technologies

MDPI

Article

3D Model Generation on Architectural Plan and Section Training through Machine Learning

Hang Zhang

Stuart Weitzman School of Design, University of Pennsylvania, Philadelphia, PA 19104, USA; kv333q@upenn.edu

Received: 30 September 2019; Accepted: 6 November 2019; Published: 15 November 2019

Abstract: Machine learning, especially the GAN (Generative Adversarial Network) model, has been developed tremendously in recent years. Since the NVIDIA Machine Learning group presented the StyleGAN in December 2018, it has become a new way for designers to make machines learn different or similar types of architectural photos, drawings, and renderings, then generate (a) similar fake images, (b) style-mixing images, and (c) truncation trick images. The author both collected and created input image data, and specially made architectural plan and section drawing inputs with a clear design purpose, then applied StyleGAN to train specific networks on these datasets. With the training process, we could look into the deep relationship between these input architectural plans or sections, then generate serialized transformation images (truncation trick images) to form the 3D (three-dimensional) model with a decent resolution (up to 1024 × 1024 × 1024 pixels). Though the results of the 3D model generation are difficult to use directly in 3D spatial modeling, these unexpected 3D forms still could inspire new design methods and greater possibilities of architectural plan and section design.

Keywords: machine learning; StyleGAN; style transfer; architectural design; 3D

1. Introduction

With the development of computer science, the emerging computational systems and environments have gradually played a more active role in aiding the creating and designing process [1], leading to the rise of CAAD (Computer-Aided Architectural Design), which was pioneered by leading architects such as Frank Gehry, Zaha Hadid, Greg Lynn, and Peter Eisenman [2].

The development of emerging computational technology brings new application possibilities in the field of architectural design. As one of the cutting-edge technologies in recent years, machine learning as a data analysis and prediction tool has been widely used in many fields, For example, Rafał Weron (2014) [3] and Cincotti et al. (2014) [4] introduced comparisons between multiple methods (traditional, neural networks, etc.) applied to electricity price forecasting, demonstrating the powerful and interesting side of machine learning in data processing. It has also caught the attention of the architects, resulting in their exploration of methods to apply machine learning to architectural design.

With regard to the architectural drawings' generation through machine learning, Huang et al. (2018) presented their results of applying pix2pixHD (High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs (Generalized Adversarial Networks)) in recognizing and generating architectural drawings, marking rooms with different colors, and then generating apartment plans through two CNNs (Convolutional Neural Networks) [5]. Campo et al. (2019) applied neural-style transfer to architectural drawings in order to explore the transformational potential of artificial intelligence and automation with respect to architecture [6]. Koh et al. (2019) used a probabilistic machine learning model trained with tile-maps to study the potential of an inference design system for architectural design [7].

These research works demonstrated that machine learning has great potential for learning and generating architectural drawings; however, all of them only focused on the application of machine learning to architectural design at the two-dimensional level, without exploring three-dimensional architectural spatial design. Steinfeld et al. (2019) recently presented their framework for an application of machine learning, a neural net trained to perform image classification, to generative architectural design by transforming the three-dimensional model data into two-dimensional multiview data [8]. However, as their paper indicated, there is still a limitation with certain practical constraints on image resolution and computing time [8], and the exploration of higher detail three-dimensional form-finding through machine learning with higher resolution remains to be done.

Based on these previous studies of architectural drawing generation through machine learning, this article mainly introduces a method of three-dimensional architectural form generation with high resolution (up to 1024 × 1024 × 1024 pixels, due to the hardware limitation, the author made up to the maximum 512 × 512 × 1024 pixels with one RTX 2080ti) by training two-dimensional architectural plan or section drawings through StyleGAN, learning the relationship between these inputs to simulate serialized transformation images which finally compose the whole three-dimensional model.

2. Materials and Methods

Since team Goodfellow proposed the principle of GAN (Generative Adversarial Network) in 2014 [9], a series of improved new GANs keep coming out during the past several years. The original GAN model contains two networks: generator (G) and discriminator (D), one to learn from the pattern of input data and the other one to distinguish the generated result from the real input images. However, the original GAN has two limitations: (1) No user control—it simply inputs a random noise, then outputs a random image. (2) The results are low-resolution and low-quality.

So far, StyleGAN is one of the most emerging and powerful GAN in two-dimensional image learning and generation of machine learning. In order to illustrate why the author specifically uses StyleGAN as machine learning process for architectural drawings training, there are several remarkable versions of improved GANs listed below to be compared their strengths and weaknesses, and Figure 1 illustrates the main logical process of each selected GANs:

(1) CGAN (2014): Conditional GAN is the first attempt to add additional information on both generator and discriminator [10]. This information could be labels tagged on original input data, resulting in images with better quality and the controllability of what the output images will look like.

(2) DCGAN (2015): Deep convolutional generative adversarial networks are the first GANs which were a major improvement on the GAN architecture, it introduced a more stable training process and higher samples of quality [11].

(3) InfoGAN (2016): Information Maximizing Generative Adversarial Nets made innovation on the basis of CGAN [12]. With the combination between information-theory and GAN, InfoGAN is able to encode meaningful image features in part of the noise vector Z such as the width of the output digit.

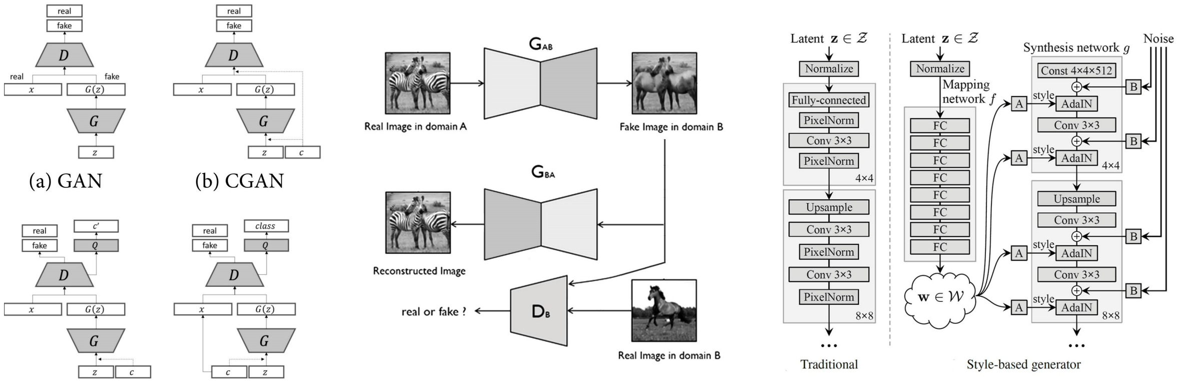
(4) ACGAN (2016): Compared with CGAN, the Auxiliary Classifier GAN added the additional information not on the discriminator (D) but to the input data, helping generator (G) to understand what the sample structure of this class is [13]. It also added an auxiliary classifier in the output section of discriminator(D) to improve the performance of conditional GAN.

