Automation in Construction 140 (2022) 104327

Contents lists available at ScienceDirect

Automation in Construction

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

Integrated vision-based automated progress monitoring of indoor construction using mask region-based convolutional neural networks and BIM

Wei Wei [a], Yujie Lu [a,b,c,*], Tao Zhong [a], Peixian Li [a], Bo Liu [a]

[a] College of Civil Engineering, Tongji University, Shanghai 200092, China
[b] Key Laboratory of Performance Evolution and Control for Engineering Structures of Ministry of Education, Tongji University, Shanghai 200092, China
[c] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 200092, China

ARTICLE INFO

Keywords:
BIM
Indoor construction
Mask R-CNN
Progress assessment
Project management

ABSTRACT

Traditional construction progress tracking relies on labor-intensive activities with time lags, potential man-made errors, and inefficient progress management, which demands for an innovative and automated progress tracking approach. This paper describes a deep learning method that utilizes image segmentation to automatically evaluate the wall construction progress of an entire floor with the progress results streamlined to BIM. The approach was applied to a case study in China for assessing plastering construction activities with high segmentation accuracy (mean average precision = 96.8%). Further improvement of Mask Region-Based Convolutional Neural Networks (Mask R-CNN) and evaluation of its superiority over other models have also been discussed. This study provides both theoretical and practical references for unmanned supervision of progress tracking and intelligent schedule management.

1. Introduction

Scientific construction management plays a critical role in ensuring successful project delivery by improving the construction quality and increasing profit. As an essential piece of construction management, progress management is closely associated with cost management. The project gains an advantage in the market competition by controlling the schedule and minimizing costs. Conventional progress management relies on manual data sorting [1] of construction drawings and site progress reports [2], which is an expensive, time-consuming, and labor-intensive task [3]. In particular, it is challenging to manage an indoor construction site considering building material piles and complex pipeline layouts, so it often requires an extra workforce and financial cost to monitor the indoor construction progress. Moreover, when evaluating construction progress, there lacks a platform of progress visualization so the project manager cannot evaluate the project progress effectively and make timely decisions. Integrating progress information into Building Information Modeling (BIM) can facilitate faster recognition of schedule delays intuitively so as to avoid extra costs. In summary, it is imperative to propose an approach for automatic progress monitoring to promote unmanned supervision for efficient construction management.

The assessment method of construction progress varies with the building structure types. The progress evaluation of prefabricated structures is relatively convenient as most structural components have been prefabricated in the factory. Compared to prefabricated structures, the progress evaluation of a cast-in-place building which involves multiple stages and a mix of wet and dry construction is even more challenging. Delay in each stage could aggregately affect overall project progress and cost overrun. Therefore, it is essential to obtain the construction progress of cast-in-place structures precisely and efficiently, which is the research focus of this study.

Advanced technologies and methods that have been applied to progress monitoring management can be categorized into three types. First, the progress information is obtained based on sensor data. The advantage of this method is that the sensors can be placed anywhere needed [4], but it usually requires a large number of card readers with a high risk of loss. Second, the construction progress is evaluated based on point cloud generation and comparison. The high-precision point cloud can be collected via laser scanners [5], but the data processing is time-

* Corresponding author at: Room A817, Building for the College of Civil Engineering, Tongji University, No. 1239 Siping Road, Yangpu District, Shanghai 200092, China.
E-mail address: Lu6@tongji.edu.cn (Y. Lu).

https://doi.org/10.1016/j.autcon.2022.104327
Received 10 February 2022; Received in revised form 1 May 2022; Accepted 2 May 2022
Available online 10 May 2022

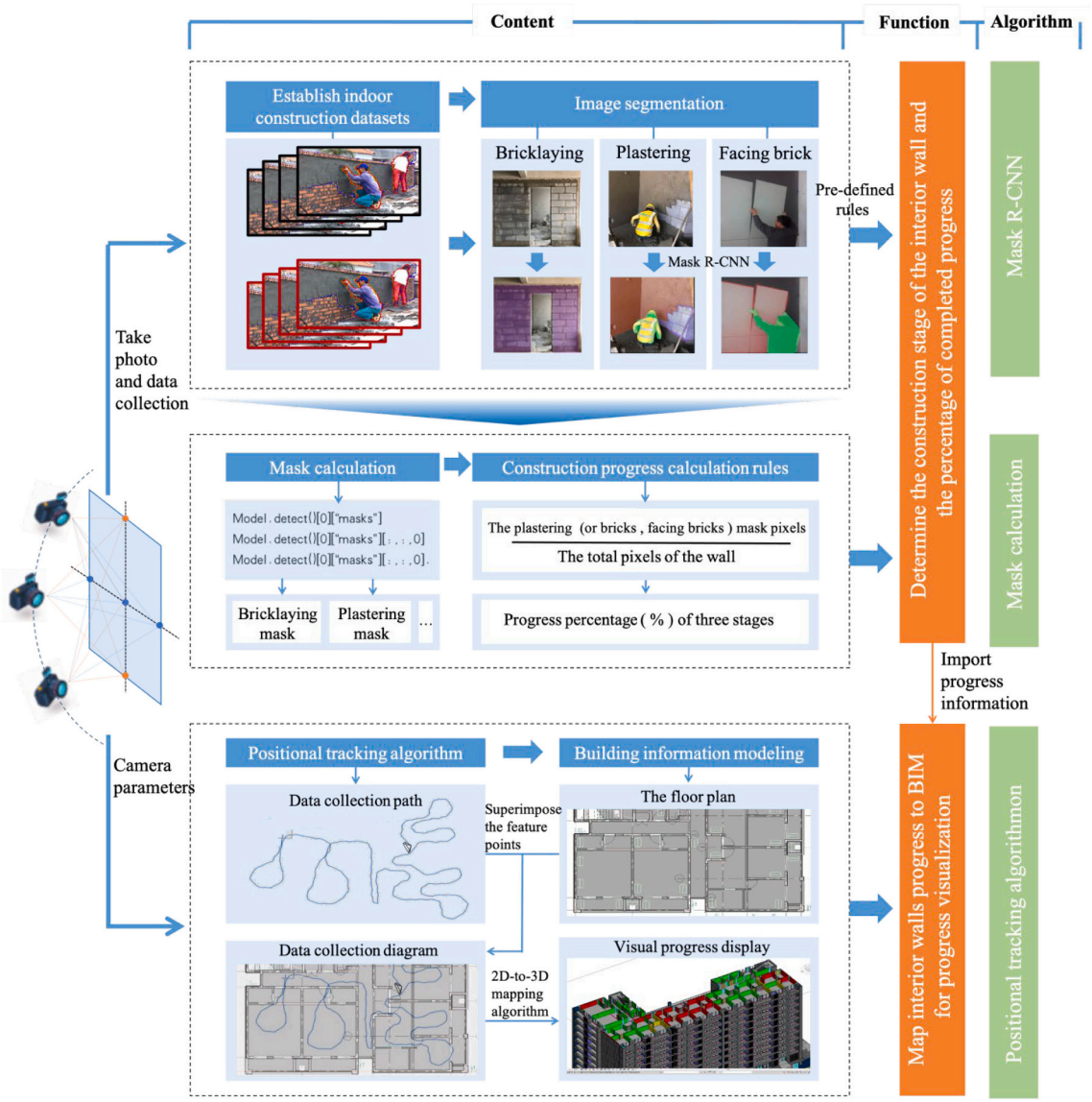


Fig. 1. Overall framework.

Table 1
Construction stages classification rules.

Construction stage	Labeling objects	Stage classification rule	Progress percentage calculation
Bricklaying	bricks and concrete	Both bricks and concrete are detected.	The pixels covered by the brick mask divided by the total pixels of the wall
Plastering	putty and plastering	Only plastering or both plastering and putty are detected.	The pixels covered by the plastering mask divided by the total pixels of the wall
Facing bricks	facing bricks and plastering	Only facing bricks or both facing bricks and plastering are detected.	The pixels covered by the facing bricks mask divided by the total pixels of the wall

consuming, and it is impossible to obtain the real-time construction progress. Third, construction progress is determined through image-assisted point clouds, which utilizes image detection to supplement the missing semantic information in the point cloud generated by Structure from Motion (SfM) [6,7]. However, SfM based on multi-angle image capture is not suitable for generating point clouds indoors because a one-shot image cannot provide indoor panoramic information. Fourth, construction progress is determined based on image processing. As a recently widely used method, computer vision (CV) based image processing is a handy approach for progress evaluation. Pictures of construction sites captured by normal cameras can provide useful progress information via object detection and segmentation algorithms. The CV-based method has unique advantages in construction progress monitoring [8].

The CV-based methods for progress evaluation include qualitative research and quantitative research. The former usually identifies key construction components, e.g., beams and columns [9–12], to classify the construction stage. The latter can determine the quantitative progress through the calculation of the construction area based on edge detection [13]. However, the edge recognition algorithm is prone to irrelevant edges around the target object, leading to low accuracy of area calculation. In addition, the current methods were limited to one-shot photos collected by a fixed camera, which can only capture a limited certain area, e.g., one room or a corridor, rather than the entire area of one floor. Herein, we developed an integrated vision-based framework, which supports automated constructed area recognition

Fig. 2. Open space and enclosed space.

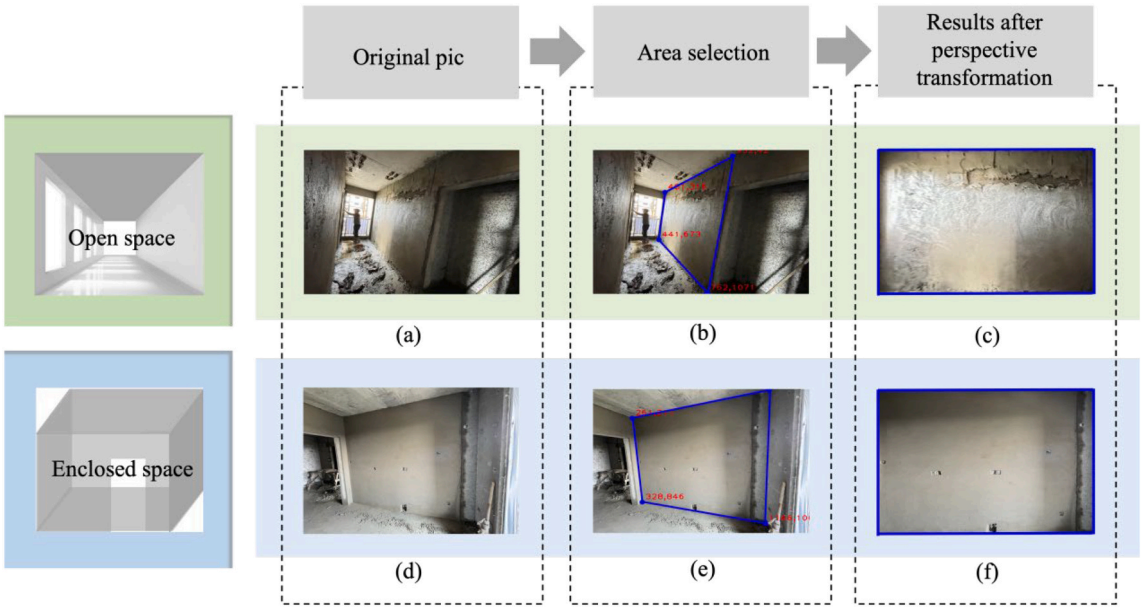


Fig. 3. Image perspective transformation.

and continuous image capture, to promote unmanned supervision of indoor construction. The specific objectives are 1) to propose an image segmentation method to quantitatively calculate wall construction area and corresponding percentile progress using an improved Mask Region-Based Convolutional Neural Networks (Mask R-CNN) algorithm; 2) to provide an evaluation method to assess the wall construction progress of the entire floor utilizing Simultaneous Localization and Mapping (SLAM) -based continuous image capture.

