A Mask R-CNN Based Approach to Automatically Construct As-is IFC BIM Objects from Digital Images

H.Q. Ying[a] and S. Lee[a]

[a]Department of Civil Engineering, The University of Hong Kong, Hong Kong
E-mail: u3004315@connect.hku.hk, sanghoon.lee@hku.hk

Abstract –

Various image-based building object recognition approaches have been developed to create as-is Building Information Models (BIMs) of existing buildings. However, existing approaches generally rely on human-designed features to automatically or semi-automatically recognize building objects, which makes them sensitive to input images and difficult to extend to new building objects. Furthermore, when constructing object geometries, most of these approaches are limited to rectangular or pre-defined surface shapes. To address these limitations, this study presents a human-designed features-free, shape constraints-free and fully automatic approach to construct as-is BIM objects from images of a building. This approach adapts Mask R-CNN, a deep convolutional neural network, to automatically recognize and segment building objects with arbitrary shapes (i.e., surface boundary shapes) from images. The segmented objects, characterized by object types and pixel masks, are further geometrically fitted to construct surface geometries. Finally, the constructed building objects are defined in the Industry Foundation Classes (IFC) data format. Three types of building objects (i.e., walls, doors, and lifts) are used in this study. Total 430 images containing these objects collected from the interiors of 4 university buildings are used to train and test the Mask R-CNN model. The test results show that the trained model is accurate and robust to recognize and segment all the three types of building objects. Furthermore, the feasibility of the proposed approach is preliminarily validated by successfully extracting IFC building objects from an image.

Keywords –

As-is BIM object; IFC; Image-based modeling; Deep learning; Mask R-CNN

1 Introduction

As-is building information models (BIMs) are characterized by containing up-to-date building information that can be used to support effective operations and maintenance of existing buildings [1, 2]. In the current practice, creating such a BIM remains a laborious, time-consuming, and costly process [1, 3]. Recent studies have focused on developing automatic approaches to create as-is BIMs. Although considerable progress has been made by exploring approaches that consume 3D point clouds [4], image-based modelling approaches have received increasing attentions. Compared to point clouds-based approaches that usually rely on a laser scanner to collect the data, image-based approaches have significant advantages in the conveniences, efficiency and cost of as-is data collection [1, 5].

Image-based as-is BIM creation generally consists of four steps [1, 6, 7]: (1) data collection, to capture images of a building and/or corresponding "location" data; (2) object recognition and construction, to recognize building objects and extract their geometries from the images; (3) geometry merging, to align constructed building objects in a common coordinate system; and (4) semantic enrichment and as-is BIM generation, to add required semantic information and save the enriched model in a specific data format (e.g., IFC or gbXML). Among these steps, the second step plays a foremost and challenging role. Most of existing approaches in this step rely on carefully hand-crafted features to recognize building objects. The commonly used features include colors [8], textures [8], edges [9], shapes [8, 10, 11], and so on. Unfortunately, these appearance-related features could vary under different environments (e.g., lighting conditions, and camera positions and poses) and/or be affected by uninterested objects (e.g., decorations and small devices that commonly exist in building interiors) in the images. Thus, these approaches are often sensitive to input images and usually require a manual pre-processing to remove noises from the input images or make the features required in downstream detection processes to be more easily extractable. Furthermore, for every new building object, corresponding detection features need to be additionally designed, which largely limits the

scalability of these approaches. In addition, when constructing object geometries, most of these approaches are limited to rectangular and pre-defined (e.g., arch) shapes. As an effort to address the first limitation, Lu et al. [12] developed a neuro-fuzzy based system, which can robustly recognize five types of building objects (i.e., beams, columns, windows, doors, and walls) in complex environments with few appearance features. However, this system is also a hand-crafted features – based approach and does not address the latter two limitations. Moreover, the system is semi-automatic. To recognize objects in an image, considerable manual efforts are required, including drawing the ground line and the ceiling line first and then orderly clicking the corners of the target objects to be recognized.

