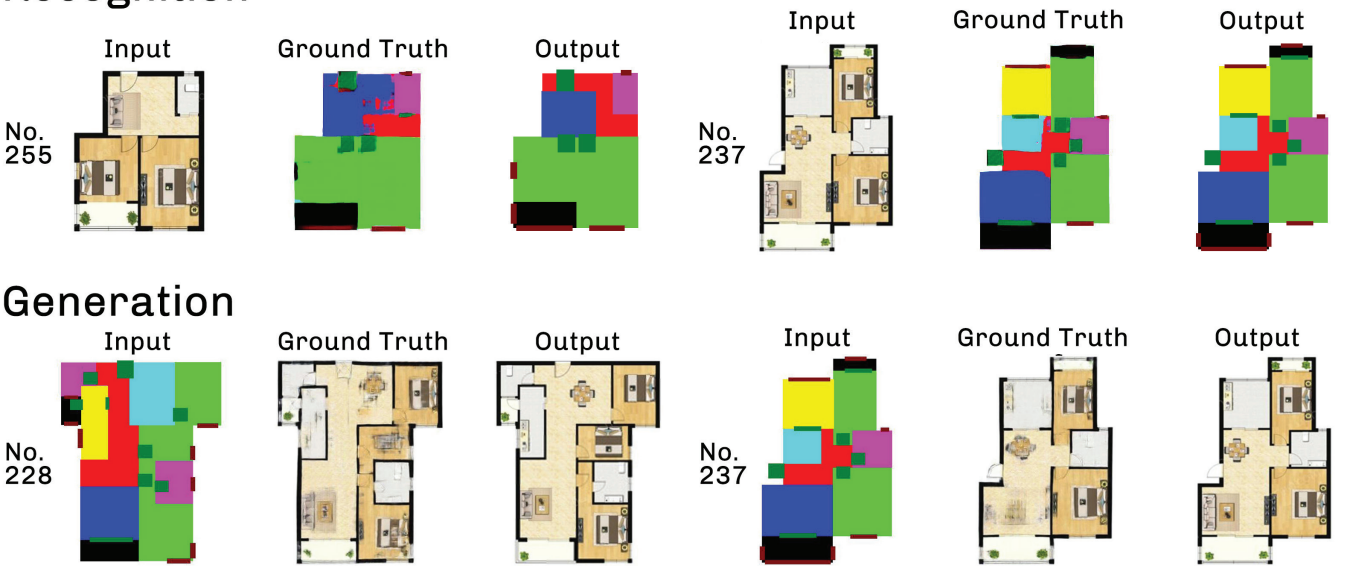
Architectural Drawings Recognition and Generation through Machine Learning

Weixin Huang
School of Architecture,
Tsinghua University

Hao Zheng
School of Design,
University of Pennsylvania

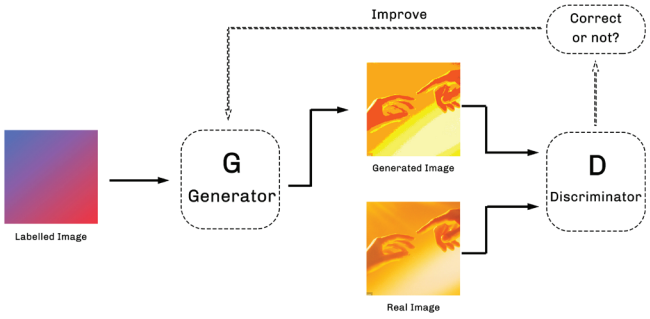


1

ABSTRACT

With the development of information technology, the ideas of programming and mass calculation were introduced into the design field, resulting in the growth of computer-aided design. With the idea of designing by data, we began to manipulate data directly, and interpret data through design works. Machine Learning as a decision making tool has been widely used in many fields. It can be used to analyze large amounts of data and predict future changes. Generative Adversarial Network (GAN) is a model framework in machine learning. It's specially designed to learn and generate output data with similar or identical characteristics. Pix2pixHD is a modified version of GAN that learns image data in pairs and generates new images based on the input. The author applied pix2pixHD in recognizing and generating architectural drawings, marking rooms with different colors and then generating apartment plans through two convolutional neural networks. Next, in order to understand how these networks work, the author analyzed their framework, and provided an explanation of the three working principles of the networks, convolution layer, residual network layer and deconvolution layer. Lastly, in order to visualize the networks in architectural drawings, the author derived data from different layer and different training epochs, and visualized the findings as gray scale images. It was found that the features of the architectural plan drawings have been gradually learned and stored as parameters in the networks. As the networks get deeper and the training epoch increases, the features in the graph become more concise and clearer. This phenomenon may be inspiring in understanding the designing behavior of humans.

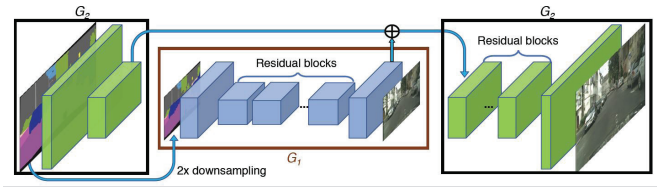
1 Apartment floor plan: recognition and generation through generative adversarial network



2

2 Workflow of GAN.

3



4

3 Pix2Pix examples by Isola et al. (2017).

4 Network architecture of Pix2PixHD by Wang et al. (2017).

RECENT DEVELOPMENT OF GENERATIVE ADVERSARIAL NETWORK (GAN)

In the past four years, Generative Adversarial Networks (GAN), as one type of machine learning algorithm, has achieved a lot of progress for generative tasks. Although there were many problems when GAN was first proposed, such as unstable training and so on, researchers improved it from the aspects of the framework, training techniques and so on, resulting in the following explosive growth.

