

A graph convolutional neural network for classification of building patterns using spatial vector data

Xiongfeng Yan^a, Tinghua Ai^{a,*}, Min Yang^a, Hongmei Yin^b

^a School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China

^b Chinese Academy of Surveying and Mapping, Beijing 100830, China



ARTICLE INFO

Keywords:

Building pattern classification
Graph convolutional neural network
Machine learning
Spatial vector data
Graph Fourier transform
Deep learning

ABSTRACT

Machine learning methods, specifically, convolutional neural networks (CNNs), have emerged as an integral part of scientific research in many disciplines. However, these powerful methods often fail to perform pattern analysis and knowledge mining with spatial vector data because in most cases, such data are not underlying grid-like or array structures but can only be modeled as graph structures. The present study introduces a novel graph convolution by converting it from the vertex domain into a point-wise product in the Fourier domain using the graph Fourier transform and convolution theorem. In addition, the graph convolutional neural network (GCNN) architecture is proposed to analyze graph-structured spatial vector data. The focus of this study is the classical task of building pattern classification, which remains limited by the use of design rules and manually extracted features for specific patterns. The spatial vector data representing grouped buildings are modeled as graphs, and indices for the characteristics of individual buildings are investigated to collect the input variables. The pattern features of these graphs are directly extracted by training labeled data. Experiments confirmed that the GCNN produces satisfactory results in terms of identifying regular and irregular patterns, and thus achieves a significant improvement over existing methods. In summary, the GCNN has considerable potential for the analysis of graph-structured spatial vector data as well as scope for further improvement.

1. Introduction

With the advancements in computing power and the advent of the big-data era, machine learning methods have emerged as an integral part of scientific research in many disciplines. As a deep supervised learning architecture, convolutional neural networks (CNNs) have shown excellent performance in various fields, such as computer vision (Krizhevsky et al., 2012), speech recognition (Abdel-Hamid et al., 2014), natural language processing (Kim, 2014), and gaming (Silver et al., 2017). The excellent performance of CNNs is attributed to the local correlation preservation, which is implemented with a small kernel size and techniques such as weight sharing and pooling (LeCun et al., 1998). The local correlation of spatial data is also significant; several well-known theories in GIS, including spatial association (Anselin, 1995) and the first law of geography (Tobler, 1970), are closely related to it. Recently, CNNs and their variants have achieved state-of-the-art performance in a variety of tasks based on spatial raster data, such as remote sensing imagery (Anwer et al., 2018; Ding et al., 2018; Gong et al., 2017; Huang et al., 2018; Paoletti et al., 2018), and unordered or rasterized point cloud (Kumar et al., 2019; Wen et al.,

2019; Yousefhussien et al., 2018). However, such powerful learning methods are rarely applied directly to unstructured spatial vector data, as the convolution requires regularity of input data (Yan and Ai, 2018).

In this study, we introduce a novel convolution operation based on a graph structure and combine it with a neural network to construct a new learning model, namely graph convolutional neural network (GCNN), to analyze unstructured spatial vector data. Although a graph is an ideal tool for modeling spatial vector data, it is difficult to define a convolution directly in the vertex domain because the neighborhood structures of the vertices may not be stationary. On the basis of the graph Fourier transform and convolution theorem, the operation can be converted into a point-wise product in the Fourier domain, which is similar to a transformation from a spatial domain convolution into a frequency domain convolution in image processing. In this process, the graph Fourier transform is a linear transformation with the bases of Laplacian eigenvectors (Hammond et al., 2011; Shuman et al., 2016). Furthermore, graph convolution based on polynomial approximation offers many advantages, such as fast computation, a small number of parameters, and spatially local connectivity, which enable the GCNN to learn discriminative features from input graph-structured data.

* Corresponding author.

E-mail addresses: xiongfeng.yan@whu.edu.cn (X. Yan), tinghuai@whu.edu.cn (T. Ai), yangmin2003@whu.edu.cn (M. Yang), mayyin@whu.edu.cn (H. Yin).

The present study explores the potential of the GCNN model to analyze graph-structured spatial vector data by classifying building perceptual patterns. Buildings are crucial components of urban spaces, and recognition and analysis of their patterns is important for the modeling and characterization of urban geographical development and landscape configuration (Gonzalez-Abraham et al., 2007; Vanderhaegen and Canters, 2017; Wu et al., 2018), as well as for the semantic classification of social functions and the forecasting of economic activities (Beck et al., 2018; Hecht et al., 2015; Steiniger et al., 2008). Furthermore, building patterns convey important cognitive information and structural knowledge that facilitate map navigation and spatial reasoning. Hence, detecting these patterns is also a critical precondition in digital cartography. Subsequently, these patterns can be preserved or enhanced as far as possible by means of map generalization and data enrichment techniques (Cetinkaya et al., 2015; Gong and Wu, 2018; He et al., 2018; Yan et al., 2017). However, current building pattern classification and recognition methods are time consuming and restrictive. They require rules to be designed and features to be extracted manually for specific patterns. The introduction of new learning methods could not only address the above-mentioned issues but also supplement attempts to solve classical problems in spatial analysis.

The remainder of this paper is organized as follows. Section 2 briefly reviews related studies on building pattern classification and the application of CNNs to graphs. Section 3 proposes a framework for classifying building patterns using the GCNN model. Section 4 presents the experimental datasets and results as well as the detailed analyses. Section 5 discusses some issues. Finally, Section 6 concludes the paper.

2. Related work

2.1. Building pattern classification

In general, building patterns refer to salient structures perceived individually or as a group in space, which can be recognized visually and named semantically (Du et al., 2016a; Pilehforooshha and Karimi, 2018). However, the description and typology vary considerably across different fields. In urban modeling, building patterns can be distinguished in terms of their socio-economic attributes and physical characteristics at the scale of a building polygon (e.g., detached and terraced houses) (Hecht et al., 2015; Henn et al., 2012; Meinel et al., 2009; Wurm et al., 2016) in addition, they can be divided into inner city areas, industrial and commercial areas, urban areas, suburban areas, and rural areas at the city block scale (Steiniger et al., 2008). In digital cartography, the patterns can be identified as certain regular geometric shapes, such as grids, circles, stars, and letters (Anders, 2006; Yang, 2008). From the perspective of visual cognition, the patterns can be classified into linear arrangements, regular grids, and irregularities (Du et al., 2016a; X. Zhang et al., 2013).

Although the taxonomy of building patterns has been studied extensively, there remains a lack of a clear and unanimous criterion, which hampers subsequent formalization and automatic identification. With regard to the formalization of building patterns, Lüscher et al. (2009) formalized high-order concepts, e.g., a pattern of terraced houses, using methods from ontological engineering. Du et al. (2016b) constructed a relational approach with three abstract levels by deriving 169 basic relations between buildings on the basis of their relative position and orientation. Yu et al. (2017) proposed a texton co-occurrence matrix method for representing and evaluating the features of building patterns with texture analysis.

In automatic building pattern identification, one of the most important existing approaches is the rule-based method, which mainly checks whether a similar structure or a specific arrangement appears among neighboring buildings. The process includes three main stages. The first stage is the representation of the spatial neighbor relationship, for which a graph is especially popular and appealing. The second stage

is the measurement of shapes. It aims to determine whether there are some geometric or semantic homogeneities among a building group, for which the most commonly used indices are size, orientation, and shape. The third stage is the definition of rules for a specific pattern, which is mainly based on angle difference, projection overlap, or other parameters. For example, Christophe and Ruas (2002) and Pilehforooshha and Karimi (2018) detected straight line building alignment for generalization purposes; X. Zhang et al. (2013) investigated two building patterns for alignment-of-center and alignment-of-side; and Du et al. (2016a) discussed some special arrangements, e.g., linear pattern, curvilinear pattern, and grid pattern.

Another strategy for classifying building patterns is based on machine learning algorithms. Some common classifiers have proven to be suitable for handing such multiple-scene problems, e.g., the support vector machine (L. Zhang et al., 2013) and random forest (He et al., 2018; Hecht et al., 2015; Jochem et al., 2018). As in the rule-based method, the first two steps of relationship representation and shape measurement are conducted; then, some features are extracted, e.g., the area difference or distribution, and these features are combined into a vector describing the group units; finally, a classifier can be employed to classify the groups into potential patterns. Such learning methods depend on the training of labeled examples rather than on manual rule definitions or formalizations for patterns. Statistical learning methods, e.g., the Markov random field (Huang et al., 2013), can also be used to model the relationship between buildings and to label or classify individual building types.