(5) WGAN (2017): Wasserstein GAN mainly solved the convergence problems. By replacing the Jensen–Shannon divergence (JS divergence) with Wasserstein distance, the loss function of WGAN is correlated with the output image quality, leading to the good result with ideal convergence [14].

(6) From pix2pix (2016) to CycleGAN (2017): pix2pix no longer inputs random noise when training, instead the inputs of discriminator(D) become paired images from class A and B. After training is finished, the output results from class B could be controlled by inputting a new image similar to class A [15]. Compared to pix2pix, CycleGAN no longer needs paired images to train, it only requires an input dataset and an output dataset. In addition, it makes constraints on networks by giving another network to train the output dataset back to the input dataset. The whole process is trying to reach to

the goal that the result-input should be the same as the original-input. CycleGAN successfully divided style and content, and it is easy to learn automatically maintaining contents when changing styles [16].

(7) From ProGAN (2017) to StyleGAN (2018): By introducing progressive training, the method that networks train from low resolution and gradually increase until reaching the max of the target, ProGAN solved the problem of GANs generating high-quality and high-resolution images [17]. However, essentially, it is a method of generating unconditional data which is hard to control the property of output images. Based on ProGAN, StyleGAN mainly modified the input of each level separately to control the visual features represented by this level without affecting other levels. The lower the level, the lower the resolution and the coarser the visual features that can be controlled. This method helps to generate different levels or multilayer style-mixing outputs. In addition, instead of using traditional random vector input, StyleGAN introduced Mapping Network and Adaptive Instance Normalization (AdaIN) to control visual features of the generated image, leading to the effect of truncation trick as a function of style scale [18].



(a) GAN (b) CGAN (c) infoGAN (d) ACGAN (e) CycleGAN (f) StyleGAN

Figure 1. These are main process diagrams of GANs (Generative Adversarial Networks): (**a**) Original GAN. (**b**) Conditional GAN. (**c**) Information maximizing GAN. (**d**) Auxiliary classifier GAN. (**e**) Cycle-consistent GAN. (**f**) StyleGAN.

Compared to the other GANs, The StyleGAN is typically talented in generating three categories of results: (a) similar fake images, (b) style-mixing images, and (c) truncation trick images. The similar fake images share common features with these input original images but look like another brand new design. The style-mixing images are the result of 2 similar fake images. One has its basic content information and the other one transfers its image style to the content one. The truncation trick images are a series of images continuing transforming style from one type to another one.

However, it is still not enough for the three-dimensional architectural form generation. To generate the 3D model from the training results of StyleGAN, the author uses the method similar to the 3D reconstruction of medical CT images [19]. The 3D model is composed of a sequence of continuously changing images. By using the pixels of the specific color gamut in the images, each single image will function as the single-layer model information.

3. Training Results and Discussion Through StyleGAN

StyleGAN's original paper offered lots of high-quality human face generated results, which is based on its new dataset of human faces called Flickr-Faces-HQ (FFHQ). In order to get ideal results, the dataset requires both a certain degree of similarity and sufficient varieties, such as the FFHQ containing only human portrait images but with vast types of differences in terms of age, ethnicity, hair, expression, image background, etc. For training architectural plan or section images through StyleGAN, it is better to constrain the input data to just plan or section but with varieties of types, functions, scales, details, and background situations. Data sheet of each training is given in Table A1 of Appendix A.

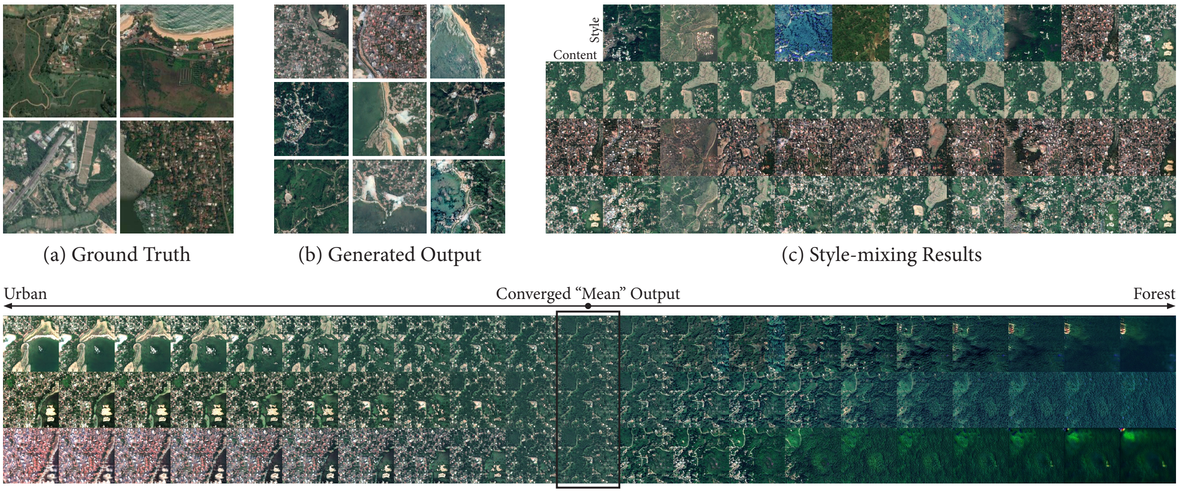
3.1. Training with Collected Datasets

In order to test StyleGAN's learning process for two-dimensional architectural drawings and evaluate its generated results based on a large and variable amount of input data, the author collected architectural drawings related to satellite urban plan, building plan, and section as input training data. These datasets and the corresponding generated results are listed below.

3.1.1. Satellite Images Training

Satellite images are one of the most accessible resources when preparing data for plan images training. The author gathered 30,000 satellite images from Google Maps containing cities, countrysides, forests, rivers, and oceans. They are on the same level of scale with 256^2 resolution.

Figure 2 shows the example of the input dataset and all three types of generated results. The training process absorbed several obvious features from satellite plan information to a certain degree then generated them out, e.g., the clear sand beach coastline crossing forest land and sea, the detailed urban buildings with various roof colors, and the combination area with small-town and forest which have road networks growing inside. Benefiting from the tremendous amount of input resources with large enough diversities, the style-mixing results are not only specific but also show a high level of differentiation degrees. Variation of output information offers smooth transformation sequences (truncation trick images), clearly shifting the style from urban to forest.



(a) Ground Truth

(b) Generated Output

(c) Style-mixing Results

(d) Truncation Trick Images

Figure 2. These are the training processes and results of collected satellite images: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.

The general training results from collected satellite images are relatively good, clear and expected output is generated with ideal convergence. However, the basic resolution is not enough (256^2), the scale of all inputs and outputs is too big that it is ambiguous to figure out what architectural details were learned and generated inside, leading to the difficulty of enlightening the architectural design.