Goodfellow et al. (2014) were known as the first team to propose the Generative Adversarial Network in machine learning. By providing training data in pairs, the program finds the most suitable parameters in the network so that the discriminator (D) has the least potential to distinguish the generated data (G) from the original data (Figure 2).

In order to solve the problem of training in the wrong direction, Mirza and Osindero (2014) proposed a refined version of GAN, Conditional Generative Adversarial Networks (CGAN). The idea of CGAN is to turn the original generation process into a conditional process, based on providing some additional information as hints. The additional information can be one-hot vectors, two-dimensional images, or even three-dimensional models. Once the training process runs toward the unexpected direction, punishment will be given to correct the training tendency according to the additional information.

Later, Zhu et al. (2016) invented the iGAN/GVM. Their work contains two kinds of additional information, one is the user's input, such as strokes, lines, stretches, and deformation, second is the boundary of objects in the image. In addition to outputting two-dimensional data (images), the program will also stick the texture in the original images to the shape of the resulting images, making the output images more real and clearer near the boundary. They used the light field information to capture the point-to-point mapping relationships, so that it is possible to repeatedly paste the textures to the output images.

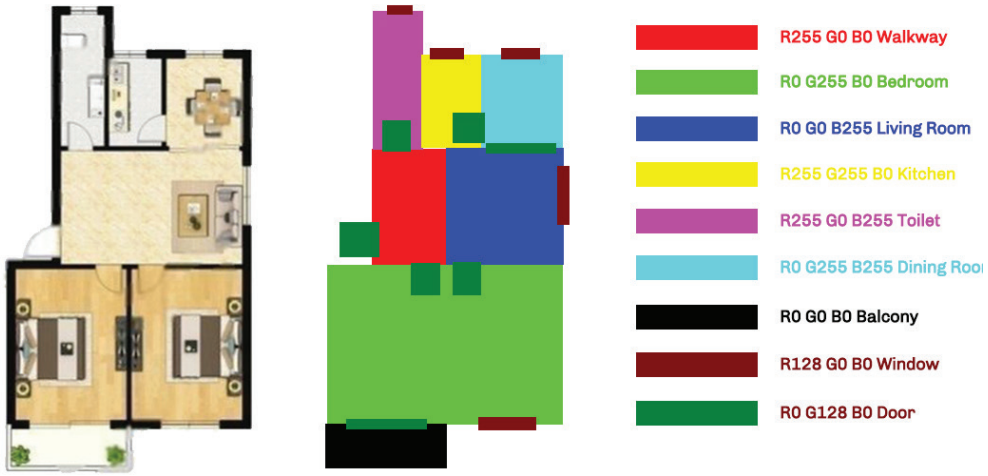
After the creation of iGAN, Isola et al. (2017) continued to work on pix2pix by generating a real photo from a partly-damaged photo, a colorful map from a black-and-white map, and an image with texture and shadow from a linear sketch. In pix2pix, the input D is a pair of images rather than a single image, and the task of D becomes the evaluation of whether those two images are the same or not. So after training, we can input an image and tell the program to generate the most possible corresponding output image (Figure 3).

Based on pix2pix, Wang et al. (2017) built a refined network called pix2pixHD, enlarging the resolution of the images into 2048*1024 instead of the previous 256*256. An input image is regarded as three parallel two-dimensional matrices, according to the width, height, and three RGB channels of the image. Then the matrices are transformed in the generator through five groups of convolution layers, then nine groups of residual network layers, and finally five groups of deconvolution layers (Figure 4).

For now, pix2pixHD is the latest and most efficient framework to process image data in pairs (Wang et al. 2017). Its ability to process large images also gives us more details when generating complex architectural drawings. The following research in this article is based on pix2pixHD framework to discuss the mapping between architectural plan drawings, which is different from previous research of mapping city images (Zheng 2018), mapping perspective images (Peng, Zhang, and Nagakura 2017), and mapping structural images (Luo, Wang, and Xu 2018).

RECOGNIZING AND GENERATING THROUGH GAN

Since GAN is a powerful tool in dealing with image data, its application in architecture, especially in recognizing and



5 Left: floor plan drawing. Middle: labeled image. Right: labeling rule.

5

generating architectural drawings, has good potential for development. A process of training and evaluating between an architectural drawing and its corresponding labeled map was carried out by the author in Python and Pytorch. In addition, to simplify the study, only a dataset of colorful floor plans of apartments collected from property website lianjia.com was tested in order to remove the influence of varying scales and styles of the drawings.

Labeling Principles

First of all, a labeling rule was created which uses different colors to represent areas with different functions (Figure 5). Colors with RGB values of only 0 or 255 were commonly used in the labeling map in order to differentiate the labels as far as possible, so all together 8 combinations of RGB values can be achieved, which are used to label walkway, bedroom, living room, kitchen, toilet, dining room, balcony, and blank areas outside the flat. Windows and doors are less important, so R:128 G:0 B:0 is used for windows and R:0 G:128 B:0 is used for doors. Since windows and doors are the connections of the other areas, their drawing layer is always on the top of the others.

One hundred fifteen image pairs such as Figure 5 were selected, sized to a fixed plotting scale, and carefully marked by three volunteer architectural students. Based on this dataset, two trainings, plan-to-map (recognizing plan drawings and producing color labeled maps) and map-to-plan (inputting color labeled maps and generating plan drawings), were tested and will be introduced in the following pages.

