Frontiers of Architectural Research (2021) **10**, 304–316

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.keaipublishing.com/foar

Research Article

Training deep convolution network with synthetic data for architectural morphological prototype classification

Chenyi Cai, Biao Li*

School of Architecture, Southeast University, Nanjing 210096, China

Received 13 July 2020; received in revised form 4 December 2020; accepted 5 December 2020

KEYWORDS
Deep convolution network;
Architectural morphology;
Prototype classification;
Feature extraction;
Generative design

Abstract The use of architectural morphological analysis and generative design is an important strategy to interpret current designs and to propose novel ones. Conventional morphological features are defined based on qualitative descriptions or manually selected indicators, which include subjective bias, thus limiting generalizability. The lack of public architectural morphological datasets also leads to setbacks in data-driven morphological analysis. This study proposed a new method for generating topology-based synthetic data via a rule-based system and for encoding morphological information to promote morphological classification via deep learning. A deep convolution network, LeNet, which was modified in the output layer, was trained with synthetic data, including five spatial prototypes (central, linear, radial, cluster, and grid). The performance of the proposed method was validated on 40 practical architectural layouts. Compared to the ground truth, the proposed method provided an encouraging accuracy of 97.5% (39/40). Interestingly, the most possible mistakes of the LeNet were also understandable according to the architect's intuitive perception. The proposed method considered the statistical and overall characteristics of the training samples. This work demonstrated the feasibility and effectiveness of the deep learning network trained with synthetic architectural patterns for morphological classification in practical architectural layouts. The findings of this work could serve as a basis for further morpho-topology studies and other social, building energy, and building structure studies related to spatial morphology.

* Corresponding author.
E-mail address: jz_generator@gmail.com (B. Li).
Peer review under responsibility of Southeast University.

https://doi.org/10.1016/j.foar.2020.12.002

1. Introduction

According to Li and Han (2011), architectural design requires an integrated balance of complex adaptive systems (CAS). One solution is generative design, which focuses on the translation and simulation of design concepts using computational models, including decision making, linkage construction, and design optimization. The computational model is established based on the large amounts of data and the extraction of rational rules to achieve novel design proposals (Li, 2012; Soddu, 1998).

From the rationalist view, the morphological approach proposes the idea that morphology has the potential to be the driving force behind the urban design process (Rossi et al., 1982). Given that architectural morphology determines the spatial footprint pattern and influences the urban fabric (Levy, 1999), many studies have linked morphological types to the aspects of the building energy, building interior, social effects, urban evolution, etc. For example, a study on urban neighborhoods (Ariga, 2005) used the morphological classification and clustering of footprint patterns over time in order to extract evolutionary patterns that result in sustainable urban neighborhoods. In addition to conceptual and qualitative studies on morphological classification, recent morphological similarity studies have been carried out quantitatively by considering shape, size, spatial proportion, and other geometrical measures.

In quantitative evaluations, researchers utilized various morphology-to-data transmission methods by selecting and adjusting indicators to represent morphology. For computational analysis and design, quantifying architectural morphology and function is important in building effective computational models. In the research on urban renewal in Roma (Tang et al., 2019), scalar indicators (e.g., block ID, plot area) and geometric indicators (e.g., edge length, shape) were used to represent block morphology for searching morphologically similar blocks. However, some missing factors remained. This is especially true for geometric information that is difficult to represent numerically, such as the composition of buildings, thus leading to subjective bias and limited generalizability.

Deep learning[1] algorithms promote new methodologies in morphological analysis and generative design. It statistically provides automatic feature extraction and learning strategies for morphological analysis and design (Li et al., 2019). Morphology-to-data transmission methods, such as image data-based (RGB), numerical labeling, and semantic segmentation (Chaillou, 2019), are dedicated to feeding samples into neural networks with continuous and informative features. In a study of typo-morphology in Lisbon (Gil et al., 2012), the k-means clustering algorithm was used to classify block and street types based on prepared plans. However, the clustering algorithm was limited in terms of data size, because the learning algorithms required a large number of training samples to ensure learning accuracy. Furthermore, the data preparation phase of the proposed process was time-consuming, because there were few shared datasets related to architectural morphology.

To develop a morphology-to-data transmission approach for morphological classification, two sets of obstacles were observed: (1) the availability of architectural morphology dataset for inputting deep learning algorithms and (2) the quantification of architectural morphological features. In the computer science field, feeding deep learning algorithms with synthetic data for the recognition of actual conditions could achieve considerable results (Srisuchinnawong et al., 2018), with the synthetic data generated based on morphological similarity and diversity. However, the application of synthetic training data for testing practical data in the architectural morphological field has yet to be fully explored.

According to the previous discussion, generating synthetic data based on topo-morphological similarity may offer an opportunity to provide effective training samples for deep learning algorithms. A fully automated feature extraction method may overcome the drawbacks of manually selecting indicators. The pictorial processing of the morphology could be taken as the data source for images, so a possible method is feature mapping of the images. An image is a 3D matrix (RGB) that contains spatial information. For example, spatially adjacent pixels have similar values, whereas pixels that are farther apart have little correlation. Therefore, the spatial information of images hides essential features that are worth extracting. Feature mapping is a transmission from the original data to feature vectors that contain the information of overall characteristics.

In a convolution neural network (CNN), the feature mapping of the images extracts features by convolution kernels. Since its introduction in 1988, LeNet (Lecun et al., 1998), the progenitor of CNN, has undergone continuous improvements and is considered one of the classic models of CNN. Neural networks are implemented for solving problems of pattern recognition of two-dimensional (2D) images (Krizhevsky et al., 2012) and multiple manifolds, such as graphs and two-dimensional (3D) models (Tom et al., 2018). Researchers have applied the manifolds to various morphological domains, such as style recognition (Yoshimura et al., 2019) and architectural element classification (Yetis and Yetkin, 2018).

The current paper aims to develop a novel method for architectural morphology-related studies. The proposed method is a combination of the rule-based systems and data-driven approaches, which generate and encode synthetic morphological training samples for neural networks performing in actual conditions. In this research, a deep CNN was trained with concise feature vectors of synthetic morphological patterns. The synthetic data were generated by extracting the volume organization rules of architectural spatial prototypes, taking five types as examples. Thus, the typological similarity was integrated with the actual

[1] Deep learning uses neural networks as its main model and is widely applied in data mining, computer vision, text processing, autonomous driving, speech recognition, and other fields.

conditions. This work further completed the classification and prototype identification of practical architectural spatial organization. This paper makes the following contributions to the literature:

- This is an interdisciplinary study that integrates architectural design with computer science and applies the latest deep learning techniques to the study of architectural morphological analysis.
- The proposed method effectively performs the automatic generation of the architectural morphological patterns as training samples for neural networks, thereby reducing the difficulties in collecting data, cleaning data, and labeling data in real applications.
- This study provides a feasible method of comprehensively and automatically encoding the overall morphological characteristics with feature mapping based on pixels, excluding artificial work on morphological indicators selection, thus avoiding subjective bias.
- The previous methods working on the same tasks are either the qualitative description or informative independent indicators. In comparison, the proposed approach considers both aspects simultaneously.
- The signified neural network output can be changed arbitrarily to meet a certain demand. Therefore, the new output can be conveniently adapted for the further application of building energy efficiency evaluation, building structure, and generative architectural design projects related to spatial morphology.

