Type Investigation in the Form of High-Rise Building Using Deep Neural Network

Jinmo Rhee(✉) and Pedro Veloso

Carnegie Mellon University, Pittsburgh, PA 15206, USA
jinmor@cmu.edu, pveloso@andrew.cmu.edu

Abstract. In this paper we propose a deep learning (DL) method to investigate existing types of high-rise buildings and to generate new ones. We collected data comprehending diverse forms of high-rise building from major cities in the world to train a generative DL model (IntroVAE) to capture morphological features. After clustering the features, we can distinguish types of high-rise buildings and use that information to generate novel high-rise building forms. This research demonstrates that generative DL models can uncover the latent types of architectural form in large datasets and can expand the typological interpretation of complex architectural forms. Besides, we demonstrate the potential of the proposed DL method for building massing design by developing a proposal of a high-rise building form based on three techniques: exploration, synthesis, and interpolation.

Keywords: Deep learning · High-rise building · Typology · Form · Neural network

1 Introduction

High-rise building is a representative architectural type of contemporary city and has its own morphological features, which can be noticeably distinguished from other architectural types such as church, residential vernacular, market, etc. These features are derived from design principles and styles and are key to the formal and typological identity of a high-rise building. Typically, high-rise buildings have been categorized by their major characteristics such as their main towers shape [1], types of structural systems [2], the position of their circulation core, their height [3], their program, or the combination of some of these [4].

In this research, we contribute to the existing taxonomies of building types by establishing formal analysis of a large database of high-rise building. By using computational methods to analyze and cluster a database encoding the form of high-rise buildings, we expect to uncover latent building types and provide opportunities to expand architectural creativity by re-interpreting and transforming their forms.

Morphological investigation of high-rise building based on the analysis of geometric features is not straightforward, because high-rise building form requires numerous and complex variables [5] representing various features and aesthetic elements derived from sources such as context and architectural styles. There is no standardized way to

D. Gerber et al. (Eds.): CAAD Futures 2021, CCIS 1465, pp. 175–189, 2022.
https://doi.org/10.1007/978-981-19-1280-1_11

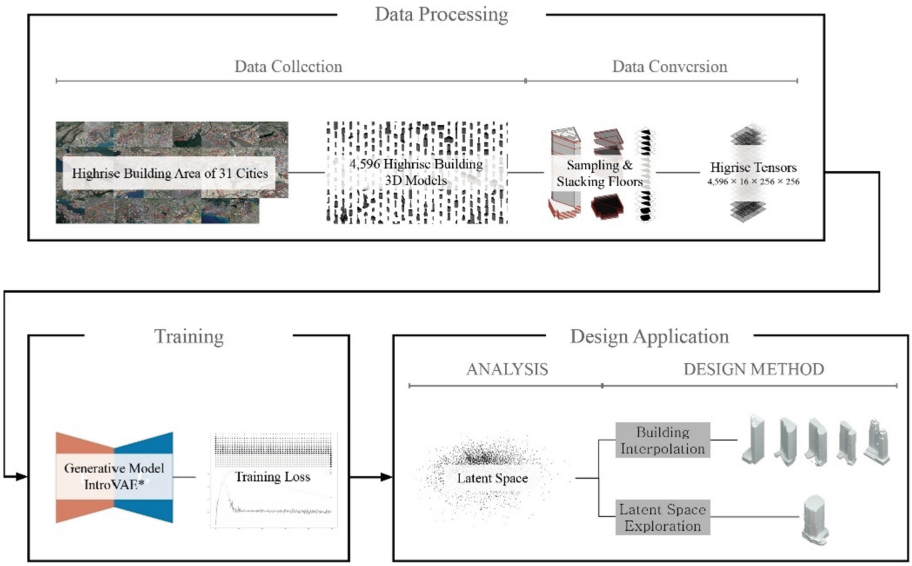

Fig. 1. Overall process of type investigation in the form of high-rise building and its design using deep neural networks.

represent these variables. For example, with the adoption of parametric and geometric modeling techniques by contemporary practitioners, the design of high-rise buildings has adopted curvy and variational shapes that are noticeably different than the rigid shapes of conventional design. Those difference makes more difficult to establish a representation for comprehensive formal analysis and typological studies of high-rise buildings.

With a system that can represent complex morphological features, we can thoroughly examine the form of varied high-rise buildings, address new formal types by re-clustering the features, and potentially explore the new formal identity in the design of high-rise buildings. Therefore, this research aims to implement generative deep learning (DL) model for grasping geometric features, discovering types of high-rise buildings, and supporting new design practices for the massing design of high-rise buildings.

This research is constituted by three steps: data collection and pre-processing, training, and design application (Fig. 1).

In the data collection and pre-processing step, we collected three-dimensional data of high-rise buildings, converted each building into a series of two-dimensional images by horizontal slicing, and stored them in a deep tensor.

