


Article

Facade Style Mixing Using Artificial Intelligence for Urban Infill

Ahmed Khairadeen Ali ^{1,2,*}  and One Jae Lee ²¹ Architectural Engineering Department, College of Engineering, University of Duhok, Duhok 42001, Iraq² Haenglim Architecture and Engineering Company, Seoul 431810, Republic of Korea; o.lee@haenglim.com

* Correspondence: ahmedshingaly@gmail.com

Abstract: Artificial intelligence and machine learning, in particular, have made rapid advances in image processing. However, their incorporation into architectural design is still in its early stages compared to other disciplines. Therefore, this paper addresses the development of an integrated bottom-up digital design approach and describes a research framework for incorporating the deep convolutional generative adversarial network (GAN) for early stage design exploration and the generation of intricate and complex alternative facade designs for urban interiors. In this paper, a novel facade design is proposed using the architectural style, size, scale, and openings of two adjacent buildings as references to create a new building design in the same neighborhood for urban infill. This newly created building contains the outline, style and shape of the two main buildings. A 2D building design is generated as an image, where (1) neighboring buildings are imported as a reference using the cell phone and (2) iFACADE decodes their spatial neighborhood. It is illustrated that iFACADE will be useful for designers in the early design phase to create new facades in relation to existing buildings in a short time, saving time and energy. Moreover, building owners can use iFACADE to show their preferred architectural facade to their architects by mixing two building styles and creating a new building. Therefore, it is presented that iFACADE can become a communication platform in the early design phases between architects and builders. The initial results define a heuristic function for generating abstract facade elements and sufficiently illustrate the desired functionality of the prototype we developed.

Keywords: artificial intelligence; generative adversarial network; machine learning; computational design; urban infill; facade design



Citation: Ali, A.K.; Lee, O.J. Facade Style Mixing Using Artificial Intelligence for Urban Infill. *Architecture* **2023**, *3*, 258–269. <https://doi.org/10.3390/architecture3020015>

Academic Editor: Avi Friedman

Received: 23 January 2023

Revised: 26 March 2023

Accepted: 9 May 2023

Published: 11 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Generally, building envelope configurations follow similar urban street configurations dictated by city policy, including building height, width, color, openings, and the ratio of open to built space [1]. Thus, when designing a new building in an existing neighborhood, architects typically look to the architecture of neighboring buildings to determine the mass of the new building. The result is that each neighborhood uses an almost typical and repetitive module for all building massing, and the architect's input in the design of new buildings is minimal due to city policy and existing neighboring buildings [2].

Conventional building design is an iterative process that is costly and time consuming: (1) draw a bubble diagram of building zones and outlines; (2) create appropriate floor plans and elevations and solicit input from clients; (3) return to the polished bubble diagram; and (4) iterate. Architects and their clients must also compromise on the quality of the design, given budget and time constraints. Therefore, there is a great demand in the architecture, design and real estate industries for automated building mass generation techniques that have immense potential [3].

Recently, there have been several experiments with generative approaches, variational autoencoders, in the field of computer vision (VAEs) [4], and generative adversarial networks (GANs) [5]. In particular, GANs show important results in various computer vision

tasks, such as image generation [6–8], image conversion [9], super-resolution [10], and text–image synthesis [11]. GANs have already been used for facade generation [8]. Pix2Pix is based on conditional GAN [5]. By using conditional vectors, we can control the categories or attributes of the generated images. Pix2Pix generates facade images with the condition of a masked image containing predefined labels for each building element. However, both GANs failed to generate facade images that look realistic.

In this paper, a novel facade design is proposed using the architectural form, height, scale and openings of two adjacent buildings as a guide to construct a modern building design with iFACADE in the same neighborhood for urban infill. The outline, style and type of the superior buildings are used in this newly constructed building. As an image, a 2D design for an urban infill building is created, where (1) neighboring buildings are imported by cell phone for reference and (2) iFACADE decodes their spatial neighborhood. In the urban neighborhood, a building design is created as an image and improved. The main contributions of this work can be summarized as follows: (1) We propose the style-based conditional generator to control the latent space. (2) Our proposed generator can generate images reflecting each feature of condition information specifying multiple classes. (3) In the experiments with a facade dataset, we show that the sounds in the style-based generator indicate additional elements, such as building windows, walls, and outline information, and that unseen types of building facades can be generated by mixing multiple types of facades with mixed conditional weights.

