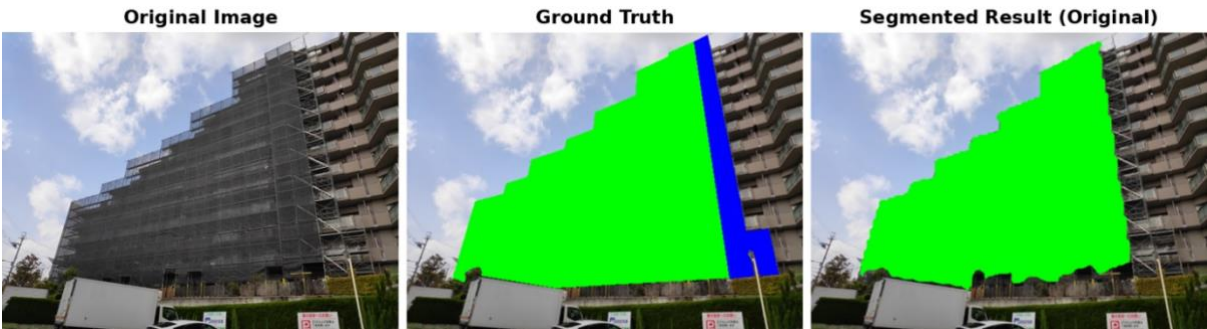


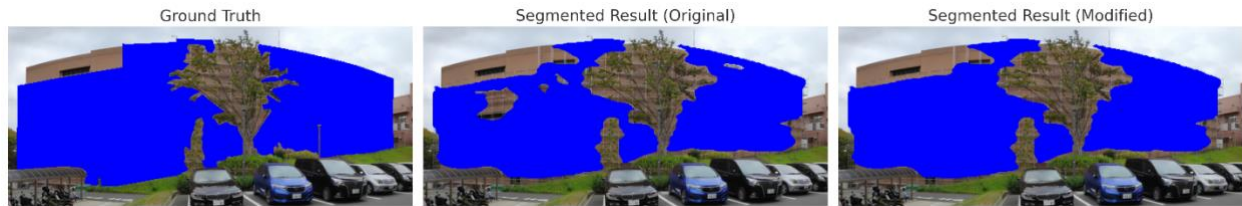
Table 4

Comparison of Segmentation Accuracy Before and After Applying the Mask Modification Algorithm for the First Time Point of Sample 1

Class	IoU (%)		
	Without Modification	With Modification	Difference
Scaffolding without Safety Sheets	81.23	84.33	+3.10

Figure 19

Visual Comparison of Segmented Masks Before and After Applying the Mask Modification Algorithm for the First Time Point of Sample 1



Sample 1 Time Point 2

In the second time point of Sample 1, the mask modification algorithm was applied to the Scaffolding with Safety Sheet class to evaluate its impact on segmentation accuracy. The performance of the modification is demonstrated in Table 5. Initially, the unmodified segmentation achieved an IoU of 93.22%. However, after applying the mask modification algorithm, the IoU slightly decreased to 93.07%, resulting in a negative difference of -0.15%. Figure 20 demonstrates the comparison between the ground truth, the original segmentation result, and the modified segmentation result of the second time point of Sample 1.

Sample 2 Time Point 1

The mask modification algorithm was applied to the first time point of Sample 2 to assess its impact on segmentation performance for both scaffolding with and without safety sheets. The comparison of segmentation accuracy before and after the modification is displayed in Table 6. For the scaffolding without safety sheets class, the IoU improved slightly from 68.19% to 68.78%, resulting in a positive difference of +0.59% after applying the modification. This indicates that the algorithm was effective in marginally refining the segmentation without introducing significant errors. However, for the Scaffolding with Safety Sheet class, the IoU remained unchanged at 85.50%, suggesting that there are no voids or outliers to be justified for this class. Overall, the segmentation accuracy improved by +0.29%, from 76.85% to 77.14%, as shown in Figure 21. The visual comparison of the original and modified segmentation results confirms these findings, with only minor changes observed between the two versions.