3.1.2. Collected Plan Drawings Training

To find out how StyleGAN tries to understand and reproduce the features of the architectural plan, 2000 plan drawings were collected to train. These input data are all in grayscale (8-bit depth) with varieties of scales, styles, and details. They are on multiple levels of scale with 512^2 resolution.

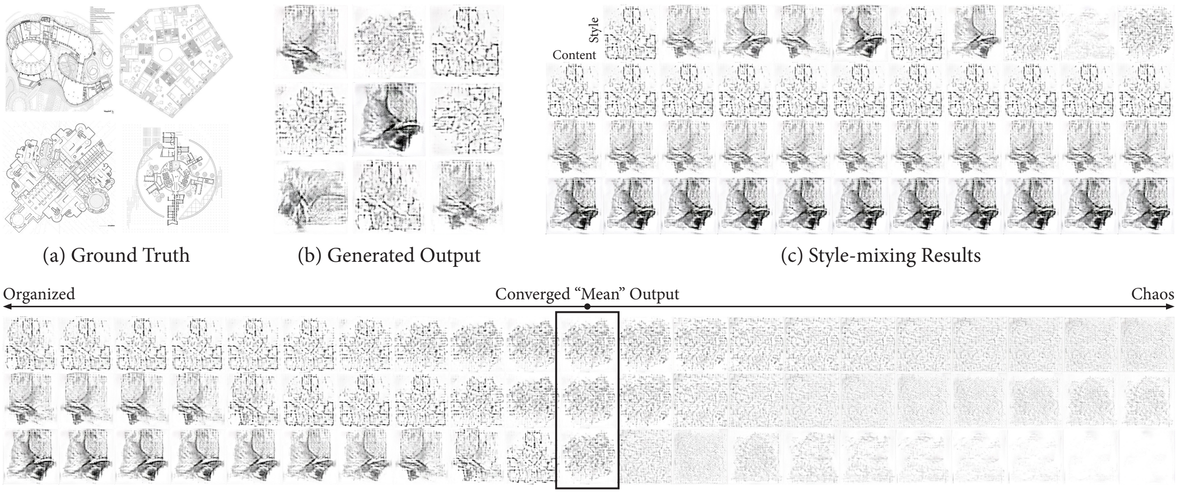
Figure 3 shows the example of the input dataset and all three types of generated results. In this turn of training, the convergence almost fails, such that generated output is limited to several types, which are all far from similar to original input. It failed to learn a feature of circle walls which are up to half of all input. The style-mixing results remain on original contents and all of the style transformations are finally becoming chaos.

Compared to the previous dataset of satellite images training, these collected plan drawings lack both amount and variations. In addition, the generated transformation sequences require clear design purpose (just like from urban to forest) otherwise the end of the transformation will always turn out to be chaos. This kind of generation (truncation trick images) is somehow StyleGAN's attempt of applying negative scaling to original results, leading to the corresponding opposite results. It will be extremely hard for GAN to expect the totally reversed situation if there are no such opposite references to learn from.

3.1.3. Collected Plan and Section Drawings Training

Based on the previous plan drawings training, it is better to clarify the goal of style transfer which, in this case, is the style between architectural plan and section, so that the training process will notice the opposite input resources then control the convergence more easily.

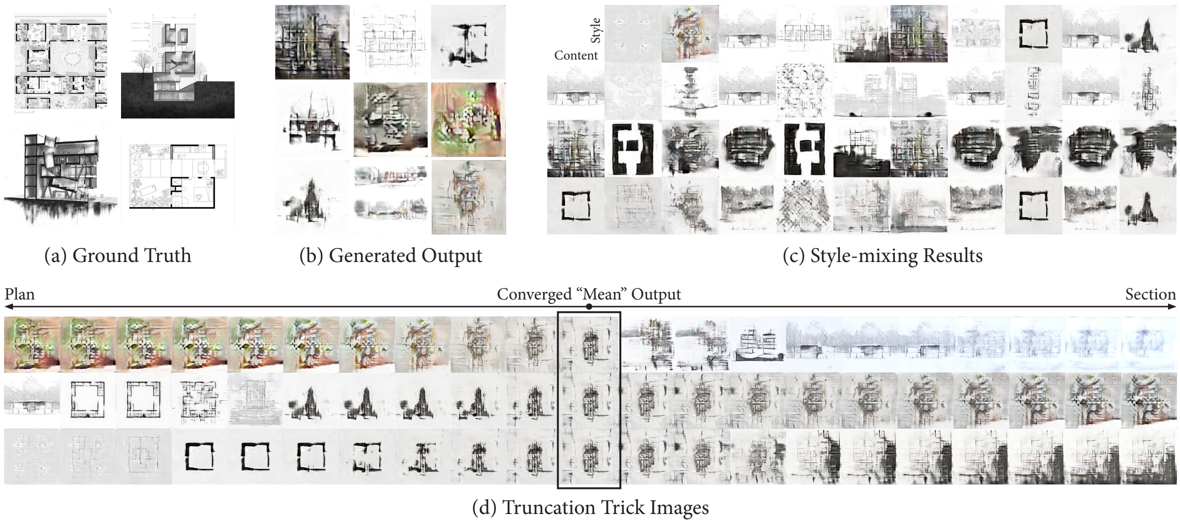
Figure 4 shows the process and results of the collected plan and section drawings training. It is trained with 5000 images containing variable styles of both plan and section drawings on 512^2 resolution. Although these generated outputs are still fuzzy and confusing, generally they give expression to basic shape and form from the original input plan and section drawings. These style-mixing results seem to be disorderly and random, however, the interaction of multiple styles is largely realized. Benefiting from more information on patterns and features of the input drawings, compared to previous plan training the style transformations from this turn are no longer converged to random chaos, also showing not linear but multiple-style transfer under two general directions.



(a) Ground Truth (b) Generated Output (c) Style-mixing Results

(d) Truncation Trick Images

Figure 3. These are the training processes and results of collected plan drawings: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.



(a) Ground Truth (b) Generated Output (c) Style-mixing Results

(d) Truncation Trick Images

Figure 4. These are the training processes and results of collected plan and section drawings: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.

3.2. Training with Designed Datasets

The previous training with collected datasets has mainly 3 limitations: (a) Output quality is not enough, leading to poor readability and identification of information. (b) Convergence is easy to fail, which makes the training result become random chaos. (c) The difficulty and ambiguity for creators to use when trying to make the design on purpose through the training process.

Based on these limitations, the author introduces a new way of preparing training datasets where designers manually create plan or section images with the specific design approach. These input data will help the machine to better understand what we expect the learning process to absorb and what kind of results we are trying to make.

3.2.1. Dynamic Circle and Square Test Training

In order to achieve the purpose of design that clearly transfers the shape and style we want from one type to another, the first step is to test on two types of simple geometric drawings which are obviously different and easily identified.