We used this tensor database of high-rise building forms to train a generative DL model called autoencoder, which contains two parts: an encoder and a decoder. This encoder compresses the tensor representation of each building into a lower-dimensional vector. The decoder learns how to reconstruct the original tensor based on this vector. By reducing the tensor representation to a lower-dimensional representation (embedding) the model also enables the discovery of the dominant morphological features in the dataset. These embeddings can be used to categorize forms of high-rise buildings into several representative types.

Besides, the decoder can not only reconstruct existing buildings but also generate new ones that are in the parameter space of the embedding. Based on the manipulation of the embedding of a trained model, we investigate three different techniques to generate a high-rise building form: exploration, synthesis, and interpolation. We introduce the potential of these techniques to support architectural creativity by developing the schematic design of two high-rise buildings.

2 Machine Learning-Integrated Architectural Design

The recent innovations and increased accessibility of DL has supported the integration of artificial intelligence technology to architectural design process. Besides, it also shed a new light on the importance of architectural data storing, collection and processing for decision-making in architecture. This section reviews research projects that used generative models from DL and image and voxel-based representations for the generation of architectural forms.

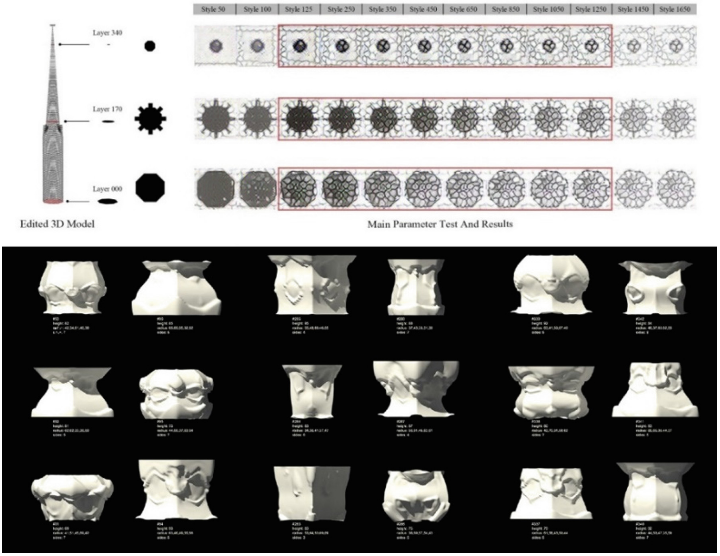

Fig. 2. (Top) Generated stylized plans with gradually changed style weights [6] and (Bottom) Capitals automatically design with machine learning [7].

The Spire of AI [6] project created a new form of architecture using Style Transfer [8] techniques from DL. The authors contoured building forms and obtained the images of horizontal sections of the forms. Then, they trained a Neural Style Transfer model with the section images. After training, users can apply a specific style to the sections and convert the pixel values of the style-transferred sections into voxels, which is a three-dimensional representation of pixels. By vertically stacking the voxel values, they can reconstruct the shape of a building with the style (Fig. 2). The idea of contouring the sections was key to represent the three-dimensional information of the form for image-based DL. Similar idea can be found in Fresh Eyes [9]. However, in this project, DL is only applied to transferring a style to other images, not the process of learning the morphological feature of buildings or synthesizing the overall form of buildings.

The automation of capital design with machine learning is introduced in the research of Artificial Intelligence Aided Architectural Design [7]. The authors trained artificial neural networks based on the detailed configuration of the Roman Corinthian order capitals. The input data format was composed of matrices that includes the sample coordinate values, surface normal vector and volume center plane deviation. The output data are displacement values. By reconstructing a three-dimensional model of an order based on this predicted displacement values, they can generate three-dimensional variations of the new capital forms based on the given input parameters, both purposeful and random (Fig. 2). This research shows the potential of integrating DL to architectural design activities. Both repeatable and predictable design activities can be easily replaced by machine learning tools in the first place, by teaching the system decision-making based on the work performed by the architects.

Voxel-based representation is another way to construct form dataset. Geometric Interpolation of Building Types project [10] illustrates a method to represent a building within a fixed number of voxels and their vectorized connections. For investigating and interpolating building types, this project employed voxels as scaffoldings of building representation. By defining the connection between the voxel points, building can be represented by its edges. The information of edges is stored in the format of tensor which deep neural network can grasp. Using parametric augmentation, the authors create a dataset based on two "types": an abstract castle structure inspired by John Hejduk's architecture and the famous building China Central Television (CCTV) designed by OMA.

David Newton's high-rise building synthesis using a generative DL model [11] is another project to use voxel-based representation of architectural form. He collected "500 building massing models located in downtown New York City" [11] and convert them into "1 × 256 × 256 voxels" [11]. Using one of the three-dimensional deep generative models, 3D-IWGAN, he trained the formal features of the buildings and was able to synthesize a new building form.

