IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 5, 2019, accepted December 18, 2019, date of publication December 23, 2019,
date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2961368

Webvr Human-Centered Indoor Layout Design Framework Using a Convolutional Neural Network and Deep Q-Learning

WEN ZHOU[1,2], (Member, IEEE), WENYING JIANG[1], WEIXIN BIAN[1], AND BIAO JIE[1]
[1]School of Computer and Information, Anhui Normal University, Anhui 241002, China
[2]Anhui Provincial Key Laboratory of Network and Information Security, Anhui 241002, China

Corresponding author: Wen Zhou (w.zhou@ahnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61902003 and Grant 61976006, and in part by the Doctoral Scientific Research Foundation of Anhui Normal University.

ABSTRACT With the rapid development of Web Virtual Reality (WebVR) technology, increasing focus has been placed on this domain. WebVR indoor scenario design studies have been of important value in both academia and industry. However, many bottlenecks still need to be overcome, such as the weak computing capacity and limited memory space available in web browsers. In particular, there are many deficiencies in virtual scenes, i.e., lack of fidelity, low automation and poor scene interactability. In this paper, we propose a novel WebVR indoor furniture layout design framework to enhance the capabilities of automatic furniture layout and significantly enhance the interactiveness of virtual scenarios. In particular, we present a hand-drawn sketch recognition scheme based on the ResNet convolutional neural network (CNN), which can strongly improve scenario interactiveness by allowing the user to conveniently add new furniture by means of free-hand drawing operations rather than tedious manual drag-and-pull operations. In addition, based on a deep Q-learning network (DQN), the best positions (states) for these pieces of furniture (agents) in virtual scenarios can be automatically determined, making it easy to satisfy popular design principles. Finally, we report experiments conducted to validate the feasibility of our proposed framework, and the results fully demonstrate that this framework is completely feasible.

INDEX TERMS WebVR, indoor scenario design, ResNet, hand-drawn, deep Q-learning network.

I. INTRODUCTION

With the rapid development of web technology, the combination of virtual reality (VR) and web technology, i.e., Web Virtual Reality (WebVR) technology, has attracted increasing attention. The potential applications of WebVR are widely varied, including applications in medical surgery, education, indoor and outdoor scenario design, and so on. Because WebVR has the advantages of being highly convenient and plugin-free, it has become one of most promising fields of web technology. In practice, most of the advances in virtual effects achieved so far in traditional VR technology have mainly relied on third-part devices, such as the Leap Motion and Kinect.

Moreover, efforts to achieve a human-centered focus in WebVR-based indoor layout design have been gaining in popularity, especially for bedroom scenarios. Correspondingly, free-style indoor design for virtual 3D space has become increasingly important. However, there are still many problems that need to be solved; e.g., because of the limited capabilities of current interactive schemes for effectively specifying and arranging 3D furniture, such 3D furniture manipulations require excessive labor-intensive dragging operations, thus strongly decreasing the interest of users who might otherwise be strongly attracted to participate in indoor layout design. Hence, there is a need to effectively eliminate many of the dull and tedious tasks currently required in layout design, for example, by enabling the automatic generation of indoor layout results in accordance with popular layout principles. In this way, users' interest can be more effectively captured to encourage them to positively engage in indoor layout design.

Nevertheless, there are still dozens of bottlenecks related to indoor scenario layout design that need to be overcome,

The associate editor coordinating the review of this manuscript and approving it for publication was Yongtao Hao.

such as the latency problem. In a WebVR application, excessively long latency times can lead to significant usability challenges, potentially causing users to lose patience and shut down the webpage. Therefore, a rapid response time is important for WebVR applications. For indoor layout design, although good results can be obtained based on some of the complex networks that are currently available (such as GoogLeNet), these networks are not well adapted to handle such design-related tasks because their response times are too slow, and thus, their high latency may destroy their value for such purposes.

As is well known, layout design is a very subjective topic, and there are few existing datasets or standard indicators for studying and assessing related methods. In fact, the majority of the existing state-of-the-art methods are focused primarily on 3D indoor scene understanding, while the needs of human-centered design have typically been ignored or disregarded. However, there are many aspects of the problem of layout design in 3D space that require attention, including the questions of how to achieve a suitable 3D rendering effect and furniture arrangement results that are consistent with popular stylistic sensibilities.

On the other hand, the pursuit of high fidelity in VR has increased the feasible scale of VR scenarios. As a result, the demands to overcome various visualization problems in WebVR have become increasingly urgent. Moreover, the analysis of the actions of virtual agents requires high computing power; by contrast, the computing capacity of web browsers is very limited. Thus, the question of how to effectively capture the characteristics of objects in a real scene is a key problem in WebVR.

In this paper, we focus on designing a better method for solving the virtual indoor furniture layout problem to effectively support human-centered indoor scenario design, i.e., we solve the virtual indoor furniture layout (VIFL) problem for designing a reasonable and personalized furniture layout in a WebVR indoor scene. This is a spatial optimization problem that is inextricably linked with both ideal scenario learning and the visualization of three-dimensional data.

Moreover, the problem of suitable human-computer interaction (HCI) is difficult to solve for such a massive application. For an online WebVR application in particular, there is a strong need to develop a means through which a good HCI effect can be conveniently realized. In practice, the users of a VIFL system will often need to add new furniture models into virtual scenes. In traditional methods, however, the ability to arrange the furniture in virtual indoor scenes such as bedrooms relies on manual drag-and-drop manipulations. Consequently, furniture rearrangement is a very labor-intensive task, and when there are a large number of furniture models available, it can be extremely time consuming for users to manually find the target furniture. Moreover, sometimes, users may not know exactly which furniture they wish to use. In such a case, it may be possible to provide a better user experience if the user is able to hand draw a sketch to describe a general idea of the furniture that is desired. However, the correct recognition of such hand-drawn abstract and amateur sketches is quite difficult to achieve. Based on the above discussion, the contributions of the present paper can be summarized as follows:

(1) We propose a method based on a convolutional neural network (CNN) for recognizing hand-drawn furniture sketches. In this way, a more intelligent HCI process can be achieved.
(2) We present an approach for determining the most appropriate positions of furniture agents in a virtual indoor scene. In essence, the objective of the VIFL problem is to find the most appropriate position (i.e., state) for each furniture agent in a virtual scenario. More concretely, the deep Q-learning method is used to optimize the positions of the furniture agents, i.e., state/action strategy selection is applied to obtain the best positions. In this paper, we assume that four different actions (forward, back, left, and right) can be employed to obtain new Q-values to identify the best position for each furniture agent. In essence, the problem is treated as a complex strategy selection problem based on environmental feedback.

This paper is organized as follows. In Section **II**, we present previous related work on hand-drawn sketch recognition and WebVR layout design. In Section **III**, we introduce our innovative approach. In Section **IV**, we explain the details of our proposed method. The results of an experimental evaluation are presented in Section **V**. Finally, Section **VI** concludes this work and presents an outlook on future studies.

II. RELATED WORKS

A. HAND-DRAWN SKETCH RECOGNITION

In the computer vision domain, the hand-drawn sketch recognition problem is always a hot topic, and numerous successful related studies have been reported in the literature. However, in these works, the research objects are generally common hand-drawn sketches; therefore, the existing sketch datasets are mainly based on general-purpose sketches, and no specialized furniture sketch dataset is available. However, the ability to correctly solve a given recognition problem heavily depends on the learning method, which must be trained on a large number of suitable samples. Although the available samples of common sketches are reasonably sufficient, there exist very few furniture sketch samples or datasets that are suitable for training a recognition tool for the VIFL problem addressed in this work. In the following, we review representative existing state-of-the-art approaches for sketch recognition.