This article is composed of seven sections. The following provides a comprehensive literature review on three types of construction progress evaluation methods: sensor data fusion, point cloud generation and comparison, and image deep learning. Section 3 introduces the overall framework and data acquisition and processing. Section 4 describes a case study in Shanghai, China, using the proposed approach. Section 5 verifies the accuracies of the used Mask R-CNN algorithm and the Positional Tracking algorithm. In the next section, we discuss the improvements made to Mask R-CNN, including loss function, optimizer, and cosine annealing, and the impact of image brightness on segmentation accuracy. Finally, the Conclusion section summarizes the whole article and provides an outlook for future research.

2. Related studies

Real-time construction progress monitoring can capture the latest construction progress and help correct schedule deviations in time and reduce construction costs. In recent years, project progress evaluation methods can be categorized into three groups: progress information acquired from sensors, progress evaluation based on point cloud generation and comparison, and progress monitoring based on image processing.

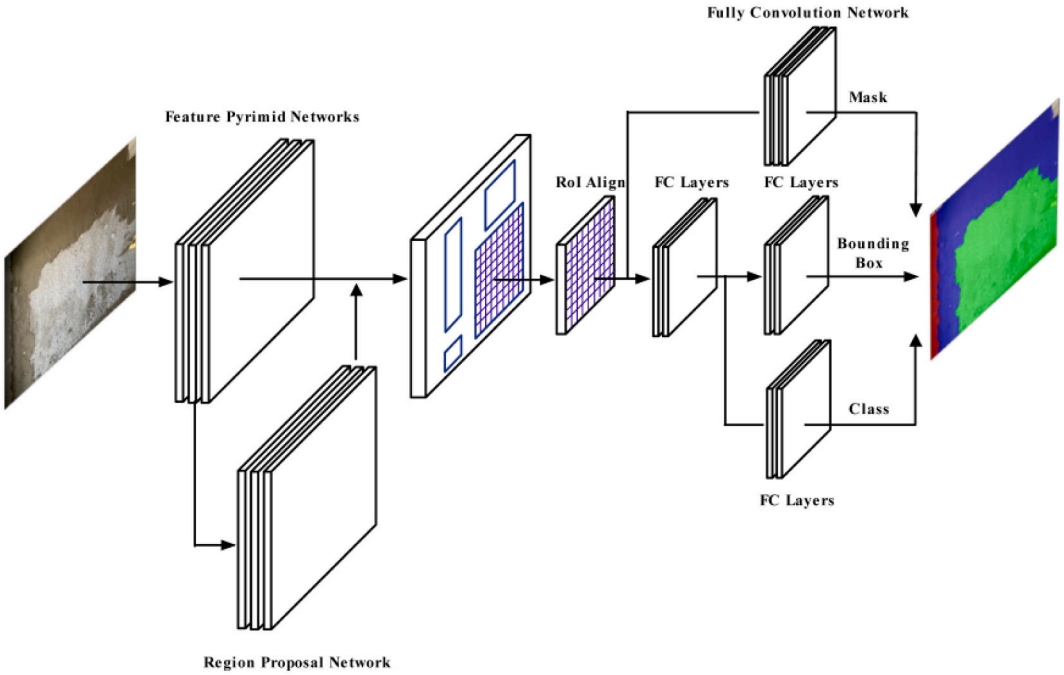


Fig. 4. Network structure of Mask R-CNN.

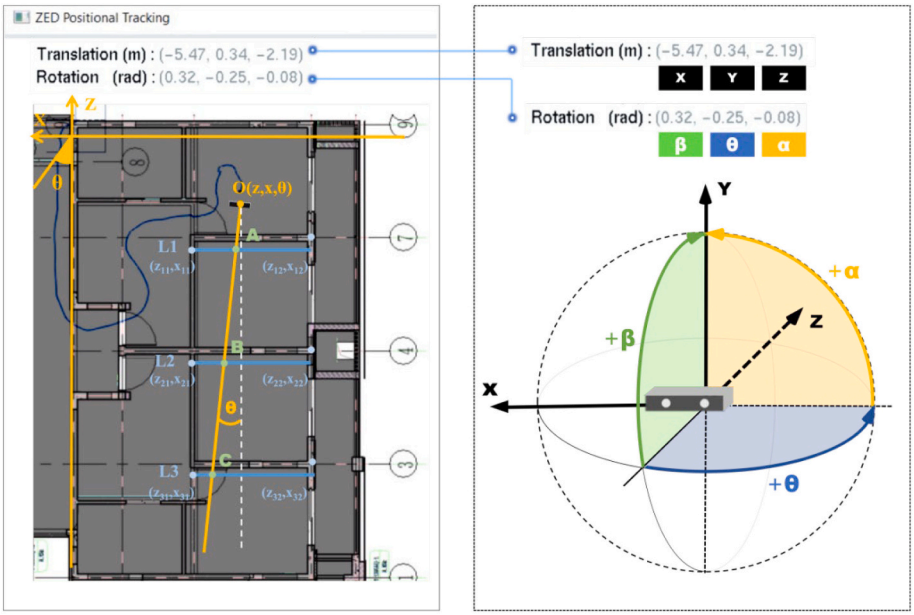


Fig. 5. Zed 2 position tracking algorithm: (a) mathematical principles; (b) camera translation and rotation.

2.1. Progress evaluation based on sensors and point cloud

Setting up sensors to acquire progress information is one of the earliest methods used in progress monitoring. Through Global Positioning System (GPS), Radio-frequency identification (RFID), Quick Response Code (QR code), and other sensors, the location and quantity information of construction components and personnel can be obtained to evaluate construction progress. GPS technology can be applied in material positioning management to effectively save the time of counting materials and improve the efficiency of schedule management [14]. RFID technology is mainly used to track components' positions and construction activities [15], and it can be combined with BIM for progress visualization display [16–18]. QR Codes are often combined with BIM for components inspection and repair [19].

However, the sensor-based approach has some limitations in indoor environments. GPS technology cannot navigate and locate due to the blocked indoor signals [20]. RFID requires signal receiving equipment and recurring installation, scanning, and maintenance [3]. QR code is suitable for prefabricated components, but there is a risk of losing labels due to collisions and frictions during the transportation and installation [21].

The most common method of construction progress evaluation is using the point cloud model generated by laser scanners or oblique photography for progress comparison. This method usually involves two

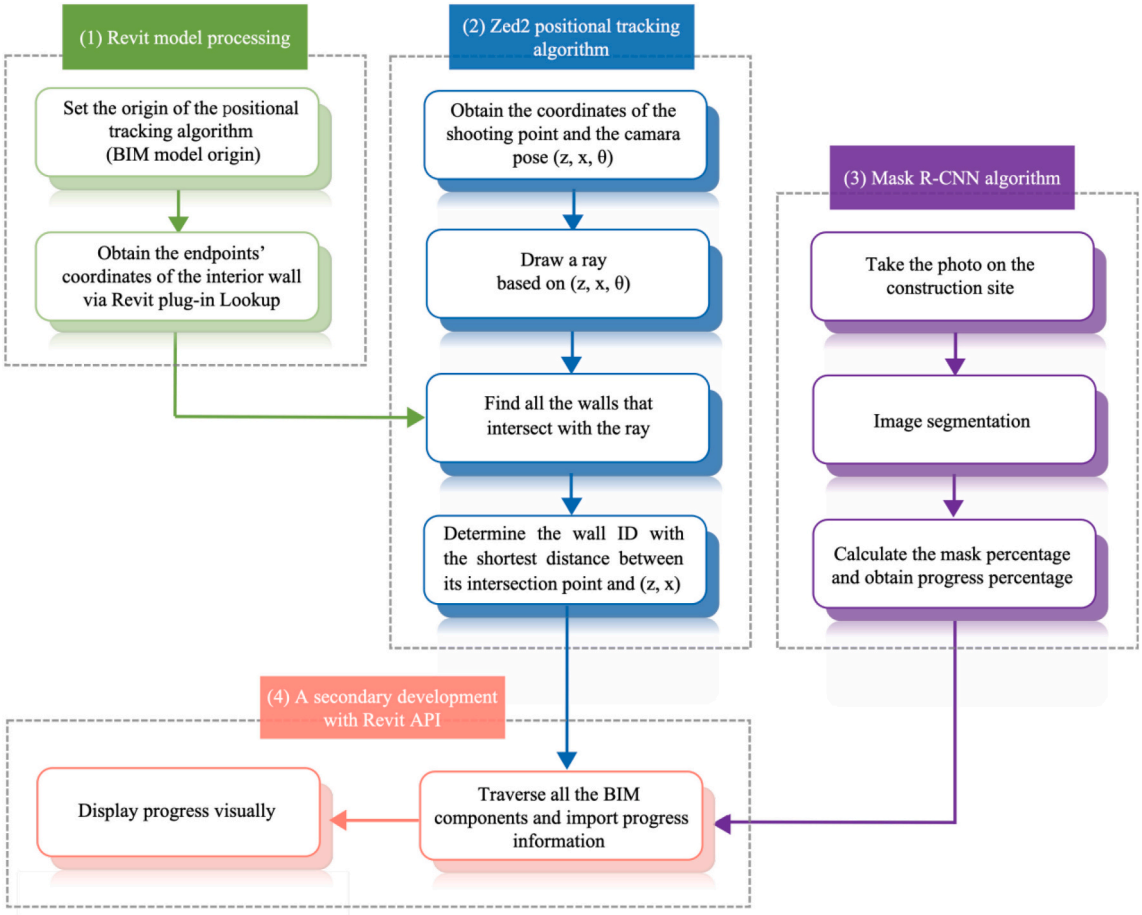


Fig. 6. 2D-to-3D mapping and progress visualization principle.

steps. First, the point cloud model obtained by oblique photography [1,6,22] or laser scanning [22,23] is registered to the as-planed BIM model through local matching, global matching [24], or manually selected feature point matching [6,24]. Second, we can obtain the visual progress differences between the actual point cloud model and the as-planned BIM model [22] through various methods, such as (1) comparing IFC 4D-bim with point cloud in txt format [24] based on support vector machine (SVM) classifier and Bayesian model [1]; (2) calculating the number of point clouds on the surface of BIM components [6], using rays that are cast from the 3D as-planned object [7].

However, three problems are prominent when using point clouds to evaluate construction progress. First, the point cloud model generated by oblique photographic images [25] usually has lower accuracy than laser scanning, but laser scanning is costly and requires professionals to define scanning configurations [26]. Second, the process of matching point cloud and BIM cannot be fully automated, and manual alignment is still time-consuming. Third, blocked building components [25] and transparent components such as windows [27] cannot be captured by laser scanners.

2.2. Progress monitoring based on image processing

Progress monitoring based on image processing has been widely used recently. In this method, the data acquisition is more convenient, and the pictures taken by the camera or mobile phone can provide rich information on construction sites, such as an object's appearance, material, etc. Moreover, image processing can be applied in both outdoor and indoor environments. Therefore, this method is adopted to evaluate indoor progress monitoring.