These precedents show the importance of defining a specific representation to capture features of architectural form for generation. Our approach uses a novel variation of stacked section images that considers specific aspects of our problem, such as building scale and parts. Similarly to [10], our research also emphasizes the importance of capturing typological features in the latent space. However, instead of synthesizing the dataset based on two building types, we use our stacked representation to explore a vast dataset of high-rise buildings from different cities.

3 Dataset

In this section we discuss our latest method for encoding of building forms into tensors. We collected three-dimensional data of high-rise buildings, converted them into two-dimensional images by slicing their three-dimensional forms, and finally a deep tensor dataset was created by stacking these sliced images.

3.1 Data Collection

The research started with the collection of three-dimensional form data of existing high-rise buildings from manually selected downtowns of 31 cities around the world and for each city focused on specific areas that have a concentration of high-rise buildings: New York City, Chicago, Atlanta, Los Angeles, Miami, Philadelphia, Pittsburgh, Boston, Seattle, San Francisco, San Diego, Houston, Dallas, Baltimore, Detroit, Indianapolis, Denver, Vancouver, Toronto, London, Paris, Riyadh, Dubai, Abu Dhabi, Hongkong, Shanghai, Taipei, Bangkok, Singapore, Honolulu, Sydney.

To facilitate a large collection of data, we automated the process of scraping high-rise building three-dimensional models from OSM (Open Street Map).

The general height threshold for high-rise building is of 25 m [12]. However, this definition implies that almost every building with more floors than 5 stories is high-rise building. If we naively accepted this threshold, it might be difficult to grasp the distinctive morphological characteristics of a high-rise building. Considering the properties of the collected dataset and intention to manifest the formal characteristics of a high-rise building in this research, we set the threshold into 70 m and picked a proper value which can provide a total of 4,956 high-rise buildings formatted as three-dimensional OBJ models. We used Rhinoceros and Grasshopper for handling and modeling three-dimensional data.

3.2 Data Processing

In order to train the image-based generative model, we developed a technique to represent each three-dimensional building as a set of two-dimensional images. The technique involves slicing and sampling 16 floors of each building and represent them as figure-ground diagrams. To do this, we sliced each building horizontally using the 3 m standard floor to floor height adopted in OSM, and put all slices into three groups based on the range of their relative heights (i.e. 0–33%, 33–66%, 60–100%). Then, we selected the first 6 floors of the first group, the 5 floors in the middle of the second group, the first two floors of the third group, and the last 3 floors of the third group. The sampling of the first six floors of the first group reflects that most buildings have the "podium" typology where their overall forms tend to have larger bases. The sampling of five floors in the middle of the second group reflects the prevailing form typically found in the midsection of high-rise buildings. The sampling of the first two floors of the third group reflects the tendency for high-rise buildings to taper towards the top. The sampling of the last three floors of the third group reflects the tendency for high-rise buildings to have a spire at the top.

We have determined the sampling of 16 floors for encoding each building as a tensor in this research. The sampling number can be larger to represent the original form of three-dimensional high-rise building model more accurately.

Each slice of the high-rise building is a diagrammatic representation of a floor plan boundary encoded as a tensor shape of shape $1 \times 256 \times 256$. By stacking 16 of these slices, we can establish a tensor to represent a high-rise building's morphological features and train a DL model with convolutional layers. The total tensor size for the entire dataset of 4,596 buildings is $4596 \times 16 \times 256 \times 256$ (Fig. 3).

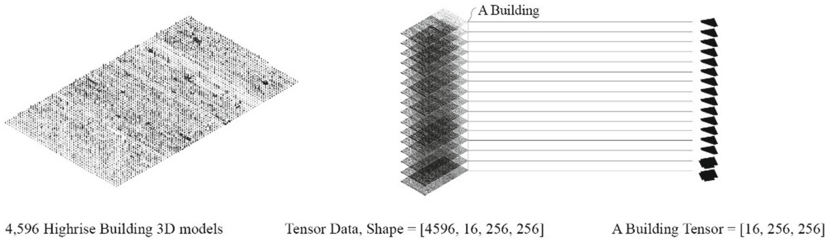

Fig. 3. Data conversion of 3D model to tensor through image format.

4 Training

4.1 Model

One of the most popular generative DL models is Generative Adversarial Network (GAN). According to [13]:

It consists of two networks: the generator network $Gen(z)$ maps latent z to data space while the discriminator network assigns probability $y = Dis(x) \in [0,1]$ that x is an actual training sample and probability $1 - y$ that x is generated by our model through $x = Gen(z)$ with $z \sim p(z)$. The GAN objective is to find the binary classifier that gives the best possible discrimination between true and generated data and simultaneously encouraging Gen to fit the true data distribution. We thus aim to maximize/minimize the binary cross entropy:

$$\mathcal{L}_{\mathcal{GAN}} = \log(Dis(x)) + \log(1 - Dis(Gen(z))) \tag{1}$$

with respect to Dis/Gen with x being a training sample and $z \sim p(z)$.