2. Literature Review

2.1. Generative Adversarial Networks

Image-to-image conversion techniques aim to acquire a function of conditional image creation that maps a source domain's input image to a target domain's corresponding image. To solve diverse image-to-image conversion tasks, Isola et al. [12] first proposed the use of conditional GANs. Their thesis has since been expanded to several scenarios: unsupervised learning [7,8,13], multi-domain image synthesis [14], and conditional image synthesis [15]. For their tasks, the above works have built dedicated architectures that include training the generator network. In other hand, our research depends heavily on utilizing the conditional StyleGAN generator.

Many articles have recently proposed different methods for studying semantic edits of the latent code. A typical technique is to identify linear directions that lead to shifts in a particular binary named attribute, such as human face young to old, or no-smile to smile [6,14]. Abdal et al. [16] learn a translation between vectors in $W+$, changing a collection of fixed named attributes. Finally, by modifying relevant components of the latent code, Collins et al. [17] perform local semantic editing.

2.2. Feature Disentanglement

Some critical problems arise with the above approaches. First and foremost, the input image must be invertible, i.e., there must exist a latent code that reconstructs the image. However, the domain of the input image must usually be the same domain on which the GAN was formed. For various image-to-image activities that convert between two separate data domains, such as segmentation graphs, drawings, edges, etc., this limitation is a significant constraint. By optimizing an extended space, such as $W+$, this limitation can be circumvented [16,17]. In this case, however, the latent space does not contain rich semantics for an ambiguous data range. For example, the latent code would not be semantically relevant, even if the input sketches could be inverted when translating between sketches and natural face images in $W+$ of a StyleGAN trained on faces. Second, the investment in GAN remains overwhelming. Although direct encoding techniques have achieved moderate performance in the restoration standard, optimization and hybrid techniques are costly and take a long time to converge.

2.3. Facade Image Generation

Based on this insight, the continuous image-to-image architecture of this research directly encodes a given current image into the desired latent vector. Thus, one can directly solve the image-to-image challenge while taking advantage of a pre-trained StyleGAN without having to accept unnecessary constraints.

CycleGAN [7] transforms image to image but without changing the visual. This work only transforms the style between the two types of images. In contrast, in this paper, by setting several areas as constraints, we exploit the structure by fluctuating transformation and type noise, including two or more types.

3. Methodology

In this paper, the current literature in the field of image-to-image translation is reviewed, and a model is proposed that can generate new building facades by mixing two existing building facades in the neighborhood for urban infill. The methodology is divided into two parts. The first part discusses the urban infill tool called iFACADE and the second part discusses the development of the trained model.

The proposed framework generates an imaginary building from a reference building. A neural network trained with elevations of real buildings can transform this into a realistic building. If you then switch between different building views, it is possible to generate different views of the same imaginary building. The iFACADE can generate new and non-existent but realistic images by using conditional neural networks that remember a specific set of features they have seen in the past: the same process we humans go through when we dream.

3.1. Urban Infill System Architecture

Since each neighborhood has its own rules for a modular building style that all buildings within the neighborhood should follow, most buildings have the same general building characteristics, colors, window openings and style. Therefore, iFACADE was developed to create a new building view based on the existing building styles of the neighborhood.

The iFACADE system is divided into five main phases: (a) input of reference buildings in the neighborhood, where two color 2D facade images of reference buildings (B1 and B2) as shown in Figure 1b (iFACADE trained model) are required as the input and architectural guidelines of the neighborhood are required as the resource test to help the architect evaluate the generated facade correctly; (b) the use of iFACADE's trained model to input reference images, extract features, and generate a new 2D image using the trained model for style blending; (c) the output of a B0 2D image containing the features of both reference images as shown in Figure 1; (d) evaluations where the architect manually evaluates the generated facade and checks if it is compatible with the architectural features of the neighborhood and submits an evaluation report; and (e) the iFACADE team re-evaluates the trained model and improves the system architecture depending on the feedback from the architect and users.

The system is built in a cloud and a front-back-end website is developed to host the trained models. It is important that the website allows users to interact from anywhere by logging into it. Currently, the system has only one feature, which is the creation of style-mix images. In the future, it could be expanded to provide more information about the regulations of each neighborhood. The user submits an evaluation report, which helps the iFACADE team to evaluate the iFACADE model, improve its system architecture, and re-train it. In addition, user-uploaded images are added to and enhance the training dataset, which will help the iFACADE team create better and more accurate facades in the future as shown in Figure 1.