Table 5

Comparison of Segmentation Accuracy Before and After Applying the Mask Modification Algorithm for the Second Time Point of Sample 1

Class	IoU (%)		
	Without Modification	With Modification	Difference
Scaffolding with Safety Sheet	93.22	93.07	-0.15

Figure 20

Visual Comparison of Segmented Masks Before and After Applying the Mask Modification Algorithm for the Second Time Point of Sample 1

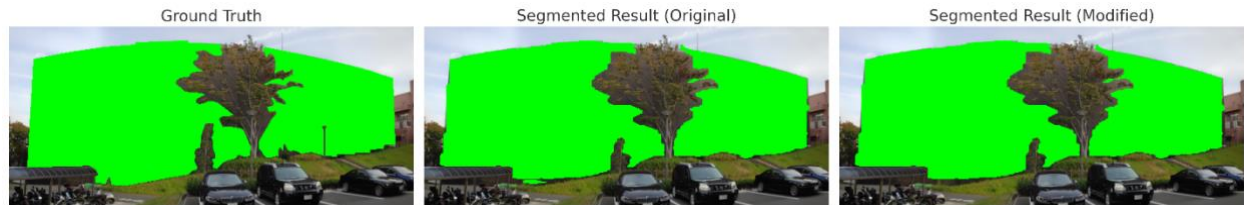


Table 6

Comparison of Segmentation Accuracy Before and After Applying the Mask Modification Algorithm for the First Time Point of Sample 2

Class	IoU (%)		
	Without Modification	With Modification	Difference
Scaffolding without Safety Sheet	68.19	68.78	+0.59
Scaffolding with Safety Sheet	85.50	85.50	0.00
Overall	76.85	77.14	+0.29

Figure 2

Visual Comparison of Segmented Masks Before and After Applying the Mask Modification Algorithm for the First Time Point of Sample 2



Sample 2 Time Point 2

The benchmark before and after modification is shown in Table 7. The Scaffolding without Safety Sheets class showed a slight improvement, with the IoU increasing from 71.17% to 72.01%, resulting in a difference of +0.84%. For the scaffolding with safety sheet class, the IoU also improved, rising from 86.61% to 87.58%, an increase of +0.97%. Overall, the segmentation accuracy for this time point improved by +0.32%, from 79.01% to 79.33%. As illustrated in Figure 22, the modifications provided subtle but consistent enhancements to the segmentation results, demonstrating the effectiveness of the algorithm in refining the scaffolding highlights and improving the overall accuracy of the segmentation. There was no modification for the third time point because there were no voids or outliers in the sample.

Actual Progress Tracking Evaluation

The performance of the proposed system was evaluated by comparing its calculated progress values against groundtruth progress derived from annotated building and scaffolding boundaries. All cases in each sample were assessed. Differences between the system output and the groundtruth were computed for each case, as shown in Table 8.

The results indicate that the proposed system produces progress estimations closely aligned with the groundtruth, with an average deviation of -6.14% across all cases. The smallest deviation was -0.89% (Sample 2 Time Point 2), while the largest was -12.97% (Sample 2 Time Point 3).

Figure 23 to Figure 27 illustrate side-by-side comparisons for all cases, with one figure showing the area at the completion of scaffolding installation and the other providing the system's visualization of the detected progress. These visualizations correspond to the numerical results in Table 8, providing a clear depiction of the alignment between the proposed system's outputs and the building boundary annotations.

Table 1
Comparison of Segmentation Accuracy Before and After Applying the Mask Modification Algorithm for the Second Time Point of Sample 2

Class	IoU (%)		
	Without Modification	With Modification	Difference
Scaffolding without Safety Sheet	71.17	72.01	+0.84
Scaffolding with Safety Sheet	86.61	87.58	+0.97
Overall	79.01	79.33	+0.32

Figure 22

Visual Comparison of Segmented Masks Before and After Applying the Mask Modification Algorithm for the Second Time Point of Sample 2