Another of the popular generative DL models is Variational Autoencoder (VAE). According to [13]:

It consists of two networks that encode a data sample x to a latent representation z and decode the latent representation back to data space, respectively:

$$z \sim Enc(x) = q(z|x), \quad x \sim Dec(z) = p(x|z) \tag{2}$$

The VAE regularizes the encoder by imposing a prior over the latent distribution $p(z)$. Typically, $z \sim N\ (0, I)$ is chosen. The VAE loss is minus the sum of the expected log likelihood (the reconstruction error) and a prior regularization term:

$$\mathcal{L}_{\text{VAE}} = -E_{q(z|x)}\left[\log\frac{p(x|z)p(Z)}{q(z|x)}\right] = \mathcal{L}_{\text{llike}}^{\text{pixel}} + \mathcal{L}_{\text{prior}} \tag{3}$$

With

$$\mathcal{L}_{\text{llike}}^{\text{pixel}} = -E_{q(z|x)}\big[\log p(x|z)\big] \tag{4}$$

$$\mathcal{L}_{\text{prior}} = D_{KL}(q(z|x)\|p(z)) \tag{5}$$

where DKL is the Kullback-Leibler divergence.

In terms of the model selection between GAN and VAE for this experiment, our main challenge was in the tradeoff between blurry images and trained latent space. Nowadays, due to the tremendous development of GAN, GAN can generate sharp synthesized images but have less freedom to explore the latent space of the model; with new input data, the latent vector cannot be preserved, and the model must be recalculated to fit the new data. In addition to characteristics of latent space, GANs are hard to train because of its unstable training process and mode collapse [14]. On the other hand, VAE usually generates blurry images, compared to the image quality generated by GAN, but it allows for more freedom to explore the latent space. Since the latent space fits to the entire given dataset, exploring it does not require new fitting calculation. Due to these tradeoffs, we selected IntroVAE as a hybrid model that conciliates the consistent latent space with high quality images from GAN (Fig. 4) [15].

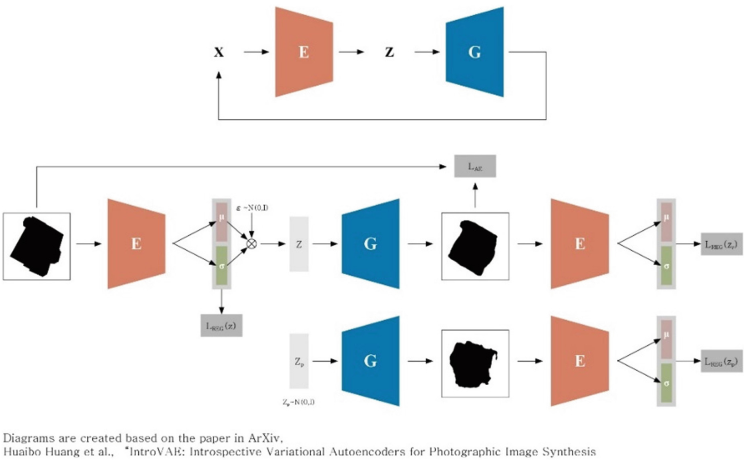

Fig. 4. Architecture of IntroVAE, re-drawn illustration fit to the dataset.

This model requires two parts in training to generate an image: one part to discriminate the generated samples from the training data, and another part to mislead the discriminator. Specifically, this model has the approximate inference model of VAEs (encoder) as the discriminator of GANs and the generator model of VAEs (decoder) as the generator of GANs. In addition to performing adversarial learning like GANs, the inference and generator models are trained jointly for the given training data to preserve the advantages of VAEs [15].

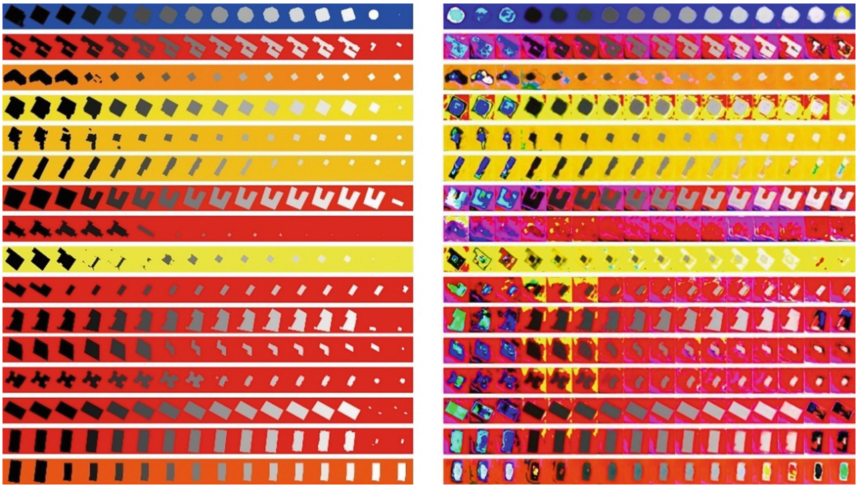

Fig. 5. Results comparison between ground truth (left) and predicted (right) images from dataset with three channels.

