Ontology Space International Journal
ISSN: 2345-6450
OISSN: 2423-7299
Spring 2024, Vol. 13, Issue 2, No. 49, Pages: 41- 53
Journal homepage: soij.qiau.ac.ir

Spatial Layout Design in Architecture Through Deep Neural Networks

Mahsa Hamouni [a], Hossein Soltanzadeh [b,*], Seyed Hadi Ghoddusifar [c], Muharram Mansoorizadeh [d]

[a] *Department of Architecture, Faculty of Art & Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran.*

[b] *Department of Architecture, Faculty of Architecture and Urban Planning, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

[c] *Department of Architecture, Faculty of Art and Architecture, South Tehran Branch, Islamic Azad University, Tehran, Iran.*

[d] *Department of Computer Science, Faculty of Engineering, Bu-Ali Sina University, Hamadan, Iran.*

Received: 01 June 2023.- Accepted: 29 August 2024

Abstract

The architectural layout design is a well-known algorithmic problem in computer-aided architectural design. It is the assignment of discrete space elements to their corresponding locations while attempting to satisfy geometrical and topological goals in their layout. This task requires maintaining consistency to ensure that the requirements are met and is exploratory and iterative in nature. The complexity of this problem has encouraged researchers to explore computational approaches for predicting the space layouts. This activity, takes place during the preliminary design phase and is highly significant, as it impacts later stages of the building lifecycle. Therefore, these methods has the potential to enhance the effectiveness and efficiency of spatial layout in design and suggested to make the work of architects easier and faster. Numerous methods have been proposed to solve the space layout design problem. Each one can be viewed as a rule-based strategy that attempts to simulate space layout design using some high-level rules. However, for producing space layout designs, in addition to the quantitative criteria that may be tested and assessed in a logical process, numerous non-quantitative elements also exist. These qualitative criteria are frequently based on a variety of factors and are challenging to describe; thus, hard-coding them would not be possible or effective. It would be best if the program could learn these rules from existing examples. Some potential solutions can be found in the rapidly expanding field of machine learning, which can serve as a tool for decision-making. Deep learning, a subfield of machine learning, can adaptively achieve goals by learning from data and interpreting experiences. The generative adversarial network (GAN), is a deep learning algorithm that has shown remarkable outcomes in the development of 2D designs. In this paper, GAN is applied to generate automated space layouts with given boundaries. A specialized training dataset, comprising 660 existing apartment layouts from Hamadan, is prepared, with each layout labelled using different colours to represent various spaces for training the model. After the model is trained, the boundary lines of 12 new apartments are tested. The performance of the model is also evaluated using two methods: the pixel accuracy measure as the quantitative method and a qualitative assessment by an expert architect based on the evaluation criteria. The results show that the proposed model successfully generates space layout plans from predefined boundaries. This issue indicates its potential for application in other cases and designs. We propose this model as a tool to facilitate the architectural layout design process, enabling architects to quickly and precisely meet client requests particularly in the projects with complex topological constraints.

Keywords: Spatial layout design; Generative design; Deep learning; GAN

1. Introduction

New computer-based methods are suggested in the expansive research area of computer-aided architectural design to make the work of architects easier. Since the 1960s, the space allocation problem has been referred to as the space layout design problem in the field of computer-aided architectural design. One of the most challenging tasks in architectural design is space layout design, which takes place in the early stages of the design process (Latha et al., 2022). It is the process of placing a set of discrete but independent spatial elements while striving to achieve geometrical, topological, and performance goals in their layout. Due to the fact that it has an impact on the next phases of the building lifecycle, it is particularly significant (Regateiro et al., 2012). This activity is exploratory and iterative in character, requiring constant maintenance to assure that various requirements are fulfilled. Therefore, the automated design technique is receiving a lot of attention because it may be a way for researchers to quickly plan complex architecture programs or find solutions for spatial design. The goal of implementing such a technique does not entail the replacement of architects, but rather the development of strong tools to quickly solve problems, and select the best tool for advancing design (Guo & Li, 2017). Topological and geometrical constraints are the resultant of subjective and objective criteria that is embedded in the spatial layouts. Objective criteria are quantitative factors that could be tested and assessed in a logical and numerical process such as architectural program, the energy efficiency, regulations, design standards, client preferences and others. The subjective criteria, however, are non-quantitative factors that related to the expertise of the designer such as the design's aesthetic, cultural, economic and social aspects. Many strategies have been proposed to solve the space layout design problem. Most of these approaches can be viewed as a rule-based strategy that makes an effort to simulate space layout design using some high-level rules (Rahbar et al., 2022). However, in addition

* Corresponding Author Email: 72soltanzadeh@gmail.com

to the quantitative criteria that may be tested and assessed in a logical process, numerous non-quantitative elements are existing for producing spatial layout designs. Although there are tools to assist architects in creating floor plans, none actively contribute to the intelligent design of the plan. They would need to follow both the explicit and implicit rules of floor plan design, which would take a great deal of intelligence. From the perspective of an architect, a successful layout is evaluated as a possible solution, that takes into account both subjective and objective factors. Due to the fact that these rules and considerations are frequently based on a variety of criteria and are challenging to specify, hard coding them would be next to impossible. Furthermore, even if it were possible, this task would take a lot of time and be highly subjective. It would be preferable if the program could learn these rules from existing examples (build precedent) rather than hard-coding them into the program. Precedents represent knowledge in design in a holistic way which provides some assurance of the success of a design as heuristic devices. This falls into the artificial intelligence (AI) problem space. The rapidly expanding filed of machine learning (ML) may offer some viable solutions (Russell & Norvig, 2016). ML is the science of programming created for computers to learn from data (Geron, 2017). It is the field of study that gives computers the ability to learn without being explicitly programmed (Ozerol & Arslan Selçuk, 2023). Deep learning is a machine learning technique that layers computing units- or neurons- into what is called an artificial neural network. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Where machine learning algorithms generally need human correction when they get something wrong, deep learning algorithms can improve their outcomes through repetition (Topuz & Çakici Alp, 2023). The automation of jobs involving the recognition of qualitative patterns in data has undergone a revolutionary change with the advent of deep neural networks. It is distinct from more traditional design computing methods. One significant distinction is that deep neural networks learn patterns through the training process rather than being structured on rules or previous algorithms. This is in contrast to earlier generative design methods. For instance, design rules that specify which forms can be created by their repeated application can be used to create new designs in shape grammar. Deep learning, however, uses generative design, where the rules for how to combine them are not predetermined. Instead, they are found by the DNN using design data. A DNN should theoretically be able to learn latent or known rules that the architects may have purposefully or intuitively implemented in their work if one has enough examples from a design collection (Imdat et al., 2018). On classification tasks, these deep discriminative models have performed so well that they have even been able to outperform human experts. Deep neural networks can do generative tasks, in which new data patterns can be created, in addition to discriminative tasks. Although they have received less studied, these "deep generative models" are becoming a significant field of study. The (GAN) is one deep generative model that has demonstrated excellent outcomes in generation designs. By giving training data in pairs, Goodfellow is credited with being the first team to propose the generative adversarial network in machine learning (Goodfellow et al., 2014).According to Goodfellow, the fundamental challenge for AI is being able to complete activities that people can accomplish with ease but formally find challenging. Such issues are resolved intuitively by people. Formally describing the process of designing in architecture is also quite challenging. An architect is capable of drawing a building plan without considering any cognitive components of the design. However, a highly intricate and complex process is unfolding in the mind behind the drawing action. To effectively train an algorithm, it would not be possible to describe these design processes. Instead of attempting to formally explain the design for autonomous design processes, an algorithm can infer meaningful solutions through drawing datasets. Today with the concept of big data, algorithms can evaluate data, and deep learning algorithms can be trained without the need to formalize processes (Goodfellow et al., 2016). GANs have shown a level of generative sophistication that has not been matched by earlier work in computer-generated design. Numerous design-related disciplines could be impacted by their capacity to learn from examples and extrapolate that learning into the creation of new instances. The generation of 2D images of plans and facades has investigated their use in architecture (Newton, 2019). In this paper, a method of using generative adversarial networks was applied to generate automated space layouts. It focuses on generating automated 2D layouts with a set footprint, windows, and an entrance door that are inputs to the suggested algorithm from the designer. To achieve this, we take the following actions: creating a specific dataset for the model, using the dataset to train the model, and then assessing the model using the test findings of the new footprint.