To address the aforementioned limitations, this study aims to develop a fully automated (i.e., without manual pre-processing of input images and human intervention in the algorithmic process), scalable (i.e., human-designed features-free), and shape constraints-free approach to construct as-is IFC BIM objects from digital images of existing buildings. This approach is based on Mask R-CNN [13], a deep neural network, to recognize and segment building object instances from images. In the remainder of this paper, the principles of Mask R-CNN are introduced first. Then the proposed approach is explained. Next, the implementation of the approach as well as a simple experiment is detailed, followed by the conclusion and discussion of future work.

2 Mask R-CNN for Instance Segmentation

In the computer vision community, instance segmentation refers to detect all interested objects in a given image while also precisely segmenting each instance [13]. It combines two classical computer vision tasks [14]: object detection, which aims to detect interested object instances and return their spatial locations (e.g., via a bounding box) with their category labels; and semantic segmentation, which aims to classify each pixel into a predefined object category list without differentiating object instances.

In the past several years, deep learning techniques have driven significant advances in various computer vision tasks including instance segmentation [14]. Compared to conventional recognition models that consume human-designed features, deep neural networks are powerful as they automatically learn important features from training data themselves [15]. Among various types of deep neural networks, Mask R-CNN surpassed prior state-of-the-art instance segmentation results [13]. It was developed based on another two powerful baseline deep neural frameworks, Faster R-CNN for object detection [16] and Fully Convolutional Network (FCN) for semantic segmentation [17], respectively. Figure 1 shows a high-level Mask R-CNN architecture.

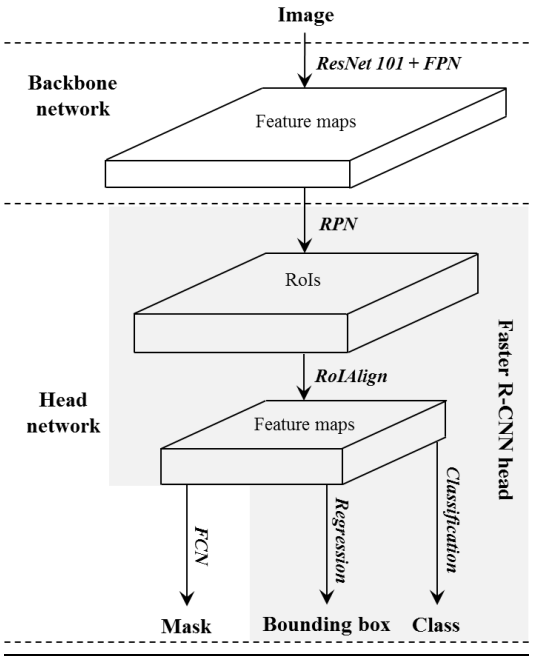

Figure 1. High-level Mask R-CNN architecture based on He et al. [13] and Ren et al. [17].

The Mask R-CNN architecture consists of two connected modules: a backbone network and a head network. The convolutional backbone network is used to extract feature representations from an input image. Then the produced feature maps are fed to the head network for succeeding three parallel tasks, namely object classification, bounding box regression and instance mask prediction. According to He et al. [13], the backbone of Mask R-CNN can achieve excellent gains in both accuracy and speed by combining ResNet [18] and Feature Pyramid Network (FPN) [19]. For the head network, Mask R-CNN was mainly designed by extending the Faster R-CNN head (see the blocks with grey background in Figure 1). The Faster R-CNN head for object detection includes two stages. In the first stage, a Region Proposal Network (RPN) is implemented on top of feature maps produced by the backbone network to propose candidate object bounding boxes (i.e. Region of Interests (RoIs)). In the second stage, high-level features are extracted from each RoI, and then object classification and bounding-box regression are performed. In this stage, Mask R-CNN adds a mask prediction branch, which is a small FCN on top of a feature map, and is parallel with existing classification and bounding box regression branches.

In this study, Mask R-CNN is adapted to achieve a fully automatic building object segmentation from images of buildings. Due to the automatic feature learning ability, Mask R-CNN based building object recognition is expected to be robust to various and complex environmental conditions, and to be scalable to customized building object types. Furthermore, Mask R-CNN allows predicting pixel-accurate masks of objects in images, which provides the potential of constructing building objects with arbitrary-shape surfaces.

