International Journal of Architecture and Urban Development
Vol. 14, No. 3, P.18-32 .Summer 2024

E-ISSN 2345-2331
Applied Research Article
DOI: 10.71711/ijaud.2024.796341

Exploring Various Residential Layout Generation Using Conditional Generative Adversarial Networks (Case Study: Apartments In Districts 1 and 2 of Hamadan City)

[1] Mahsa Hamouni, [2] Hossein Soltanzadeh, [3] Seyed Hadi Ghoddusifar, [4] Muharram Mansoorizadeh*

[1] Ph.D. Student, Department of Architecture, Faculty of Art & Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran.
[2] Professor, Department of Architecture, Faculty of Architecture and Urban Planning, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*
[3] Assistant Professor, Department of Architecture, Faculty of Art and Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran.
[4] Associate Professor, Department of Computer Science, Faculty of Engineering, Bu-Ali Sina University, Hamadan, Iran.

Recieved 2023.11.11 ; Accepted 2024.10.29

ABSTRACT:Artificial intelligence technology has become an influential and trending topic in architectural layout design. The core technology of AI, machine learning, has attracted the attention of architects as a decision-making tool. The focus of many studies that apply machine learning to layout design is using the generative adversarial network (GAN within a given boundary. Previous research demonstrates that training a GAN with labels can help a computer understand how spatial elements relate and the logical relationship between spatial elements and boundaries. However, this paper applied conditional GAN to generate space layouts with given boundaries and supplementary conditions. The supplementary conditions provide designers control over the generated layout plans by satisfying input boundary and user requirements. It also allows designers to generate different layout plans within the same boundary. To achieve this, a method for dividing image channels is proposed so that both given boundaries and supplementary conditions become the model's input. The dataset consists of 660 apartment plans in Hamadan. The dataset is split into a training set and a test set. The training set includes 594 images, and the test set includes 66 (10%) of the images. After training the model with the training set, the model is tested using the test set. Finally, the model outputs are evaluated based on quantitative and qualitative methods. The results show that the supplementary conditions provide further guidance to the model for space layout generation based on user preferences and reduce the image quality problems of the synthetic images.

Keywords: *Spatial layout design, Machine learning, image–to–image translation, Conditional GAN.*

INTRODUCTION

Computer-aided architectural design was introduced in the 1950s and has since experienced different periods of development (Caetano et al., 2020). Space layout design is one of the research branches of computer-aided architectural design. It involves determining possible locations and dimensions for interrelated objects that adhere to all design requirements (Michalek et al., 2002). In other words, it proposes a layout following topological and geometrical constraints. Many objective and subjective factors influence these constraints. Objective factors could be defined as numerical rules such as design standards. However, subjective factors like aesthetics are based on the designers' expertise (Rahbar et al., 2019). There are two algorithmic approaches to generating design solutions. First, the rule-based method views the design process as an optimization problem and applies human-defined rules to find a solution that satisfies the requirements, such as a genetic algorithm (Zheng & Yuan, 2021). Researchers have used these algorithms to improve their design work in architectural research for many years.

Users of this approach must explicitly describe the numerical rules and objective function f(x) to teach the algorithms to discover the solutions. This means that to express a design's quality, designers must provide a clear evaluation function. It is challenging for designers to comprehend and mathematically express precise numerical rules and objective functions (Whitley, 1994). As mentioned before, in

*Corresponding Author Email: ***hos.soltanzadeh@iauctb.ac.ir*** ***ORCID: 0000-0003-3974-5205***

addition to the objective criteria that may be tested and assessed logically, numerous subjective factors deal with the designer's experience and expertise. Thus, hard-coding them would not be possible or effective. There are still limitations even if architects could hardcode these rules in a computer program. Because each designer has some degree of subjectivity in their design decisions, this scenario has the drawback of resulting in a biased deterministic intelligence.

The second approach to generating design solutions is the data-driven method that contains artificial intelligence and machine learning (Hong et al., 2020). In contrast to the previously described approach, this data-driven method has shown that it can derive information from data and use that knowledge to make decisions (Creswell et al., 2018). The intersection of architecture and artificial intelligence (AI) is a multidisciplinary domain that has experienced technological advancements. AI integration in architectural design is not a novel idea. In reality, one of the earliest projects to express a desire to introduce an automated assistant in design and architecture was the MIT CAD project (1959–1970) (Cardoso Llach, 2015). As the fourth industrial revolution advances, interest in artificial intelligence and big data is fast growing, systems are being created for intellectualization in various disciplines, according to Yoon et al. (2018). Architecture is one of these fields where big data may be used to support a system for making predictions by analyzing the outputs (Özerol & Selçuk, 2022). With the advent of machine learning and more powerful computers in the scientific field, there are even more chances to apply data-driven methods to address the space layout problem. The design has extensively used artificial intelligence technologies, particularly machine learning methods like neural networks.

In this method, the model could learn non-numerical rules from the dataset (Hong et al., 2020). An algorithm can progressively learn patterns in a context by observing large amounts of data and then determining which rules are most likely to produce an appropriate output. With this method, architectural rules could be learned without the necessity for enforcement of what those rules are. The inferred outputs may depend on the input data due to this statistical approach to intelligence. A biased dataset generates non-generalizable outputs. By forcing bias in a dataset, however, this dependence can be positively used in architecture. This would result in exploring specific rules and features (Ferrando, 2018). Machine learning can conduct statistics on empirical data and generate probability density estimates for ill-defined problems that are difficult to define (Newton, 2019). Machine learning is distinct from other algorithms as a decision-making tool in space layout generation. A machine learning model is trained using a dataset, and the trained model is then used to generate output images from the input images.

In contrast to rule-based approaches, the mappings between inputs and outputs are computed through machine learning by taking the input controlling factors and the output solutions as training data. To be more precise, the procedure is shown as a network of neurons that describes the computational relationships between the inputs and outputs (McCulloch & Pitts, 1943). To optimize the neural network's parameters, the training process is executed by a program with backpropagation once the input and output design data have been provided (Werbos, 1974). Using this network, the user can add new feature data and receive feedback on the output design data. By implementing this method, the user develops a data translator that simplifies input data conversion into the output design by feeding the neural network only the design features and their associated design outcomes instead of the objective functions. Fig. 1 shows the difference between rule-based and machine-learning methods (Zheng & Yuan, 2021). The generative adversarial network (GAN) has shown remarkable outcomes in developing 2D designs as a machine learning model. Recently, relevant studies have applied GAN to layout generation and proved its effectiveness.

GANs have shown a generative sophistication level that previous studies have not equaled. Numerous design-related disciplines could be impacted by the capacity of GANs to learn from examples and extrapolate that learning into creating new instances. Goodfellow and his colleagues are the first team to propose the generative adversarial network in machine learning (Goodfellow et al., 2014). According to Goodfellow, the fundamental challenge for artificial intelligence is finding a solution for easy tasks that people can accomplish but find challenging to explain. Such issues are resolved intuitively by people. Formally describing the process of designing in architecture is also quite challenging. An architect can design a building plan without thinking about any cognitive aspects of the design. The process that is taking place in the mind behind the drawing, however, is extremely detailed and sophisticated. To train an algorithm, it would not be possible or effective to describe these design processes. Instead of attempting to explain the design for automated design processes formally, an algorithm can infer meaningful solutions through drawing datasets. The dataset concept allows algorithms to evaluate data, and deep learning algorithms can be trained without describing processes (Goodfellow et al., 2016). By giving image data in pairs, GANs have been applied to a wide variety of tasks.