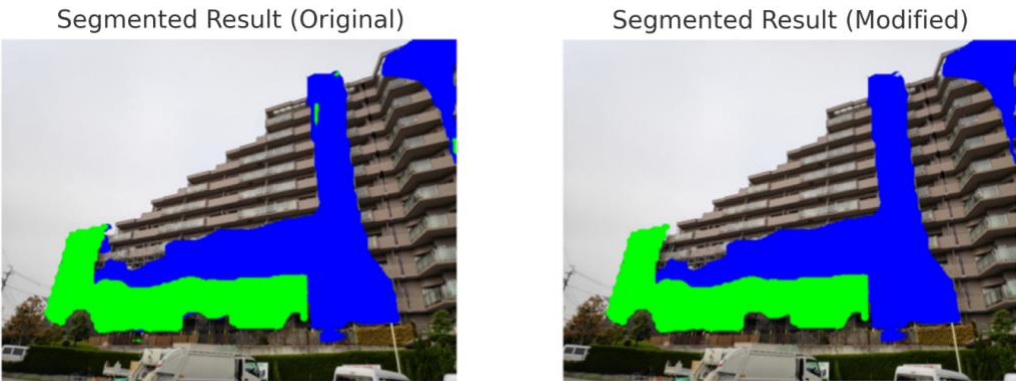


Table 8

Comparison Between the Proposed System’s Calculated Progress and the Groundtruth Progress.

Cases	Actual Progress (%)		
	Proposed System	Groundtruth	Difference
Sample 1 Time Point 1	65.42	75.19	-9.77
Sample 1 Time Point 2	96.16	99.68	-3.52
Sample 2 Time Point 1	24.86	28.40	-3.54
Sample 2 Time Point 2	55.05	55.94	-0.89
Sample 2 Time Point 3	84.00	96.97	-12.97
Average			-6.14

Figure 23

Base Mask (Top) and Calculated Actual Progress (Bottom) of the First Time Point of Sample 1



Figure 24

Base Mask (Top) and Calculated Actual Progress (Bottom) of the Second Time Point of Sample 1



Figure 25

Base Mask (Left) and Calculated Actual Progress (Right) of the First Time Point of Sample 2



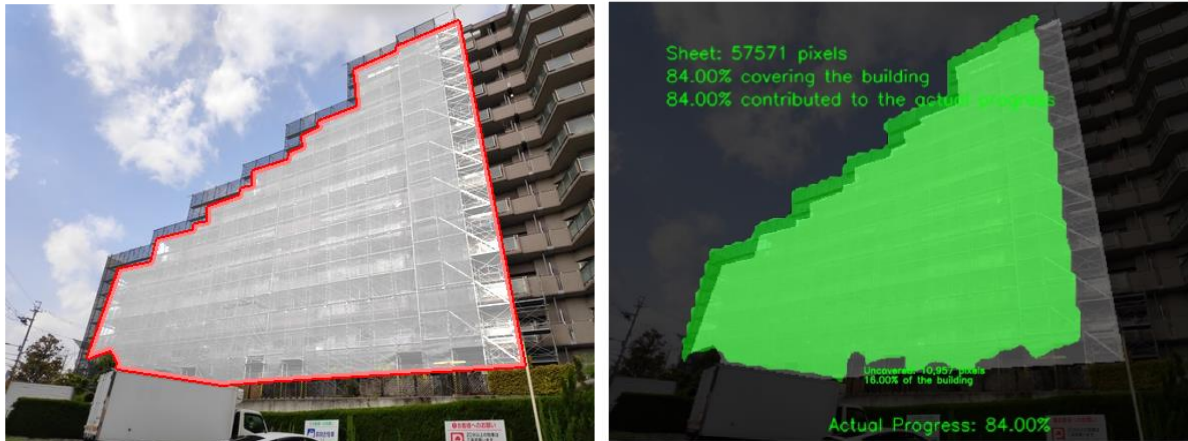
Figure 26

Base Mask (Left) and Calculated Actual Progress (Right) of the Second Time Point of Sample 2



Figure 27

Base Mask (Left) and Calculated Actual Progress (Right) of the Third Time Point of Sample 2



DISCUSSION

The model showed good generalization, with only slight performance declines from validation to test sets. Overall, mAP decreased from 0.496 to 0.480 and F1 from 0.531 to 0.511, due to more challenging unseen examples. For the scaffolding with safety sheets class, performance improved on the test set (mAP: 0.527→0.552; F1: 0.558→0.580), with higher recall indicating better true positive detection, likely due to more distinct visual features. In contrast, the Scaffolding without Safety Sheets class performance declined (mAP: 0.464→0.407; F1: 0.503→0.442), likely due to visual ambiguity and limited training data, as sites typically install safety sheets promptly for safety compliance. Additional data or augmentation may improve detection for this class.