3 The Proposed Approach

The proposed approach aims to automatically extract IFC building objects from images of existing buildings to support as-is BIM construction. It consists of three modules (see Figure 2): building objects recognition and segmentation, building object geometry construction, and IFC BIM object generation.

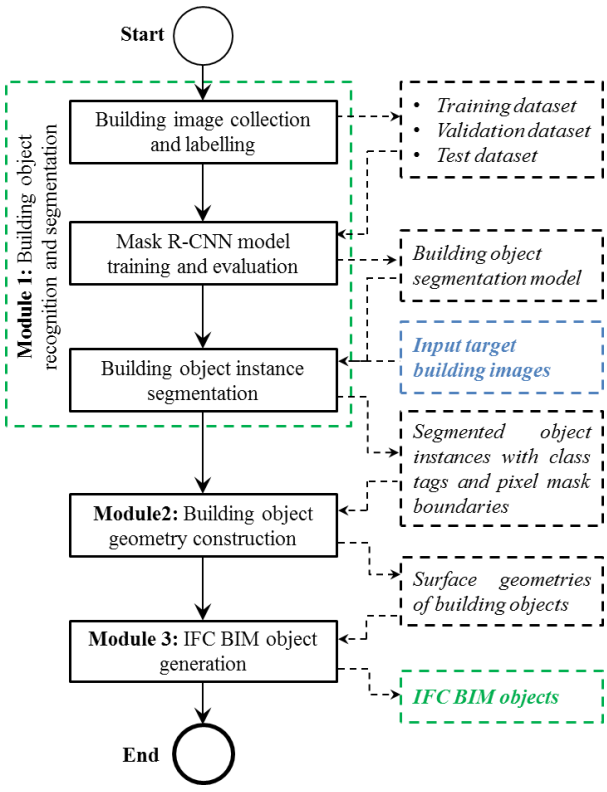

Figure 2. The proposed approach

3.1 Module 1: Building Object Recognition and Segmentation

This module takes building images as an input and produces a list of object instances segmented from each image. Each output instance consists of an object class and a mask boundary of the instance in the image. To link the pixel dimensions with real-world dimensions for object surface construction in the next module, a ruler with a known length (e.g., 1 meter) is used as a reference object in each image. Specifically, this module contains three successive steps: building image collection and labelling (Section 3.1.1), Mask R-CNN model training (Section 3.1.2), and building object instance segmentation (Section 3.1.3).

3.1.1 Building Image Collection and Labelling

Building images (most images, if not all of them, should contain the reference ruler) are collected and labeled to fit the Mask R-CNN model for building object segmentation. To ensure the generalization and the robustness of the model, images of building objects in diverse environmental conditions (e.g., building facades and building interiors, various lighting conditions, and spaces with various usages) need to be considered. Images can be conveniently collected by using handheld digital cameras or smartphones. For each image, all the building objects and the ruler (if present) need to be manually annotated by outlining their masks and adding corresponding class tags. In this study, images are annotated by the "VGG Image Annotator" web tool [20].

3.1.2 Mask R-CNN Model Training and Evaluation

The whole labeled image dataset is randomly split into the following three subsets: a training subset, a validation subset, and a test subset. The Mask R-CNN model is trained based on the image – annotation pairs in the training subset. The goal of the training is to find optimal weight parameters of Mask R-CNN that can map the training images to corresponding annotations with minimal loss. According to He et al. [13], the loss function of Mask R-CNN is defined as a multi-task loss (L) which refers to the sum of the classification loss (L_{cls}), the bounding-box loss (L_{box}), and the mask loss (L_{mask}). Details of L_{cls} and L_{box} can be found in Girshick [21] and details of L_{mask} in He et al. [13]. The validation subset is used to inspect the training process to minimize overfitting. Generally, various training strategies involving the configurations of hyperparameters need to be implemented to obtain an optimal model. For a trained model (i.e., with minimal training loss on the premise of minimal overfitting) obtained under a specific training strategy, it is further evaluated with the test subset. The model that can accurately and robustly segment building objects and the ruler in the test subset is identified and used in the downstream processes.