On the other hand, images have traditionally played a crucial cognitive role in the creative process of generating architectural designs (Lawson & Dorst, 2009), so these developments in artificial intelligence research are especially encouraging for architecture. Architects have used images as sources of inspiration, metaphors, analytical tools, and justifications (Purcell & Gero, 1998). Architects use images deliberately and unconsciously in their cognitive processes to show the fundamental qualities of their architecture by representing their perspectives, memories, and origins (Olgiati, 2013). Architects can start their designs and overcome inertia using images as precedents (Huang et al., 2021). A useful tool for decision-making during the architectural design process is precedent. They are used as a guide for developing space layouts in both professional and educational settings (Grover et al., 2018). Generative adversarial networks (GANs) provide this opportunity for learning from images and extrapolating that learning into generating new instances. It can find a solution based on expert knowledge. Generative adversarial network is one method that intersects design and machine learning.

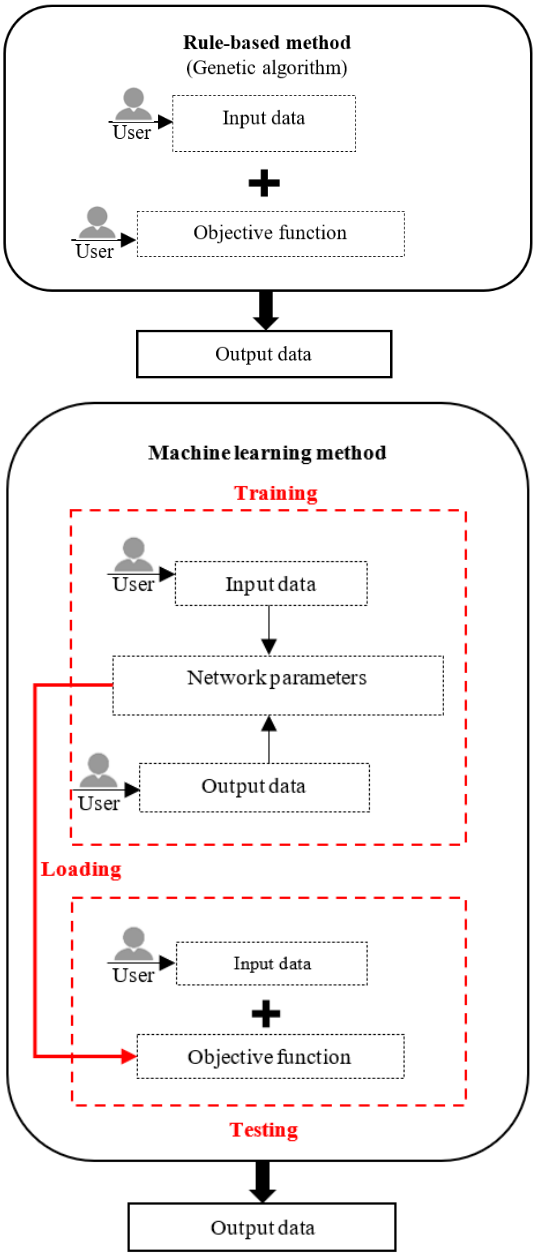


Fig. 1: Difference between Rule-based and machine learning method procedure (Source: Zheng & Yuan, 2021)

In contrast to conventional rule-based generative techniques, such systems use a more holistic and nondeterministic approach to mapping and generating spatial predictions for floor designs. Furthermore, it has been suggested that this system can automate the time-consuming process of incorporating design rules into generative models (Rodrigues & Duarte, 2022). This paper applied a conditional GAN (c-GAN) to generate automated space layouts with given boundaries and user requirements. Conditional GAN was introduced by Mirza and Osindero (2014). It focuses on generating automated 2D layouts with a fixed footprint, windows, and an entrance door that is input to the suggested algorithm from the designer.

In contrast to previous methods that applied c-GAN for generating space layouts with given boundaries, this method guided the model further by adding room centroids as supplementary conditions. These additional conditions give designers control over the generated layout plan and make the model flexible. It also allows designers to generate different space layout plans within the same boundary. In other words, the original Pix2Pix model can only support one image input with three channel images. In this method, the input image channel was modified from three to six channels, allowing for the simultaneous input of two images and enabling interaction with the model by adding supplementary conditions (Fig.2). Fig.3 shows the research methodology flowchart. For this purpose, the following actions are taken: preparing a specific dataset for training the model consisting of 660 apartment layout plans in Hamadan, using the dataset to train the model, testing the model, and then assessing the model using the test findings.

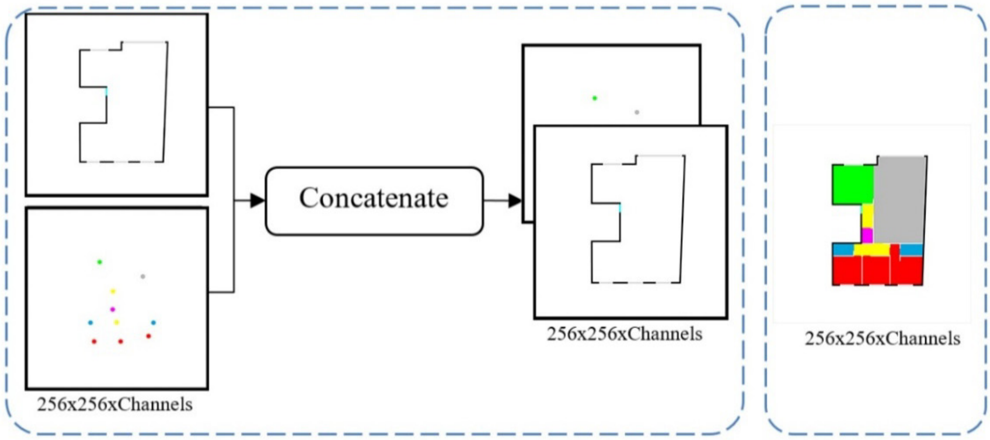


Fig. 2: Achieve the simultaneous input of two images (Source:Authors)

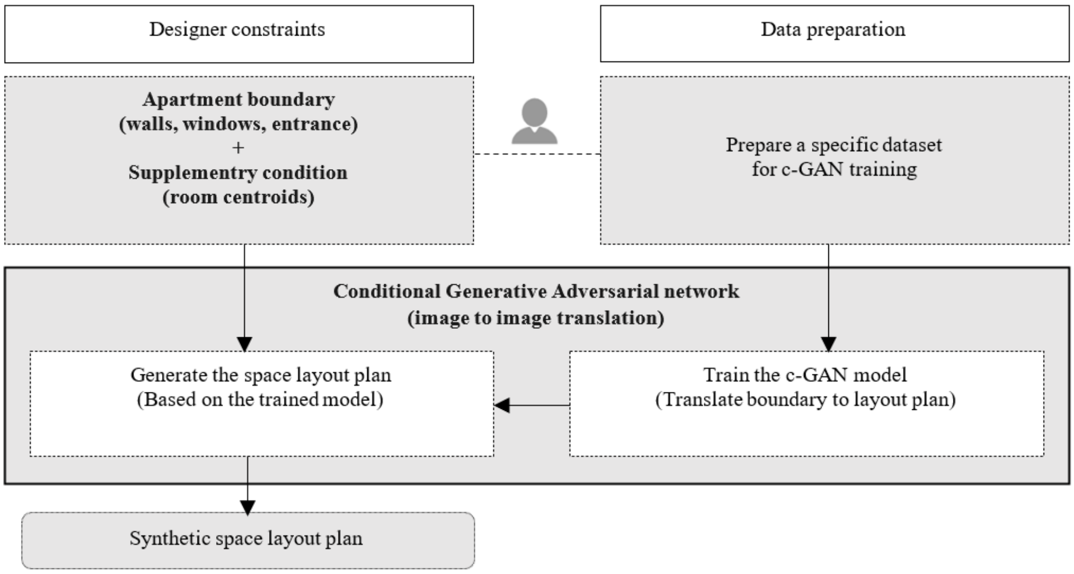


Fig. 3: Research methodology flowchart (Source:Authors)