In the context of professional computer-aided design (CAD) or artistic drawing, Lu *et al.* [1], Jabal *et al.* [2] Zitnick *et al.* [3], Bae *et al.* [30], Kara *et al.* [31], and Milosevic *et al.* [32], Zhou *et al.* [33], [34] conducted extensive research on sketch recognition. Their proposed methods yield excellent results, but they require high-quality sketches, which are very difficult for typical amateur users

to produce. Eitz *et al.* [4] proposed a new framework for recognizing common sketches. In addition, these authors released an open-source sketch dataset to help others train their own tools, and this dataset also contains some furniture sketches. Later, Schneider *et al.* [5] adopted a support vector machine (SVM) classifier and considered different hand-drawn features extracted from images to be used for representation. Li *et al.* [6] demonstrated that fusing various local features using a multikernel learning approach helps to improve the performance of sketch recognition. They also examined the performance achieved with many different features and found that the Histogram of Oriented Gradients (HOG) descriptor is generally the best feature descriptor for sketches.

Notably, CNNs have recently demonstrated impressive performance for many recognition tasks across many different disciplines; CNNs have dominated a number of visual recognition challenges, producing the top benchmark results. One important advantage of a CNN compared with other conventional classifiers, such as SVM and Bayesian decision tree classifiers, is its close coupling of representation learning and classification, which endows the learned feature representations with extremely high discriminatory capabilities.

Simonyan *et al.* [7] proposed a deeper network with smaller filters that is preferable for image recognition. Despite these advances, most existing image recognition methods based on CNNs are optimized for images and not for sketches; consequently, their performance for sketches is suboptimal. Yu *et al.* [8] proposed a new CNN-based method that can demonstrate even better recognition performance than that of humans, thus achieving an enormous breakthrough for sketch recognition. However, this method is more suitable for distinguishing sketches with salient between-class discrepancies, e.g., human and animal sketches. By contrast, there are few between-class differences in many common furniture sketches, such as table and bed sketches; therefore, this approach is not suitable for our requirements.

As another related method, Wang *et al.* [9] designed a novel scheme for cross-domain matching that employs a variant of the Siamese network in which the shape view image branch and the sketch branch have the same architecture, without any special treatment of the unique features of sketches. However, this method is a solution for sketch-based shape retrieval rather than sketch recognition.

Moreover, deeper neural networks are often more difficult to train. To address this problem, He *et al.* [10] proposed a novel residual neural network called ResNet to facilitate the training of networks that are substantially deeper than those used previously. In this paper, we adopt this promising network architecture for furniture sketch recognition based on a limited number of furniture sketch samples.

B. INDOOR LAYOUT DESIGN

Scene design is a popular topic of research in many fields, such as city planning, 3D scene modeling, and architectural design. Many important related research results have been reported, for instance, those of Parish *et al.* [11], Muller *et al.* [12], Chen *et al.* [13], and Merrell *et al.* [14]. As an important aspect of scene design, the VIFL problem has received considerable attention in recent years. Using traditional manual methods, furniture layout is often a very time-consuming and labor-intensive task. In particular, each 3D furniture model requires tedious rotation in six degrees of freedom and translation in large spatial dimensions; consequently, relying solely on a mouse for such manual manipulations of 3D objects is very inconvenient. Moreover, the majority of people lack indoor design expertise and experience and consequently must likely expend a great deal of effort to achieve a reasonable furniture layout. Even for professional designers, continuously repeating similar operations while arranging furniture is a dull and tedious task. Therefore, the ability to generate popular VIFL results for amateur users would be an attractive option. Moreover, most professional users would also find it convenient to be able to perform necessary revision based on certain popular VIFL principles to generate personalized design results at a lower cost. To this end, it would be desirable to develop an automatic furniture layout tool that can generate popular and reasonable design results.

In essence, automatic furniture layout is a problem of indoor scene optimization. A layout evaluation function has been proposed by Yu *et al.* [15] for finding suitable furniture arrangements based on the simulated annealing method. However, as the variety of different furniture models in a virtual 3D room environment increases, the number of iterations will also increase, considerably reducing the convergence speed and resulting in a very long execution time and high latency. Hence, this approach is clearly not suitable for a web-based application.

Meanwhile, with the rapid development of the 3D technologies available in the web environment [16], [17], massive 3D indoor design systems have emerged, which are very welcome because they allow users to freely perform indoor design tasks without needing to install any plugins. This new trend has motivated researchers to extend the existing offline methods of furniture layout to online applications. However, online furniture layout requires the ability to provide real-time response while also generating reasonable layout results.

A method of synthesizing 3D object arrangements from examples has been proposed by Fisher *et al.* [18]; this method considers dozens of plausible new scenes using a machine learning approach. Xu *et al.* [19] have presented a framework that can automatically generated a vivid 3D scene from a hand-drawn sketch. Moreover, Xu *et al.* [20] have proposed a system for automatically creating 3D indoor scenes based on room shapes entered by users. In this approach, 3D object models are arranged based on the relationships between the models and the room. Song *et al.* [21] have also proposed automatic furniture layout algorithms to help users rapidly generate reasonable layout results. Example-based reasoning and a distant field method are used to measure

the relationships among every piece of furniture to obtain better positions for the furniture in a virtual room. Additionally, Li *et al.* [22] have proposed the novel recursive neural network and variational autoencoder (RvNN-VAE) method for generating plausible 3D indoor scenes based on input consisting of 2D sketches or images. A learning approach is utilized to generate the closest feasible layout design. In fact, many similar methods of scenario synthesis have also been proposed by, e.g., Wang *et al.* [23], Song et al. [24], Xu *et al.* [25], Yu *et al.* [26], and Wu *et al.* [27]. Although these approaches can generate excellent results in the relevant domain, there is an obvious disadvantage that urgently requires a solution, i.e., the lack of sufficient examples of indoor furniture arrangements for training a related learning framework. In addition, the arrangement of indoor scenes is relatively subjective; consequently, we limit our objective to providing only basic or common arrangement results rather than enabling rich individualized design capabilities since the ability to generate common designs that can meet the habitual requirements of ordinary users is the first priority.

In this paper, based on the extensive research reported in the literature summarized above, to further enhance the interactiveness of 3D indoor scenario design, we develop a tool for hand-drawn sketch recognition to allow new furniture to be conveniently added into a scene. In addition, a deep Q-learning action strategy scheme is employed to determine the optimal states for the various furniture agents. In particular, the motivation of our work is to make WebVR applications more suitable for indoor layout design tasks to effectively achieve human-centered design results. Moreover, we abandon the case-based inference method in favor of the rule-based reinforcement learning approach. In this way, we can effectively decrease the dependence on specific indoor layout cases.

III. OVERVIEW OF THE PROPOSED FRAMEWORK

In this section, an overview of the proposed framework is presented. The framework addresses two main problems: sketch recognition and indoor furniture layout design.

For sketch recognition, the framework comprises two stages, i.e., an online stage and an offline stage. In the offline stage, the complete pipeline must be trained to correctly predict a label for each input sketch. In this paper, the ResNet50 CNN architecture is used to classify and describe furniture sketches, which often show large within-class differences and small between-class differences. In the online stage, several tasks need to be completed. Sketch preprocessing ensures that each sketch is of a suitable size to be utilized in subsequent analysis. In addition, in practice, when a hand-drawn sketch is input, the requested furniture should be provided immediately, using the prediction scheme based on the trained model, to allow furniture to be conveniently added into virtual room scenes.

For furniture layout design, to correctly handle the relationships between the room and the various pieces of furniture, we divide furniture into two types, i.e., coupled-type furniture and free-type furniture. For example, for a bedroom, a sofa is free-type furniture because its position depends only on the room itself. By contrast, the positioning of a nightstand is strongly related to the placement of the bed; in some sense, a nightstand can be viewed as a bed accessory. Therefore, a nightstand is considered to be coupled-type furniture. Notably, the classification of different furniture types depends on the specific design goals. A great deal of time can be saved by defining coupled-type furniture; by contrast, if all furniture is treated as free-type furniture, it will take a very long time to complete the layout design for a whole bedroom. In essence, which furniture should be of the coupled type strongly depends on the indoor design principles applied. For coupled-type furniture, there exists an obvious master-slave relationship; for instance, consider the relationships between a TV and and its TV stand and between a dresser and its stool. Specifically, the location of accessory furniture is typically restricted by that of the master furniture. Therefore, we need only to place the master furniture; once this is done, the position of the slave furniture is determined. By contrast, for free-type furniture, we need to consider only the relationship between each piece of furniture and the room itself. The distance field concept [21] has previously been adopted to measure the relationships between furniture and the room, i.e., the Q-values of different states. Based on this approach, we can arrange every piece of furniture in a room scenario to specify suitable furniture locations to complete the layout task.