3.1.3 Building Object Instance Segmentation

Once the Mask R-CNN model is well trained, it can perform the building object segmentation on input

images. Although the model could segment building objects captured in various camera poses, the input images are required to be captured right in front of the target building objects to construct their surface geometries as accurate as possible in Module 2. For the same reason, each input image should contain a well-placed (i.e., vertically or horizontally) ruler (the same ruler used in the training process). By using the scikit-image package (http://scikit-image.org/), the boundary of the predicted mask of each segmented object instance is extracted as a polygon consisting of pixel points in image coordinates.

3.2 Module 2: Building Object Geometry Construction

This module takes the pixel-level mask polygons of objects generated from Module 1 as an input, and produces the surface geometries of objects in two steps: shape extraction (Section 3.2.1) and coordinate transformation (Section 3.2.2).

3.2.1 Shape Extraction

The pixel-level mask polygon of an object is in great detail in terms of shape representation. This step simplifies the shape representation by detecting corner pixels, and then constructing boundary edges (see the example in Figure 3).

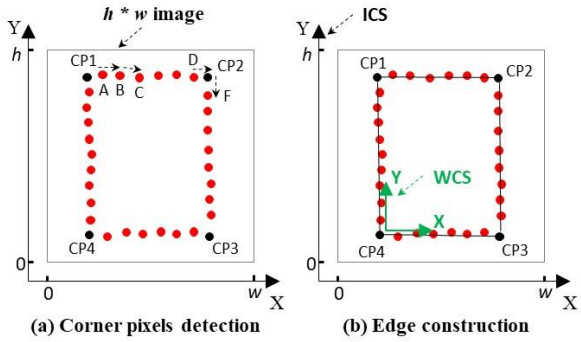

Figure 3. Object shape extraction from a segmented mask

A corner pixel refers to the pixel shared by two adjacent edges to be constructed. Corner pixels are detected by the following procedure. First, for any two adjacent pixels in a mask polygon, the direction vector from one pixel to the other is computed. Second, the angle between two adjacent direction vectors is calculated. Each angle involves a group of three successive pixels. When an angle is larger than a pre-set threshold (e.g., 10 degrees), the corresponding second pixel is recognized as a corner pixel (see CP1, CP2, CP3, and CP4 in Figure 3(a)). Once all the corner pixels are detected, an edge is constructed by fitting two adjacent corner pixels and other pixels between the two corner pixels with a line segment or a circular arc. These two geometric primitives are used in this study is because they are common in representing building object shapes and easy to be processed. As a result, the object shape can be extracted and approximately represented by the constructed edges.

3.2.2 Coordinate Transformation

This step transforms the extracted object shapes represented in an image coordinate system (ICS) into a world coordinate system (WCS), which is used to define the objects' geometries with real dimensions. In each image, objects (e.g., a wall and its hosting openings) on a common plane are assigned with a common WCS. The WCS of an object is set up by taking the lower left point as the origin and keeping x-axis and y-axis of the ICS (see Figure 3 (b)). In this manner, the transformation only involves a translation. To calculate the real dimensions of building objects, the segmented ruler with a known length (i.e., 1m) is used to calculate the pixel dimension by:

$$d_p = \frac{1}{\sqrt[2]{(x_{r\_CP1} - x_{r\_CP2})^2 + (y_{r\_CP1} - y_{r\_CP2})^2}} \quad (1)$$

Where (x_{r_CP1}, y_{r_CP1}) and (x_{r_CP2}, y_{r_CP2}) refers to the image coordinates of two endpoint pixels of a long edge of the segmented ruler instance, respectively.

Then for any $\boldsymbol{P}$ (x_{imag}, y_{imag}), its world coordinates (x_{world}, y_{world}) can be computed by

$$\begin{bmatrix} x_{world} \\ y_{world} \end{bmatrix} = d_p \times \begin{bmatrix} x_{image} - x_{o\_image} \\ y_{image} - y_{o\_image} \end{bmatrix} \quad (2)$$

Where $(x_{o_{imag}}, y_{o\_imag})$ refers to the image coordinates of the origin of the world coordinate system.