2. Research Background

2.1 Rule-based methods

Studies on the autonomous production of plans in architectural design started in the 1960s. The problem of space layout design has been considered in a variety of ways. Liggett and Mitchell described a software system that uses quadratic assignment, which is focused on finding the best places for a group of related objects, to generate high-quality solutions to both small and large space planning challenges concerned with operating efficiency (Liggett & Mitchell, 1981). In order to solve the challenge of office layout planning, Jagielski and Gero developed a genetic programming approach, a subset of the evolutionary (Jagielski & Gero, 1997). Jo and Gero suggested a design approach based on an evolutionary/genetic algorithm for developing large office layouts (Jo & Gero, 1998). The use of a genetic engineering-based extension to genetic algorithms for space layout design challenges was described by Gero and Kazakov (Gero & Kazakov, 1997). With a prototypical example in the context of the 2D spatial design of homes, Rosenman presented a case-based model of design that

uses an evolutionary approach for the adaptation of previously stored design solutions (Rosenman, 1999). A mathematical programming approach was used by Medjdoub and Yannou to create an architectural conceptual CAD that, crucially, avoids the inherent combinatorial complexity of practical space layout issues (Medjdoub & Yannou, 2000). Michalek et al provided a model that provides a method for optimizing the arrangement of architectural floor plans that benefits from the effectiveness of gradient-based algorithms and uses evolutionary algorithms to make discrete decisions (Michalek et al, 2002). By including the physics of motion into the space plan elements, Arvin and House employed physically-based space planning to automate the conceptual design process. This methodology offers a responsive design process, enabling a designer to make choices that have instantaneous effects on the entire design (Arvin & House, 2002). An issue with architectural layout that uses evolutionary algorithms to decide how to arrange units was taken into consideration by Bausys and Pankrasovaite (Bausys & Pankrasovaite, 2005). With the help of the layout evolutionary multi-objective optimization and interactive evolutionary computation, Inoue and Takagi developed a framework for a room layout planning support system (Inoue & Takagi, 2008). Merrell et al described a technique that uses a Bayesian network trained on real world data to automatically generate building layouts (Merrell et al., 2010). Based on a squarified treemap algorithm, Marson and Musse presented a revolutionary method for creating house floor designs with semantic information (Marson & Musse, 2010). Thakur et al demonstrated a technique that uses evolutionary algorithms to create an architectural layout for a single flat with consistently shaped spaces (Thakur et al., 2010). Koenig and Schneider concentrated on interactive evolutionary algorithms and computer-based generative solutions for architectural layout issues (Koenig & Schneider, 2012). Liu et al suggested a method for automatically generating various floor plans with corresponding 3D geometry that all adhere to the design requirements and constraints (Liu et al., 2013). Bao et al used simulated annealing to create candidate layouts in order to tackle the issue of generating and investigating building layouts that satisfy specific requirements. By employing quadratic programming to remedy broken hard constraints, they restore the validity of the arrangement (Bao et al., 2013). Rodrigues et al demonstrated a prototype tool for the space planning stage that uses evolutionary algorithms to automatically develop various floor plans in accordance with the preferences and needs of the architect (Rodrigues et al., 2014). Helme and Derix provided an overview of a number of experimental hybrid semi-automated techniques that have been used in design processes where users can direct physics-based simulations to provide input for programmatic distribution of layouts (Helme & Derix, 2014). Utilizing a simulated annealing algorithm, Yi et al proposed a design decision-making methodology for configuration based on optimal environmental performance (Yi et al., 2014). Two evolutionary algorithm-based techniques for creating rectangular architectural layouts were presented by Koenig and Knecht (Koenig & Knecht, 2014). By combining a rectangular Voronoi subdivision with a genetic algorithm, Chatzikonstantinou introduced a new method for creating architectural layout configurations that is appropriate for use in computational optimization scenarios (Chatzikonstantinou, 2014). Dino demonstrated a design tool for evolutionary algorithm-based space layout optimization (Dino, 2016). Hua demonstrated a technique for creating irregular floor plans automatically. It creates topologically viable layouts by using simulated annealing and sub-graph matching. (Hua, 2016). Guo and Li used a technique for automatically creating a spatial architectural plan using a multi-agent topology finding system (Guo & Li, 2017). Using an interactive evolutionary algorithm, Bahrehmand et al introduced an interactive layout solver that aids designers in layout planning by offering personalized space configurations based on architectural guidelines and user preferences (Bahrehmand et al., 2017). Bisht described a technique that makes use of graph theory to convert an adjacency graph into a dimensioned floor design (Bisht, 2022). In Table 1, an overview of these methods (Weber et al., 2022) is presented.

2.2 Machine learning methods

With regard to the methods described above, it can be deduced that each one can be viewed as a rule-based strategy that makes an effort to simulate space layout design using a few high-level rules. In contrast to previously described approaches, Machine learning models have shown that they can derive information from data and use that knowledge to make decisions (Creswell, 2018). On the other hand, there is a long history of architectural education and practice that emphasizes learning from precedent. Built precedent can be a useful tool for decision-making in the architectural design process. They are utilized as a reference for developing space layouts in both professional and educational settings (Grover et al., 2018). There has been a resurgence of interest in referential automated layout techniques as a result of advancements in computing and ML technology. This is a technique to imitate how architects make decisions based on their expertise and experience. There are a number of algorithmic techniques for referential design, but the most popular are machine learning algorithms that employ deep neural networks, such as GANs, to discover closest matches (Weber et al., 2022). In order to recognize spaces in a layout and to create a furnished plan from a labelled image, Huang and Zheng used the Conditional generative adversarial network (c-Gan) algorithm for both tasks. They showed how GAN might be used to generate furniture layouts and recognise floor plans. Based on the room program and its opening location, their model would design the infill of rooms using colored patches (Huang & Zheng, 2018). In the same year, Nathan Peters advocated using GANs to address program repartition in a single-family modular homes based on the house footprint in his Harvard thesis (Peters, 2017). Rahbar translated the image of an apartment footprint directly into a labelled layout design using a conditional GAN model (Rahbar et al., 2019). Three key steps of the pix2pix algorithm were used by

Chaillou. A parcel outline was used to generate a building footprint in the initial step. In the second stage (program repartition), each area inside the footprint was assigned a color. The furnished layout was generated at the third stage (Chaillou, 2019). In Table 2, an overview of these methods is presented.