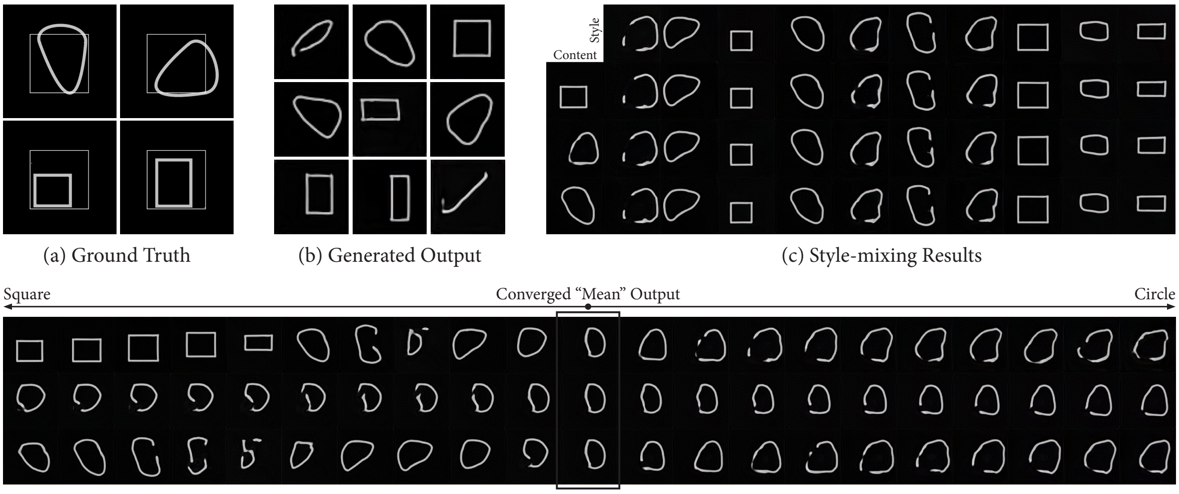
Figure 5 shows the process and results of the designed circle and square drawings training. The input dataset is made with 10,000 circle or square shapes of drawings on a dynamic layout with 512^2 resolution. The "dynamic layout" means these input circles or squares are not constrained to be all the same shape and location but randomly growing. Although style-mixing results share the same problem with results from plan drawings training, these basic generated outputs hold decent fidelity and clarity. In addition, these truncation trick images show not only clear transfer direction but also good quality of continuous transformation.

3.2.2. Dynamic Single Drawing Variations Training

Based on the circle and square test training, the author designed one single architectural drawing with several parts that keep changing to create dynamic variations for training.

These 10,000 input drawings are all sections on 512^2 resolution with fixed rock bed, water line, and foundation. In contrast, their bridges, walls, and roofs are varied within a reasonable distance. In order to make sure there is a clear designed transfer direction, this dataset is generally divided into two main styles: one has a quadrilateral interior space with a polygonal roof and another has a domed roof forming an interior space with a semicircle.

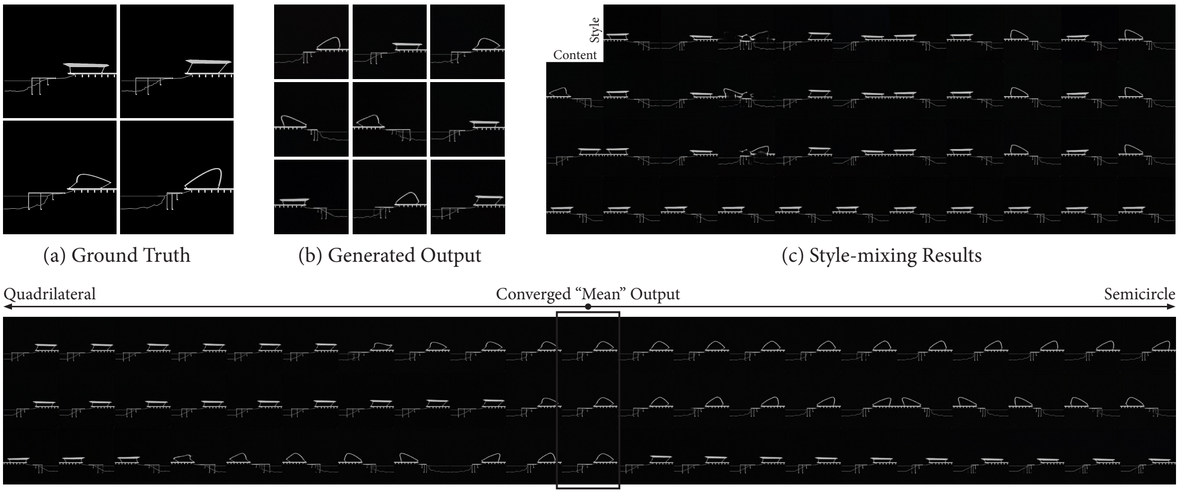
Figure 6 shows the process and results of designed single drawing variations training. The training of this turn learned to generate almost the same section drawings as the input data, however, these style-mixing results are still not ideal due to the limitation of the dataset which was constrained to only one single main style. On the other hand, these style transformations show a clear transition direction from quadrilateral form to semicircle form, and some part of them even jump through multiple styles.



(a) Ground Truth (b) Generated Output (c) Style-mixing Results

(d) Truncation Trick Images

Figure 5. These are the training processes and results of designed circle and square drawings: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.



(a) Ground Truth (b) Generated Output (c) Style-mixing Results

(d) Truncation Trick Images

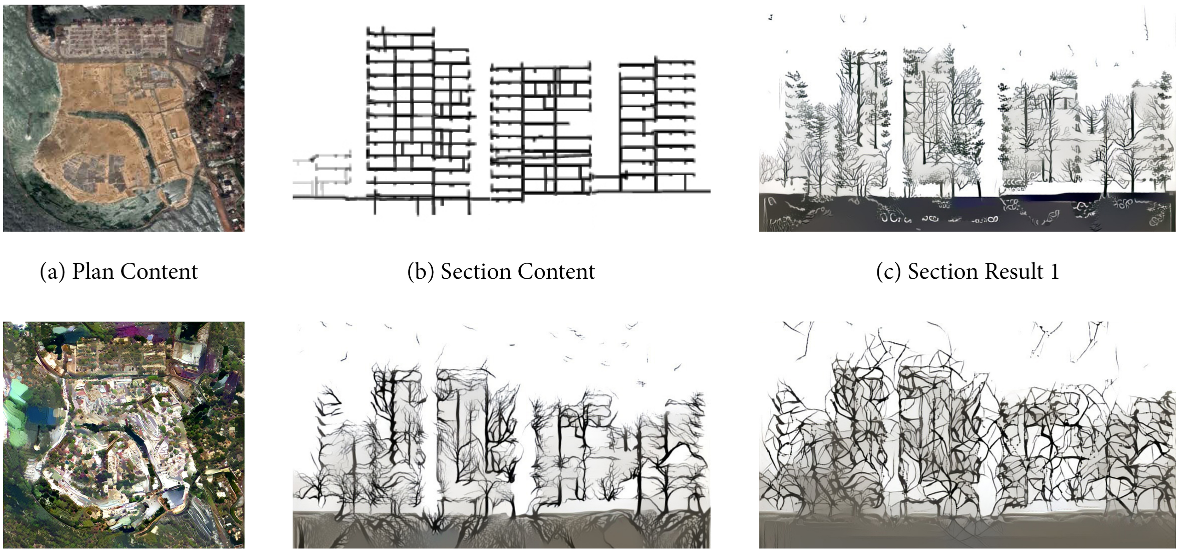
Figure 6. These are the training processes and results of designed single drawing variations: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.