The progress evaluation based on CV can be summarized into three types. First, the construction stage is identified based on image feature extraction, including the material feature, texture feature, time-series feature, etc. Many studies used improved SVM to identify the objects' material types automatically [28–30], and then combined the information obtained from the image with BIM or point cloud model [3,12,31,32]. Some researchers developed various texture feature extraction methods to recognize the construction stages of drywall construction [33]. In addition, Han, et al. [12] determined the status of the building components (construction completed or no construction) using materials classification and detection. Second, the construction stage is identified by recognizing key objects or personnel activities. Hamledari, et al. [11] classified the construction stage by detecting key construction objects like drywall studs and power outlets. Martinez, et al. [34] used Faster R-CNN to identify workers and key equipment in images and used finite state machines to model task sequences, improving activity tracking accuracy in low-resolution images. Third, the construction progress is assessed by identifying the number or condition of construction components. For example, Deng, et al. identified the specific boundary of regular components such as tiles and bricks based on an improved edge detection algorithm and then calculated the progress by the area ratio of tiles or the number ratio of bricks [13,32]. Wang, et al. [21] identified and tracked the prefabricated wall through Mask R-CNN and Deep SORT algorithm, obtained the wall position and installation time, and then updated the progress in a BIM model.

Nevertheless, there are two limitations in the image-based progress assessment method. First, most studies only recognized a certain stage of the wall construction without calculating the specific constructed area which is essential to evaluate workers' productivity. Although [13] proposed a method of construction area calculation based on multi-edge

```
Input:
    pic_id, ray_z, ray_x, ray_θ
Output:
    Progress, right_id
BEGIN
    /* Step 1: Calculate plaster construction progress */
    get_certain_mask(pic_id) // get the mask value of a certain category, eg:sum(mask_plaster)
    progress←np.sum(mask_plaster)/(np.sum(mask_plaster)+np.sum(mask_putty))
1   /* Step 2: Establish Ray equation and Wall equation */
2   Wall←{"L_i":[(z_i1,x_i1),(z_i2,x_i2)]}←known item
3   ray_angle←math.pi-ray_θ
4   f(z,x)←x-ray_x-(z-ray_z)tan(ray_angle)]==0 // Ray equation
5   g_i(z,x)←(z-z_i2)(x_i1-x_i2)/(z_i1-z_i2)+x_i2==0 // Wall equation
6   /* Step 3: Solve for all walls intersecting the ray equation and calculate intersection points */
7   id_list ←{}, point_list ←{}
8   for key,value in Wall.items() do
9       id_list.append(key)
10      point_list.append(value)
11  end for
12  dieta_z_ij←z_ij-ray_z, dieta_x_ij←x_ij-ray_x
13  cross_wall_number ←{}, cross_wall_id ←{}
14  /* Step 4: Calculate the distances of the intersection points from the camera position */
15  for k in range(len(Wall)) do
16      k←k+1
17      for g in range(1,3) do
18          if dieta_z_kg*cos(ray_angle)>=0 || dieta_x_kg*sin(ray_angle)>=0 || f(z_k1,x_k1)f(z_k2,x_k2)<=0 then
19              cross_wall_number.append(k)
20              cross_wall_id.append(id_list[k])
21              break
22          end if
23      end for
24  end for
25  distance_list ←{}
26  /* Step5: Output the id of the wall in BIM corresponding to the wall in the photo */
27  for j in cross_wall_number do
28      j←j+1
29      print solve([f(z,x)==0, g_j(z,x)==0],[z, x])
30      dist←math.pow(ray_z-z,2)+math.pow(ray_x-x,2)
31      distance_list.append(dist)
32  end for
33  Wall_id←cross_wall_id[distance_list.index(min(distance_list))]
34  END
```

Fig. 7. Pseudo-code of 2D-to-3D mapping.

recognition, area detection accuracy can be influenced by irrelevant edges of indoor sundries [13,32]. Second, current research monitored indoor construction progress through photos collected by a fixed camera, which only obtains the progress of a fixed area, such as one room, rather than the entire floor [12,35].

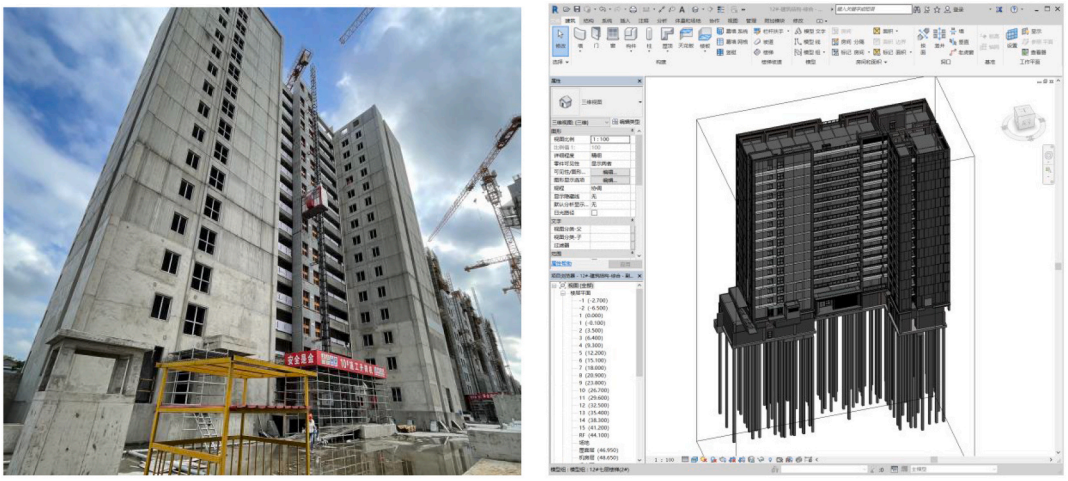
3. Methods

3.1. Research framework

The proposed automated assessment framework of indoor construction progress based on Mask R-CNN and BIM consists of two steps (Fig. 1): (1) determine the construction stage of the interior wall and the quantitative progress and (2) integrate the progress of interior walls into BIM.

For the first step, we established a dataset for indoor construction components segmentation and manually took construction pictures using Zed 2, a hand-held camera with a positioning function. The interior wall construction progress can be classified into three phases: bricklaying, plastering, and facing brick (Table 1), and the labeling objects in each of the three stages are bricks and concrete, putty, and plastering, and facing bricks and plastering, respectively. Then, we calculated the progress percentage of interior walls based on the mask matrix calculation. Taking plastering as an example, it is defined as plastering phase when only plastering or both plastering and putty are detected, and the plastering progress percentage (%) is calculated as the pixels covered by the plastering mask divided by the total pixels of the wall.

For the second step, Zed 2 positional tracking algorithm was used to track the camera and extract the camera position and pose information (3D coordinates and 3D rotation angle) when taking the images (Fig. 5). After that, we proposed a mathematical algorithm based on the camera pose and position to determine the mapping relationship between the walls in the images and the Revit model components. Finally, we used



(a) (b)

Fig. 8. Case building: (a) exterior look at the time of data collection; (b) BIM model.

(a) (b)

Fig. 9. Data collection with Zed 2 camera: (a) data collection; (b) camera parameters. (Resource: https://www.stereolabs.com/zed-2i/).

the Revit API to integrate the progress information obtained in the first step into the Revit model (as shown in Fig. 6).

As the walls around the staircases are inclined and many temporary supports block the view during the construction, it is challenging to shoot and calibrate images of staircases. Therefore, the scope of this study excludes stairwells, and we categorized the indoor construction spaces into two types (Fig. 2): (1) open spaces such as corridors and (2) enclosed spaces such as rooms.

3.2. Image pre-processing

During data collection and processing, we corrected the images taken by Zed 2 using a perspective transformation algorithm to eliminate image distortion and angle effects. Fig. 3 shows the process of image perspective transformation in an enclosed room and open space. After taking the construction site pictures, four corner points of the wall were selected (Fig. 3 (b) and (e)) to perform a perspective transformation. Fig. 3(c) and (f) present the results after correcting the image distortion and eliminating angular deviations influence.

3.3. Image segmentation and mask R-CNN

Computer vision algorithms such as object detection and image segmentation have developed rapidly. Compared with object detection which recognizes objects using bounding boxes, image segmentation can detect the boundary of an object using a mask and accurately classify each pixel. The common segmentation algorithms include DeepMask [36], Fully Convolutional Instance-aware Semantic Segmentation [37], Mask R-CNN, and so on. However, the first two algorithms are not suitable for segmenting overlapping objects. Mask R-CNN [38] is a segmentation algorithm with satisfactory instance segmentation results [39], which can accurately segment overlapping objects [40]. Furthermore, this algorithm is suitable for identifying objects with irregular shapes, such as plastering. Therefore, we use Mask R-CNN (Fig. 4) to segment instances in the scenario of indoor construction. Mask R-CNN uses Feature Pyramid Networks (FPN) to extract image features, applies Region Proposal Networks (RPN) to create anchor boxes of regions of interest (RoI), and then uses network branches for bounding box classification and regression. In addition, the Mask branch carries out mask segmentation through full convolution layers. Furthermore, RoI Align in Mask R-CNN can connect the two-stage networks to make the image and mask correspond accurately and improve segmentation accuracy [38].

The loss function of Mask R-CNN is composed of three parts: classification loss L_{cls}, bounding box regression loss L_{box}, and k-th mask loss L_{mask}. The progress percentage in this study can be obtained by mask calculation. Therefore, the accuracy of the mask and L_{mask} are essential

Fig. 10. Data collection diagram. Point A is the origin. Points B and C are two feature points with known true coordinates.

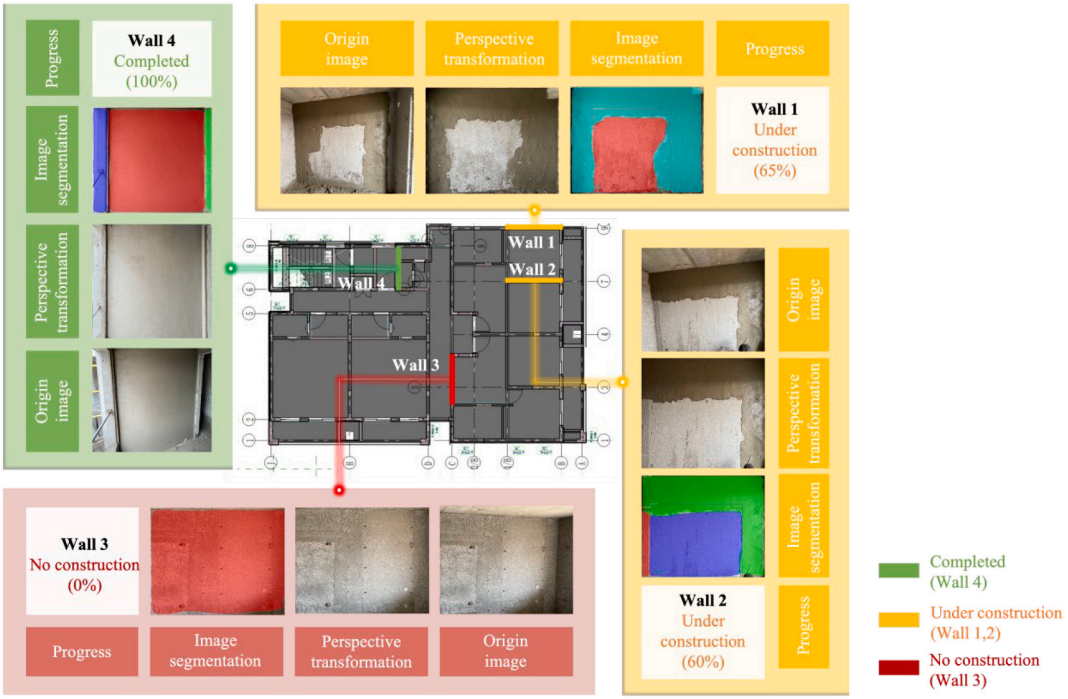


Fig. 11. Images segmentation results in the plastering stage.

factors for progress calculation. The common L_{mask} includes Cross-Entropy Loss (*CE Loss*) [38] (Formula 1) and Dice Loss [41] (Formula 2) which are particularly suitable for multi-category detection.

$$CE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(y_{ij}log\left(\widehat{y_{ij}}\right)\right) \tag{1}$$

Where N is the number of pixels in the picture, m is the number of detection categories, and y_{ij} is the probability of classifying pixel i as category j.

$$\text{Dice Loss} = 1 - D = \frac{2\sum_{i=1}^{N}\sum_{j=1}^{m}p_{ij}q_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{m}p_{ij}^2 + \sum_{i=1}^{N}\sum_{j=1}^{m}q_{ij}^2} \tag{2}$$

Where N is the number of pixels in the picture, m is the number of detection categories, p_{ij} is the predicted binary segmentation volume, and q_{ij} is the ground truth of binary volume.