Since each extracted shape is defined by a combination of line segments and/or circular arcs, the transformation of a shape essentially refers to transform all relevant geometric primitives into the corresponding WCS. For a line segment, its two endpoints are transformed via Equation (2). For a circular arc defined by a center, a radius, two trimming points and an arc direction, the center and the two trimming points need to be transformed via Equation (2).

3.3 Module 3: IFC BIM Object Generation

This module defines the constructed building objects as IFC objects. For each building object, only the visible sections of surfaces (i.e., exposed to a space or the outdoors) are constructed. Thus, the IFC concept of space boundary (SB) (i.e., IfcRelSpaceBoundary) [22], which defines the surface partition of a building object that bounds a space or contacts outdoors, is proper to

store the constructed information. Figure 4 shows the IFC entities and data structure used to define a SB in the IFC4 specification. To be more specific, each building object is defined by an IfcRelSpaceBoundary instance. To store the object type information, this instance references a building element instance that matches the type via the attribute "RelatedBuildingElement". For example, for a wall object, an IfcWallStandardCase instance is created and linked to the IfcRelSpaceBoundary instance via that attribute. The object surface geometry is defined by the IfcRelSpaceBoundary instance via the attribute "ConnectionGeometry" (see Figure 4). It is noteworthy that the created IFC SB instances do not constitute a valid IFC model as other IFC instances (e.g., IfcProject, IfcSite, IfcBuilding and IfcSpace) required by the IFC4 specification are not yet included. All the missing instances can be automatically inferred and added by using the semantic enrichment approach developed by Ying et al. [6].

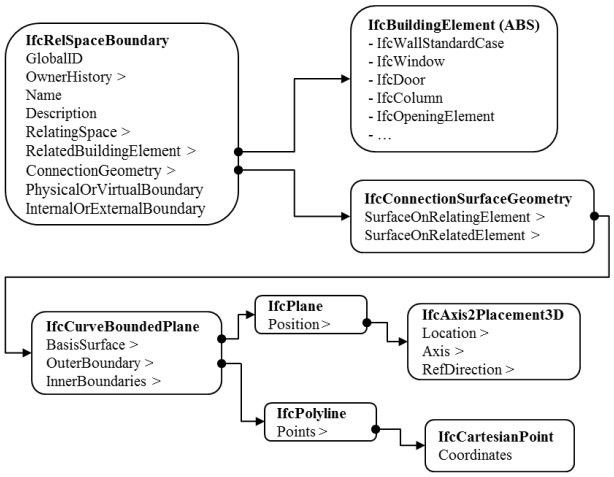

Figure 4. Instantiation diagram to define a space boundary in IFC4 schema

4 Implementation and Experiment

A prototype system implementing the proposed approach was developed in multiple programing languages. Module 1 was implemented in Python. Module 2 and Module 3 were implemented in C#. In the rest parts of this section, the implementation details of Mask R-CNN for building object segmentation are first elaborated and then a preliminary experiment of the entire approach is presented.

4.1 Mask R-CNN Model Implementation

4.1.1 Dataset Preparation

The authors created a 2D image dataset to train a Mask R-CNN model for building object segmentation. The dataset includes 430 images from interiors of 4 multifunctional buildings in the University of Hong Kong with significantly different decoration designs (see some examples in Figure 5). All the images were captured by a smartphone and their sizes are 3024×4032. The dataset also includes the ground truth annotation of each building object contained in each image. The authors randomly split the annotated dataset into three sub datasets: 80% (344 images) for training, 10% (43 images) for validation, and 10% (43 images) for testing. Table 1 shows the number of building object instances in each sub dataset. In this test, only three types of building objects (i.e. walls, doors and lifts) and a reference ruler were considered. Other objects such as columns, beams and windows will be taken into account later.