The mask modification algorithm generally enhanced segmentation by filling small voids and removing noise. For example, in Sample 1 Time Point 1, IoU for the Scaffolding without Safety Sheets class improved by +3.10%. However, in Sample 1 Time Point 2, it mistakenly removed valid scattered segments (Figure 28), causing a -0.15% IoU drop for the scaffolding with safety sheets class. In Sample 2 Time Point 2, it failed to remove large misclassifications (Figure 29), highlighting its limitation for prominent or complex errors. Human oversight is recommended before applying the algorithm in such cases.

Actual progress tracking was evaluated against ground-truth progress derived from annotated

building and scaffolding boundaries (Table 8). Across five cases, the proposed system achieved an average deviation of -6.14% from ground truth, with the smallest difference of -0.89% and the largest of -12.97%. This consistent underestimation trend suggests that the model tends to slightly under-detect scaffolding areas, particularly in cases without safety sheets, where visual contrast and edge definition are weaker. Such conditions make the Scaffolding without Safety Sheets class more difficult to identify accurately, resulting in lower pixel coverage and consequently smaller progress values. Nonetheless, the deviation remains within an acceptable range for practical applications, demonstrating the robustness of the proposed approach while highlighting an area for improvement in future model refinement.

The performance of the proposed model is consistent with the results reported in previous studies. Ding et al. (2025) achieved IoU values ranging from 82.13% to 92.74%, Lin et al. (2023) reported a precision of 75%, and Dzung et al. (2024) obtained an mAP of 0.88 for construction-related segmentation tasks. In comparison, our model achieved IoU values between 81.23% and 93.22% for the first sample, which featured less visual complexity, and 70.82% to 82.84% for the second, more complex sample. These results demonstrate that our approach performs on par with or even slightly better than the range of state-of-the-art research, depending on scene complexity.

The proposed automatic process provides clear advantages over conventional area-based estimation, which relies on manual annotation and fixed ratios. By automatically detecting, classifying, and refining scaffolding regions, the system enables consistent and repeatable progress measurement while reducing human intervention. Moreover, its ability to distinguish between scaffolding with and without safety sheets allows for more detailed and practical progress evaluation on active construction sites. Overall, the system demonstrates strong potential for automated scaffolding tracking with real-time performance (>30 FPS).

CONCLUSION

This study evaluated a deep learning-based system for tracking scaffolding progress on construction sites, distinguishing between Scaffolding with Safety Sheets and Scaffolding without Safety Sheets classes. The methodology combined real-time image segmentation, case studies, and a mask modification algorithm to enhance detection accuracy and efficiency, aiming to reduce reliance on manual inspections and improve construction safety monitoring.

Figure 3

Portion of the Scaffolding Correctly Segmented but Removed by the Mask Modification Algorithm due to its Scattered Appearance



Figure 4

Wrong Segmentation Mask Not Removed by the Mask Modification Algorithm



The system achieved strong segmentation performance for the scaffolding with safety sheets class, with an mAP of 0.552 and F1 score of 0.580 on the test set, indicating good generalization. Performance for the Scaffolding without Safety Sheets class was lower (mAP: 0.407; F1: 0.442), highlighting the need for additional training data and improved detection techniques for visually ambiguous configurations.

The mask modification algorithm yielded mixed outcomes, improving IoU by +3.10% in some cases but occasionally removing valid segments, underscoring the importance of parameter tuning or human oversight in complex or fragmented scenes.

Progress tracking, evaluated against groundtruth from annotated building and scaffolding boundaries, showed an average deviation of -6.14% across five cases. The smallest deviation was -0.89%, while the largest (-12.97%) resulted from under-detection of the Scaffolding without Safety Sheets class, a feature underrepresented in training data.

