

The following publication Li, R., Chi, H.-L., Peng, Z., Li, X., & Chan, A. P. C. (2023). Automatic tower crane layout planning system for high-rise building construction using generative adversarial network. *Advanced Engineering Informatics*, 58, 102202 is available at <https://dx.doi.org/10.1016/j.aei.2023.102202>.

Automatic Tower Crane Layout Planning for High-Rise Building Construction Using Generative Adversarial Network

Rongyan LI¹, Hung-Lin CHI^{1,*}, Zhenyu PENG¹, Xiao LI², Albert P. C. CHAN¹

¹Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Hong Kong

²Department of Civil Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong

*Corresponding author, e-mail addresses: hung-lin.chi@polyu.edu.hk

Abstract

With the rise of high-rise building projects, tower crane layout planning (TCLP) is increasingly crucial to avoid costs, safety issues, and productivity deficiencies. Current optimization approaches require manual data extraction and become more complex as projects grow. To further alleviate the planning burden, an automatic TCLP system is proposed, using a generative adversarial network (GAN) called CraneGAN. It generates tower crane layouts from drawing inputs, eliminating the need for manual information extraction. CraneGAN is trained on a high-quality dataset and evaluated based on computational and transportation times. By adjusting hyperparameters and applying data augmentation, CraneGAN achieves robust and efficient results compared to genetic algorithms (GA) and the exact analytics method. After validating through a numerical analysis for construction projects, this proposed approach overcomes complexity limitations and streamlines the manual data extraction process to a better facilitate layout planning decision-making.

Keywords:

Tower crane; Crane location; Generative adversarial network; Computer vision; Automatic design; Image-to-image translation

27 **Highlights:**

- 28 • An image-based automatic tower crane layout planning approach is proposed;
- 29 • A highly scalable crane layout prediction network, CraneGAN, is developed;
- 30 • The results of CraneGAN in transportation time outperformed GA in 66.67% of cases;
- 31 • Taking images as input also saves manual data extraction time for planning preparation;
- 32 • CraneGAN prediction remains in polynomial time while maintaining the layout quality.

1. Introduction

Rapid urbanization has increased the number of high-rise residential buildings in city areas [1]. As crucial transportation machines in high-rise building construction projects, tower cranes deliver construction components, temporary machinery, or materials vertically and horizontally, governing the installation and construction schedule [2][3]. According to the statistics from Rider Levett Bucknall (RLB), the number of operation tower cranes in 14 major cities across the U.S. and Canada increased 7.04% in Quarter 1 (Q1) 2023, -0.62% in Q3 2022 and 4.50% in Q1 2022, compared with it in Q3 2022, Q1 2022 and Q3 2021 separately [4]. Despite the temporary impact of the Covid-19 pandemic on the construction market, the available data indicates a recovery in the construction sector and an upward trajectory in the utilization of tower cranes. Proper tower crane layout planning (TCLP) can accelerate the construction process by reducing transportation time to improve transportation efficiency and avoiding secondary handling to save construction costs [5][6]. In the construction process, contractors mainly rely on their experience to decide on a tower crane's location, which sometimes leads to significant deviations in the decision quality when facing various building layouts or newly involved lifting approaches, such as modular construction [7]. Therefore, efficiently solving TCLP problems in high-rise building construction projects is a crucial and practical scientific problem. Currently, scholars are applying mathematical approaches to obtain the optimal locations of tower cranes. Those approaches systematically consider diverse site situations and constraints to address the TCLP challenge, further aiding in the decision-making that was previously done empirically.

With the development of many mathematical theories and computer-aided algorithms, TCLP problems are being solved with improved outcomes, primarily through optimization approaches [8]. The optimization approaches used in TCLP problems can be categorized into analytical and numerical approaches. The former includes the branch-and-bound method and dynamic programming, taking considerable computing time to traverse possible solutions and come up with optimal outcomes based on the given constraints [9]. The solutions are precise in achieving planned objectives, while the main drawbacks are their time-consuming processes and inflexibility to change. To overcome such drawbacks, numerical approaches, such as metaheuristics (e.g., genetic algorithm (GA) [10], and firefly algorithm (FA) [11]) and hybrid-metaheuristics algorithms (e.g., an artificial neural network (ANN)-GA method [12] and the particle bee algorithm (PBA) [13]), are used to find near-optimal solutions within tolerable computational time. These algorithms were fast when performing optimization calculations but had unstable accuracies and success rates. The computational time increased exponentially when facing large-scale and high-complexity cases. These issues make them unable to realize real-time and flexible solutions. In addition, with all those solutions, explicit manual data input of construction projects is required, which usually takes hours for a skilled operator to complete.

To overcome the aforementioned challenges, the latest deep learning approaches can be a future improvement for solely or cooperatively solving crane layout problems. A generative adversarial network (GAN) consists of generator and discriminator models to generate realistic synthetic information with the desired style on images and videos [14]. GANs have been successfully adopted for various design tasks, such as automated architectural home design [15], interior layout design [16], and shear wall design [17]. They all showed promising problem-solving results in planning efficiency and effectiveness. This reveals that GANs are able to learn the design or planning patterns

based on the graphical input instead of those with exact analytical information, simplifying and accelerating the utilization process. In construction, two-dimensional (2D) drawings are still the mainstream visualization and communication aids for engineers in construction design [18]. Hence, this study proposes a GAN-based automatic TCLP system to improve planning efficiency and performance by learning from existing layout planning drawings for minimal transportation time. A developed GAN network, CraneGAN, was trained among the system and evaluated with the existing heuristic optimization approach to determine its performance. A numerical analysis was applied for the system validation. Hyper-parameter adjustment and data augmentation were applied in CraneGAN, further improving the TCLP design. This study explored the application of advanced computer vision technology (GAN) in construction engineering problems (TCLP), improving reliability in engineering decision-making.

The remaining sections of the paper are organized as follows: Section 2 reviews the existing TCLP methods and automated design technologies. Section 3 illustrates the proposed framework and architecture of the CraneGAN-based automatic TCLP system. Section 4 elaborates on the implementation and evaluation of CraneGAN. Section 5 presents a real case application based on the uses of the proposed system. Section 6 discusses the improvements of the CraneGAN performance by hyper-parameter adjustment and data augmentation and whether the case complexity can affect the CraneGAN performance. Finally, Section 7 presents the key findings and future research directions.

2. Literature Review

2.1 Technologies for Tower Crane Layout Planning (TCLP)

TCLP is one of the construction site layout planning problems required to determine the tower crane layout considering complicated construction situations under specific optimization objectives and constraints, such as a limited material supply area, crane capacity, and operation range [19]. To quantify this problem, TCLP problems are usually formed as mathematical models with objective functions to seek time, cost, or safety optimality under various constraints. In 1984, a dynamic programming model was proposed to find tower cranes' optimal selection and location based on their rental costs [9]. Later, mixed integer programming (MIP) [5], mixed integer linear programming (MILP), [25] and binary mixed integer linear programming (BMILP) [26] models were introduced to formulate TCLP problems with constraints and objectives to minimize the travel time or costs. Realizing the minimal transportation time and crane operating costs are the current mainstream optimization objectives. The minimal transportation time is determined based on the crane hook movement, and the minimum total operating costs are calculated by multiplying the transportation time by the cost per unit time [25]. Given that transportation time is a positive influencing factor of the operation cost, it is usually selected as the sole optimization objective for TCLP studies. For constructing a mathematical model of the TCLP problem, the composition of the constraints, influencing factors, and optimization objectives gradually become comprehensive with research development.

Many optimization approaches have been used to find the exact and approximate optimal layout for tower cranes. The branch-and-bound method was introduced to obtain the exact optimal planning results considering the priority for urgent materials and the minimal time for traversing all possible locations [26]. To further reduce the computational time, metaheuristics and hybrid-

metaheuristics algorithms have been used to find the near-optimal layout within a tolerable time consumption. The GA is a commonly used algorithm for obtaining the near-optimal tower crane layout [10]. Based on the objective of a minimal cost, Tam et al. [28] and Tam and Tong [12] applied the GA to determine the layout both for the tower crane and the material supply area. Marzouk and Abubakr [10] utilized the GA to address TCLP problems by minimizing the transportation time and the crane operational and rental costs separately.

In addition to the GA, Wang et al. [11] proposed an automated TCLP system by applying the firefly algorithm (FA). In 2018, Ali et al. [29] proposed four newly developed metaheuristic algorithms for TCLP: colliding bodies optimization (CBO), enhanced colliding bodies optimization (ECBO), vibrating particles system (VPS), and enhanced vibrating particles system (EVPS). ECBO performed better than the other three methods. In 2020, an upgraded sine cosine algorithm (USCA) was used to obtain the optimal locations of cranes by adding agent memory and a harmony-search-based side constraint approach. A comparison with the results of the PSO, VPS, CBO, whale optimization algorithm (WOA), slap swarm algorithm (SSA), and sine cosine algorithm (SCA) showed that the USCA had a faster convergence speed and superior stability [30].

Hybrid-metaheuristics algorithms have also been applied. In 2002, an ANN-GA approach was utilized. The ANN was introduced as a method to predict the non-linear crane operation time based on specific factors, such as the loading point, angular movement, weight, and hoisting height. The GA was still used to find the optimal layout of tower cranes [11]. In 2014, particle bee algorithm (PBA), which combined honeybees (bee algorithm, BA) and birds (particle swarm optimization, PSO), was introduced to minimize crane operating costs by determining the locations of tower cranes [13].

In addition, several scholars integrated mathematical and simulation approaches to visualize and evaluate their proposed TCLPs. After obtaining calculated TCLP, Building Information Modeling (BIM)-based simulation was used to visualize and assess the results for avoiding collision during lifting [11]. Agent-based simulation (ABS) model was also used to evaluate the interaction of tower cranes, material supply locations, lifting tasks, and other agents for optimal combinations of tower crane layouts [23]. Further, a Java-based ABS tool was developed for crane operation planning in overlapping areas [31]. To involve users in the planning process, a virtual reality (VR) tool was introduced in TCLP for human-in-the-loop simulation, integrating multi-criteria and dynamic situations during crane layout planning for high-rise modular construction [32]. Moreover, reinforcement learning (RL)-based simulation approach was applied in tower crane-related tasks in 3D virtual environment with six modeling strategies to train the RL agent, enhancing the guidance of crane lifting [33]. Those simulation-based approaches required existing BIM models for construction projects and exact manual input involved. In addition, the human-in-the-loop simulation needs experts to guarantee the decision quality of TCLP.

Overall, the uses of the aforementioned optimization algorithms were based on input accurately extracted data from building targets and related site environments, which is a labor-intensive collection task and challenging to automate. In addition, when dealing with large-scale and high-complexity constructions, the computational time grows exponentially. Therefore, applying an image-based decision-making approach instead of exact model forming and solving is a reasonable and workable approach to avoid massive data collection efforts, potentially improving the decision efficiency.

2.2 Generative Adversarial Network (GAN) and Image-to-Image Translation

A GAN is a deep learning framework that has contributed to numerous computer vision problems [34], such as image generation [35], video generation [36], image translation [37], and auto-design [38]. Generally, a GAN model includes a generator and a discriminator with their corresponding neural networks learning from a labeled dataset [39]. For example, with image generation, the generator aims to produce a synthetic image that is similar to real images in the labeled dataset, while the discriminator attempts to distinguish whether the synthetic image is really a fake image [40]. Such adversarial activities during the iterative network training processes improve the realism level of the image synthesis and finally result in an image generated with relatively rich realistic features [41]. Specifically, the generator is trained to learn a probability-distributed pattern so that it can generate different synthetic images to trick the discriminator. After introducing probabilistic noise, a synthetic image is generated with a certain probability. At the same time, based on the labeled dataset with real images, the discriminator also learns a probability-distributed pattern that can calculate a loss that reflects the similarity between the input image (either synthetic or real image) and the real image. The loss will cause the generator to update its own probability distribution. Once the discriminator and generator reach the set number of epochs or the set loss, the training will stop [42].

Image-to-image translation is a task of computer vision that converts an input image to another desired image style, such as transferring a real image to a cotton style or changing a person's facial expression from joy to frustration [14]. Therefore, the generation process needs to be more controllable to realize style transfer. However, the main aim of a plain GAN is to make the synthetic images more realistic, and it is hardly restricted to images with a certain style or content. Conditional GAN (CGAN) was thus proposed and became the main generation paradigm that could realize category-specific image generation. Compared with plain GANs, the CGAN generator model input consists of a random noise vector and a conditional vector (e.g., an image of a face waiting for its expression to be changed) [39][44]. Pix2pix is a special CGAN framework that reduces the noise input and uses paired datasets, showing remarkable results for image-to-image translation tasks [14]. Later, pix2pixHD was proposed to generate more refined and precise images with multi-scaled discriminators and a coarse-to-fine generator [54]. Paired data (e.g., a real image plus its desired style-transferred one) enables the CGAN to accurately learn the matching relationships between the data with less data consumption. However, there is poor data accessibility in many tasks. To address this problem, several frameworks, including CycleGAN [45], discover cross-domain relations with GAN (DiscoGAN) [46] and dual GAN [47], were proposed to adapt to unpaired image-to-image translation. For a TCLP, paired datasets can be scientifically and easily obtained by planning documents to improve training accuracy. Therefore, the pix2pix and pix2pixHD frameworks were selected in this study, and the generator and discriminator were trained based on paired databases.

Based on existing studies, many scholars have focused on obtaining a near-optimal location for the tower crane in a shorter computational time. Several emerging optimization algorithms with the objective of saving time were also applied to address this problem and performed well. However, these applied optimization technologies were based on the input of exact characteristic data of buildings and their components' properties, which requires information to be extracted from documents manually. The extraction procedure is time-consuming, and due to different design intentions and drawing styles, it is sometimes difficult to directly obtain precise information based

207 on computer-aided design (CAD) documents. Therefore, in this research, an image-based automatic
208 TCLP system is proposed, which uses a GAN-based framework to realize the automated design of
209 a tower crane installation location to save data retrieval efforts from the existing construction
210 documentation.