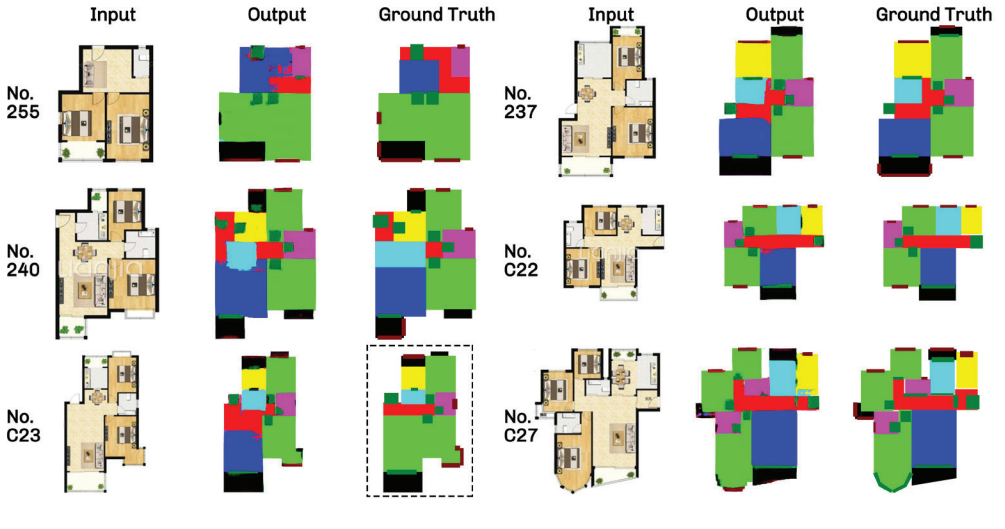
Recognizing

After dividing 115 images into a training set with 100 images and a testing set with 15 images, our team first trained the network using plan drawings as input and color labeled maps as output. The program is supposed to take in a plan drawing and recognize it by producing a map with different colors that represent different functional areas. The whole training process was carried out with one NVIDIA TITAN X graphics card, and it took 80 seconds for one epoch with 100 images, so totally 1.8 hours for one network.

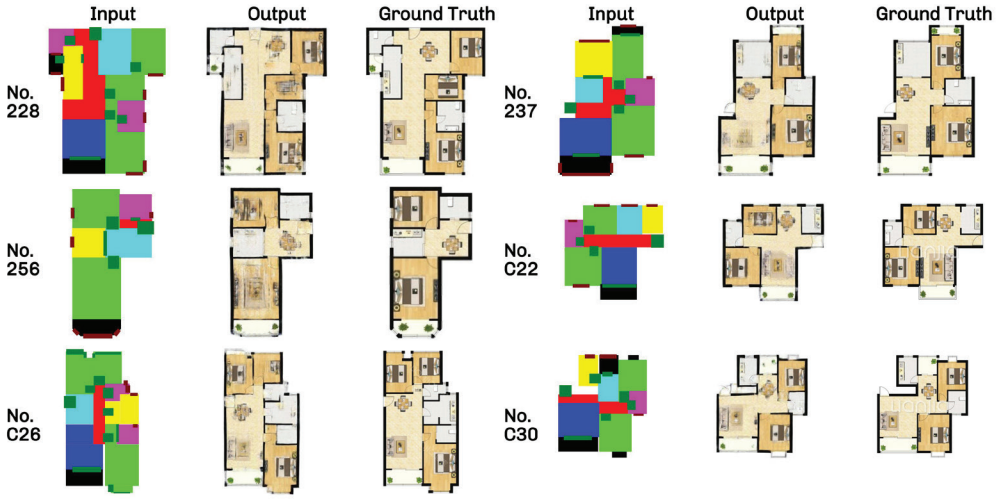
Figure 6 shows the selected results from the testing set. It performs well in recognizing areas of bedroom, kitchen, toilet, and balcony, whose boundaries are clear because there are usually walls to separate them from each other and specific furniture inside, as in No.237 and No.C22. However, for walkway, living room, and dining room, the network may not be able to tell them apart since there is usually no clear boundary between them, as in No.255 where the areas of walkway and living room are mixed together. But actually, a test asking multiple architects to mark No.255 showed different results between the areas of walkway and living room, so it's also hard for humans to distinguish these two areas in No.255. Also in No.240 the boundary between dining room, walkway, and living room is not clear, and different architects may give different answers based on their own understanding. This uncertainty somehow reflects the similarity in human cognition and machine learning results.

In No.C27, the shape of this floor plan contains an ellipse and a triangle, but most floor plans in the training set are orthogonal. As a result, the prediction does not perfectly match the original plan. Adding more images with irregular boundary shapes into the training set may help to solve this problem.

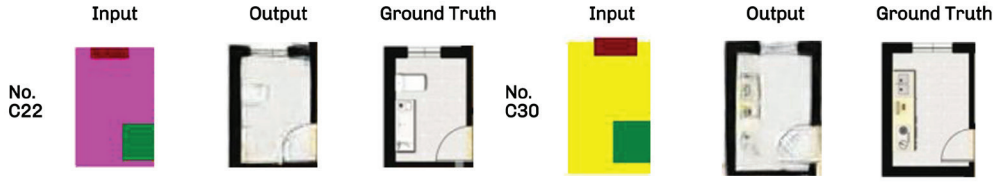
It is also interesting to see that in No.C23, an error from our volunteer was found by the trained network. The living room and parts of a walkway are labeled as a bedroom by the volunteer, but the network successfully recognized this area. The performance of the network even exceeds that of a human in some images. Later, we double checked all images and found four wrongly-marked image pairs in



6



7



8

6 Result of recognizing.

7 Result of generating.

8 Detailed generating result of toilet and kitchen.

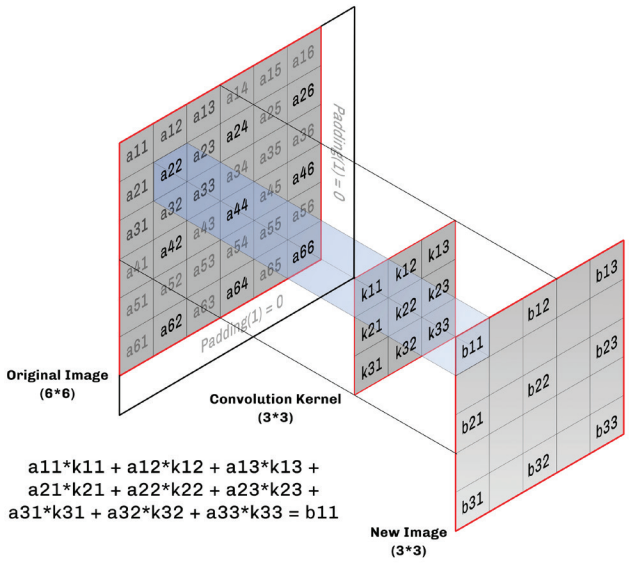
the training set, but those errors didn't lead the training process in the wrong direction, which demonstrates the fault-tolerant ability of the network.

