Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs

Morteza Rahbar, Mohammadjavad Mahdavinejad, Mohammadreza Bemanian, Amir Hossein Davaie Markazi & Ludger Hovestadt

To cite this article: Morteza Rahbar, Mohammadjavad Mahdavinejad, Mohammadreza Bemanian, Amir Hossein Davaie Markazi & Ludger Hovestadt (2019) Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs, Applied Artificial Intelligence, 33:8, 689-705, DOI: 10.1080/08839514.2019.1592919

To link to this article: https://doi.org/10.1080/08839514.2019.1592919

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 8, 689–705
https://doi.org/10.1080/08839514.2019.1592919

Taylor & Francis
Taylor & Francis Group

Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs

Morteza Rahbar[a,b], Mohammadjavad Mahdavinejad[a], Mohammadreza Bemanian[a], Amir Hossein Davaie Markazi[c], and Ludger Hovestadt[b]

[a]Department of Architecture, Tarbiat Modares University, Tehran, Iran; [b]Chair for Computer Aided Architectural Design, Department for Architecture, ETH Zürich, Zurich, Switzerland; [c]School of Mechanical Engineering, Iran University of Science & Technology, Tehran, Iran

ABSTRACT

In this paper, a data-driven generative method is applied to generate synthetic space allocation probability layout. This generated layout could be helpful in the early stage of an architectural design. For this task, a specific training dataset is generated which is used to train the cGAN model. The training dataset consists of 300 existing apartment layouts which are coloured in a set of low feature representation. The cGAN model is trained with this dataset and the trained model is evaluated based on the quality of its generated layouts regarding the five pre-defined topological and geometrical benchmarks.

Introduction

Space allocation is a well-known algorithmic problem in computer-aided architectural design. The target of space allocation problem (SAP) is to define an algorithm that could propose a layout (space arrangement) based on topological and geometrical constraints. The topological and geometrical constraints are influenced by different objective and subjective agents. Objective agents are factors that could be defined as numerical objective functions such as architectural program, the energy efficiency of the project, municipality regulations, design standards, client preferences, etc. In opposite, the subjective agents are factors that deal with the mentality of the designer. These factors are more based on the designer's experience than numerical rules. For instance, the aesthetic aspects of the design or some user's environmental behaviors are among subjective agents that influence the topological and geometrical constraints.

In Figure 1 a sample apartment plan is illustrated with the representation of its space allocation in colors and a topological graph of space connectivity. The space allocation representation complies to both topological and

CONTACT Mohammadjavad Mahdavinejad mahdavinejad@modares.ac.ir Department of Architecture, Tarbiat Modares University, Tehran Iran

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

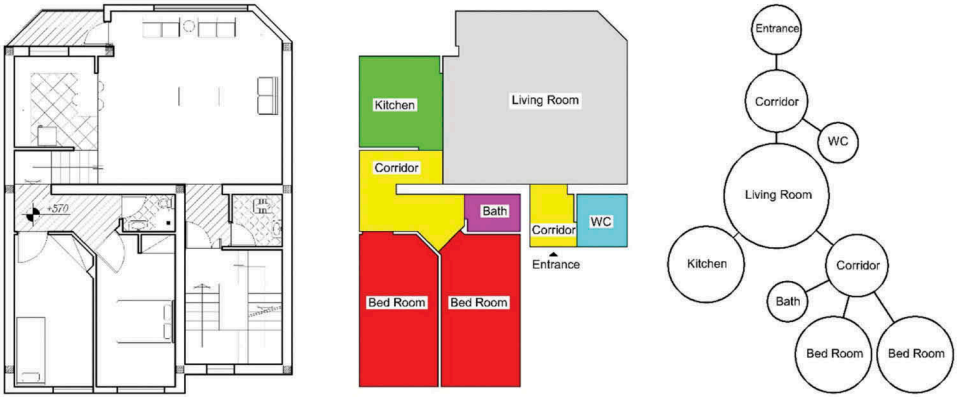


Figure 1. A sample plan of an apartment (Left), Representation of the space allocation with colors (Middle), Topological connectivity of spaces (Right).

geometrical constraints. The size of the rooms, their area and proportion are some geometrical constraints, and the hierarchal arrangement of the space is the topological constraint which they both derive from resultant of objective and subjective agents.

Space allocation problem has been one of the main streams of computer scientist's research since the 1960s. Most of these research relies on solving two important issues: the generation algorithm of variant layouts and a cost function algorithm for evaluating each generated layout. From architect's intuition perspective, a good layout is evaluated as the best possible solution based on all subjective and objective factors. In this case, we usually have several possible solutions which underscore that SAP has an inherent uncertainty which could not necessarily be modeled as a deterministic mathematical relation of spaces. To date, most of the SAP research has simplified the complex evaluation cost function to a specific objective factor and solve it with an optimizer (Bausys and Pankrasovaite 2005; Gero and Kazakov 1997; Gero and Vladimir 1998; Grason 1970, 1971; Jagielski and Gero 1997; Jo and Gero 1998; Levin 1964; Liggett and William 1981; Michalek et al. 2002).