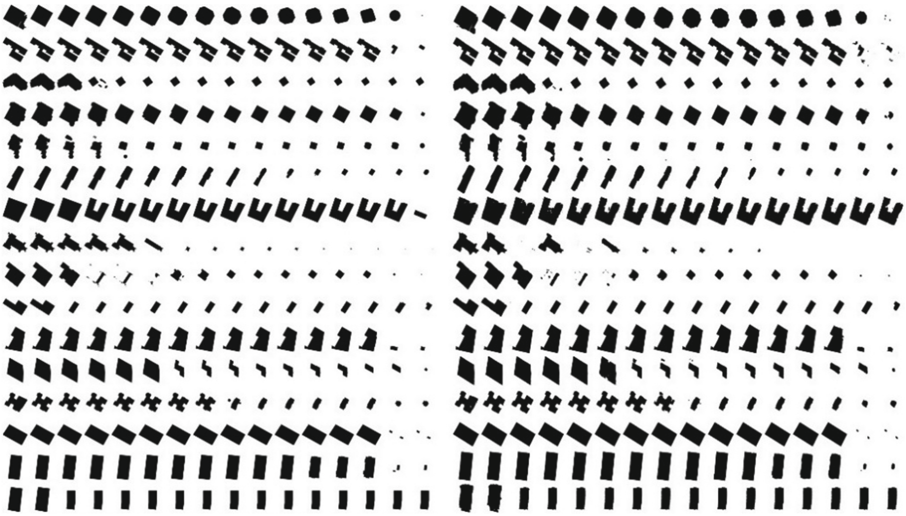

Fig. 6. Results comparison between label (left) and predicted (right) image from dataset with one channel.

4.2 Training

On the onset of the training process, we tested a dataset that included actual height of buildings represented by background colors. This dataset has 3 channels (i.e. RGB) instead of 1. Our initial was that color information would be helpful for easing learning by providing more distinguished features of each building and floor. After training the model based on this dataset for 500 epochs, based on comparison between the original

(a) and decoded (b) images of test dataset (Fig. 5), we found too much randomness, noise, and often incorrect colors on the images. Color information was difficult to learn, because floor plan shape and height value did not have a strong correlation.

To overcome the challenge with colors, we switched to a grayscale representation of the buildings and trained the model for 500 epochs. Based on comparison between the original (a) and decoded (b) images of test dataset (Fig. 6), the decoded images were still a little bit noisy and had minor errors, but they had higher accuracy and sharp edges. For the final learning, the hybrid model was trained on a computer with the following specifications: 'Intel(R) Core (TM) i7-8700k @ 3.70 GHz', 64 GB memory, and two GTX-1080ti graphic processing units. It took almost 200 h to train the data for 1300 epochs. Learning rate of encoder and decoder were 2e−4, lambda for L1 loss was 100, hidden dimension was 10, beta1 and beta2 were 0.5 and 0.999 respectively, and batch size was 128.

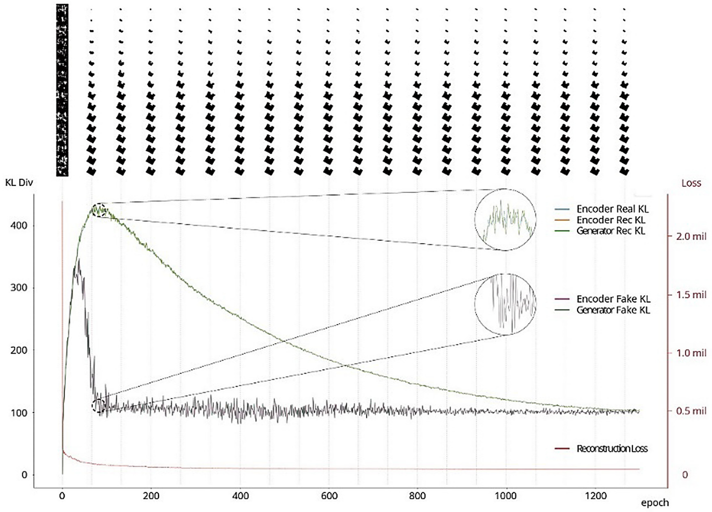

Fig. 7. Losses during training process.

Figure 7 illustrates the quality of the reconstructed sample building diagrams from the test dataset with regard to the losses of the reconstruction and the Kullback–Leibler (KL) – divergences [16]. After almost 20 epochs, the losses started to converge to a stable stage in which their values fluctuate slightly around a balance line [15]. KL divergence between the latent distribution of the decoder and a normal distribution started to converge after almost 100 epochs. After this point, the KL-divergences continuously decreases with little fluctuation of their values. KL divergences of encoder and decoder's face image also decreases and converge after almost 60 epochs. Compared to these divergences from 0 to 60 epochs, the width of fluctuation of these divergences are drastically smaller. Encoder and decoder's reconstruction and real image KL-divergences converged to fake image's KL-divergences.