Table 1
Automatic space layout creation methods.

Author(s)	Typology	Optimizer	Inputs	Output
Gero	Residential	Genetic Algorithm	Number and areas of rooms	Floorplan, based on grid
Rosenman et al	Residential	Genetic Algorithm	Tree representation of program	Floorplan, differentiated rooms connected
Arvin and House	Public	Physically Based	Area, adjacency	Modeling architectural design objectives in physically based space planning
Inoue and Takagi	Residential	Evolutionary algorithm	Area, location preference	Assigned program on existing layout, differentiated boundaries
Merrell et al	Residential	1. Baysian network for Program generation, 2. Metropolis algorithm	Area, foot print, aspect ratio, adjacency, adjacency type	Program layout
Bao, et al	Residential, Office	Quadratic Programming, simulated annealing	Boundary, total floor area, courtyards	Massing with specified floor area
Yi et al	Residential	Simulated annealing	Programmatic units	Aggregation of modular
Chatzikonstantinou	Residential	Rectangular Voronoi Subdivision, Genetic Algorithm	Area, weighted adjacency matrix	Volumetric Arrangement of layout
Hua	Residential	Mathematical Programming	raster image or vector graphic, area, adjacency	Layout on input image or vector graphic.
Medjdoub and Yannou	Residential	Mathematical Programming	Adjacency, area, min width/depth,	Assigned program on existing layout
Michalek et al	Residential	Genetic Algorithm, Mathematical Programming	Areas, adjacency	Design topology (with adjacencies) (tree) and assigned program existing layout
Bausys and Pankrasovaite	Residential	Genetic Algorithm	Program description (with min and max size), bounding box	Layout in bounding box
Marson and Musse	Residential	Squarified Tree map KD Tree	Areas, connectivity	Assigned program on existing layout
Liggett and Mitchell	Office	Quadratic assignment	Areas, adjacency	Assigned program on existing layout
Jo & Gero	Office	Genetic Algorithm	Areas, adjacency	Assigned program on existing layout
Gero & Kazakov	Office	Genetic Algorithm	Areas, adjacency	Assigned program on existing layout
Jagielski and Gero	Office	Genetic Algorithm	Areas, adjacency	Assigned program on existing layout
Thakur et al	Residential	Genetic Algorithm	Area	Assigned program on existing layout
Guo and Li	Residential	Agent based	Program graph, area of rooms	Generated layout assigned to Grid voxel
Koenig & Schneider	Residential	Genetic Algorithm	Connectivity, hierarchy	Assigned program on existing layout
Liu et al	Residential	Non-linear least squares	Connectivity, Areas, Wall fabrication specification	Rooms inside boundary, precast concrete walls
Koenig and Knecht	Residential	Genetic Algorithm	Connectivity, hierarchy	Assigned program on existing layout
Rodrigues	Residential	Mathematical Programming	Areas, connectivity	Assigned program on existing layout
Dino	Office	Genetic Algorithm	Program description, geometric properties	Room tiles in existing grid
Bisht	Residential	Graph theory	Dimensional constraints, adjacency	Program layout
Bahrehmand et al	Residential	Genetic Algorithm	Areas, adjacency, window door or entrance.	Program layout

Table 2
Machine learning methods in the literature.

Author(s)	Typology	Matching	Inputs	Output
Huang & Zheng	Residential	c-GAN	Labelled image of floorplans	Furnished floorplan
Peters	Residential	c-GAN	Boundary	Rooms color coded in boundary, manual tracing for vectors
Chaillou	Residential	c-GAN	Boundary of building	Rooms color coded in boundary manual tracing for vectors
Rahbar et al	Residential	c-GAN	Boundary of building	Rooms color coded in boundary

3. Research Methodology

GAN has recently emerged as the best architecture for the synthesis of images, on the basis of the dataset used to train the GAN model. Image-to-image translation is one of the uses for generative adversarial networks. The goal of image-to-image translation is to create a new image that is connected to the mapping relationship between the input and output images using a neural network model. Deep generative models have shown superior performance over competing methods by learning from and interpreting data to synthesis designs with a level of sophistication and adaptability. In this study, a conditional generative adversarial network named Pix2Pix is used to support a model for residential layout generation. In Figure 1 shows the research methodology flowchart. This paper's primary objective is to suggest an approach for predicting the possibility of a space layout using data-driven processes. The procedure and the representation's outcomes could be used for the new field of architectural probability problems. Figure 2 illustrates the research procedure diagram

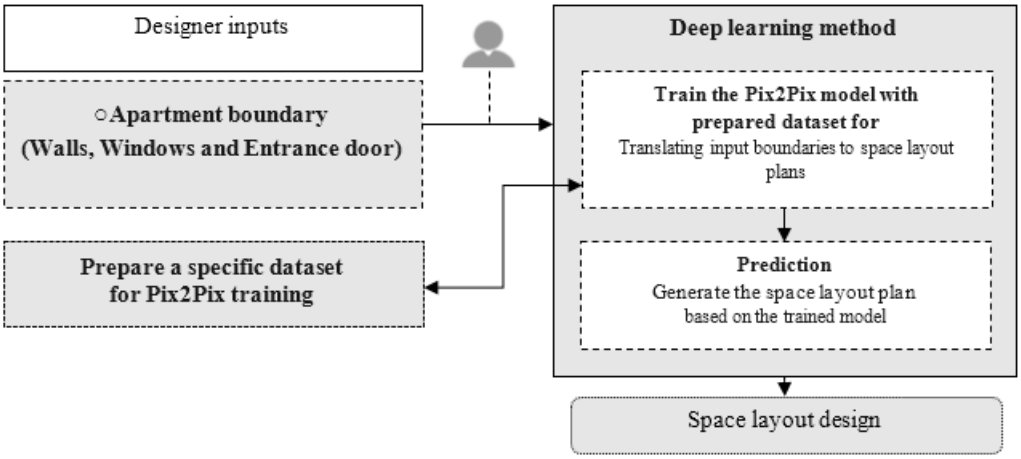


Fig. 1. Proposed automated design methodology.