In this paper, the experiments are based on probabilistic data-driven models for generating layouts. Our hypothesis is that each space allocation color representation implies its topological and geometrical constraints which are the resultant of some subjective and objective agents. These subjective and objective agents are hidden semantics that is embedded in the layouts. Instead of using a rule-based optimization model a data-driven prediction modeling approach is applied. Specifically, a conditional generative adversarial network (Isola, Zhu et al. 2017) is trained with the prepared dataset. Since the innovation of generative adversarial networks (GAN) with Ian Goodfellow's influential paper (Goodfellow et al. 2014) in 2014, different branches of GANs for solving different problems have emerged. Conditional generative adversarial networks (cGAN) is one of the main streams (Mirza

and Osindero 2014). In this research, cGAN is trained with a specific dataset which could be used for predicting the probability of space allocation in a given boundary. Our contributions are:

- Generating specific architectural layout dataset for deep learning models
- Train cGAN for space allocation probability layout
- Experiment a specific low-level feature presentation to acquire precise solutions

The main goal of this paper is to propose a practical method predicting the probability of space allocation layout based on data-driven processes. The method and the results of the gradient low feature representation could be applied to the new domain of architectural probability tasks. There are few related works that will be considered in section 5.

Background

The arrangements of predefined functions of an architectural project could have multiple possibilities. An architect proposes a layout design based on his own experience, knowledge, design capabilities and influential parameters of the design. If we consider a large-scale project such as hospitals with the complicated relation of space, then we could encounter a very hard task and architect's final proposed layout is just what he could design based on his experience, knowledge, and capabilities. This is the case that CAAD could help. And modeling this task is the challenging part.

In the last few decades, many scholars have worked on defining an algorithm for architectural layout design. The trace dates back to 1960s where Levin (Levin 1964) wrote a book on finding the optimum architectural layout using the graphs. The application of graphs was one of the main streams in architectural layout design at the early stages. Grason (Grason 1970, 1971) also used graphs for floor plan representation and computerized space planning. According to Liggett (Liggett 2000), there are another two main approaches to space planning in addition to graph theory. Both approaches rely on optimization techniques but differ on the number of objective functions. For instance, Liggett & William (Liggett and William 1981) apply quadratic assignment optimization with single cost function of maximizing workflow efficiency between activities. Since the 1990s, many scholars focused on applying evolutionary algorithms (Bonnaire and Riff 2002; Michalek et al. 2002) for solving space planning problems. Specifically, genetic algorithm (Bausys and Pankrasovaite 2005; Gero and Vladimir 1998; Jo and Gero 1998; Wong and Chan 2009) and genetic programming (Gero and Kazakov 1997; Jagielski and Gero 1997) were the main interest. In recent years some works have focused on multi-objective

optimization (Inoue and Takagi 2008) (Inoue and Takagi 2009) and hybrid optimization techniques (Rodrigues et al. 2013).

The overall progress of previous works is admirable and each of them has solved a part of the problem and reached to a specific goal. However, almost all of them have focused around defining a logical and mathematical relation between the spaces, which basically differs to how human intelligence deals with a space planning task. There are several parameters such as local municipality regulations, the meaning of beauty, client's preferences, cultural factors, regional and religious aspects, design trends and so many other similar parameters that they are all hidden rules directly influencing the layout design. The human intelligence and designers respect to these factors based on their experience, knowledge and design capabilities. The proposed layout design of the designer could change from time to time and place to place. The variation of possible solutions proves that the layout design lies on an uncertainty base and there is more than one absolute final answer. It is difficult to quantify the complex parameters of space planning problem and it requires a new type of interaction to leverage human intuition and computational capabilities in partnership.

Overview of Our Approach

Regarding the possibility of multiple solutions for a space planning task, the generative algorithm requires a probabilistic model instead of a deterministic optimized mathematical model. The algorithm should have the capability of improvement exactly as designers gain experience in their design. Regarding this fact, GANs are studied as a model to generate synthetic layouts. GAN stands for generative adversarial networks and it was first introduced by Ian Goodfellow in his paper (Goodfellow et al. 2014) in 2014. GAN consists of two neural networks playing and min-max game in an iterative optimization process. The first network generates a synthetic data from random noise and the second network discriminates the real and fake data. Since Ian Goodfellow's paper, many other scholars worked on different branches of GANs. Conditional GAN is one the popular branches of GAN presented by Philip Isola (Isola, Zhu et al. 2017) in 2017. In his paper (Isola, Zhu et al. 2017) he defines the automatic image-to-image translation as the problem of translating one possible representation of a scene into another, given sufficient training data. The pixel paired method applies convolution and deconvolution layers and translates a pixel with a specific value to a new value.