2.2. Application of convolutional neural networks to graphs

As mentioned above, CNNs often fail to analyze spatial vector data because of their regularity requirements for data structures. To generalize CNNs to general structures such as graphs, two potential solutions are proposed: the spatial approach and the spectral approach. The spatial approach directly conducts the convolution operation in the vertex domain but faces two challenges: (1) the definition of a receptive field, and (2) the ordering of vertices. Against this background, Sandryhaila and Mouraj (2013) and Such et al. (2017) attempted to redefine the convolution as a polynomial function of the graph adjacency matrix. Further, Niepert et al. (2016) mapped the unordered graph space to a linear vector space through neighborhood assembly and graph normalization, whereby the vertices are assigned according to their structural roles within the graph. In addition, some charting-based methods have also been developed, e.g., geodesic CNN (Masci et al., 2015) and mixture model network (Monti et al., 2017). Further details can be found in the review by Bronstein et al. (2017).

In contrast to the spatial approach, the spectral approach aims to realize convolution of graphs by means of the graph Fourier transform. This theory has been studied for decades in the field of signal processing (Chung, 1997; Hammond et al., 2011; Sandryhaila and Mouraj, 2013, 2014; Shuman et al., 2013, 2016). Its core idea is to expand the graph signal by its internal structures (i.e., Laplacian eigenvectors) rather than by simple oscillation functions (i.e., cosine and sine), and provide an analytic perspective of the spectra. Zhu and Michael (2012) theoretically explained why the Laplacian eigenvectors are used as the Fourier bases and discussed whether they are meaningful for arbitrary graphs. On the basis of the graph Fourier transform, researchers have attempted to generalize and extend CNNs to graph-structured data. For example, Duvenaud et al. (2015), Li et al. (2016) and Jain et al. (2016) introduced some machine learning frameworks for specific problems; Bruna et al. (2014) used a spectral convolution to define a multilayer neural network model, which is similar to the classical CNN; and Defferrard et al. (2016) and Kipf and Welling (2017) focused on a fast and canonical operation for spectral representation, e.g., locality of the convolution kernel.

In recent years, generalized CNNs have been applied to many graph-structured datasets, including Euclidean data, such as data for traffic

forecasting (Yu et al., 2018), 3D shape recognition and retrieval (Wang et al., 2018), and spatial pattern representation (Zhu and Liu, 2018), as well as non-Euclidean data, such as social media data (Rahimi et al., 2018), brain connectivity networks (Ktena et al., 2018), and chemical molecular structures (Gilmer et al., 2017). As typical data modeled by graph structures, spatial vector data can also be analyzed and processed using these novel concepts and technologies.

3. Methodology

A variety of specific patterns, e.g., regular or irregular patterns, can be distinguished from the spatial distribution of buildings. Regular groups are characterized by homogeneous structure, and they are regularly spaced and arranged visually, e.g., new formal planning settlements. Conversely, irregular groups have various shapes or an unstructured layout.

As with image processing, determining which pattern a building group visually belongs to is essentially an issue of classification. An analogy is drawn between a building group and an image, where a building is considered a pixel, and its semantic attributes and geometric characteristics are equivalent to color channels. In this study, CNN is applied to graphs for feature extraction and pattern classification of building groups.

3.1. Framework for building pattern classification

The overall framework for the classification of building patterns using GCNN consists of the following three parts (Fig. 1):

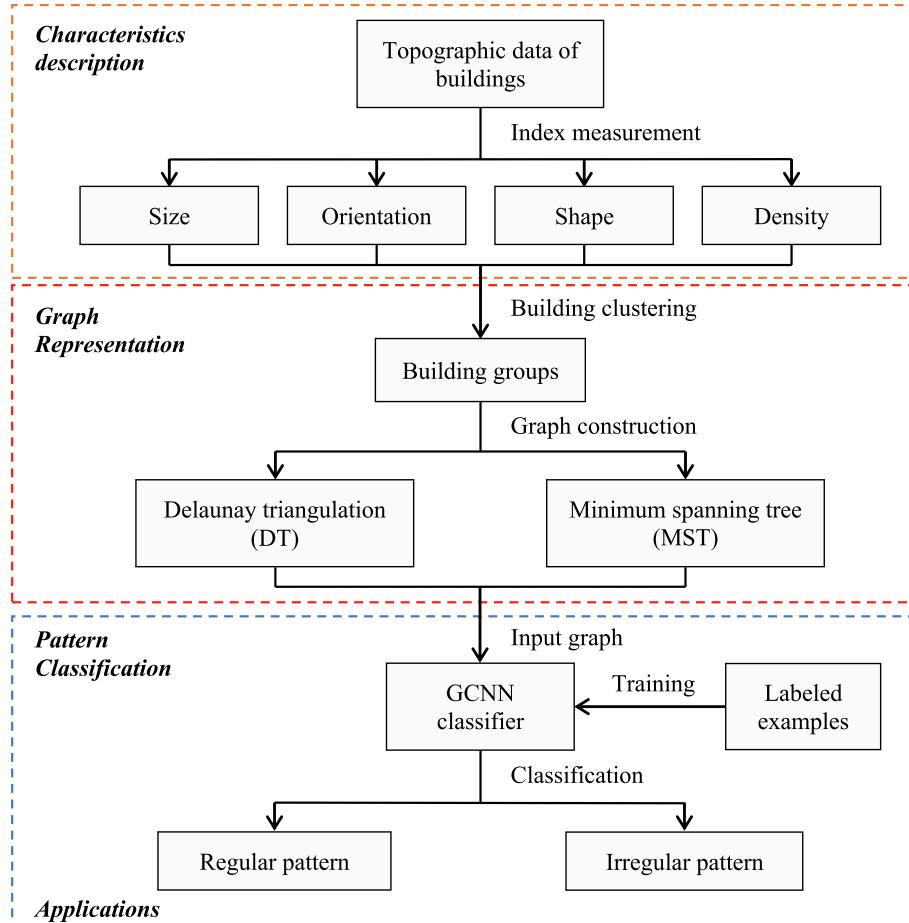


Fig. 1. Overall framework for pattern classification of buildings using GCNN.

- *Characteristics description for an individual building.* This preparatory part uses a set of indices to measure the semantic and geometric characteristics of an individual building.
- *Graph representation for a building group.* This part, which is also a preparatory stage, discovers the relational representation between buildings in a group after grouping adjacent or similar buildings with spatial clustering technologies.
- *Classification of patterns.* This part trains the GCNN model with sufficient examples labeled artificially, whereby the structural features for pattern classification can be extracted. Then, the patterns for any input building group that goes through the first two stages can be identified.

3.2. Description indices for individual buildings

The process of visually recognizing and discriminating perceptual patterns is usually controlled instinctively and subconsciously by the rules of psychological cognition, such as the Gestalt principles. A set of principles was developed to explain perceptual organization in Gestalt psychology, e.g., common fate and common orientation (Li et al., 2004). These organization principles can be described by Bertin's visual variables, including position, size, orientation, shape, value, color, and texture (Bertin, 1983). For the description of individual buildings, this study focus on three variables: size, orientation, and shape.

Many indices are used to compute each visual variable. For example, the orientation variable can be obtained by the orientation of a significant edge or the weighted orientation of all the edges (Duchêne et al., 2003). In particular, the shape variable of the building polygon is extremely complicated, diversified, and even subjective (Angel et al.,

2010; Basaraner and Cetinkaya, 2017; Li et al., 2013; Lombardo, 2014; Peura and Iivarinen, 1997). Moreover, there are some repetitions or strong correlations between these descriptive indices proposed by different researchers, e.g., two indices are reciprocal. Feature engineering can be used to analyze the importance of each index or the correlation between the indices, e.g., principle component analysis (Burghardt and Steiniger, 2005; Wei et al., 2018). In this study, some significant and representative indices are selected as shape descriptors by fully considering their potential relevance and relative ease of computation, use and interpretation.

In addition, semantic attributes and some high response variables, e.g., density, also reflect the status and form of a building (Batty and Longley, 1994). A valuable index for measuring the density of an individual building is the area ratio of the building to its impact area (Zhang et al., 2008). The impact area is defined as a space tessellation that captures the manner in which proximate buildings compete for space and reflects the entire pattern of a building group. For example, a building group with a regular pattern induces a homogeneous distribution of perceived densities. Furthermore, the density reveals the open space ratio and the building intensity in a certain region, which is conducive to the evaluation of the social function and development of the region (Batty, 2008).