RESEARCH BACKGROUND

Review of the data-driven methods: Methods not based on GAN

Architectural design images make up most of the research objects in the studies on machine learning applied to architecture. Wu et al. apply a data-driven method for generating floor plans with given boundaries. It includes two steps. One step concerns defining topological constraints, which determine how rooms connect and where they are located. Another step determines the geometrical constraints using CNNs to calculate room sizes and wall positions. To train networks for automated floor plan generation with a given boundary as input, they create the RPLAN dataset with more than 80k floor plans. They suggested a process that starts with finding the room locations. Then, using an encoder-decoder network and certain rules, walls are placed, and windows are added. Users can select the location of specific rooms using this generation method, but additional constraints are not considered (Wu et al., 2019).

For generating floor plans, Hu et al. present a method that integrates deep learning convolutional neural networks and graph neural network generative modeling on the RPLAN dataset. Layout graphs represent design constraints such as type and number of rooms. Their approach enables users to define high-level requirements that are then transformed into a layout graph. Then, the RPLAN dataset is searched for floor plans that satisfy the criteria in layout graphs. These floor plans are retrieved and ranked by how closely their boundaries match the user-provided boundary. By adjusting the graphs such that the room nodes are located inside the building, the chosen layout graph is transferred to the boundary. Finally, the model generates corresponding floor plan images for each layout graph and the input boundary (Hu et al., 2020). Eisenstadt et al. suggest an adaptive approach to facilitate retrofitting room arrangements in frames. Their research proposed a method for generating various floor plans during the initial phases of architectural conceptual design (Eisenstadt et al., 2019). Graph neural networks are used by Chang et al. to generate volumetric designs conditioned on an input program graph (Chang et al., 2021). Merrell et al. propose an approach to generate building layouts automatically. In their work, two methods are used: making an architectural program that lists the interior spaces and then converting the architectural program into a layout plan. The generation of architectural programs involves training a Bayesian network. With this data-driven methodology, relationships between attributes that represent semantic structures, such as common adjacencies between various room types, could be learned. An optimization is used to create the floor plans. This results in floor plans that are visually credible and correspond to some high-level requirements, but many possible client and site aspects may also be disregarded in the process (Merrell et al., 2010).

Methods based on GAN

Machine learning has significantly influenced architectural design. Evolutionary methods, including genetic and heuristic algorithms, were mostly used until deep learning was introduced. Due to limitations, most researchers have used generative design (i.e., form-finding, space layouts, and material-based learning), which complies with generative adversarial networks. One of the first researchers to study this topic was Hao Zheng. In 2018, he used a conditional GAN to demonstrate that building plans, urban plans, and satellite images could be generated from conditional input like footprints. He successfully generated plausible apartment plans and explained the working rules. In the next study, Huang and Zheng used a conditional GAN to identify rooms in a layout plan and generate a furnished plan from a color-labeled image. They trained the model using a dataset of 115 image pairs, including an architectural plan and a color-labeled map where the colors denote different room types. The plans were subsequently generated based on the attributes discovered by GANs (Huang & Zheng, 2018). Based on the footprint, Nathan used GANs in his Harvard thesis to address program repartition in single-family modular homes. Without specified fenestration, his model converted a footprint into programmatic patches of color (Peters, 2018). As et al. applied infoGAN to generate graph structures that indicate the single-family houses' layout. For generative concept design in architecture, they implemented deep learning methods. Data were acquired for this graph-based research from analyzing the spatial relationships of fifteen homes in Revit. They were converted into graph forms, with nodes denoting rooms and edges denoting the connections among them. High-performing sub-graphs of the 3D model were then identified using deep neural networks, which were later composed into new conceptual designs.

Additionally, using the same design data, the study trained generative adversarial networks to generate novel designs in graph representation (As et al., 2018). Newton presents a GAN-based technique using a generative adversarial network to generate new designs from examples. One hundred façade images were utilized to train the model, and 15 images were prepared to test the facade designs (Newton, 2019).

To generate a layout design from the image of an apartment footprint, Rahbar applied a conditional GAN model. It focuses on generating automated 2D layouts with a fixed footprint, windows, and an entry door that is input to the algorithm from the designer. A dataset of 300 apartment layouts was created to train the model (Rahbar et al., 2019). The work of Chaillou, who developed ArchiGAN, has another application of c-GAN. The area on a project site (building footprints) that the building occupies is all that is needed to get relevant outputs.

The entire procedure consisted of three key stages of the model. In the first step, they used a parcel outline to generate a footprint of the building. In the second step, program repartition, every space inside the footprint was given a color. In the third step, the furnished layout was generated. He employed the model to develop multiunit residential layouts within site boundaries (Chaillou, 2019). Zheng et al. conducted a generation experiment only on the boundary condition of an apartment as input. The authors used two different house plan datasets from China and Japan to train two models. To learn a building's house design, they built a c-GAN-based neural network, which then generated the house plan layout inside a predetermined boundary. The findings demonstrated that even with just the building boundary as input, GANs can still generate outcomes. A comparison of the results found that the Chinese home plans were simpler to grasp, indicating that they were more systematic and template-based (Zheng et al., 2020). The c-GAN model was used by Liu et al. to generate a campus layout on the site plan scale using campus boundaries and nearby roadways. They have attained reasonable findings with a data set that includes the site plans of elementary schools and universities. Preliminary design goals were defined for this study, and it was ensured that the data set size complied with the design requirements (Liu et al., 2021). To generate floor plans, Nauata et al. suggest a generative adversarial network based on the graph constraint that takes an architectural constraint (the type and count of rooms with their spatial adjacencies) as bubble diagrams. Compared with other methodologies, their approach receives high scores for realism and diversity in user studies. The most frequent failure modes are misaligned, inaccessible, and incorrect room geometry and size when given a specified room type. Given an input graph, the generated layout's shape changes depending on the ground truth since the boundary is not considered (Nauata et al., 2020). They also introduce a novel generative adversarial layout refinement network. It enables iterative refinement, where a previously generated layout becomes the next input constraint. The iterative generator improves a metric of choice by controlling input constraints during iterative layout refinement (Nauata et al., 2021). Liu et al. use a c-GAN model to design and analyze the layouts of Chinese private gardens. Their contribution relates to enhancing the training process with a limited data set (30 gardens) through consecutive stages in which the number of labels (7, 9, and 10 labels for each training) is increased gradually (Liu et al., 2022). Rahbar and colleagues suggest a hybrid approach that combines two separate methods. To generate a graph layout that meets the topological requirements, agent-based modeling's hierarchical phases must first be simulated. Using a deep learning approach, a conditional GAN model converts the heat maps into layout plans (Rahbar et al., 2022).