An overview of the proposed framework is shown in Figure 1.

In general, the proposed framework can be described as follows:

1. Online pipeline: This stage can be divided into two steps. The first step is online input sketch drawing recognition. As is well known, for indoor scenarios, there are dozens of different possible furniture models that may appear in a given room; therefore, input sketch recognition should be an iterative procedure. As shown in the top left part of Figure 1, the system performs three different operations. First, when a furniture sketch is input, it is subjected to preprocessing and recognition (based on the training results from the offline pipeline). The relevant furniture models are then presented, and the user can select which piece of furniture should be added to the indoor scene. When more furniture needs to be added to the scene, additional inquiries can be made to continue to add new pieces of furniture. Clearly, this is an iterative process. Moreover, as shown in the top right part of Figure 1, the relationships between the furniture and the room are defined to determine suitable positions based on the principles of indoor layout design. For two pieces of furniture with a free positioning relationship, e.g., a chair and a sofa or a stand and a table, the deep Q-learning scheme must be used to determine reasonable positions for these pieces of furniture in the indoor scene. In turn, the positions

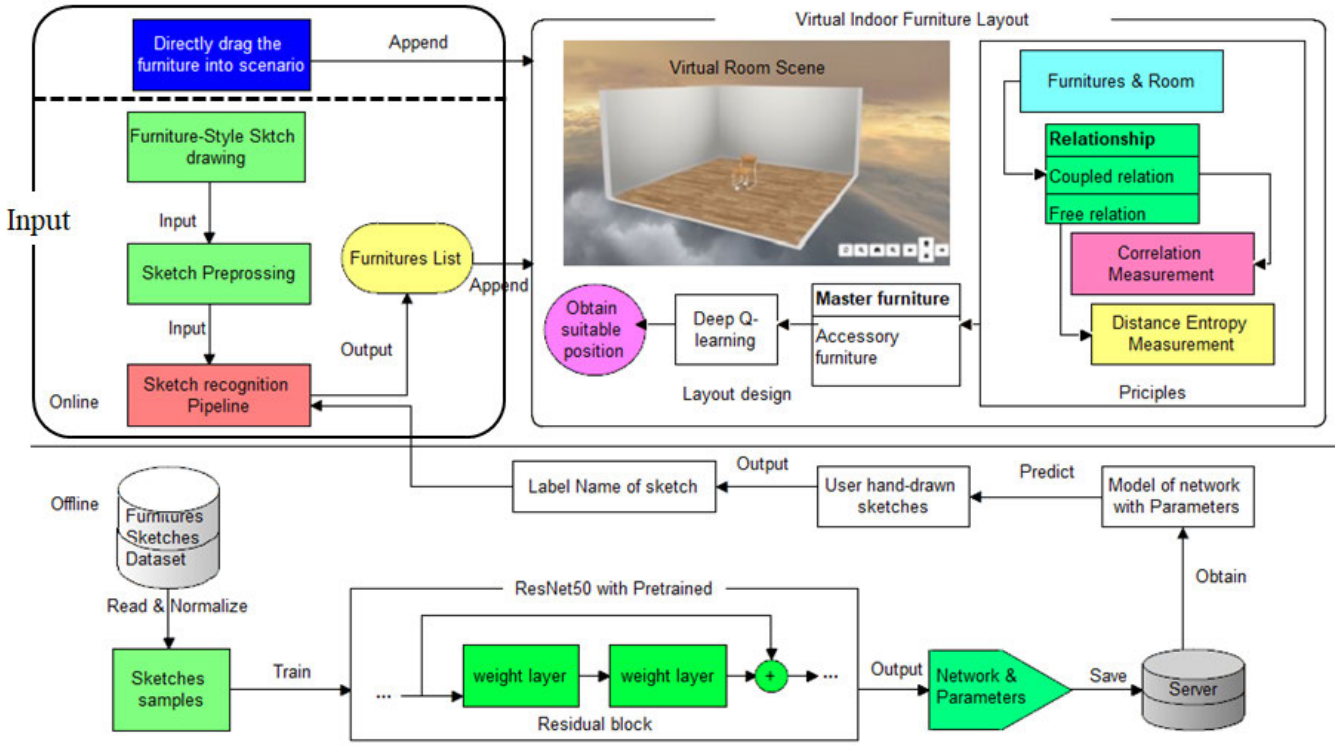

FIGURE 1. An overview of the proposed framework. It consists of two stages, i.e., an online stage and an offline stage. The online stage involves the input of sketch drawings by the user, the output of recognition results and the display of the final automatically generated layout results. In the offline stage, the training pipeline is executed to provide the online tool with the ability to produce suitable prediction results. The offline stage is mainly run on the server side, whereas the online tasks are mainly executed in a web browser.

of pieces of furniture with master-slave relationships, such as a chair and a table or a bed and a nightstand, can then be easily rearranged. Thus, layout design is performed based on common layout principles.

2. Offline pipeline: The purpose of this stage is to train an existing network to obtain a suitable model for predicting the correct labels for furniture sketches. In this study, the pretrained ResNet50 architecture has been utilized as the base network model.

IV. DESCRIPTION OF THE PROPOSED FRAMEWORK

In the previous section, an overview of the proposed framework has been presented. Next, the details of the proposed framework will be specified.

A. RECOGNITION OF FURNITURE SKETCHES

To train the model for the recognition task, we must conduct a suitable supervised learning process. This process consists of three main steps, i.e., preprocessing, training, and testing, as shown in Figure 2. These steps are described as follows:

(1). First, we need to collect furniture sketches to form a training dataset. In this paper, we select ten different categories of furniture from the Eitz sketch dataset [4], which is one of most famous sketch datasets in academia.In every category associated with furniture (e.g., armchair, bed, bench, chair, door, and table), there are 80 different sketches. Of course, however, the categories of furniture that may appear in a bedroom extend far beyond those listed above. For categories for which no sketch samples are available, users must still rely only on direct dragging rather than sketch-based recognition. For large sketches, we scale them down in size, e.g., $1111 \times 1111 \rightarrow 128 \times 128$. To prevent a scaled-down sketch from becoming unclear or blurry, an adaptive thinning algorithm [28] is used to generate a ten-pixel-wide skeleton of the sketch to augment the strokes of the original sketch; in this way, we can effectively obtain high-resolution scaled sketches.

(2). Rapid response is very important for web-based applications. In this paper, we adopt the well-known, powerful ResNet50 network, which has achieved great success in many fields, including image recognition, as the base network for training. In comparison to other networks with similarly good performance, ResNet-based models require less time for training and prediction. In particular, we utilize a well-known ResNet model that has been pretrained on the ImageNet dataset, which greatly accelerates the whole training process.