Finally, a total of eleven indices (except for the position index) are considered for the size, orientation, shape, and density variables to describe individual buildings, and all or some of them are used as input variables for the model. Table 1 summarizes them, and the geometric shapes are shown in Fig. 2.

3.3. Graph representation for building group

An undirected and connected graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where \mathcal{V} and \mathcal{E} are finite sets of $|\mathcal{V}| = N$ vertices and edges respectively, and $\mathcal{W} \in \mathbb{R}^{N \times N}$ is an adjacency matrix that encodes the weights of the edges between every pair of vertices. Each vertex includes several variables representing the graph signals. In this study, the vertex variables represent the description indices as summarized above, and the

dimensions of the input variables equal the number of indices considered.

To construct the graph for the building data, the first step is to cluster the buildings into different groups under spatial context constraints, such as road networks. Spatial clustering, which has been studied extensively, is beyond the scope of the present study (Cetinkaya et al., 2015; Deng et al., 2018; Wang et al., 2015; Yan et al., 2008). The present study focuses on modeling and classifying building groups after spatial clustering, e.g., building blocks separated by road networks. This modeling process involves the connection mode and the distance measurement.

Many proximity graphs can be chosen to model the relations between the buildings in a group, such as k-nearest neighbor graph (k-NNG) and Gabriel graph. Different graph types have different degrees of connectivity, which correspond to different adjacency matrices. Two common graphs in the GIS field, namely Delaunay triangulation (DT) and minimum spanning tree (MST), are considered in this study. Their sets of edges satisfy the inclusion relationship $\mathcal{E}_{MST} \subseteq \mathcal{E}_{DT}$. Specifically, to generate the DT and MST, buildings are represented as vertices by using their centroids, and the line that connects two buildings' centroids is represented as an edge.

The distance discussed here is essentially the weight that is associated with an edge. With regard to its measurement, many previous studies have considered two factors: shape similarity and spatial proximity (Ai and Guo, 2007; Deng et al., 2018; Yan et al., 2008). The shape similarity distance mainly measures the homogeneity between buildings. The descriptions among the buildings are included implicitly in the learning of the graph upon summarizing these input variables for individual buildings, as will be discussed in Section 3.4. Therefore, only the proximity is considered, and the weight of an edge is updated with the nearest distance between the outlines of the corresponding pair of buildings. An alternative is the visible distance, which considers the orientation for calculating the average distance in the visible scope between two buildings (Ai and Guo, 2007). However, it is computationally more expensive and hence will be investigated in a future study.

Table 1

Description of spatial characteristics of individual buildings along with their equations and short descriptions.

Variable	Index	Notation/equation	Description
Position	Centroid	$(c_x, c_y) = \frac{1}{N} \sum_{i=1}^N (x_i, y_i)$	Arithmetic mean position of all building vertices, where N is the number of vertices
Size	Perimeter	P_b	Building perimeter
	Area	A_b	Building area
	Mean radius	$\frac{1}{N} \sum_{i=1}^N R_i$	Average distance from each vertex of the building to its centroid (Peura and Iivarinen, 1997)
Orientation	SBRO	—	Orientation of the smallest bounding rectangle (SBR)
	WSWO	—	The most frequent orientation among the wall orientations of the building with a small tolerance (Duchêne et al., 2003)
Shape	Compactness/circularity	$\frac{4\pi A_b}{P_b^2}$	Quadratic relationship between the area and the perimeter (Basaraner and Cetinkaya, 2017)
	Fractality	$1 - \frac{\log(A_b)}{2\log(P_b)}$	Logarithmic relationship between the area and the perimeter (Basaraner and Cetinkaya, 2017)
	Elongation	$\frac{L_{sbr}}{W_{sbr}}$	Length-width ratio of the building SBR (Burghardt and Steiniger, 2005)
	Concavity	$\frac{A_b}{A_{ch}}$	Area ratio of the building to its convex hull (CH) (Basaraner and Cetinkaya, 2017)
	Overlap index	$\frac{A_b \cap eac}{A_b \cup eac}$	Area ratio of the intersection and union between the building and its equal area circle (EAC) (Li et al., 2013)
Density	Area ratio	$\frac{A_b}{A_{ia}}$	Area ratio of the building to its impact area (IA) (Zhang et al., 2008). The impact area is approximated by a Voronoi-like diagram that is generated by skeletonizing the gap space among buildings on the basis of Delaunay triangulation

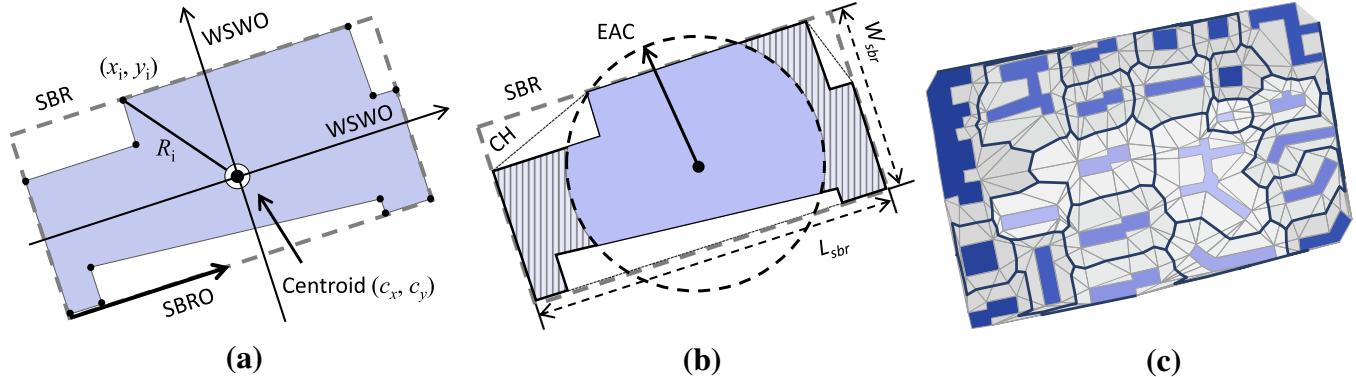


Fig. 2. Illustrations of indices for describing an individual building: (a) building position, size and two orientation indices of SBRO and WSWO, where SBR represents the smallest bounding rectangle and R_i denotes the distance from the vertex (x_i, y_i) to the centroid (c_x, c_y) ; (b) building shape indices, where CH and EAC represent the convex hull and the equal area circle, and L_{sbr} and W_{sbr} denote the length and width of the SBR, respectively; (c) building density index, where the colors denote the density values of the building.

Finally, a graph for a given building group can be constructed with DT or MST, as shown in Fig. 3.

In addition to the basic form of the vertices and edges, the graph can be represented by its Laplacian matrix \mathcal{L} , defined as $\mathcal{L} = D - \mathcal{W}$, where $D = \text{diag}(d_0, \dots, d_{N-1})$ is the degree matrix formed by the degrees $d_i = \sum_j \mathcal{W}_{ij}$ of vertex i . Because \mathcal{L} is a symmetric semi-definite matrix, it has a complete set of orthonormal eigenvectors, denoted as $\{\chi_\ell\}_{\ell=0}^{N-1}$, with associated nonnegative eigenvalues $0 \leq \lambda_0 \leq \dots \leq \lambda_{N-1}$. \mathcal{L} is diagonalized by the matrix of eigenvectors $\mathcal{X} = [\chi_0, \dots, \chi_{N-1}]$ such that $\mathcal{L} = \mathcal{X} \Lambda \mathcal{X}^T$ where Λ is the diagonal matrix of eigenvalues. A normalized version is $\tilde{\mathcal{L}} = I_N - D^{-1/2} \mathcal{W} D^{-1/2}$, where I_N is the identity matrix of size N , and accordingly the range of eigenvalues is $[-1, 1]$. In this study, \mathcal{L} stores the connections between buildings and their spatial distances.

3.4. Graph convolutional neural network

In this section, a convolution operation on a graph is defined with the graph Fourier transform, and the learning architecture of a GCNN is built for the classification of building patterns.

3.4.1. Graph Fourier transform

The Fourier transform, which is an effective tool for signal analysis and image processing, decomposes the original function (e.g., a signal or image) into frequencies. The process is essentially a linear transformation of a function using the orthogonal complex exponentials $e^{i\omega t}$ as the decomposition bases.