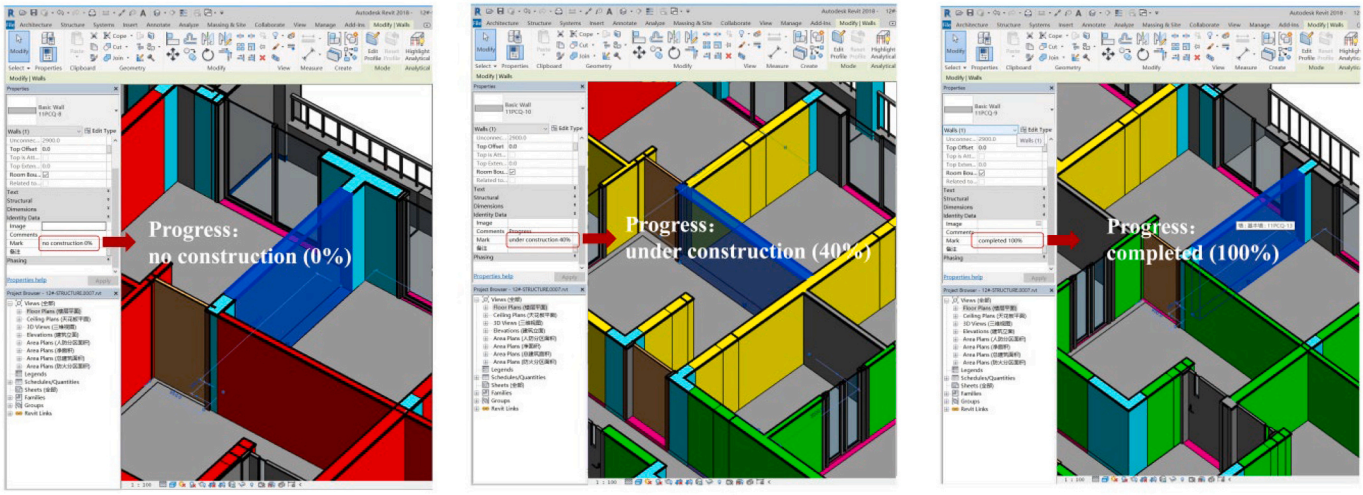


Fig. 12. Visualization of the construction progress of walls: (a) no construction; (b) under construction; (c) completed.

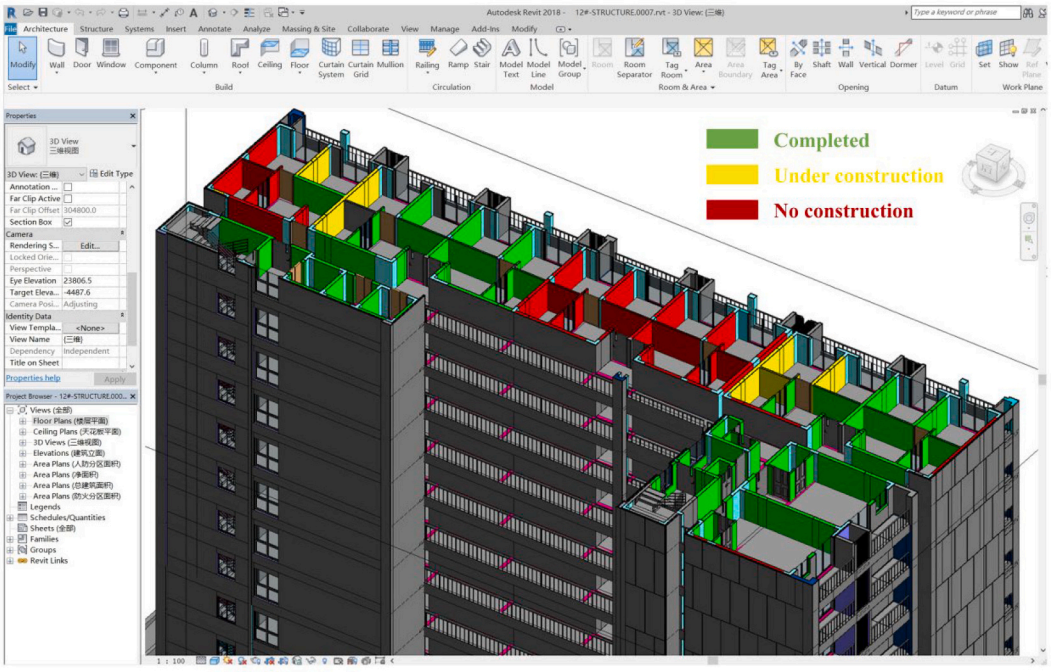


Fig. 13. Visualization of the construction progress of the entire floor.

3.4. 2D-to-3D mapping and progress visualization

When taking images with Zed 2 camera, we used a positional tracking algorithm to track the camera movement path in real-time and record camera pose information (Fig. 5a). Taking the original position of the Zed 2 camera as the origin, we could obtain the 3D coordinates (x, y, z) and the 3D rotation angle information (β, θ, α) of the camera left lens relative to the origin in real-time during the movement of the camera. The positive direction of the 3D coordinates and rotation angle are shown in Fig. 5b.

To map the wall components in 2D images to the wall components in 3D BIM models, we proposed a procedure using Revit model processing, Zed 2 positional tracking algorithm, Mask R-CNN algorithm, and a secondary development with Revit API. Fig. 6 shows the procedure in four steps: the first step is to obtain the BIM components information; the next step is to determine the relationship between the components in the pictures and the components in BIM; the following is to recognize the construction progress; in the end, components progress is mapped into BIM using a Plug-in (Revit API).

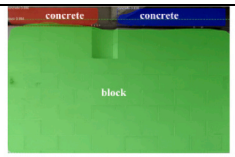
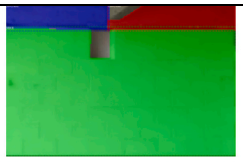
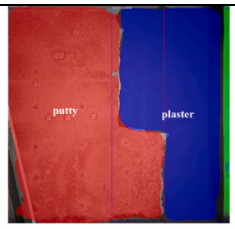
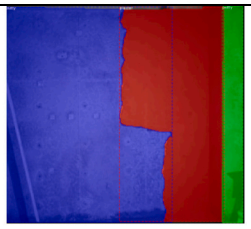
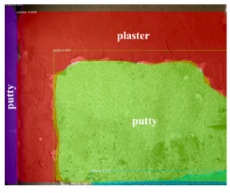
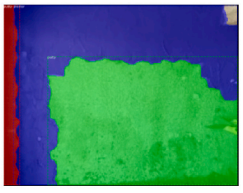
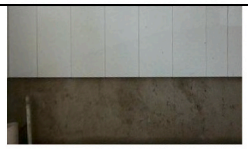
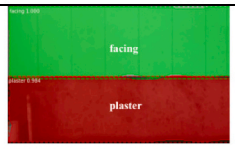
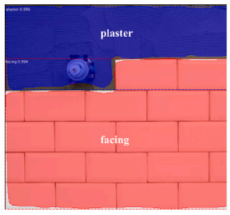
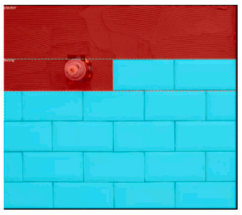
3.4.1. The component information is acquired through processing the Revit model

We set the origin of the project in the Revit model as the original position of the positional tracking algorithm. Then the Revit plug-in Lookup was installed to obtain the interior wall ID and the endpoints' coordinates of the wall positioning line {"L_i":[(z_{i1},x_{i1}),(z_{i2},x_{i2})]} to establish the wall equation (as shown in Fig. 5a).

3.4.2. The Revit component ID corresponding to the image is obtained based on the positional tracking algorithm

We drew a ray based on the coordinates of the shooting point and the camera pose (z, x, θ). According to the geometric relationship between the ray equation and the wall equation, we could find all the walls that intersect with the ray and then determine the wall ID with the shortest

Table 2
Images segmentation results in three stages.

Stage	Origin image	Perspective transformation	Image segmentation	Ground truth	Mask distribution
Block					Up and down distribution
					Left and right distribution
Plaster					Left and right and mosaic distribution
					Enclosing distribution
Facing					Up and down distribution
					Up and down and mosaic distribution

distance between its intersection point and (z, x). The algorithm principle is shown in Fig. 7. It is worth noticing that we took (z, x) as the origin of the coordinates. If one of the points (dieta_z_{i1}, dieta_x_{i1}) or (dieta_z_{i2}, dieta_x_{i2}) is in the same quadrant with the ray equation and meanwhile the two endpoints of the wall are on different sides of the ray, then the ray is considered to intersect the wall, that is "dieta_z_{kg} * cos (ray_angle) >= 0 || dieta_x_{kg} * sin (ray_angle) >= 0 || f (z_{k1},x_{k1}) f (z_{k2}, x_{k2}) <= 0" (Fig. 7).

3.4.3. The construction progress is determined by image segmentation

We used the Mask R-CNN algorithm to segment images taken on the construction site to obtain the indoor construction phase and progress percentage of each wall and then calculated a certain floor's overall construction progress.

3.4.4. The progress information is integrated into BIM

Based on the wall ID obtained from step 2, we used a Revit API to traverse all the BIM components and imported the progress information (obtained from step 3) into the corresponding component properties. Finally, we wrote a color-coding algorithm to modify the component color automatically according to its progress information—completed (green), under construction (yellow), and no construction (red).

3.5. Data collection and processing

In this study, we took Building #10 at the Pujiang Community Project in Shanghai as a case study (Fig. 8) which is a 16-storey shear wall structure—15 floors above ground and 1 floor underground. At the time of data collection, this building was under construction—the completed height was 47.10 m and the completed gross floor area was 9163.64 m^2.