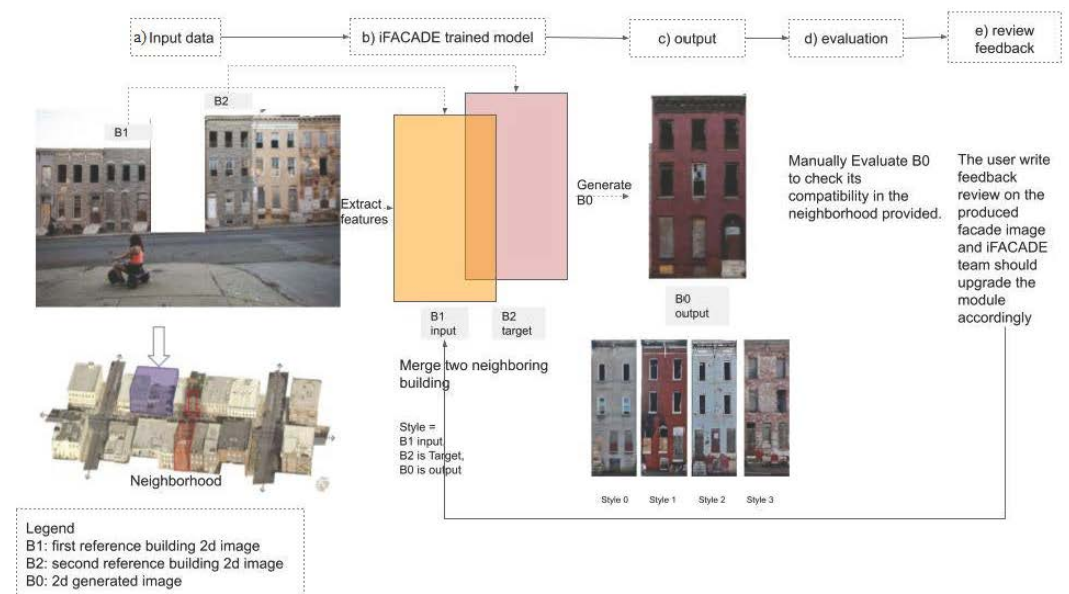


Figure 1. The iFACADE system architecture.

3.2. Style Mixing Trained Model System Architecture

In this subsection, the architecture of the model trained by iFACADE and the methods used to control the facade style are explained. Therefore, the main elements of the model architecture are explained, including the generator, the discriminator, the style mixture, the latent space control, and the dataset used to train the model.

In general, the iFACADE model consists of a generator that produces a fake facade image as a function of a real image and a discriminator that determines whether the input image is fake or real.

3.2.1. Conditional Style-Based Generator

In this subsection, we explain the model generator and custom adaptive instance normalization (AdaIN) architecture. The generator used in iFACADE is a conditional style-based neural network that uses transpose convolution to upscale input images. Compared to the traditional StyleGAN [18], the iFACADE generator controls the latent space of the conditional StyleGAN generator by inputting the conditional information (y) to the mapping function (f) in the fully connected (FC) layer. This method is adopted from [18].

This research performed a nonlinear conversion of the latent vector z to conditional style information into two scalars (y_s and y_b), where s stands for scalar and b stands for bias to control an adaptive instance normalization layer (AdaIN). The AdaIN operation is defined as

$$y = (y_s, y_b) = f(w) \quad (1)$$

$$\text{AdaIN}(x_i, y) = y_s \frac{x_i - \mu(x)}{\sigma(x)} + y_b \quad (2)$$

where x_i is a normalized instance that we apply AdaIN to, y is a set of two scalars (y_s, y_b) that control the “style” of the generated image, and $f(w)$ represents a learning affine transformation.

In the proposed generator, the AdaIN operation is normalized to map networks that show various styles. The dimensions of feature map z are half of the conditional style value. The objective function used in the StyleGAN of this research is adopted from [18] that uses the hinge version of the standard adversarial loss [6], defined as

$$L(G, D) = E_q(y)[E_q(x|y)[\max(0, 1 - D(x, y))]] + E_q(y)[E_p(z)[\max(0, 1 + D(G(z, y), y))]] \quad (3)$$

$$L(G, D^*) = -E_q(y)[E_p(z)[D^*(G(z, y), y)]] \quad (4)$$

where $p(z)$ is the standard Gaussian distribution. The generator's conditional mapping function z is fed to multiple fully connected (FC) layers. The latent code of a traditional StyleGAN generator is fed only the input layer (z) on the left. iFACADE adds additional conditional information (y) to the StyleGAN generator (left). The conditional code (y) gives more control on the style as depicted in Figure 2.