For graph-structured data, the eigenvectors $\{\chi_\ell\}_{\ell=0}^{N-1}$ of the Laplacian matrix \mathcal{L} that satisfy the orthogonality condition are used as the decomposition bases instead of the complex exponentials. The Fourier transform of a given signal $f(n)$ (e.g., an input variable in this study) on a graph is defined as

$$\hat{f}(\lambda_\ell) = \sum_{n=0}^{N-1} \chi_\ell^T(n) f(n) = \mathcal{X}^T f \quad (1)$$

The inverse Fourier transformation is given by

$$f(n) = \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell) \chi_\ell(n) = \mathcal{X} \hat{f} \quad (2)$$

This definition is precisely analogous to the classical case, and it can be interpreted as an expansion of $f(n)$ in terms of the eigenvectors of

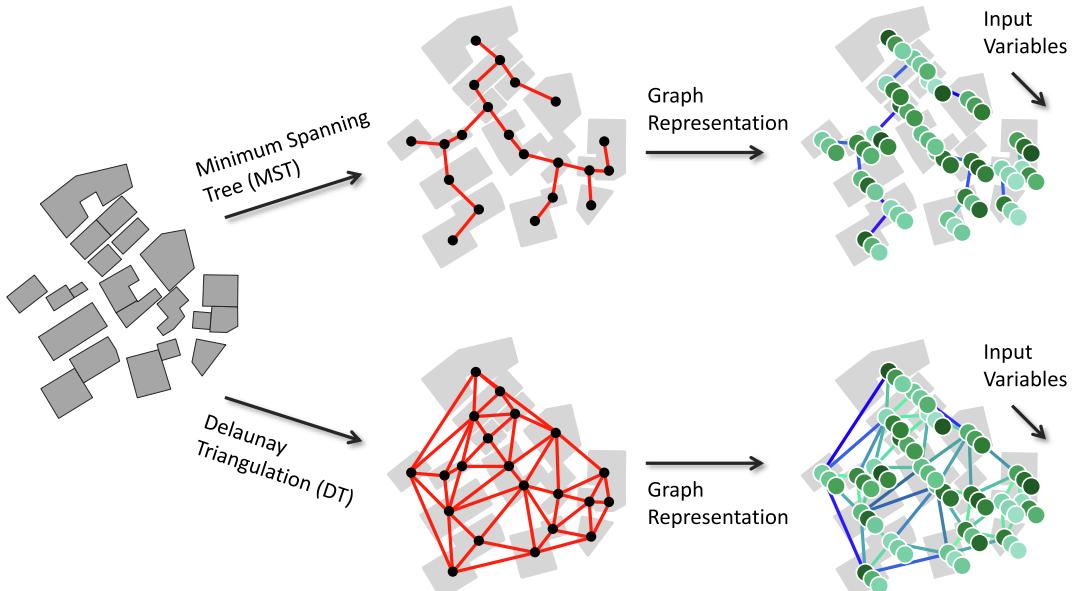


Fig. 3. Graph representation for a building group constructed with DT and MST. In the constructed graphs, each vertex includes three dimensions representing three input variables.

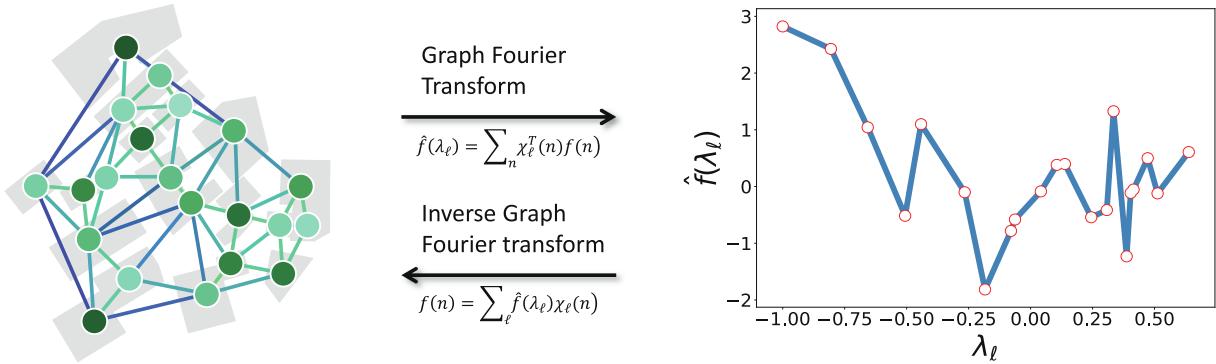


Fig. 4. Fourier transform and its inverse transform for graph-structured data.

the Laplacian (Hammond et al., 2011; Shuman et al., 2016), as seen in Fig. 4.

3.4.2. Convolution operation on graphs

On the basis of the graph Fourier transform and convolution theorem, the convolution can be first converted into a point-wise product in the Fourier domain, and then reconverted into the vertex domain as follows:

$$\begin{aligned} f * g &= \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell) \hat{g}(\lambda_\ell) \chi_\ell(n) = \mathcal{X}((\mathcal{X}^T f) \cdot (\mathcal{X}^T g)) \\ &= \mathcal{X} \text{diag}(\hat{g}(\lambda_0), \dots, \hat{g}(\lambda_{N-1})) \mathcal{X}^T f \end{aligned} \quad (3)$$

Fig. 5 shows this graph convolution operation of two graph signals $f(n)$ and $g(n)$; $g(n)$ and its transform, $\hat{g}(\lambda_\ell)$, is considered as a convolution kernel. This kernel can be designed by a set of free parameters $\{\theta_\ell\}_{\ell=0}^{N-1}$ in the Fourier domain, i.e., Laplacian eigenspace. Furthermore, it can be considered a function of eigenvalues, denoted as $\mathcal{G}(\Lambda)$. Then, the convolution is expressed as

$$f * g = \mathcal{X} \text{diag}(\theta_0, \dots, \theta_{N-1}) \mathcal{X}^T f = \mathcal{X} \mathcal{G}(\Lambda) \mathcal{X}^T f \quad (4)$$

3.4.3. Polynomial approximation for fast localized convolution

The convolution on a graph as described above has two limitations: (1) eigendecomposition is performed in each operation, which incurs considerable computational costs; (2) without considering the locality in space, the variable value of a vertex can be related to the global vertices after this operation, which is not consistent with the local connections of CNNs (Defferrard et al., 2016).

To address these two problems, Hammond et al. (2011) proposed a fast localized convolution based on low-order polynomial approximation that represents $\mathcal{G}(\Lambda)$ as a polynomial function of the eigenvalues:

$$\mathcal{G}(\Lambda) = \sum_{k=0}^K \theta_k \Lambda^k \quad (5)$$

where $\{\theta_k\}_{k=0}^K$ is a vector of polynomial coefficients and K is the polynomial order. K can be a small positive integer, e.g., 3. Then, the convolution is rewritten as

$$f * g = \mathcal{X} \left(\sum_{k=0}^K \theta_k \Lambda^k \right) \mathcal{X}^T f = \left(\sum_{k=0}^K \theta_k (\mathcal{X} \Lambda^k \mathcal{X}^T) \right) f = \sum_{k=0}^K \theta_k \mathcal{L}^k f \quad (6)$$