For data collection, we used a Zed 2 binocular depth-sensing stereo camera produced by Stereo labs in San Francisco, the USA. The camera has relatively high accuracy at a low price compared to other binocular

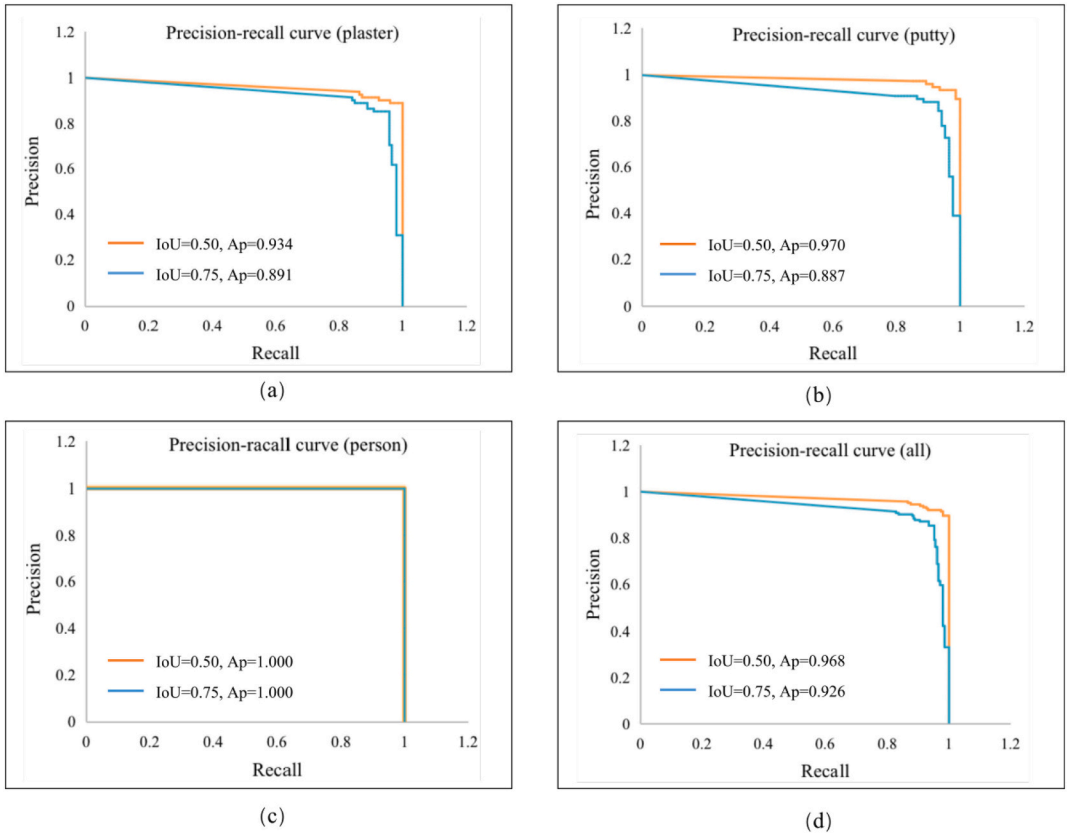


Fig. 14. Precision-Recall curve: (a) plaster; (b) putty; (c) person; (d) all types.

cameras on the market, and it supports open-source algorithms and secondary development. The camera with a size of 175 × 30 × 33 mm (6.89 × 1.18 × 1.3″) can reach the depth range of 0.2–20 m (0.65–65 ft), and the depth field of view can reach a maximum of 110° (H) x 70° (V) x 120° (D). The output resolution side by side is: 2× (2208 × 1242) @15fps, 2× (1920 × 1080) @30fps, 2× (1280 × 720) @60fps, and 2× (672 × 376) @100fps. The data collection process and camera parameters are shown in Fig. 9.

This research focuses on the plastering phase, and the pictures were collected from public video clips online and the actual construction sites, adding up to a dataset of 738 pictures. The dataset labeled by Labelme was divided into a training set (500 pictures), a validation set (160 pictures), and a test set (78 pictures) following a ratio of 7:2:1. Mask R-CNN applies a transfer learning on the MS-COCO dataset, and the parameter settings were batch_size = 4, backbone = resnet101, and epoch = 120. We improved the loss function, optimizer, and cosine annealing used in the Mask R-CNN algorithm (details in Section 6) to increase the segmentation accuracy. The environment configuration was tensorflow_gpu 1.5.0 + CUDA 9.0 + cudnn 7.0 + keras 2.1.0 + numpy 1.16.0. In the data collection process, the movement path of the camera was clockwise from the origin along the wall. The data collection path (Fig. 10) can be obtained through the following process. First, point A was set as both the starting point in the positional tracking algorithm and the origin coordinate of the BIM model. Second, we selected two points that the camera had to pass through on the floor plan, that is, B and C, whose true coordinates were known from the BIM model. Third, we moved the camera and obtained the predicted coordinates of all points along the movement path using the positional tracking algorithm. Among these points, two points whose predicted coordinates are closest to the true coordinates of B and C were selected as the corresponding B′ and C′. Finally, the data collection diagram can be obtained by superimposing A, B′, and C′ on the movement path to the corresponding points A, B, and C on the BIM floor plan.

4. Results

4.1. Calculation of progress percentage

The case building was in the indoor plastering phase at the time of data collection. After implementing the proposed indoor progress automated evaluation framework, we identified three typical types of progress evaluation results of the plastering construction phase, as shown in Fig. 11. Wall 1 and Wall 2 were under construction (during plastering), and the progress percentage was 65% and 60%, respectively, indicated in yellow. Wall 3 has not started plastering, and the progress percentage was 0%, marked in red. Wall 4 has finished plastering, and the progress percentage was 100%, indicated in green.

Summarizing the progress information of all the walls except for the stairwell on the 12th floor of Building # 10, there were 101 walls in total, and each wall had two working surfaces. Type A wall is located at the edge of the building and only the interior surface needs to be plastered. Therefore, the Type A wall only records one working surface (8 working surfaces in total). Type B wall has both surfaces indoors, and each wall has two working surfaces needing to be plastered (186 working surfaces in total). According to the above rules, a total of 194 working surfaces of 101 walls need plastering progress calculation. Some calculation parameters and results are shown in Appendix A. The coordinates and angles of the shooting points can be obtained from the Zed 2 positional tracking algorithm. The Revit property panel can retrieve the wall area, and the Revit Lookup-Location Curve function can check the two ends coordinates of the wall. The calculation results show that the total wall area that needs progress evaluation is 1223.505 m^2 and the completed construction area is 800.662m^2, resulting in a plastering progress percentage of the entire floor to be 65%.

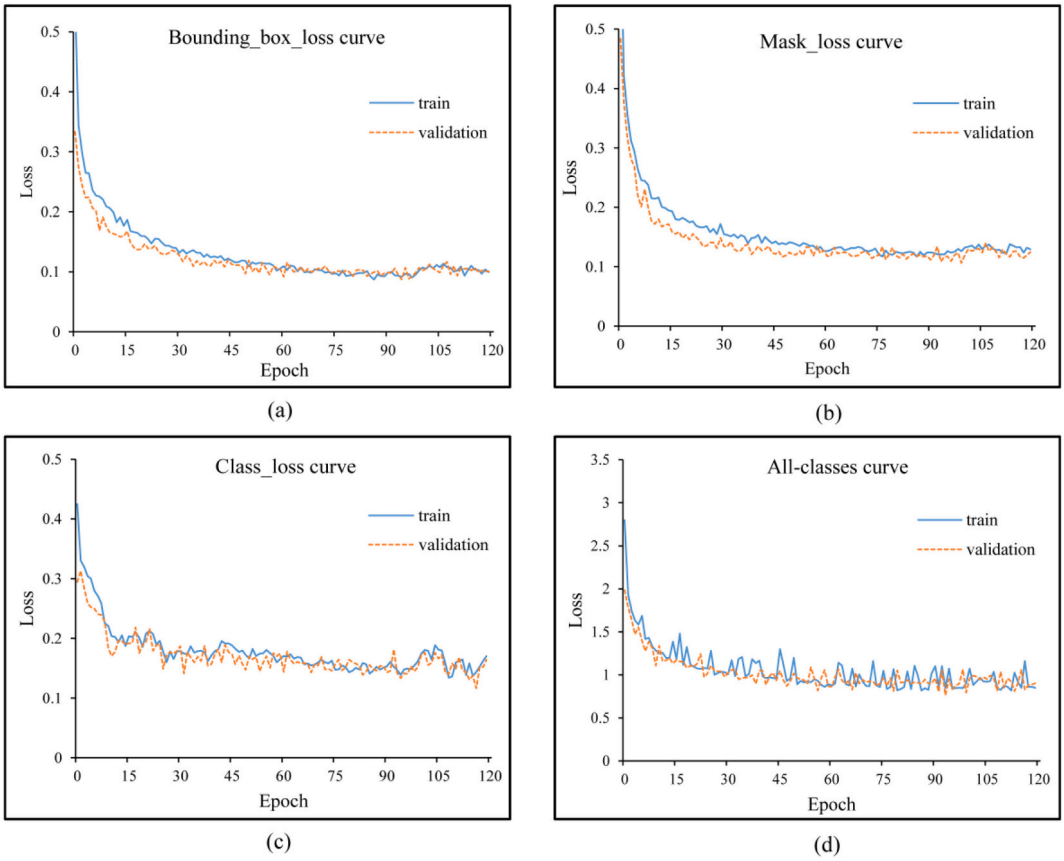


Fig. 15. Loss curve of training and validation process: (a) bounding_box_loss curve; (b) mask_loss curve; (c) class_loss curve; (d) all_classes curve.

4.2. Visualization

We used the Revit API secondary development program to traverse the wall components in the Revit model according to the wall ID and then automatically imported the progress information into the component properties. Then the interior wall construction progress information (such as "under construction, 60%") could be displayed in the component property bar. The component's color was automatically changed to red, yellow, and green according to the construction progress as shown in Fig. 12. After all the component progress information was imported, we got a visual display of the overall progress of the 12th standard floor of the building, as shown in Fig. 13.

Since the studied project was in the indoor plastering phase, the bricklaying process has been completed, and it would be a long time before the facing brick phase starts. This study only established the dataset of the plastering phase and showed the progress assessment results of the plastering construction. But the progress assessment of bricklaying and facing brick construction follows the same framework. We collected several pictures of bricklaying and facing brick construction on the Internet and tested them using the improved Mask R-CNN algorithm. The test results in Table 2 show that our algorithm achieved accurate detection and segmentation for various forms of mask distribution, such as up and down, left and right, mosaic, enclosing, etc.

5. Validation

5.1. Accuracy of mask R-CNN

Mean Average Precision (mAP) and Precision-Recall curve (P-R curve) of the test set are normally used to evaluate the application effect of Mask R-CNN. The precision and recall rates represent the model's performance to detect positive samples accurately and cover all positive samples, respectively. The Average Precision (AP) rate which is numerically equivalent to the area enclosed by the P-R curve, represents the overall detection effect of the model. Intersection Over Union (IoU) is the ratio of the overlap area to the union area of predict results and ground truth. In the tasks of segmentation, IoU is usually used to define positive/negative samples. For example, a sample would be defined as a positive sample when its IoU is higher than the IoU threshold [42,43].

Fig. 14 shows the P-R curves of plaster, putty, person, and all types with AP and mAP values in different IoU thresholds (IoU = 0.5 and 0.75) in the test set of 78 images using the improved Mask R-CNN algorithm. The AP of person reached 100% because of the transfer learning on the MS-COCO dataset which includes enormous samples of people. While APs for putty and plaster were 97.0% (IoU = 0.5) and 93.4% (IoU = 0.5). The mAPs of all types were 96.8% (IoU = 0.5) and 92.6% (IoU = 0.75), which are sufficient for indoor progress evaluation.

The loss curves of the training set and the validation set are shown in Fig. 15. The model was run for 120 epochs, and each epoch includes 125 iterations. In the initial epochs, the class-loss and mask-loss fluctuated dramatically. This problem is probably due to the small batch_size, resulting in fewer pictures in each iteration, and a few pictures with a high loss would dominate the overall loss. As the iterations increased, the loss became stable and converged in the end with a loss of 0.9 for the validation set.