2. Concept model of generative design

Two generative design approaches are taken in the computational model construction: rule-based system and data-driven approaches. With the rule-based system, the architect builds a physical or mathematical simulation model and solves specific architectural problems under well-defined rules (Li, 2012). It is an analytical problem-solving method that can help the architect to efficiently complete the linkage construction and optimization steps. In contrast, the data-driven approach creates statistical models and mines through vast amounts of data. It solves the problems of prediction, recommendation, and feature extraction. Furthermore, it helps to summarize and identify non-logical rules, such as personal preference, subjective tendency, style definition, and hand-drawn images, which support architects during the design decision step.

2.1. Rule-based system in generative design

Digital technology allows architectural elements to be freely described and interconnected without losing the systematic mapping (Hovestadt, 2010). With the rule-based system approach, designers need to manually define and quantify features, such as the architectural concept, the building elements, the physical environment of the building, and the rules of the elemental association. Furthermore, such algorithms such as evolutionary algorithms, multi-agent systems (Caiet al., 2019), and integer programming (Hua et al., 2019), are required to match design tasks. Then, computational models are constructed to simulate the self-organization of architectural elements and to achieve design generation in dynamic processes (Tang et al., 2019).

2.2. Data-driven methods in generative design

Data-driven methods use a statistical approach to describe and solve problems consisting of a flexible number of elements, connections, and variables (Hovestadt, 2010). Digital techniques have been widely used in architecture, generating large amounts of precise data. For example, 3D scanning, depth detection, and synchronous location and mapping (SLAM) technologies have been increasingly used along with online maps. These techniques provide architects with large amounts of data quickly and efficiently. They also help establish a database of architectural cases with feature information and retrieve cases with similar features (Tang et al., 2019). With a data-driven approach, designers need to extract features from abundant of training data by designing a network that can support design decision-making through data analysis methods.

2.3. Neural network in generative design

Similar to conventional feature engineering, rule-based systems are based on pre-defined rules and translate design rules into programming principles. It means that the information synthesis is limited by the architect's stock knowledge and experiences, which may result in subjective bias and inadequate linkage construction of design variables. In comparison, neural network algorithms can efficiently parse out general features from large amounts of data and apply those to new data in testing. Neural networks are used for feature extraction, prediction, and clustering (Nielsen, 2015). However, limited by the available data, the application of the neural network in the architecture field is restricted. Nevertheless, it is an effective way to communicate and simulate the architectural design concept by integrating the two approaches. It supports database construction and feature extraction and then guides the computational model to accurately correlate design elements and variables, ultimately completing design evaluation and in-depth optimization.

Two key points in the research of the application of deep learning in architecture are (1) the accessibility of training samples and (2) the information encoding of sample features. Previously published studies construct a training dataset by crawling online data or manual semantic segmentation, which are time-consuming and have a limited sample size (Chaillou, 2019; Huang and Zheng, 2018; Liu et al., 2019). This process can also generate subjective noise. A more efficient and accurate approach is the use of an automatic data preprocess pipeline, which can feed the architectural morphology to the algorithms.

Fig. 1 represents the general workflow of our work. First, five architectural spatial organization prototypes were extracted. The architectural morphological patterns were generated based on the prototype abstraction. Second, the features of the generated samples were mapped into feature vectors for the neural network input dataset. Third,

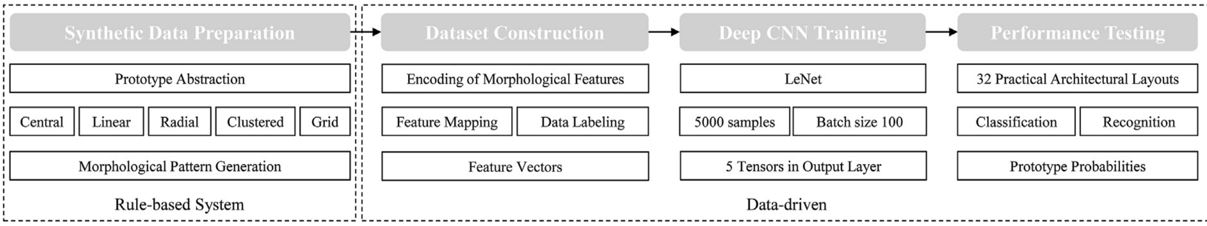


Fig. 1 The workflow of architectural morphological prototype classification based on neural network.