(3) The network and parameters are saved to disk to allow the whole network to be conveniently read out during the testing stage. Finally, prediction can be performed using the trained model to obtain the correct label

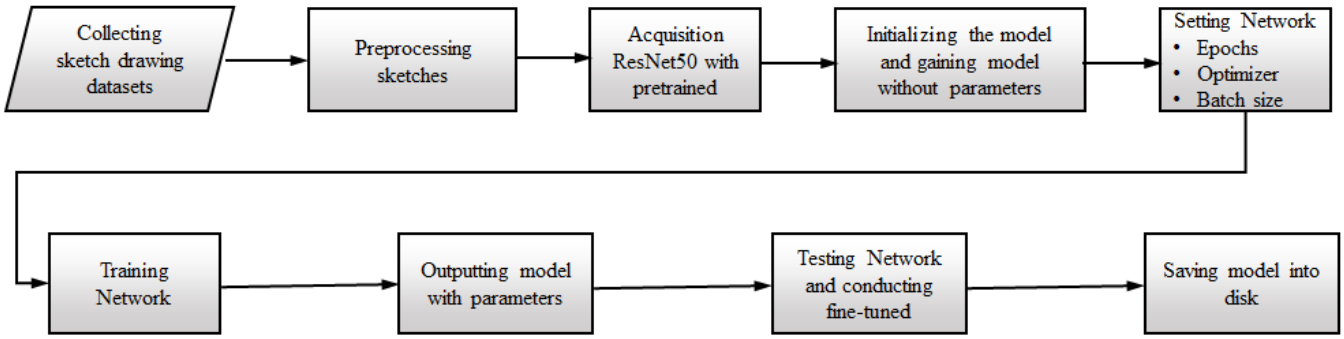

FIGURE 2. The main flow of the model training process for recognizing hand-drawn sketches. The ResNet50 architecture used in this study was first pretrained on the ImageNet dataset before being fine-tuned on sketch data.

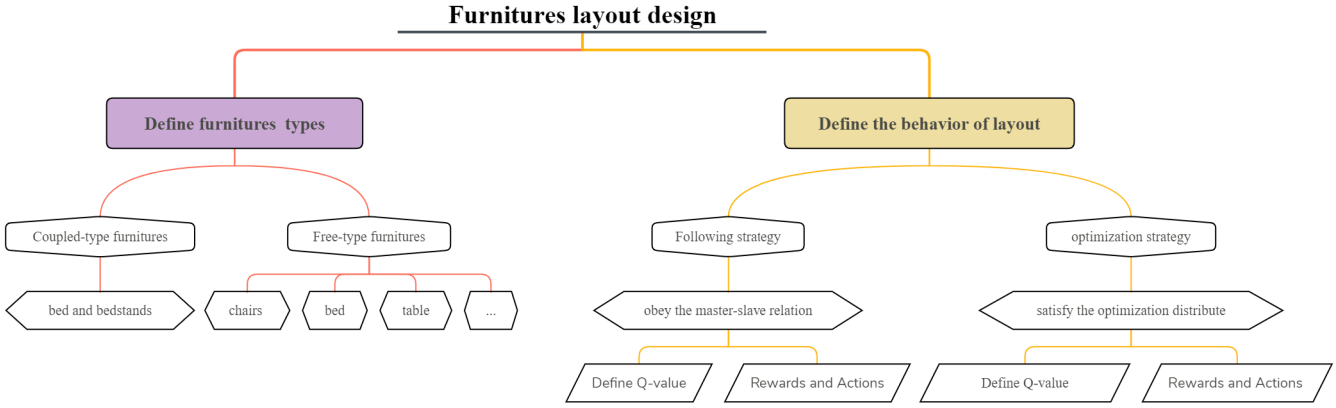

FIGURE 3. The layout principles defined for indoor scenarios.

TABLE 1. The training parameters for the ResNet50 model.

Parameter	Specification
Input Size	128×128
Number of Categories	7
Number of Samples	560
Batch Size	20
Number of Epochs	500
Optimizer	Adadelta
Loss Metric	Cross Entropy

for each input sketch. In this way, we can effectively complete the recognition task.

The parameters used to further train the pretrained ResNet50 network are shown in Table 1.

B. FOUNDATIONS OF FURNITURE LAYOUT DESIGN

In essence, the procedure for layout design is to conduct a corresponding action strategy selection process for every piece of furniture in a virtual room. We consider four different possible actions: forward, back, left and right. The architecture of the proposed approach is shown in Figure 3.

1) DEFINING THE Q-VALUE OF A STATE

In reinforcement learning, Q-learning is an important off-policy method that is widely used in computer games, intelligent control, robotics and other fields. Q-learning enables an agent to learn the mapping relationship from states to actions by trial and error through continuous interaction with the environment so as to maximize its long-term cumulative reward. The most significant difference between Q-learning and other machine learning methods, such as CNNs, is that it requires no pretraining; it relies only on the information gained by interacting with the complex environment. Additionally, Q-learning can consider teacher signals issued in various states; consequently, it has wide application prospects for solving various complex decision optimization problems. Fortunately, the layout design problem is also essentially a classical multitarget optimization problem in 3D space. This problem requires consideration of the relationships between the room and individual pieces of furniture and also involves many complex relationships among multiple pieces of furniture.

Consider a piece of furniture (denoted by Δ) in a virtual room (denoted by X). As shown in the blue rectangle in Figure 4, this piece of furniture can be moved forward, backward, left or right in the process of layout design. For interaction with the environment, we define a function to measure the Q-value of every point $\delta = (x, y, z) \in \Delta$, which denotes the spatial coordinate position of Δ in the room. It is always true that $y = 0$ because the piece of furniture must

TABLE 2. Q-table of state/action tuples.

State	Action	Reward
P_1	F	$\Phi(X_f, F)$
P_2	B	$\Phi(X_f, B)$
P_3	L	$\Phi(X_f, L)$
...	...	...
P_{size}	R	$\Phi(X_f, R)$

be sitting on the floor (we do not consider objects on the walls, such as picture frames). Then, $\forall\delta \in \Delta$, the distance field E [21] can be represented as follows:

$$D(P_i, P_j) = \parallel P_i - P_j \parallel_2 \tag{1}$$

$$E(\delta) = \sum_{n=0}^{N} \alpha \times D(\delta, \delta^n_{door}) + \beta \times \sum_{m=0}^{M} D(\delta, \delta^m_{window}) + \gamma \times D(\delta, \delta_{center}) \tag{2}$$

Here, P_i and P_j denote two different points in 3D space. The parameters α, β, and γ are experimental values representing the weights for doors, windows, and the center point of the room, respectively; in this paper, $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 1$. δ^n_{door} denotes the center point of the n^{th} door (the center point of the black part of the door in Figure 4); similarly, δ^m_{window} and δ_{center} denote the center points of the m^{th} window and the entire room, respectively. N and M denotes the total numbers of doors and windows, respectively. If $N = 0$, then there are no doors in the room; in this case, $\alpha = 0$.

To compute the zone of E, the possible actions must be considered. In this paper, we assume that there are four different possible actions $A = F, B, L, R$ for every piece of furniture, i.e., forward (F), backward (B), left (L), and right (R). In addition, the bounding rectangle of a piece of furniture consists of four line segments (as shown in Figure 4, they are denoted by X_{red}, X_{purple}, X_{black}, and X_{blue}). Therefore, the reward can be expressed as follows.

$$\Phi(X_f, a) = \begin{cases} \epsilon \times \sum_{i=0}^{w} E(\delta^i_{purple}) - E(\delta^i_{red}) & a = F \mid B \\ \epsilon \times \sum_{i=0}^{h} E(\delta^i_{black}) - E(\delta^i_{blue}) & a = R \mid L \end{cases} \tag{3}$$

Here, w and h denote the width and height, respectively, of furniture object X_f. When $a = F$ or R, $\epsilon = 1$; otherwise, $\epsilon = -1$. In addition, δ^i_{purple} and δ^i_{red} denote the i^{th} points on the purple line and red line, respectively, of the bounding rectangle (as shown in Figure 4). Finally, we can generate a Q-table that records the Q-value of every state, as shown in Table 2.