This study has several limitations that suggest directions for future improvement. First, the proposed system was validated using single-viewpoint images captured mainly from outside the building, which may restrict its applicability in cases where scaffolding is not fully visible from a single perspective. Integrating multiple viewpoints could be an interesting direction for improving detection performance. Second, although the dataset includes diverse environmental and lighting conditions, the sample distribution among classes is somewhat unbalanced, particularly for scaffolding without safety sheets, which is harder to capture, resulting in slightly reduced detection accuracy. Future studies should address these issues by incorporating multi-view image data, extending the framework to integrate 3D or point-cloud information for volumetric progress estimation, and expanding the dataset to improve class balance and robustness. These advancements will further enhance the practicality and scalability of automated progress-tracking systems in complex construction environments.

DISCLOSURES AND ACKNOWLEDGEMENTS

The authors wish to acknowledge the funding support received through King Mongkut's University of Technology North Bangkok: Fundamental Fund [grant number KMUTNB-FF-66-39].

REFERENCES

- Bec, J., Blazik-Borowa, E., & Szer, J. (2023). Analysis of scaffolding harmonic excitation. *Bulletin of the Polish Academy of Sciences Technical Sciences*, Article 144577. <https://journals.pan.pl/dlibra/publication/144577/edition/126209/content>
- Blazik-Borowa, E., Robak, A., Pieńko, M., & Czepiżak, D. (2023). The measurement of axial forces in scaffolding standards. *Measurement: Journal of the International Measurement Confederation*, 223, Article 113770. <https://doi.org/10.1016/j.measurement.2023.113770>
- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. *Proceedings of the IEEE International Conference on Computer Vision, 2019*, 9156–9165. <https://doi.org/10.1109/ICCV.2019.00925>
- Bradski, G. (2000). The openCV library. *Dr. Dobbs's Journal of Software Tools*, 25(11), 120–123.
- Chern, W. C., Kim, T., Asari, V. K., & Kim, H. (2024). Image hashing-based shallow object tracker for construction worker monitoring in scaffolding scenes. *Automation in Construction*, 166, Article 105604. <https://doi.org/10.1016/j.autcon.2024.105604>
- Choo, H., Lee, B., Kim, H., & Choi, B. (2023). Automated detection of construction work at heights and deployment of safety hooks using IMU with a barometer. *Automation in Construction*, 147, Article 104714. <https://doi.org/10.1016/j.autcon.2022.104714>

- Chua, W. P., & Cheah, C. C. (2024). Deep-learning-based automated building construction progress monitoring for prefabricated prefinished volumetric construction. *Sensors*, 24(21), Article 7074. <https://doi.org/10.3390/s24217074>
- Ding, Y., Liu, M., Zhang, M., & Luo, X. (2025). Removing visual occlusion of construction scaffolds via a two-step method combining semantic segmentation and image inpainting. *Engineering Applications of Artificial Intelligence*, 142, Article 109983. <https://doi.org/10.1016/j.engappai.2024.109983>
- Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99, 52–58. <https://doi.org/10.1016/j.autcon.2018.11.028>
- Dzeng, R. J., Cheng, C. W., & Cheng, C. Y. (2024). A scaffolding assembly deficiency detection system with deep learning and augmented reality. *Buildings*, 14(2), Article 385. <https://doi.org/10.3390/buildings14020385>
- Ekanayake, B., Wong, J. K. W., Fini, A. A. F., Smith, P., & Thengane, V. (2024). Deep learning-based computer vision in project management: Automating indoor construction progress monitoring. *Project Leadership and Society*, 5, Article 100149. <https://doi.org/10.1016/j.plas.2024.100149>
- Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., & Azorin-Lopez, J. (2016). PointNet: A 3D convolutional neural network for real-time object class recognition. *Proceedings of the International Joint Conference on Neural Networks*, 2016, 1578–1584. <https://doi.org/10.1109/IJCNN.2016.7727386>
- Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2015). Automated progress monitoring using unordered daily construction photographs and IFC-based Building Information Models. *Journal of Computing in Civil Engineering*, 29(1), Article 04014025. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000205](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205)
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <http://www.deeplearningbook.org/>
- Guan, H., Su, Y., Sun, X., Xu, G., Li, W., Ma, Q., Wu, X., Wu, J., Liu, L., & Guo, Q. (2020). A marker-free method for registering multi-scan terrestrial laser scanning data in forest environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166, 82–94. <https://doi.org/10.1016/j.isprsjprs.2020.06.002>
- Gupta, S., & Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161, 466–474. <https://doi.org/10.1016/j.procs.2019.11.146>
- Hamledari, H., McCabe, B., Davari, S., & Shahi, A. (2017). Automated schedule and progress updating of IFC-based 4D BIMs. *Journal of Computing in Civil Engineering*, 31(4), Article 04017012. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000660](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000660)
- Hannan Qureshi, A., Alaloul, W. S., Wing, W. K., Saad, S., Ammad, S., & Musarat, M. A. (2022). Factors impacting the implementation process of automated construction progress monitoring. *Ain Shams Engineering Journal*, 13(6), Article 101808. <https://doi.org/10.1016/j.asej.2022.101808>
- Harris, C. R., Millman, K. J., der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2980–2988). <https://doi.org/10.1109/ICCV.2017.322>