Karadag et al. present a machine learning-based model to generate classroom layouts for educational buildings. In this project, a c-GAN model was applied. The first model generated zones based on given boundaries and the second model was trained to generate arrangements for classroom furniture based on given zonings (Karadag et al., 2022). By demonstrating that DCGANs can learn and reproduce floor designs in the style of Andrea Palladio, Uzun and colleagues accomplished a significant advancement. The reproduced floor layouts are statistically comparable to the original floor plans (Uzun et al., 2020). Zhao applied a c-GAN model to generate space layouts for hospital floor plans. The model can generate high-quality color-coded masks linked to various spatial functions in a hospital using a bounding box as inputs, which has practical engineering applications. The limitations imposed by the training data, which human designers produced, result in layouts similar to human design (Zhao et al., 2021). Table 1 presents an overview of these data-driven methods (Source: Authors). In conclusion, recent research has used various neural networks to learn from architectural design. This shows that converting the design data into images and feeding it into neural networks is practical (Weber et al., 2022).

Table 1: Data-driven methods in literature (Source: Authors)

Author(s)	Date	Matching	Inputs	Output
Huang & Zheng	2018	Conditional GAN	Labeled image of floor plans	Furnished floor plan
Peters	2018	Conditional GAN	Building boundary	Color-coded rooms in boundary manual tracing for vectors
Rahbar et al.	2019	Conditional GAN	Building boundary	A colored synthetic layout plan
Chaillou	2019	Conditional GAN	Building boundary	Color-coded rooms in boundary manual tracing for vectors
Wu et al.	2019	1. Convolutional neural network to locate room 2. Convolutional neural network for wall placement	Entrance, Building boundary	Wall map, vector representation of the layout
Zheng	2020	Conditional GAN	Building boundary	Furnished apartment floor plan
Hu et al.	2020	1. Graph neural network 2. Convolutional neural network floor plan image	Entrance, building boundary, number and type of rooms	Floor plan image, vectorized floor plan
Nauata et al.	2020	Convolutional message-passing neural networks	Bubble diagram (Program graph)	Fitted rectangles as rooms
Chang et al.	2021	1. Graph neural network 2. Graph neural network voxel graph	Program Graph, User input during generation	Volumetric pixel grid representation of a program
Nauata et al.	2021	1. Relational GAN 2. Convolutional message-passing neural networks	Program graph	Vector representation of the layout
Rahbar et al.	2022	1. Agent-based modeling for the program graph 2. C-GAN for generating a layout plan	Building Boundary Program Graph	Color-coded rooms in boundary
Karadag et al.	2022	Conditional GAN	Boundary of building	Zonings based on given boundaries Furnished floor plan based on given zonings

MATERIALS AND METHODS

Generative Adversarial Networks (GANs) have demonstrated the ability to generate new image examples from the learned distribution of a training set. Recently, GAN-derived research has focused on image-generation tasks in computer vision. This paper investigates using c-GAN to extend the original GAN framework to generate apartment space layout plans. In image-to-image translation problems, c-GANs are appropriate for learning a mapping from an input image to a target image. For this purpose, the following actions are taken: creating a specific dataset for training the model, using the dataset to train the model, testing the model, and then assessing the model using the test findings. This paper suggests an approach for generating layout plans using additional conditions to satisfy both input boundary and user requirements. Requirements are encoded in "centroids" as condition images and stacked with the boundary input to train the model to generate layout plans that meet both the input boundary and user requirements. The centroids represent room types (indicated by color) and their locations.

Image-to-image translation based on c-GAN

GANs have recently emerged as powerful tools for generating images, consisting of a generator and discriminator networks. It is founded on the idea that these two neural networks compete. A generator model named G is used to generate new images. An image can be classified as real (data set images) or fake (synthetic images) using a discriminator model named D. A set of data is used to train the D model to identify images. When trained, the model can distinguish the differences between a real example taken from the dataset and a fake image that isn't from the dataset. The G model is trained to generate images that look like images from the dataset. The D model gives some feedback on the output quality of the G model as it generates images. In reaction, the G model adjusts, generating images that are even more realistic. The G model tries to trick the D model, while the D model tries to recognize the fake images. In a competitive phase, the two networks are simultaneously trained. Through this feedback loop, both models enhance during training until the generated samples can no longer be recognized (Chaillou, 2019).

The difference between the generated sample and the images from the dataset is calculated throughout this training process by the loss function. The loss function is then sent to the network as feedback (backpropagation) to enhance the generated outputs. GANs can be divided into conditional GANs (c-GAN) and unconditional GANs. Besides tweaking the training dataset, unconditional GANs have limited control over the generated outputs. Unconditional GANs generate outputs based on random noise vectors. However, it is impossible to directly control the modes of the generated outputs for unconditioned generative models. Conditional GANs provide finer control over the output by allowing the user to input additional information to guide the model in generating the desired outputs. It depends on some auxiliary information **y**, which conditions the discriminator and generator. The generator and discriminator both receive this information as additional input. As an image can be used as the supplementary condition, c-GANs are appropriate for learning how to map from an input to an output image. This type of GAN generates space layouts that satisfy both the input boundary and additional conditions.

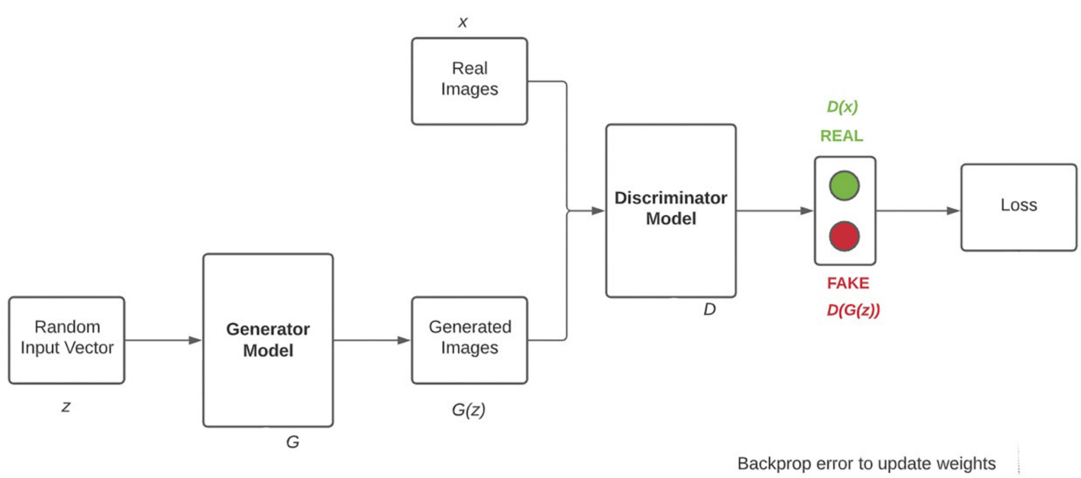


Fig. 4: The structure of Conditional GAN (Source:Authors)

Fig.4 shows the structure of the conditional GAN. According to Isola et al. (2017), c-GANs can solve these tasks. They suggested Pix2Pix, an image-to-image translation method that generates new images using the semantic data specified in the input image. It became a practical basis for complicated image-to-image translation tasks, including converting sketches to images, labeling scenes, day to night, and more. Pix2Pix is a type of c-GAN. Wang et al. (2018) provide pix2pixHD by improving the pix2pix framework further. The authors suggest adding low-dimensional feature channels as input to enhance control over the synthetic images. This model generates space layouts that satisfy the input boundary and additional conditions.