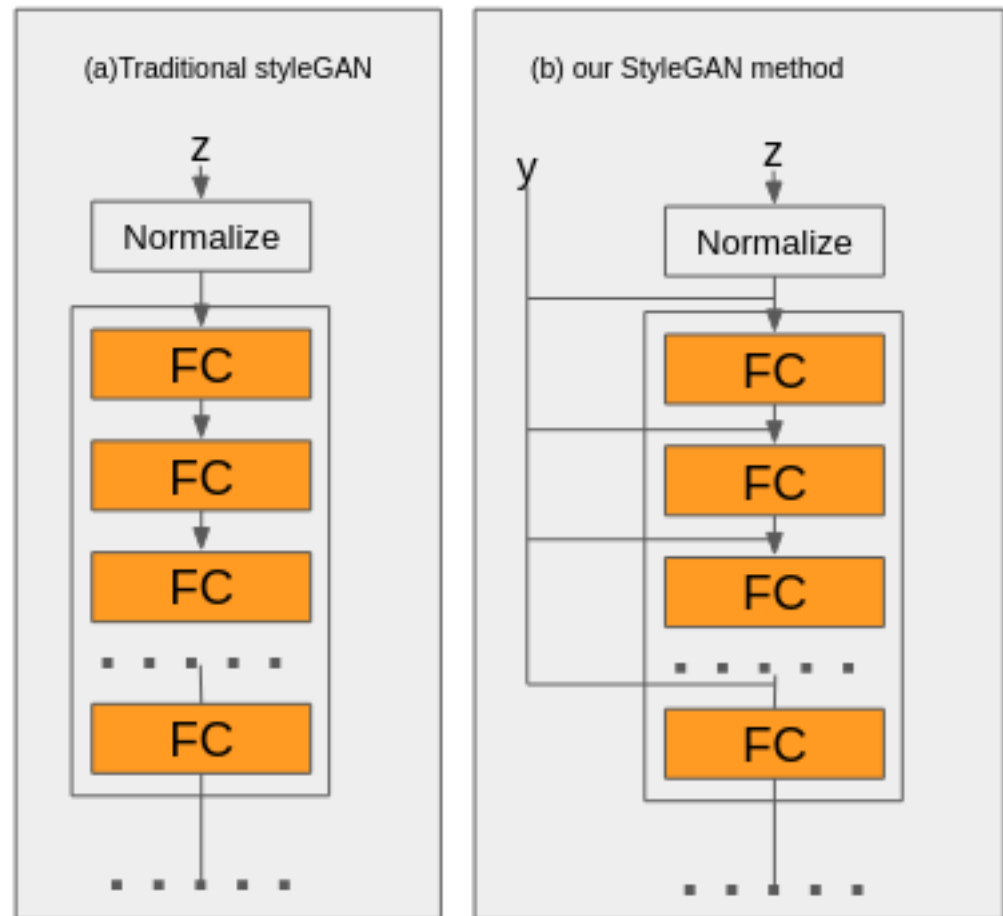


Figure 2. The latent code of a traditional StyleGAN generator is fed only the input layer (z) on the left. iFACADE adds additional conditional information (y) to the StyleGAN generator (left). The conditional code (y) gives more control on the style.

3.2.2. Conditional Discriminator

The discriminator is an unsupervised classifier that determines if the input image is true or fake. The model architecture of discriminator used in iFACADE is adopted by a projection discriminator proposed by [6]. Different from the approach of concatenating the embedded conditional vectors into feature vectors, the projection discriminator integrates conditional vectors.

3.2.3. Style Transfer

To further encourage the styles to localize, we employ mixing regularization, where a given percentage of images are generated using two random latent codes instead of one during training. It switches from one latent code to another at a randomly selected point in

the synthesis network when producing such an image. To be specific, we run two latent codes z_1, z_2 through the mapping network, and have the corresponding w_1, w_2 control the styles so that w_1 applies before the crossover point and w_2 after it. This regularization technique prevents the network from assuming that adjacent styles are correlated.

As a basis for our metric, we use a perception-based pairwise image distance that is calculated as a weighted difference between two Visual Geometry Group from Oxford (VGG16) [19] embeddings, where the weights are fit so that the metric agrees with human perceptual similarity judgments. If we subdivide a latent space interpolation path into linear segments, we can define the total perceptual length of this segmented path as the sum of perceptual differences over each segment as reported by the image distance metric. The average perceptual path length in latent space is Z over all possible endpoints.