- Hou, L., Zhao, C., Wu, C., Moon, S., & Wang, X. (2017). Discrete firefly algorithm for scaffolding construction scheduling. *Journal of Computing in Civil Engineering*, 31(3), 1–15. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000639](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000639)
- Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2019). *RandLA-Net: Efficient semantic segmentation of large-scale point clouds*. Arxiv. <http://arxiv.org/abs/1911.11236>
- Hui, L., Park, M., & Brilakis, I. (2014). Automated in-place brick counting for facade construction progress estimation. *Computing in Civil and Building Engineering*, 29(6), 958–965. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000423](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000423)
- Hüthwohl, P., & Brilakis, I. (2018). Detecting healthy concrete surfaces. *Advanced Engineering Informatics*, 37, 150–162. <https://doi.org/10.1016/j.aei.2018.05.004>
- Kavaliauskas, P., Fernandez, J. B., McGuinness, K., & Jurelionis, A. (2022). Automation of construction progress monitoring by integrating 3D point cloud data with an IFC-based BIM model. *Buildings*, 12(10), Article 1754. <https://doi.org/10.3390/buildings12101754>
- Khosakitchalert, C., & Seghier, T. E. (2025). Evaluating and optimizing acoustical reverberation time and material cost for classrooms using Building Information Modeling (BIM) and Generative Design (GD) tools. *Nakhara: Journal of Environmental Design and Planning*, 24(1), Article 504. <https://doi.org/10.54028/NJ202524504>
- Khosakitchalert, C., & Yabuki, N. (2023). Conversion challenges: A case study of converting a post and lintel structure to a precast concrete structure using Building Information Modeling (BIM). *Nakhara: Journal of Environmental Design and Planning*, 22(3), Article 316. <https://doi.org/10.54028/NJ202322316>
- Kim, H., & Kim, C. (2020). Deep-learning-based classification of point clouds for bridge inspection. *Remote Sensing*, 12(22), Article 3757. <https://doi.org/10.3390/rs12223757>
- Kim, J., Chung, D., Kim, Y., & Kim, H. (2022). Deep learning-based 3D reconstruction of scaffolds using a robot dog. *Automation in Construction*, 134, Article 104092. <https://doi.org/10.1016/j.autcon.2021.104092>
- Kim, K., Cho, Y. K., & Kim, K. (2018a). BIM-based decision-making framework for scaffolding planning. *Journal of Management in Engineering*, 34(6), 1–37. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000656](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000656)
- Kim, K., Cho, Y., & Kim, K. (2018b). BIM-driven automated decision support system for safety planning of temporary structures. *Journal of Construction Engineering and Management*, 144(8), 1–11. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001519](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001519)
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). *Segment anything*. Arxiv. <http://arxiv.org/abs/2304.02643>
- Lin, P., Lin, J. J., & Hsieh, S. (2023). Construction site scaffolding completeness detection based on mask R- CNN and hough transform. *30th EG-ICE: International Conference on Intelligent Computing in Engineering* (pp. 1–10). <https://doi.org/10.48550/arXiv.2503.14716>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). *Microsoft COCO: Common objects in context*. ArXiv. <https://arxiv.org/abs/1405.0312>

