APARTMENT FLOOR PLANS GENERATION VIA GENERATIVE ADVERSARIAL NETWORKS

HAO ZHENG1, KEYAO AN2, JINGXUAN WEI3 and YUE REN4
1,2,3 University of Pennsylvania, Philadelphia, USA
1 zhao@design.upenn.edu  2,3 {skylaran|weijingx}@seas.upenn.edu
4 University College London, London, UK
4 yue.ren.19@ucl.ac.uk

Abstract. When drawing architectural plans, designers should always define every detail, so the images can contain enough information to support design. This process usually costs much time in the early design stage when the design boundary has not been finally determined. Thus the designers spend a lot of time working forward and backward drawing sketches for different site conditions. Meanwhile, Machine Learning, as a decision-making tool, has been widely used in many fields. Generative Adversarial Network (GAN) is a model frame in machine learning, specially designed to learn and generate image data. Therefore, this research aims to apply GAN in creating architectural plan drawings, helping designers automatically generate the predicted details of apartment floor plans with given boundaries. Through the machine learning of image pairs that show the boundary and the details of plan drawings, the learning program will build a model to learn and generate image data. This automatic design tool can help release the heavy load of architects in the early design stage, quickly providing a preview of design solutions for architectural plans.

Keywords. Machine Learning; Artificial Intelligence; Architectural Design; Interior Design.

1. Introduction

1.1. BACKGROUND

In architectural design, the design process usually follows specific criteria, arranging objects inside space. Despite the cultural and aesthetic considerations, architectural design can be regarded as a box of geometric operations, inputting the design boundary (available space for the design) and outputting the layout of the contents inside the boundary.

As a presentation of architectural design, projected drawings, especially floor plan drawings, have been widely used since the Renaissance (Alberti 1988). With the rapid development of Computer-Aided Design (CAD) technology from the 1960s (Carpo 2017), drawing digitally has been widely practiced by architects.
Thus, digital image files showing the plan drawings have become the most popular format to communicate between architects and clients.

1.2. PROBLEM STATEMENT

The interior design, the most commonly practiced field in our daily life, however, is subject to the architectural design. For example, the boundary for an apartment might change during the overall design process of the whole floor. Thus a small change in the design boundary will cause a redesign request to the entire interior layout, in which he designers spend a lot of time working forward and backward drawing sketches for different site conditions. Still, most of the floor plans they provide will be discarded because of the adjustment of the boundary.

1.3. PROJECT GOAL

Thus, in the age of Artificial Intelligence, we propose a Machine Learning method to make the computer provide the interior design layout in the early design stage, releasing the heavy load from architects. Generative Adversarial Network (GAN) (Goodfellow, Pouget-Abadie, et al. 2014) supports the learning and generating of real-world images, including architectural floor plan images. The final goal of this research is to build a tool that takes images showing the design boundary as the input and outputs images showing the detailed interior design inside the boundary.

1.4. LITERATURE REVIEW

Previous research about the usage of GAN in architectural design includes the transformation of city maps to satellite images (Zheng 2018); the generation of furniture layout (Huang and Zheng 2018); the recognition of different rooms in architectural plans (Zheng and Huang 2018), and the recognition of different architectural elements (doors, windows, etc.) (Kvochick 2018). This research will fill in the gap of generating detailed apartment floor plans by only providing the design boundary, showing the possibility of applying Machine Learning in more complicated and creative design works.

2. Methodology

2.1. DATA LABELLING AND CLEANING

Two datasets from previous research were used in the learning process. The first dataset contains 1279 images of apartment floor plans in Japan (Liu, Wu et al. 2017), and the second dataset contains 112 images of apartment floor plans in China (Huang and Zheng 2018). The design and drawing styles are different; thus, two GAN models will be trained separately using these two data sets. However, the images in the data set only contain the design layouts without the boundaries, so the first step is to produce the boundary images based on the plan drawings.