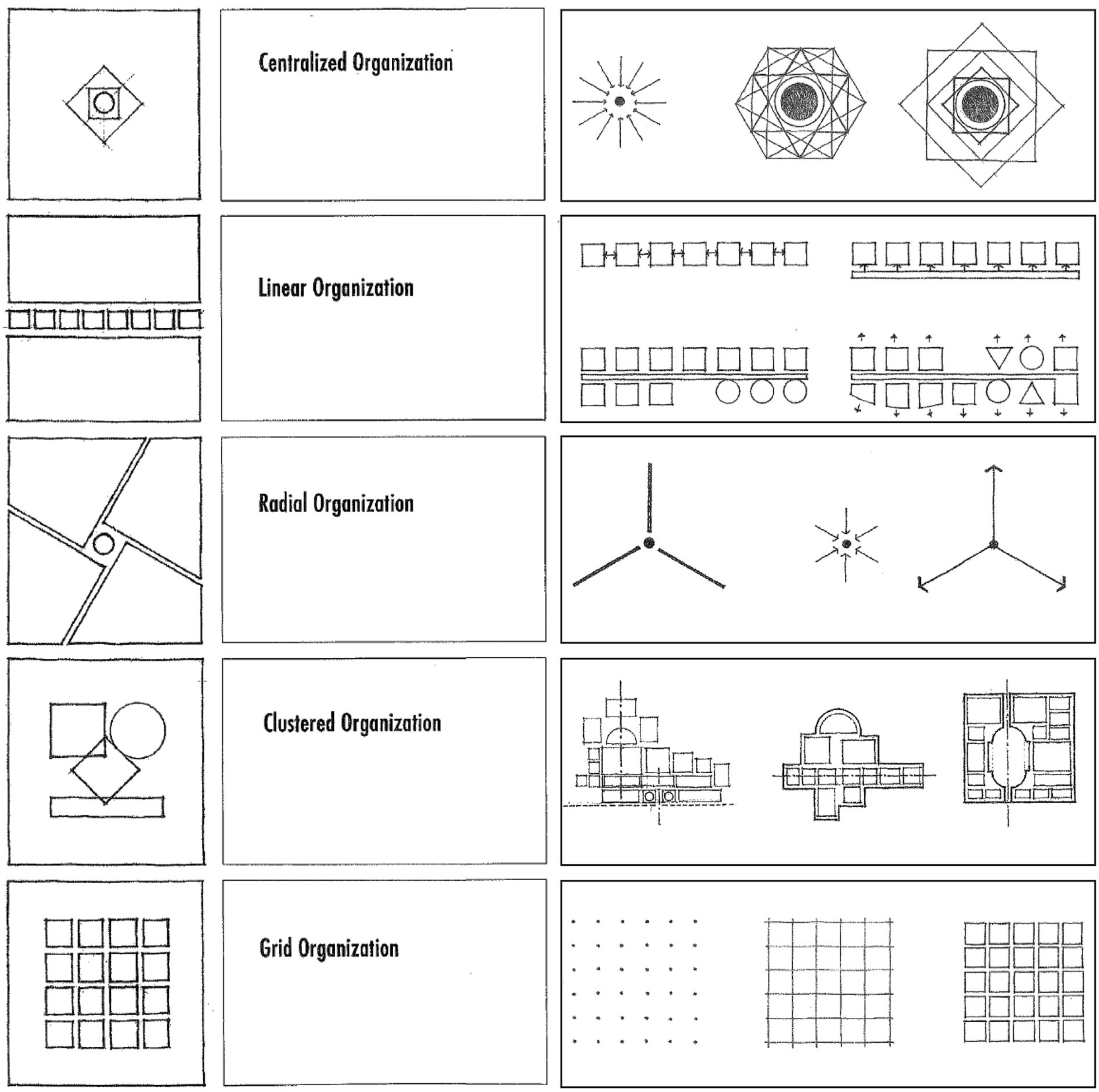


Fig. 2 Five spatial prototypes of architectural morphological organizations.

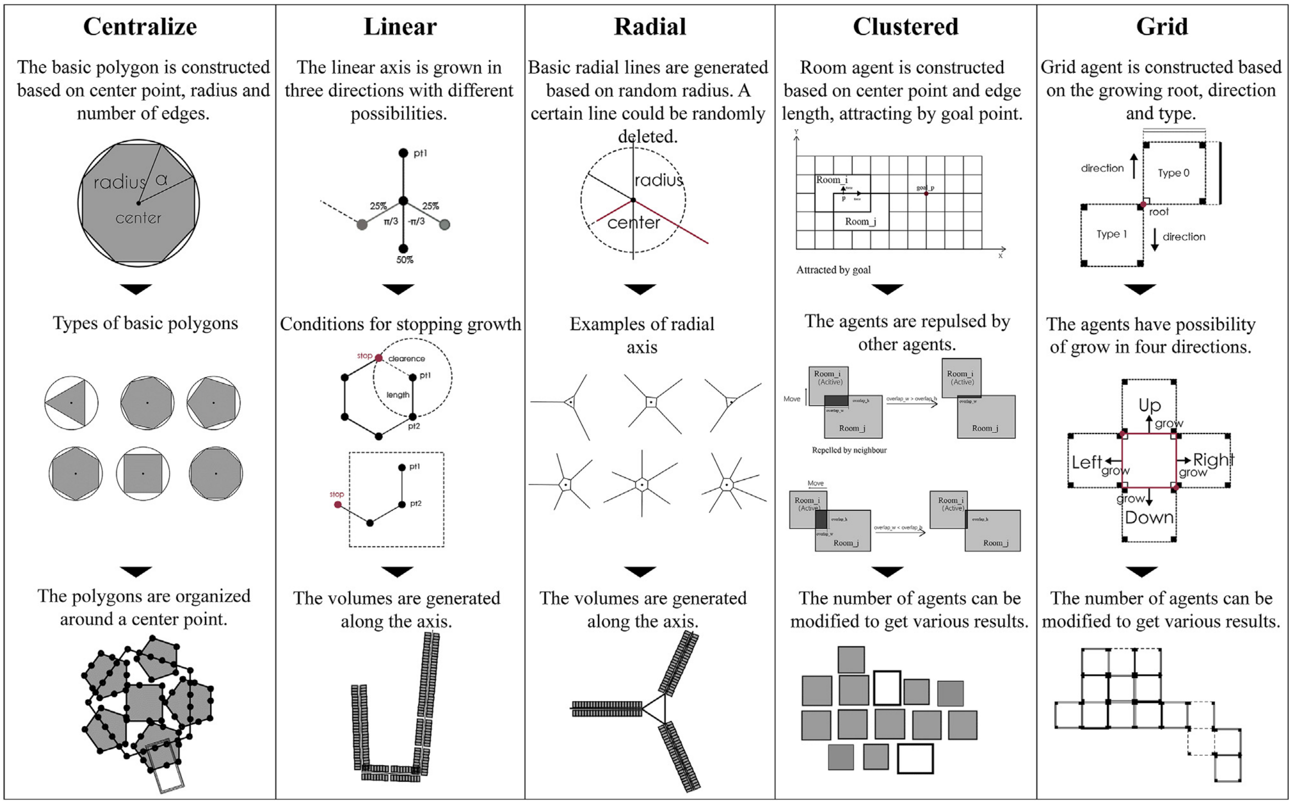


Fig. 3 Generation rules and process of the five types of morphological pattern samples.

the output layer modified LeNet was trained compared with a simple fully-connected neural network. Finally, 40 practical architectural layouts were taken to test the performance of the trained neural networks for morphological prototype classification.

3. Architectural morphological pattern generation

In the work entitled "Architecture: Form, Space, and Order," Cheng (2005) elaborated on the basic principles and syntax of architectural design as a classical theory in which the architectural space is interconnected and combined into a coherent prototype, including centralized, linear, radial, clustered, and grid (Fig. 2). It reflects the architects' general judgment on prototype classification of architectural layouts. We used the above-mentioned five spatial prototypes in this work.

Training neural networks require adequate input data in continuous features. We used algorithms, such as the multi-agent system (Cai et al., 2019) and L-system (Chan and Chiu, 2000), which are bottom-up methods to obtain a community by simulating the action and the relation of components. A technique called data augmentation is used for enlarging the dataset by artificially adding the variation of training samples to obtain a robust network (Goodfellow, Bengio and Courville,2016). The conventional data augmentation operations, such as rotation, mirroring, etc., are used for making minor changes to existing datasets in order to acquire more samples to reduce overfitting. To augment our training samples, we used the generative method to obtain various morphologies based on the topological similarity. We added random function to parameters, such as direction, radius, edge type, angle, etc. Hence, the samples are topologically similar but different in detail in terms of morphology, thereby ensuring the training quality. Fig. 3[2] shows the generation logic of the five types of morphological patterns.

1. The centralized pattern consisted of multiple basic polygons arranged around the center point. The basic polygon was constructed from three elements: center, radius, and the number of sides. The centralized morphological patterns were obtained after the triple recursion of the basic polygons.
2. The linear patterns were generated by designing the linear axis. The growth of the axis through the Lindenmayer System (L-system) was simulated, starting from the growth point, with three possible directions of growth: forward, left, and right with certain angles. The growth stopped if the growth point was out of the panel or too close to the original growth point. Then, the rooms were generated along the axis.

[2] The L-system is a type of formal grammar. It has been used to describe the behavior of plant cells and applied to model the morphology of various organisms (Source: Wikipedia).