- Mani, G. F., Feniosky, P. M., & Savarese, S. (2009). D4AR-A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. *Electronic Journal of Information Technology in Construction*, 14, 129–153. https://www.itcon.org/papers/2009_13.content.06965.pdf
- Matarneh, S., Elghaish, F., Pour Rahimian, F., Abdellatef, E., & Abrishami, S. (2024). Evaluation and optimisation of pre-trained CNN models for asphalt pavement crack detection and classification. *Automation in Construction*, 160, Article 105297. <https://doi.org/10.1016/j.autcon.2024.105297>
- Omara, H., Mahdjoubia, L., & Kheder, G. (2018). Towards an automated photogrammetry-based approach for monitoring and controlling construction site activities. *Computers in Industry Journal*, 98, 172–182. <https://doi.org/10.1016/j.compind.2018.03.012>
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., ... Bojanowski, P. (2023). *DINOv2: Learning robust visual features without supervision*. Arxiv. <http://arxiv.org/abs/2304.07193>
- Paoletti, M. E., Haut, J. M., Plaza, J., & Plaza, A. (2019). Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158, 279–317. <https://doi.org/10.1016/j.isprsjprs.2019.09.006>
- Park, M., Tran, D. Q., Bak, J., & Park, S. (2023). Small and overlapping worker detection at construction sites. *Automation in Construction*, 151, Article 104856. <https://doi.org/10.1016/j.autcon.2023.104856>
- Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., & Zhou, X. (2020). Deep snake for real-time instance segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 8530–8539). <https://doi.org/10.1109/CVPR42600.2020.00856>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779–788). <https://doi.org/10.1109/CVPR.2016.91>
- Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77, 157–173. <https://doi.org/10.1007/s11263-007-0090-8>
- Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM handbook: A guide to Building Information Modeling for owners, managers, designers, engineers, and contractors* (third). John Wiley & Sons.
- Saovana, N., Yabuki, N., & Fukuda, T. (2020). Development of an unwanted-feature removal system for Structure from Motion of repetitive infrastructure piers using deep learning. *Advanced Engineering Informatics*, 46, Article 101169. <https://doi.org/10.1016/j.aei.2020.101169>
- Saovana, N., Yabuki, N., & Fukuda, T. (2021). Automated point cloud classification using an image-based instance segmentation for structure from motion. *Automation in Construction*, 129, Article 103804. <https://doi.org/10.1016/j.autcon.2021.103804>
- Tochaiwat, K., & Seniwong, P. (2025). Sales rate prediction for condominiums in the Bangkok Metropolitan Region using deep learning: Identification of determinants and model validation. *Nakhara: Journal of Environmental Design and Planning*, 24(1), Article 502. <https://doi.org/10.54028/NJ202524502>
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Guillard, E., Yu, T., & the scikit-image contributors. (2014). Scikit-image: Image processing in Python. *PeerJ*, 2, Article e453. <https://doi.org/10.7717/peerj.453>

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Mulbregt, P. van. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272.
<https://www.nature.com/articles/s41592-019-0686-2>

Wang, B., Chen, Z., Li, M., Wang, Q., Yin, C., & Cheng, J. C. P. (2024). Omni-scan2BIM: A ready-to-use scan2BIM approach based on vision foundation models for MEP scenes. *Automation in Construction*, 162, Article 105384.
<https://doi.org/10.1016/j.autcon.2024.105384>

Xu, Y., Tuttas, S., Hoegner, L., & Stilla, U. (2018). Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Automation in Construction*, 85, 76–95.
<https://doi.org/10.1016/j.autcon.2017.09.014>

Yang, F., Odashima, S., Masui, S., & Jiang, S. (2023). Hard to track objects with irregular motions and similar appearances? Make it easier by buffering the matching space. *Proceedings - 2023 IEEE Winter Conference on Applications of Computer Vision, WACV 2023* (pp. 4788–4797).
<https://doi.org/10.1109/WACV56688.2023.00478>

Yin, Z., & Caldas, C. (2022). Scaffolding in industrial construction projects: Current practices, issues, and potential solutions. *International Journal of Construction Management*, 22(13), 2554–2563.
<https://doi.org/10.1080/15623599.2020.1808562>