In Table 2, P_i denotes the i^{th} possible position of a piece of furniture in the virtual room, which strongly depends on the size of the room. For example, in a 100×100 room, there are theoretically 10000 possible states for a piece of furniture. However, as is well known, there are some positions where it is not possible or reasonable to place furniture, such as within the zones of the doors and windows in an indoor scenario. For example, no furniture should be placed within the black, gray

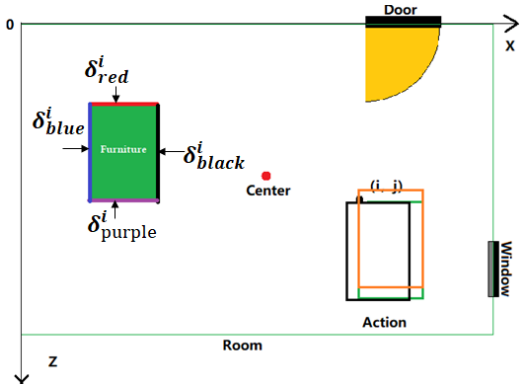

FIGURE 4. A figure illustrating the indoor layout design problem. There are three types of key regions, i.e., the regions of doors and windows and the center point of the room (the red point in the figure). In addition, the green rectangle represents the placement of the piece of furniture in the current state.

or yellow zones in Figure 4; therefore, we should discard or abandon these locations. Notably, the center of the room is generally also a poor location for placing furniture. Hence, $\forall\Delta \in X$, for a given action a', the Q-value can be computed as shown in equation 4.

$$Q(S', a') = \sum_{\delta \in \Delta} E(\delta) \tag{4}$$

In addition, the Q-value $Q(S')$ of a given state S' is represented as shown in equation 5.

$$Q(S') = \max_{a'}^{n} Q(S', a') \tag{5}$$

Here, n denotes the number of possible actions that can be taken from the given state S. In this paper, $n = 4$.

Figure 5 illustrates the variations in the distance field. Dark-colored zones are not good regions for placing furniture; instead, furniture generally should be placed in light-colored zones, such as yellow zones. Note that as furniture is added to a scene, salient changes occur in the whole distance entropy distribution.

2) DEEP Q-LEARNING NETWORK FOR LAYOUT DESIGN

In Figure 5, the black rectangles denote forbidden regions for the placement of any furniture; they are occupied by doors, windows or other furniture. Concretely, if $\forall\delta_i \in X_f$ there is overlap with these zones (black rectangles in Figure 4), then $Q(P_{current}) = 0$ ($P_{current}$ is the center position of the furniture and can also be understood as the location of the furniture). Next, we must obtain the maximum Q-value among all possible locations in the virtual room. Of course, when the location of the furniture is outside of the room (more specifically, $\exists\delta_i \in X_f$, $\delta_i \notin X$ or $X_f \bigcap X \neq X_f$), the Q-value should be set to 0.

In addition to the relationships between the room and the furniture, there may also be rich and complex connections among multiple pieces of furniture, for instance, a bed and two nightstands, whose positions are restricted by the position of the bed. We refer to furniture of this kind as coupled-type

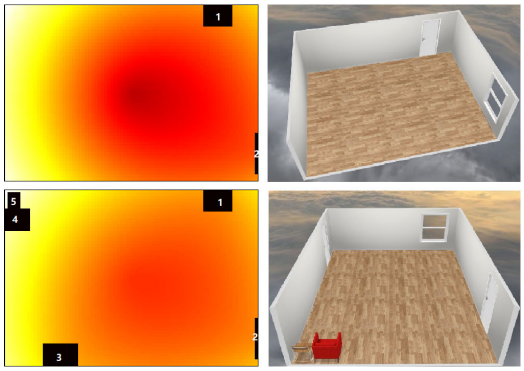

FIGURE 5. A figure comparing the distance entropy of different layouts. In the top row, the model includes only a window and a door. In the top left figure, black rectangle no. 1 represents the shape of the region occupied by the door in the room, where the energy entropy is zero; no furniture can be placed in this region. Rectangle no. 2 represents the region occupied by the window model; similarly, it is not possible to place any furniture in this region. In the bottom row, three objects are arranged in the room, i.e., a door and two chairs (see the bottom right figure). In the bottom left figure, rectangles no. 3, 4, and 5 represent the positions occupied by the three newly added models, generating zero distance entropy.

furniture. In such a case, we arrange only the master furniture, and the accessory furniture can then be automatically placed in the corresponding positions. We refer to the other kind of furniture, such as a table, as free-type furniture.

Q-learning is a model-free off-policy algorithm for estimating the long-term expected return of executing an action from a given state. These estimated returns are known as Q-values. A higher Q-value indicates that an action a is judged to yield better long-term results when starting from a state s. Q-values are learned iteratively by updating the current Q-value estimate towards the observed reward R plus the maximum Q-value over all possible actions a' resulting in a state s'. This can be easily represented as shown in equation 6.

$$Q(s, a; \theta) \leftarrow r_t + \xi \max_{a'} Q(s', a'; \theta) \tag{6}$$

Here, the purpose of the semicolon is to separate the neural network parameters θ from the state and action. Clearly, the term θ is constantly changing; thus, the Q-value for a given state and action is also constantly changing. In addition, the term r_t is the reward for the new state at time t, which can be obtained in accordance with equation 3 (see Table 2). The factor ξ denotes a discount rate or decay rate; in this paper, we assume that this parameter does not change, i.e., $\xi = 1$.

Recently, deep CNNs have often been exploited to make full use of the ability to optimize the relationships between stochastic actions (e.g., the ϵ greedy algorithm) and predicted actions (i.e., deep Q-learning networks). In contrast to the traditional Q-learning method, a CNN optimizer can be utilized to predict concrete actions rather than stochastic strategies. Consider a loss function L that represents the discrepancies between the Q-values of stochastically generated actions and the Q-values of predictions; a CNN optimizer can be utilized to minimize this loss function L. In this way, through training

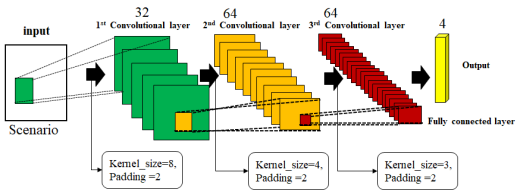

FIGURE 6. Architecture of the CNN to be trained.

the CNN, we easily obtain the ideal network parameters; hence, we can predict a concrete action based on the current state of an agent, e.g., the position of a piece of furniture in a virtual scene. L can be represented as shown in equation 7.

$$L = \| r_t + \xi \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta) \|_2 \tag{7}$$

Here,the notations θ and θ' represent two different sets of network parameters; as mentioned above, during the training stage, the parameters of the neural network are constantly changing.Otherwise, the definitions of the parameters in equation 7 are essentially the same as those in equation 6. Notably, the loss function L has two components: one is the Q-value based on the stochastic strategy (i.e., the ϵ greedy strategy), as specified in equation 6, and the other is the result predicted based on the network with parameters θ.

C. TRAINING STAGE FOR A DEEP Q-LEARNING NETWORK

In this section, we mainly present the details of how to train the deep Q-learning network (DQN). The architecture of the CNN to be trained is shown in Figure 6, and an overview of the whole training pipeline is illustrated in Figure 7.

The training process can be divided into three stages. The first is the preprocessing stage, i.e., the necessary preprocessing operations, which include discretizing the scene into a grid. In reality, the possible states of the actual scene are continuous. However, to reduce the time required to compute the optimal state for each agent (i.e., each individual piece of furniture), we introduce the idea of a spatial grid from the field of fluid simulations to generate a reliable and valid discrete model of the scene. When the majority of the volume of an agent (i.e., a piece of furniture) belongs to a certain grid cell, we consider that it should placed in this grid cell. Thus, the state space of the scene can be naturally represented by some limited number of discrete grid cells. In addition, we need to effectively adjust the size of every input image. In this paper, the size of each input image is uniformly scaled to 600×400. An overly large input image size would considerably reduce the training speed and exert an undesirable effect on the final results.