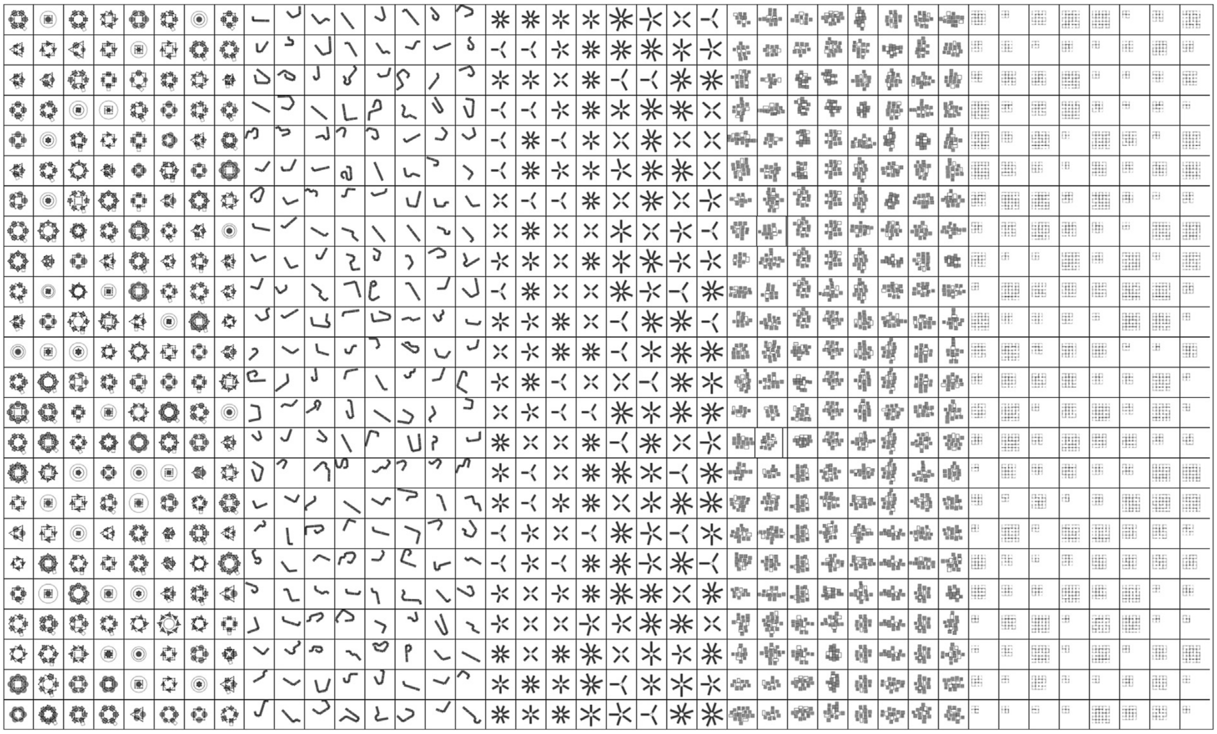


Fig. 4 Examples of generated architectural morphological patterns.

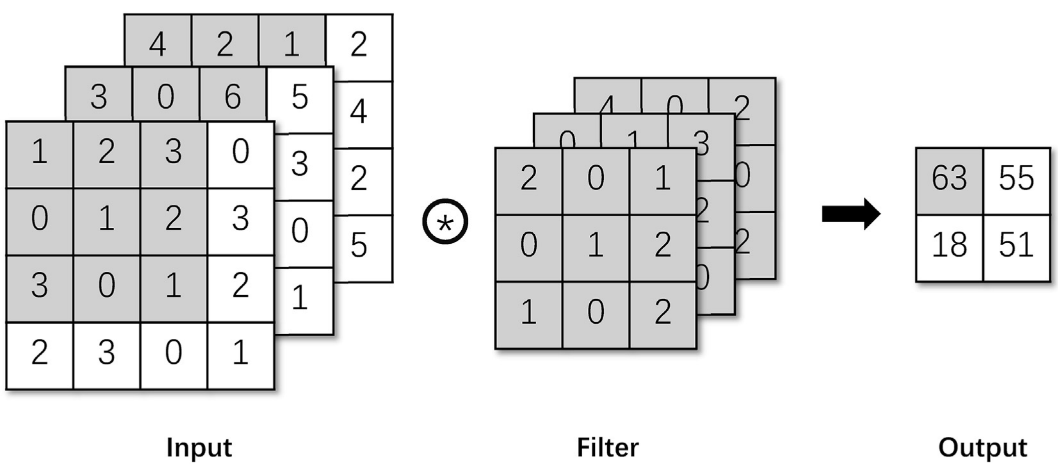


Fig. 5 Fundamental filters in convolutional operations.

3. The logic for generating radial pattern axis was similar to the centralized pattern except that the vertices were randomly removed to obtain irregular radials.
4. In the clustered pattern, the rooms and courtyards were simulated by rectangular multi-intelligent agents, with both attraction and repulsion forces between each other, thus obtaining a balanced situation.
5. The grid pattern was generated through grid agents (constructed from the growth point and direction) with three types of edges: thick, bilinear, and dotted. This was done to ensure that each agent had eight morphological possibilities. Here, the growth of the grids was based on the growth of the agents.

Fig. 4 shows examples of the five types of architectural morphological patterns with the following details.

4. Neural network model training and prototype classification

The application of the neural network model can be divided into two steps. The first step involved training by simulating the signal propagation between neurons, thus extracting the general features of the training sample. The second step was testing in which the trained network was applied to the new samples. Due to the computational mechanism of neural networks, we do not need to explicitly define the

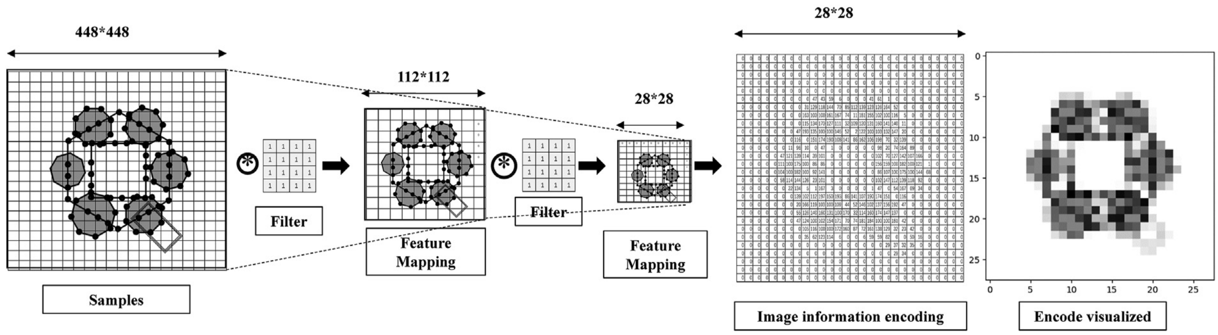


Fig. 6 Overview of the morphology-to-data process for generating samples.