Figure 2 illustrates two examples of image to image translation presented in Phillip Isola's paper (Isola, Zhu et al. 2017). In the first pair of images an input day scene is translated to an output night and in the second pair, an input black and white image is translated into a color image. The image to image translation could also be used in the field of semantic segmentation. Few semantic experiments have

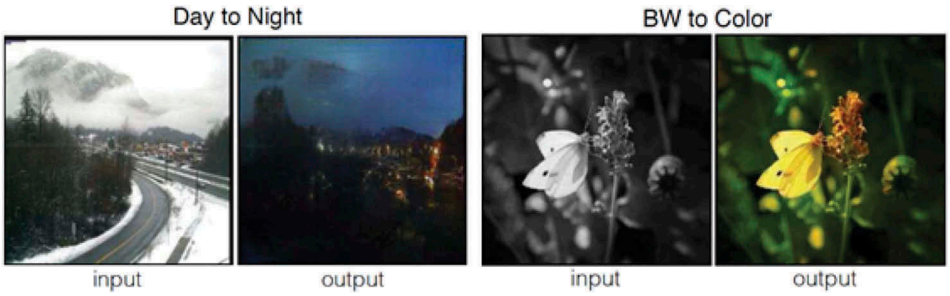


Figure 2. Image to image translation examples (Isola, Zhu et al. 2017).

been carried out by Isola in his paper such as translation of semantic labels to street scene and aerial photos to semantic maps (Figure 3). In both cases, the translation could be done in the other direction as well.

In the field of architectural layout and space allocation, we deal with hidden semantics laid in the relation of spaces and their composition. In segmentation semantics, each pixel correlates to its paired pixel on the other image but in hidden semantics, there is no specific correlation of pixels meaning. In Figure 4 an architectural plan is illustrated and on its counter side, the rooms are segmented. This is a sample of semantic segmentation and room detection in an architectural layout. The algorithm could exactly detect which pixel is a part of a wall or a room.

But how about an algorithm which could allocate different spaces given an existing boundary. In this case, there is no direct correlation between the paired pixels and instead, the algorithm should decide where to allocate each

Figure 3. Semantic segmentation with Image to image translation algorithm (Isola, Zhu et al. 2017).

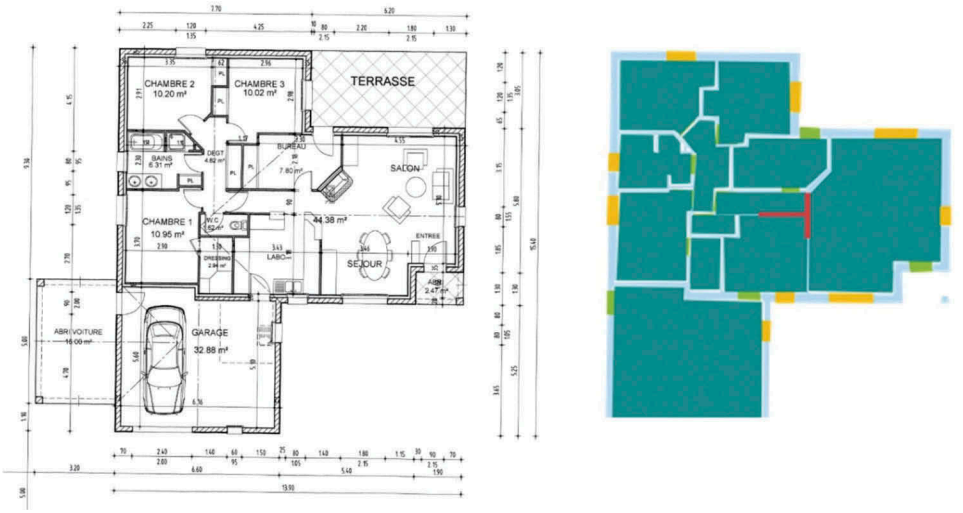


Figure 4. Segmentation of architectural layout spaces.

of the given spaces based on geometrical and topological constrains. This is what we call it as hidden semantics in an architectural layout. As mentioned before the design task is a non-deterministic task which could have several solutions so instead of a final design solution we are looking for a probability heat map. In this research, conditional GAN is used as a method to mimic the human-driven design process. The cGAN generates space allocation probability heat maps and instead of an exact drawing of each space, it colors each pixel regarding the experienced it has gained from the training dataset. To train the cGAN model, in the first step two publicly available architectural layout datasets (de Las Heras et al. 2015) (Rakuten 2014) were studied and none of them were useful for the space allocation task. These two datasets are prepared for a semantic pixel wised room detection task. Due to the lack of reasonable dataset, a specific dataset is prepared which will be described in detail in the next section.