In conclusion, the network works well in recognizing architectural plan drawings. Compared to the training set of thousands of images commonly used in other research, a training set with 100 images is enough for the network to learn and summarize the knowledge of architectural plan drawings of specific apartments.

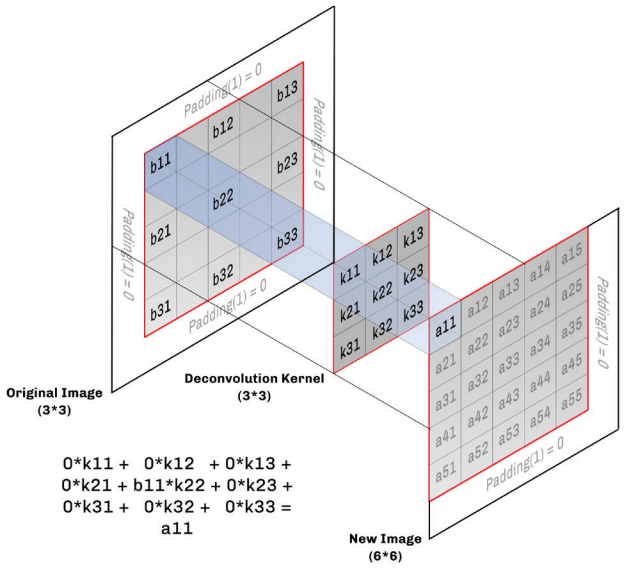
Generating

Next, instead of regarding plan drawings as input images, our team then trained another network using color labeled maps as input images and plan drawings as output images. When evaluating, the program should generate a plan drawing according to the input labeled map.

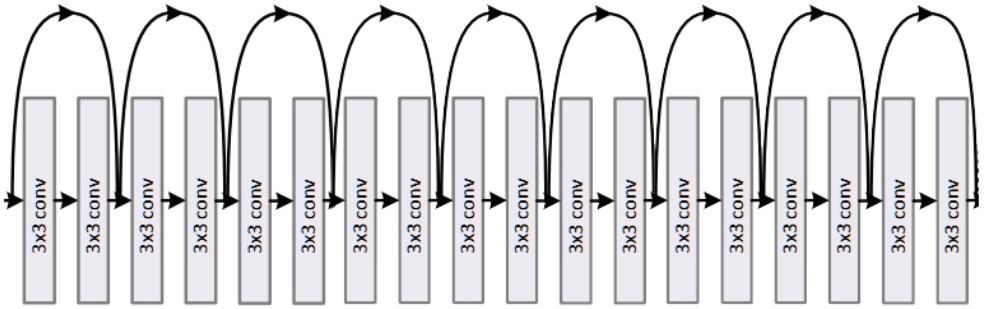
Figure 7 shows selected results from the testing set. All six selected images show clear generation of the kitchen and toilet areas, including accurate positions of kitchenware and sanitary ware and correct direction of door openings (Figure 8). The high quality of these results is not surprising since there is not much uncertainty in the positioning of fixtures and doors in the training set.



9



11



10

9 Convolution layer and kernel.

10 Nine groups of ResNet.

11 Deconvolution layer and kernel.

However, when generating the area of living room, the positions of the sofa and the TV are not always clear, as in the output of No.237 and No.C22, since facing either right or left seems reasonable. But in No.C26 and No.C30, facing the other direction is impossible because of the existence of a door and walkway, so the positions of the sofa and TV are very clear in these two output images. Similar results happen in the bedroom of No.228 and No.256, which are also reasonable.

Another point is that the position of the dining table in No.228 and No.C26 is slightly different from that in the original images. But a survey shows that more architects thought the generated position was more reasonable because it leaves more space for the walkway and door. This somehow shows the reliability of the network in design.

In conclusion, the network has the potential to learn the rules of design effectively. Both the very certain rules that a design needs to follow and the uncertain situations that provide flexibility can be reflected by the network. Architects can release their hands from simple or even complex design work by inputting labeling information to the program and getting detailed design plans as feedback.

WORKING PRINCIPLES

Based on the dataset and experiments above, in the following two sections, the working principles and core algorithms in the generator of GAN are explored, from the whole framework to certain neurons.

Convolution Layer

Image data is actually a combination of three two-dimensional matrices which represent the RGB channels of pixels in the image. When the network takes in the image data, the matrices will go through a series of calculations and finally come out as a new set of matrices. We call each set of calculations layer, and each single calculation neuron.

The first section of layers includes five groups of convolution layer sets; each contains one convolution layer, one batch normalization layer, and one ReLU layer. As Figure 9 shows, the original image will be multiplied by a convolution kernel matrix in the convolution layer, and become a new matrix. This operation will be carried out every two pixels, so the size of the new matrix is half of the original image in

width and height. This calculation enables the combination of information in neighbor pixels, and further summarization of the information in the image. Usually, a convolution layer contains multiple convolution kernels, and each kernel produces a new matrix. All new matrices arrange in a line, resulting in a three-dimensional matrix, which is the final outcome of a convolution layer.

Then, one batch normalization layer and one ReLU layer will act as a data coordinator to normalize the numbers and produce the activation matrices for the next layer.