3.2.3. Design Corresponding Style Input through Style Transfer GAN

In order to make the training set more abundant to fill the deficiency of learning resources for style-mixing, multiple styles of input are required. However, each style of the training input should not be arbitrarily designed but should be corresponding to each other with some degree of shared information.

The "neural style transfer" proposed by Gatys et al. (2015) provides a single image style transfer method [20]. This GAN is used to blend two single images (a content one and a style one) together so the output image will look like the original content one but formed with the style of the input style one.

Figure 7 shows the input and output of plan and section drawing training through neural style transfer GAN. With the style input of one satellite image containing both urban and forest, this style transfer GAN turns the content image of sandy island into an island city with white color houses and trees growing inside. What is more, based on the content image of the designed buildings section drawing, one style image of the forest's elevation drawing keeps transferring its style, turning the content input into the architectural section with branches growing on the floors and walls. There are 3 outputs referring to 3 levels of style influence strength, from result 1 to result 3 the style is more and more deepened. As a result, with the appropriate configuration of training, we can generate the ideal corresponding style image as a supplement to StyleGAN's training set.



(a) Plan Content (b) Section Content (c) Section Result 1

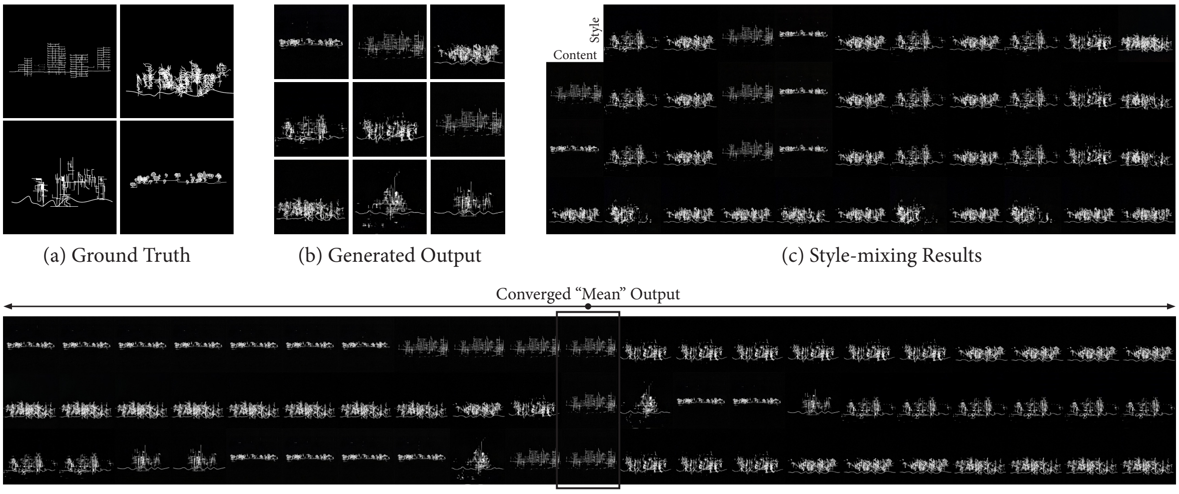
(d) Plan Result (e) Section Result 2 (f) Section Result 3

Figure 7. These are single plan and section drawing training examples through "neural style transfer": (**a**) Plan content input. (**b**) Section content input. (**c**) Section generated output 1. (**d**) Plan generated output. (**e**) Section generated output 2. (**f**) Section generated output 3.

3.2.4. Dynamic Multiple-Style Drawings Training

Through the training of neural style transfer GAN, multiple-style section drawings were designed to make up the input dataset for this new turn of training. This dataset has 4 categories: (a) Tall buildings. (b) Tall buildings with forest style (branches growing inside). (c) Tower buildings. (d) Forest with trees growing. Each category has 5000 images on 512^2 resolution, and the architectural parts within these images are constantly changing locations.

Figure 8 shows the process and results of designed multiple-style drawing training. Overall, in this round of training, the convergence of generated output and style-mixing situation compared to the previous one is not improved much. However, it is worth noting that the style transformation results show StyleGAN's ability to learn both shared and opposite features in the four styles and then calculate the continuous variations between them.

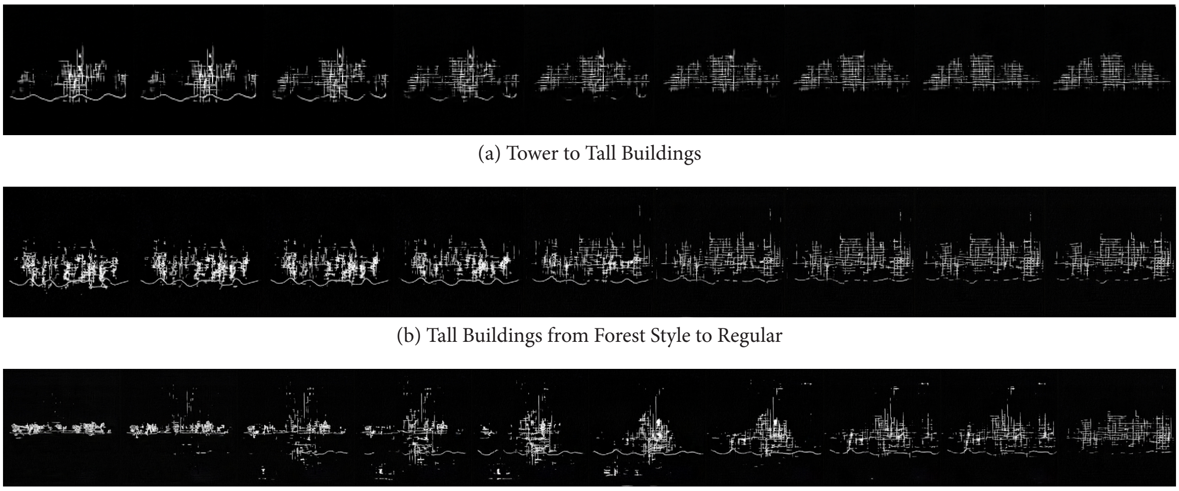


(a) Ground Truth (b) Generated Output (c) Style-mixing Results

(d) Truncation Trick Images

Figure 8. These are the training processes and results of multiple-style drawings: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Contents, styles, and mixed results. (**d**) Continuing transforming images stack.