3. CraneGAN-Based Automatic Tower Crane Layout Planning

System

3.1 Planning Process

The construction site layout is a complex design problem for generating reasonable layout arrangements for the production and temporary facilities at the construction site with a specific execution sequence [50] [51]. After obtaining the original information required for the design of the construction site layout (such as the building plan and construction conditions), the location of vertical transportation machinery needs to be prioritized and determined. It is because they significantly affect the location of the on-site sidewalk, mixing plant, and other temporary facilities. A tower crane is one of the vertical transportation machinery. This prudent technical decision-making issue strives to determine the tower crane's type, number, and optimal location based on construction objectives considering various constraints [51]. The essential decision process of TCLP problem solving is shown in Fig. 1. The building plan, construction conditions, schedule, and resources are known information ahead of the decision-making process. Firstly, the content of TCLP is identified, which mainly determines potential types, numbers, and optimal layouts of tower cranes. Then, the objectives of TCLP are identified from the construction cost, efficiency, and safety perspectives based on the project's requirements. Thirdly, main detailed constraints are explored to achieve those objectives, including economic, technical, and safety constraints. To quantify these constraints, the crucial influencing factors, such as construction height limitations, are considered to facilitate setting up the boundary conditions of the TCLP. In actual construction site layout planning, the location of the tower crane is determined advanced to maximize the coverage of the project operations, not only the building construction area but also the material processing, unloading, and material storage areas. Other temporary equipment, such as construction hoists and steel processing yards, will be arranged afterward. Considering the potential influencing factors, a series of alternative solutions for TCLP can be generated. Finally, the optimal TCLP is selected based on the above construction objectives. This study considers the required objectives, constraints, and essential influencing factors in the dataset generation phase, leading CraneGAN to learn the layout planning patterns to generate proper plans in practical cases further.

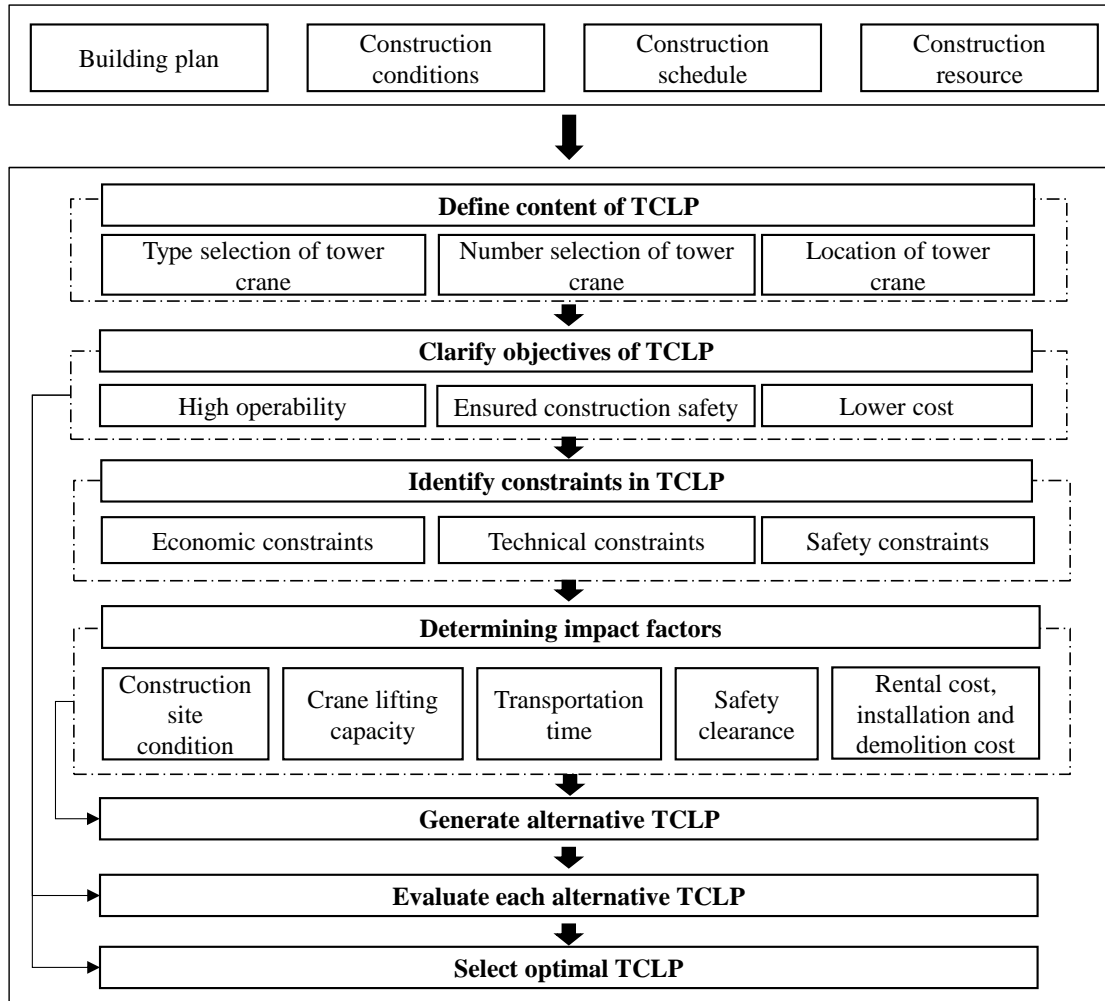


Fig. 1 Flowchart of the tower crane layout planning (TCLP) process

3.2 Framework

In this study, a CraneGAN-based automatic TCLP system is proposed, which can ease the data collection efforts through direct utilization of construction drawings and tackle the exponentially increasing processing complexity. As shown in Fig. 2, the automatic TCLP system consists of three parts: I. Extraction - The lifting task information is extracted from CAD drawings of construction site conditions, building plans, and the potential tower crane installation options identified to form pending design images for layout planning; II. Generation - The trained CraneGAN is applied to predict a workable tower crane layout based on the massive dataset training and improvement approaches with the pending design images as input. III. Identification - The generated tower crane layout from Part II. is identified, evaluated, and finally visualized in a clear image. Once the generated tower crane layout cannot be identified or evaluated as a failed drawing, it will force CraneGAN to generate another plan until it is qualified.

The proposed system combines image-based data collection with the well-trained CraneGAN to obtain an appropriate tower crane layout. The primary aspect of this approach is that it transfers the TCLP problem into an image-to-image translation problem instead of mathematical model calculation, which avoids manual construction data collection and potential complexity scaling

problems. In addition, the objectives and constraints of the generated tower crane layout depend on the training dataset of CraneGAN, which can be extended and replaced. Therefore, this proposed system is an extensible framework that can provide reliable and acceptable results based on different objectives and constraints for construction managers and contractors, supporting further layout planning decision-making.

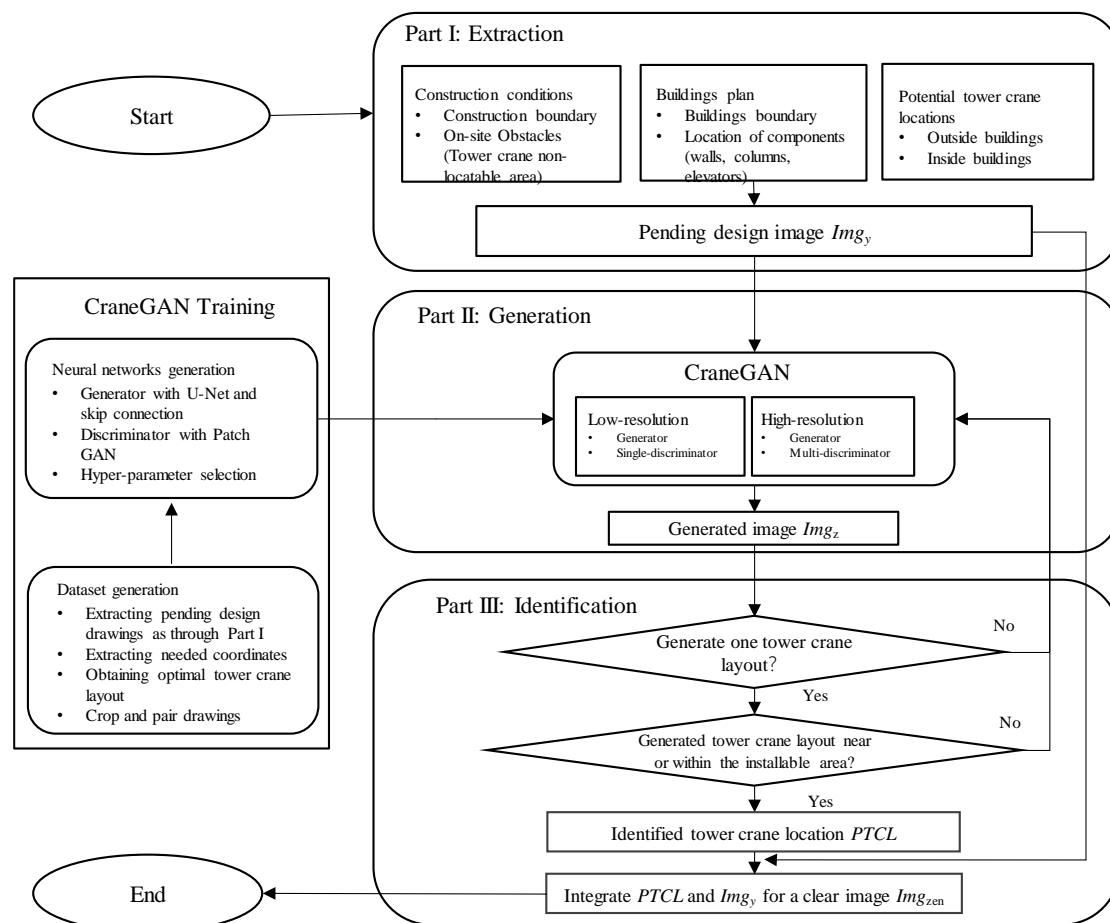


Fig. 2 Framework of the automatic tower crane layout planning (TCLP) system

3.3 Part I: Extraction

For high-rise buildings, tower cranes are usually located at the elevator shafts for better internal climbing or attached outside of the buildings on the land with good bearing capacity to complete transportation tasks [62]. Also, construction sites sometimes have temporary facilities with pre-determined locations, resulting in cranes that can only be positioned in a defined area. Moreover, a crane lift task mainly consists of four essential elements, including cranes, construction components for transportation, and the load and unload places when calculating the hook transportation time [50]. Therefore, those affecting information on construction site conditions and the potential location of the tower crane needs to be considered when deciding on the tower crane layout, which is required to be displayed in the pending design drawings for the automatic TCLP system. In Part I, a pre-processing approach is applied to emphasize the relationship of the tower crane layout with

those affected factors mentioned before. Hence, the extracted image includes the location of construction and building boundaries, the recognized obstacles (such as existing buildings and pre-determined temporary facilities), potential tower crane locations, the building plans, and the material yard location. A pending design drawing is extracted from the raw construction materials, as shown in Fig. 3.

The legend and color of the pending design images are unified in this study, which is the same as in the dataset to enable the CraneGAN in Part II to recognize and apply the learned planning patterns to predict a workable location for the tower crane. In addition, different colors in RGB are used to distinguish the information on the potential location of the tower crane (Blue), materials supply area (Green), and other construction information (Black) in the pending design images.

During the extraction process, the generation of the pending design images is image-based which can be realized in software like AutoCAD without requiring any explicit pre-processed data. AutoCAD has become the most commonly used CAD software, reaching about 40% of the global CAD/CAE/CAM software market [53]. Due to its simplicity and popularity, this approach targets using AutoCAD or similar design software in TCLP for the decision maker. So far, the pending design images as input of CraneGAN have been generated.

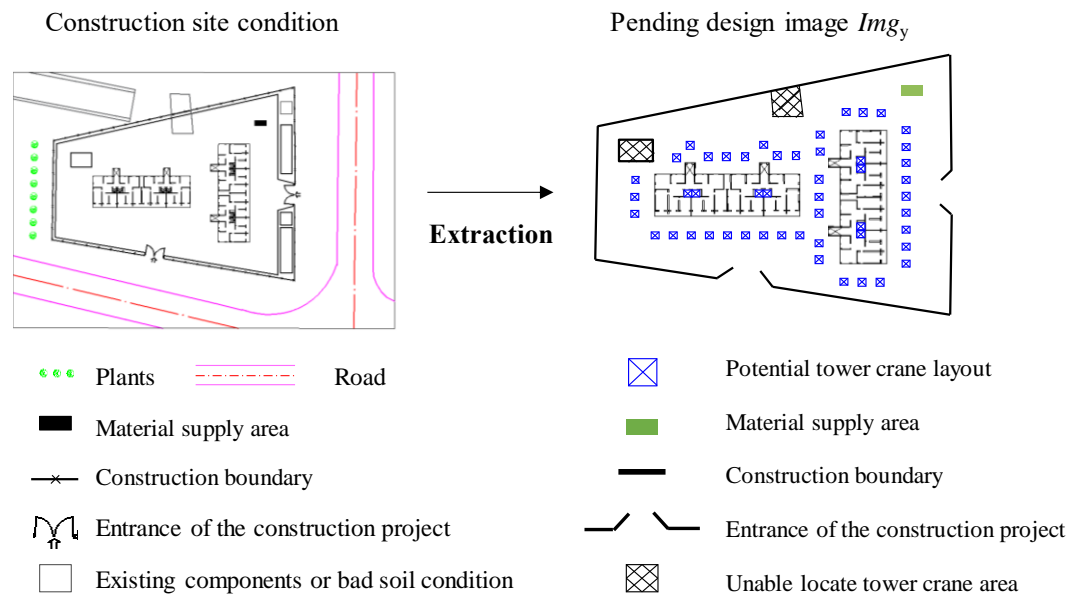


Fig. 3 The extraction processes of the images for TCLP

3.4 Part II: Generation

3.4.1 CraneGAN Basics and Training Process

In the context of TCLP problems, the primary focus lies in generating a proper tower crane layout using graphical information obtained from construction projects, which is the objective of Part II. A strong correspondence exists between the input construction project information and the resultant tower crane layout. Hence, the proposed generator is supposed to read and understand the construction information to predict the tower crane layout. Compared with basic GANs, CGANs are more suitable for TCLP problems, as the input image is a conditional information for generating image instead of a simple noise vector. And they can efficiently learn the mapping between the input

images and target images in paired datasets, enabling it to generate realistic and coherent synthetic images based on given input conditions. In addition, in the architectural construction engineering (AEC) industry, CGANs have successfully been applied for architectural layout design [55] and hospital operating department layouts [56]. Pix2pix and pix2pixHD are typical frameworks that produce high-quality synthetic images, detailed diversity, and more robust applicability [14] [54]. Therefore, CraneGAN adapts the characteristics of these frameworks with the paired datasets, including the loss function, generator model, and discriminator model, as shown in Table 1. To be suitable for different sizes of the input images, CraneGAN is available in two versions, a low-resolution version for small-size images (256×256) and a high-resolution version for large-size images (1024×1024). Further details regarding the network architectures are elaborated in the following sub-sections.