Training and Evaluation Procedure

cGAN

The cGAN algorithm consists of two adversarial models. The first one is the generator model named G which generates synthetic data and the second one is a discriminator model named D which distinguishes the real data from the fake data. The generator model in cGAN algorithm builds a mapping function from a prior noise vector z and input image x to a data space. The discriminator model outputs a single scalar representation expressing the probability of G(x) as a fake or real image (Isola, Zhu et al. 2017) (Mirza and

Osindero 2014). Each of these models has its own objective function. The G model tries to generate more realistic synthetic images similar to the target images and the D model should try to distinguish it as fake data. The G model tries to generate real and high précised images to fool the D model as a real image and meanwhile, the D model tries to distinguish it as fake. In the cGAN iterative training process, the D and G model enter a competitive game where each of these models tries to minimize or maximize the objective function. The overall objective function of a cGAN can be expressed as:

$$\mathcal{L}_{cGAN}(G.D) = \mathbb{E}_{x.y}[\log D(x.y)] + \mathbb{E}_{x.z}[\log(1 - D(x.G(x.z)))]$$

In this objective function, the G tries to minimize the objective and the D tries to maximize the objective function. This is called the min-max game between the two models. Some scholars mix the min-max objective function with an L2 distance objective function to force the G model to generate a synthetic data near the ground truth output as well as trying to fool the D model (Pathak et al. 2016). Some other scholars apply the L1 distance objective function instead of L2 to reduce the blurring effect (Isola, Zhu et al. 2017). In this paper, we adopt the L1 approach to reduce the blurriness of architectural layout generated design. The L1 objective function and the final objective function are as follows:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x.y.z}[\| y - G(x.z)\|_1]$$

$$G = \arg\min_G \max_D \mathcal{L}_{cGAN}(G.D) + \lambda\mathcal{L}_{L1}(G)$$

The cGAN model used in this research is an Image-to-Image translation based in Tensorflow machine learning framework and it could be accessed from (Hesse 2017). This model uses a "U-Net"-based architecture as the generator and a convolutional "PatchGAN" classifier as a discriminator. The final answer of the discriminator is a number between 0 and 1 which indicates the probability of the real and fake answer.

Dataset

This research investigates how a probability space allocation map could be proposed based on a given boundary. Regarding the procedure of cGAN training, a pair of same-sized images are required as training dataset. This training dataset is an important part of our research and we investigate that how low representation of the training dataset could affect the quality of the synthetic generated data. The dataset is a pair of images, the first is the boundary representation of the project and the second one is the available layout design. The first image is loaded as an input and the second image is loaded as a conditional target image. The cGAN training is actually the

procedure of customizing the parameters of both generator and discriminator models to cover the mapping of input to output images in all over the dataset. To avoid under-fitting and over-fitting of the model, 300 apartment plans are collected and each of their spaces is labeled with a specific color.

The procedure of generating the pair images dataset from an existing apartment is illustrated in Figure 5. On the left side, there is a sample apartment plan. The middle image is generated for the input image of the cGAN. It only has the boundary wall of the apartment. The staircase, columns, and terraces are excluded. The walls are colored in black, and the windows are colored in light grey. The location of the door entrance is marked with a thin black line. These are the low feature of the input image. For the output image, a colored layout is generated. In this layout, the bedrooms are labeled as red colors, the kitchen as green, the corridor as yellow, the bathroom as purple and WC as blue. It also has the walls and windows of the input image. The RGB colors are the low features that network is trained with. In the cGAN training procedure, the network learns how to map the input images to their corresponding output images based on the pixel-wise method (Figure 6).

In Figure 6, a sample corner of an input image and the corresponding output image is illustrated. After the training, the cGAN learns how to map the low features of the input pixel to the low features of the corresponding output pixel. Here the walls and the windows are exactly mapped to their own colors but the white part at this corner is mapped to red color which is a bedroom. The question is how could the model recognize that this white part is not a bathroom or a kitchen. The answer is that the up-sampling and down-sampling of the two networks in cGAN change the image dimensions.

Figure 5. Labeling the Floor plans for cGAN training (Left: Original plan, Middle: Input image, Right: labeled output image).

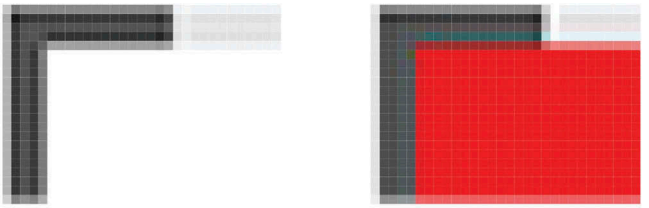


Figure 6. Pixel-wise Learning in cGAN.