Figure 5. Varieties of collected building interior images in the dataset

Table 1. The number of building objects in each dataset

Dataset	Wall	Door	Lift	Ruler
Training dataset	599	443	34	99
Validation dataset	68	49	6	11
Test dataset	73	53	2	12

Whole dataset	740	545	42	122

4.1.2 Training

The authors implemented the Mask R-CNN model using an open-source package built on Keras and Tensorflow developed by Matterport [23]. ResNet-101 + FPN are used as the backbone network. Parameters of the sub models of the Mask R-CNN (e.g., ResNet-101, FPN, RPN, FCN etc.) are left in default settings in Matterport's implementation, which basically follows the suggestions in He et al. [13]. It is noteworthy that all the input images (3024×4032) are automatically resized as 1024×1024 for training in the Matterport's implementation. Instead of training the model from scratch, the authors used the weights of a pre-trained Mask R-CNN model on MS COCO (http://cocodataset.org/#home) to initialize the model. In addition, to make the resulting model more robust, a set of image argumentation operations (e.g., horizontal flipping and Gaussian blurring) were randomly used to introduce variety in the training images. The model was trained on a desktop with 61GB RAM and a NVIDIA Tesla K80 GPU with 12 GB memory. The training process consists of two stages. In the first stage, only the head layers that do not use pre-trained weights from MS COCO are trained to adapt the building object segmentation task. In the second stage, all layers are trained to fine-tune the weights to achieve the best performance. The authors set the batch size of 2, the learning momentum of 0.9, and the weight decay of 0.0001 for both training stages, and the learning rate of 0.001 for the first stage and the learning rate of 0.0001 for the second stage. The authors adopted an early stopping strategy to minimize overfitting with the validation dataset. The model was trained with 60 epochs to record the full learning curve. Finally, the model trained at the 50th epoch was selected as the model began to get overfitting after 50 epochs (i.e., the validation loss began to increase).

4.1.3 Assessment

The performance of the trained model was evaluated with the hold-out test dataset from two aspects: (1) the object classification accuracy, which aimed to evaluate the performance of the model in terms of building object and ruler recognition; and (2) the object segmentation accuracy, which aimed to evaluate the performance of the model in terms of the instance mask generation. A positive classification of an instance is acknowledged if the following two criteria are satisfied: (1) the predicted mask of the instance has an overlap with the ground truth; and (2) the predicted class of the instance is correct. Table 2 shows the overall classification accuracy and the object-level classification accuracy with precision and recall. The results show that the trained model can accurately and robustly recognize the building objects and the reference ruler.

Table 2. Classification accuracy of the trained model on the test dataset

		P	N	Precision	Recall
Wall	T	72	0	97.3%	100%
	F	2	0		
Door	T	48	0	100%	90.6%
	F	0	5		
Lift	T	2	0	100%	100%
	F	0	0		
Ruler	T	11	0	100%	91.7%
	F	0	1		
Overall	T	133	0	98.5%	95.7%
	F	2	6		

Note: P – Positive; N – Negative; T – True; F – False; Precision = TP / (TP + FP); Recall = TP / (TP + FN).

For the segmentation accuracy evaluation, the authors use the metric called mAP (the mean of average precision values of all classes), which is commonly used in computer vision community to evaluate the performance of an object detector. Given that the quality of a predicted mask would have a significant effect on corresponding surface construction, the authors set a large threshold value - 0.75 - of IoU (Intersection over Union: the ratio between the intersection and the union of the predicted mask and the ground truth mask of an instance) for the mAP computation. The mAP $^{IoU = 0.75}$ of the trained model on the test dataset reaches 0.912, which indicates that the trained model can generate effective masks.

4.2 Experiment

A new image (i.e., not in the dataset used for the Mask R-CNN model training, validation and testing) taken in an interior space at night, as shown in Figure 6(a), is used to validate the feasibility of the proposed approach. The image mainly contains three building objects, a reference ruler, as well as a noisy object – the emergency exit sign. The three building objects (from left to right in Figure 6 (a)) includes a wall showing a small surface region, a wall showing the entire surface, and a door showing the entire surface. The latter two are the target building objects to be constructed. Figure 6(b) – (f) shows the whole flow of generating interested IFC objects from the image by the proposed approach. In Figure 6 (b), all three building objects and the ruler are correctly segmented by the trained Mask R-CNN model. Then the two target building objects are further distinguished from the segmented wall instance with a partial surface region, and proceed to the downstream

steps with the ruler to construct surface geometries (see Figure 6(c) – (e)). Finally, two corresponding IFC objects are successfully generated (see Figure 6(f)).