Table 1 Characteristics of CraneGAN

Version	Input image size	Loss function	Generator	Discriminator
Low-resolution	256×256	$L_{GAN} + \lambda_1 \cdot L_1$	U-net with skip connection	Single discriminator with Patch GAN
High-resolution	1024×1024	$L_{GAN} + \lambda_2 \cdot L_{FM}$	U-net with skip connection	Multi-discriminator with Patch GAN

The CraneGAN generator aims to generate a synthetic image with an appropriate tower crane layout. The training process of CraneGAN based on paired datasets is shown in Fig. 4. Each data pair includes a conditional image and its labeled one. The conditional image is the same with the extracted image in Part I. The labeled image adds the optimal tower crane layout while keeping the conditional image information. The input of the CraneGAN generator model is the conditional image (pending design drawings). The generator then generates a synthetic image with a predicted optimal tower crane layout as the output. Based on the paired dataset, the discriminator will score the degree of truthfulness of the synthetic image based on the corresponding labeled image and send a loss to the generator. Subsequently, based on the score, the generator updates its parameters in the neural networks to generate more realistic images to trick the discriminator further. After multiple training epochs, the generator achieves high-quality tower crane layout generation with a further reduced loss.

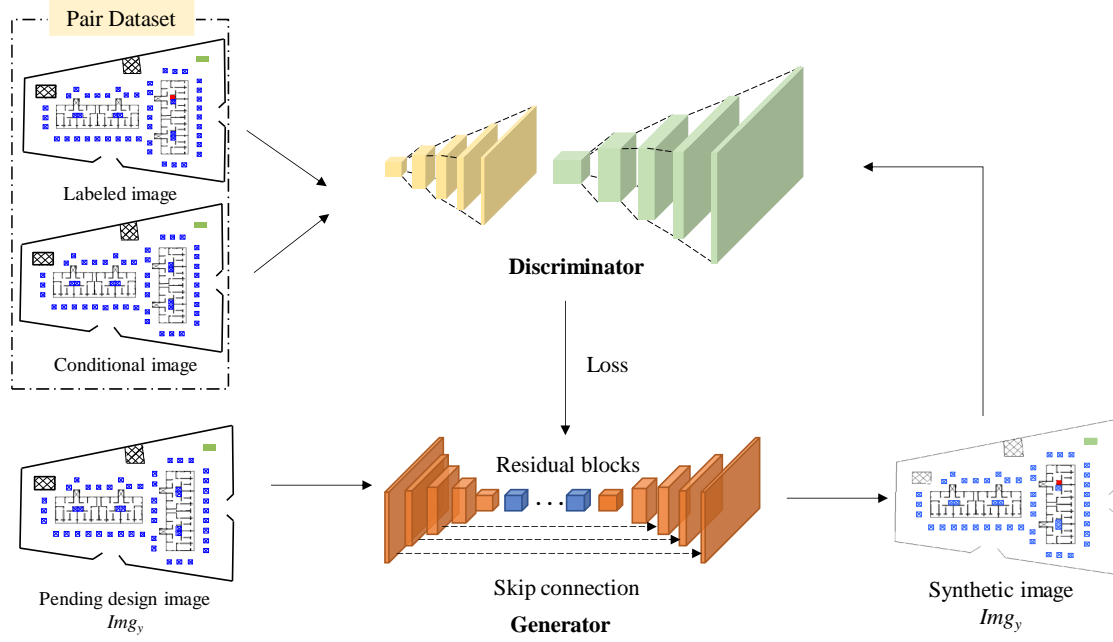


Fig. 4 CraneGAN training process

3.4.2 CraneGAN Network

The CraneGAN network consists of a loss function and neural network architectures for a generator and a discriminator in low- and high-resolution version. Based on the objectives of the generator and discriminator mentioned above, the loss function of CraneGAN is:

$$\min_G \max_D L_{GAN}(G, D) = E_{(x,y)}[\log D(x, y)] + E_{(x,z)}[\log(1 - D(x, G(x, z)))] \quad (1)$$

where G refers to the generator model, D refers to the discriminator model, x refers to the real dataset, y refers to the corresponding labels, and z refers to the synthetic image by the generator. It is set to maximize the discrimination and minimize the likelihood of truthfulness (probability distribution) between the real dataset and the synthetic image.

In previous related work, it was found that adding a traditional loss to the objective of the GAN can increase the accuracy and quality of the synthetic images [14]. Thus, losses L_1 and L_2 are usually used as the model evaluation criterion:

$$L_1(G) = \frac{\sum_{n=1}^n ||y - G(x, z)||}{n} \quad (2)$$

$$L_2(G) = \frac{\sum_{n=1}^n (y - G(x, z))^2}{n} \quad (3)$$

L_1 is the mean absolute error (MAE), which is the mean value of the distance between the predicted and actual values, and L_2 is the mean square error (MSE), which is the mean value of the square of the distance between the predicted and actual values. After selection, L_1 was chosen, as it yielded a clearer picture than L_2 . Therefore, for the low-resolution-version CraneGAN, the loss

function for the generator is as follows:

$$L_{CraneGAN-l}(G) = \arg \min_G \max_D L_{GAN}(G, D) + \lambda_1 \cdot L_1(G) \quad (4)$$

For the high-resolution version, the loss function is improved by adding a feature matching loss L_{FM} , which can stabilize the training [54], expressed as follows:

$$L_{FM}(G, D_K) = E_{(x,y)} \sum_{i=1}^T \frac{1}{N_i} \left[\left\| D_k^{(i)}(x, y) - D_k^{(i)}(x, G(s)) \right\|_1 \right] \quad (5)$$

Therefore, the generator loss in the high-resolution-version CraneGAN is:

$$L_{CraneGAN-h}(G) = \min_G \left(\max_{D_1, D_2} \sum_{k=1,2} L_{GAN}(G, D_K) + \lambda_2 \sum_{k=1,2} L_{FM}(G, D_K) \right) \quad (6)$$

where $D_k^{(i)}$ refers to the features in the i^{th} layer of the discriminator D_K , T refers to the number of total layers, N_i refers to the total number of elements in the layer, λ_2 is a crucial parameter that controls the importance of L_{GAN} and L_{FM} . Its value will be discussed later when adopting data augmentation approaches to achieve better performance.

In solving image translation problems, an encoder-decoder network is usually applied in the generator neural network architecture [54]. First, the network's input is processed through a series of layers with downsampling until a bottleneck layer, which is reversed with those upsampling afterward. To better transport the low-level information, U-Net was proposed, which has the same number of convolutional layers in the downsampling and upsampling processes with skip connections. The shallow downsampling layers can capture simple information about the image, such as the boundary and color. The receptive field of the deep layer is large, there are more convolution operations, and several abstract features are also captured. The skip connection between each layer and its corresponding mirror layer can directly deliver the feature of the downsampling layers to the upsampling layers. Therefore, in CraneGAN, the generator applies the U-Net network with a skip connection as the prominent architecture, which is feasible for feature reservation. The architecture is shown in **Error! Reference source not found.**

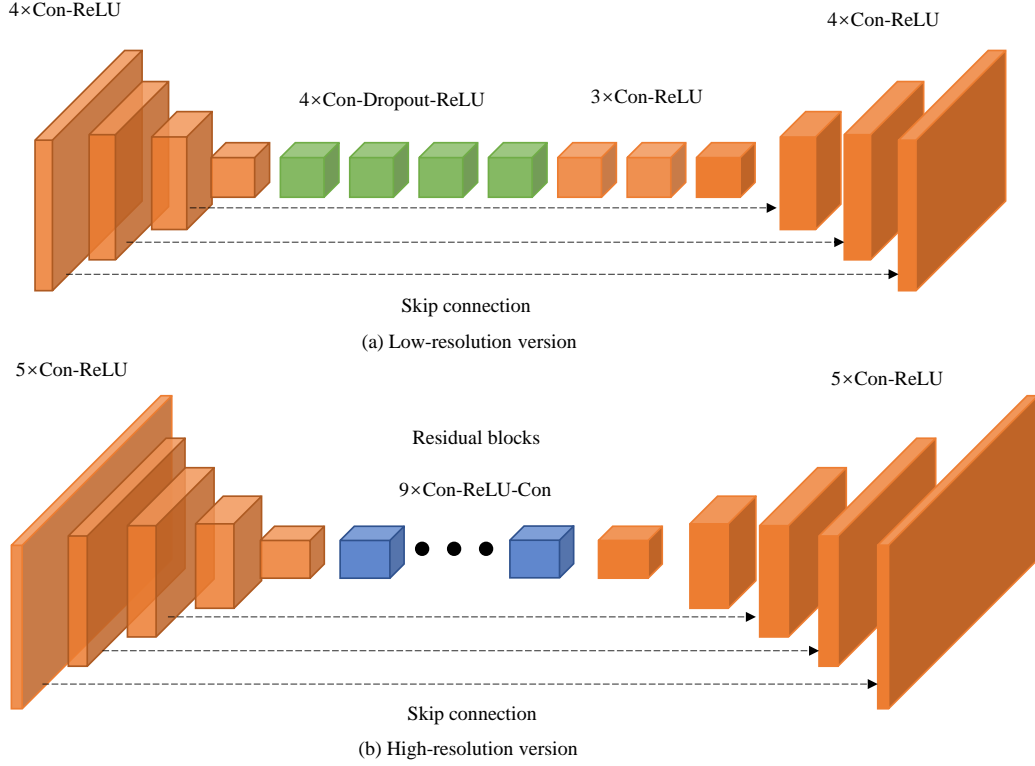


Fig. 5 Network architectures of CraneGAN generators

The network architecture of the CraneGAN discriminator is shown in Fig. 6. It applies a patch-based fully convolutional network named *PatchGAN*. Its task is to distinguish whether the image in an $N \times N$ patch is real or synthetic and score the corresponding patch of the image. Then, it averages all responses from the convolutional patches across the image to obtain the final output. Based on experiments in pix2pix [14], N is encouraged to be smaller than the total size of the image (256×256) to come out with high-quality results, as it needs fewer parameters, runs faster, and is suitable for most large images. When it comes to high-resolution images with sizes of 1024×1024 , a large receptive field is needed. The large receptive field needs a deeper network or larger convolutional kernels, which leads to a higher network capacity requirement and a larger memory footprint for training. Therefore, multi-scale discriminators were proposed in the high-resolution-version CraneGAN, where the difference only appears on the operated image scales. The larger-scale discriminator has the largest receptive field with a global view of the image, supervising the generator for generating a globally consistent image. The smaller one has the smallest receptive field with a detail preference for the generator.

CraneGAN applies a single discriminator for low-resolution images and a multi-scale discriminator for high-resolution images with *PatchGAN*. For the low-resolution version CraneGAN discriminator, there are four downsampling layers with a Con-ReLU structure and five layers with a convolution structure for the one-dimensional output. For the high-resolution version, there are five layers in the larger discriminator, with the first four layers as Con-ReLU, and the last layer is only a convolutional layer. For the smaller discriminator, there are six layers. Compared with the larger discriminator, the difference in the layer structure is in the last layer. It adds an average pool for downsampling after the convolutional layer.

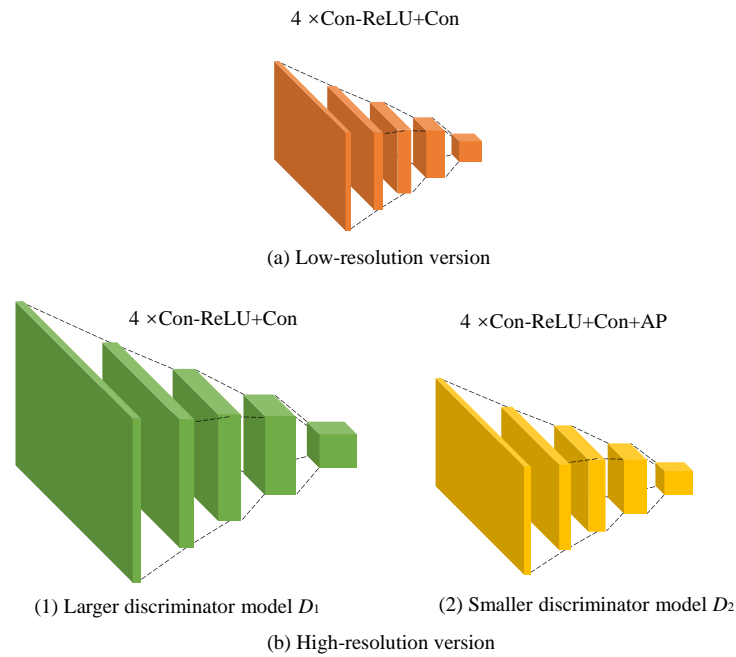


Fig. 6 Network architectures of CraneGAN discriminators

3.4.3 Dataset Generation

Training based on accurate and high-quality datasets is necessary for a data-driven approach to generate the required results [58]. Moreover, the dataset's quality is commonly evaluated from five aspects: availability, usability, reliability, relevance, and presentation quality [59]. As tower crane layout pairs are essential, this study needs to collect a high-quality dataset for tower crane layouts. Considering the limited availability and accessibility of datasets for TCLP in real-world construction projects, several steps to data collection with several standardized considerations are conducted. Firstly, construction projects that meet the requirements for building height and type are carefully selected. According to the *Code for fire protection design of buildings (2018 edition)*, high-rise building refers to residential buildings over 27m and non-single-story factory buildings, warehouses, and other civil buildings over 24m [66]. Subsequently, rigorous data preprocessing is executed on the selected projects, including standardizing the annotation of drawings to facilitate subsequent automated data extraction. The preprocessed drawings encompass vital architectural components related to tower crane positioning, such as roads, entrances, existing structures, construction boundaries, the installation locations of construction components, and so on. Lastly, during the process of obtaining label images, the same mathematical model is utilized to obtain the optimal tower crane layout based on actual construction projects, with minimizing transportation time as the optimization objective while simultaneously considering safety (e.g., safety distance between crane foundation and building boundary) and accessibility constraints (e.g., the land bearing capacity). This process reflects the availability and reliability of the dataset generation for tolerable generation time, reliable data source, and guaranteed accuracy.