First, each image is resized to a bounding area of 500 pixels X 500 pixels, and placed in the middle of a canvas of 512 pixels X 512 pixels, leaving a white margin of 6 pixels in each side. Then, a Photoshop script automatically detects the continuous white pixels from margin to the original image, and find the boundary
of the plan. Last, by inversely selecting the area and filling black pixels inside the boundary, the boundary image is produced (figure 1). Thus, the generated boundary image and the resized plan image together become an image pair for the training of GAN.

Figure 1. Image resizing and boundary detecting.

After the production of the boundary images, we found there are two types of images that are not suitable for learning in the first dataset (figure 2). One is for having an unrelated rectangular boundary box, and the other one is for having a separate bounding colored area. Thus we removed nine images from the first dataset. There are no such problems in the second dataset.

Figure 2. Invalid image pairs.

2.2. NETWORK STRUCTURE

In this research, the mapping between the input boundary image and the output plan drawing is structurally precise. Image-based neural network GAN (Generative Adversarial Network) with convolution and deconvolution kernels was used as the framework. Based on the conditional GAN invented by (Mirza and Osindero 2014), pix2pixHD (Isola, Zhu, et al. 2017), an open-source project, was finally chosen as the algorithm for this research.

Figure 3 shows the network structure of pix2pixHD, in which two neural networks, Generator (G) and Discriminator (D), are trained simultaneously. The Generator acts to transform an input image to an output image with the same size, using convolutional, residual, and deconvolutional layers. The Discriminator works to distinguish the image generated by the Generator from the ground truth image. The Generator feeds forward the generated result to the Discriminator, while the Discriminator feeds back the loss and gradient to the Generator. Thus the Generator is trained to generate the fake images closer to the ground truth,
while the Discriminator is trained to tell the fake image better apart. The two networks are “competing” with each other, so this system is called “adversarial.”

![Figure 3. Training a conditional GAN (Isola, Zhu, et al. 2017).](image)

However, to apply pix2pixHD, in addition to provide the program with image pairs described previously, some important hyper-parameters also need to be defined. First, we don’t offer instance maps; thus, the function to read and use instance maps should be turned off. Second, we want the program to use RGB colors as input directly, so we set the label_nc as 0. Third, during experiments we found the early training epochs with constant learning rate would not improve the network after 50 epochs, thus we set the learning rate for the first 50 epochs of the training process as a constant value, but the learning rate for the following 50 epochs as decaying values. All other settings are the same as the default settings.

2.3. NETWORK TRAINING

During the training process, the loss values of the Generator and the Discriminator were recorded. Figure 4 and figure 5 show the loss of the two models. Generally speaking, in both models, the Generator loss and the Discriminator loss did not converge, since the two neural networks were, in fact, competing for each other. To be specific, when the loss of the Generator was lower, the loss of the Discriminator would be higher. This phenomenon indicates that the training was successful; the whole system gradually got improved during the balance of the Generator and the Discriminator. What’s more, the loss in model 1 was comparatively higher than the loss in model 2, because the variance of dataset 1 is higher than that of dataset 2.

![Figure 4. Training Loss of Model 1. Right: generator loss; Middle: discriminator loss on fake images; Left: discriminator loss on rake images.](image)
Since the loss cannot tell us whether the model has been converged or not, in each training epoch, an input image was sent to the neural network, and the output image was recorded. By comparing the generated image with the ground truth image, we can decide whether the performance is satisfied and whether to stop the training.

Figure 6 and figure 7 show the image pairs in each training epoch for model 1 and model 2. At the beginning of the training, the generated images were very blurry. As the training went on, the quality of the generated images was gradually improved. Especially in model 2, the synthesized image in epoch 100 performed nicely, showing a clear pattern of different rooms and furniture. Thus, we decided to stop the training in epoch 100 and stored the final models for the following predictions.

Figure 6. Testing image pair in each training epoch for model 1.
To train the two models, it took a Titan X GPU 33 hours of training for the first data set, and 2.7 hours for the second data set. The time cost is acceptable for most of the cases in the architectural design industry.