Regarding the geometry construction accuracy in the result, the areas of the constructed wall surface and door surface are 8.69 m^2 and 4.78 m^2 respectively. Compared to the real values (measured with a laser range finder), 7.97 m^2 for the wall and 4.184 m^2 for the door, the errors are +9% and +13.97% accordingly. As seen in Figure 6, these errors are mainly caused from the imperfection in the reference ruler placement and its segmentation (Figure 6(b)) as well as the process of determining building object shapes from the extracted mask boundaries (from Figure 6(c) to Figure 6(d)). In future, the Mask R-CNN model needs to be improved to generate more accurate masks for segmented objects.

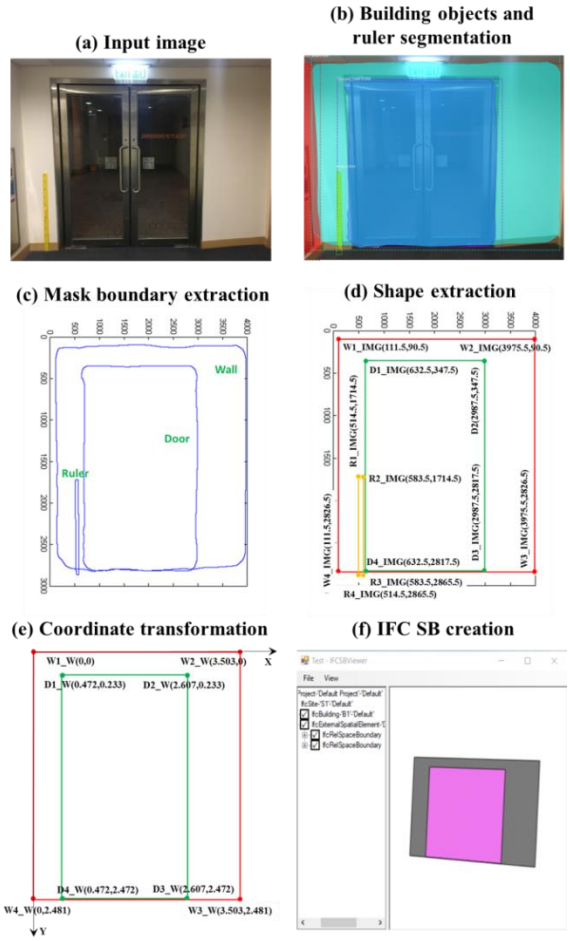

Figure 6. The flow of processing an image by the proposed approach

5 Conclusion and Future Work

In this study, a fully automatic approach is presented to construct IFC building objects from digital images of existing buildings. Unlike previous image-based approaches that rely on human-designed features to recognize building objects, the proposed approach automatically learns important features from training data themselves to segment building object instances. This makes the approach to be robust in various building indoor conditions, and to be efficiently scalable to customized building object types. Furthermore, the approach is able to construct building objects with complex surface shapes. The proposed approach was implemented with three modules: Mask R-CNN based building object recognition and segmentation, building object geometry construction and IFC BIM object generation. 430 images containing three types of building objects (i.e., walls, doors, and lifts) were collected and used to train and test the Mask R-CNN model. The test results show that the trained Mask R-CNN model can accurately and robustly recognize the three types of building objects and the reference ruler (with average precision of 98.5% and average recall of 95.7%), and can generate effective masks for the recognized building object and ruler instances (with mAP $^{IoU = 0.75}$ of 0.912). Based on the trained Mask R-CNN model, the feasibility of the entire approach was examined by successfully extracting IFC BIM objects from an image, while the geometry construction accuracy still has room for further improvement. It is expected that the approach can be useful for researchers and practitioners to develop semantically rich as-is BIMs of existing buildings.

In future, the approach will be further improved in the following aspects. First, the Mask R-CNN model will be improved to generate more accurate masks of segmented objects to support accurate surface geometry construction. Second, the size of the image dataset will be increased with more images from different environments (e.g., building exteriors) and different building types (e.g., residential buildings and commercial buildings) to improve the performance and enhance the generalization of the Mask R-CNN model. Other common building objects such as windows, beams, columns, and curtain walls will be included. Third, more flexible and accurate dimension measurement method (e.g., photogrammetry technique) will be introduced to address the constraint on input image capturing (i.e., taking images right in front of target building objects). Finally, the performance of the proposed approach will be further examined by more case studies.