5.2. Accuracy of positional tracking

We evaluated the positional tracking algorithm by setting calibration points. As shown in Fig. 16, the hand-held Zed 2 camera started from the origin, and we selected three calibration points with known coordinates in the room. During the data collection, three calibration points were passed in sequence, and the camera returned to the origin in the end, forming a closed loop. The selection of the calibration point position is

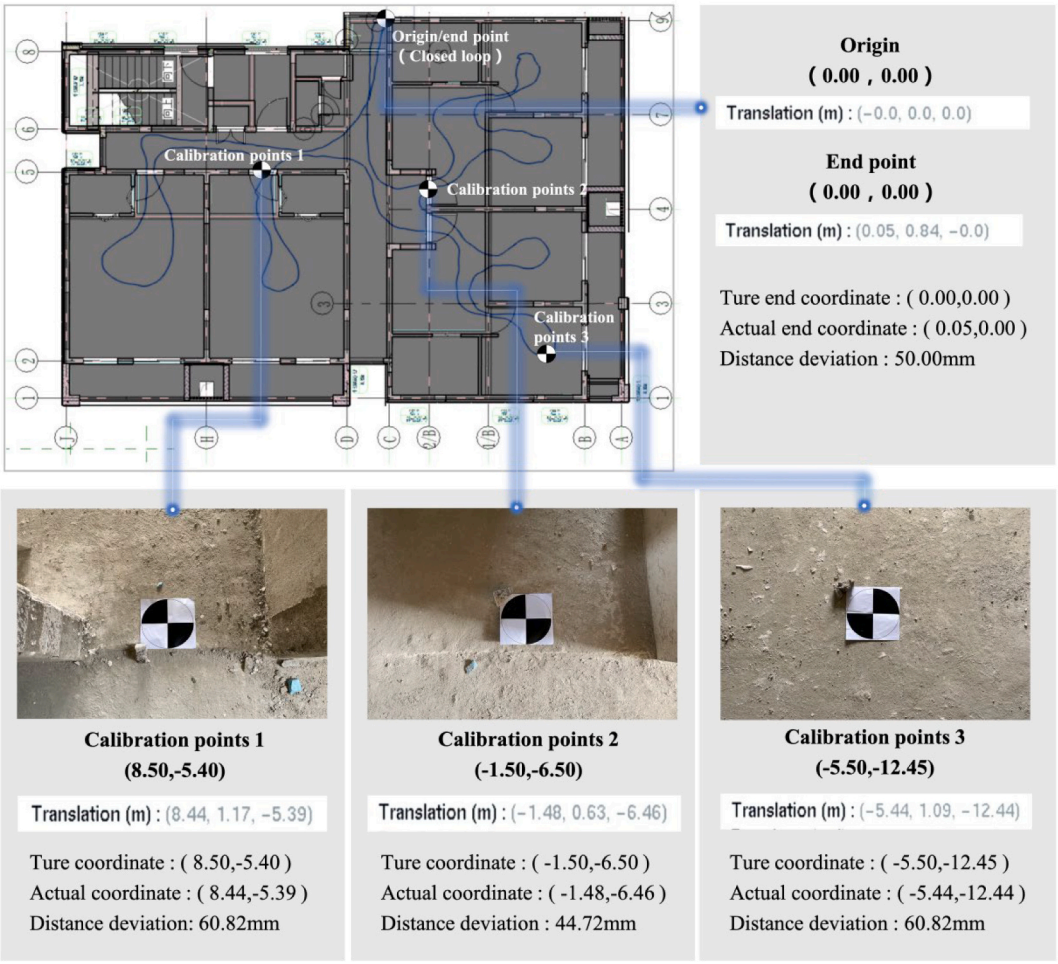


Fig. 16. Evaluation of the positional tracking algorithm. True coordinate: the known coordinates of the calibration point; actual coordinate: the coordinates obtained by positional tracking algorithm.

Table 3
Results of loss function tests.

Test	L_{mask}	IoU	AP -plaster	AP -putty	AP -person	mAP	Imp-AP -plaster	Imp-AP -putty	Imp-AP -person	Imp-mAP
Test1	CE Loss	0.5	87.5%	98.1%	100%	95.2%	Baseline			
		0.75	71.4%	87.7%	100%	86.4%	Baseline			
Test2	Dice Loss	0.5	95.0%	95.4%	100%	96.8%	7.5%↑	2.7%↓	0%	1.6%↑
		0.75	70.5%	76.6%	100%	82.4%	0.9%↓	11.1%↓	0%	4.0%↓
Test3	CE Loss +Dice Loss	0.5	93.3%	98.3%	100%	97.2%	5.8%↑	0.2%↑	0%	2.0%↑
		0.75	77.1%	87.8%	100%	88.3%	5.7%↑	0.1%↑	0%	1.9%↑

Note: mAP in the table is the highest among the mAPs in all the epochs.

based on the principle of uniform distribution and easy access to actual coordinate positions. Calibration points 1 and 2 are in the middle of the door frames, and calibration point 3 is in the center of the room. We calculated 1) the distance deviation between the known coordinates of the calibration point and the coordinates obtained by the positional tracking algorithm and 2) the distance between the origin and the endpoint (which should be zero), resulting in a maximum distance deviation of 60.82 mm. The positional tracking algorithm is used to determine which wall the photo represents by calculating the minimum distance between the camera and wall. In practice, the camera is usually placed over 3–4 m away from the wall to ensure complete coverage of the wall. A distance deviation of 60 mm, which is about 2% of 3 m, would not influence the mapping of components to BIM.

6. Discussion

6.1. Loss function improvement

The loss function of the mask which is used to evaluate and optimize the segmentation results of the mask is an essential factor influencing the segmentation accuracy. CE Loss (Test 1), the default loss function of Mask R-CNN, is suitable for detecting objects with the same level of difficulty because CE Loss considers the contribution of each pixel equally. When the difficulty level of some detected objects is high (complex samples), e.g., plaster accounts for 50–80% pixels and has an irregular shape, Dice Loss (Test 2) is a more effective function that can strengthen the training of complex samples [41]. However, for particularly complex situations, the contribution of each pixel also needs to be

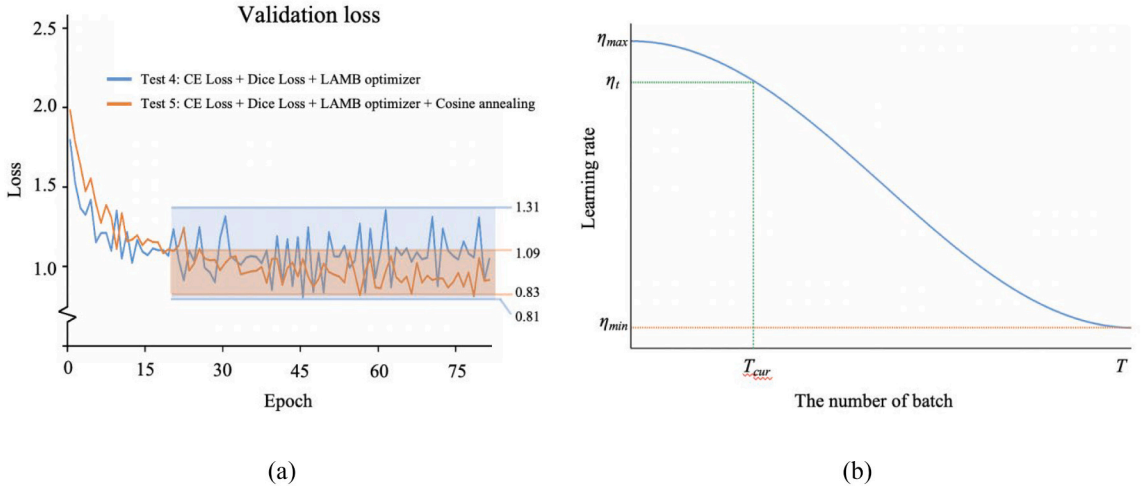


Fig. 17. Validation loss and cosine annealing function: (a) validation loss of tests 4 and 5; (b) cosine annealing function. η_{min} and η_{max} are the upper and lower limits of the learning rate, T is the total number of batches in the training process, and T_{cur} is the number of batches that have been iterated.

considered while considering the unbalanced level of detection difficulty (Test 3). We designed three loss function tests and the results are shown in Table 3.

Test 2 (L_{mask} = Dice Loss) resulted in an AP (IoU = 0.5) of 95.0% for plaster, which was 7.5% higher than the AP in Test 1. The APs (IoU = 0.5) of plaster and putty in Test 2 were roughly equal, showing that the problem of unbalanced level of detection difficulty can be solved. However, the mAP (IoU = 0.75) declined compared to Test 1. The reason is that Dice Loss weakens the training of easy samples (for instance, the AP of putty in Test 2 is lower than the baseline) while strengthening the training of complex samples, resulting in the reduction of mAP. In Test 3, we used Dice Loss plus CE Loss as L_{mask}, leading to the increased APs of all classes.

Imp-AP is the increased AP value compared with the baseline. Imp-mAP is the increased mAP value compared with the baseline.

6.2. Hyperparameter optimization

Optimizers, such as Stochastic Gradient Descent (SGD) and Layer-wise Adaptive Moments optimizer for Batching training (LAMB), are important factors in hyperparameter optimization, which can adjust and update the parameters of the loss function to make the loss value approach the global minimum. SGD optimizer, the default optimizer of Mask R-CNN, can quickly make the loss converge to an approximate local minimum [44] by updating the network parameters with a fixed learning rate after a small batch of training data. However, the fixed learning rate may not be the optimal value, resulting in a low accuracy, so we need to adaptively adjust the optimal learning rate to ensure that each layer of the network effectively converges [45] in our dataset. Therefore, we used the LAMB optimizer (Test 4) which can individually optimize and modify the learning rate of each layer in the neural network through a layer-wise strategy to obtain the optimal learning rate [46]. The results are shown in Table 5. When we used the optimizer with a high global learning rate, the loss function was prone to oscillation. This problem can be solved by a cosine annealing schedule [47] (Test 5) which can make the loss stable by reducing the learning rate smoothly (Fig. 17b).

In Test 4, the algorithm parameters were batch size = 4, learning rate = 5e-4, and weight_decay = 0.02 (official recommendation). The detection accuracy was improved and the mAPs were 96.8% (IoU = 0.5) and 91.2% (IoU = 0.75). But in the later epochs, the loss function of the validation dataset oscillated obviously, as shown in Fig. 17a. The reason is that LAMB with a high learning rate makes the loss function diverge due to some samples with higher errors. Under the effect of the cosine annealing schedule in Test 5, the oscillation amplitude of the loss function was significantly reduced in the later epochs (Fig. 17a). The mAPs were 97.2% (IoU = 0.5) and 92.2% (IoU = 0.75) which were 2% and 5.8% higher than the baseline.

6.3. Image brightness equalization

As the brightness of construction images is often uneven due to rain, occlusion, etc., and the plaster color changes naturally over time, these situations may lead to a reduction in segmentation accuracy, as shown in Table 4. The image brightness equalization algorithm can effectively solve the above problems by reducing the brightness difference of the images. The algorithm will first perform pixel difference and convolution on the RGB image with uneven brightness to establish a pixel matrix "A" of the same dimension as the original image. Then, the pixel values of the low-brightness and high-brightness areas of matrix "A" are taken as positive and negative values, respectively. The pixel values of the remaining positions are taken as zero. Finally, the matrix "A" and the original image matrix are added to balance the image brightness.

On top of the parameter setting of Test 5, we applied the image brightness equalization algorithm to process the test set of 78 pictures. The images after brightness equalization and their predicted results are shown in Table 4. The algorithm improved the segmentation effect significantly compared with the predicted results of original images from three perspectives: incomplete object recognition, incomplete mask recognition, and overlapping masks. The mAP (IoU = 0.75) of Test 6 achieved the highest value of 92.6% among all the tests, which was 0.4% higher than Test 5 and 6.2% higher than the baseline (Table 5).