There are four steps for generating the paired dataset with the optimal tower crane layouts based on the raw design data, as shown in Fig. . The first step is done in Part I, to obtain the pending

design drawings. Secondly, the relevant digital information in the pending design drawings is extracted as inputs for the mathematical model. Thirdly, the tower crane layout with the minimum transportation time is calculated and identified based on the known construction information and displayed as a full-filled red rectangle. Finally, the sizes of the images are cropped to meet the requirements of the different CraneGAN versions. The size of each input image for different training networks is specified to avoid tensor mismatch problems.

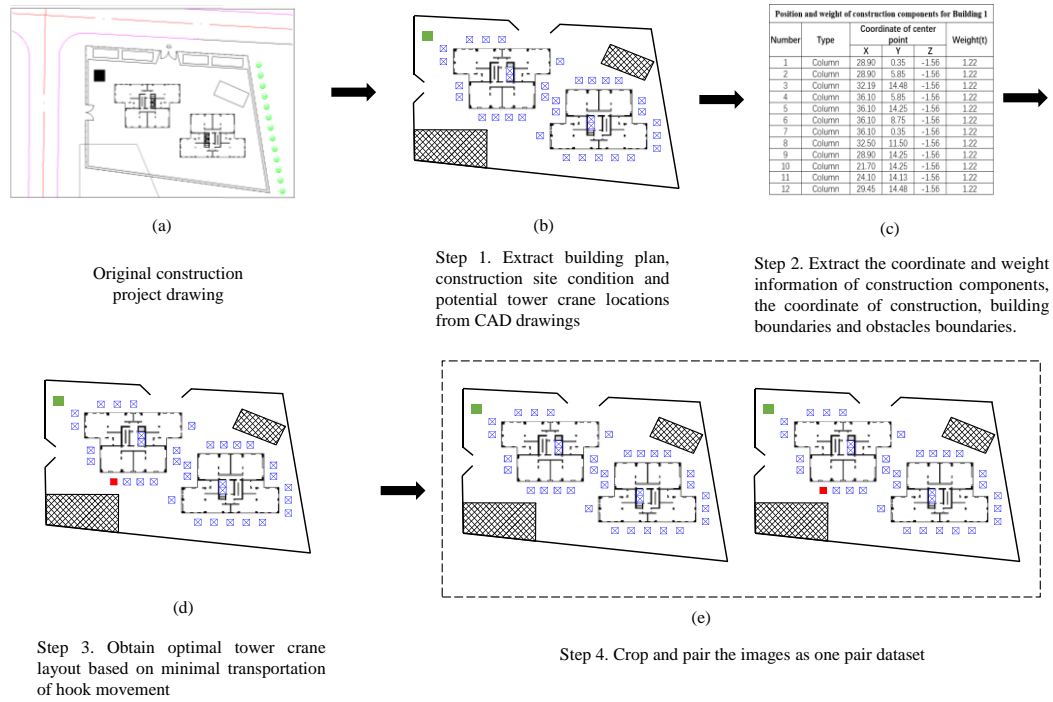


Fig. 7 Process of CraneGAN dataset generation

Crane transportation time denotes the temporal duration encompassing the transportation of construction components by the tower crane, commencing from the material storage area and culminating with the completion of unloading operations. There are two main reasons for selecting it as the optimal objective. Firstly, crane transportation time refers to the overall time required to complete all the transportation tasks for construction components, indicating the efficiency and productivity of crane transportation. Secondly, the choice of the objective function was influenced by the available data and the practical feasibility of implementation. Minimizing the cumulative time of transportation tasks is a widely used criterion in TCLP problem-solving and has demonstrated practical applicability in [8] and [65]. For each transportation task, the crane transportation time consists of loading, lifting, unloading, and idle time. The mathematical model is inspired by crane hook movement [61]. The hook movement of lifting tasks can be divided into horizontal and vertical movements, as shown in Fig. 8. The crane is located at K to deliver construction components from the supply area $S(x_s, y_s, z_s)$ to the demand area $D(x_d, y_d, z_d)$.

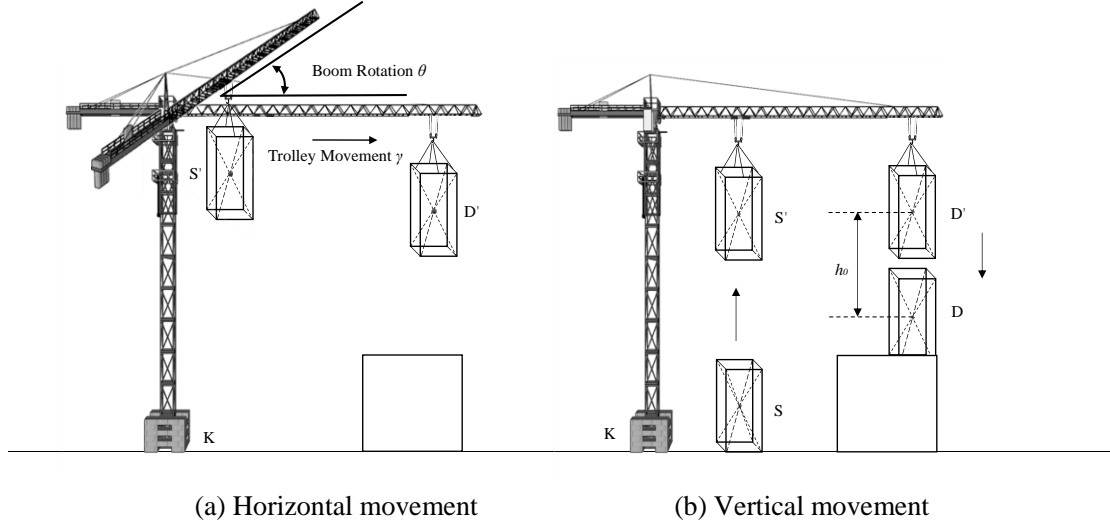


Fig. 8 Hook movement of a tower crane

The crane lifting time of hook's horizontal boom and trolley movements are calculated as follows:

$$T_{\theta} = \frac{\theta}{v_{\theta}} \quad (7)$$

$$T_{\gamma} = \frac{\gamma}{v_{\gamma}} \quad (8)$$

$$T_h = \alpha(\beta_{\gamma} \cdot T_{\gamma} + \beta_{\theta} \cdot T_{\theta}) + \max\{(1 - \alpha)(\beta_{\gamma} \cdot T_{\gamma} + \beta_{\theta} \cdot T_{\theta}), (\beta_{\theta} \cdot T_{\gamma} + \beta_{\gamma} \cdot T_{\theta})\} \quad (9)$$

where $0 \leq \theta \leq \pi$, T_{θ} is the time spent on the boom movement, θ refers to the angle of boom rotation in the horizontal direction, v_{θ} is the slewing speed of the jib (rad/min), T_{γ} is the time spent on the trolley movement in the horizontal direction, γ is the trolley moving distance between the supply and demand areas in the horizontal direction, v_{γ} is the radial speed of the trolley (m/min), and T_h is the time spent on horizontal movement. In Eq. (9), the parameter α , which is between 0 and 1, is the degree of simultaneous trolley movement and boom rotation horizontally. The parameters β_{γ} and β_{θ} represent whether the trolley movement or boom rotation occurred first. When β_{γ} equals 0 and β_{θ} equals 1, the boom rotation occurs earlier than the trolley movement. When the β_{γ} equals 1 and β_{θ} equals 0, the sequence of the movements is reversed.

The transportation time of the vertical hook movement is as follows:

$$T_v = \frac{|z_d - z_s| + 2h_0}{v_h} \quad (10)$$

where T_v refers to the time spent on vertical movement; h_0 refers to a safe height; and v_h refers to the vertical lifting speed with loaded construction components (m/min).

To calculate the total transportation time of the tower crane, the vertical movement is assumed to be able to coincide with the horizontal movement, and the degree of synchronization is reflected by the parameter γ , as follows:

$$T_t = \gamma \cdot T_v + \max\{(1 - \gamma) \cdot T_v, T_h\} \quad (11)$$

where T_t is the total time of the hook movement for lifting one construction component, T_v is the time spent on vertical movement, and T_h refers to the time spent on horizontal movement.

With the consideration of the crane loading, unloading, and idle time, the total time of a building hoisting scheme can be defined as follows:

$$T_{t_n} = \sum_{i=1}^I \sum_{j=1}^J \sum_{l=1}^L T_{t_{i,j,l}} + I \cdot J \cdot L \cdot T^u + I \cdot J \cdot L \cdot T^w \quad (12)$$

where T_{t_n} is the total transportation time of the whole construction process, I is the total number of components, J is the total number of building stories, L represents the total number of component types, T^u is the loading and unloading time, and T^w is the idling time. Based on the hook movement models, the optimal layout of the tower crane can be calculated based on minimizing the total transportation time. A further detailed explanation of the model can be found elsewhere in [61].

Instead of selecting the total schedule for transportation tasks, this study applied to minimize crane transportation times for the sum of individual transportation times as an objective function. The reason is the total schedule for transportation tasks required to coordinate the construction materials, labor, and machinery to complete the construction activity. Lack of any one of labor, materials, and machinery will delay the construction schedule. The total schedule cannot directly reflect the efficiency and productivity of the crane transportation. Moreover, in the actual transportation process, the crane loading, unloading, and idle time vary in different tasks, relying on various unpredictable factors (such as workers' experience and relevant construction schedule). Therefore, the transportation time is an appreciated number that refers to the exact time-consuming in actual construction, but it is reasonable to reflect the transportation productivity.

In conclusion, the main difference between CraneGAN and the traditional GAN lies in their specific applications and corresponding neural networks. CraneGAN is a variant of traditional GAN specifically designed to predict tower crane layout in construction projects, extract features from vast TCLP data, and make predictions in new construction project cases. For the neural networks, the distinction between CraneGAN and traditional GAN lies in the input mechanism, loss function, and architecture of their generator and discriminator neural networks. The input image of CraneGAN is the graphical information of construction projects as condition information instead of a simple noise vector. Also, CraneGAN is trained based on the paired dataset, suiting the characteristics of the TCLP problem for generating tower crane layouts using the corresponding graphical information of construction projects. Then, the loss function in different versions is improved by adding L_1 loss or feature matching loss L_{FM} . Also, the hyper-parameter for the loss function is fine-tuned to improve the prediction accuracy and quality. Finally, compared with traditional GAN, the generator of CraneGAN applied U-net with a skip connection to avoid the feature disappearing. Moreover, Patch GAN was used to accelerate the training process with fewer parameters in the discriminator [14] [54].

3.5 Part III: Identification

Given the uncertainties of generative processes, the results from CraneGAN sometimes need further post-processing to identify the tower crane layout and enhance image clarity. In addition, noise may influence the regions of the potential layout areas, and color intensities may be rendered insufficient. Therefore, as a post-processing process shown in Fig. 9, Part III identifies the deviation of pixel color intensity between generated drawings Img_z and pending design drawings Img_y to generate comprehensive and high-quality layout results Img_{zen} . The red color is used to represent the suggested optimal crane layout location. Its intensity, ranging from 125 to 255, was selected. If there are no or fewer red pixels detected in the generated results, the failure generation flag is raised to the generator in Part II. Then, the generator re-generates a new image and sends it back again. Also, the pixels of the potential crane location options are recognized by their blue colors in the generated results. The blue intensity, which ranges from 150 to 255, was selected. The location of the predicted tower crane layout (in red color) can be further consolidated by calculating and identifying the minimum distance between the group of red pixels and those in blue. The center point of one of the blue pixel groups, which got the closest distance to that of the red pixels, can be selected as the accurate prediction result location. Finally, such identified tower crane layout is re-drawn as a red square in the post-processed image to present the results.

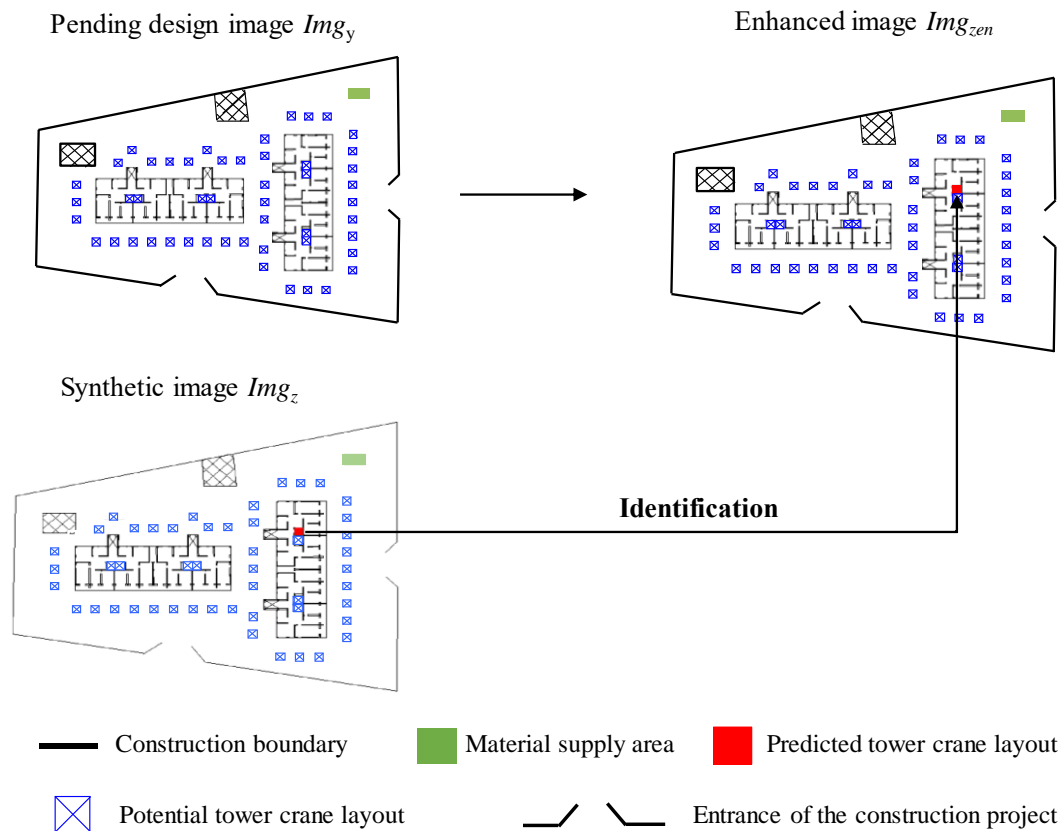


Fig. 9 The process of Part III: Identification

4 Implementation and Performance Evaluation

4.1 Environment Setting

The training, testing, and validation process of the proposed automatic TCLP system were all conducted on a server running the Windows 10.0 x64 operation system. The specifications and configuration were as follows:

- CPU: 11th Gen Inter® Core™ i9-11900@2.5GHz;
- RAM: 64 GB of memory;
- GPU: NVIDIA GeForce RTX 3090 Ti and NVIDIA Tesla P100;
- Hard Disk: 512-GB SSD and 10-TB HDD;
- Run-on CPU: None;
- Run-on GPU: CraneGAN, pre- and post-processing, rendering.