In this procedure, the pixels and their neighbors are taken into account and the final low feature output of a pixel is the result of the interaction of each pixel and its neighbors. Considering all of the images in the dataset, customizing the parameters of the two networks to avoid underfitting and overfitting is the challenging part. Here, a sample representations of input low features are generated to train the cGAN.

In Figure 7 a sample generated pair image for cGAN training dataset is represented. In this image, the gradient colors have been used to emphasize both the openings and the entrance. The gradient brightness varies based on the size of the opening, and the entrance area is colored with a gradient yellow color.

In Figure 8, a part of the 300 dataset is presented. To specify the size of each apartment, all of the plans in the dataset is normalized based on their area. The boundary wall of all dataset is also set to 20 cm width so there is no difference in the thickness of the walls. The cGAN model is trained on all of the 300 dataset which is described in the next section.

Training the cGAN

The cGAN is trained based on the prepared dataset and after the training, the trained model could generate synthetic images similar to target images. In Figure 9 a sample of training procedure is presented.

As illustrated in Figure 9, the D model (discriminator) learns to classify the pair of images as real or fake. Meanwhile, the G model (generator) learn to fool the discriminator to classify the generated image as a real image. In this research, the architecture of the two models from Isola's paper (Isola, Zhu et al. 2017) is

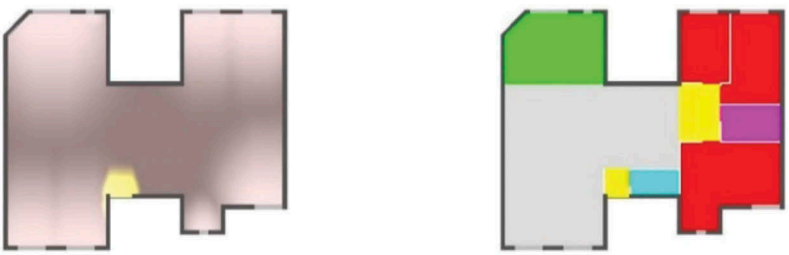


Figure 7. Generated pair images for cGAN training dataset.

Figure 8. A part of the third training dataset.

adopted. The G models is a U-Net-based generator and the D model is a convolutional Patch-GAN classifier. The U-Net in this model uses the convolutional and deconvolutional layer with the possibility of skipping between the layers. The generator downsamples the dimensionality of the input image to a vector and upsamples the dimension back to its original size. In this process, a new image is generated which is called the synthetic image. In the next step, the generated image and the input are loaded into the discriminator model as a tuple. The discriminator uses a Patch-GAN classifier which classifies this tuple as fake or real only by checking and penalizing a small patch of the tuple's images. This Patch-GAN makes it possible to classify a higher resolution of an image in a reasonable amount of time.

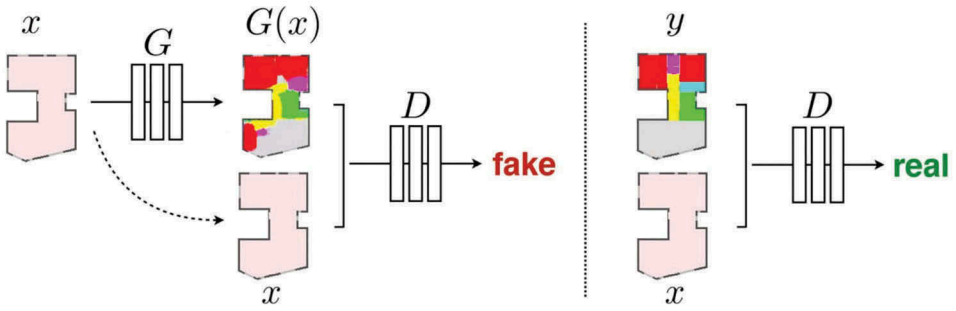


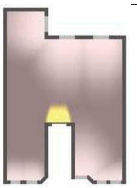
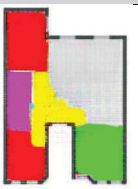
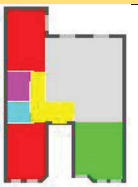
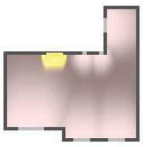
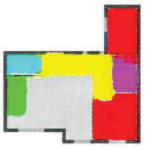
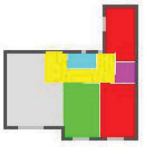
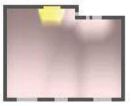
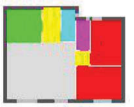
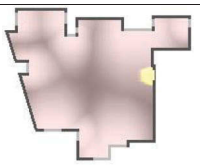
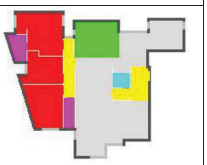
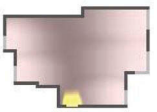
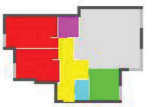
Figure 9. The procedure of training a conditional GAN to map a given boundary to a space allocation map.