3. Results

3.1. TESTING DATA GENERATING USING JAPANESE MODEL

Figure 8 and figure 9 show two Chinese apartments generated by Model 1 (Japanese training dataset). The apartment in figure 8 is an example, while the model does not generate very well. It is hard to tell the positions of the living room, the kitchen, and the bedrooms. The apartment in figure 9 is an example, while the model generates pretty well. Even though the kitchen size in the synthesized image is different from that in the real image, all other parts, including the living room and bedrooms, match perfectly. And the overall layout makes sense to architectural designers.
Figure 9. Testing image pair 2. Using the Japanese model to predict Chinese data. Left: input label; Middle: synthesized image; Right: real image.

Figure 10 and figure 11 show two Japanese apartments generated by Model 1. The apartment in figure 10 is an example, while the model performs poorly. The layout in the synthesized image is blurring. It is hard to tell apart rooms in the image. Thus, this synthesized image does not help to predict the design case. The apartment in figure 11 is an example, while the model generates well. Even though the size of the living room in the synthesized image is different from that of the real image, all other parts, including the living room and bedrooms, almost match, both the synthesized images and the real images show designs with a similar drawing style.

Therefore, the model trained with the Japanese dataset is unstable when feeding different design boundaries.

3.2. TESTING DATA GENERATING USING CHINESE MODEL

Figure 12 and figure 13 show the image pairs deserved by using Model 2 (Chinese training dataset) to generate the Japanese apartment. They both perform poorly.
The synthesized image in figure 12 can show the basic structure of the apartment. But the application of the room and the interior decorations are generated wrongly. While the synthesized image in figure 13 even cannot generate the basic structure of the apartment.

Figure 12. Testing image pair 5. Using the Chinese model to predict Japanese data. Left: input label; Middle: synthesized image; Right: real image.

Figure 13. Testing image pair 6. Using the Chinese model to predict Japanese data. Left: input label; Middle: synthesized image; Right: real image.

Figure 14 and figure 15 show the image pairs deserved by using Model 2 to generate the Chinese apartment. The synthesized image in figure 14 performs substantially. It can not only provide a clear structure of the apartment but is also very similar to the original input image. Besides, the synthesized image in figure 15 shows a different but reasonable structure. The pattern is blurry but recognizable by architects. And the drawing style in both image pairs is highly uniform, indicating a perfect match between the training and testing images.

Figure 14. Testing image pair 7. Using the Chinese model to predict Chinese data. Left: input label; Middle: synthesized image; Right: real image.
Therefore, the model trained with the Chinese dataset performs poorly with the Japanese apartment boundaries but nicely with Chinese apartment boundaries. This phenomenon shows the design strategies for Chinese apartments are more uniform.

3.3. MODEL COMPARISON

Figure 16 shows the Japanese apartment generated by both models. The layout in the synthesized image by the Japanese model is clear to tell the interior design, including the positions of the living room and the bedroom. However, the synthesized image by the Chinese model is not ideal. It is hard to recognize the bedroom pattern. This synthesized image has worse performance than the real image. And the predicted layout does not make sense to the designer. Thus, in terms of generating Japanese apartments, the Japanese model works better than the Chinese model.

Figure 17 shows the image pairs deserved by using both models to generate the Chinese apartment. The synthesized image using the Chinese model performs better than that using the Japanese model. It makes a precise prediction in both the layout design and drawing patterns. However, the synthesized image using the Japanese model generates a clear pattern of the apartment, but it does poorly in providing the internal furnishings and the application of each room.
Therefore, the design style, as well as the drawing style in the two datasets, are different and unique. The models perform well when feeding in similar input images as the training images.

4. Conclusion and Discussion

In conclusion, Generative Adversarial Networks successfully learn and generate apartment floor plans only based on the design boundaries. By training the network with plan drawings from different design styles, the neural network determines various features; thus, it generates plans with varying patterns of design and layouts. The quality and the unity of the patterns in the training images directly influence the resolution of the generated images. To achieve a better result, the training dataset should have uniform drawing styles as well as simple design rules.

In the future, the cooperation between Artificial Intelligence and Architectural Design will become more and more frequent, thus resulting in the birth of a new design concept - Architectural Intelligence, in which the computer not only aids but also decides the design.

References