After the computation of all five groups of layers, the size of the image will be greatly reduced to width/16 * height/16, with 1024 layers of information, so the final size of the data will be 16 * 16 * 1024. All information in the original image is summarized and stored in the new three-dimensional matrix, ready for the next group of calculations.

Residual Network Layer

The second part in the network is nine groups of residual network layers (ResNet). One ResNet contains two sets of convolution layers, but instead of directly linking convolution layers, ResNet has a back door to skip two layers if the result is growing worse (Figure 10). It processes the network into deeper layers, while making sure the overfitting problem does not occur.

Deconvolution Layer

Compared to the effect of a convolution layer to make the image smaller, the deconvolution layer is a reversed operation, enlarging the image back to the original size, while reducing the number of two-dimensional matrices.

Figure 11 illustrates the computation principles of a deconvolution layer. The source pixels are arranged separately, and the same rule of multiplication is applied to the kernel. Each deconvolution layer will make the matrices double the size in height and width, but reduce the number of matrices. After going through five deconvolution layers, the data with the size of width/16 * height/16 * 1024 will be a size of width * height * 3, same as the original image.

In the generator network, image data will be folded into a smaller image with many layers of information, then be unfolded back to another image of the same size but with only 3 channels of color information. In the network, thousands of kernels work together to summarize and then explain the features, which are the main parameters that the network should learn from the training set.

VISUALIZING THE NETWORK

After understanding how the network trains and processes image data, our team visualized each matrix in the whole network to see what kinds of visual features the network has recognized and generated.

Network for Recognition

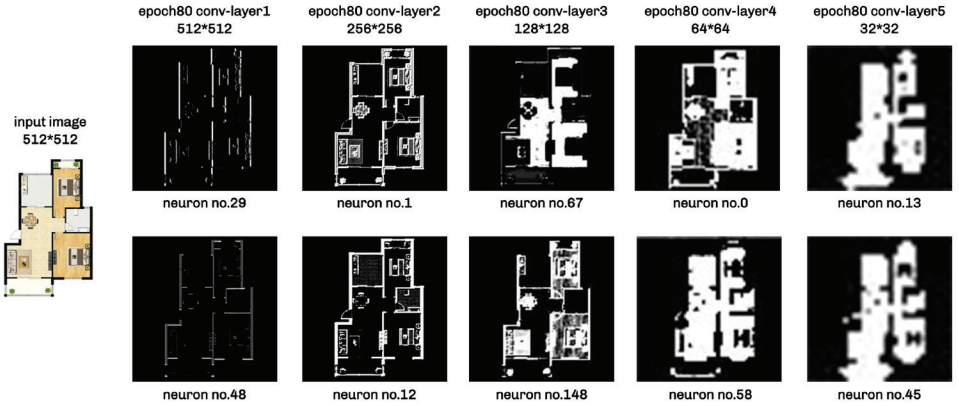
In order to activate the kernels in the network, a testing image was inputted. Then a series of black-and-white images was translated from the two-dimensional matrices, which were the result of the original image after passing through each layer. The pixel with more extreme values (0 or 255) means it's more activated into two groups.

Figure 12 and Figure 13 show the selected translated images from the recognition network. The network was trained 80 times (epochs), and its loss value reached a relatively low and stable number, so we thought this training process was completed.

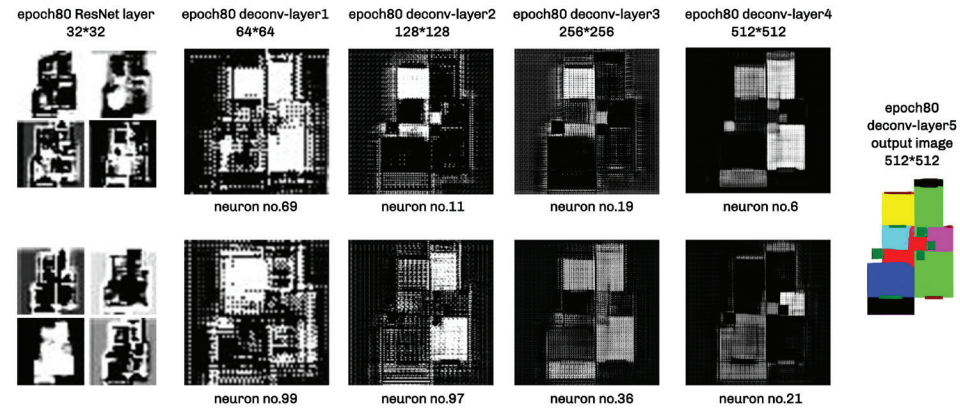
In Figure 12, as the convolution layer (conv-layer) goes deeper, the activation of the image becomes more and more conspicuous, and more and more features are activated. Neuron No.29 in conv-layer 1 indicates that only features like vertical walls are detected, but more features like the edges of tables and beds are activated in Neuron No.1 in conv-layer 2. What's more, Neuron No.67 in conv-layer 3 shows the paving pattern of bedrooms and living room is detected, and in Neuron No.0 in conv-layer 4, the features of the paving, walls, and furniture edges can be activated together. In the last conv-layer, it seems all features are summarized and combined into one matrix. So the aim of this convolution process is to condense and re-encode the information and features in the original image, and to prepare the data for the later deconvolution layers. This is more like the learning process of humans, from concrete entities to abstract concepts as we think deeper.

Figure 13 shows the translated images in ResNet layers and deconvolution layers. The matrices don't change much in ResNet layers because of the protection mechanism of the overfitting problem. The author tried to shut down the back door in ResNet, but this caused the vanishing gradient problem as a result when back propagating. Here, the ResNet is necessary although it takes some computation.