4. Case Study

4.1. Model Training Dataset

This research trained a conditional generative adversarial network, using customized StyleGAN [18] to reconstruct the facade images from the CMP dataset. It uses a total of 720 images of facade that were adopted from the Center for Machine Perception (CMP) [20], eTraining for Interpreting Images of Man-Made Scenes (eTRIMS) [21] and EuroCity Persons (ECP) datasets [22]. The CMP dataset contains 606 pairs of annotated and real images of facade images. The images are from different international cities, but they share a similar modern architecture style with minor detailed architectural style differences that are neglected in this paper. We processed the images manually and chose the best 420 images and erased the rest. The additional images were collected from the eTRIMS database, which contained 60 facade images, and the Ecole Centrale Paris facade database. The facade images collected were processed to 128×128 pixels with 3 channels, and divided to 80 percent training, 15 percent test and 5 percent validation. The facade images constraints are the following 12 classes: facade, molding, cornice, pillar, window, door, sill, blind, balcony, shop, decoration, and background.

To increase the training speed, the images resolution were decreased to 128×128 pixels with three channels; they are not suitable for high-resolution image generation. This research normalizes the image color values to $[-1, 1]$ before feeding the image into the model.

4.2. iFACADE Model Training

This research used Tensorflow to implement the model training. We also started to generate resolutions from 8×8 . The models were optimized by stochastic gradient descent. For all experiments, the learning rate was fixed at 0.002, which updates the generator once for each discriminator update.

We implemented the proposed architecture in Tensorflow using a workstation with a NVIDIA 2080 Ti GPU. Our model uses StyleGAN [6] with the ADAM optimizer ($b_1 = 0.5$, $b_2 = 0.999$) and was trained for 11 days and 6 h. The learning rates of the generator and discriminator were both 0.0001. The stack size was 4. We set the number of critics to 1 and used leaky RLUs ($\alpha = 0.1$) for all nonlinearities, except the last one in the generator, where the trained model used the hyperbolic tangent. Styles are configured per image by two normalization parameters computed per feature map, and these parameters are computed deterministically based on a global style vector w . The input image was 256×256 pixels in size. We used a one-hot vector as the conditional vector. After the conditional vector was sent to input the image size, it was concatenated with an input image in the center of the encoder. We used StyleGAN as the discriminator. In this study, the generator was changed once after five updates of the discriminator. For training with a 32-bit bath size, we used Adam optimization and 20 iteration epochs. We created test photos with a resolution of 512×512 .

5. Results

The edition has numerous features that differ from the target, has a slightly different color, and can be guessed, especially in the regions where the inscriptions are sparse. However, most of the key architectural features are in place. This leads us to believe that we can design whole new lettering plans and create realistic looking facades from them. This could be very useful for an architect, who can sketch a design for a building and then quickly prototype the textures (perhaps several dozens since they are so easy to create).

What happens when you combine feature settings from two different images? Since style injection is performed separately on each layer, this can easily be done by inserting the w vector from building B1 into a set of layers and the w vector from building B2 into the remaining layers. This results in some planes being configured according to the parameters of building B1 and others according to those of building B2. This is what can be seen in the figure above. In each row, we take the leftmost image and swap a group of its style parameters with the image in the corresponding column. In the first three rows, we swap the coarse style parameters from the source; in the second two rows, we swap the medium ones; and in the last row, we imported only the fine style parameters from the alternate image “note that these are not real images, just various artificial drawings from the z -distribution, which are then converted to a w -vector using the mapping network” as shown in Figure 3.