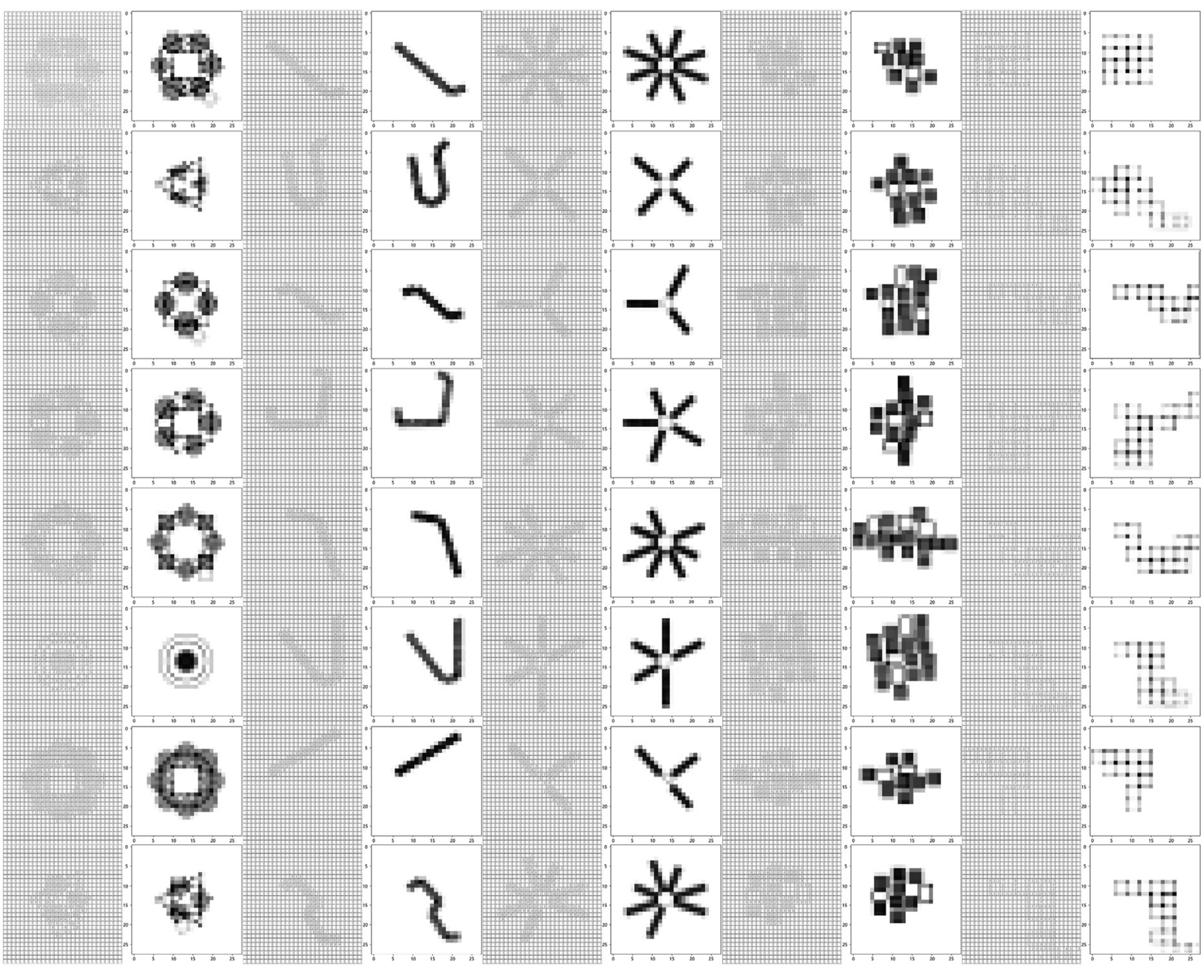


Fig. 7 Partial visualization of the architectural morphological datasets.

function or process for target morphology. We just need to define the input data and output targets. In other words, the classification of neural networks can solve such a problem when the architectural morphology cannot be exhausted under a particular classification due to their diversity and ambiguity of definition.

4.1. Morphological pattern information encoding

Based on the idea of feature mapping, the image was processed with a "filter operation" for feature mapping in the convolution process. The process of feature mapping can be described as taking a certain size kernel (3 × 3 as an example in Fig. 5), sliding it over the image pixels, and performing operations, such as multiplication and accumulation, to obtain a feature matrix.

In this experiment, the size of the generated morphology sample was 448 $\times$ 448 pixels. To make the neural network operation more efficient and to prevent the loss of image features, the image information was encoded in this experiment by sweeping the image in two layers with filters of size 4 $\times$ 4 and stride 4. As the samples were in black and white, we encoded them based on a single channel, taking the brightness intensity average and finally encoding the architectural morphology information in the sample as 28 $\times$ 28 feature mapping data (Fig. 6). We took 1000 generated samples for each type of spatial organization pattern in the experiment. The spatial prototypes were labeled by 0—4 according to the morphological pattern. The

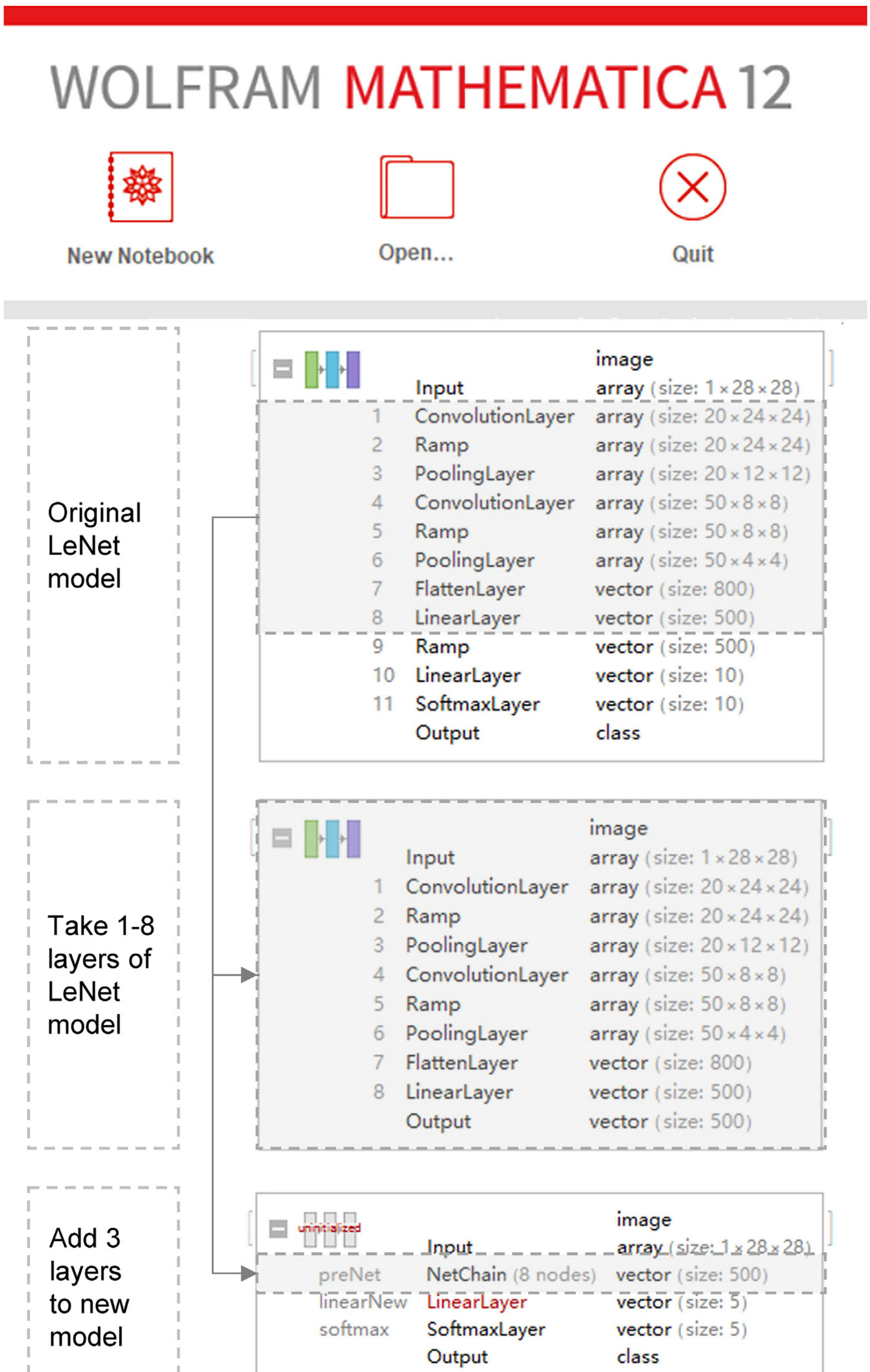


Fig. 8 Structure of the original LeNet and its modification.