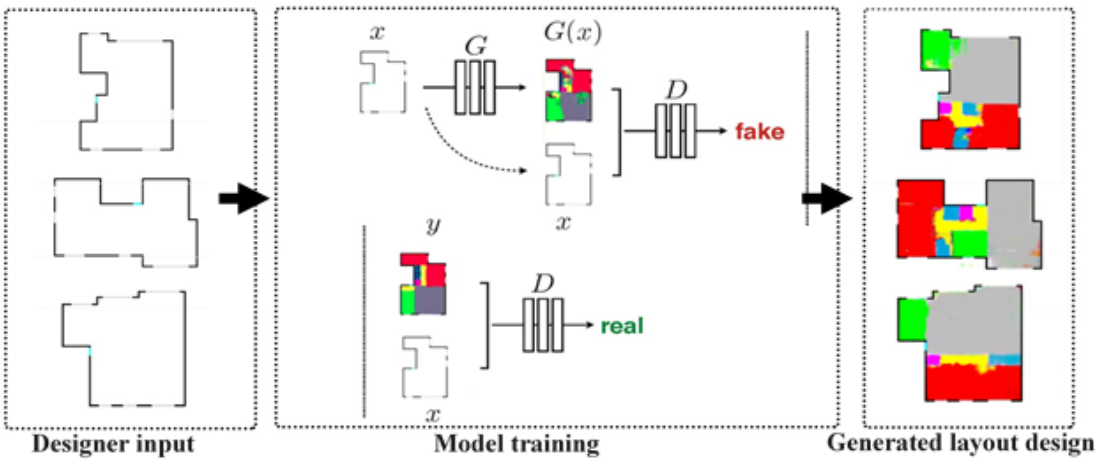


Fig. 2. Research methodology procedure.

3.1 Pix2Pix

Pix2Pix is a form of Conditional generative adversarial network for general purpose image-to-image translation.
By providing training data in pairs, Ian Goodfellow and his colleague proposed the generative adversarial networks (Goodfellow et al., 2014). The GAN model is founded on the idea that a generator and a discriminator, two neural networks, compete with one another. A generator model named G is for creating new logical synthetic images and a discriminator model named D is for classifying images as real (from the data set) or fake (generated). A set of data is used to train the D model to recognize images. When properly trained, this model can distinguish the difference between a real example taken from the dataset and a fake image that isn't from the dataset. The G model, however, is trained to generate images that look like images from the same 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 producing images that are even more realistic. The G model tries to deceive the D model while the D model tries to recognize the fake images. The two networks are trained concurrently in an adversarial phase. A GAN gradually improves its ability to create relevant synthetic images through this feedback loop (Chaillou, 2019). GANs are generative models that learn the mapping G: z → y from a random noise vector z to an output image y (Goodfellow et al., 2014). The Conditional GAN is an extension of the GAN architecture that gives users control over the output generated image by basing it on an input source image. As opposed to this, Conditional GANs learn a mapping from observed image x and random noise vector z, to y, G: {x; z} → y (Figure 3).

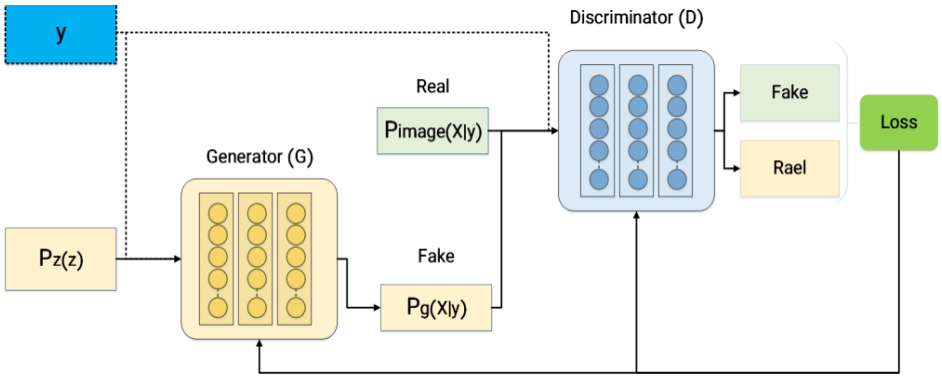


Fig. 3. The architecture of Conditional GAN.

An adversarially trained discriminator, D, which is trained to do as well as feasible at spotting the generator's fakes, cannot separate the outputs from the real images that the generator, G, produces. The objective of a conditional GAN is made up of two part: L1 loss and adversarial loss. The adversarial loss can be expressed as:

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

where an adversarial D seeks to maximize this aim while G seeks to minimize it. Conditional GANs learn a conditional generative model in the same way that GANs learn a generative model of the data. This makes it possible to conditionally generate an output image that matches an input image for image-to-image translation tasks.
It has been useful in the previous approaches to mix the GAN objective with a more traditional loss, like L2 distance. The generator's task is to not only deceive the discriminator but also to be close to the ground truth output in an L2 sense, while the discriminator's job stays unchanged. Phillip Isola and his colleague explore investigate this possibility, using L1 distance rather than L2 as L1 encourages less blurring. L1 distance is added to the generator loss to encourage the low- frequency correctness of the generated image:

$$L_{L1}(G) = E_{x,y,z}[\| y - G(x,z) \|_1]$$

And the final objective function is as follows:

$$G = \arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G)$$

Pix2Pix was presented by Phillip Isola in 2017. The generation of an image is dependent on a given image in this application of conditional GANs to the problem of image-to-image translation. To map the input to the corresponding output set y, the input x sets are loaded into the trained Pix2Pix models. It developed into a viable foundation for complex image-to-image translation tasks, such as translating maps to satellite photos, black-and-white photos to color, product drawings to product photos, sketches to photos, labels to scenes, day to night and more. A translation of labels to scenes is shown in Figure 4. In the case of Pix2Pix, the discriminator is given a source image and a target image, and it then determines if the target is an acceptable translation of the source image or not (Isola, 2017).

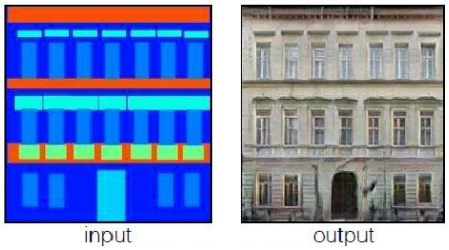


Fig. 4. Translating of label to scene example.

3.2 Preparation of the data set

Regarding the procedure of Pix2Pix training, a pair of same-sized images are required as training dataset. In the scope of the study, the data set is prepared to train Pix2pix model. . This training dataset includes 660 ideal apartment plans of Hamadan. The similar social, economic, and cultural conditions were effective in the selection of plans. So the plans of the District 1 and District 2 of Hamadan were collected. The training dataset contains a pair of images, the first is the boundary representation of the project as an input and the second one is the available layout design as a conditional target image. First, the real cases of apartment floor plans are collected. Second, the collected floor plans are labeled manually using the different colors to represent spaces, i.e. color labeled map. AutoCAD software is used to generate the source and target image representations based on actual sizes, each layout is precisely drawn. As the vector drawing is transformed into a raster image, the drawings are proportionally resized. A sample of preparing source image (input) from an apartment floor plan is illustrated in Figure 5. An example apartment plan is located on the left. For Pix2Pix's input picture, the right image is created. It only has the apartment's boundary wall. There are no differences in the thickness of the boundary wall for each dataset. The skylights in the plan of northern apartments, columns and staircases and are not included. Terraces are regarded as an extension of adjacent areas. Walls, windows and entrance of footprint are the input image's low features. For the output picture, the appropriate image (a colored layout as target) is produced. Each space is labeled with a specific color. The colors used to label the spaces are shown in Figure 6. The low-level features used to train the network are the RGB colors. The network learns how to map the input images to their matching output images during the Pix2Pix training process. Figure 7 displays a portion of the 660 collection. All of the plans in the dataset are normalized based on their area in order to define the size of each apartment. The Pix2pix model is trained on all of the 660 dataset which is described in the next section.