However, it is still unclear to understand how exactly StyleGAN transfers one style to another. This inner process requires deeper exploration of the transformation and smaller truncated dimensions. Figure 9 shows a more detailed and intuitive process of transfer from one designed architectural section to another style. This process can capture the gradual style changes between two completely different sections (e.g., from tower to tall buildings), or the transfer between the second style section generated from the original section and the original section itself (e.g., from forest style tall buildings to original tall buildings), or even jump multiple times between different styles (e.g., from trees to tower to tall buildings).



(a) Tower to Tall Buildings

(b) Tall Buildings from Forest Style to Regular

(c) Trees to Tower to Tall Buildings

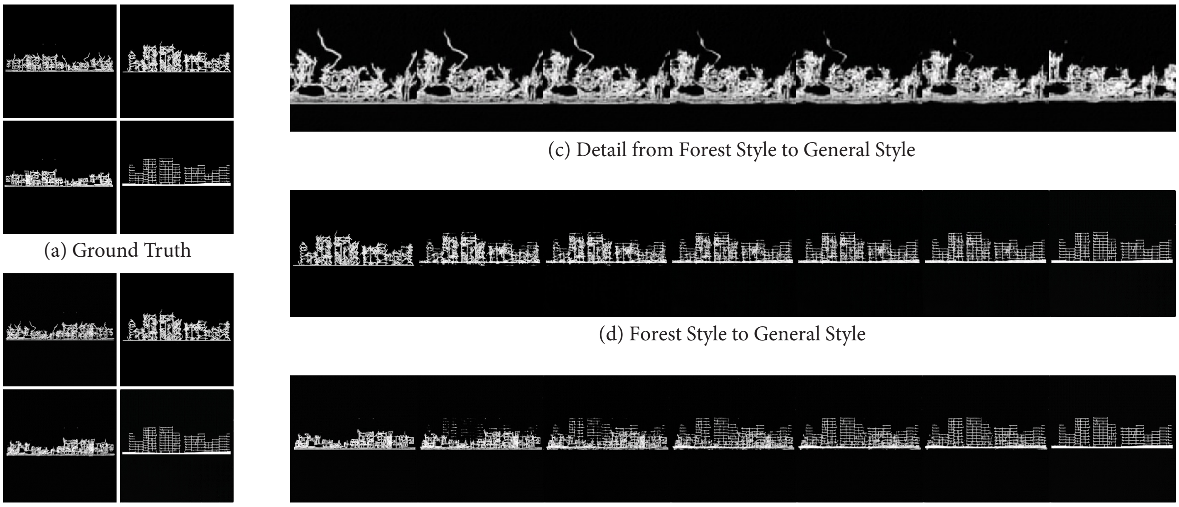
Figure 9. These are style transformations of multiple-style drawings training on a smaller truncated scale: (**a**) Style transfer from tower to tall buildings. (**b**) Style transfer from forest style tall buildings to normal tall buildings. (**c**) Style transfer from trees to tower to tall buildings.

3.2.5. Static Multiple-Style Drawings Training

Although dynamic dataset offers sufficient learning resources and sample diversity, its generated results are always hard to simulate the original data clearly and accurately, not to mention implementing the exact style transfer gradient between the designed input drawings that we expected.

In order to restrict StyleGAN from learning simulating specific input data with explicit design goals, a static type of training data works much better than the dynamic one. In each category of styles, input images with static type will no longer change but remain consistent.

Figure 10 shows the process and results of static multiple-style drawings training. In the dataset of this StyleGAN's training, regular architectural sections of two scales (high-rise and bungalow) were the input, as well as corresponding data of sections with overgrown branches obtained through the transfer of forest style by neural style transfer GAN. There are 3000 images of each category in the input collection, which are all exactly the same and have a resolution of 512^2.


(a) Ground Truth

(b) Generated Output

(c) Detail from Forest Style to General Style

(d) Forest Style to General Style

(e) Bungalow to High-rise

Figure 10. These are the training processes and results of static multiple-style drawings: (**a**) Example of input data. (**b**) Selected output fake images. (**c**) Detail of style transfer from forest style to general style. (**d**) Style transfer from forest style to general style. (**e**) Style transfer from bungalow to high-rise.

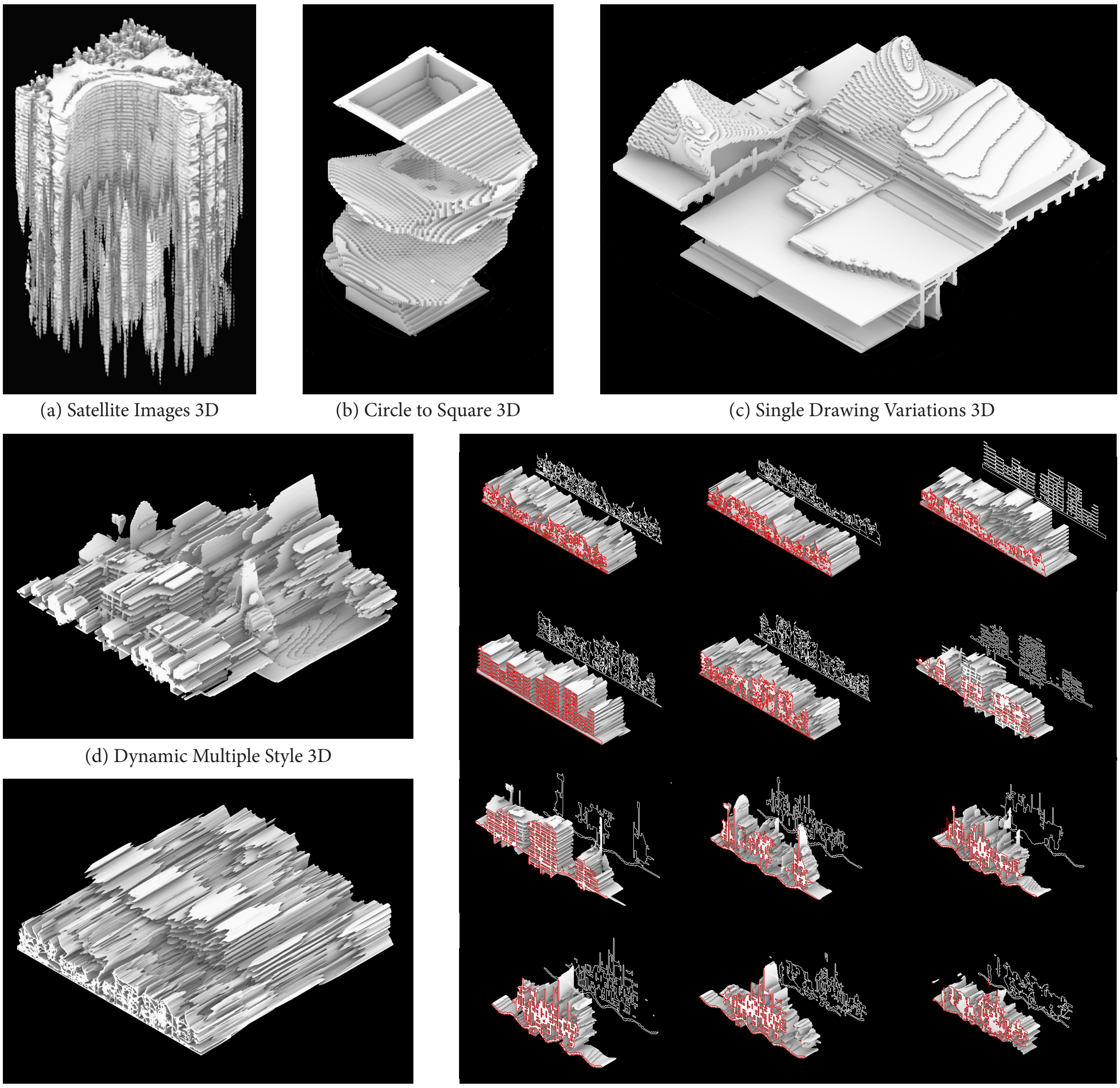
It is expected that the generated output is almost identical to the input, and the style transformation will clearly meet the expectation of style transfer purpose we try to achieve. The transition process from forest style to general style is very clean and straightforward, helping us to explore and master the in-between situation of two original inputs. As for the style transformation from high-rise to bungalow, due to the lack of learning samples in the upper space, the data higher than bungalow could only gradually disappear during the transition.