6.4. Comparison with other algorithms

The performance of the improved Mask R-CNN is compared with four outstanding segmentation algorithms, i.e., Mask R-CNN [38] with the R-101-FPN backbone, UNet++ [48], YOLACT [49] with the R-101-FPN backbone, and PointRend [50] with the backbone of R-101-FPN. To ensure a fair comparison, five algorithms were evaluated with the same plaster dataset that was divided into a training set (500 pictures), a validation set (160 pictures), and a test set (78 pictures) following a ratio of 7:2:1. Besides, all algorithms were trained using images with short sides of 800 pixels and the parameter settings were batch_size = 4 and epoch = 120. The comparison results are shown in Table 6. For the overall evaluation of the model, improved Mask R-CNN outperformed other algorithms with the highest mAP, AP- person (IoU = 0.75), and AP - putty (IoU = 0.75). For the plaster detection, both improved Mask R-

Table 4
Image brightness equalization examples.

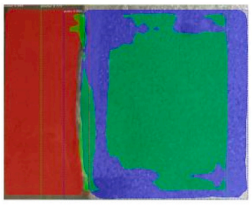
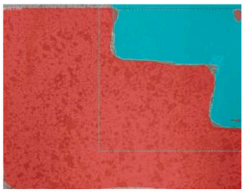
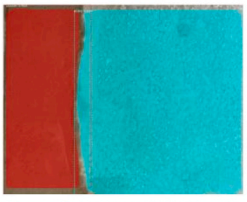
	Example 1	Example 2	Example 3
Original images			
Predicted results of original images			
Brightness equalization			
Predicted results of brightness equalization			
Ground truth			
Solved problems	Incomplete object recognition	Incomplete mask recognition	Overlapping masks

Table 5
Results of six tests of Mask R-CNN algorithm improvement.

Test	Content	mAP_{50}	mAP_{75}	Imp_{50}	Imp_{75}
1	CE Loss + SGD optimizer	95.2%	86.4%	baseline	baseline
2	Dice Loss + SGD optimizer	96.8%	82.4%	1.6%↑	4.0%↓
3	CE Loss+Dice Loss + SGD optimizer	97.2%	88.3%	2.0%↑	2.9%↑
4	CE Loss+Dice Loss + LAMB optimizer	96.8%	91.2%	1.6%↑	4.8%↑
5	CE Loss + Dice Loss + LAMB optimizer+Cosine annealing	97.2%	92.2%	2.0%↑	5.8%↑
6	CE Loss + Dice Loss + LAMB optimizer + Cosine annealing + Image brightness equalization	96.8%	92.6% (highest)	1.6%↑	6.2%↑

Note:mAP_{50} is the mean average precision value when IoU = 0.5, mAP_{75} is the mean average precision value when IoU = 0.75, Imp_{50} is the increased mAP_{50} compared with the baseline, and Imp_{75} is the increased mAP_{75} compared with the baseline.

CNN and PointRend ranked top two among all algorithms.

7. Conclusions

Traditional construction progress management is mainly carried out through manual inspection and filling out the progress reports. The low level of automation makes it difficult to obtain and update project progress in real-time. The indoor construction progress management is especially time-consuming and costly due to the chaotic indoor construction sites. This paper described a framework based on a computer vision algorithm and BIM to automatically evaluate indoor construction progress in real-time to avoid time lags in progress monitoring and reduce labor-intensive activities. The framework involves two steps. First, the Mask R-CNN algorithm was applied to segment images to determine the construction stage and the quantitative progress. Then, a SLAM-based photo recognition method was proposed to integrate progress information into BIM. The framework was successfully applied to a construction project in Shanghai where the mean Average Precision of plaster segmentation reached 95.3%.

Table 6
Comparison results of different models on the plaster dataset.

Test	Model	Backbone	Style	IoU	AP - plaster	AP- putty	AP- person	mAP	Imp-mAP
1	Mask R-CNN (ICCV'2017)	R-101-FPN	Tensorflow	0.5	87.5%	98.1%	100%	95.2%	Baseline
				0.75	71.4%	87.7%	100%	86.4%	Baseline
2	UNet++ (CVPR'2018)	–	PyTorch	0.5	88.3%	89.6%	100%	92.6%	2.6%↓
				0.75	79.2%	81.8%	100%	87.0%	0.6%↑
3	YOLACT (ICCV'2019)	R-101-FPN	PyTorch	0.5	94.1%	94.4%	96.0%	94.8%	0.4%↓
				0.75	84.2%	76.0%	79.2%	79.8%	6.6%↓
4	PointRend (CVPR'2020)	R-50-FPN	Caffe	0.5	96.1%	94.7%	96.0%	95.6%	0.4%↑
				0.75	**91.3%**	84.3%	96.0%	90.6%	4.2%↑
5	Improved Mask R-CNN	R-101-FPN	Tensorflow	0.5	93.4%	97.0%	100%	96.8%	1.6%↑
				0.75	89.1%	**88.7%**	**100%**	**92.6%**	**6.2%↑**

Note: Bold numbers represent the highest value (IoU = 0.75) in each column.

This study contributes to the body of knowledge from the following two aspects. (1)To automate the wall progress evaluation for the cast-in-place construction, we established an evaluation framework that enables us to precisely recognize wall progress in a whole layer with streamlined BIM visualization. (2) Different from the current method which utilized edge detection for construction area calculation, we proposed an image segmentation method to quantitatively calculate wall construction area and percentile progress based on the improved Mask R-CNN in three aspects: the loss function, optimizer, and cosine annealing. This study not only provides a theoretical reference for the automated progress tracking of indoor construction but also contributes to promoting a feasible approach that connects both physical activities and virtual models for digital twin construction. The proposed method in this study can be utilized for automated construction progress tracking management and potentially facilitate the unmanned inspection process with tremendous time and effort savings reduced from the exhaustive manual data processing. Besides, the approach could also contribute to the progress payment management by linking the completed progress with the unit-price of a project to calculate construction cost occurred, enabling project managers to control and avoid cost overruns when the actual schedule deviates from the plan. Furthermore, this study also demonstrated a straightforward method that integrates the evaluation results into digital models displayed in the cloud for progress visualization, making the stakeholders intuitively understand construction progress and efficiently recommend reasonable decisions.

Nevertheless, there are some limitations in this study despite its contributions. First, this study only collected data of plastering construction. Future research could expand the dataset of bricklaying, facing brick, and wall finishing for multiple scenarios of progress evaluation. Second, the path sequence was not considered during image collection and path optimization with the highest efficiency is supposed to be carried out in future works.

Declaration of Competing Interest

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

Acknowledgment

The authors are financially sponsored by the National Natural Science Foundation of China (52078374, 52108090), Fundamental Research Funds for the Central Universities (22120210288), the Peak Discipline Construction Project of Shanghai (No. 2021-CE-06), and the Shanghai Pujiang Program (2020PJD074).

Appendix A. Information statistics and progress calculation results

No.	Shooting point coordinates(z,x,θ)	Wall id	Endpoints' coordinates of wall (z,x) (m)	Area of one surface (m^2)	Progress percentage of one surface	Construction area of one surface (m^2)	Progress of the entire floor(%)
1	−12.41,-1.93,-0.26(−15)	1,461,080	(−13.40,-6.90) (−13.40,-3.30)	7.830	No construction (0%)	0	
2	−11.54,-5.22,0.35(20)	1,461,085	(−13.36,-3.30) (−13.36,0.00)	7.830	No construction(0%)	0	
3	−5.64,-5.03,0.19(11)	1,461,110	(−6.60,-6.90) (−6.60,-3.50)	7.830	Completed(100%)	7.830	
4	−9.43,-5.78,2.95(169)	1,461,110	(−6.60,-6.90) (−6.60,-3.50)	7.830	Completed(100%)	7.830	
5	−5.04,-2.23, −0.10(−6)	1,461,114	(−6.60,-3.50) (−6.60,0.00)	7.830	Completed(100%)	7.830	
6	−10.13,-2.37,-3.11 (−178)	1,461,114	(−6.60,-3.50) (−6.60,0.00)	7.830	Completed(100%)	7.830	65%
7	−2.41,-5.24, −0.10(−6)	1,461,292	(−3.30,-6.90) (−3.30,-3.50)	7.830	Under construction (75%)	5.873	
8	−5.95,-5.41, 3.12(179)	1,461,292	(−3.30,-6.90) (−3.30,-3.50)	7.830	Under construction (32%)	2.506	
9	3.62,-2.76, 1.69(97)	1,461,255	(6.90,-3.50) (6.90,-1.60)	4.440	Completed(100%)	4.440	
10	10.44,-2.59, 0.16(9)	1,461,255	(6.90,-3.50) (6.90,-1.60)	4.440	Completed(100%)	4.440	
11	16.80,-2.68, −3.44(−197)	1,461,232	(20.10,-3.50) (20.10,-1.60)	7.440	Completed(100%)	7.440	
12		1,461,132		7.440	Completed(100%)	7.440	

(*continued on next page*)

(*continued*)

No.	Shooting point coordinates(z,x,θ)	Wall id	Endpoints' coordinates of wall (z,x) (m)	Area of one surface (m^2)	Progress percentage of one surface	Construction area of one surface (m^2)	Progress of the entire floor(%)
	23.44,-2.51, −0.26(−15)		(20.10,-3.50) (20.10–1.60)				
13	7.44,-5.79, −3.33(−191)	1,461,245	(10.20,-6.90) (10.20,-3.50)	7.830	Completed(100%)	7.830	
14	13.21,-6.14, 3.19(183)	1,461,245	(10.20,-6.90) (10.20,-3.50)	7.830	Completed(100%)	7.830	
15	−6.45,1.43, −0.65(−37)	1,461,144	(−5.30,1.30) (−12.10,1.30)	13.220	Completed(100%)	13.220	
16	−10.24,5.57, 1.52(87)	1,461,144	(−5.30,1.30) (−12.10,1.30)	13.220	Completed(100%)	13.220	
17	−10.20,2.35, −1.48(−85)	1,461,289	(−5.30,6.25) (−12.10,6.25)	13.220	Completed(100%)	13.220	
18	−10.20,10.35,1.64(94)	1,461,189	(−5.30,6.25) (−12.10,6.25)	13.220	Completed(100%)	13.220	
19	0.89,-5.26, −1.62(−93)	1,461,177	(3.80,-6.90) (3.80,-3.30)	7.830	No construction(0%)	0	
20	5.84,-4,98, 0.05(3)	1,461,177	(3.80,-6.90) (3.80,-3.30)	7.830	No construction(0%)	0	
...							
Sum:				1223.505		800.662	

Note: The construction area of one surface = The area of one surface * construction progress percentage of one surface. For example, Construction area of the No.1 surface = 7.830 m^3 * 0% = 0 m^3.
The total area = the sum area of all the surfaces.
The total construction area = the sum area of all the surfaces that has completed construction.
Construction progress of the entire floor = The total construction area divided by the total area.

References

[1] M. Golparvar-Fard, F. Pena-Mora, S. Savarese, Automated progress monitoring using unordered daily construction photographs and IFC-based building information models, J. Comput. Civ. Eng. 29 (1) (2015) 04014025, https://doi.org/10.1061/(asce)cp.1943-5487.0000205.

[2] M. Abdelsayed, R. Navon, An information sharing, internet-based, system for project control, Civ. Eng. Syst. 16 (3) (1999) 211–233, https://doi.org/10.1080/02630259908970263.

[3] B. Ekanayake, J.K.-W. Wong, A.A.F. Fini, P. Smith, Computer vision-based interior construction progress monitoring: a literature review and future research directions, Autom. Constr. 127 (2021), 103705, https://doi.org/10.1016/j.autcon.2021.103705.