Acknowledgement

The work described in this paper was supported by a grant from the Research Grants Council Early Career

Scheme of the Hong Kong Special Administrative Region, China (Project No. HKU 27203016).

References

[1] Lu Q. and Lee S. Image-based technologies for constructing as-is building information models for existing buildings. *Journal of Computing in Civil Engineering*, 31(4), p.04017005, 2017.

[2] Becerik-Gerber B., Jazizadeh F., and Li N., et al. Application areas and data requirements for BIM-enabled facilities management. *Journal of Construction and Engineering Management*, 138(3): 431-442, 2011.

[3] Tang P., Huber D., Akinci B., Lipman R. and Lytle A. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, 19(7): 829-843, 2010.

[4] Pătrăucean V., Armeni I., Nahangi M., Yeung J., Brilakis I., and Haas C. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2): 162-171, 2015.

[5] Bhatla A., Choe S.Y., Fierro O., and Leite F. Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras. *Automation in Construction*, 28: 116-127, 2012.

[6] Ying H., Lu Q., Zhou H., and Lee S. A framework for constructing semantic as-is building energy models (BEMs) for existing buildings using digital images. In *Proceedings of 35th International Symposium on Automation and Robotics in Construction*, pages 309-316, Berlin, Germany, 2018.

[7] Ying H., Zhou H., Lu Q., Lee S., and Hong. Y. Semantic Enrichment of As-is BIMs for Building Energy Simulation. In: Mutis I., Hartmann T. (eds) *Advances in Informatics and Computing in Civil and Construction Engineering*, pages 733-740, Springer, Cham, 2019.

[8] Oskouie P., Becerik-Gerber B., and Soibelman L. Automated recognition of building façades for creation of As-Is Mock-Up 3D models. *Journal of Computing in Civil Engineering*, 31(6), p.04017059, 2017.

[9] Zhu Z. and Brilakis I. Concrete column recognition in images and videos. *Journal of Computing in Civil Engineering*, 24(6): 478-487, 2010.

[10] Neuhausen M. and König M. Automatic window detection in facade images. *Automation in Construction*, 96: 527-539, 2018.

[11] Cao J., Metzmacher H., and O'Donnell J., et al. Facade geometry generation from low-resolution aerial photographs for building energy modeling. *Building and Environment*, 123: 601-624, 2017.

[12] Lu Q., Lee S., and Chen L. Image-driven fuzzy-based system to construct as-is IFC BIM objects. *Automation in Construction*, 92: 68-87, 2018.

[13] He K., Gkioxari G., Dollár P., and Girshick R. Mask R-CNN. In *Proceedings of International Conference on Computer Vision*, pages 2980-2988, Venice, Italy, 2017.

[14] Liu L., Ouyang W., Wang X., Fieguth P., Chen J., Liu X., and Pietikäinen M. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[15] LeCun Y., Bengio Y. and Hinton G. Deep learning. *Nature*, 521(7553): 436, 2015.

[16] Ren S., He K., Girshick R., and Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of Advances in neural information processing systems*, pages 91-99, Montreal, Canada, 2015.

[17] Long J., Shelhamer E., and Darrell T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431-3440, Boston, USA, 2015.

[18] He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770-778, Las Vegas, USA, 2016.

[19] Lin T.-Y., Doll´ar P., Girshick R., He K., Hariharan B., and Belongie S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 2017.

[20] Dutta A., Gupta A. and Zisserman A. VGG Image Annotator (VIA). Online: http://www.robots.ox.ac.uk/~vgg/software/via/, Accessed: 18/1/2019.

[21] Girshick R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440-1448, Santiago, Chile, 2015.

[22] BuildingSMART. IfcRelSpaceBoundary. Online: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcproductextension/lexical/ifcrelspaceboundary.htm, Accessed: 18/1/2019.

[23] Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Online: https://github.com/matterport/Mask_RCNN, Accessed: 15/11/2018.