For the CraneGAN implementation, the algorithms were developed using Python 3.6 and PyTorch 1.10.0 as a base deep learning development framework.

4.2 Implementation

4.2.1. Dataset Generation

The dataset in this study consisted of 1884 pairs of building drawings and corresponding layouts from open-sourced architectural design and research institutes in China. It included 1648 images of building plans in the training set and 236 in the testing set. The CraneGAN in low- and high-resolution versions were trained and evaluated individually. The input image size with a low resolution was 256×256 , and that for a high resolution was 1024×1024 .

Each pair of images in the corresponding dataset were composed of conditional images and labeled images, as shown in

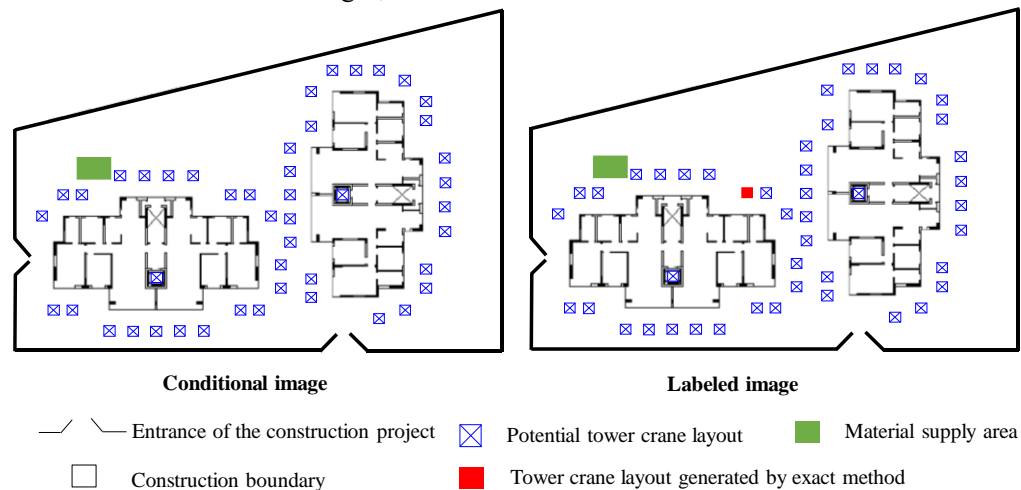


Fig. . In the images, the filled green rectangle refers to the location of the material supply area. And the blue rectangles with center crosses represent the potential layouts of the tower crane for the high-rise building. The solid red rectangle refers to the optimal location of the tower crane, considering the minimum transportation time for the given material supply and building plan. The black objects

represent the construction boundary. The CraneGAN was trained to generate an image with maximum similarity to the corresponding labeled image.

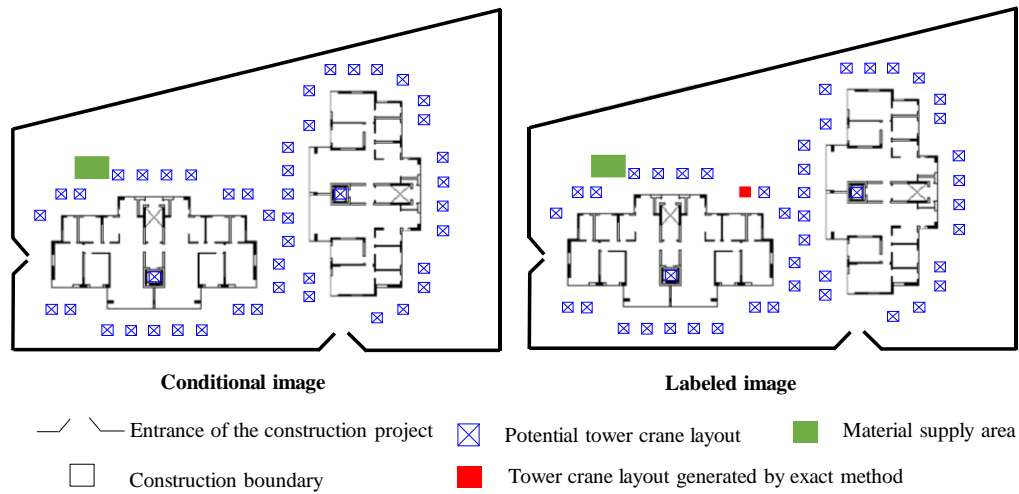


Fig. 10 Paired dataset for CraneGAN training

4.2.2. CraneGAN Training Results

In CraneGAN training, the batch size, learning rate, and epochs were set as 1, 0.0002, and 200, respectively.

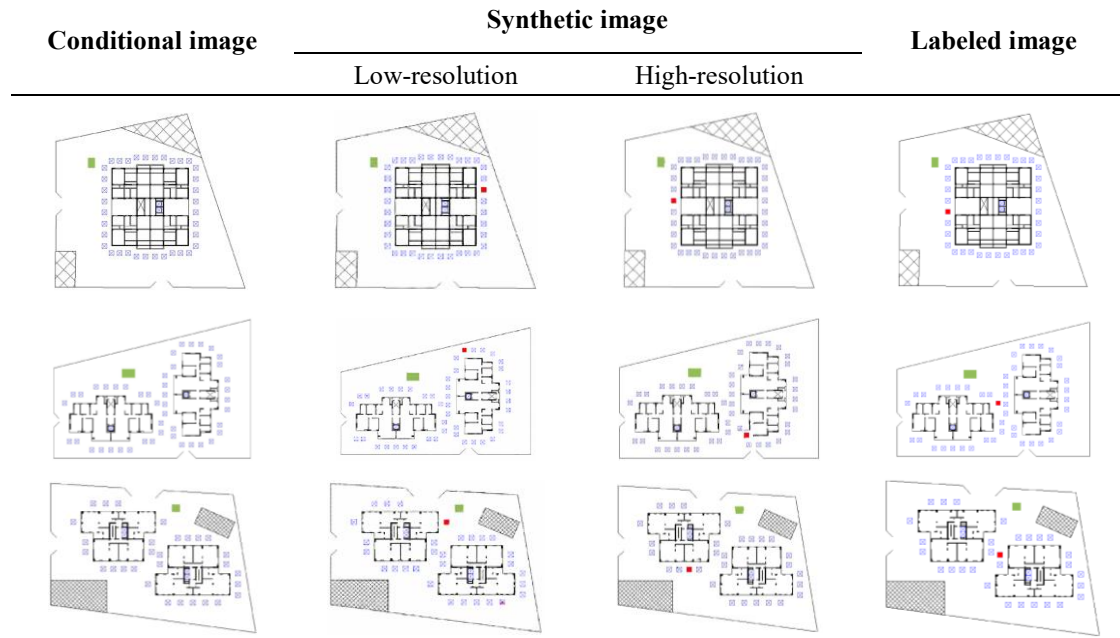


Fig. shows the resulting examples of the CraneGAN output. The conditional images as input images are listed in the first column. Their generated synthetic images are in the middle column, followed by labeled images identified with the optimal layouts of the tower crane in the third column. CraneGAN could generate reasonable layouts for the attached tower crane, matching the optimized images in both low- and high-resolution versions. The size of the synthetic image was the same as that of the input image. Also, the synthetic images generated by the high-resolution-version CraneGAN were clearer than those by the low-resolution-version CraneGAN.

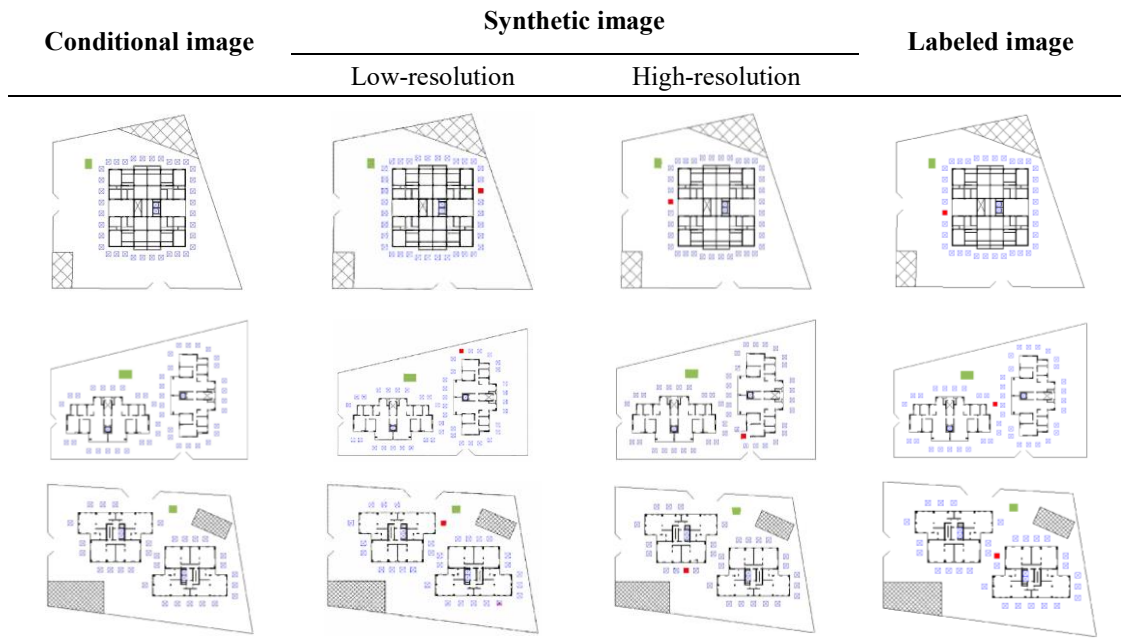


Fig. 11 Examples of CraneGAN training results

A computer-vision-based evaluation was selected to assess the quality of the synthetic images from CraneGAN on the test dataset. The similarity between the synthetic image and the lable image were evaluated. For the CraneGAN configurations, the generators with λ_1 equal to 30 and λ_2 equal to 5 after data augmentation were selected for the low-resolution and high-resolution versions, respectively.

To evaluate the accuracy of the synthetic images, pixel accuracy (PA), a notable image accuracy assessment metric commonly used in CNN applications, was selected [63]. PA is the number of pixels with the correct prediction category as a proportion of the total number of pixels derived from the confusion matrix shown in Table 2 [64]. In the confusion matrix, each column represents the predicted value (synthetic image), and each row represents the conditional image. Thus, the calculation of PA is as follows:

$$PA = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

Table 2 Confusion matrix for pixel accuracy (PA)

Confusion matrix		Synthetic image	
		Positive	Negative
Conditional image	True	TP	FN
	False	FP	TN

Note: TP represents true positive; FN represents false negative; FP represents false positive; and TN represents true negative.

Thirty images from the test dataset were selected to evaluate the quality of the synthetic image from CraneGAN, as shown in **Error! Reference source not found..** The average and standard

deviation in the high-resolution version were slightly better than those in the low-resolution version. However, the average PAs of CraneGAN in the two versions were all above 98%, and the standard deviations were all below 0.25%, revealing the high accuracy and quality of the synthetic images.

Table 3 PA results from CraneGAN training

	Number of images	Average	Standard deviation
Low-resolution	30	99.23%	0.22%
High-resolution	30	99.15%	0.14%

4.3 Performance Evaluation

4.3.1. Computational Time

When solving TCLP optimization problems, exact analytical methods, which search exhaustively for an optimal solution, can be considered for a precise result. Alternatively, heuristic algorithms can be adopted to save time finding near-optimal solutions. The exact analytical method was used to build the baseline for TCLP problem solving, and GA was selected as a comparative method for its widespread use and effectiveness in solving TCLP as an operation research solver, as demonstrated in those research [10][67]. Therefore, the exact analytical, GA, and CraneGAN methods were separately applied in different construction projects for obtaining TCLP, and the computational time was recorded. The computational time for the exact method was selected as the baseline to reflect the improvement of using GA and CraneGAN. For the GA method, the initial population size was set to 100, and the length of the chromosomes was 2, which refers to the x and y coordinates of the tower crane layout. The evolution algebra was set to 100, referring to the termination condition. The crossover probability was set to 0.7, and the mutation probability was set to 0.3. The objective function was the same as Eq. (12) for calculating crane transportation time. Moreover, the algorithm iterates through the selection, crossover, mutation, and replacement steps until a termination condition is met. The proposed CraneGAN used in the comparison (as shown in Table 4 and Table 5) was the high-resolution version with data augmentation, and the hyper-parameter value was set to 5. Related data augmentation details are discussed in Section 5.

Table 4. Computational time in using the exact analytical method, genetic algorithm (GA), and CraneGAN to solve TCLP problems

Image No.	Computational time (s)		
	Exact method	GA	CraneGAN
1	30.58	10.52	2.18
2	26.62	9.09	2.04
3	26.71	9.18	1.85
4	41.36	9.50	2.06
5	48.01	9.38	2.34
6	37.67	9.26	2.41
7	46.07	9.66	2.23
8	52.25	9.72	1.96
9	51.29	9.38	1.94
10	37.70	9.98	2.45

11	41.00	9.56	2.08
12	45.74	9.98	2.13
13	32.32	9.61	2.33
14	34.72	9.26	2.42
15	23.94	9.03	1.95
Average	38.39	9.57	2.16
Standard deviation	8.85	0.42	0.19

To reduce the error caused by the sampling randomness, 15 typical building plans with different layout complexities were selected for evaluation. As shown in Table 4, the average computational times were 38.29, 9.57, and 2.16 s when using the exact analytical method, GA, and CraneGAN, respectively. Using CraneGAN could significantly reduce the computational time by 5.64% and 22.57% compared to using the exact analytical method and GA. The potential tower crane location is related to the building boundary, which affects the computational time of the exact analytical method when the number of potential crane locations is increased. Moreover, the exact method and the GA were all based on known building component locations, weights, and related properties, which required further time-consuming data collection processes. These were even excluded from the current computational time comparison.