Moreover, the ϵ greedy strategy, which is often used to identify relevant actions, is used to optimize our network. Specifically, ϵ is expressed as shown in equation 8.

$$\epsilon = \max_{0 \le step \le eps\_decay\_steps} \{eps_{min}, eps_{max} - (eps_{max} - eps_{min}) \times \frac{step}{eps\_decay\_steps}\} \tag{8}$$

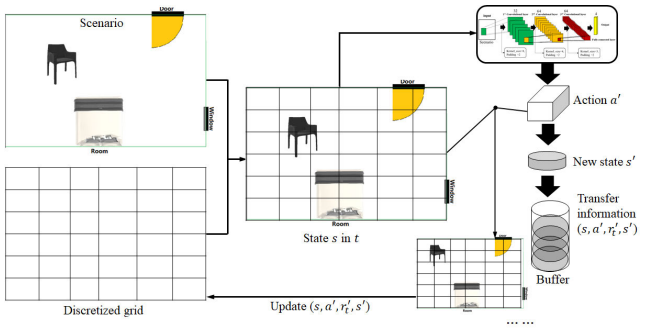

FIGURE 7. The training procedure for the DQN. First, preprocessing of the input images is performed, including the establishment of a discretized grid to resize the input images. In addition, transfer information is saved in a prepared buffer to avoid overgeneralization.

Here, the parameters eps_{min}, eps_{max}, and eps_decay_steps have constant values; in this paper, we set them to $eps_{min} = 0.5$, $eps_{max} = 1$, and $eps\_decay\_steps = 500000$, respectively. The ϵ greedy strategy is applied as shown in equation 9.

$$a' = \begin{cases} N(num\_actions) & r \leq \epsilon \\ a & otherwise \end{cases} \tag{9}$$

Here, the quantity $0 \leq r \leq 1$ is a randomly generated number, and N is a stochastic function that generates a random integer in the range specified by the argument, i.e., $num\_actions$. The quantity $num\_actions$ denotes the number of possible actions; in this paper, $num\_actions = 4$.

In practice, prepared buffer storage is used to collect sets of related transfer information on the state, action, and reward for each agent. In this paper, the number of sets of transfer information that is saved is set to 500000. Every such set of transfer information can be viewed as a tuple. Using the approach discussed above, we can perform the relevant training task. In addition, a termination condition must also be specified. When a specified maximum number of training epochs is reached, the training stage will terminate. However, it is unlikely that every agent will need the same amount of training time. Therefore, we also define a new condition to terminate training. A cumulative reward function, denoted by $J(\theta)$, is utilized to measure whether it is necessary to execute the next state; in this way, we can ensure that the goal of the maximum reward is reached.

$$J(\theta) = \sum_{t=0}^{T} \gamma^{T-t} \times r_t \tag{10}$$

Here, θ denotes the parameters of the DQN, and $\gamma \in [0, 1]$ is a discount factor. When $\gamma = 0$, only the instantaneous reward is considered; to accelerate convergence, we choose to place relatively little emphasis on the long-term reward by setting $\gamma = 0.1$. We consider that if the cumulative reward $J(\theta)$ starts to decrease at time $T + 1$ (specifically, if the derivative of J is less than 0), then the optimal position for the specified agent (i.e., a piece of furniture) has been found.

Algorithm 1 Agent layout optimization based on a DQN

Input: At time $t = 0$, the first existing agent $i = 0$ is in state s_0^0
Output: For n agents in the scene, their optimal states can be represented as $S_{opt} = \{s_{opt}^0, \ldots, s_{opt}^{n-1}\}$
Initialize: The number of agents $n = 1$, $S_{opt} \leftarrow \emptyset$, the initial parameters θ of the DQN, the maximum allowed $T \leftarrow 1000$, $t \leftarrow 0$, $i \leftarrow 0$
for *the* i_{th} *agent* s_i^t *at time* t **do**
  As described in Section **IV-C**, train the DQN
  if *a new agent is added to the scene* **then**
    $n \leftarrow n + 1$
  end
  while $t<T$ **do**
    Based on the current agent state S_i^0, predict a new state S_i^t and compute the reward r_t
    In accordance with **Equation 10**, compute the cumulative reward J
    if $r_{t+1} < J$ **then**
      $s_{opt}^i \leftarrow s_i^t$
      break
    else
      $J \leftarrow \gamma \times J + r_{t+1}$
    end
    $t \leftarrow t + 1$
  end
  $S_{opt} \leftarrow S_{opt} + s_{opt}^i$
  if $i \geq n$ **then**
    Layout is complete for all agents
    break
  else
    $i \leftarrow i + 1$
  end
end

D. TESTING BASED ON THE DQN

Once the DQN has been trained, it can be used to generate reliable predictions consisting of a valid, concrete action for every agent at time T. As mentioned above, our goal is to obtain the best position for every agent; therefore, in accordance with Equation 10, we utilize the DQN for action prediction. Once the termination condition is met, the optimal arrangement can be acquired. The complete procedure of the proposed method is described in detail in Algorithm 1.

E. ADJUSTING FURNITURE POSES

Determining the pose of a piece of 3D furniture is an important problem. The pose of a piece of furniture when it is initially placed in an indoor scene is always constant. However, the method proposed above can be used to modify only the positions of pieces of furniture, not their poses. Nevertheless, furniture pose adjustment is also very important.

Function-behavior-state (FBS) modeling [29] is a traditional design concept that has been widely applied in many design applications, including CAD and scenario layout.

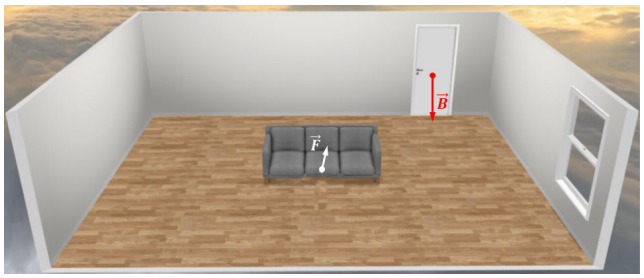

FIGURE 8. The FBS-based design concept in an indoor scenario.

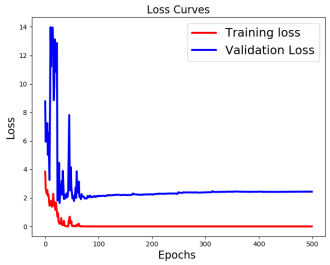

FIGURE 9. The loss curves from the training stage for furniture sketch recognition.

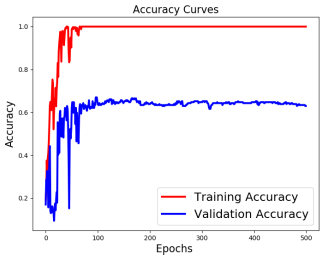

FIGURE 10. A comparison between the training accuracy and the validation accuracy.

The basic principle is that the state of an object is determined by both its function and its behavior. In indoor layout design, every piece of furniture has salient functional characteristics; for instance, the purpose of a chair is to be sat on. Furthermore, the behavior strongly relies on users' choices regarding, e.g., how to conveniently achieve the action of sitting. Ideally, in design studies, function and behavior should be uniformly consistent. Otherwise, the state of the object should be changed; i.e., in this paper, the pose of the piece of furniture should be adjusted. For ease of describing these concepts, two different vectors are defined: the function vector $\vec{F}$ for the piece of furniture and the behavior vector $\vec{B}$ for the room. The definitions of these vectors are illustrated in Figure 8. To conveniently handle pose adjustment in an indoor scenario, there are four different possible directions for these vectors. Moreover, $\vec{B}$ is determined by the position of the door, e.g., when the door is aligned along the x-axis (see Figure 4), $\vec{B}$ is aligned along the z-axis. Then, we can compute the rotation angle β as shown in equation 11.

$$\beta = \arccos(\vec{B} \bullet \vec{F}) \tag{11}$$

Here, the notation $\bullet$ represents the point multiplication operation between the vectors $\vec{B}$ and $\vec{F}$. Finally, a geometrical rotation transformation can be performed based on β to obtain a suitable pose for the piece of furniture.

V. EXPERIMENTS

In this section, the training and testing results related to the proposed framework are presented, and comparisons are performed to demonstrate its superiority.