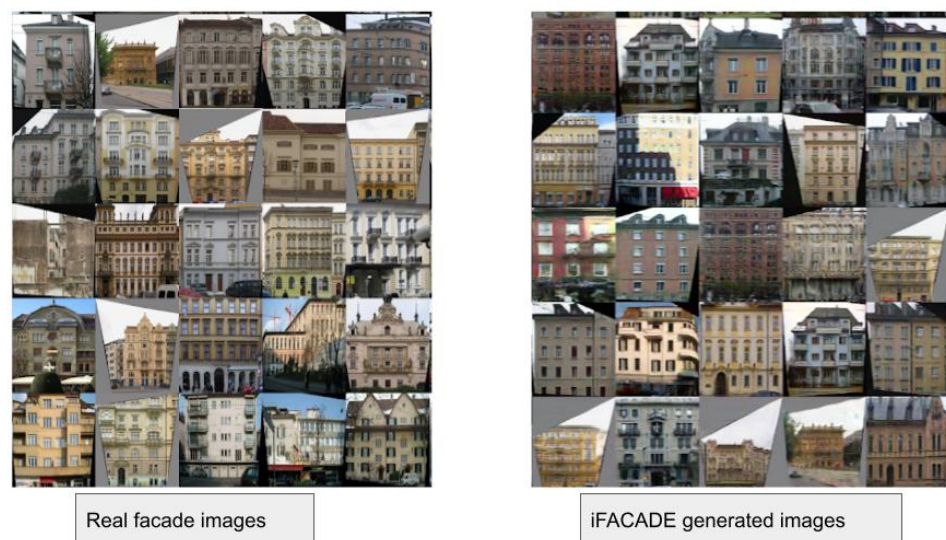


Figure 3. Example images generated from inputs containing condition information specifying (a) one, (b) two, and (c) three domains, respectively. All images are generated with the same style noise. Our model can generate images that include class information from an input that includes condition information that specifies multiple classes.

We find that the style-based generator (E) significantly improves the Fréchet input distance (FID), a metric for evaluating the quality of images created with a generative model, over the traditional generator (B), by almost 20 percent, which corroborates the extensive ImageNet measurements from parallel work (6, 5). Figure 2 shows an uncensored set of new images generated by our generator from the Flickr-Faces-HQ dataset (FFHQ). As confirmed by the FIDs, the average quality is high, and even accessories such as glasses and hats are successfully synthesized. All FIDs in this article are computed without the cropping trick, and we use it only for illustration in Figure 2 and the video. All images are generated at a resolution of 1024 as illustrated in Figure 4.

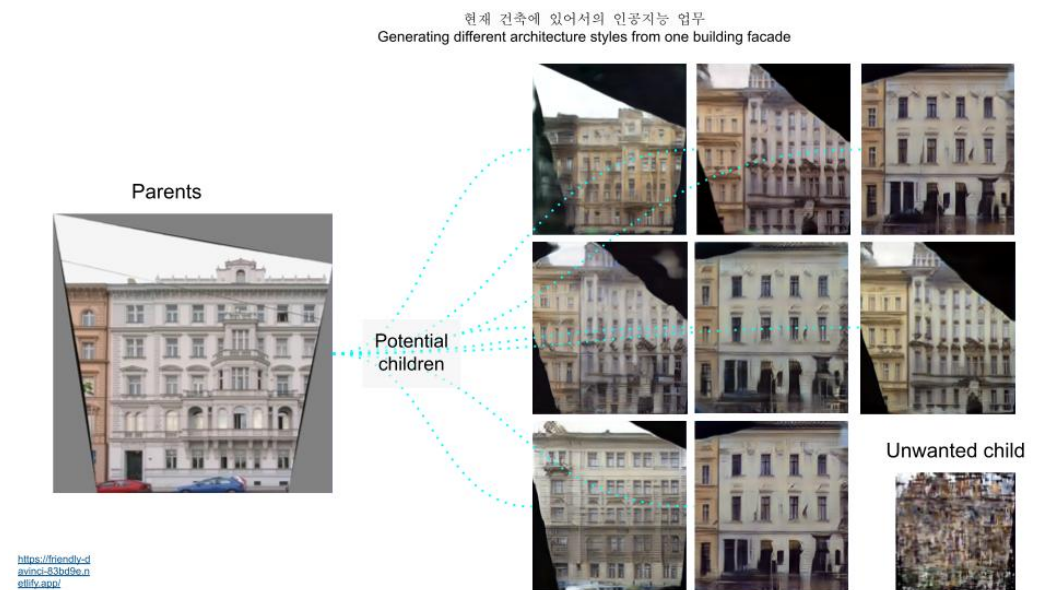


Figure 4. Generating different architecture styles from one building facade by manipulating latent space.

Examples of stochastic variation are as follows: (a) Two generated images. (b) Magnification with different realizations of the input noise. While the overall appearance is almost identical, the individual architectural elements are placed very differently. (c) Standard deviation of each pixel over 100 different realizations, showing clearly which parts of the images are affected by the noise. training significantly improves localization, as evidenced by improved FIDs in scenarios where multiple latents are mixed at test time. Figure 3 shows examples of images synthesized by mixing two latent codes at different scales. We can see that each subset of styles controls meaningful high-level attributes of the image.