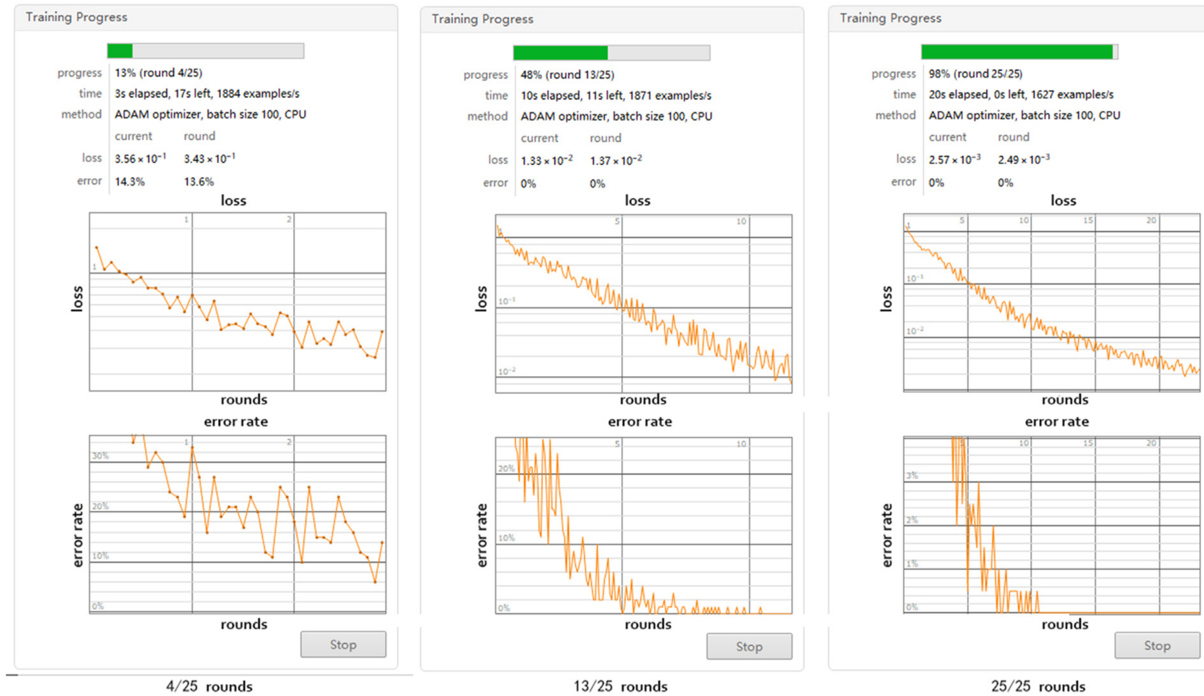


Fig. 9 Gradual reduction of learning errors and recognition failure during training.

ground truth number was the goal for the neural network output to be trained as close as possible. Therefore, the dataset featuring the architectural morphology was constructed from 5000 (28×28) encoded feature data with labels. Fig. 7 shows a visualization of the architectural morphology encoded information.

4.2. Training of LeNet

The basic function for weights and deviations between neural layers is expressed as formula (1), where $A^{(n)}$ indicates neurons in the *nth* neural layer, $W^{(n)}$ indicates the weight propagated between each neuron and the previous neural layer, $b^{(n)}$ indicates the bias propagated between each neuron and the previous neural layer, and X indicates the data in the previous layer. The backpropagation of errors in the neural network is expressed as formula (2), where E indicates the error between estimation and target, and σ indicates the learning rate.

$$A^{(n)} = X \cdot W^{(n)} + b^{(n)} \tag{1}$$

$$\Delta W = \sigma \frac{\partial E}{\partial W} \tag{2}$$

The neural network training continuously optimizes weights (W) and bias (b) using gradient descent and error inverse propagation method, so that the distance (E) between the output and the target becomes progressively smaller. In this way, the output gradually gets closer to the optimization goal. The neural network testing uses the W-b matrix to obtain the training step and then operate with the test data. The output of the five neurons in the output layer represents the probability of the corresponding prototype.

We used a deep convolution neural network, LeNet, which was designed for training with MNIST[3]dataset. The original LeNet structure has a total of 11 layers, including two convolutional layers. The input layer size was 28×28 with one channel, as the samples were black and white images. The output layer size was 10 with one channel. As the output layer of this experiment required 5 neurons, instead of 10, therefore, we took 1–8 layers of the LeNet. Then, we added three new layers: a linear layer, a softmax layer, and an output layer, with five neurons to constitute the neural network adopted for this experiment (Fig. 8). This was implemented on Mathematica (Wolfram Research, Champaign), and we used the ADAM optimizer. The batch size was 100. In less than 20 s, the neural network was trained after 25 rounds. Fig. 9 illustrates the gradual decrease in recognition failure as the number of training iterations increases.

4.3. Testing of the trained neural networks for morphological prototype classification

A total of 40 practical building layouts were selected to test the performance of the neural network. These samples were chosen from the reference, and the author used these examples to illustrate the spatial organization prototypes. Therefore, each sample was labeled with the ground truth according to the book. Due to the diversity of architectural morphology, some samples could be considered to belong to more than one prototype depending on the architect's

[3] It contains thousands of black and white images of handwritten 0–9 numbers. The trained neural networks can recognize new handwritten numbers. The accuracy of recognition is often used to measure the performance of specific neural networks.

different interpretation. As a comparison study, we tested the 40 samples through a simple, trained, fully-connected neural network with three layers (an input layer, a hidden layer, and an output layer) and the trained LeNet.