Regarding the time consumption standard deviation, the highest to lowest was in the order of the exact analytical method, GA, and CraneGAN. The reason was that the standard deviation was affected by the variation in the intensity and boundaries of constructions. The construction's intensity and the boundary's size were proportional to the number of delivery tasks and potential tower crane locations, which required different computational times. The comprehensive search algorithm was most severely affected. Therefore, the CraneGAN and GA performances were more stable than that of the exhaust traversal algorithm in the exact method. These results proved that CraneGAN could efficiently and stably generate a nearly optimized tower crane layout, which was even superior in time consumption to that of the GA.

4.3.2. Crane Transportation Time

The crane transportation time is an essential metric for evaluating the performance of a TCLP. As shown in Table 5, using the exact analytical method could identify the maximum and minimum transportation times. Then, the maximum transportation time was selected as the baseline for evaluating the results of GA. The difference ratio was the ratio of the difference between the transportation time obtained using the corresponding method and the minimum transportation time.

Table 5. Transportation time and difference ratio using the exact analytical method, GA, and CraneGAN to solve TCLP problems

Image No.	Exact analytical method		GA	CraneGAN	
	Maximum transportation time	Minimum transportation time	Transportation time	Transportation time	Difference Ratio with GA (%)
1	3531.70	2185.50	2155.50	2253.60	4.55

2	3253.60	2324.40	2407.40	2752.50	14.33
3	4653.70	1253.70	2698.40	1373.22	-49.11
4	2255.60	1231.30	2762.70	1234.54	-55.31
5	2311.50	1524.10	1737.50	1924.69	10.77
6	3048.90	1465.00	1908.30	1416.22	-25.79
7	3676.10	1683.70	1752.30	2313.22	32.01
8	2965.40	1430.70	2165.70	1566.92	-27.65
9	3311.50	1098.80	2050.20	2000.76	-2.41
10	2747.50	1349.30	1390.60	1547.63	11.29
11	3605.50	1063.30	2360.00	1646.11	-30.25
12	1317.10	930.52	2587.10	1949.47	-24.65
13	2549.30	1140.00	2628.80	1282.87	-51.20
14	1547.70	783.62	2039.70	852.77	-58.19
15	2503.60	842.56	2358.20	1613.50	-31.58

In **Error! Reference source not found.**, 93.33% of the results obtained by CraneGAN were within positive 15% of the difference ratios of the results obtained using the GA algorithm, which was also not far from the minimum transportation times. In 73.33% of the cases, the difference ratio between the results did not exceed 5% of the total transportation time. In addition, in 66.67% of the cases, the results using CraneGAN outperformed the results using the GA. Hence, the quality of the CraneGAN performance was tolerable, and a more stable and robust outcome can be anticipated when facing highly complex layout planning cases.

5 Numerical analysis

A real construction project was utilized to validate the proposed automatic TCLP system. The tower crane layout based on the minimum transportation time was generated using the exact analytical method, the GA optimization, and the proposed system with appropriate hyper-parameter adjustments. The generated results were demonstrated by comparing them with the actual tower crane location in practice.

The selected project is a public housing construction project in Hong Kong, as shown in Fig. 12. It consists of five blocks of high-rise buildings with heights of around 120 m. Block 7, the highest building, is selected as the case to obtain the optimal tower crane layout. It covers an area of 72×57 m and 123.8 m in height, and one single tower crane is required for essential lifting tasks. The tower crane is supposed to be located at the elevator shaft or the surrounding area of the building, which is 3 m outside the building boundary, based on practical experience.

The application procedure of the proposed TCLP system is illustrated in Fig. 12. The construction conditions are shown in Fig. 12(a). After drawing extraction, the conditional image for the TCLP was generated, as shown in Fig. 12(b). The extraction processes required approximately 5 mins to obtain the design images as input for CraneGAN. Compared with the necessary numerical data collection from related construction documents, applying the proposed TCLP system with CraneGAN on images directly reduced the conventional data fetching and processing time, which took 2 to 3 hours to complete for this building. As for the layout generation, it took about 3 seconds to load a pre-trained generator and generate the results, as shown in Fig. 12(c). For the result post-

processing and identification, it took 6 seconds to identify the generated tower crane layout. The final layout was thus confirmed, as shown in Fig. 12(d).

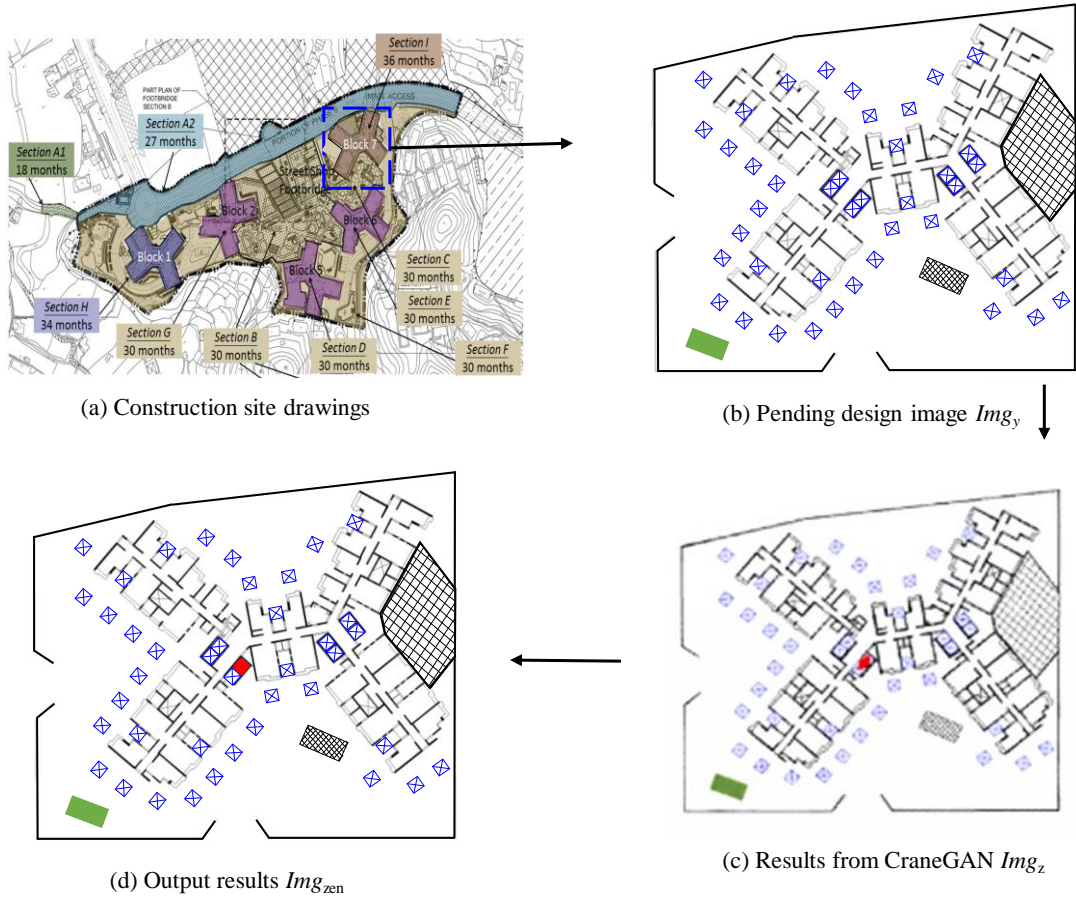


Fig. 12 Application procedure of the proposed TCLP system

The TCLP results obtained using the proposed CraneGAN, GA, exact analytical method, and actual layout determined in this construction project were collected and displayed in

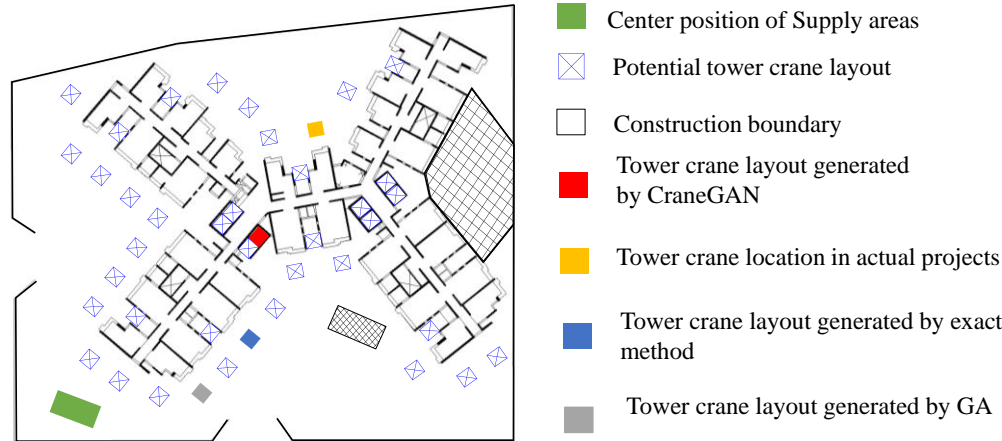


Fig. . The CraneGAN's result was located within the potential tower crane locations, which meets the requirements for this project. The performance of the computational and transportation times is shown in Table 6. The computational times for applying the exact analytical method, GA, and CraneGAN were 33.24, 10.75, and 2.33s, respectively. In comparison, the proposed system

showed better and more stable performances, with the lowest computational time. The transportation times, which reflect the design quality, when using the exact analytical method, GA, and CraneGAN were 3386.92, 5472.31, and 5039.24 per standard floor, respectively. As for the actual tower crane layout, the calculated transportation time was 8265.25. Compared with the actual location in the project, CraneGAN improved the transportation time by 39.03%. This was a further 7.91% improvement in transportation time compared to that of the GA.

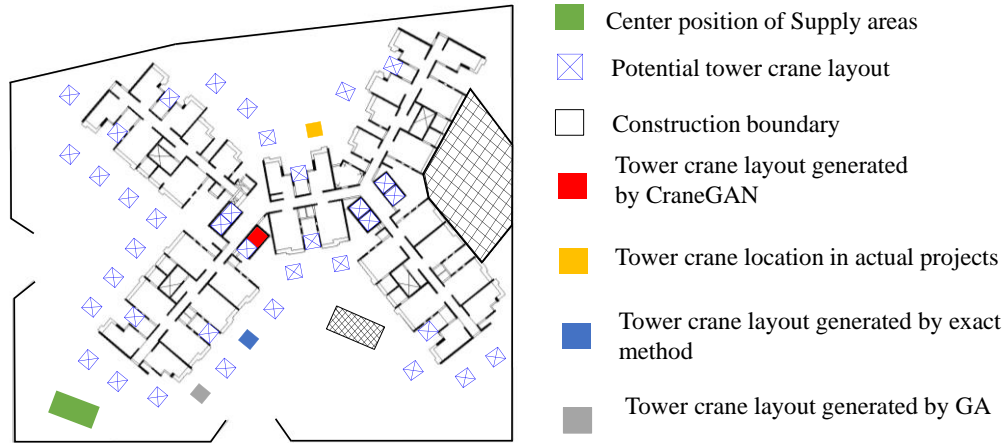


Fig. 13 Layout results from different TCPL approaches and the actual layout in the project

Table 6 Transportation and computational time comparison of different TCLP approaches and the actual layout in the project

	Exact method	GA	CraneGAN	Actual
Transportation time	3386.92	5472.31	5039.24	8265.25
Computational time (s)	33.24	10.75	2.33	/

To accelerate and easy the process of application, several distinctive aspects set the automatic CraneGAN-based TCLP system apart from previous research in terms of computational burden. Firstly, the proposed system reduces this burden by utilizing drawing input as the foundation for tower crane layout generation, eliminating the need for manual information extraction. This eliminates extensive manual data collection and preprocessing, resulting in a streamlined computational process. Moreover, although the proposed system necessitates a certain level of computational effort for training neural networks, exceeding that of other mathematical TCLP methods, CraneGAN effectively redistributes this burden and maintains a relatively minor and consistent level of calculations in practical applications. This efficient allocation mitigates the problem of calculation explosion. The performance evaluation of transportation time and computational time validates this approach. Moreover, this data-driven method can be readily applied to extensive data from digital twins or generate training scenarios for TCLP, offering valuable support to inexperienced decision-makers.

6 Discussion

To obtain a higher performance of CraneGAN, improvements were conducted targeting the

dataset and CraneGAN hyper-parameters to further identify their abilities in solving TCLP problems. The analyses of hyper-parameters and applying data augmentation in the dataset were presented in this section. In addition, the influences of construction layouts with different complexities were also discussed.

6.1 Hyper-Parameter Adjustment

In the loss function, λ_1 and λ_2 are two important hyper-parameters that need to be adjusted to realize better performances of CraneGAN, details shown in Section 3.4.1. Therefore, obtaining suitable values of these two hyper-parameters is necessary for TCLP problems. The results of applying different values are shown in Table 7 and Table 8.

Table 7 Comparison of results using different λ_1 for low-resolution dataset

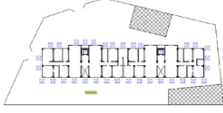



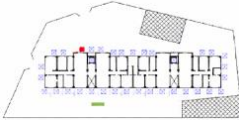


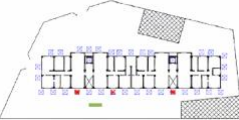
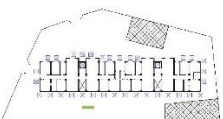
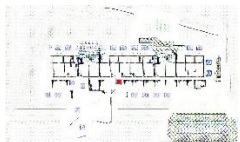


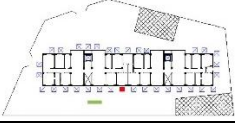
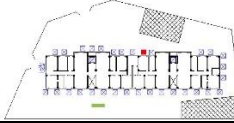
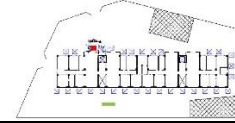
			
Input	$\lambda_1 = 0$	$\lambda_1 = 10$	$\lambda_1 = 30$
Transportation time	2963.72	2772.77	2963.86
Computational time (s)	1.98	2.15	1.86
			
$\lambda_1 = 50$	$\lambda_1 = 100$	$\lambda_1 = 120$	$\lambda_1 = 150$
3243.12	4481.14	4571.83	/
2.04	2.03	2.15	2.17

Table 7 shows the results of specifying different values of λ_1 for the loss function of the low-resolution-version CraneGAN dataset. When λ_1 equaled 0, 10, 30, 50, 100 and 120, proper tower crane layouts were generated, and the outcomes were stable, with tolerable image quality. As λ_1 increased to over 120, the generator showed that it failed to predict the tower crane layout, while the other information, like the construction boundaries, was learned. Therefore, as illustrated in Eq. (4), using too high of the L_1 weight in the loss function is not appropriate for improving the image quality in the low-resolution-version CraneGAN.