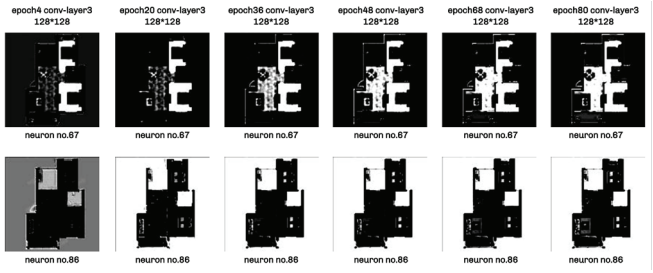
Next comes five groups of deconvolution layers (deconv-layer). In the recognition network, the final aim is to map the floor plans to the color labeled maps, whose colors are usually continuous and compact. Neuron No.69 in deconv-layer 1 shows the chaos situation when the computation of matrices in the first deconvolution layer completes. But as



12



13



14

12 Translated images in convolution layer for recognition network.

13 Translated images in ResNet and deconvolution layer for recognition network.

14 Translated images of selected neurons in different training epochs for recognition network.

Table 1 Possibilities that two areas are activated together for recognition network.

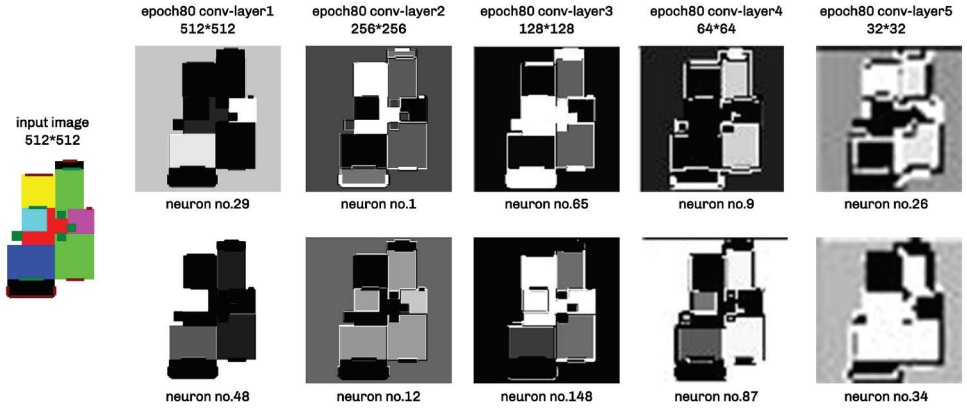
	kitchen	upper balcony	upper bedroom	dining room	walkway	toilet	living room	lower balcony	lower bedroom
kitchen	1	0.45	0.38	0.27	0.31	0.28	0.25	0.36	0.39
upper balcony	0.45	1	0.59	0.48	0.62	0.5	0.73	1	0.48
upper bedroom	0.38	0.59	1	0.39	0.5	0.39	0.52	0.52	1
dining room	0.27	0.48	0.39	1	0.42	0.39	0.47	0.45	0.34
walkway	0.31	0.62	0.5	0.42	1	0.52	0.53	0.58	0.42
toilet	0.28	0.5	0.39	0.39	0.52	1	0.53	0.5	0.25
living room	0.25	0.73	0.52	0.47	0.53	0.53	1	0.73	0.44
lower balcony	0.36	1	0.52	0.47	0.58	0.5	0.73	1	0.47
lower bedroom	0.39	0.48	1	0.34	0.42	0.25	0.44	0.47	1

Table 1

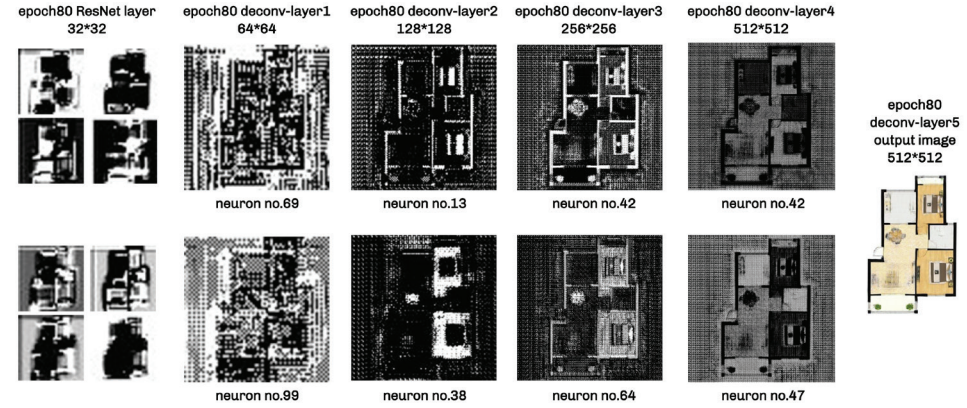
the network goes deeper, the boundaries between different areas become clearer, and the noise gradually disappears. Finally, a clean map showing the prediction of different areas comes out as the output of the last deconvolution layer.

The gradual change of a specific neuron in different training epochs can also be illustrated (Figure 14). For a total of 80 training epochs, samples in epochs 4, 20, 36, 48, 68, and 80 in conv-layer 3 are selected. For example, Neuron No.67 in epoch 4 shows a clear activation of the paving pattern in bedroom area. With the training going on, the activation of the paving pattern in the living room also becomes clearer. In the final training of epoch 80, the image of Neuron No.67 shows an equivalent weight of both the paving patterns, which are reasonable because the positions of the two areas may have some connections. We can understand this as a procedure of learning like humans, as the more experience accumulated in the learning process, the better the knowledge will match the reality. It is easier to understand the effect of training epochs on performance by referring to the human learning process.