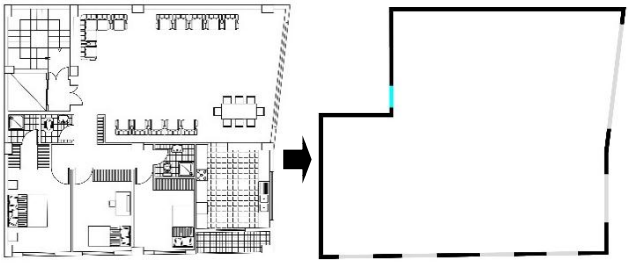


Figure 5. Preparation of the source image (footprint) from the original plan.

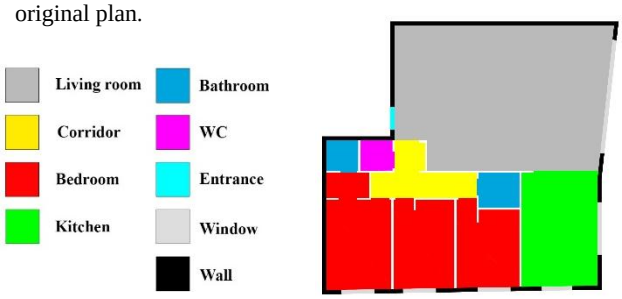


Fig. 6. Labeling the floor plan for Pix2Pix training, labeled output image (Right) and Label colors of spaces (Left).

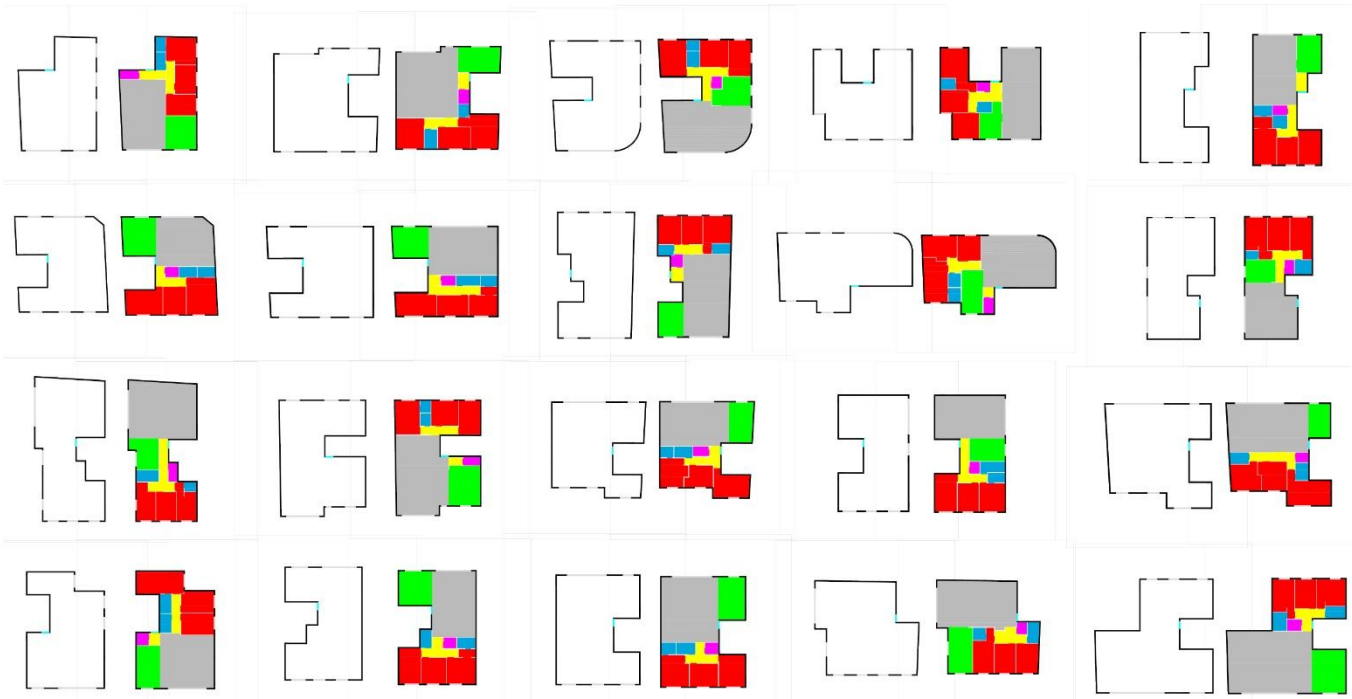


Fig. 7. A part of the training dataset.

3.3 Training the Pix2Pix

Following the data preparation phase, the data set is used to train a Pix2Pix GAN model in Keras. The open-source software library known as Keras offers an interface for TensorFlow, an open-source framework for numerical

computation and large-scale machine learning used for artificial neural networks. Each image is imported, resized, and then divided into "footprint plan" and "color labeled plan" sections. The output consists of images with a width and height of 256x256 pixel. Figure 7 illustrates the generator and discriminator parts that made up the model's architecture. In order to account for mapping of footprint to color labeled plan pictures over the whole training data, the discriminator and generator models' parameters are changed during the Pix2Pix training process. This model's discriminator employs a "PatchGAN" classifier from a convolutional neural network. Instead of using the entire area for classification of images, the PatchGAN uses patches of a particular size. This trains the generator to produce more realistic images. Pix2Pix uses a "U-Net-based" architecture as its image-to-image translation. The discriminator gains the ability to classify a pair of images as real or fake. While this is happening, the generator learns how to deceive the discriminator into the generated image is a real image. The U-Net generator is considered an encoder-decoder model. It is a structure that uses a skip connection to link the encoder layer and the decoder layer directly. In comparison to a simple encoder-decoder architecture, the skip connection enables learning that is more stable. The model generates a space layout plan from a footprint. It employs a convolutional and de-convolutional layer with the option of layer skipping. Each source image that is input is divided into a number of tokens by the generator. The generator performs the encoding-decoding process by first down sampling or encoding the input image to a bottleneck layer and then up sampling or decoding the bottleneck representation to the output image's size. In this procedure, a new image is produced using the experience it has gained from the training dataset (48). The generated image and the input are inserted as a pair into the discriminator model in the next phase. The discriminator model simply checks and penalizes a small patch of the image pair to determine whether the provided images are fake or real. It takes data from both the training set and the generator. To train GAN models, many thousands of iterations are required.

4. Results and Evaluation

In this section, we evaluate the outputs of the model. There is no established method for evaluating the performance of GAN algorithms; nevertheless, there are numerous alternative methods that can be employed. The two categories of GAN evaluation methodologies are qualitative and quantitative approaches. In this paper, we applied both approaches to comprehend GAN behavior. Manually inspecting and judging the generated samples (rating and preference judgment) is one of the most common and effective qualitative assessments for evaluating GANs. An expert architect does the qualitative evaluation of the model in this paper. In addition to qualitative evaluation, the effectiveness of GAN algorithm productions can be measured using a variety of quantitative techniques. Some of these methods assess the resolution quality of the GAN-generated image, while others use statistical calculations to determine whether the training dataset and the GAN outputs are similar (Borji, 2019). In this study the pixel accuracy measure is proposed for the evaluation of the trained model. Pixel accuracy is calculated by dividing the number of correctly mapped pixels from an input image of a particular class in the synthesized image, by all pixels in that class.