The computation is accelerated as the eigendecomposition

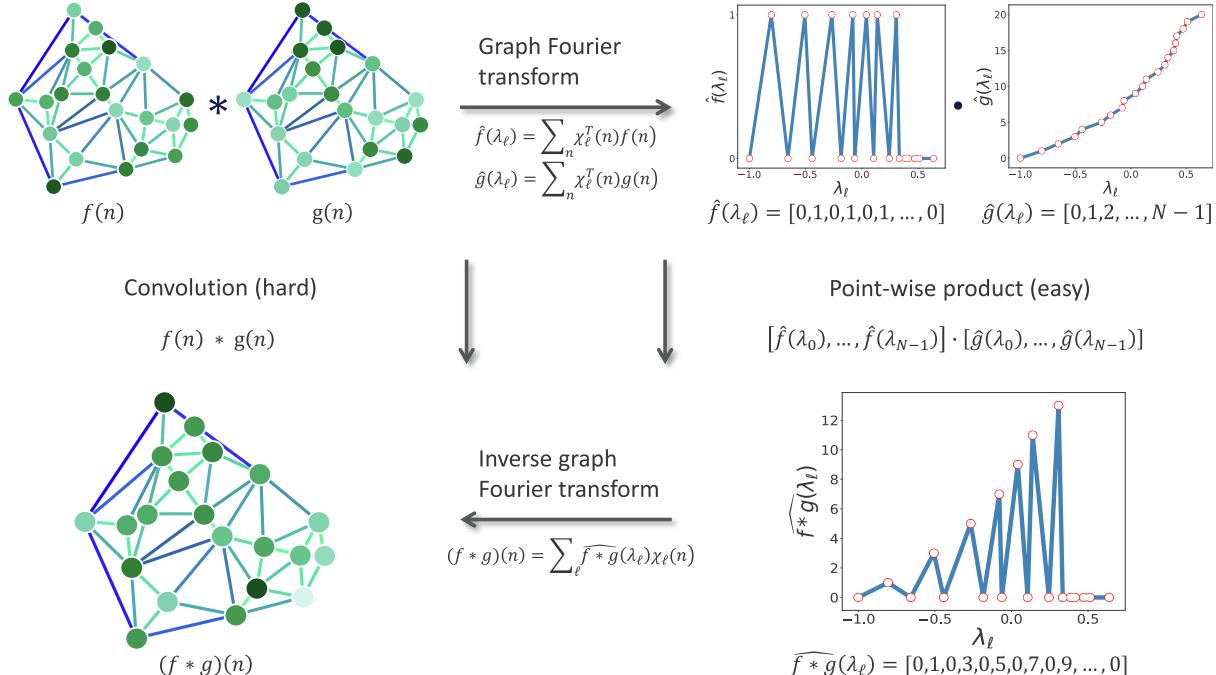


Fig. 5. Graphical illustration of the convolution of two graph signals $f(n)$ and $g(n)$. Their Fourier transforms are $\hat{f}(\lambda_\ell) = [0, 1, 0, 1, 0, 1, \dots, 0]$ and $\hat{g}(\lambda_\ell) = [0, 1, 2, \dots, N - 1]$, respectively.

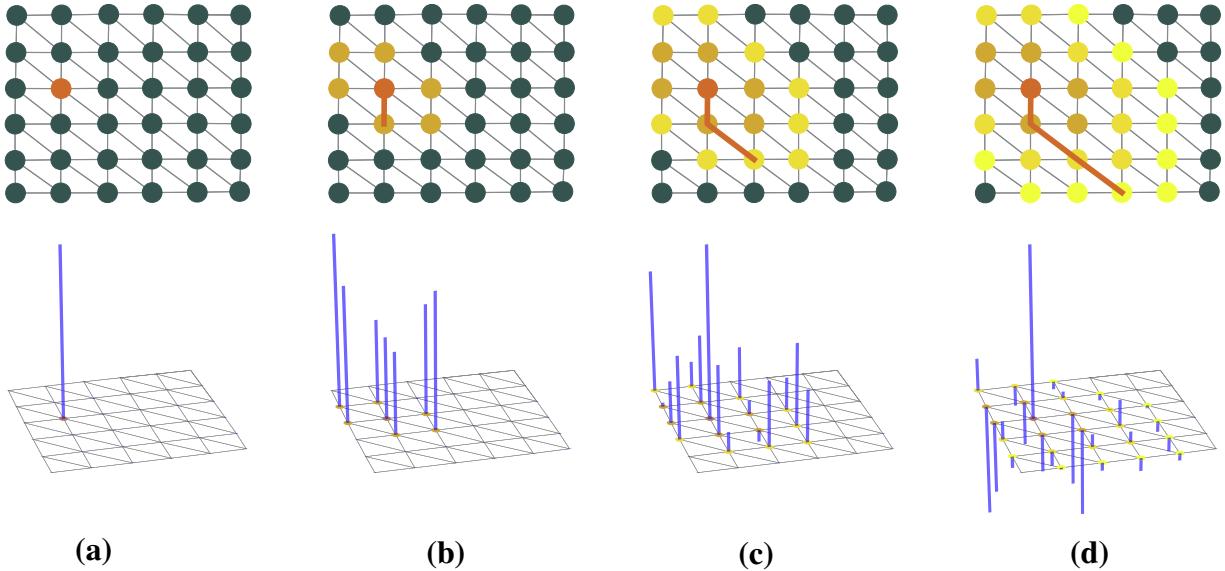


Fig. 6. Localized convolution based on polynomial approximation: (a) a graph signal that has a value on only one vertex; (b), (c), and (d) respectively show the neighbors of this vertex (yellow vertices) and the distance from the farthest neighbor to it (brown line) in the top row, and the convolved results (blue lines) in the bottom row with 1-order, 2-order, and 3-order polynomial kernels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

operation is avoided and the convolution is conducted by K multiplications of the sparse matrix \mathcal{L} . In addition, the variable values of a vertex are related only to its K -order neighboring vertices (as seen in Fig. 6), which means that the graph convolution exhibits the local connection property in space. In other words, the locality of space is equivalent to the smoothness in the Fourier domain: the smoother the eigenvalues, the lower is the polynomial order and the smaller are the neighborhoods in space, and vice versa.

3.4.4. Architecture of the convolutional neural network on graphs

The present study proposes a learning model on graph-structured data for the classification of building patterns by combining the graph convolution with the computational mode of the neural network. The model consists of multiple convolutional and pooling layers and one fully connected layer. The architecture of the model with two convolutional layers is shown in Fig. 7.

The j^{th} output graph of the $(l+1)^{th}$ layer activations, i.e., the j^{th} dimensional feature, follows a layer-wise propagation rule that is

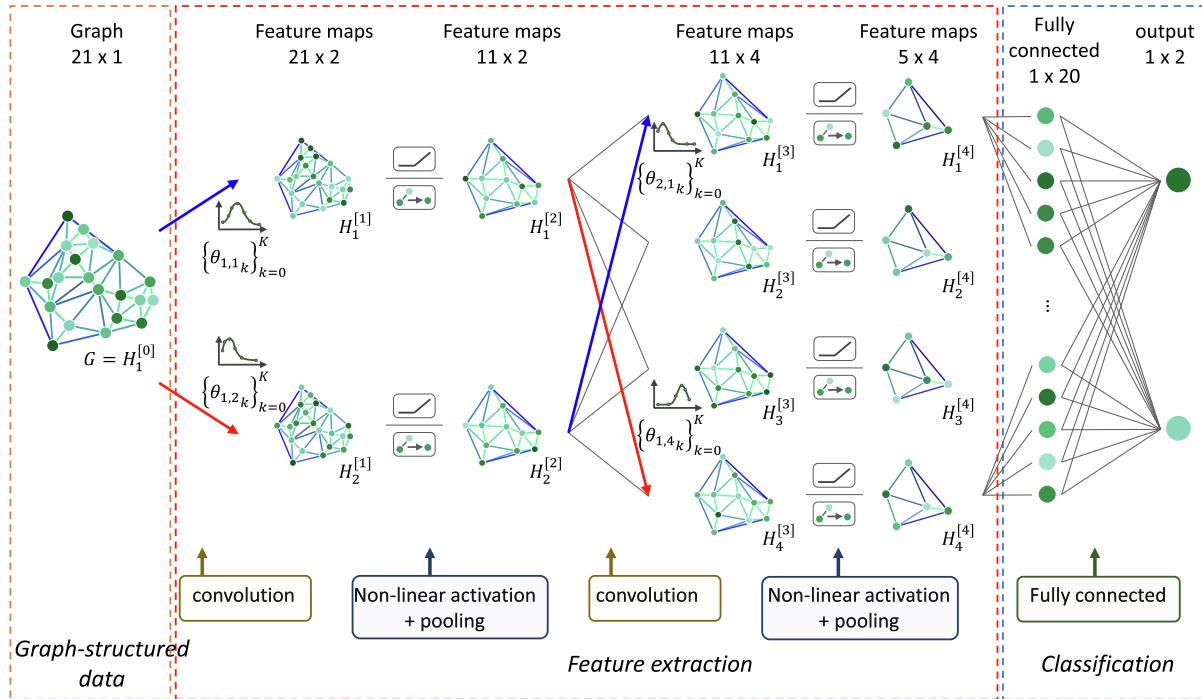


Fig. 7. Architecture of the convolutional neural network on graph-structured data. The left volume stores the graph representation of an input group including 21 buildings and one variable. The middle volume stores the features maps of two convolutional layers (including 1×2 and 2×4 K -order polynomial kernels) and two pooling layers. The right volume stores the fully connected layer and class scores.

expressed as

$$H_j^{[l+1]} = \sigma \left(\sum_{i=1}^{F_{in}} \left(\sum_{k=0}^K \theta_{i,j,k} \mathcal{L}^k H_i^{[l]} \right) + b_j^{[l]} \right) \quad (7)$$

where $\sigma(\cdot)$ denotes a non-linear activation function, e.g., $ReLU(\cdot) = \max(0, \cdot)$; $H_i^{[l]}$ denotes the i^{th} input graph of the l^{th} layer activations; and $\theta_{i,j,k}$ and $b_j^{[l]}$ are the trainable $F_{in} \times F_{out}$ vector of K -order polynomial coefficients and $1 \times F_{out}$ vector of bias in the l^{th} layer, respectively.