Preparation of the data set

This study's dataset is a collection of 660 apartment floor plans in Hamadan of Districts 1 and 2. Proper and standard floor plans are selected to create the dataset. The following requirements are met by each floor plan included in the final dataset: The floor plan's total area is between 120 and 175 m2, there is a living room, and there are between 7 and 10 spaces. Regarding the procedure of c-GAN training, a pair of same-sized images is required as a training dataset. A pair of images was created for each floor plan in the dataset to provide the network with two separate image domains to learn a mapping between inputs and outputs to perform an image-to-image translation task.

For this purpose, AutoCAD software was used to generate the training dataset, which consists of these image pairs. Interior areas, exterior walls, windows, and the entrance are some information stored in the input images. The staircases, furniture, interior doors, and columns are removed from the floor plans. Terraces are regarded as an extension of adjacent areas. There are no differences in the thickness of the boundary wall for each dataset. Beyond the data saved in the input images, the ground truth images include further information indicating the interior walls and the room types. Each space is labeled with a specific color during data preparation (Fig.5). The colors used to label the input and ground truth are shown in Table 2. The images were finally split into two groups: training data and test data. Of the total images, 594 are in the training data, and 66 (10%) are in the test data. Each image pair consists of an apartment's boundary and a color-labeled image (Fig.6). The images are all 256 × 256 pixels in size. As previously mentioned, centroid images are also needed as additional conditions. The color-labeled images as ground truth images of a test set are utilized to extract the room centroids as an additional condition (Fig.7). To achieve this, the primary step is to eliminate the walls, windows, and entrance, leaving only the rooms. After extracting different room shapes, the thresholding function receives this image, and a center is calculated for each room shape located in a specific position (x, y). Furthermore, the number of adjacency points for each centroid in this experiment has been set to 50, determining the centroids' size in the condition images. This paper suggests an approach for generating space layout plans using centroid images as an additional condition to satisfy both input boundary and user requirements (Fig.8). In the next section, the c-GAN model is trained on all of the 594 pair images with condition images, which is explained.

Training the c-GAN model

The c-GAN model is trained using the training data set following the data preparation step. The process of optimizing machine learning model parameters using input and output training datasets is known as model training. By providing the centroid images as supplementary conditions during training, the experiment aims to determine how adding conditions to the c-GAN model affects it. The RGB images for training are loaded using the Python Imaging Library (PIL). The data loader converts these images to tensors before being fed into the networks. The tensors have the shape (C, H, W). H and W represent the image's height and width, while C denotes the number of channels. When running the scripts, the user can choose a configurable number of input channels for the generator and discriminator layers. To do this, a directory holding the centroid images that serve as extra conditions is loaded by modifying the data loader. The two input tensors from an image pair consisting of an input and a condition image are concatenated along the C dimension to create a new tensor of shape (2C, H, W). The condition image was stacked on the input image. In this instance, the boundary indicates that the input image comes first and the condition image is secondary. The concatenated tensor is fed into the network, and the added condition images are displayed during training.

Table 2: Colors used for labeling (Source: Authors)

Color	RGB	Label
	255,255,255	External area
	0,0,0	Exterior Wall
	222,222,222	Interior wall
	192,192,192	Windows
	0,255,255	Entrance
	0,255,0	Kitchen
	153,153,153	Living room
	255,255,0	Corridor
	255,0,0	Bedroom
	0,124,165	Bathroom
	255,0,255	WC

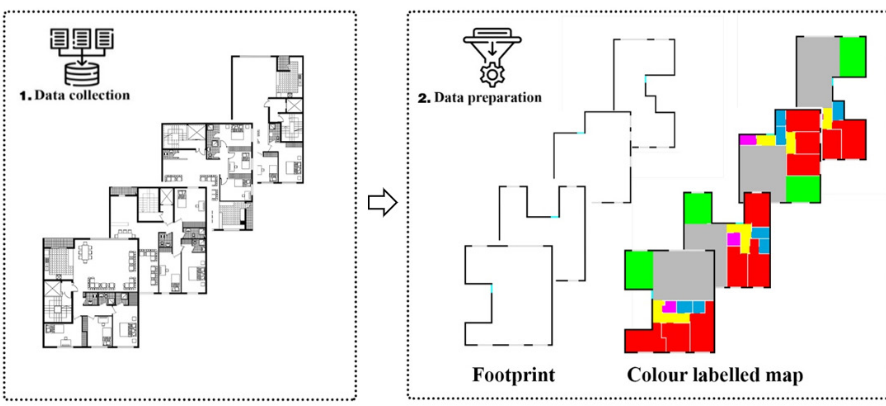


Fig. 5: Data preparation for the model training (Source:Authors)

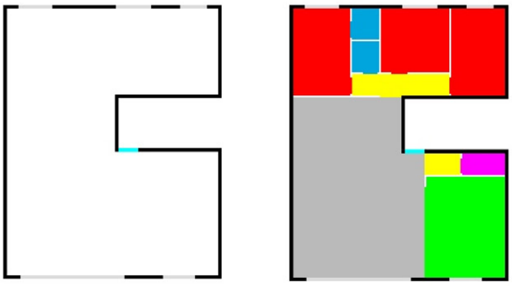


Fig. 6: Building boundary as input (Left) labeled image as ground truth (Right) (Source:Authors)

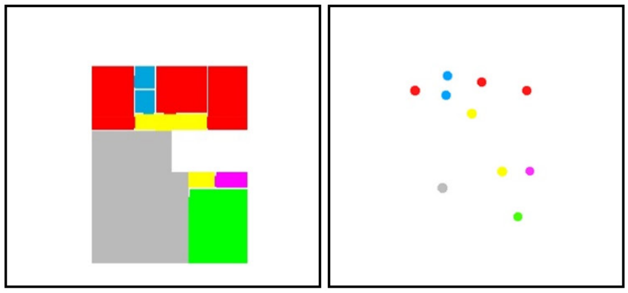


Fig. 7: Labeled image as ground truth (Left), centroid image as additional condition (Right) (Source:Authors)

a) Input boundary b) Additional condition c) Space layout plan

Fig. 8: Generating a space layout plan from boundary with additional conditions. Boundary image as input (Fig.8a) centroid image as additional condition (Fig.8b) Generated layout plan (Fig.8c) (Source:Authors)

RESULTS AND DISCUSSIONS

This section evaluates the outputs of the model. There is no established method for evaluating the performance of GAN algorithms, so Three attributes of GAN assessment criteria are explored to establish a framework for evaluating this model. First, an effective criteria rewards model can generate samples similar to real instances. Some methods use statistical calculations to determine whether the real instances and the GAN outputs are similar. Second, the output image is synthesized from a centroid image, and an input boundary must adhere to the requirements specified in the centroid image. As a result, the total number of rooms, the type of each room, and their locations should all correspond to the conditions provided by the centroids. The colors used in the output should match those in the condition images. Third, the hidden semantics in the real dataset should be represented in the generated layout plans. This relates to how rooms are arranged with respect to each other, their sizes and shapes, and where interior walls are located. The structural similarity method (SSIM) can measure the first feature. The second feature is also better performed by a computer. The third feature is better determined by human judgment because it either includes some subjectivity or is challenging for a computer to comprehend. Both quantitative and qualitative evaluations are used in this project. Quantitative evaluation includes the SSIM method to calculate the percentage of overlap between the real images and the output images. Methods such as room count deviation and room type accuracy assess how well outputs satisfy the centroid conditions. Qualitative evaluation includes a human expert assessment to examine some aspects of the image quality of the synthesized images as well as the usability of the layouts.