The neural network predicted the respective probabilities of five output neurons and ultimately took the highest probability as the output result. The result's accuracy reflects the quality of the synthetic training samples and the

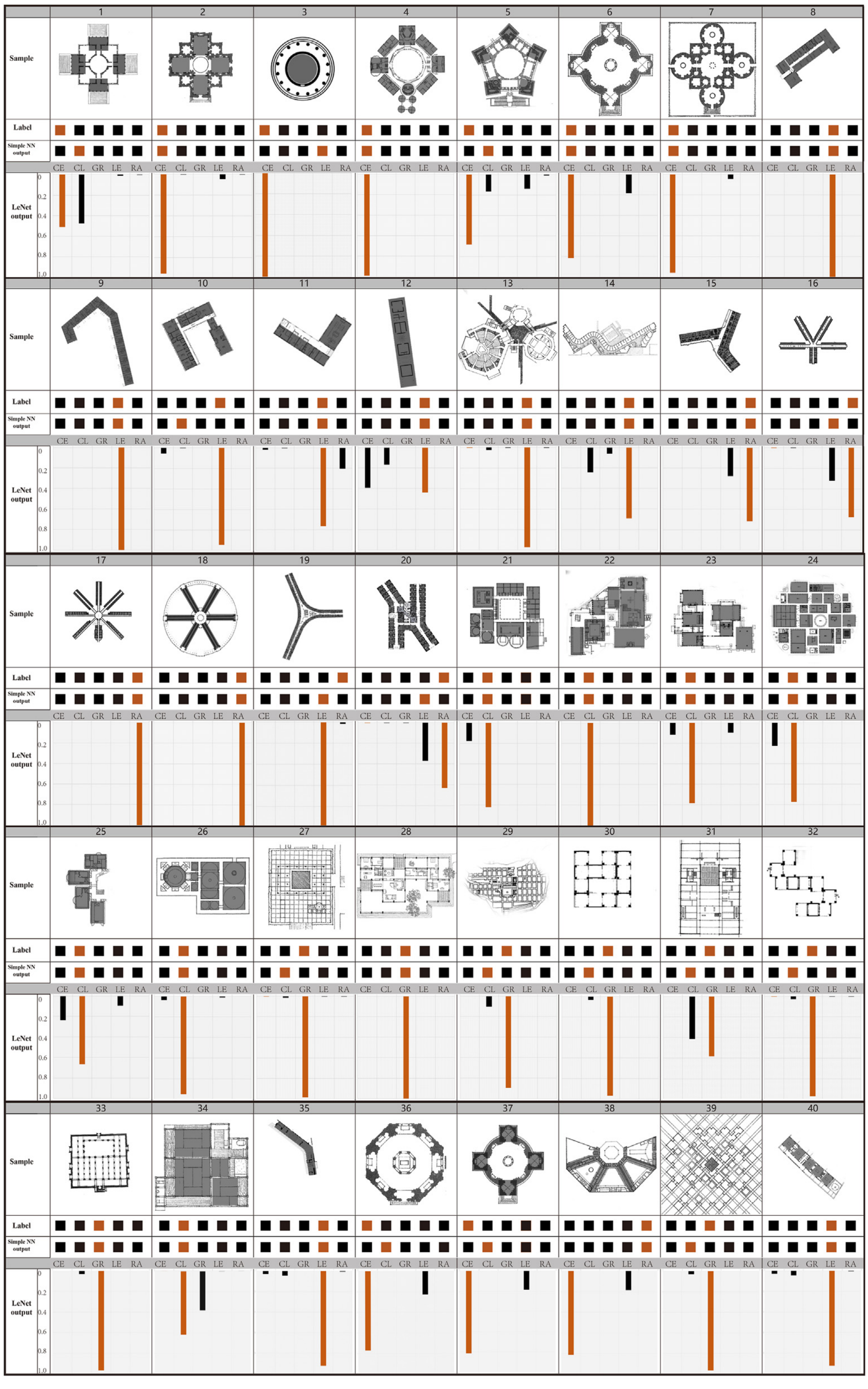


Fig. 10 The outputs of the simple neural network and the LeNet on the architectural morphological prototype classification.

performance of the neural networks. The size of the neural network input and output can be modified. The results can be used as conditions for further programs, such as concept strategy and case retrieval.

5. Results and discussion

The fully-connected neural network correctly generated outputs in 25 out of 40 samples (Fig. 10). The deep convolutional neural network LeNet identified 31 out of 40 samples correctly, compared to the ground truth labels. The failed case involved identifying a radial prototype, case 19, as a linear prototype (Fig. 10). The deep convolutional neural network performed better than the simple fully-connected neural network, as it had a more developed structure.

Interestingly, we can intuitively see that case 19 can be classified as a linear prototype as it shows linear morphology. Furthermore, some other samples could be classified into more than one prototype, which is also reflected in the LeNet output. For example, some architects would take case 1 as a clustered prototype, although it is labeled as a centralized prototype; cases 15, 16, and 20 could be taken as a linear prototype while being labeled as a radial prototype; and cases 24 and 25 have the possibility of being the centralized prototype while being labeled as a clustered prototype.

Fig. 11 shows the most possible mistake of the LeNet compared with the architect's definition variety, taking the eight cases with high secondary probability output as examples. For example, in case 1, the probability of a centralized prototype is 0.51, whereas that of a clustered prototype is 0.47; in case 20, the probability of a radial prototype is 0.62, whereas that of the linear prototype is 0.37. This is a surprising outcome wherein the neural network's output in terms of architectural morphological prototype shows high accuracy and similarity mistakes with architects. Architects take more than one possibility when

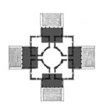
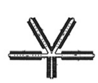
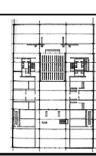
Case	Image	Architect's divergence classification	LeNet output
1		Centralized Clustered	Centralized : 51.09% Clustered : 47.78%
12		Linear	Linear : 43.77% Centralized : 39.38% Clustered : 16.76%
15		Radial Linear	Radial : 72.08% Linear : 27.91%
16		Radial Linear	Radial : 67.59% Linear : 32.19%
20		Radial Linear	Radial : 62.92% Linear : 37.04%
24		Clustered Centralized	Clustered : 77.26% Centralized : 22.72%
25		Clustered Centralized	Clustered : 66.84% Centralized : 23.58%
32		Grid	Grid : 58.24% Clustered : 41.32%

Fig. 11 Most possible mistakes of the LeNet compared with the architect's possible classification.

classifying certain samples, and the LeNet reflects a similar decision process.

As the training samples were generated based on a specific prototype, they integrated both topological similarity and morphological diversity simultaneously. Compared with the conventional data augmentation methods, the proposed method provides a more flexible and diverse enhanced data source. We transmitted the 448×448-pixel image to 28×28 feature mapping data, thus boosting the efficiency of the neural network. In the calculation process of the LeNet, the input data were calculated through kernels of the $n \times n$ matrix containing weight values. This is a robust technique for image processing. In other words, the neural network is sensitive to the distribution of the pixels in the image. For these reasons, the neural networks performed well in classification based on the generated samples.

Compared with other quantitative classification approaches, encoding morphology into feature data based on feature mapping is a fully automated method that includes several aspects of the morphological criteria without the need to balance the weights of indicators. Training with synthetic data saves effort in collecting, selecting, and labeling data. The training target could also flexibly perform a certain task; for example, the training labels could be modified to represent the building energy performance, structural type, etc.

The limitations of the proposed method are related to the highly automated process, which increases the difficulties of emphasizing the influence of a certain aspect by adjusting the weights. First, the generated samples could be more detailed, and more types are needed to achieve a trained network and architectural morphology with higher accuracy and better diversity, respectively. Second, given that the samples are image-based and only contains 2D information, 3D information like building height and building form, may be lost. This drawback could be overcome by adding one more dimension, such as the grayscale dimension in the picture to represent the building height or the use of voxels instead of pixels. Third, for other tasks that need different training samples, the generative system should be redeveloped in order to obtain task-oriented synthetic data.