[4] C.Z. Li, F. Xue, X. Li, J. Hong, G.Q. Shen, An internet of things-enabled BIM platform for on-site assembly services in prefabricated construction, Autom. Constr. 89 (2018) 146–161, https://doi.org/10.1016/j.autcon.2018.01.001.

[5] F. Bosché, M. Ahmed, Y. Turkan, C.T. Haas, R. Haas, The value of integrating scan-to-BIM and scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: the case of cylindrical MEP components, Autom. Constr. 49 (2015) 201–213, https://doi.org/10.1016/j.autcon.2014.05.014.

[6] A. Braun, S. Tuttas, A. Borrmann, U. Stilla, Improving progress monitoring by fusing point clouds, semantic data and computer vision, Autom. Constr. 116 (2020), 103210, https://doi.org/10.1016/j.autcon.2020.103210.

[7] M. Kopsida, I. Brilakis, Real-time volume-to-plane comparison for mixed reality–based progress monitoring, J. Comput. Civ. Eng. 34 (4) (2020) 04020016, https://doi.org/10.1061/(asce)cp.1943-5487.0000896.

[8] K. Asadi, H. Ramshankar, M. Noghabaei, K. Han, Real-time image localization and registration with BIM using perspective alignment for indoor monitoring of construction, J. Comput. Civ. Eng. 33 (5) (2019) 04019031, https://doi.org/10.1061/(asce)cp.1943-5487.0000847.

[9] G.A. Atkinson, W. Zhang, M.F. Hansen, M.L. Holloway, A.A. Napier, Image segmentation of underfloor scenes using a mask regions convolutional neural network with two-stage transfer learning, Autom. Constr. 113 (2020), 103118, https://doi.org/10.1016/j.autcon.2020.103118.

[10] P. Byvshev, P.-A. Truong, Y. Xiao, Image-based renovation progress inspection with deep siamese networks, in: Proceedings of the 2020 12th International Conference on Machine Learning and Computing, 2020, pp. 96–104, https://doi.org/10.1145/3383972.3384036.

[11] H. Hamledari, B. McCabe, S. Davari, Automated computer vision-based detection of components of under-construction indoor partitions, Autom. Constr. 74 (2017) 78–94, https://doi.org/10.1016/j.autcon.2016.11.009.

[12] K.K. Han, M. Golparvar-Fard, Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs, Autom. Constr. 53 (2015) 44–57, https://doi.org/10.1016/j.autcon.2015.02.007.

[13] H. Deng, H. Hong, D. Luo, Y. Deng, C. Su, Automatic indoor construction process monitoring for tiles based on BIM and computer vision, J. Constr. Eng. Manag. 146 (1) (2020) 04019095, https://doi.org/10.1061/(asce)co.1943-7862.0001744.

[14] C.H. Caldas, D.G. Torrent, C.T. Haas, Using global positioning system to improve materials-locating processes on industrial projects, J. Constr. Eng. Manag. 132 (7) (2006) 741–749, https://doi.org/10.1061/(asce)0733-9364(2006)132:7(741).

[15] A. Shahi, M. Safa, C.T. Haas, J.S. West, Data fusion process management for automated construction progress estimation, J. Comput. Civ. Eng. 29 (6) (2015) 04014098, https://doi.org/10.1061/(asce)cp.1943-5487.0000436.

[16] K. Song, S.N. Pollalis, F. Peña-Mora, Project dashboard: concurrent visual representation method of project metrics on 3D building models, Comp. Civil Eng. 2005 (2005) 1–12, https://doi.org/10.1061/40794(179)147.

[17] E. Ergen, B. Akinci, B. East, J. Kirby, Tracking components and maintenance history within a facility utilizing radio frequency identification technology, J. Comput. Civ. Eng. 21 (1) (2007) 11–20, https://doi.org/10.1061/(asce)0887-3801(2007)21:1(11).

[18] C. Kim, H. Kim, J. Ryu, C. Kim, Ubiquitous sensor network for construction material monitoring, J. Constr. Eng. Manag. 137 (2) (2011) 158–165, https://doi.org/10.1061/(asce)co.1943-7862.0000257.

[19] T.M. Lorenzo, B. Benedetta, C. Manuele, T. Davide, BIM and QR-code. A synergic application in construction site management, Proc. Eng. 85 (2014) 520–528, https://doi.org/10.1016/j.proeng.2014.10.579.

[20] A. Ibrahim, A. Sabet, M. Golparvar-Fard, BIM-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform, in: Proceedings of the European Conference on Computing in Construction, Crete, Greece, 2019, pp. 10–12, https://doi.org/10.35490/ec3.2019.195.

[21] Z. Wang, Q. Zhang, B. Yang, T. Wu, K. Lei, B. Zhang, T. Fang, Vision-based framework for automatic progress monitoring of precast walls by using surveillance videos during the construction phase, J. Comput. Civ. Eng. 35 (1) (2021) 04020056, https://doi.org/10.1061/(asce)cp.1943-5487.0000933.

[22] S. Kim, S. Kim, D.-E. Lee, Sustainable application of hybrid point cloud and BIM method for tracking construction progress, Sustainability 12 (10) (2020) 4106, https://doi.org/10.3390/su12104106.

[23] R. Maalek, D.D. Lichti, J.Y. Ruwanpura, Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction, Remote Sens. 11 (9) (2019) 1102, https://doi.org/10.3390/rs11091102.

[24] H. Son, C. Kim, Y. Kwon Cho, Automated schedule updates using as-built data and a 4D building information model, J. Manag. Eng. 33 (4) (2017), 04017012, https://doi.org/10.1061/(asce)me.1943-5479.0000528.

[25] Z. Pučko, N. Šuman, D. Rebolj, Automated continuous construction progress monitoring using multiple workplace real time 3D scans, Adv. Eng. Inform. 38 (2018) 27–40, https://doi.org/10.1016/j.aei.2018.06.001.

[26] A. Shahi, J.M. Cardona, C.T. Haas, J.S. West, G.L. Caldwell, Activity-based data fusion for automated progress tracking of construction projects, in: Construction Research Congress 2012: Construction Challenges in a Flat World, 2012, pp. 838–847, https://doi.org/10.1061/9780784412329.085.

[27] S. Pu, G. Vosselman, Extracting windows from terrestrial laser scanning, Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 36 (2007) 12–14, in: https://www.isprs.org/proceedings/xxxvi/3-W52/ls2007.pdf.

[28] H. Mahami, N. Ghassemi, M.T. Darbandy, A. Shoeibi, S. Hussain, F. Nasirzadeh, R. Alizadehsani, D. Nahavandi, A. Khosravi, S. Nahavandi, Material Recognition

for Automated Progress Monitoring using Deep Learning Methods, arXiv preprint arXiv:2006.16344, 2020, https://doi.org/10.48550/arXiv.2006.16344.
[29] W.S. Alaloul, A.H. Qureshi, Material classification via machine learning techniques: construction projects progress monitoring, artificial neural networks and deep learning-applications and perspective, IntechOpen (2021), https://doi.org/10.5772/intechopen.96354.
[30] A. Dimitrov, M. Golparvar-Fard, Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections, Adv. Eng. Inform. 28 (1) (2014) 37–49, https://doi.org/10.1016/j.aei.2013.11.002.
[31] J. Park, J. Chen, Y.K. Cho, Point Cloud Information Modeling (PCIM): An Innovative Framework for As-Is Information Modeling of Construction Sites, Construction Research Congress 2020: Computer applications, American Society of Civil Engineers Reston, VA, 2020, pp. 1319–1326, https://doi.org/10.1061/9780784482865.139.
[32] L. Hui, M.-W. Park, I. Brilakis, Automated brick counting for façade construction progress estimation, J. Comput. Civ. Eng. 29 (6) (2015) 04014091, https://doi.org/10.1061/9780784413616.119.
[33] C. Kropp, C. Koch, M. König, Drywall state detection in image data for automatic indoor progress monitoring, Comp. Civil Build. Eng. 2014 (2014) 347–354, https://doi.org/10.1061/9780784413616.044.
[34] P. Martinez, B. Barkokebas, F. Hamzeh, M. Al-Hussein, R. Ahmad, A vision-based approach for automatic progress tracking of floor paneling in offsite construction facilities, Autom. Constr. 125 (2021), 103620, https://doi.org/10.1016/j.autcon.2021.103620.
[35] K. Asadi, K. Han, Real-time image-to-BIM registration using perspective alignment for automated construction monitoring, Const. Res. Congress 2018 (2018) 388–397, https://doi.org/10.1061/9780784481264.038.
[36] P.O. Pinheiro, R. Collobert, P. Dollár, Learning to Segment Object Candidates, arXiv Preprint. arXiv:1506.06204, 2015.
[37] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2017) 2359–2367, https://doi.org/10.1109/cvpr.2017.472.
[38] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969, https://doi.org/10.48550/arXiv.1703.06870.
[39] Y. Qiao, M. Truman, S. Sukkarieh, Cattle segmentation and contour extraction based on mask R-CNN for precision livestock farming, Comput. Electron. Agric. 165 (2019), 104958, https://doi.org/10.1016/j.compag.2019.104958.
[40] Q. Zhang, X. Chang, S.B. Bian, Vehicle-damage-detection segmentation algorithm based on improved mask RCNN, IEEE Access 8 (2020) 6997–7004, https://doi.org/10.1109/access.2020.2964055.
[41] F. Milletari, N. Navab, S. Ahmadi, V-net, fully convolutional neural networks for volumetric medical image segmentation, in: Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), 2016, pp. 565–571, https://doi.org/10.1109/3dv.2016.79.
[42] M.A. Rahman, Y. Wang, Optimizing intersection-over-union in deep neural networks for image segmentation, in: International Symposium on Visual Computing, Springer, 2016, pp. 234–244, https://doi.org/10.1007/978-3-319-50835-1_22.
[43] L. Zhu, Z. Xie, L. Liu, B. Tao, W. Tao, Iou-uniform r-cnn: breaking through the limitations of rpn, Pattern Recogn. 112 (2021), 107816, https://doi.org/10.1016/j.patcog.2021.107816.
[44] Z. Allen-Zhu, Natasha 2: Faster non-convex optimization than sgd, in: Advances in Neural Information Processing Systems, 2018, pp. 2675–2686. ISBN 1510860967 9781510860964.
[45] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, K. Keutzer, Imagenet training in minutes, in: Proceedings of the 47th International Conference on Parallel Processing, 2018, pp. 1–10, https://doi.org/10.1145/3225058.3225069.
[46] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large Batch Optimization for Deep Learning: Training Bert in 76 Minutes, arXiv preprint arXiv:1904.00962, 2019, https://doi.org/10.48550/arXiv.1904.00962.
[47] I. Loshchilov, F. Hutter, Sgdr: Stochastic Gradient Descent with Warm Restarts, arXiv Preprint arXiv:1608.03983, 2016, https://doi.org/10.48550/arXiv.1608.03983.
[48] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer, 2018, pp. 3–11, https://doi.org/10.1007/978-3-030-00889-5_1.
[49] D. Bolya, C. Zhou, F. Xiao, Y.J. Lee, Yolact: real-time instance segmentation, Proc. IEEE/CVF Int. Conf. Comp. Vision (2019) 9157–9166, https://doi.org/10.1109/iccv.2019.00925.
[50] A. Kirillov, Y. Wu, K. He, R. Girshick, Pointrend: image segmentation as rendering, Proc. IEEE/CVF Conf. Comp. Vision Pattern Recognition (2020) 9799–9808, https://doi.org/10.1109/cvpr42600.2020.00982.