Quantitative evaluation

For quantitative evaluation, 66 boundary images from the test data are used. The test set also extracts the room centroid as additional conditions. As previously mentioned, the output should adhere to both the input boundaries and additional conditions. An effective criteria rewards model that can generate samples that are similar to real instances. The quantitative evaluation method SSIM utilized in this paper is the structural similarity approach. The structural similarity technique (SSIM), a widely used

Table 3: Average SSIM scores on the test set for the model (Source: Authors)

Average SSIM scores (%)
0.94%

evaluation tool, is used to calculate the percentage of overlap between the target and output images (Wang et al., 2004). SSIM relates to how effectively the human visual system functions and is perceived. The images do not overlap when the SSIM value is set to "0." However, when it is "1.00," the images completely overlap, indicating they are joined together identically (Karadag et al., 2022). In this experiment, outputs of the model and ground truth images of the test data were analyzed to examine the degree of similarity through the SSIM method. The average SSIM score on the test set for the model is 0.94% (Table 3). The results show that the proposed model has been successful in generating output. This issue indicates the flexibility and efficiency of the model for space layout generation with additional conditions. The next quantitative evaluation is related to determining how effectively the output images adhere to their centroid conditions. The output image should have the same number of rooms as the centroids in the condition image. Each centroid should also contain the room types indicated by the colors. Failure modes that could occur include assigning the wrong type to a room, not creating a room at a centroid, and generating several rooms at one centroid. Two measures are employed to identify failure modes: room count deviation and room type accuracy. Room count deviation value is the absolute value of the extra or absent room counts divided by the total number of centroids. A room is an uninterrupted area of color surrounded by interior walls. The absence of walls or openings is permissible if the room boundary is clearly defined. The synthesized image will have as many rooms as the number of centroids in the condition image if there is a 0% deviation. The average room count deviation score for synthesized samples from the test data is 1.89%. Room type accuracy is calculated by dividing the number of rooms of the correct type (accurately located) by the total number of centroids. Each centroid is examined to determine which room in the generated image contains it to obtain this value. If the room type corresponds with the type the centroid indicates, the room is considered in the right location. When a generated floor plan has an accuracy of 100%, it means that it follows the room types and approximate locations that each centroid provides. The average type accuracy score for synthesized samples from the test data is 96.89%. As it matches the conditions in most of the generated images, the scores demonstrate the ability of the model to comprehend the additional conditions correctly. The most common reason for room count deviations from the centroid images is that the model encloses parts of a room with walls.

Qualitative evaluation

This study uses qualitative evaluation based on human expert judgment. As previously mentioned, the model should capture the semantic relationships of real floor plans. For this purpose, the criteria are defined: the usability of the generated layout and the image quality. Usability relates to the dimensions and shapes of rooms, the location of interior walls, and the arrangement of rooms relative to each other. In summary, usability relates to how well a floor plan understands the semantic relationships between different types of rooms and is appropriate for its intended use as a residential building. Image quality is related to the evaluator's examination of color bleeding and blurriness. For the expert evaluation, six output images generated during the test phase were randomly selected (Table 4). Based on the criteria, an expert architect is asked to assign a score between 1 and 10 on a linear scale to each layout plan image. Table 5 summarizes the evaluation based on the six different criteria. The evaluator is asked to assess outputs from a general perspective, including examining the synthetic layout plans and analyzing the room details. For instance, layouts detrimental to usability include placing a bathroom in the center of the living room or blocking the entry door by placing the rooms. Placing rooms in inappropriate locations that would make it difficult to move between them is another example that might reduce usability. The average score of the image quality is 9.08. The usability of generated layouts is rated 9.06 on average. The outcome demonstrates that the centroids give the model enough information to allow the generator to respect the additional conditions. The model trained on centroids appears to have significantly fewer problems with color bleeding and partially rendered elements. It is assumed that the additional centroid condition guides the generator by giving it additional data, resulting in a clearer structure. Feeding additional information shows that the generator may concentrate more on refining the details than trying to generate a synthetic layout plan. The scores show that, in addition to achieving an acceptable level of image quality, the model trained on centroids also performs well in terms of usability. This paper presents a new idea for layout generation: users can control the synthesized layouts by inputting different centroid conditions.

Table 4: Outputs of the trained model (Source: Authors)

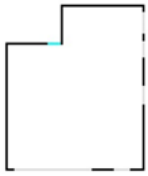
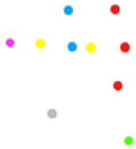
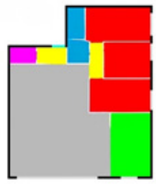
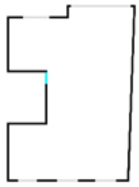
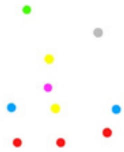
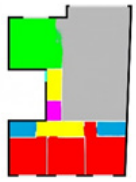
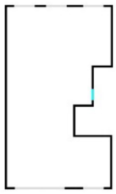
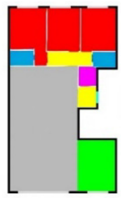
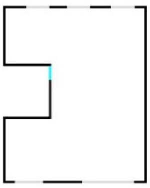
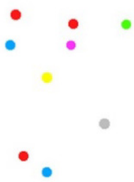
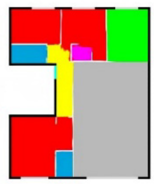
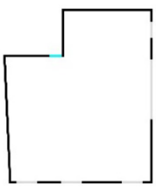
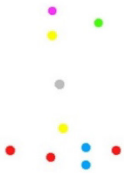
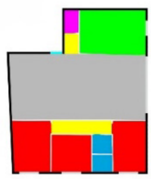
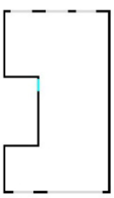
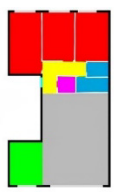
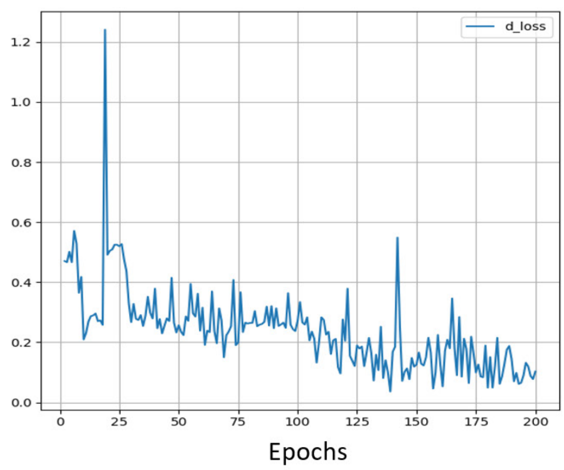
	Test data's boundary	Additional condition	Outputs of the model with additional condition	Usability score	Quality score
				8.87	9.25
				9.75	8.25
				9	9.75
				8.62	8.5
				9.12	9.25
				9	9.5
	Colors used for labeling				