These style transformations can help us to explore the spatial evolution of the fusion of sections with different attributes or functional styles in architectural design, and inspire us to design new section forms and structures under mixed styles.

4. Attempts of 3D Generation

To generate the 3D model from the training results of StyleGAN, the author uses a method similar to the 3D reconstruction of medical CT images. The 3D model is composed of a sequence of continuously changing images. By using the pixels of the specific color gamut in the images, each single image will function as the single-layer model information.

Figure 11 shows 3D models generated from truncation trick images of previous training results. Benefiting from the continuous variation of the transformation, we can observe the spatial form of the transition state in the transformation of the architectural section style more intuitively through the generated 3D model. In the 3D model from satellite images training, the shape of the urban landscape is relatively clear at the top, while at the bottom, when the transition to forest occurs, the whole model starts to become fragmented and chaotic. The 3D model from single drawing variations training shows how dynamic versus static data in a dataset can affect the generated results. Compared to the other dynamic data, the rock bed, water line, and foundation are all fixed elements in the input images, as a result, they remain in their original form. Both dynamic and static multiple-style 3D models exhibit considerable complexity and spatial hierarchy due to the diversity of input learning resources.



(a) Satellite Images 3D

(b) Circle to Square 3D

(c) Single Drawing Variations 3D

(d) Dynamic Multiple Style 3D

(e) Static Multiple Style 3D

(f) Split Model from both Dynamic and Static Multiple Style 3D

Figure 11. These are the 3D models generated from truncation trick images of previous training results: (**a**) 3D model generated from satellite images training. (**b**) 3D model generated from circle to square training. (**c**) 3D model generated from single drawing variations training. (**d**) 3D model generated from dynamic multiple-style training. (**e**) 3D model generated from static multiple-style training. (**f**) Split 3D model from both dynamic and static multiple-style training (the red line represents the section at the front and the white line represents the section at the back).

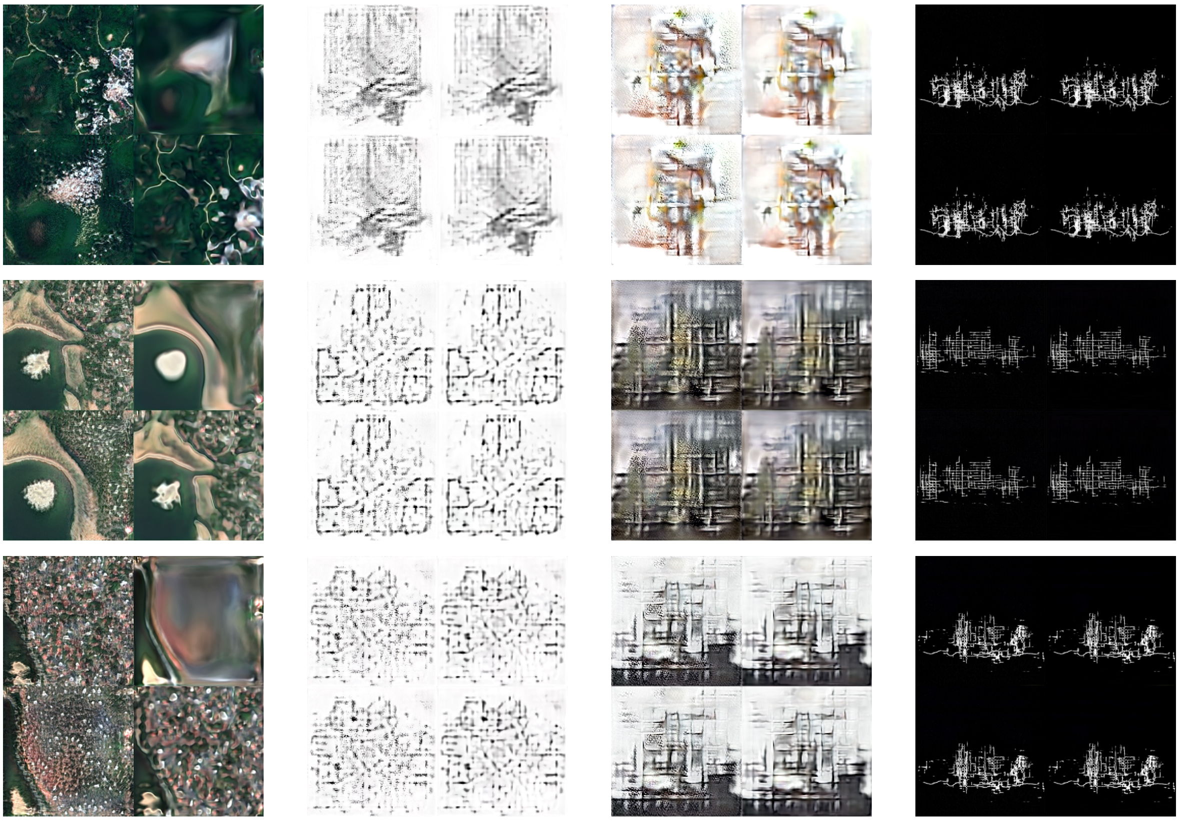
However, the main problem of these generated 3D models is that the architectural forms are more or less like simply stretching between two different styles of sections. Even with the precise control of each designed section node (like the split model in Figure 11), the generated spaces between them are still difficult to be directly used as architectural design references in a 3D scale.