Table 8 Comparison of results using different λ_2 for high-resolution dataset

			
Input image	$\lambda_2 = 0$	$\lambda_2 = 1$	$\lambda_2 = 3$

Transportation time	/	/	/
Computational time	2.53	2.45	2.87
(s)			
			
$\lambda_2 = 5$	$\lambda_2 = 10$	$\lambda_2 = 15$	$\lambda_2 = 20$
2772.82	2735.71	3243.17	/
2.86	2.75	2.43	2.68

As shown in **Error! Reference source not found.**, when λ_2 equaled 0, the synthetic image showed that the generator could not learn the global view of the input image, and the image quality was compromised, resulting in massive noise. As λ_2 equaled to 5 and 10, the quality of the synthetic image gradually improved until it reached a plateau, with considerable improvements in the image clarity and tower crane layout design logic. When λ_2 reached 15, the prediction failed again. Therefore, as illustrated in Eq. (6), adding feature matching in the loss function was necessary to improve the learning degree of the tower crane layout design logic at particular scales of λ_2 . Nevertheless, the image clarity in the high-resolution version was better than that in the low-resolution version.

Table 9 Results of success prediction rate, computational time, and average transportation time improvement for CraneGAN hyper-parameter adjustments

		Success prediction rate (%)	Average computational time (s)	Average transportation time improvement (%)
λ_1	0	100.00	2.15	9.81
	10	100.00	1.45	18.91
	30	83.33	1.84	13.780
	50	82.76	1.83	8.92
	100	96.55	1.76	8.67
	120	80.00	2.24	19.53
	150	100.00	2.00	11.130
λ_2	0	0	/	/
	1	0	/	/
	3	33.33	2.66	30.46
	5	96.67	2.85	14.27
	10	86.67	2.68	22.54
	15	100	2.74	16.27
	20	83.33	2.84	22.61

Note: The success prediction rate refers to the tower crane being located in potential locations. The average transportation time improvement treated the transportation time calculated in the GA as a baseline.

To further determine the performances with different λ_1 and λ_2 , 30 typical images in the dataset were selected for prediction and statistical analysis. The results are expressed in Table 9, including the success prediction rate, average computational time, and average transportation time

improvement for different λ_1 and λ_2 in the CraneGAN loss function. In terms of the success prediction rate, when λ_1 was 0, 10, 100, or 150 and λ_2 was 1, 5 or 15, the success prediction rate was higher than 90%, reflecting a stable prediction performance through generators of CraneGAN. For all combinations of λ_1 and λ_2 , the computational times were all within 3 s, indicating that the results could be generated in a tolerable amount of time. When λ_1 was 120 and λ_2 was 10, the transportation times were improved by the highest percentages in both the low-resolution-version and high-resolution-version CraneGAN compared with those of the GA. However, the success prediction rate when λ_1 was 100 was 96.55%, indicating that in a few cases, it was challenging to produce prediction results. Therefore, the generator with λ_1 equal to 10 should be applied for the low-resolution-version CraneGAN to guarantee prediction stability and superiority. The generator with λ_2 equal to 10 and 15 were selected in the high-resolution version. λ_2 equal to 120 will be firstly used to predict tower crane layout planning to pursue higher construction efficiency. Once the corresponding generator cannot predict suitable results, the generator with λ_2 equal to 10 can be applied alternatively for stable prediction.

6.2 Data Augmentation

Data augmentation is a common method for increasing the performance of GAN generators. Manipulating existing datasets by adding disturbances (such as image panning or rotating) can facilitate the neural networks identifying and extracting key features in data [60]. This study expanded the training dataset from 312 to 1648 images by vertically and horizontally flipping and rotating images 180 degrees to generate more image instances in the dataset. Such data augmenting technique was implemented on the CraneGAN to achieve further enhancements. As shown in Fig. 14, the success prediction rate after the data augmentation increased when λ_1 equaled 50, 100 and 150, stayed flat when it was 10, and declined when had other values in the low-resolution version CraneGAN. In terms of the high-resolution version, when λ_2 equaled 3, 5, 10 and 15, the success rate after data augmentation showed a consistent increase, while it was worse only when λ_2 was 15.

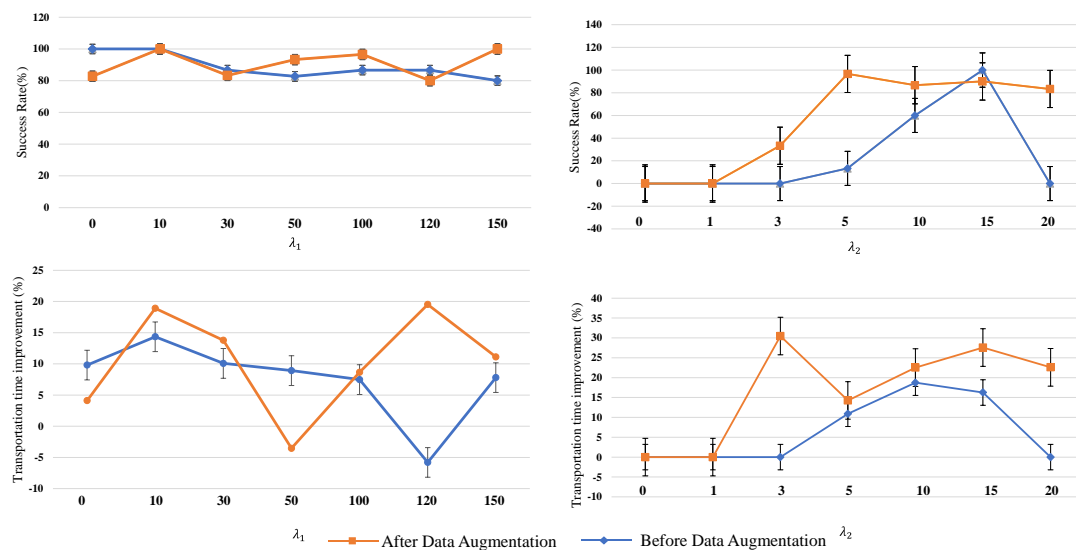
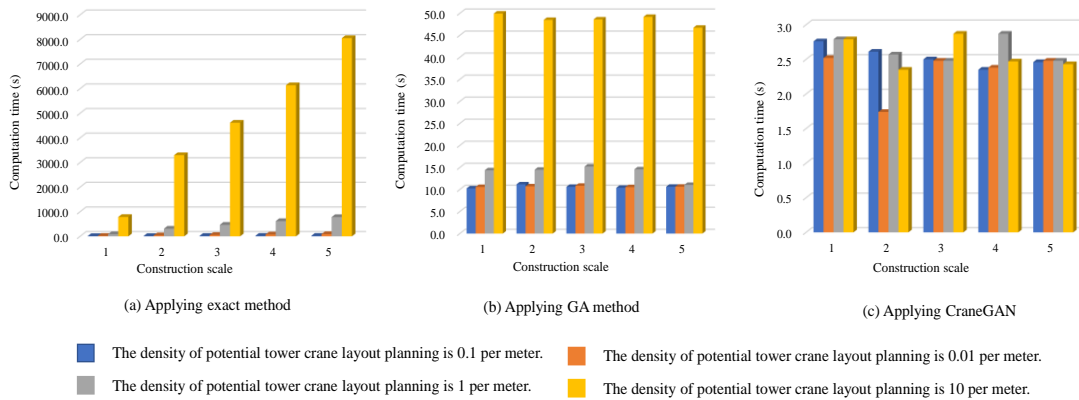


Fig. 14 Prediction results before and after the data augmentation for CraneGAN

Fig. 14 also shows the results of the average ratio of transportation time improvement before and after augmentation. The ratio of the difference between the generated results before and after data augmentation to that before data augmentation is referred to as the transportation time improvement. When λ_1 equaled 10, 30, 120 and 150, the average ratios of transportation time improvement were positive, reflecting better performance in the low-resolution version CraneGAN. As for the high-resolution version, except λ_2 equaled 0 and 1, the performance of transportation time improvement positively increased after data augmentation. Hence, these results indicated that data augmentation could enhance CraneGAN as long as the hyper-parameters are appropriately selected, especially for the high-resolution version.

6.3 Complexity

When facing large-scale construction projects, the proliferation of viable tower crane locations and the associated information pertaining to the construction project (such as the augmentation of construction components) undergo an exponential surge. Consequently, this surge gives rise to challenges associated with time complexity, causing the computational time to rise when using traditional mathematical methods. Hence, this section discusses the influence of CraneGAN under different case complexities, comparing with those applying the exact analytical method and GA. As shown in Fig. 15, constructions under five site scales and four resolutions in layout planning search space were selected for comparison. The TCLP computational time using the exact analytical method showed a significant increasing trend as the building scale increased, ranging from 10.90 to 8052.77 s. The computational time of using GA reflected a gentle rise from 10.254 to 49.954 s, while the proposed CraneGAN exhibited a stable performance of generating results in around 2.5 s. These results indicated that the proposed system could ease the exponentially increasing time consumption for large-scale construction and improve decision-making efficiency. Distinguished from conventional operations research methods, the proposed CraneGAN leverages image-based approaches for TCLP, offering complementary advantages to operations research methodologies. CraneGAN provides a viable solution to mitigate the computational challenges arising from large volumes of data to offline processing, thereby addressing the issue of online calculation explosion.



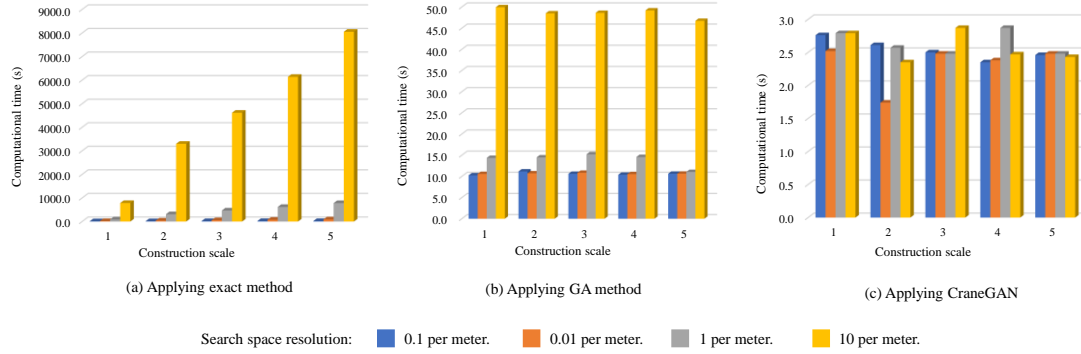


Fig. 15 Results of computational time under different scale complexities and search resolutions

7 Conclusion

In summary, this study introduced an image-based automatic tower crane layout planning (TCLP) system for high-rise buildings. CraneGAN was developed to generate reasonable and efficient tower crane layouts based on image input instead of manually interpreted data. The transportation and computational times were selected as metrics for the performance evaluation, and a numerical analysis was conducted to validate the proposed system. Compared with applying the exact analytical method and the genetic algorithm (GA), the proposed system prevents the manual data extraction process, consuming around 2 hours in the study case. In addition, the proposed system improved the transportation time by 39.03% compared to the actual layout plan in the project, which is a 7.91% improvement of GA's results. The computational time was also remarkably reduced, requiring only 6.02% and 21.68% of the consumptions by the exact analytical method and GA.

Additionally, after training generators based on various influencing factors λ_1 and λ_2 , the generator of λ_1 equal to 10 and a combination of λ_2 equal to 10 and 15 after data augmentation were selected in CraneGAN, as they achieved a 100% success prediction rate and improved the average transportation time by more than 10%. Data augmentation was used to improve the performance of CraneGAN based on adequately configuring the hyper-parameters. The impact of the ever-increasing complexity on the layout planning tasks was discussed through a performance comparison between the exact analytic method, GA, and CraneGAN. The outcomes of the proposed CraneGAN showed a comparatively stable performance and acceptable quality when facing large-scale construction cases.

This study makes contributions in four key aspects. Firstly, the proposed automatic TCLP system introduces a novel approach as the first GAN-based solver for TCLP, making decision based on image input. Secondly, the CraneGAN model offers a scalable layout prediction network that accommodates various constraints and optimal objectives, allowing customization based on specific construction requirements. Thirdly, the data-driven automatic TCLP system addresses the time complexity challenges prevalent in large-scale construction projects, reducing into polynomial processing time while maintaining acceptable optimality levels. Lastly, this research expands the theoretical foundation for intelligent layout planning and contributes to the advancement of intelligent construction management in digital twin environments and related training systems. It provides managers with efficient and effective decision support, enabling enhanced decision-making capabilities.

For future studies, improvements on CraneGAN are possible. Several influencing factors were

not discussed in this study, such as environmental conditions, which can be expanded by adding constraints or changing objective functions while generating the paired dataset. In addition, despite the limited availability and accessibility of real-world large construction project data in this dataset, there exists the potential for gradual expansion of the application scope of the proposed system. Furthermore, the uncertainty and sensitivity of the GAN-based methods are significant, and an efficient approach to identify proper hyper-parameter configurations is encouraged. Moreover, there is potential to expand upon the current study by developing a comprehensive simulation framework encompassing the entire construction process, leveraging the provided tower crane positioning information. Finally, the proposed automatic TCLP system could also be applied to other construction facilities' layout planning once their crucial constraints and preferred planning style have been determined.

Acknowledgment

This work was supported by the Research Grants Council of Hong Kong (grant number PolyU 25221519). Also, we would like to thank Mr. Wang Baokang for helping to extract the information from the construction drawings, which is an essential process to build up the dataset.