6. Discussion

It is presented that iFACADE will be useful for designers in the early design phase to create new facades in a short time depending on existing buildings, saving time and energy. Moreover, building owners can use iFACADE to show their preferred architectural facade to their architects by mixing two architectural styles and creating a new building. Therefore, it is illustrated that iFACADE can become a communication platform in the early design stages between architects and builders.

Figure 4 shows that our model generates images from the input noise styles and the state vectors representing each class. We can see that our model can generate any class of images, thanks to the condition vectors. However, although the sounds and styles are fixed, the generated facades have distorted images of food because they have few round patterns. Figure 5 shows the images generated from a fixed style, a state vector, and randomly sampled noise. Each image is generated by our model with different random noises. We can see that random noise plays a role in representing differences, such as food topping. Figure 4 shows that the images generated by our model are generated simultaneously with state vectors representing two or more classes. We can see that our model can generate any feature, even if there are multiple condition vectors:

$$\text{targeted facade} = G(x) \text{variant of target facade} = G(x + \epsilon) \quad (5)$$

where G is the generator, and ϵ is a vector whose components are small numbers that are sampled randomly.

The iFACADE does not work in neighborhoods that do not follow the modular architectural style because the generated facades have characteristics of reference facades. It also cannot be applied to new neighborhoods where buildings do not yet exist, as it relies

on prefabricated facades as a reference to generate a new architectural style. In addition, iFACADE cannot control the specific characteristics of individual building elements. For example, it cannot change the style of the window openings or the height of the porch according to the user's preferences. For this reason, iFACADE will control each individual building element according to the user's wishes in the future. To address these limitations, this research can utilize StyleGAN 2 [23], which provides better control on the latent space and image quality. In addition, the trained dataset could be increased with different architecture styles to give options to the user to choose the desired style according to their preferences. Furthermore, the user could upload a collection of images, and the model could perform architecture style transfer to the targeted facade using [7,24]. This method needs to be tested to find the applicability and quality of generated facade images.



Figure 5. Evaluation criteria of generated facade images using trained model.

To measure the quality of generated facade images, quantitative measures are used to evaluate the trained model and generate new facades. A comparison between real facade images and generated facade images was conducted. The main architectural characteristics were analyzed. The evaluation process depends on the following features: number of floors, walls, windows, clarity of the facade, and materials. The real facade image had 4 to 5 floors, and the generated facade images also maintained the same range of floors. The front wall of the real facade images used were flat, and the generated facade images had the same features. The windows in the real facade were in the range of 3, 4 and 5 windows per floor and the generated facade also maintained the same number of windows per floor as depicted in Figure 5. The real images used had 720 pixel clarity. However, the generated facade lost 15 percent of its clarity due to the limited graphics card used for training. It is arguable that using fewer images for training will give better quality images, but it will limit the variation of facade features shown in Table 1. The materials of the facade in the real and generated images are stone and marble with materials of off-white colors. The number of floors remained limited. The facade wall in the generated images is of an apartment and parallel to the front view, which follows the real image. The window details are varied because they belong to all real images and not to a target image in the trained data-set. The clarity of the facade is good enough for a conceptual illustration. The material and the color of the facade vary each time the model runs to generate a new image as illustrated in Figure 5.

Table 1. Evaluating the generated facade image architecture features.

Item	Real Facade	Generated Facade
number of floors	4 floors 5 floors	4 floors 5 floors
walls	plain facade wall	plain facade wall
windows	(3, 4, 5) per floor	(3, 4, 5) per floor
clarity of the facade	very clear	15 percent blurry
material	stone and marble	variety of off white color

7. Conclusions

In this research, a machine learning tool is proposed that can mix the facade style of two reference facade images and generate a unique facade design that can be used in urban redensification. Experiments show that the proposed model can generate different facade images with conditional vectors. The proposed tool could also be useful for facade designers, as it is able to quickly convert an architect's simple building sketch into a prototype texture. The main contributions of this research are that (a) iFACADE can mix facade styles to generate a new facade image with its own features, (b) the latent space of the generated image can be used to control the style details of the architectural elements of the output image, and (c) our proposed generator can generate images representing each feature from the condition information defining multiple classes. In the future, iFACADE could be extended to a mobile app that can host a trained model, where the user can simply take a photo of two desired building views, and the app generates a mixed-style architectural facade. Then, the user can project the generated facade onto the unbuilt space using augmented reality technology. Thus, our proposed generator based on conditional style has great potential to solve problems that are still unsolved in food images. Finally, we hope that our proposed model will contribute to further studies on architectural design. Moreover, iFACADE should be extended to generate 3D facade elements in addition to 2D images so that it can be directly used in the early design stages of architecture and increase automation.