4.1 Evaluation through the quantitative method

After the training process, we use the pixel accuracy measure to evaluate the model's outputs. When external areas, exterior walls, windows and the entry door pixels retaining their labels and interior areas are mapped to the color other than the values used for other classes, this is considered a correct mapping. Since the boundary where the classes meet is only a smaller piece of an image, errors are likely to occur along this area. The accuracy scores are calculated separately for five classes: External area, exterior walls, windows, entry door and interior area. The use of correct colors in the right locations relates to the label pixel accuracy score. We need to define a tolerance level that allows for some deviations from the real values since the model could generate pixels that resemble the colors used in the ground truth images but do not have exactly the same RGB values. As shown by the label pixel accuracy values in Table 3, external areas and interior areas being correctly rendered. The boundary is not shifted either as that would affect the accuracy scores for the external areas. It seems like the exterior walls, windows and the entry door pixels are to some extent overtaken by the bordering other classes, since visual inspection of the synthesized images does not show any indications of the model using the wrong colors. The entry door pixels seem to suffer from surrounding classes bleeding into them the most as the accuracy scores are considerably lower than the scores for the other classes. The entry door also occupies a smaller area of the image, so any pixel deviations will affect the accuracy percentages more.

Table 3
Average label pixel accuracy scores.

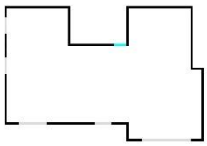
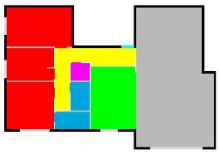
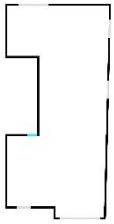
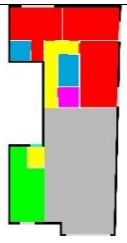
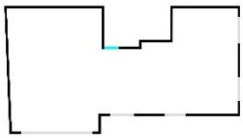
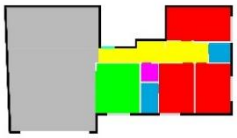
Average label pixel accuracy (%)				
External area	Exterior walls	Windows	Entry door	Interior area
100	88.4	76.8	57.7	100

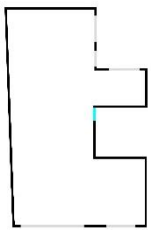
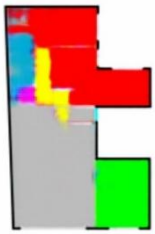
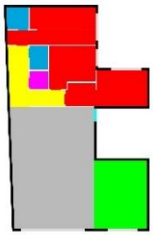
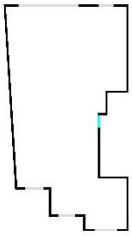
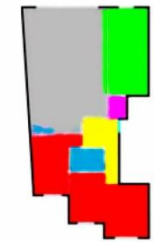
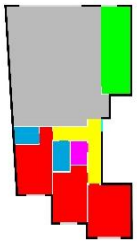
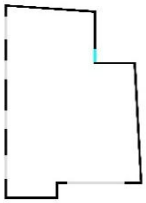
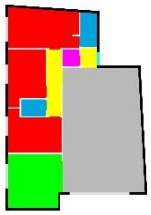
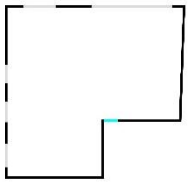
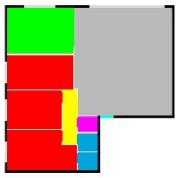
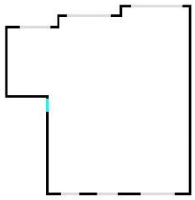
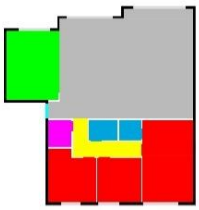
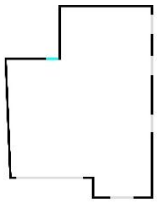
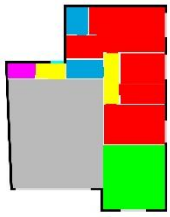
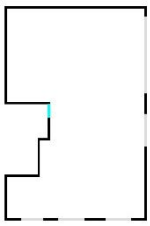
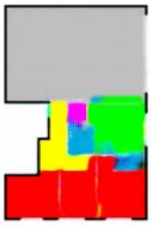
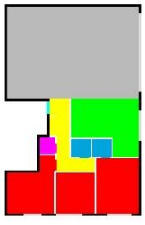
4.2. Evaluation through the qualitative method

The qualitative evaluation of the model is done by an expert architect after the training phase. To this end, 12 different types of the test data's boundary lines with different dimensions and designs are given to the model as input datasets. These boundary lines were not present in the training data set before. Table 4 shows the output generated by the model based on its learning process. The experimental data including boundary lines are shown in the first column. The second column displays the model's output. The third column shows a sample layout which designed by an expert architect. There is also a column of average score resulting from the evaluation of the model by

an expert architect. The scores range from 1 to 7 for each criterion with numbers from 1= extremely weak to 7= extremely strong. In Table 5 the scores of each criterion for the 12 test data is presented. Since there are always multiple solutions for each case, the only significant criterion for the evaluation of the model is the learning of the rules and latent basic concepts for production of an architectural spatial layout. For this purpose, several evaluation criteria in two categories, namely topological and geometrical, are used to evaluate the model's ability by an expert architect. The geometrical criteria included the correct dimensions of the spaces and the correct proportions of the spaces in relation to the total area. The topological criteria included the correct orientation of spaces, the correct zoning relative to the entrance, the correct arrangement of the spaces relative to each other and relative to the location of the windows and openings. The dimensions criterion evaluated the correct dimensions of the spaces. The proportions criterion, evaluated the area of each space in relation to the total area. The orientation criteria, evaluated the correct orientation of each spaces. The entrances criterion evaluated the correct division of the spaces into two public and private zones according to the location of the entrance. The arrangements criterion evaluated the correct location of each space in relation to windows, openings and other spaces. In sample 1, there are no elongated and disproportionate rooms, and the model preserved the dimensions of each layout. In sample 2, however, all spaces, including the WC, the bathroom, the living room, the bedrooms and the kitchen, are proportionate to the total area. In sample 4 the area of the WC and that of the bathroom are not compatible with desirable spatial proportions either. In sample 5, the dimension of master's bathroom shows that the model has not correctly learned the dimensions. In real life cases, the location of an entrance is such that it connects a public space to a private one. As can be seen, in all samples, the arrangement relative to the entrance has completely separated the public and private spaces. All samples have windows in the living area, kitchen, and bedrooms. That shows that the model has accurately figured out how to arrange the spaces in relation to the windows. A further indication that the model has successfully learned the proper arrangement of the areas relative to one another is the lack of a bathroom or WC in the middle of the kitchen, living room, or bedrooms. The scores of the evaluation criteria and the resulting output showed that some of the desirable drawing and layout design patterns have been appropriately learned by the model. The correct zoning relative to the location of the entrance, the correct arrangement of the spaces that require light adjacent to windows and the correct arrangement of the spaces relative to each other and the correct orientation of spaces are among the well-learned instances. This study showed that the production of a spatial arrangement layout by the model was successful. Simultaneous consideration of and attention to all impactful factors in the design process, especially the latent design rules and patterns pertaining to the experiences of the designer, is one of the benefits of utilizing generative adversarial networks in solving design problems.