The pooling layers calculate the statistics of local neighborhoods into a single vertex in the output to reduce the complexity of the model. Such neighborhoods at different scales can be constructed by using multi-resolution graph clustering, e.g., spectral clustering (Henaff et al., 2015) or Graclus multilevel clustering (Dhillon et al., 2007). Considering the computational efficiency, the pooling layer is optional in this architecture. The fully connected layer is the same as that in CNNs. The probability distribution across the potential patterns is obtained through forward propagation in this architecture for an input graph of a building group. Finally, the most probable pattern can be chosen as the final classification result.

During the training of the GCNN, each input example includes two sets of outputs: the predicted value \hat{y} by this model and the actual label value y that encodes the pattern class. The performance is monitored and measured by the loss function $J(\hat{y}, y)$, i.e., the difference between these two values, which can be calculated in many ways, e.g., cross-entropy.

The network is optimized by minimizing the loss function for each example or a batch of examples, and the parameters are updated by the gradient back-propagation algorithm. In this process, the gradient computation of the loss function and activation function is the same as it is in the case of CNNs, except for the convolution operation. As seen from Eqs. (6) and (7), the convolution operation in each layer only involves the addition and multiplication of the input graph and its parameters; thus, the two gradients are obtained easily without considering the activation function $\sigma(\cdot)$:

$$\frac{\partial J}{\partial \theta_{i,j,k}} = \mathcal{L}^k H_i^{[l]} \frac{\partial J}{\partial H_j^{[l+1]}}, \text{ and } \frac{\partial J}{\partial H_i^{[l]}} = \sum_{j=1}^{F_{out}} \left(\frac{\partial J}{\partial H_j^{[l+1]}} \left(\sum_{k=0}^K \theta_{i,j,k} \mathcal{L}^k \right) \right)$$

The entire network has the same learning complexity as the classical CNN (LeCun et al., 1998).

4. Experiments and results

The proposed GCNN is implemented with Python in TensorFlow. All the code has been released under a GPL license and distributed through GitHub (<https://github.com/xiongfengyan/gcnn>).

This section describes a series of experiments on large building datasets, conducted to test the effectiveness, robustness, and generalization ability of the proposed method.

4.1. Experimental settings and data preprocessing

The experimental data were extracted from a large-scale 1:2000 topological map of Guangzhou City, which is located in southern China. This experimental region covers areas of different graphic and geographical characteristics, including urban, suburban, and rural areas.

The buildings were divided into different groups using a road and river network and proximity-based spatial clustering technologies (Deng et al., 2018), with 20–128 buildings per group. All the groups were manually identified as regular or irregular patterns. Each group was estimated by at least three participants to ensure the correctness of the classification results. Ambiguous groups and mixed groups were discarded to obtain more stable training. For example, a group was considered ambiguous if one participant voted it as a regular pattern while the other two voted it as an irregular pattern, and it was considered mixed if one part of the group belonged to a regular pattern while the other part belonged to an irregular pattern. Finally, 2647 and 2646 available groups were labeled for regular and irregular patterns, respectively, and they included a total of 318,641 buildings, as seen in Fig. 8.

Each building group can serve as an example for the GCNN, and all the examples were randomly split into training, validation, and test sets in a proportion of 6:2:2. The fully connected layer contains a fixed

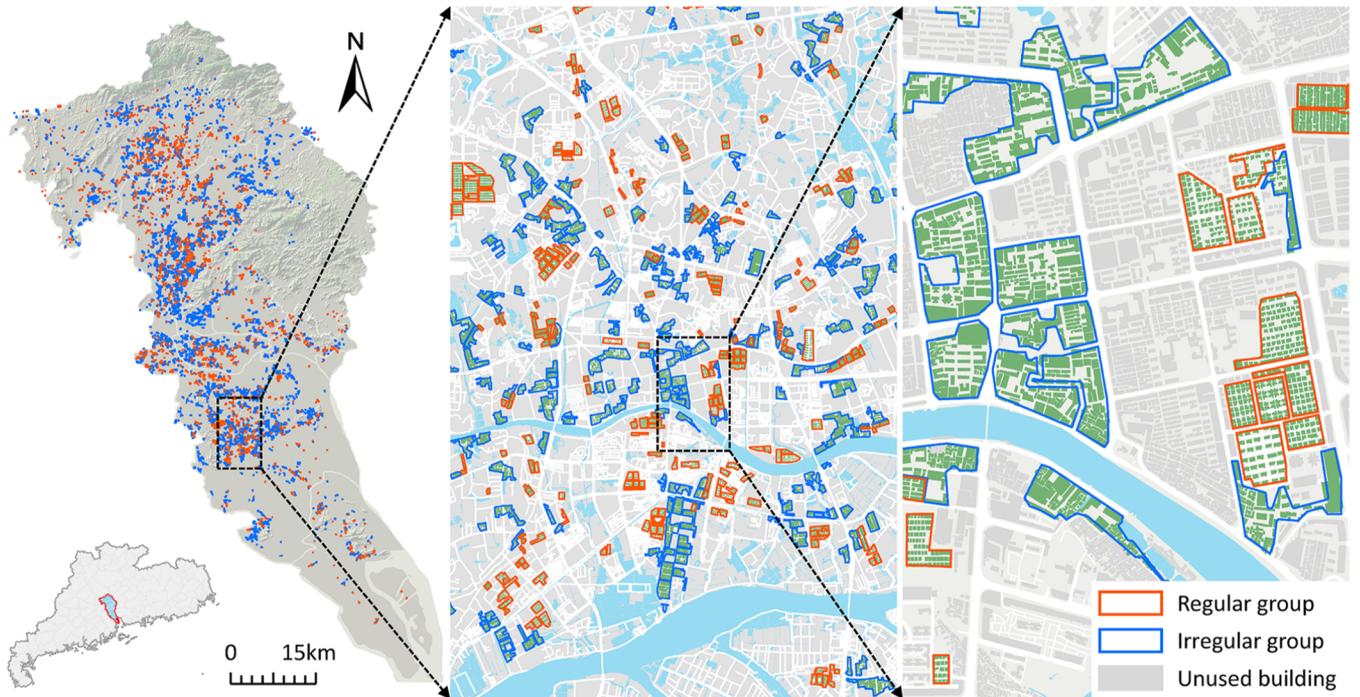


Fig. 8. Experimental dataset of Guangzhou City.

Table 2

Confusion matrix of the pattern classification experiment with the Guangzhou City dataset.

Number of examples = 1059	Actual regular	Actual irregular
Predicted regular	523	7
Predicted irregular	7	522

number of neurons, whereas the number of buildings in each group could be different; hence, fake vertices were added to cases for which the numbers were insufficient compared to the maximum. No edges were connected to these fake vertices and the input variables were padded with zeros. The input variables were normalized using the Z-score method, where the arithmetic mean and standard deviation were calculated by all buildings in the training set.

4.2. Pattern classification

For a pattern classification task, the model outputs a probability vector $\{P_i\}_{i=0}^{M-1}$ across M classes. The class with the largest P is the final classification result. In general, if this classification is the same as the label, it is correct; otherwise, it is incorrect. In addition, we can set a stricter condition and only accept those classes with the probability greater than a given threshold (e.g., 0.8) as the correct results. Finally, the accuracy is measured by the ratio of the number of examples classified correctly to the total number of examples tested.

Delaunay triangulation was used to represent these examples as input graphs. A shallow GCNN architecture with five convolutional layers and one fully connected layer was used to test the datasets. Each convolutional layer contained 16 third-order polynomial kernels. The Adam algorithm was selected for the optimization, with a learning rate of 0.001 and an exponential decay rate of 0.95. The hyper-parameters of regularization weight, dropout probability, and mini-batch size in the model took values of $5e-4$, 0.5, and 100, respectively.

After 50 training epochs, the accuracies of the training and test sets were 99.15% and 98.68%, respectively, and the losses were 0.074 and 0.127, respectively, with the corresponding confusion matrix shown in Table 2. The close accuracies indicate that the GCNN trained by the training set has a strong generalization ability and good applicability to other datasets. Furthermore, when we set 0.8 as the acceptance threshold, the test accuracy was 98.02%, which was close to the general case, clearly indicating that most of the classification results are correct.