Author Contributions: Conceptualization, A.K.A.; methodology, O.J.L.; software, A.K.A.; validation, A.K.A.; formal analysis, O.J.L.; investigation, O.J.L.; resources, O.J.L.; data curation, A.K.A.; writing—original draft preparation, A.K.A.; writing—review and editing, A.K.A.; visualization, A.K.A.; supervision, O.J.L.; project administration, O.J.L.; funding acquisition, O.J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Haenglim for Architecture and Engineering Company.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work is supported by Haenglim Architecture and Engineering Company. The first author would like to thank Chung Ang University and the Korean Government Scholarship program.

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

References

- Adamus-Matuszyńska, A.; Michnik, J.; Polok, G. A Systemic Approach to City Image Building. The Case of Katowice City. *Sustainability* **2019**, *11*, 4470. [\[CrossRef\]](#)
- Talen, E. *City Rules: How Regulations Affect Urban Form*; Island Press: Washington, DC, USA, 2012.
- Touloupaki, E.; Theodosiou, T. Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization—A review. *Energies* **2017**, *10*, 637. [\[CrossRef\]](#)
- García-Ordás, M.T.; Benítez-Andrades, J.A.; García-Rodríguez, I.; Benavides, C.; Alaiz-Moretón, H. Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data. *Sensors* **2020**, *20*, 1214. [\[CrossRef\]](#) [\[PubMed\]](#)
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [\[CrossRef\]](#)
- Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.
- Almahairi, A.; Rajeswar, S.; Sordoni, A.; Bachman, P.; Courville, A. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *arXiv* **2018**, arXiv:1802.10151.
- Zhu, J.Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A.A.; Wang, O.; Shechtman, E. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers: Burlington, MA, USA 2017; pp. 465–476.
- Zhang, Y.; Yin, Y.; Zimmermann, R.; Wang, G.; Varadarajan, J.; Ng, S.K. An Enhanced GAN Model for Automatic Satellite-to-Map Image Conversion. *IEEE Access* **2020**, *8*, 176704–176716. [\[CrossRef\]](#)
- Bulat, A.; Yang, J.; Tzimiropoulos, G. To learn image super-resolution, use a gan to learn how to do image degradation first. In Proceedings of the European Conference on Computer Vision (ECCV), 2018, Munich, Germany, 8–14 September 2018; pp. 185–200.
- Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative adversarial text to image synthesis. *arXiv* **2016**, arXiv:1605.05396.
- 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, 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
- 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, 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 2223–2232.
- Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8789–8797.
- Mao, Q.; Lee, H.Y.; Tseng, H.Y.; Ma, S.; Yang, M.H. Mode seeking generative adversarial networks for diverse image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 1429–1437.
- Abdal, R.; Qin, Y.; Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In Proceedings of the IEEE International Conference on Computer Vision, 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 4432–4441.
- Collins, E.; Bala, R.; Price, B.; Susstrunk, S. Editing in Style: Uncovering the Local Semantics of GANs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, Seattle, WA, USA, 14–19 June 2020; pp. 5771–5780.
- Horita, D.; Shimoda, W.; Yanai, K. Unseen food creation by mixing existing food images with conditional stylegan. In Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management, 2019, Nice, France, 21–25 October 2019; pp. 19–24.

19. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
20. Tylecek, R. *The Cmp Facade Database*; Technical Report, CTU-CMP-2012-24; Czech Technical University: Prague, Czech Republic, 2012.
21. Korc, F.; Förstner, W. *eTRIMS Image Database for Interpreting Images of Man-Made Scenes*; Technical Report, TR-IGG-P-2009-01; Department of Photogrammetry, University of Bonn: Bonn, Germany, 2009.
22. Braun, M.; Krebs, S.; Flohr, F.; Gavrilu, D.M. The eurocity persons dataset: A novel benchmark for object detection. *arXiv* **2018**, arXiv:1805.07193.
23. Viazovetskyi, Y.; Ivashkin, V.; Kashin, E. Stylegan2 distillation for feed-forward image manipulation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XXII 16*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 170–186.
24. Lin, C.T.; Huang, S.W.; Wu, Y.Y.; Lai, S.H. GAN-based day-to-night image style transfer for nighttime vehicle detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 951–963. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.