Table 5: Scores of the generated layouts according to the evaluation criteria (Source: Authors)

Test data	Qualitative evaluation criteria						Usability score	Quality score
	Usability				Image Quality			
	Arrangement of rooms	Dimension of rooms	Shape of rooms	Positions of interior walls	No Color bleeding	No Blurriness		
1	8.5	8.5	9.5	9	9.5	9	8.87	9.25
2	9.5	10	10	9.5	8	8.5	9.75	8.25
3	9	9.5	9.5	8	10	9.5	9	9.75
4	8	9	9	8.5	8.5	8.5	8.62	8.5
5	8.5	9.5	9.5	9	9	9.5	9.12	9.25
6	9	8.5	9.5	9	9.5	9.5	9	9.5

Fig.9 displays the discriminator and generator losses for the models trained with centroids. It suggests that the discriminator loss is higher at the end of training. This indicates that the generated images are more realistic since they can often fool the discriminator by seeming authentic. Additionally, it was observed that the losses of the generator and discriminator are convergent towards a fixed value. In summary, the model exhibits good performance as evaluated by criteria, loss function diagrams, and findings.



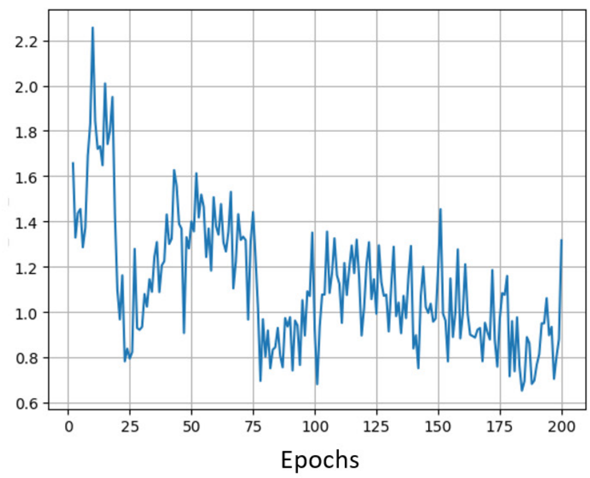


Fig.9: Discriminator loss (left), Generator loss (Right)

CONCLUSION

Big data can be processed by machine learning models, which then convert it into design knowledge that is saved in the neural network. Machine learning can quickly and efficiently provide design solutions after training the network. Machine learning networks provide greater flexibility in generation and self-update capabilities than traditional code-based solutions. This study applied the conditional GAN among the machine learning frameworks to generate automated space layouts with given boundaries and additional conditions. GAN has shown remarkable outcomes in the development of 2D designs.

In contrast to previous methods that applied GANs for generating space layouts, this model was further guided by incorporating supplementary conditions. The results indicate that the added centroid conditions defining the locations of particular types of rooms provide further guidance to the model and reduce image quality problems in the synthesized samples during testing. The additional conditions give users control over what is included in the synthetic space layout plans, and the generated plans also adhere to the boundaries of buildings. Users can generate different floor plan layouts within the same building boundaries by giving various centroid conditions. Consequently, additional conditions benefit the generation floor plan in terms of training and model flexibility. Feeding additional conditions is useful if the designer wants to interact with the model. It has been discovered that altering the centroid images can significantly impact the space layout and provide a new building layout that closely complies with the apartment layout rules. Elements in the synthesized images were found to be relatively complete through analysis, and the machine could understand the correspondence relationship between the centroid of the room and the building layout. The experiment's outcomes met our expectations based on our main question, which allows the model to interact with new conditions by providing input. When the user knows the exact location of each room, specifying the room centroids could be helpful. If the user would like suggestions from the early stages of their planning, using a model that is only trained with boundaries could be more appropriate. Higher-quality images with reduced color bleeding can also be generated. For future work, adding more specific details is suggested. For instance, displaying doors or even door swings gives additional information about the ease of movement. Another possible basis for further research could be presenting information, including desired room area and proportion. These suggested examples can be integrated into the model if the appropriate data can be gathered.

AUTHOR CONTRIBUTIONS

M. Hamouni: Literature review, conceptualization, data curation, model training, validation, preparation of main manuscript, and editing. Prof. H. Soltanzadeh, Prof. S.H. Ghoddusiar, and Prof. M. Mansorizadeh: Supervision, Project administration, Formal analysis.

ACKNOWLEDGEMENT

This article is taken from the Ph.D. thesis of the first author, Mahsa Hamouni, entitled "Investigating a Generative Model of architectural layout design based on deep neural networks," which was guided by Prof. Hossein Soltanzadeh and advised by Prof. Seyed Hadi Ghoddusiar and Prof. Muharram Mansorizadeh at the Islamic Azad University of Tehran, South Branch.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the authors have witnessed ethical issues, including plagiarism, informed consent, misconduct, data fabrication or falsification, double publication and submission, and complete redundancy.

REFERENCES

As, I., Pal, S., & Basu, P. (2018). Artificial intelligence in architecture: Generating conceptual design via deep learning. *International Journal of Architectural Computing*. https://doi.org/10.1177/1478077118800982

Caetano, I., Santos, L., & Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural R*esearch, 9(2), 287-300. DOI:10.1016/j.foar.2019.12.008.

Cardoso Llach, D. (2015). Builders of the Vision: Software and the Imagination of Design (1st ed.). Routledge. 1-199. https://www.academia.edu/14855933/Builders_of_the_Vision_Software_and_the_Imagination_of_Design

Chaillou, S. (2019). Architecture, towards a new approach, Harvard. https://doi.org/10.9783/9781949057027-006

Chang, K., Cheng, C., Luo, J., Murata, Sh., Nourbakhsh, M. & Tsuji, Y. (2021). Building-GAN: Graph-Conditioned Architectural Volumetric Design Generation. https://doi.org/10.48550/arXiv.2104.13316

Creswell, A., T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A., & Bharath. A. (2018). "Generative Adversarial Networks: An Overview." IEEE Signal Processing Magazine 35, no. 1: 53–65. https://doi.org/10.1109/MSP.2017.2765202.

Eisenstadt, V., Langenhan, C., Althoff, KD., & Dengel, A. (2020). Improved and Visually Enhanced Case-Based Retrieval of Room Configurations for Assistance in Architectural Design Education. In: Watson, I., Weber, R. (eds) Case-Based Reasoning Research and Development. ICCBR 2020. Lecture Notes in Computer Science(), vol 12311. Springer, Cham. https://doi.org/10.1007/978-3-030-58342-2_14.

Eisenstadt, V., Langenhan, C., & Althoff, K-D. (2019). Generation of Floor Plan Variations with Convolutional Neural Networks and Case-based Reasoning – An Approach for Unsupervised Adaptation of Room Configurations within a Framework for Support of Early Conceptual Design. 10.5151/proceedings-ecaadesigradi2019_648.

Ferrando, C. (2018). *Towards a Machine Learning Framework in Space Syntax* (Doctoral dissertation, Carnegie Mellon University).

https://kilthub.cmu.edu/articles/thesis/Towards_a_Machine_Learning_Framework_in_Space_Syntax/7178417?file=13205726

Goodfellow, I., Bengio, Y., & Courville, A. (2016) Deep Learning. MIT Press, Cambridge, MA. http://www.deeplearningbook.org.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative Adversarial Networks. Advances in Neural Information Processing Systems. 3. 10.1145/3422622.