Results & Evaluation

As mentioned previously, the input x sets are loaded into the trained cGAN models to map the input to the correspondent output set y. After the training, the model is evaluated based on test results. For the test sets, 10 different boundary lines of architectural layouts with different dimensions and concepts are loaded as input datasets. These boundary lines are absolutely new to the trained models and they do not exist in their training dataset. The training model generates an output based on what it has learned. In Table 1, the generated output is presented. There is also a column of expert design which illustrates a sample layout designed by an architect based on his experience. As discussed earlier there is no exact final solution for a design task and there are multiple possible solutions.

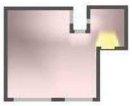
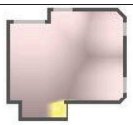
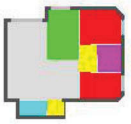
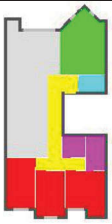
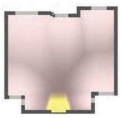
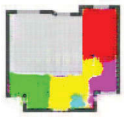
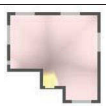
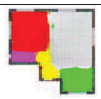
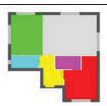
The outputs of each test data are evaluated based on topological and geometrical criteria. For this task five benchmark of 'Orthogonal design', 'Dimensions of the spaces', 'Proportion of space's area', 'Entrance recognition' and 'Logics of space allocation' is considered. 'Orthogonal design' considers how accurate each model, designs spaces based on orthogonal lines. In this study, all of the training dataset have orthogonal lines and it is important to evaluate how accurate the model has learned this feature. 'Dimensions of the spaces', evaluates if the generated layout of each model has spaces with correct dimension. For example, a bedroom with 1*1.5-m dimension shows that the model has not learned the dimension feature properly. 'Proportion of space's area' checks whether the model generates each space with standard area based on what it has learned. For example, a 70-sqm apartment with three bedrooms of overall 55 m and only 15 m left expresses that the model has not learned the proportions of the spaces correctly. 'Entrance recognition', evaluates if the model has correctly detected the entrance location and it has proposed a corridor at the entrance of the apartment. 'Logics of space allocation' checks the location of each singular space and its relation with the other spaces. The location of each space should correspond to the windows, openings and other spaces. For example, if the model generates a bathroom exactly at the middle of a kitchen or if it generates

Table 1. Output results of the trained cGAN and an expert designer.

	Test data's boundary	Synthetic plan generated by the trained model	Expert design
1		 B1 B2 B3 B4 B5 4.5 5 4.5 5 5 **Validity Index: 4.8**	 B1 B2 B3 B4 B5 5 5 5 5 5 **Validity Index: 5**
2		 B1 B2 B3 B4 B5 5 4.5 4.5 5 5 **Validity Index: 4.8**	 B1 B2 B3 B4 B5 5 5 5 4.5 5 **Validity Index: 4.9**
3		 B1 B2 B3 B4 B5 4.5 4 4 5 5 **Validity Index: 4.5**	 B1 B2 B3 B4 B5 5 5 5 5 4.5 **Validity Index: 4.9**
4		 B1 B2 B3 B4 B5 5 4 4.5 5 5 **Validity Index: 4.7**	 B1 B2 B3 B4 B5 5 5 5 5 4.5 **Validity Index: 4.9**
5		 B1 B2 B3 B4 B5 4 5 4.5 5 5 **Validity Index: 4.7**	 B1 B2 B3 B4 B5 5 5 5 5 5 **Validity Index: 5**

(*Continued*)

Table 1. (Continued).

6		 B1 B2 B3 B4 B5 4.5 4 5 5 5 **Validity Index: 4.7**	 B1 B2 B3 B4 B5 5 4.5 5 5 4.5 **Validity Index: 4.8**
7		 B1 B2 B3 B4 B5 4 4.5 4.5 5 5 **Validity Index: 4.6**	 B1 B2 B3 B4 B5 5 5 5 5 4.5 **Validity Index: 4.9**
8		 B1 B2 B3 B4 B5 4 4.5 5 5 5 **Validity Index: 4.7**	 B1 B2 B3 B4 B5 5 5 5 5 4.5 **Validity Index: 4.9**
9		 B1 B2 B3 B4 B5 4 4 4.5 5 5 **Validity Index: 4.5**	 B1 B2 B3 B4 B5 5 5 5 5 4.5 **Validity Index: 4.9**
10		 B1 B2 B3 B4 B5 4.5 4.5 5 5 5 **Validity Index: 4.8**	 B1 B2 B3 B4 B5 5 5 5 5 5 **Validity Index: 5**

a bedroom with no windows, it shows that the model has not learned the logics of space allocation properly. In Table 2 the 'Orthogonal design' evaluation result is marked as B1, 'Dimensions of the spaces' is marked as B2, 'Proportion of space's area' is marked as B3, 'Entrance recognition' is marked as B4 and 'Logics of space allocation' is marked as B5. Finally, the Validity Index is computed by averaging the benchmark's evaluation results.