4.3 Typical Form of High-Rise Building Using Latent Space

Encoded vectors of each building tensor keep the morphological features of the building with the reduced data dimension: every building can be represented with 10 floats in the latent space by the reduced features. However, 10-dimensional space is still hard to visualize. In order to visualize latent space which has 10 dimensions in to 3-dimensional space, we employ t-SNE (t-distributed stochastic neighbor embedding) [17] to reduce dimensionality. The hyper-parameters are: perplexity 35, learning rate 100, iterations 1500. All building in the dataset can be placed and represented in 3-dimensional space as a data point. A data point in the latent space is a high-rise building. The location of the point represents its morphological characteristics, and the distance among the points represents the degree of morphological similarity. The shorter the distance, the more similar form of high-rise buildings.

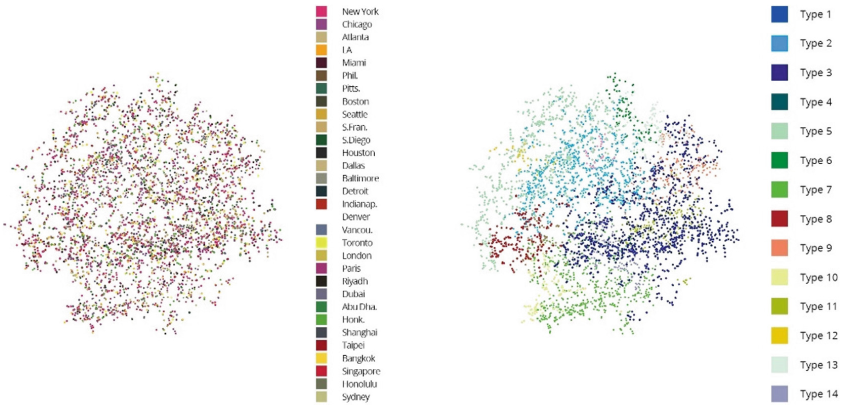

Fig. 8. Visualizations of the latent space of encoded high-rise dataset by cities (left) and types (right).

We discovered a total of 14 types of high-rise buildings by clustering the data points of the buildings in the latent space with Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [18]. Figure 8 demonstrates two different visualizations of the latent space of encoded high-rise dataset: by cities and types.

By tracking the nearest data point of each cluster's center, the typical form of each type can be investigated (Fig. 9). Type 1 has a parallel configuration of a tower and podium. Type 2 is the simplest box shape tower. Type 3 is a twin tower sharing a podium. Type 4 is a narrow rectangular tower with many irregularities. Type 5 is a cake-shaped tower without a podium. Type 6 is a tower with subtraction. Type 7 is a circular-shaped tower. Type 8 has also parallel configuration of a tower and podium like type 1, but the podium is large and tall. Type 9 is a simple slim tower. Type 10 is similar to type 5 with a podium. Type 11 is similar to type 4 with a podium. Type 12 is like type 8 with a podium. Type 13 is like type 4 with lower irregularities. Type 14 is a simple tween tower without a podium (Table 1).

Fig. 9. Typical form of each type of high-rise building.

Table 1. Feature comparison of each type.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Profile Shape	**Rect.**	■	■		■	■	■		■	■	■	■	■	■	
	Circle			■				■							■
Vertical Shape	**Regular**	■	■	■			■	■	■	■			■		■
	Irregular				■	■					■	■		■	
Tower Ratio	**Normal**	■	■	■			■	■	■		■				■
	Narrow				■	■						■		■	
	Skinny									■			■		
Number of Tower	**1**	■	■		■	■	■	■	■	■	■	■	■	■	
	2			■											■
Podium	**0**		■		■	■	■	■		■				■	■
	1	■		■					■		■	■	■		

Fig. 10. Interface of Deeprise for high-rise building form analysis and generation.

5 Design Application

We developed this model as a design plug-in prototype of Grasshopper in Rhinoceros, a popular three-dimensional modeling tool in architecture and product design. The prototype, called "Deeprise", has its own interface (Fig. 10) and deploys the trained model with the learned high-rise building form for design and runs it in back-end to provide three different approaches for high-rise building form generation.

5.1 Three Different Method for Form Generation

Designers can randomly generate a building by changing the sliders in the interface. The slider will change the seed value of random function and produce a random vector with the same length as the hidden dimension of the model. Figure 11 shows a schematic design example of high-rise building based on the randomly generated building forms from Deeprise interface. After retrieving the contour lines from Deeprise, designers can use lofting to create a buildings form.