A. ENVIRONMENT

We implemented our proposed framework using the Python language. More specifically, the Python-based Django framework was adopted for web programming. In addition, we used the open-source TensorFlow library for network training. Fortunately, the ResNet50 network has been integrated into the Keras package, and we can directly call this network to perform related tasks. In these experiments, the program was executed on a PC running the Windows 10 OS with an Intel Core I7-7700HQ processor, 8 GB of physical memory, and an NVIDIA GeForce GTX 1060 with 6 GB of memory.

To train the network, we collected related furniture sketch samples from the Eitz dataset [4]. The total number of collected samples is 800, corresponding to 10 different categories (80 per category). We divided these samples into a training set (70%) and a validation set (30%).

TABLE 3. Comparison of furniture sketch recognition results obtained using various methods.

Method	Avg. accuracy
HOG-SVM [4]	0.56
Multikernel SVM [6]	0.66
Ours	0.68
Human [8]	0.73

B. SKETCH-BASED RECOGNITION OF FURNITURE

The objective of the furniture sketch recognition task is to retrieve related furniture models based on input sketches to provide a better user experience. The results obtained during the training stage in terms of loss and accuracy indicators are shown in Figure 9 and Figure 10, respectively. In the testing stage, to better validate our proposed method, our method was compared with two other published methods: HOG-SVM [4] and multikernel SVM [6]. Many methods exist that can be used to perform this task, but for online recognition applications, we must first consider methods that can effectively handle online requests. From the user perspective, it is meaningless to use a method that can achieve slightly higher accuracy if doing so results in a much longer wait time. Moreover, because the number of available furniture sketches is very limited, we do not compare our method with other deep learning methods, for which a very large number of samples are often used for network training. The comparison of the result is presented in Table 3.

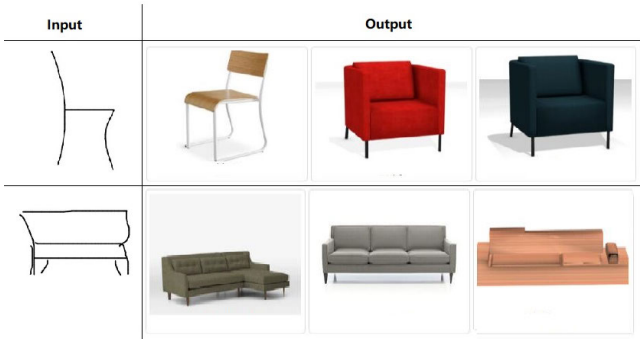

FIGURE 11. Online test results based on our proposed method.

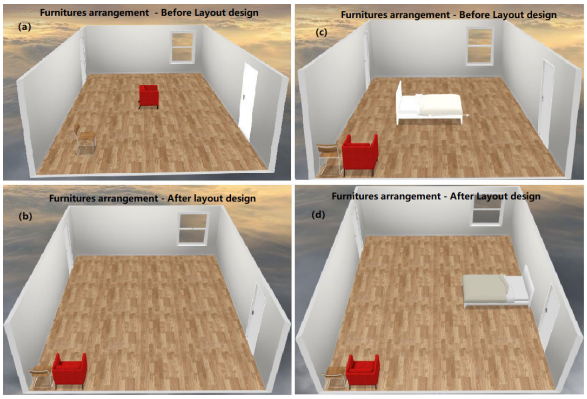

FIGURE 12. Comparison of results before and after layout design based on our proposed approach. In the top left figure (a), two different chairs have been added to the indoor scene. The layout result obtained using the proposed framework is shown in the bottom left figure (b); clearly, these two pieces of furniture have been arranged into new positions. As another example, we further add a bed into the scene, as shown in the top right figure (c); the subsequent layout result is shown in the bottom right figure (d).

As Table 3 shows, the performance of our method is approaching the human level. In fact, if we could collect a greater quantity of samples, we could further improve our results and potentially even surpass the human level. The final test results are shown in Figure 11.

As Figure 11 illustrates, we are able to conveniently retrieve relevant furniture models based on freely hand-drawn sketches. For the majority of users, this capability will provide an easier and more convenient method of adding desired furniture into virtual scenes.

C. INDOOR LAYOUT RESULTS

For indoor layout design, we adopt the deep Q-learning method to enable online execution of the design task. Examples of the layout results are shown in Figure 12.

From Figure 12, it is not difficult to see that the layout design task can be smoothly and correctly completed. Intuitively, the furniture is initially randomly placed in the space of the virtual room, and it is very inconvenient to manually manipulate such 3D objects with a mouse. Moreover, simple drag-and-pull manipulation of such furniture objects is a very dull and time-consuming task. By contrast, our proposed approach allows a reasonable, conventional layout design

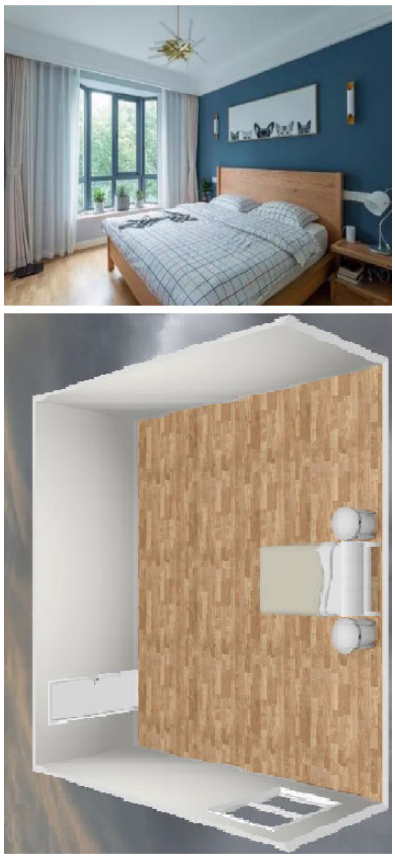

FIGURE 13. A comparison between the layout design results of the proposed framework and photographs of real existing indoor layouts.

to be rapidly completed, thus considerably enhancing the convenience of the design task.

However, the assessment of the final result is inherently subjective; correspondingly, there is a lack of effective quantitative criteria for evaluating the final result. Thus, it is difficult to objectively measure the success of the outcome. Therefore, we instead present a qualitative comparison with photographs of real existing indoor layouts to validate the feasibility of our framework.

Figure 13 presents a comparison between the layout design results and photographs of real existing layout designs retrieved from the Internet. We find the following similarities:

1. The bed is placed near the window. In this case, the layout design result is very close to the photograph.
2. The bed is placed in the center of the room, at a certain distance from the door. In addition, the nightstands are placed on either side of the bed. This result is also very similar to the photograph.

Clearly, the design results of our framework are highly consistent with popular layout design principles. Therefore, this qualitative comparison effectively demonstrates the feasibility of the proposed framework.

D. DISCUSSION

We have proposed a framework for completing popular indoor layout design tasks. Related experiments validate the

feasibility of our proposed method. However, several problems still exist that will need to be solved. First, indoor layout design is highly subjective, and it is difficult to perform quantitative comparisons for a variety of scenarios, such as different room sizes or room types, which often generate different results. Thus, it would be desirable to develop an objective metric for evaluating layout results. Second, indoor layout design in 3D space requires not only the 3D positioning of furniture but also many other factors, such as indoor illumination. In particular, the illumination conditions in a room have a great effect on the aesthetics of its layout and consequently must be considered in human-centered indoor layout design. Third, in this study, we have achieved layout result based only on certain popular design principles; we have not fully considered the implementation of various personalized design rules, such as the desire to place a sofa near the window for reading purposes. Additionally, wall-based layout design, e.g., the layout of family photos on the wall, has not been addressed, although this should also be a goal of human-centered layout design. Finally, to broaden the scope of application, more room types should be considered, e.g., dining rooms, living rooms, kitchens, or even the whole house.