The trained model in Table 1 is based on the dataset illustrated in Figure 8. The test data's boundary column is illustrating the input of test dataset. The two outputs, one from the cGAN model and another is a sample designed by an expert architect is evaluated based on the five benchmarks. The validity Index of each evaluation is computed. In Table 2 a summary of benchmark evaluation and the validity index of the 10 test dataset is presented. In this table, the 10 generated layouts are evaluated independently with discreet numbers from 1 = very weak to 5 = strong in each defined benchmark. The validity Index of each test data and also the average validity index of the trained model is computed.

The validity index and benchmark scores of the test data demonstrate that the trained model gained full score in both entrance recognition and the logics of space allocation and in some test evaluations it has also gained a better score comparing to the expert design. In the first three benchmarks, we could see different scores which indicate that the models require more training dataset.

Related Works

Researchers from artificial intelligence community have already put many efforts to generate synthetic data with deep learning techniques. A group of researchers in Stanford University use the generative adversarial networks to design shoes (Deverall et al. 2017). They use trained GAN models with shoe dataset to generate synthetic shoe designs for preliminary steps. A group of scholars in Berkeley AI Research Laboratory, propose Cycle-Consistent Adversarial Networks for an unpaired image to image translation (Zhu et al. 2017). In their research, they

Table 2. Evaluation of the generated layouts based on the five benchmarks.

Test Dataset	Orthogonal design	Dimensions of the spaces	Proportion of space's area	Entrance recognition	Logics of space allocation	Validity Index
1	4.5	5	4.5	5	5	4.8
2	5	4.5	4.5	5	5	4.8
3	4.5	4	4	5	5	4.5
4	5	4	4.5	5	5	4.7
5	4	5	4.5	5	5	4.7
6	4.5	4	5	5	5	4.7
7	4	4.5	4.5	5	5	4.6
8	4	4.5	5	5	5	4.7
9	4	4	4.5	5	5	4.5
10	4.5	4.5	5	5	5	4.8
Average	4.4	4.4	4.6	5	5	4.68

present a method which could generate synthetic data similar to input image but in a new style. Leon Gatys and his research team worked on generating synthetic artistic style photos (Gatys et al. 2015). Conditional cycle-GANs were also used to generate synthetic face photos (Lu et al. 2017).

More specifically in the field of architectural design, some scholars study the application of deep learning in segmentation tasks. Samuel Dodge applies fully convolutional networks (FCN) for wall segmentation in architectural layouts. In his paper 'Parsing floor plan images' (Dodge et al. 2017), he proposes an FCN-2s with a two-pixel stride layer architecture to gain the best result in wall segmentation. Some other scholars (Ahmed et al. 2012) use the deep learning architectures for automatic room detection task. In the field of architectural generative design, the Autodesk company is leading a research group focusing on generative design & AI. Autodesk's new Toronto office is one of the first examples of a generatively designed office space (Nagy et al. 2017).

Discussion and Future Works

The experiments in this study demonstrate the possibility of using data-driven models for generative architectural designs. The cGAN is used to generate probability space allocation map which is helpful in early stages of design. Instead of generating the final floor plan, generating SP probability maps are investigated which are more abstract and requires less dataset. The experiment of this research revealed that the low-feature representation of the dataset and the number of the dataset is a key success in the final generated layout.

On our future works, we will extend the number of training dataset to test its effect on the quality of the results. We will also work on new representations of dataset's low-features to generate more reasonable probabilistic maps. In this study, our input and target images had 256*256 pixel and we will work on higher resolution images which could help the cGAN to learn more details but it would also take more time for training. We will also test the effect of cGANs hyper-parameters in the quality of the results.

Acknowledgments

This paper is part of Ph.D. thesis entitled "Architectural layout design optimization" conducted by the first author at Tarbiat Modares University under the supervision of Prof. Mohammadjavad Mahdavinejad, Prof. Mohammadreza Bemanian, Prof. Amirhossein Davaiemarkazi and partially done at ETH Zurich under the supervision of Prof. Ludger Hovestadt.

Glossary

Up-sampling: Increasing the size of the image (data) with de-convolutional layers

Down-sampling: Reducing the size of input image (data) with convolutional layers

GAN: Generative Adversarial Networks

cGAN: Conditional Generative Adversarial Network

SAP: Space Allocation Problem

L1 loss function: Also known as Least Absolute Error (LAE) is the process of minimizing the sum of the square of the differences (S) between the target value (y_i) and the estimated values ($f(x_i)$):

$$S = \sum_{i=1}^{n} |y_i - f(x_i)|$$

L2 Loss function: Also known as Least Square Error (LSE) is the process of minimizing the sum of the square of the differences (S) between the target value (y_i) and the estimated values ($f(x_i)$):

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Overfitting: Occurs when the machine learning model fits the training dataset too well and it disables the model to generalize to a new set of test data.