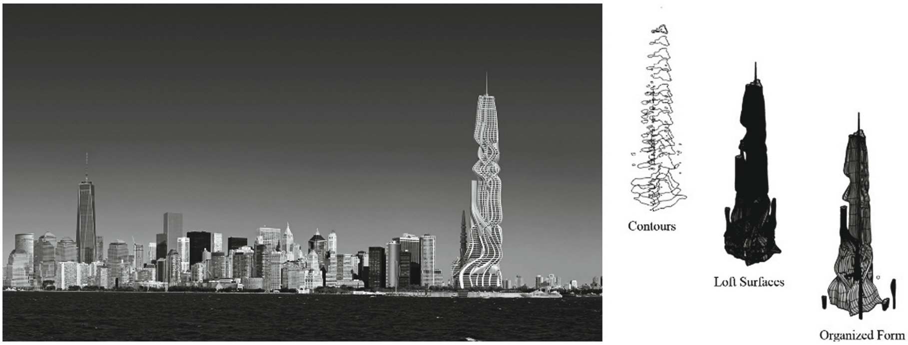

Fig. 11. Design example using random exploration from Deeprise and its design process.

The second method to generate a high-rise building form is synthesis. Designers can synthesize a vector in the latent space by changing slider and assigning a value to each dimension of the latent space. This method allows designers to control the form of a high-rise building with higher precision. After they roughly explores the latent space to discover a proper form of a high-rise building, they can digitally sculpt the form in detail by changing sliders a bit (Fig. 12).

The last method for generation of a high-rise building form is interpolation. Instead of editing a form of high-rise building by manually changing values in a vector, this method can generate a series of interpolated forms between two buildings. Through the SLERP (Spherical Linear Interpolation) function, it generates hybrid forms of two buildings with different combination ratios of them (Fig. 13). This method is significant for architectural design, because designers can mix two different types of buildings by changing the parameters of the interpolation without having to manage the larger parameter space of the model directly. Figure 14 demonstrates a design scenario where the form consists of 75% of a building and 25% of another.

Fig. 12. Series of synthesized forms by slight changes of Z-vectors.

Fig. 13. Series of interpolated forms between two buildings.

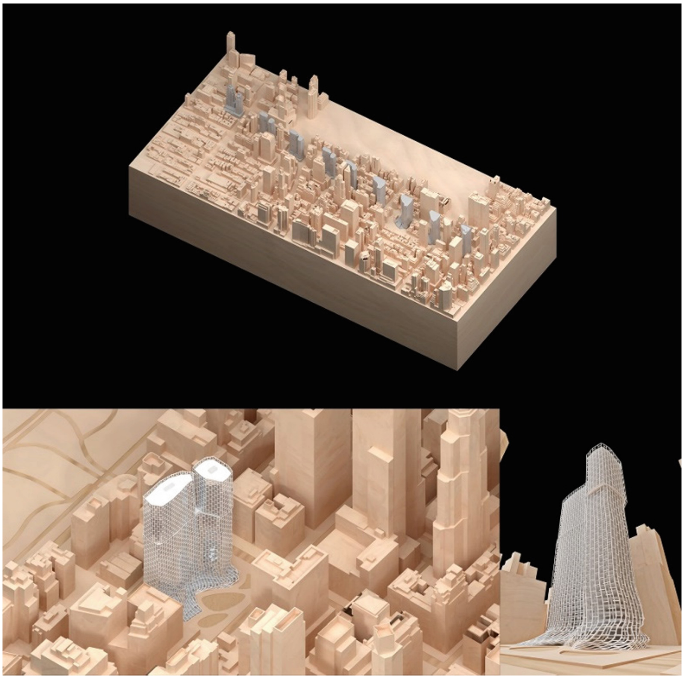

Fig. 14. Design example using interpolation.

6 Conclusion

We have categorized high-rise buildings according to their morphological characteristics, discovered typical types of them, and generate new high-rise buildings using a generative DL model. To train the model, we used a dataset with custom representation of morphological information of high-rise buildings. Based on this, we conducted design experiments with the trained model to explore new high-rise building forms for schematic architectural design.

6.1 Discussion and Contribution

This research illustrates a new methodological approach for the analysis of architectural morphology in design. The new design types are drawn by advanced statistical techniques applied to databases with three-dimensional information of high-rise buildings. The method of representation and interpretation of architectural form in this research supports a complex and thorough analysis of the existing buildings. With this method, designers can expand their knowledge of architectural forms by uncovering the latent form and types from the real world. Furthermore, this research demonstrates the potential that the generative DL model can be used not only for directly creating the design results but also for analyzing design objects. Rather than independently generating high-rise building forms with a trained model, this research addressed the integration of formal analysis and generation through the latent space. In this sense, DL could be developed into an efficient tool to help designers to analyze their design problems and generate solutions that are both creative and grounded on real-world data.

Lastly, the expanded exploration coverage of high-dimensional design solution space through DL can provide vast opportunities to discover new architectural forms. Instead of interpreting morphological features of several high-rise buildings and abstracting their formal principles, the generation methods in this paper demonstrates the potential to synthesize a new high-rise form by exploring the design solution space where complex formal principles exist.

6.2 Challenges and Future Study

There are three aspects to be considered for future steps: diversity of building types, using data that extrapolate geometry, and learning directly from a three-dimensional representation of buildings.