6. Conclusion

Computers can rapidly perform 4- or even 10-digit operations, because they are extremely powerful in terms of following basic instructions. However, the human brain can easily distinguish information contained in pictures, such as faces, cats, and furniture. In comparison, such a task is quite difficult for a computer (Rashid, 2016). In relation to this, one of the goals of neural networks is to solve non-explicit instruction problems, such as morphological analysis.

Morphological similarity analysis and classification represent a useful analysis framework in many studies, such as those in the typo-morphological, historical evolution, predesign contextualization, and building energy performance fields. The quantitative descriptions of architectural morphology provide a baseline for in-depth building interpretation and have received attention in morphological studies in recent years. Conventional methods based on an intuitive perspective of forms often consist of statistical calculations, weight decisions, and indicator selections, which are used for the integrated description of a building. It leads to subjective bias and may greatly be influenced by the researcher's knowledge.

Morphological analysis is promoted by the recent success of deep learning methods in which the intrinsic features can be extracted/learned automatically from a large amount of data. The morphological features are quantified by high-dimensional feature data through a deep CNN, which contains the overall information of the morphological characteristics, rather than one-to-one correlations of the features.

We applied a rule-based system to generate morphological patterns, including five spatial prototypes, to construct a synthetic training dataset. This was done to supplement the insufficient architectural morphological datasets and to save efforts on preprocessing the training datasets. To complete the morphology-to-data transmission, we used the convolutional approach based on pixels to quantify the morphological features. A total of 5000 synthetic training samples served as inputs for training the networks. The performance of the proposed method was validated on 40 practical architectural layouts. Compared to the ground truth from the reference book, the modified LeNet provided an encouraging accuracy of 97% (39/40), whereas that of the simple, fully-connected neural network was 62.5% (25/40). Interestingly, the only mistake of the LeNet was similar to that of some experienced architects. Furthermore, the most possible mistakes of the LeNet were similar to the experienced designers' mistakes when intuitively observing the testing samples. The LeNet output shows similarity with the architects' definition variety on some samples. The results indicate that, by developing a proper training dataset, the neural network output can be highly accurate while still maintaining the diversity of the morphological definition. The proposed method, therefore, can serve as a basis for further architectural typo-morphology-related studies.

This work demonstrated the feasibility and power of using the deep learning network in architectural morphology. The finding of this work can help promote morphological design in the future and potentially facilitate a greater understanding of architects' designs. Our future work will focus on technological improvements and the following application scenarios:

- More dimensions, such as grayscale value, can be added to the data source to indicate volume heights and improve the model performance.
- More information could be added to the data source according to specific scenes. For example, the generated samples could include more details about green or vertical spaces.
- The training targets could be modified according to specific application scenarios, such as morphology-

related building performance evaluations and building structure types, by changing the label interpretation.

- Further, the output of the neural network could be used as one of the input parameters for generative designs based on the rule-based system. For example, the conceptual hand-drawing scripts could serve as inputs for the trained neural network to obtain similarity comparisons for case-based studies. With the combination of generative designs, the proposed method could facilitate the decision-making and promote further building development.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript submitted.

Acknowledgments

This research was funded by National Key R&D Program of China (2017YFC0702302 in 2017YFC0702300), National Natural Science Foundation of China (Project No. 51978139) and Fundamental Research Funds for the Central Universities and Postgraduate Research Innovation Program of Jiangsu Province (Project No. KYCX20_0110).

References

Ariga, T., 2005. Morphology, sustainable evolution of inner-urban neighborhoods in san francisco. J. Asian Architect. Build E ng. 4 (1), 143–150.

Cai, C., Tang, P., Li, B., 2019. Intelligent generation of architectural layout inheriting spatial features of Chinese garden based on prototype and multi-agent system - a case study on Lotus teahouse in yixing. Paper presented at the. In: Proceedings of the 24th CAADRIA Conference. Victoria University of Wellington, Wellington, New Zealand.

Chaillou, S., 2019. AI+ Architecture: towards a New Approach. Harvard University.

Chan, C.W., Chiu, M.L., 2000. A simulation study of urban growth patterns with fractal geometry. Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia 55–64.

Cheng, D., 2005. Architecture: Form, Space and Order. Tianjin University Press, Tianjin.

Gil, J., Beirão, J.N., Montenegro, N., Duarte, J.P., 2012. On the discovery of urban typologies: data mining the many dimensions of urban form. Urban Morphol. 16 (1), 27–40.

Goodfellow Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge.

Hovestadt, L., 2010. Beyond the Grid: Architecture and Information Technology: Applications of a Digital Architectonic. Birkhauser, Basel.

Hua, H., Hovestadt, L., Tang, P., Li, B., 2019. Integer programming for urban design. Eur. J. Oper. Res. 274 (3), 1125–1137.

Huang, W., Zheng, H., 2018. Architectural drawings recognition and generation through machine learning. Paper presented at the. In: Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (Mexico City, Mexico).

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25 (2).

Lecun, Y., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Levy, A., 1999. Urban morphology and the problem of the modern urban fabric: some questions for research. Urban Morphol. 3, 79–85.

Li, B., 2012. GenerativeDesign: Exploration of Computer-Aided Architectural Design Methods Based on Complex Adaptive Systems. Southeastern University Press.

Li, B., Han, D., 2011. Technical comprehension of architectural generative design and its prospects. Architectural Journal (06), 96–100.

Li, B., Zhang, J., Ludger, H., Guo, Z., 2019. Analyzing the black box of design with algorithm model. Architect 94–99, 01.

Liu, Y., Lin, W., Deng, Q., Liang, L., 2019. Exploring the building form generation by neural networks –Taking the free university of berlin as an example. Paper presented at the Proccedings of 2019. In: National Conference on Architecture's Digital Technologies in Education Research.

Nielsen, M.A., 2015. Neural Networks and Deep Learning, vol. 2018. Determination press, San Francisco, CA, USA.

Rashid, T., 2016. Make Your Own Neural Network. CreateSpace Independent Publishing Platform.

Rossi, A., Eisenman, P., Ghirardo, D.Y., Ockman, J., 1982. The Architecture of the City. The MIT Press, Cambridge.

Soddu, C., 1998. Argenia, a natural generative design. Generative art: Proceedings of GA 98, 7–41.

Srisuchinnawong, A., Sakulkueakulsuk, B., Chaisangmongkon, W., 2018. Word recognition in captured images by CNN trained with synthetic images. Paper presented at the. In: 2018 22nd International Computer Science and Engineering Conference (ICSEC).

Tang, P., Li, H., Wang, X., 2019. Generative design on conservation and inheritance of traditional architecture and settlement based on machine learning: a case study on the urban renewal design of roma termini railway station. Architect (1), 100–105.

Tom, K., Paul, G., Anthony, S., Wonka, P., Mitra, N., 2018. FrankenGAN: Guided Detail Synthesis for Building Mass-Models Using Style-Synchronized GANs. arXiv preprint arXiv:1806.07179.

Yetiş, G., Yetkin, O., 2018. A novel approach for classification of structural elements in a 3D model by supervised learning. Paper presented at the. In: Proceedings of the 36th eCAADe Conference (Poland).

Yoshimura, Y., Cai, B.Y., Wang, Z., Ratti, C., 2019. Deep Learning Architect: Classification for Architectural Design through the Eye of Artificial Intelligence (Paper presented at the annual conference on computers).