5. Noise Effect on Different Layers

StyleGAN found a potential benefit of the ProGAN progressive layers. The lower the level, the lower the resolution, and the coarser the visual features that can be controlled. Based on this, StyleGAN adds per-pixel noise after each convolution to implement stochastic variation, then is able to control noise effect on different visual features separately.

Figure 12 shows the effect of noise inputs generated from previous training results on different layers. Each set of results has four graphs: (1) Upper Left—noise is applied to all layers. (2) Upper Right—no noise applied. (2) Lower Left—noise in fine layers only (64^2–1024^2). (2) Lower Right—noise in coarse layers only (4^2–32^2).

For the satellite images training, it is obvious that the whole generated results will become muddy without any noise (Upper Right). In addition, under the influence of only the noise in fine layers (Lower Left), the output simply adds some random details according to the original frame. Only when the coarse noise is applied will the original pixel block develop into the result with a geometric frame.



(a) Satellite Images (b) Collected Plans (c) Collected Plan and Section (d) Designed Multiple Style

Figure 12. These are noise effects on different layers: (**a**) Noise effect from satellite images training. (**b**) Noise effect from collected plan drawings training. (**c**) Noise effect from collected plan and section training. (**d**) Noise effect from designed multiple-style drawings training.

When all levels of noise are applied (Upper Left), both clear frame and sufficient details will be generated. As for the rest of the training results, they do not have a large number of datasets full of variability as in satellite images training, leading to the scarcity of stochastic elements, so the noise effect is not obvious.

6. Conclusions

StyleGAN has a strong ability for image learning and reconstruction simulation. It is good at absorbing the pixel characteristics of images in the input dataset, producing highly similar imitation outputs, and even summarizing the whole input data into one converged ”mean output“. In addition, it can realize the cross-influence among the input data through the style-mixing or style transformation.

Based on these generated results trained from various architectural plan and section datasets, there is no doubt that the effect of style transformation (truncation trick images) is the most stable and ideal compared with other types of output. In contrast, the style-mixing failed to generate results in most of the previous training trials. It is not because the architectural plan and section drawings are not suitable for the generation of style-mixing. Just as in the training of those collected data, the result of style-mixing is always relatively better. In several training trials of designed data, the diversity of the datasets is so low that they are directly identified as belonging to the same style by StyleGAN.

The main application of StyleGAN in this paper is to learn and simulate the relationship between two architectural plans or sections, so as to explore the intermediate state between the two styles and generate serialized transformation pictures accordingly to build a 3D model. Despite the fact that the results of 3D model generation are difficult to be directly used in 3D spatial modeling due to the illogicality of stretching-like form, those transformation sequences between the two or multiple styles still produce unexpected results that go beyond human thought patterns, to inspire us with the new design method and more potential possibilities for architectural plan and section design.

7. Image Credits

Figure 1 (f) StyleGAN: © T. Karras, S. Laine, and T. Aila (2018).
All other drawings and images by the author.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. This is the data sheet of each training process. ″Style-mixing“ refers to whether this training could make style-mixing or not. The ″3D“ refers to whether this training could generate clear serialized images to form a 3D model or not. The ″Inputs Number“ refers to how many images has been set into the training dataset. The ″Resolution“ refers to the resolution of the training images. The ″Kimg“ refers to the iterations of training process. The ″Time/hours“ refers to how many hours it will take for the single RTX 2080ti the author use to reach the purpose iteration.

Name	Style-Mixing	3D	Inputs Number	Resolution	Kimg	Time/Hours
Satellite Image	yes	yes	30,000	256^2	6126	30.5
Plan Drawings	no	no	2000	512^2	7140	69.5
Plans and Sections	yes	no	5000	512^2	6526	55.5
Circle and Square	no	yes	10,000	512^2	4705	26.2
Single Drawing Variations	yes	yes	10,000	512^2	4705	17.1
Dynamic Multiple-Style	no	yes	20,000	512^2	6526	48.3
Static Multiple-Style	no	yes	12,000	512^2	5726	31.5

References

1. Harry, W. Chairman's introduction. In *Computer-Aided Architectural Design Futures, Proceedings of the International Conference on Computer-Aided Architectural Design, Delft, The Netherlands, 18–19 September 1985*; Pipes, A.; Ed.; Butterworths: London, UK; Boston, MA, USA; pp. 2–3.
2. Salman H S.; Laing R.; Conniff A. The impact of computer aided architectural design programs on conceptual design in an educational context. *Des. Stud.* **2014**, *35*, 412–439. [CrossRef]
3. Weron R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [CrossRef]
4. Cincotti, S.; Gallo, G.; Ponta, L.; Raberto, M. Modeling and forecasting of electricity spot-prices: Computational intelligence vs classical econometrics. *AI Commun.* **2014**, *27*, 301–314.

5. Huang W.; Zheng H. Architectural drawings recognition and generation through machine learning. In Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, Mexico City, Mexico, 18–20 October 2018; Anzalone, P., Signore, M., Wit, A., Eds.; Association for Computer Aided Design in Architecture (ACADIA): Fargo, ND, USA, 2018.
6. Del Campo, M.; Manninger, S.; Sanche, M.; Wang, L. The Church of AI—An examination of architecture in a posthuman design ecology. In Proceedings of the 24th CAADRIA Conference, Wellington, New Zealand, 15–18 April 2019; Haeusler, M., Schnabel, M., Fukuda, T., Eds.; The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA): Hong Kong, China, 2019.
7. Koh, I.; Amorim, P.; Huang, J. Machinic Design Inference: From Pokemon to Architecture—A Probabilistic Machine Learning Model for Generative Design using Game Levels Abstractions. In *Computer-Aided Architectural Design, Proceedings of the 18th International Conference, CAAD Futures 2019, Daejeon, Korea, 26–28 June 2019*; Ji-Hyun, L.; Ed.. Springer: Berlin/Heidelberg, Germany, 2019.
8. Kyle, S.; Kat P.; Adam M. Fresh Eyes A Framework for the Application of Machine Learning to Generative Architectural Design, and a Report of Activities at Smartgeometry 2018. In Proceedings of the 24th CAADRIA Conference, Wellington, New Zealand, 15–18 April 2019; Haeusler, M., Schnabel, M., Fukuda, T., Eds.; The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA): Hong Kong, China, 2019.
9. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **2014**, 2672–2680.
10. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
11. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
12. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2016**, 2172–2180.
13. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Austalia, 6–11 August 2017; Volume 70, pp. 2642–2651.
14. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
15. Isola, P.; Zhu, J.; Y.; Zhou, T.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
16. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
17. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
18. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4401–4410.
19. Wang, H.; Pu, X. 3D medical CT images reconstruction based on VTK and visual C++. In Proceedings of the 2009 3rd International Conference on Bioinformatics and Biomedical Engineering, Beijing, China, 11–16 June 2009; pp. 1–4.
20. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.