VI. CONCLUSION

In this paper, we propose a framework to support indoor furniture layout design in a virtual room. There are two important tasks to consider, i.e., the interactive addition of new furniture into a scene and the intelligent completion of the indoor layout design. We adopt a CNN for furniture sketch recognition to achieve the function of more interactive and convenient furniture addition. Moreover, the deep Q-learning method is used to develop a new pipeline for layout design based on state-action selection. The results of related experiments performed to validate our proposed framework show that it is completely feasible.

However, our proposed approach still has certain limitations. More concretely, in regard to furniture sketch recognition, there are not currently sufficient samples available for analysis. In the future, we plan to collect a greater quantity of samples to train our network. In addition, a more complex network architecture should be used to further improve the accuracy. Moreover, for the layout design task, the Asynchronous Advantage Actor-Critic (A3C) method will be considered to further improve the performance and obtain better results. In addition, as is well known, the proper layout of indoor scenes inherently involves certain subjective factors, such as individual preferences and habits. Therefore, further effort is still required to enable more personalized design of indoor layouts based on, e.g., the learning of individual style preferences using a CNN. Furthermore, to some extent, effective layout design is likely to require further study of existing exemplary style samples, such as photographs selected by individuals to illustrate their favorite design styles, rather than relying solely on agent-based decisions. In addition, in the future, further research will need to be conducted to address indoor illumination layout design and wall-based layout design.

REFERENCES

[1] T. Lu, C. L. Tai, and F. Su, "A new recognition model for electronic architectural drawings," *Comput.-Aided Des.*, vol. 37, no. 10, pp. 1053–1069, 2005.
[2] M. F. A. Jabal, M. S. M. Rahim, and N. Z. S. Othman, "A comparative study on extraction and recognition method of CAD data from CAD drawings," in *Proc. Int. Conf. Inf. Manage. Eng.*, 2009, pp. 709–713.
[3] C. L. Zitnick and D. Parikh, "Bringing semantics into focus using visual abstraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3009–3016.
[4] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" in *Proc. SIGGRAPHR*, 2012.
[5] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using Fisher vectors," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–9, 2014.
[6] Y. Li, T. M. Hospedales, S. Gong, and Y.-Z. Song, "Free-hand sketch recognition by multi-kernel feature learning," *Comput. Vis. Image Understand.*, vol. 137, pp. 1–11, Aug. 2015.
[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[8] Q. Yu, Y. Yang, F. Liu, Yi-Zhe Song, T. Xiang, and T. M. Hospedales, "Sketch-a-Net: A deep neural network that beats humans," *Int. J. Comput. Vis.*, vol. 122, pp. 1–15, May 2017.
[9] F. Wang, L. Kang, and Y. Li, "Sketch-based 3D shape retrieval using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1875–1883.
[10] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[11] Y. I. Parish and P. Müller, "Procedural modeling of cities," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 301–308.
[12] P. Muller, P. Wonka, and S. Haegler, "Procedural modeling of buildings," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2006, vol. 25, no. 3, pp. 614–623.
[13] G. Chen, G. Esch, and P. Wonka, "Interactive procedural street modeling," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2008, vol. 27, no. 3.
[14] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2010, vol. 29, no. 6, pp. 1–12.
[15] L. Yu, S. Yeung, and C. Tang, "Make it home: Automatic optimization of furniture arrangement," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2011, vol. 30, no. 4, pp. 1–11.
[16] P. Quax, J. Liesenborgs, and A. Barzan, "Remote rendering solutions using Web technologies," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4383–4410, 2016.
[17] M. Zorrilla, A. Martin, and J. R. Sanchez, "HTML5-based system for interoperable 3D digital home applications," *Multimedia Tools Appl.*, vol. 71, no. 2, pp. 533–553, 2014.
[18] M. Fisher, D. Ritchie, and M. Savva, "Example-based synthesis of 3D object arrangements," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2012, vol. 31, no. 6.
[19] K. Xu, K. Chen, and H. Fu, "Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models," in *Proc. Int. Conf. Comput. Graph. Interact. Techn.*, 2013, vol. 32, no. 4, pp. 1–15.
[20] W. Xu, B. Wang, and D.-M. Yan, "Wall grid structure for interior scene synthesis," *Comput. Graph.*, vol. 46, pp. 231–243, Feb. 2015.
[21] P. Song, Y. Zheng, and J. Jia, "Web3D-based automatic furniture layout system using recursive case-based reasoning and floor field," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 5051–5079, 2019.
[22] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, C. Tu, B. Chen, D. Cohen-Or, and H. Zhang, "GRAINS: Generative recursive autoencoders for INdoor scenes," *ACM Trans. Graph.*, vol. 38, no. 2, pp. 12–31, 2019.
[23] K. Wang, M. Savva, and A. X. Chang, "Deep convolutional priors for indoor scene synthesis," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 70–83, 2018.
[24] S. Song, F. Yu, and A. Zeng, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1746–1754.

[25] K. Xu, K. Chen, and H. Fu, "Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 123–135, 2013.
[26] L.-F. Yu, S.-K. Yeung, D. Terzopoulos, T. F. Chan, S. J. Osher, and C.-K. Tang, "Make it home: Automatic optimization of furniture arrangement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 86–100, 2011.
[27] J. Wu, C. Zhang, and T. Xue, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 82–90.
[28] H. Chatbri and K. Kameyama, "Towards making thinning algorithms robust against noise in sketch images," in *Proc. 21st IEEE Int. Conf. Pattern Recognit. (ICPR)*, Tsukuba, Japan, Nov. 2012, pp. 3030–3033.
[29] Y. Umeda, M. Ishii, and M. Yoshioka, "Supporting conceptual design based on the function-behavior-state modeler," *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 10, no. 4, pp. 275–288, 1996.
[30] S. H. Bae, R. Balakrishnan, and K. Singh, "EverybodyLovesSketch:3D sketching for a broader audience," in *Proc. 22nd Annu. ACM Symp. User Interface Softw. Technol.*, Victoria, BC, Canada, 2009.
[31] L. B. Kara and K. Shimada, "Sketch-based 3D-shape creation for industrial styling design," *IEEE Comput. Graph. Appl.*, vol. 27, no. 1, pp. 60–71, Jan./Feb. 2007.
[32] B. Milosevic, F. Bertini, E. Farella, and S. Morigi, "A SmartPen for 3D interaction and sketch-based surface modeling," *Int. J. Adv. Manuf. Technol.*, vol. 84, pp. 1625–1645, May 2016.
[33] W. Zhou and J. Jia, "A learning framework for shape retrieval based on multilayer perceptrons," *Pattern Recognit. Lett.*, vol. 1, no. 117, pp. 119–130, 2019.
[34] W. Zhou, J. Jia, C. Huang, and Y. Cheng, "Web3D learning framework for 3D shape retrieval based on hybrid convolutional neural networks," *Tsinghua Sci. Technol.*, vol. 25, no. 1, pp. 93–102, 2020.

WEN ZHOU (M'18) received the Ph.D. degree from the School of Software Engineering, Tongji University, in 2018. Since November 2018, he has been with the School of Computer and Information, Anhui Normal University, Wuhu, China, where he is currently a Lecturer. His research interests include medical image analysis, WebVR visualization, virtual reality, and machine learning. He is also a member of the Chinese Computer Federation (CCF) and the Chinese Association for Artificial Intelligence (CAAI).

WENYING JIANG received the B.Sc. degree in digital media technology from Huaibei Normal University, in 2019. She is currently pursuing the master's degree in computer science and technology with Anhui Normal University, China. Her research interests include machine learning and 3-D visualization.

WEIXIN BIAN received the Ph.D. degree in computer science from the China University of Mining and Technology, Xuzhou, China, in 2018. He joined the School of Computer and Information, Anhui Normal University, in 2005, and is currently an Associate Professor. His research interests include machine learning and digital image analysis.

BIAO JIE received the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, China, in 2015. He joined the School of Computer and Information, Anhui Normal University, in 2006, and is currently a Professor. His research interests include machine learning and medical image analysis.