References

- [1]. Cai, S., Ma, Z., Skibniewski, M. J., & Bao, S. (2019). Construction automation and robotics for high-rise buildings over the past decades: a comprehensive review. *Advanced Engineering Informatics*, 42, 100989. doi: 10.1016/j.aei.2019.100989
- [2]. Zhang, Z., & Pan, W. (2021). Multi-criteria decision analysis for tower crane layout planning in high-rise modular integrated construction. *Automation in Construction*, 127. doi: 10.1016/j.autcon.2021.103709
- [3]. Cheng, Z., & Hammad, A. (2012). Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency. *Advanced Engineering Informatics*, 26(2), 396-410. doi: 10.1016/j.aei.2012.01.003
- [4]. Rider Levett Bucknall Company, (2023, April 9). RLB CRANE INDEX NORTH AMERICA Q1 2022- Q1 2023, <https://www.rlb.com/americas>
- [5]. Ji, Y., & Leite, F. (2020). Optimized Planning Approach for Multiple Tower Cranes and Material Supply Points Using Mixed-Integer Programming. *Journal of Construction Engineering and Management*, 146(3). doi:10.1061/(ASCE)CO.1943-7862.0001781
- [6]. Moussavi Nadoushani, Z. S., Hammad, A. W. A., & Akbarnezhad, A. (2017). Location Optimization of Tower Crane and Allocation of Material Supply Points in a Construction Site Considering Operating and Rental Costs. *Journal of Construction Engineering and Management*, 143(1). doi:10.1061/(ASCE)CO.1943-7862.0001215
- [7]. Lin, J., Fu, Y., Li, R., & Lai, W. (2020). An Algorithm for Optimizing the Location and Type Selection of Attached Tower Cranes Based on Value Engineering. *International Conference on Construction and Real Estate Management 2020*. doi: 10.1061/9780784483237.013
- [8]. Hussein, M., & Zayed, T. (2021). Crane operations and planning in modular integrated construction: Mixed review of literature. *Automation in Construction*, 122. doi: 10.1016/j.autcon.2020.103466

- [9]. Furusaka, S. (1984). A model for the selection of the optimum crane for construction sites. *Construction Management and Economics*, 2(2), 157-176. doi:10.1080/01446198400000015
- [10]. Abdelmegid, M. A., Shawki, K. M., & Abdel-Khalek, H. (2015). GA optimization model for solving tower crane location problem in construction sites. *Alexandria Engineering Journal*, 54(3), 519-526. doi: 10.1016/j.aej.2015.05.011
- [11]. Wang, J., Zhang, X., Shou, W., Wang, X., Xu, B., Kim, M. J., & Wu, P. (2015). A BIM-based approach for automated tower crane layout planning. *Automation in Construction*, 59, 168-178. doi: 10.1016/j.autcon.2015.05.006
- [12]. Tam, C. M., & Tong, T. K. L. (2003). GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction. *Construction Management and Economics*, 21(3), 257-266. doi:10.1080/0144619032000049665
- [13]. Lien, L. C., & Cheng, M. Y. (2014). Particle bee algorithm for tower crane layout with material quantity supply and demand optimization. *Automation in Construction*, 45, 25-32. doi: 10.1016/j.autcon.2014.05.002
- [14]. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision & Pattern Recognition*. IEEE. doi: 10.48550/arXiv.1611.07004
- [15]. Huang, W., & Zheng, H. (2018). Architectural drawings recognition and generation through machine learning. *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*. doi: 10.52842/conf.acadia.2018.156
- [16]. Zhao, B., Yin, W., Meng, L., & Sigal, L. (2020). Layout2image: Image Generation from Layout. *International Journal of Computer Vision*, 128 (10-11), 2418-2435. doi:10.1007/s11263-020-01300-7
- [17]. Liao, W., Lu, X., Huang, Y., Zheng, Z., & Lin, Y. (2021). Automated structural design of shear wall residential buildings using generative adversarial networks. *Automation in Construction*, 132. doi: 10.1016/j.autcon.2021.103931
- [18]. Yang, B., Liu, B., Zhu, D., Zhang, B., Wang, Z., & Lei, K. (2020). Semiautomatic structural BIM-model generation methodology using CAD construction drawings. *Journal of Computing in Civil Engineering*, 34(3), 04020006. doi:10.1061/(ASCE)CP.1943-5487.0000885
- [19]. Ho, C. O., Nie, T., Su, L., Yang, Z., Schwegler, B., & Calvez, P. (2021). Graph-based algorithmic design and decision-making framework for district heating and cooling plant positioning and network planning. *Advanced Engineering Informatics*, 50, 101420-. doi: 10.1016/j.aei.2021.101420
- [20]. Kim, M., Ham, Y., Koo, C., & Kim, T. W. (2023). Simulating travel paths of construction site workers via deep reinforcement learning considering their spatial cognition and wayfinding behavior. *Automation in Construction*, 147, 104715. doi: 10.1016/j.autcon.2022.104715
- [21]. Mawdesley, M. J., Al-Jibouri, S. H., & Yang, H. (2002). Genetic algorithms for construction site layout in project planning. *Journal of Construction Engineering and Management*, 128(5), 418-426. doi: 10.1061/(ASCE)0733-9364(2002)128:5(418)
- [22]. Yahya, M., & Saka, M. P. (2014). Construction site layout planning using multi-objective artificial bee colony algorithm with levy flights. *Automation in Construction*, 38(3), 14-29. doi: 10.1016/j.autcon.2013.11.001
- [23]. Younes, A., & Marzouk, M. (2018). Tower cranes layout planning using agent-based simulation considering activity conflicts. *Automation in Construction*, 93, 348-360. doi:

10.1016/j.autcon.2018.05.030

[24]. Riga, K., Jahr, K., Thielen, C., & Borrmann, A. (2020). Mixed integer programming for dynamic tower crane and storage area optimization on construction sites. *Automation in Construction*, 120. doi: 10.1016/j.autcon.2020.103259

[25]. Huang, C., Wong, C. K., & Tam, C. M. (2011). Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming. *Automation in Construction*, 20(5), 571-580. doi: 10.1016/j.autcon.2010.11.023

[26]. Huang, C., & Wong, C. K. (2018). Optimization of crane setup location and servicing schedule for urgent material requests with non-homogeneous and non-fixed material supply. *Automation in Construction*, 89, 183-198. doi: 10.1016/j.autcon.2018.01.015

[27]. Huang, C., & Wong, C. K. (2015). Optimisation of site layout planning for multiple construction stages with safety considerations and requirements. *Automation in Construction*, 53, 58-68. doi: 10.1016/j.autcon.2015.03.005

[28]. Tam, C. M., Tong, T. K. L., & Chan, W. K. W. (2001). Genetic algorithm for optimizing supply locations around tower crane. *Journal of Construction Engineering and Management*, 127(4), 315-320. doi:10.1061/(ASCE)0733-9364(2001)127:4(315)

[29]. Kaveh, A., & Vazirinia, Y. (2018). Optimization of tower crane location and material quantity between supply and demand points: A comparative study. *Periodica Polytechnica Civil Engineering*, 62(3). doi:10.3311/PPci.11816

[30]. Kaveh, A., & Vazirinia, Y. (2020). An upgraded sine cosine algorithm for tower crane selection and layout problem. *Periodica Polytechnica Civil Engineering*, 64(2), 325-343. doi:10.3311/PPci.15363

[31]. Khodabandelu, A., Park, J. W., & Arteaga, C. (2020). Crane operation planning in overlapping areas through dynamic supply selection. *Automation in Construction*, 117, 103253. doi: 10.1016/j.autcon.2020.103253

[32]. Zhang, Z., & Pan, W. (2021). Virtual reality supported interactive tower crane layout planning for high-rise modular integrated construction. *Automation in Construction*, 130, 103854. doi: 10.1016/j.autcon.2021.103854

[33]. Cho, S. H., & S. U. Han. (2022). Reinforcement learning-based simulation and automation for tower crane 3D lift planning. *Automation in Construction*, 144 (October): 104620. doi: 10.1016/j.autcon.2022.104620

[34]. Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., & Wood, K. L. (2011). Computer-based design synthesis research: An overview. *Journal of Computing and Information Science in Engineering*, 11(2). doi:10.1115/1.3593409

[35]. Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. doi: 10.48550/arXiv.1809.11096

[36]. Zablotskaia, P., Siarohin, A., Zhao, B., & Sigal, L. (2019). Dwnet: Dense warp-based network for pose-guided human video generation. doi:10.48550/arXiv.1910.09139

[37]. Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8789-8797). doi: 10.48550/arXiv.1711.09020

[38]. Ahmed, S., Weber, M., Liwicki, M., Langenhan, C., Dengel, A., & Petzold, F. (2014). Automatic analysis and sketch-based retrieval of architectural floor plans. *Pattern Recognition*

- Letters, 35(1), 91-100. doi: 10.1016/j.patrec.2013.04.005
- [39]. Ikeno, K., Fukuda, T., & Yabuki, N. (2021). An enhanced 3d model and generative adversarial network for automated generation of horizontal building mask images and cloudless aerial photographs. *Advanced Engineering Informatics*, 50. doi: 10.1016/j.aei.2021.101380
- [40]. Xu, K., Kong, X., Wang, Q., Yang, S., Huang, N., & Wang, J. (2022). A bearing fault diagnosis method without fault data in new working condition combined dynamic model with deep learning. *Advanced Engineering Informatics*, 54, 101795. doi: 10.1016/j.aei.2022.101795
- [41]. Ni, F., He, Z., Jiang, S., Wang, W., & Zhang, J. (2022). A Generative adversarial learning strategy for enhanced lightweight crack delineation networks. *Advanced Engineering Informatics*, 52, 101575. doi: 10.1016/j.aei.2022.101575
- [42]. Kong, T., Fang, W., Love, P. E., Luo, H., Xu, S., & Li, H. (2021). Computer vision and long short-term memory: Learning to predict unsafe behaviour in construction. *Advanced Engineering Informatics*, 50, 101400. doi: 10.1016/j.aei.2021.101400
- [43]. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. doi: 10.48550/arXiv.1411.1784
- [44]. Miyato, T., & Koyama, M. (2018). cGANs with projection discriminator. doi: 10.48550/arXiv:1802.05637
- [45]. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232). doi: 10.48550/arXiv.1703.10593
- [46]. Kim, T., Cha, M., Kim, H., Lee, J. K., & Kim, J. (2017, July). Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning* (pp. 1857-1865). PMLR. doi: 10.48550/arXiv.1703.05192
- [47]. Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2849-2857). doi: 10.48550/arXiv.1704.02510
- [48]. Jia, S., Liu, C., Guan, X., Wu, H., Zeng, D., & Guo, J. (2022). Bidirectional interaction between BIM and construction processes using a multisource geospatial data enabled point cloud model. *Automation in Construction*, 134, 104096. doi: 10.1016/j.autcon.2021.104096
- [49]. Li, K., Luo, H., & Skibniewski, M. J. (2019). A non-centralized adaptive method for dynamic planning of construction components storage areas. *Advanced Engineering Informatics*, 39, 80-94. doi: 10.1016/j.aei.2018.12.001
- [50]. Hawarneh, A. A., Bendak, S., & Ghanim, F. (2020). Construction site layout planning problem: past, present and future. *Expert Systems with Applications*. doi: 10.1016/j.eswa.2020.114247
- [51]. Zhang, Z., Pan, W., & Pan, M. (2021). Critical considerations on tower crane layout planning for high-rise modular integrated construction. *Engineering, construction, and architectural management*, 2022, Vol.29(7), p.2615-2634. doi: 10.1108/ECAM-03-2021-0192
- [52]. Farajmandi, M., Ali, M., Hermann, R. and AbouRizk, S. (2020). "A decision support tool for planning module installation in industrial construction", *Engineering, Construction and Architectural Management*, Vol. 27 No. 9, pp. 2615-2641. doi: 10.1108/ECAM-01-2019-0069
- [53]. Ma, K., Mao, Z., He, D., & Zhang, Y. (2020). Design a network architectural teaching system by auto CAD. *Computer-Aided Design and Applications*, 17(S2), 1-10. doi: 10.14733/cadaps.2020.S2.1-10
- [54]. Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution

- image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 8798-8807. doi: 10.48550/arXiv.1711.11585
- [55].Rahbar, M., Mahdavinejad, M., Markazi, A. H. D., & Bemanian, M. (2022). Architectural layout design through deep learning and agent-based modeling: A hybrid approach. *Journal of Building Engineering*, 47. doi: 10.1016/j.jobbe.2021.103822
- [56].Zhao, C., Yang, J., Xiong, W., & Li, J. (2021). Two Generative Design Methods of Hospital Operating Department Layouts Based on Healthcare Systematic Layout Planning and Generative Adversarial Network. *Journal of Shanghai Jiaotong University (Science)*, 26(1), 103-115. doi: 10.1007/s12204-021-2265-9
- [57].Wang, Z., & Liu, J. C. (2021). Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training. *International Journal on Document Analysis and Recognition*, 24(1-2), 63-75. doi: 10.1007/s10032-020-00360-2
- [58].Zhou, W., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*, 13(4). doi: 10.1109/TIP.2003.819861
- [59].Li, C., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*, 14(1), 21-3. doi: 10.5334/dsj-2015-002
- [60].Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). doi: 10.1186/s40537-019-0197-0
- [61].Zhang, P., Harris, F. C., Olomolaiye, P. O., & Holt, G. D. (1999). Location optimization for a group of tower cranes. *Journal of Construction Engineering and Management*, 125(2), 115-122. doi: 10.1061/(ASCE)0733-9364(1999)125:2(115)
- [62].Lu, Y., Gao, M., Liang, T., He, Z., Feng, F., & Pan, F. (2022). Wind-induced vibration assessment of tower cranes attached to high-rise buildings under construction. *Automation in Construction*, 135. doi: 10.1016/j.autcon.2022.104132
- [63].Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review. *Remote Sensing*, 13(13). doi: 10.3390/rs13132450
- [64].Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1), 98-136. doi: 10.1007/s11263-014-0733-5
- [65].Hosseini, M., Beiranvand, P., Dadgar, M.R. and Olfati, A. (2017). A mathematical model for optimal tower crane layout planning, *Decision Science Letters*, Vol. 6 No. 4, pp. 377-386. doi: 10.5267/j.dsl.2017.2.001
- [66].GB 50016-2014 Code for fire protection design of buildings (2018 edition), China Planning Press, Beijing, in Chinese.
- [67].Lu, Y., & Zhu, Y. (2021). Integrating hoisting efficiency into construction site layout plan model for prefabricated construction. *Journal of construction engineering and management* (10), 147. doi: 10.1061/(ASCE)CO.1943-7862.0002158