Table 4
Outputs of the trained Pix2Pix.

	Test data's boundary line	Output of the Model	Sample layout design	Average Score
1				**6.83**
2				**6.66**
3				**6.91**

4				**6.58**
5				**6.75**
6				**7**
7				**6.91**
8				**6.66**
9				**6.75**
10				**6.83**

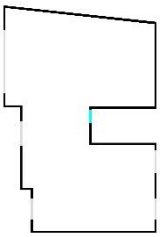
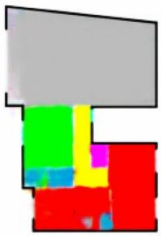
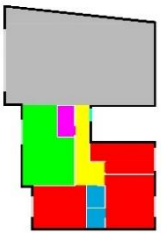
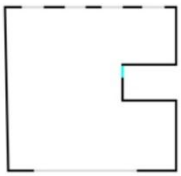
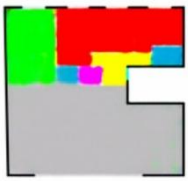
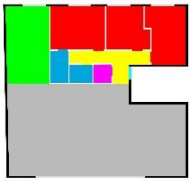
11				**6.91**
12				**6.75**

Table 5
The test data's scores of each criterion.

Test data	Evaluation criteria						Average score
	Topological				Geometrical		
	Zoning relative to the entrance	arrangement relative to each other	arrangement relative to the windows and openings	orientation preferences	Space dimension	Space area	
1	**7**	**7**	**7**	**7**	**6.5**	**6.5**	**6.83**
2	**7**	**6.5**	**6.5**	**7**	**6.5**	**6.5**	**6.66**
3	**7**	**7**	**7**	**6.5**	**7**	**7**	**6.91**
4	**7**	**6**	**7**	**7**	**6.5**	**6**	**6.58**
5	**7**	**6.5**	**7**	**7**	**6.5**	**6.5**	**6.75**
6	**7**	**7**	**7**	**7**	**7**	**7**	**7**
7	**7**	**7**	**6.5**	**7**	**7**	**7**	**6.91**
8	**7**	**6.5**	**7**	**7**	**6**	**6.5**	**6.66**
9	**7**	**7**	**7**	**7**	**6**	**6.5**	**6.75**
10	**7**	**6.5**	**7**	**6.5**	**7**	**7**	**6.83**
11	**7**	**7**	**6.5**	**7**	**7**	**7**	**6.91**
12	**7**	**7**	**7**	**7**	**6**	**6.5**	**6.75**

5. Discussion

With the evolution of the digital age, the profession of architecture has developed new ways to resolve its design problems. These new ways have enabled designers to gain a different perspective and allow them to increase design quality. Recently the latest technological developments, machine learning and deep learning, have received considerable attention in the field of architecture because of the powerful learning and forcasting potentials of neural networks. The experiments in this study demonstrate the possibility of using these models for generative architectural designs. The Pix2Pix is used to generate space layout plan which is helpful in preliminary stages of design. Instead of generating the final floor plan, generating space layout plans are investigated which are more abstract and requires less dataset. In GANs models, the more training data is provided, the more accurate the output from the artificial intelligence becomes and the closer it gets to the user's expectations. In this study, the training data consisted of 660 images, which exceeded the number used in previous studies. The results of this study are consistent with the user expectations and desirable drawing patterns. Therefore, the most important requirement for improving responsiveness and achieving ore desirable geometrical criteria is increasing the quantity of training data in the process of training the artificial intelligence model.

6. Conclusion

In this study, the use of the generative adversarial network for 2D architectural design problems was investigated. Based on the given boundaries, generative adversarial network successfully learn both subjective and objective criteria in the built precedent on its own and improve the performance to generate space layout designs. The validation by an expert shows that the proposed model has

been successful in generating space layout plans to satisfy topological and geometrical constraints. This issue indicates the flexibility and efficiency of the model for use in other cases and designs. The neural network learns multiple features through dataset training, and as a result, develops space layout plans with hidden semantic. The research findings of the training models show that the algorithm learns the procedure of translating an input image (given boundaries) to an output image (space layout plan). It mimics the fundamental conversion patterns in the input and output images of the training dataset. Based on the dataset that was presented to the algorithm, the Pix2Pix model fully organized the topology of the spaces that were generated. In general, more training datasets with authorized layout designs might produce layouts that are better in geometrical constraints. Preprocessing and training all need for designer involvement. As a result, it is not entirely automated, and post processing (such as vector drawing) is still required for the pixel-based end results to be employed as architectural representations. This type of model assists architects in reducing time spent on repetitive tasks especially in the project with complex topological constraints.

References

Arvin, S.A., House, D.H., Modeling architectural design objectives in physically based space planning, Automation in Construction 11 (2002) 213–225, https:// doi.org/10.1016/S0926-5805(00)00099-6.

As, I,. Pal, S., Basu, P., (2018). Artificial intelligence in architecture: Generating conceptual design via deep learning. International Journal of Architectural Computing. 16. 306-327.

Bahrehmand, T. Batard, R. Marques, A. Evans, J. Blat, Optimizing layout using spatial quality metrics and user preferences, Graphical Models 93 (2017) 25–38, https://doi.org/10.1016/j.gmod.2017.08.003.

Bao, F., Yan, D.M., Mitra, N.J., Wonka, P., Generating and exploring good building layouts, ACM Transactions on Graphics 32 (2013), https://doi.org/ 10.1145/ 2461912.2461977.

Bausys, R., and I. Pankrasovaite. 2005. Optimization of architectural layout by the improved genetic algorithm. Journal of Civil Engineering and Management 11 (1):13–21.doi:10.3846/13923730.2005. 9636328.

Bisht, S., Transforming an Adjacency Graph into Dimensioned Floorplan 0, 2022, pp.1–18, https://doi .org/10.1111/cgf.14451.

Borji, A. (2019). Pros and cons of GAN evaluation measures. Computer Vision and Image Understanding, 179, 41-65.

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

Chatzikonstantinou, I., A 3-dimensional architectural layout generation procedure for optimization applications, in: DC-RVD, Proceedings of 2014 ECAADe Conference Vol. 1, 2014, pp. 287–296. http:// papers. Cumincad .org/ cgi-bin/works/paper/ecaade2014_163.

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

Dino, I.G., An evolutionary approach for 3D architectural space layout design exploration, Automation in Construction 69 (2016) 131–150, https://doi.org/ 10.1016/j.autcon.2016.05.020.