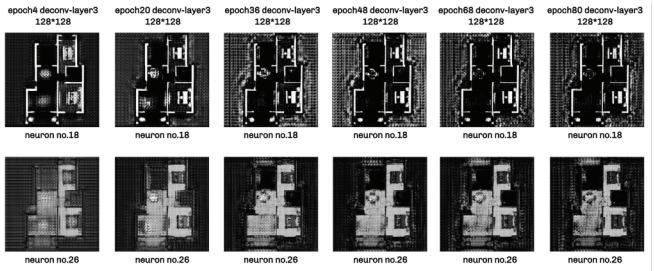
What's more, through the analysis of the last deconv-layer, a table was produced showing which two areas have a greater possibility to be activated together (Table 1). Numbers greater than 0.7 are highlighted. It shows a larger possibility for areas with the same types of functions to be activated together, such as the upper and lower balcony, and the upper and lower bedroom. This is because of the detection of similar patterns in the network. However, the possibility of living room and balcony being activated together is also comparatively high. Since there is no similarity in pattern between these two kinds of areas, it could be said that the design of living rooms and balconies, such



15



16



17

15 Translated images in convolution layer for generation network.

16 Translated images in ResNet and deconvolution layer for generation network.

17 Translated images of selected neurons in different training epochs for generation network.

Table 2 Possibilities that two areas are activated together for generation network.

	kitchen	upper balcony	upper bedroom	dining room	walkway	toilet	living room	lower balcony	lower bedroom
kitchen	1	0.33	0.38	0.22	0.56	0.38	0.14	0.33	0.38
upper balcony	0.33	1	0.48	0.36	0.47	0.39	0.49	1	0.48
upper bedroom	0.38	0.48	1	0.44	0.25	0.09	0.31	0.48	1
dining room	0.22	0.36	0.44	1	0.09	0.27	0.48	0.36	0.44
walkway	0.56	0.47	0.25	0.09	1	0.55	0.28	0.47	0.25
toilet	0.38	0.39	0.09	0.27	0.55	1	0.45	0.39	0.09
living room	0.14	0.49	0.31	0.48	0.28	0.45	1	0.49	0.31
lower balcony	0.33	1	0.48	0.36	0.47	0.39	0.49	1	0.48
lower bedroom	0.38	0.48	1	0.44	0.25	0.09	0.31	0.48	1

Table 2

as their positions, may be highly related.

In conclusion, by visualizing the recognition network, certain similarities are found in the learning method and process between the GAN machine learning algorithm and human cognition. It might be interesting to dig it deeper in other types of data to the relationship of machine learning and human cognition in architectural design problems.

Network for Generation

Next, same test was done for the generation network.

Figure 15 shows the translated images in conv-layer. In generation networks, the features of input images are easier to recognize and understand compared to the recognition network. The edges and differences between each area are quite clear, and almost no noise exists in the conv-layers. The features being activated change from simple color blocks to the combination of color blocks and their boundary lines, which further supports the former conclusion.

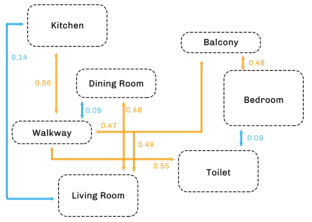
However, after the computation in the ResNet, the same thing happens in the deconv-layer 1, images are in a chaotic situation with much noise (Figure 16). As the network goes deeper, the noise gradually disappears and the true generation of the floor plan begins to show up. As shown in Neuron No.42 of deconv-layer 3, distinguishable edges of walls and furniture are activated, which means the generation network acted properly in deconv-layers.

The same test of translating images from different training epochs was done for the generation network (Figure 17). But here we thought it was more valuable to activate the

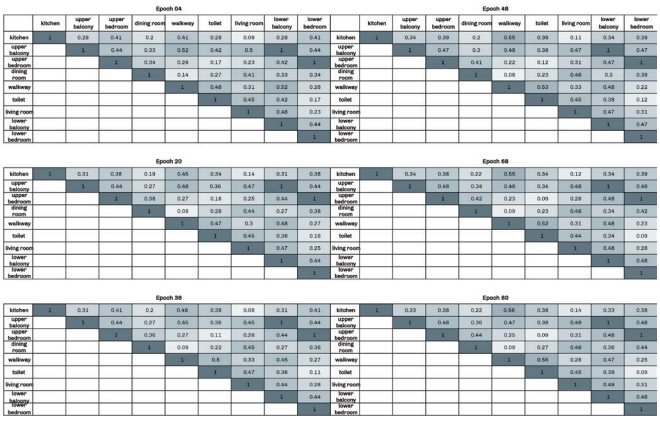
18 Summary of highlighted activation possibilities.

19 Possibility tables in different training epochs.

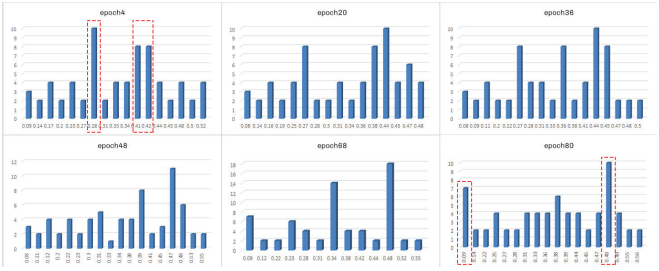
20 Possibility distribution in different training epochs.



18



19



20

deconv-layer because the deconv-layer is more complex and contains more unique information than the conv-layer in the generation network. The same phenomenon in the generation network was found in the translated images of different training epochs as in the recognition network. Neuron No.18 in epoch 4 shows a very blurry generation of the living room and the dining room. But when it proceeds to epoch 36, the dining table becomes clear, and in epoch 48, the blurry area in living room disappears, which means this neuron regards the generation of living room as an unrelated factor to that of other areas, and excludes its weight. In epoch 80, however, the former highly activated dining area becomes less activated, because of the same reason. This shows the ability of the network in self-correcting and evolving as the training time increases.

As shown in table 2, while the two balcony areas and bedroom areas still keep a very large possibility (100%) to be activated together, the possibility of dining room and walkway appears very small. This represents the ability of the network to distinguish these two controversial areas. Meanwhile, the possibilities of the toilet and two bedrooms are both very small, this indicates the preference of designing toilets and bedrooms close together in the dataset, so the network adjusted its parameters to distinguish them apart in a very early stage.