Grover, R., Emmitt, S. & Copping, A. (2018). The typological learning framework: the application of structured precedent design knowledge in the architectural design studio. International Journal of Technology and Design Education. 28. 10.1007/s10798-017-9421-4.

Hong, T., Wang, Z., Luo, X., & Zhang, W. (2020). State-of-the-Art on Research and Applications of Machine Learning in the Building Life Cycle. Energy and Buildings. 212. 10.1016/j.enbuild.2020.109831.

Hu, R., Huang, Z., Tang, Y., Van Kaick, O. & Huang, H. (2020). Graph2Plan: learning floorplan generation from layout graphs. ACM Transactions on Graphics. 39. 10.1145/3386569.3392391.

Huang, J., Johanes, M., Chando Kim, F., Doumpioti, Ch. & Holz, G. (2021). On GANs, NLP and Architecture: Combining Human and Machine Intelligences for the Generation and Evaluation of Meaningful Designs, Technology|Architecture + Design, 5:2, 207-224, DOI: 10.1080/24751448.2021.1967060.

Huang, W., & Zheng, H. (2018). Architectural Drawings Recognition and Generation through Machine Learning. Doi:10.52842 /conf.acadia.2018.156.

Isola, Ph., Zhu, J-Y., Zhou, T., & Efros, A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. 5967-5976. https://doi.org/10.48550/arXiv.1611.07004

Karadag, I., Zeynel Güzelci, O., & Alaçam, S. (2022). "EDU-AI: a twofold machine learning model to support classroom layout generation," Construction Innovation. 10.1108/CI-02-2022-0034.

Lawson, B., & Dorst, K. (2009). Design Expertise (1st ed.). Routledge. https://doi.org/10.4324/9781315072043

Liu, Y., Luo, Y., Deng, Q., Zhou, X. (2021). Exploration of Campus Layout Based on Generative Adversarial Network. In: Yuan, P.F., Yao, J., Yan, C., Wang, X., Leach, N. (eds) Proceedings of the 2020 DigitalFUTURES. CDRF 2020. Springer, Singapore. https://doi.org/10.1007/978-981-33-4400-6_16.

Liu, Y., Fang, Ch., Yang, Z., Wang, X., Zhou, Z., Deng, Q. & Liang, L. (2022). Exploration on Machine Learning Layout Generation of Chinese Private Garden in Southern Yangtze. In: Yuan, P.F., Chai, H., Yan, C., Leach, N. (eds) Proceedings of the 2021 DigitalFUTURES. CDRF 2021. Springer, Singapore. https://doi.org/10.1007/978-981-16-5983-6_4.

Michalek, J. Choudhary, R., Papalambros, P. (2002). Architectural Layout Design Optimization. Engineering Optimization.34 (5):461–84. https://doi.org/10.1080/03052150214016

Mirza, M. & Osindero, S. (2014). Conditional Generative Adversarial Nets. https://doi.org/10.48550/arXiv.1411.1784

McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 115–133. http://dx.doi.org/10.1007/BF02478259.

Merrell, P., Schkufza, E., & Koltun, V. (2010). Computer-generated residential building layouts, ACM Transactions on Graphics 29. 1–12, https://doi.org/ 10.1145/1882261.1866203.

Nauata, N., Chang, K., Cheng, C., Mori, G & Furukawa, Y. (2020). House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation. 10.1007/978-3-030-58452-8_10.

Nauata, N., Hosseini, S., Chang, K., Chu, H., Cheng, C. & Furukawa, Y. (2021). House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. 13627-13636. https://doi.org/10.48550/arXiv.2103.02574

Newton, D. (2019). Generative Deep Learning in Architectural Design. Technology|Architecture + Design. 3. https://doi.org/10.1080/24751448.2019.1640536

Olgiati, V., (2013). The images of architects. Publisher: Quart Publishers. Editor: Valerio Olgiati. Pages: 424. Language: English. ISBN: 978-3-03761-069-5.

Özerol, G., & Arslan Selçuk, S. (2022). Machine learning in the discipline of architecture: A review on the research trends between 2014 and 2020. International Journal of Architectural Computing. https://doi.org/10.1177/14780771221100102

Peters, N., (2018). Enabling Alternative Architectures: Collaborative Frameworks for Participatory Design, Harvard GSD Thesis. https://books.google.com/books/about/Enabling_Alternative_Architectures.html?id=KRxnwAEACAAJ

Purcell, T., & Gero, J. (1998). Drawings and the design process. Design Studies. DOI 19. 10.1016/S0142-694X(98)00015-5.

Rahbar, M., Mahdavinejad, M., Bemanian, M., Davaie Markazi, A.H. & Hovestadt, L. (2019). Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs. Applied Artificial Intelligence. 33. 1-17. https://doi.org/10.1080/08839514.2019.1592919

Rahbar, M., Mahdavinejad, M., Markazi, A. H., & Bemanian, M. (2022). Architectural layout design through deep learning and agent-based modeling: A hybrid approach. Journal of Building Engineering, 47, 103822. https://doi.org/10.1016/j.jobe.2021.103822

Rodrigues, R. C., & Duarte, R. B. (2022). Generating floor plans with deep learning: A cross-validation assessment over different dataset sizes. *International Journal of Architectural Computing*. https://doi.org/10.1177/14780771221120842

Uzun, C., Çolakoğlu, M., Inceoğlu, A. (2020). GAN as a generative architectural plan layout tool: A case study for training DCGAN with Palladian Plans and evaluation of DCGAN outputs. A|Z ITU Journal of Faculty of Architecture. 17. 185-198. 10.5505/itujfa.2020.54037.

Wang, T.-C., Liu, Y., Zhu, J., 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, Doi: 10.1109/CVPR.2018.00917

Wang, Z., Bovik, A.C., Sheikh, H.R., & Simoncelli, E.P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. Image Processing, IEEE Transactions on. 13. 600 - 612. 10.1109/TIP.2003.819861.

Weber, R., Mueller, C.,& Reinhart, C.(2022) Automated floor plan generation in architectural design. *Automation in Construction*. 104385, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2022.104385.

Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University. https://www.researchgate.net/publication/279233597

Whitley, D. (1994). A Genetic Algorithm Tutorial. Statistics and Computing, 4, 65-85. https://doi.org/10.1007/BF00175354.

Wu, W., Xiao-Ming, Fu., Tang, R., Wang, Y., Qi, Y & Liu, L. (2019). Data-driven interior plan generation for residential buildings. ACM Transactions on Graphics. 38. https://doi.org/10.1145/3355089.3356556

Yoon, S-w., Kim, I. & Lee, K-C. (2018). The Architectural Design of Storage System for Power Data Management. 736-738. 10.1109/BigComp.2018.00139

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. J. Shanghai Jiaotong Univ. (Sci.) 26, 103–115. https://doi.org/10.1007/s12204-021-2265-9

Zheng, H., & Yuan, P. F. (2021). A generative architectural and urban design method through artificial neural networks. *Building and Environment,* 205, 108178. https://doi.org/10.1016/j.buildenv.2021.108178.

Zheng, H., An, K., Wei, J., & Ren, Y. (2020). Apartment Floor Plans Generation via Generative Adversarial Networks. In Proceedings of the RE: Anthropocene, Design in the Age of Humans: Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2020), Chulalongkorn University, Bangkok, Thailand. 10.52842/conf.caadria.2020.2.599.