Gero, J. S., and V. A. Kazakov. 1997. Learning and re-using information in space layout planning problems using genetic engineering. Artificial Intelligence in Engineering 11 (3):329–34. doi:10.1016/S0954-181 0(96)00051-9.

Geron, A., Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2019). (2nd ed.). O'Reilly.

Goodfellow, I., et al. (2016) Deep Learning. MIT Press, Cambridge, MA.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y. 2014, Generative adversarial nets. NIPS 2014 (Conference on Neural Information Processing Systems.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio. Y., Generative adversarial nets. In NIPS, 2014. 2, 4, 6, 7.

Grover, R., Emmitt, S. & Copping, A. The typological learning framework: the application of structured precedent design knowledge in the architectural design studio. Int J Technol Des Educ 28, 1019–1038 (2018). https://doi.org/10.1007/s10798-017-9421-4.

Guo, Z., Li, B., Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system, Frontiers of Architectural Research 6 (2017) 53–62, https://doi. org/10.1016/j.foar.2016.11.003.

Helme, L., Derix, C., Spatial configuration: semi-automatic methods for layout generation in practice, The Journal of Space Syntax 5 (1) (2014) 35–49. ISSN:2044-7507http://joss.bartlett.ucl.ac.uk/journal/index.php / j oss/article/vie w/201/pdf.

Hua, H., Irregular architectural layout synthesis with graphical inputs, Automation in Construction 72 (2016) 388–396, https://doi.org/10.1016/j. autcon.2016.09.009.

Huang, W. and Zheng, H. (2018), "Architectural drawings recognition and generation through machine learning", Recalibration on Imprecision and Infidelity - Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2018, doi: 10.52842/conf.acadia.2018.156.

Inoue, M., Takagi, H., Layout algorithm for an EC-based room layout planning support system, in: SMCia 2008 : IEEE Conference on Soft Computing in Industrial Applications, 2008, pp. 165–170, https://doi.org/10.1109/ SMCIA.2008.5045954.

Jagielski, R., and J. S. Gero. 1997. A genetic programming approach to the space layout planning problem. In CAAD futures. Dordrecht: Springer; pp. 875-884.

Jo, J. H., and J. S. Gero. 1998. Space Layout Planning using an evolutionary approach. Artificial Intelligence in Engineering 12 (3):149–62. Doi:10. 1016/ S0954-1810(97)00037-X.

Koenig, R., Knecht, K., Comparing two evolutionary algorithm based methods for layout generation: dense packing versus subdivision, AIEDAM - Artificial Intelligence for Engineering Design, Analysis and Manufacturing 28 (2014) 285–299, https://doi.org/10.1017/S0890060414000237.

Koenig, R., Schneider, S., Hierarchical structuring of layout problems in an interactive evolutionary layout system, AIEDAM - Artificial Intelligence for Engineering Design, Analysis and Manufacturing 26 (2012) 129–142, https://doi. org/ 10.1017/S0890060 412000030.

Latha, H., Patil, S. & Kini, P.G. Influence of architectural space layout and building perimeter on the energy performance of buildings: A systematic literature review. Int J Energy Environ Eng (2022).

Liggett, R. S., and J. M. William. 1981. Optimal Space Planning in Practice. Computer-Aided Design 13 (5):277–88.

Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S., Rent3D: floor-plan priors for monocular layout estimation, in: Proceedings of the IEEE Computer Conference on Computer Vision and Pattern Recognition. 07-12-June, 2013, pp. 3413–3421, https://doi.org/10.1109/CVPR.2015.7298963.

Marson, F., Musse, S.R., Automatic real-time generation of floor plans based on Squarified Treemaps algorithm, International Journal of Computer Games Technology 2010 (2010), 624817, https:/ /doi.org /10. 1155/ 2010 /624 817.

Medjdoub, B., Yannou, B., Separating topology and geometry in space planning, CAD Computer-Aided Design 32 (2000) 39–61, https://doi.org/10.1016/S0010- 4485(99)00084-6.

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

Michalek, J. J. Choudhary, R., Papalambros, P. 2002. Architectural Layout Design Optimization. Engineering Optimization.34 (5):461–84. doi: 10.1080/ 03052 1502 1 4 016.

Newton, D. (2019). Generative Deep Learning in Architectural Design. Technology|Architecture + Design. 3. 176-189. 10 .1080/ 2475 1448. 2019. 1 6 40536.

Ozerol, G., & Arslan Selçuk, S. Machine learning in the discipline of architecture: A review on the research trends between 2014 and 2020, (2022). International Journal of Architectural Computing. https://doi.org/10.1177_14780 771221100102.

Peters, N., Enabling Alternative Architectures: Collaborative Frameworks for Participatory Design, Harvard GSD Thesis, 2017.

Phillip, I., Zhu, J., Zhou, T., A. Efros, A., "Image-to-image translation with conditional adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134. 2017.

Rahbar, M., Mahdavinejad, M., Bemanian, M., Davaie Markazi, AH., Hovestadt, L., (2019) Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs, Applied Artificial Intelligence, 33:8.

Rahbar, M., Mahdavinejad, M., Davaie Markazi, A.H., Bemanian, M., Architectural layout design through deep learning and agent-based modeling: A hybrid approach, Journal of Building Engineering, Volume 47, 2022.

Regateiro, F., Bento, J., Dias, J., Floor plan design using block algebra and constraint satisfaction, Advanced Engineering Informatics, Volume 26, Issue 2, 2012, Pages 361-382.

Rodrigues, E., Rodrigues, A., Gomes, ´A., Automated approach for design generation and thermal assessment of alternative floor plans, Energy and Buildings 81 (2014) 170–181, https://doi.org/10.1016/j.enbuild.2014.06.016.

Rosenman, M., Case-based evolutionary design, in: C. Fonlupt, J.-K. Hao, E. Lutton, M. Schoenauer, E. Ronald (Eds.), Artificial Intelligence for Engineering Design, Analysis and Manufacturing 14, Cambridge University Press, 2000, pp. 17–29.

Russell, S. J., Norvig. P., 2016. Artificial Intelligence: A Modern Approach. 3rd ed. Boston: Pearson Education.

Thakur, M.K., Kumari, M., Das, M., Architectural layout planning using Genetic Algorithms, in: Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology. ICCSIT 2010. 4, 2010, pp. 5–11, https: //doi.org/ 10.1109/ ICCSIT. 2010 .5565165.

Topuz, B., & Çakici Alp, N. Machine learning in architecture. Automation in Construction, (2023). 154,105012.https://doi.org/10.1016/j.autcon.2023.10 5012Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. (2004), "Image quality assessment: from error visibility to structural similarities on Image Processing, Vol. 13 No. 4, pp. 600-612.

Weber, R., Mueller, C., Reinhart, C., Automated floor plan generation in architectural design, Automation in Construction, Volume 140, 2022, 104385, ISSN 0926-5805, https://doi.org/ 10.1016/ j.autcon.2022. 104385.

Yi, H., Yi, Y.K., Performance based architectural design optimization: automated 3D space layout using simulated annealing, in: 2014 ASHRAE/IBPSA-USA Building Simulation Conference, 2014, pp. 292–299.http://www.scopus.com/inward/record.url? scp= 84938862924&partnerID=8YFLogxK.