Fig. 9 shows the accuracies and loss changes of the validation set over time to better evaluate the stability of the proposed model. It can be seen that, with the training steps, the accuracy increases while the loss value decreases gradually, and that both converge after approximately 400 steps of training. With further training, the accuracy with

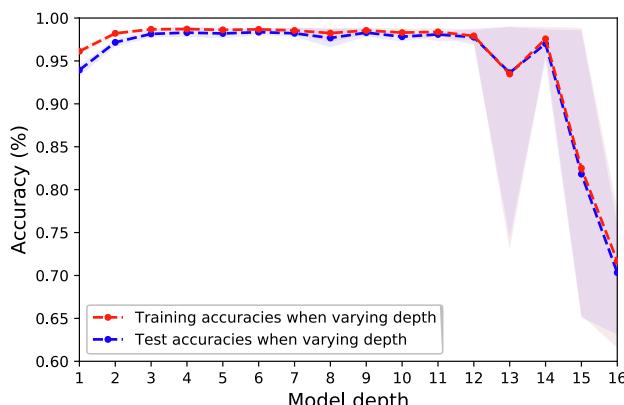


Fig. 9. Accuracies and loss changes of the validation set over time.

the acceptance threshold of 0.8 gradually approaches the general accuracy, which indicates that the model is still being optimized step by step. Finally, both accuracies exceed 98%, which indicates that the model is well generalized on the validation set.

More specifically, Fig. 10 shows the activations of the used GCNN architecture in these experiments. The left volume stores the input graph of the building group and its 11 variables, the middle volume records the five-layer feature maps, and the right volume holds the probabilities for each class. As with CNNs, the input to-be-recognized graph is filtered by several convolutional layers and finally mapped to output probabilities by the fully connected layer. As seen in Fig. 10, the Fourier-transformed values of the feature maps become more stable and sparser during this processing. Moreover, in most of the feature maps, only the first few eigenvalues hold the transformed values while the latter ones are zero or close to zero. This finding can be understood as follows: the model retains the main features extracted from the training set and gradually ignores other information. At the same time, the convolution kernels do not need to be designed manually in advance; they continuously calibrate their weights through the training of a large number of labeled examples. The process of calibrating the weights until they are stable is a convergence process, which enables the model to distinguish the specific patterns from different input variables and connection modes.

To further test the validity and generalization ability of the model, an additional experiment was conducted using the trained model described above to classify the building patterns in another region. This experimental region was extracted from Shanghai City, which is located in eastern China and includes residential and commercial areas. The dataset was preprocessed in the same way as the Guangzhou dataset, and 1135 groups were obtained, with a total of 68,663 buildings. All the groups were evaluated and a probability value was given to observe the pattern distribution of the entire region, as shown in Fig. 11. The actual labels were determined by the participants' votes and the classification accuracy was 94.71%; the confusion matrix is shown in Table 3. As seen in Fig. 11, the newly planned residential houses in the lower left corner have similar structures and neat arrangements, and they achieved high probabilities of being classified as regular patterns. A science park is in the top right corner with complex buildings and different shapes, which was classified as an irregular pattern. This result is consistent with our visual perception, which indicates that the classification ability of the model is mainly driven by the training data. As long as a group is recognized visually as a certain type of pattern, the model can distinguish it correctly even if the geographical area changes.

Furthermore, a type of noise that has a significant impact on the classification is found by analyzing these misclassified building groups, as shown in Fig. 12(a). When a narrow building extends along the edge of a group, the performance begins to deteriorate. As a result, the group is mistakenly classified as an irregular pattern, whereas it belongs to a regular pattern by manual classification. Moreover, the ambiguous and mixed cases (as discussed earlier) possibly cause unsatisfactory classification results, as shown in Fig. 12(b) and (c). In the Guangzhou dataset, these types of groups were discarded for stable training, but they were evaluated during the additional experiment, which could be the reason for the slightly lower accuracy of the additional experimental dataset compared to the training accuracy.

4.3. Parameter sensitivity analysis

The model was experimentally investigated in terms of its sensitivity to several influential parameters, including the model depth, graph structure, polygonal order K , and input variables.

As only the convolutional layer and fully connected layer are used in our architecture, the model can be varied by adding or reducing the convolutional layer, and the model depth represents the number of convolutional layers. Each convolutional layer also contains 16 third-

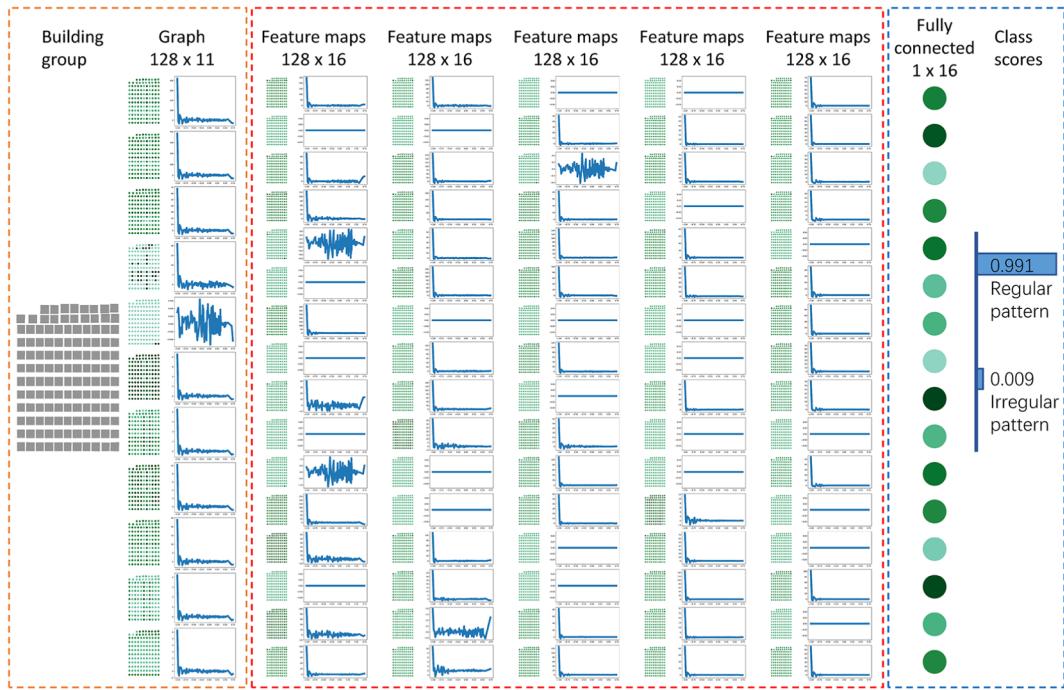


Fig. 10. Activations of the GCNN architecture used in our experiments. The values on the graph vertices are represented in green, while the blue lines denote the corresponding Fourier transforms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

order polynomial kernel, and the optimization algorithm and other hyper-parameters remain unchanged. Fig. 13 reports the results of a 5-fold cross-validation experiment on the Guangzhou dataset. The performance initially improves with increasing depth and then stabilizes.

However, it deteriorates sharply when the model is deeper than 12 layers. The inherent instability of the gradients in the back-propagation algorithm may be the plausible cause of deep GCNN learning difficulties (the vanishing or exploding gradients problem), which also arise in the