Underfitting: Occurs when the machine learning model does not fit the training dataset well enough that it could not generate acceptable output results on the training data.

Training a model: Is the process of optimizing the parameters of machine learning models based on input and output training dataset.

Hyperparameters: There are some parameters in machine learning models whose value are set before learning process starts. The other values are set through iterative optimization process.

References

Ahmed, S., Liwicki, M., Weber, M., & Dengel, A. 2012. Automatic room detection and room labeling from architectural floor plans. 10th IAPR International Workshop on Document Analysis Systems (DAS- 2012). 2012 Mar 27. IEEE: 339–43.

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.

Bonnaire, X., and M. C. Riff. 2002. A self-adaptable distributed evolutionary algorithm to tackle space planning problems. *International Workshop on Applied Parallel Computing* 403–10, Springer, Berlin, Heidelberg.

de Las Heras, L.-P. Terrades, O. R., Robles, S., Sánchez, G. 2015. CVC-FP and SGT: A new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition (IJDAR)*. 18(1):15–30. doi:10.1007/s10032-014-0236-5.

Deverall, J., Lee, J., Ayala, M. 2017. Using Generative Adversarial Networks to Design Shoes: The Preliminary Steps. *Stanford University CS231n*.

Dodge, S., Xu, J., Stenger, B. 2017. Parsing floor plan images. Fifteenth IAPR International Conference on Machine Vision Applications (MVA-2017), IEEE: 358–61.

Gatys, L. A. et al. 2015. A neural algorithm of artistic style. *Nature Communications* arXiv preprint arXiv:1508.06576.

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/S0954-1810(96)00051-9.

Gero, J. S., and A. K. Vladimir. 1998. Evolving design genes in space layout planning problems. *Artificial Intelligence in Engineering* 12 (3):163–76. doi:10.1016/S0954-1810(97)00022-8.

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 27) proceedings. *Advances in Neural Information Processing Systems*:2672–80.

Grason, J. 1970. A dual linear graph representation for space-filling location problems of the floor plan type. *Emerging Methods in Environmental Design and Planning*: 170–178. MIT Press, Cambridge, MA.

Grason, J. 1971. An approach to computerized space planning using graph theory. *Proceedings of the 8th Design Automation Workshop*, *Atlantic City, ACM*: 170–78

Hesse, C. 2017. Image-to-Image Translation in Tensorflow. *from https://github.com/affinelayer/pix2pix-tensorflow*.

Inoue, M., and H. Takagi. 2008. Layout algorithm for an EC-based room layout planning support system. *IEEE Conference on Soft Computing in Industrial* Applications *(SMCia'08)*, IEEE: 165–70 doi: 10.1093/jjco/hyn005

Inoue, M., and H. Takagi. 2009. EMO-based architectural room floor planning. Systems, Man and IEEE *International Conference on Cybernetics (SMC 2009)*, IEEE: 518–23

Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition.

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.

Levin, P. H. 1964. Use of graphs to decide the optimum layout of buildings. *The Architects Journal* 7:809–15.

Liggett, R. S. 2000. Automated facilities layout- past- present- future. *Automation in Construction* 9 (2):197–215. doi:10.1016/S0926-5805(99)00005-9.

Liggett, R. S., and J. M. William. 1981. Optimal Space Planning in Practice. *Computer-Aided Design* 13 (5):277–88. doi:10.1016/0010-4485(81)90317-1.

Lu, Y., Tai, Y. W., Tang, C. K. 2017. Conditional cyclegan for attribute guided face image generation. arXiv preprint arXiv:1705.09966.

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

Mirza, M., and S. Osindero. 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Benjamin, D. 2017. Project Discover: An application of generative design for architectural space planning. *In Proceedings of the Symposium on Simulation for Architecture and Urban Design (p. 7). Society for Computer Simulation International.*

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A. 2016. Context encoders: Feature learning by inpainting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, Nevada, United States. 2536–44.

Rakuten. 2014. Floor plan from Rakuten Real Estate and pixel-wise wall label. *Rakuten*. Online: https://rit.rakuten.co.jp/news/article/information/2017/1128_01/

Rodrigues, E., Gaspar, A. R., Gomes, Á. 2013. An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. *Automation in Construction* 35:482–98. doi:10.1016/j.autcon.2013.06.005.

Wong, S. S. Y., and K. C. C. Chan. 2009. EvoArch - An evolutionary algorithm for architectural layout design. *Computer-Aided Design* 41 (9):649–67. doi:10.1016/j.cad.2009.04.005.

Zhu, J.-Y., Park, T., Isola, P., Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593.