This research only used high-rise buildings to capture the morphological features of architectural form. If more architectural types, such as churches, markets, airports, etc., are available, the potential of architectural design exploration with DL can be expanded. For this goal, not only different DL models but also different ways to represent architectural form into learnable data format should be explored.

Besides, we intend to extrapolate geometric and physical characteristics of buildings in order to explore other criteria and relationships that affect the built form. By integrating geometric and other social, economic, cultural, and environmental data, form can be more broadly interpreted as a phenomenon of human activities. As a result, Deeprise will be one step closer to analyze morphological principles and features related to our society.

Technically, the most challenging part of this research is converting three-dimensional form data into stacks of two-dimensional images for learning and then reconstructing three-dimensional forms. These conversion and reconstruction process incur to a certain amount of loss of the original morphological features. Specifically, since the original height information of each floor boundary image was approximated by relative values, the reconstructed form was segmented by extruded geometries in the modeling software.

In the recent years, deep learning research has been pushing the boundaries of representation beyond the structured domain of images. These advancements enable the design of deep learning workflows that do not require conversions between three and two-dimensional data. Some of the examples include geometric representations such as voxels [19], meshes [20], and point clouds [21], which are available in DL libraries such as Pytorch3D [22]. Learning with these representations showed good performance in well-structured geometric domains with small scale variance, such as in models of bodies and faces. Applying these techniques to the domain of building morphology will require systematic exploration and experimentation.

References

1. Szolomicki, J., Golasz-Szolomicka, H.: Technological advances and trends in modern high-rise buildings. Buildings **9**, 193 (2019). https://doi.org/10.3390/buildings9090193
2. Kovacevic, I., Dzidic, S.: Modern structural concepts for high-rise buildings. In: САВРЕМЕНА ТЕОРИЈА И ПРАКСА У ГРАДИТЕЉСТВУ (2018).https://doi.org/10.7251/STP1813549K
3. CTBUH, T.C. on T.B. and U.H.: Height Criteria (2020). https://www.ctbuh.org/resource/height
4. Moussavi, F., Ciancarella, M., Scelsa, J.A., Crettier, M., Kilalea, K. (eds.): The Function of Style. Actar, Barcelona (2014)
5. Donath, D., Lobos, D.: Massing study support. A new tool for early stages of architectural design, p. 8 (2008)
6. Ren, Y., Zheng, H.: The spire of AI - voxel-based 3D neural style transfer. In: Holzer, D., Nakapan, W., Globa, A., Koh, I. (eds.) RE: Anthropocene, Design in the Age of Humans - Proceedings of the 25th CAADRIA Conference - Volume 2, Chulalongkorn University, Bangkok, Thailand, 5–6 August 2020, pp. 619–628. CUMINCAD (2020)
7. Cudzik, J., Radziszewski, K.: Artificial intelligence aided architectural design. In: AI for Design and Built Environment, pp. 77–84 (2018)
8. Gatys, L., Ecker, A., Bethge, M.: A neural algorithm of artistic style. arXiv (2015). https://doi.org/10.1167/16.12.326
9. Steinfeld, K., Park, K., Menges, A., Walker, S.: Fresh eyes. In: Lee, J.-H. (ed.) CAAD Futures 2019. CCIS, vol. 1028, pp. 32–46. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-8410-3_3
10. De Miguel Rodríguez, J., Villafañe, M., Piškorec, L., Sancho Caparrini, F.: Generation of geometric interpolations of building types with deep variational autoencoders. Des. Sci.**6**, E34 (2020).https://doi.org/10.1017/dsj.2020.31
11. Newton, D.: Generative deep learning in architectural design. Technol.|Architect. + Des. **3**(2), 176–189 (2019). https://doi.org/10.1080/24751448.2019.1640536
12. Craighead, G.: High-Rise Security and Fire Life Safety, 3rd edn. Butterworth-Heinemann (2009)
13. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. arXiv:1512.09300 [cs, stat] (2016)
14. Thanh-Tung, H., Tran, T.: On catastrophic forgetting and mode collapse in generative adversarial networks. arXiv:1807.04015 [cs, stat] (2020)
15. Huang, H., Li, Z., He, R., Sun, Z., Tan, T.: IntroVAE: introspective variational autoencoders for photographic image synthesis. arXiv:1807.06358 [cs, stat]. (2018)
16. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
17. van der Maaten, L., Hinton, G.: Viualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
18. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise (1996)
19. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, pp. 82–90. Curran Associates Inc., Red Hook (2016)
20. Cheng, S., Bronstein, M., Zhou, Y., Kotsia, I., Pantic, M., Zafeiriou, S.: MeshGAN: non-linear 3D morphable models of faces. arXiv:1903.10384 [cs] (2019)
21. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. arXiv:1707.02392 [cs] (2017)
22. Facebook: Pytorch3D. https://github.com/facebookresearch/pytorch3d