Figure 18 shows the summary of large or small activation possibilities. Walkway and living room are the core components in this graph, since they have links to most of the other areas. This might reveal the designing sequences of apartment plans, in which the walkway and living room come first, then other rooms.

Figure 19 shows the evolution of possibility tables in different training epochs. Since the matrices are symmetrical, only the upper triangle is illustrated. Generally speaking, with the training going on, the possibilities turn from even numbers to more extreme numbers (Figure 20), which indicates the improvement during the training process. In epoch 4, most numbers are distributed within 0.28 to 0.42, but in epoch 80, the number distribution in 0.09 and 0.48 increases a lot, this may indicate that during the training process, the network gradually learns to tell areas apart or combine them together.

To be specific, the possibility of toilet and bedrooms is 0.17 in epoch 4, and gradually decreases to 0.09 in epoch 80, while that of kitchen and walkway increases from 0.41 in epoch 4 to 0.56 in epoch 80. This demonstrates the learning process of understanding the relationships between different areas that occurs in the generation network. .

On the other hand, it could be see that many of the co-activation possibilities are not changing significantly, which may indicate that the knowledge of apartment plan design and the training process is more complex than what can be revealed by the statistical relationship between spaces, and requires more in-depth exploration in the future.

CONCLUSION

Pix2pixHD, an application of Generative Adversarial Networks (GAN) is tentatively applied in recognizing and generating architectural drawings. The experiments are successful, and can be further developed into prototypes of a powerful tools for drawing review, digitalization, and drawing assistance. Also, by understanding the working principles and visualizing sample networks, designers can verify and summarize their design techniques and ideas, and get further inspiration through this process.

Analysis of the recognizing and generating process, as well as the training process of the GAN has been tentatively

carried out, revealing some interesting phenomena. Because of the complexity of neural networks, it is believed that there will be more in-depth associations that lie in the network, which will provide a valuable understanding of architectural floor plan design. Further in-depth studies could be carried out to explore the mechanism that lies in the network.

Through the analysis of the process of network training and information processing, it is interesting to find that, compared to a human learning process, a machine learning algorithm has similar characteristics, such as to dig abstract concepts from concrete entities, and to extract accurate standards from blurry understandings.

It could be seen that in the future, artificial intelligence may play more and more active roles in not only repetitive works, but also creative works. It is highly possible that human ability would be greatly expanded when combined with artificial intelligence. The next step of this research would be to develop networks to recognize and generate architectural drawings faster and more reliably, which could be applied in releasing architects from repetitive works, and enhance exploration of creative design solutions.

ACKNOWLEDGEMENTS

This paper is the continuing research of 'Understanding and Visualizing Generative Adversarial Networks in Architectural Drawings' by the authors. The previous article was published as a short paper in CAADRIA 2018.

REFERENCES

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Nets." In *Advances in Neural Information Processing Systems 27*. Montreal, QC: NIPS.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. "Image-to-image Translation with Conditional Adversarial Networks." arXiv preprint. arXiv:1611.07004

Luo, Dan, Jinsong Wang, and Weiguo Xu. 2018. "Robotic Automatic Generation of Performance Model for Non-Uniform Limear Material via Deep Learning." In *Learning, Prototyping and Adapting, Proceedings of the 23rd International Conference on Computer-Aided Architectural Design Research in Asia.* Beijing: CAADRIA.

Mirza, Mehdi, and Simon Osindero. 2014. "Conditional Generative Adversarial Nets." arXiv preprint. arXiv:1411.1784.

Peng, Wenzhe, Fan Zhang, and Takehiko Nagakura. 2017. "Machines' Perception of Space: Employing 3D Isovist Methods and a Convolutional Neural Network in Architectural Space Classification." In *Disciplines & Disruption: Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture*, edited by Takehiko Nagakura, Skylar Tibbits, Mariana Ibanez, and Caitlin Mueller, 474–81. Cambridge, MA: ACADIA.

Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2017. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs." arXiv preprint. arXiv:1711.11585.

Zheng, Hao. 2018. "Drawing with Bots Human-computer Collaborative Drawing Experiments." In *Learning, Prototyping and Adapting, Proceedings of the 23rd International Conference on Computer-Aided Architectural Design Research in Asia.* Beijing: CAADRIA.

Zhu, Jun-Yan, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. "Generative Visual Manipulation on the Natural Image Manifold." In *Proceedings of the 14th European Conference on Computer Vision*, part V, edited by Bastian Liebe, Jiri Matas, Nicu Sebe, and Max Welling, 597–613. Amsterdam: ECCV.

IMAGE CREDITS

Figure 3: © Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017.

Figure 4: © Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2017.

All other drawings and images by the authors.

Weixin Huang is an Associate Professor in the School of Architecture, Tsinghua University. He is the Associate Director of the Digital Architectural Technology Education Committee of China, a committee member of Computer Aided Architectural Design Research in Asia (CAADRIA), and one of the founders of Digital Architectural Design Association (DADA) of the Architectural Society of China (ASC). He received his Ph.D. from Kyoto University, Japan. His research focuses on digital design of architectural & structural integrated systems, big data spatio-temporal behavior analysis, and design cognition.

Hao Zheng is currently a Ph.D. student at the University of Pennsylvania, School of Design. He is a programmer and design researcher, specializing in machine learning, robotic technology, mixed reality, and generative design. He holds a Master of Architecture degree from the University of California, Berkeley, and a Bachelor of Architecture degree from Shanghai Jiao Tong University. Before joining UPenn, Hao worked as a research assistant at Tsinghua University with a concentration on robotic assembly and machine learning and at UC Berkeley with a concentration on bio-material 3D printing and deep learning.