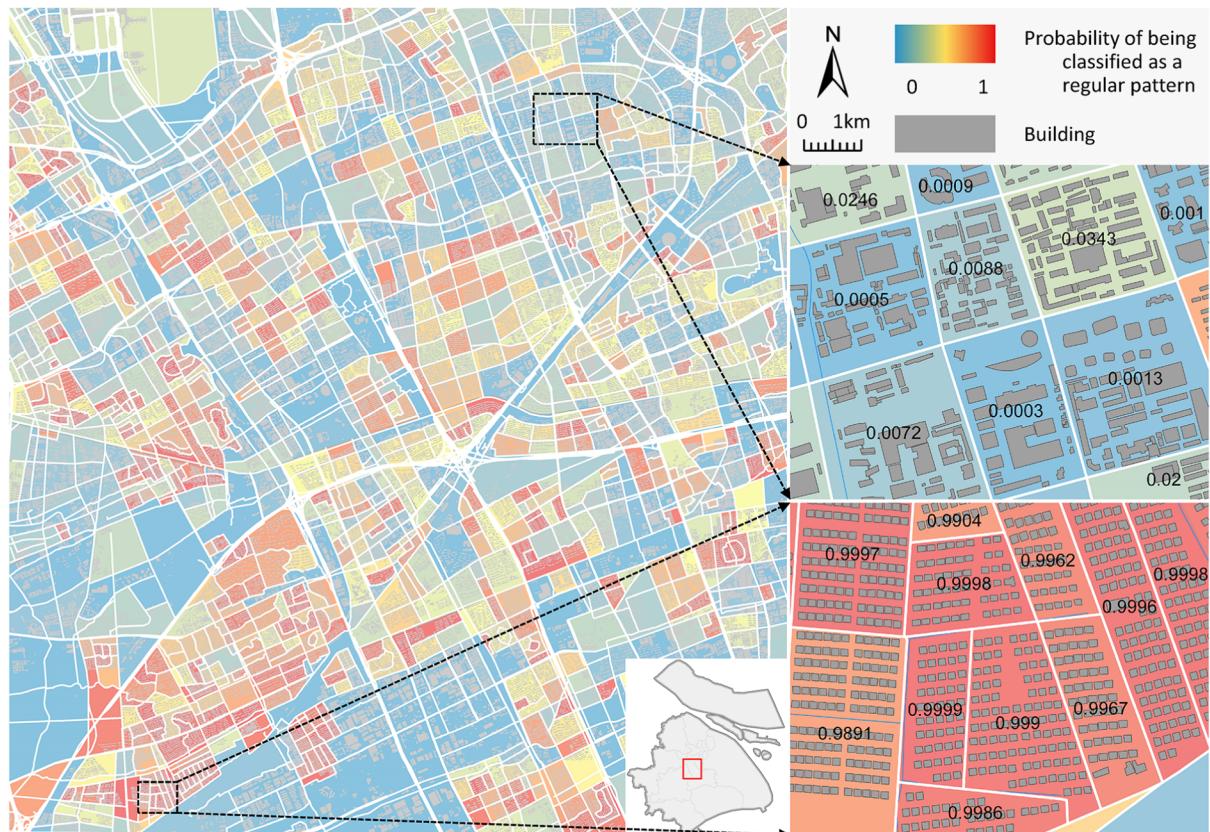


Fig. 11. Classification results of the building groups in the Shanghai dataset. The labeled values represent the output probability of being classified as a regular pattern.

Table 3

Confusion matrix of the additional experiment with the Shanghai dataset.

Number of examples = 1135	Actual regular	Actual irregular
Predicted regular	534	20
Predicted irregular	40	541

case of CNNs. A possible solution is to use residual connections to train deep networks (He et al., 2016).

Fig. 14 shows the performances on the validation dataset with different polynomial orders K (i.e., from one to six) when using the DT and MST to construct the graph. By comparison, the graph structure does not influence the classification performance significantly, whereas the polynomial order K does. When K is 1, the performance is poor; it begins to improve as K increases to 2. At the same time, different graph structures have different sensitivities to K . In other words, the K value for the best performance of the model constructed by DT is 3, whereas that for the model constructed by MST is 4. This finding can be explained by the fact that MST is sparser than DT; thus, larger neighborhoods are required to detect the local features. In addition, the larger the value of K , the more complex and time consuming is the training.

Fig. 15 provides a comparison by using only one descriptive index at a time or all the indices except for one as the input variables to investigate the importance of the input variables to pattern classification. When using ten indices as the input variables, the accuracy reached 98% regardless of which index was not used. This finding indicates that no index is indispensable; they are related and can be replaced or supplemented by each other on the basis of the approximation of the proposed model. When using one index as an input variable, the size and density indices showed the best performance while the directional indices performed poorly. On the one hand, this finding could be attributed to the orientation of the building polygon, which is complicated and difficult to quantify; on the other hand, it indicates the importance of the area and density in describing the perceived patterns. The accuracies of the shape indices ranged from 85% to 95%.

4.4. Comparative analysis

To better examine the advantages and disadvantages of the proposed method, two existing methods were implemented for comparison: a support vector machine (SVM) based on several input variables that describe the characteristics of building groups (L. Zhang et al.,

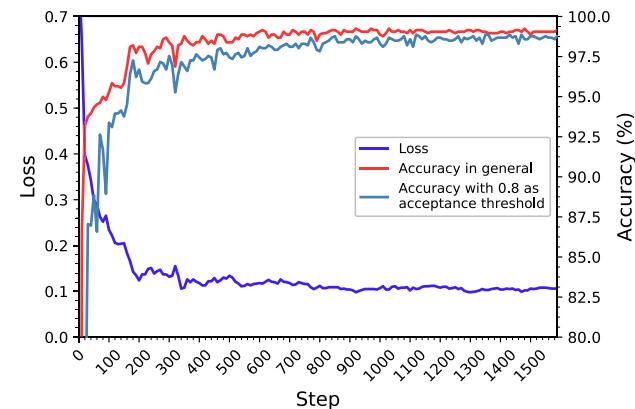


Fig. 13. Model performances varying the model depth. The mean classification performances of 5-fold cross-validation are plotted as the dashed line and their variances are shown by the upper and lower bounds.

2013), and a random forest (RF), which mainly considers the Gestalt principles of proximity, continuity, and integrality as input variables (Hecht et al., 2015; He et al., 2018; Jochem et al., 2018).

For fair comparison, only the area, SBR0, compactness, and concavity indices were considered as input variables for the GCNN method. For the SVM and RF methods, five features were extracted to describe the building group, including the standard deviation of the weights of all the edges, the area ratio of the building to the smallest bounding rectangle of the group (He et al., 2018), area difference, orientation difference, and similarity difference. The area difference was calculated by taking the ratio of the standard deviation σ_A to the average the building areas within the group \bar{A} , denoted as $\sqrt{\sum(A_i - \bar{A})^2/N}/\bar{A}$, and the orientation difference was calculated in a similar manner (L. Zhang et al., 2013). The turning function (Arkin et al., 1991) was adopted to measure the similarity S_{ij} between buildings i and j , and the similarity difference of the group was derived from $\sum S_{ij}/N(N - 1)$.

A part of the Guangzhou dataset was used as the training set, and another part of the Guangzhou dataset and the entire Shanghai dataset were used as two separated test sets. The results are summarized in Table 4.

A comparison of these accuracies on the Guangzhou dataset shows that all the three methods achieve good classification performance. Although only four indices were used for the GCNN, its accuracy still reached 98.0%, which is higher than that of the other two methods. Further, comparing these accuracies on the Shanghai dataset, the

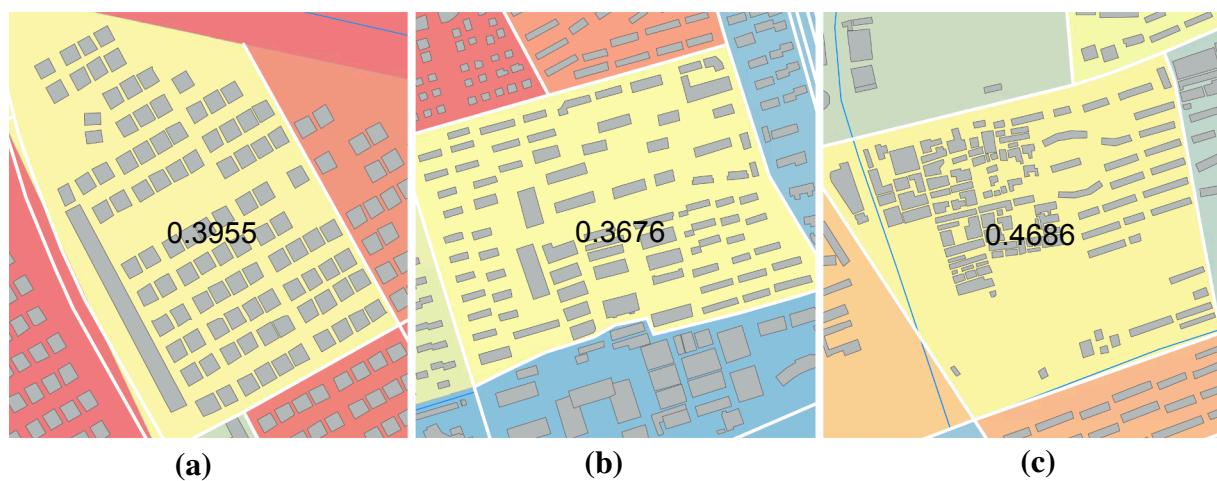


Fig. 12. Examples of misclassified groups: (a) a group with a lower probability of being classified as regular pattern owing to interference from narrow buildings, (b) an ambiguous group, and (c) a mixed group. The labeled value has the same meaning as in Fig. 11.

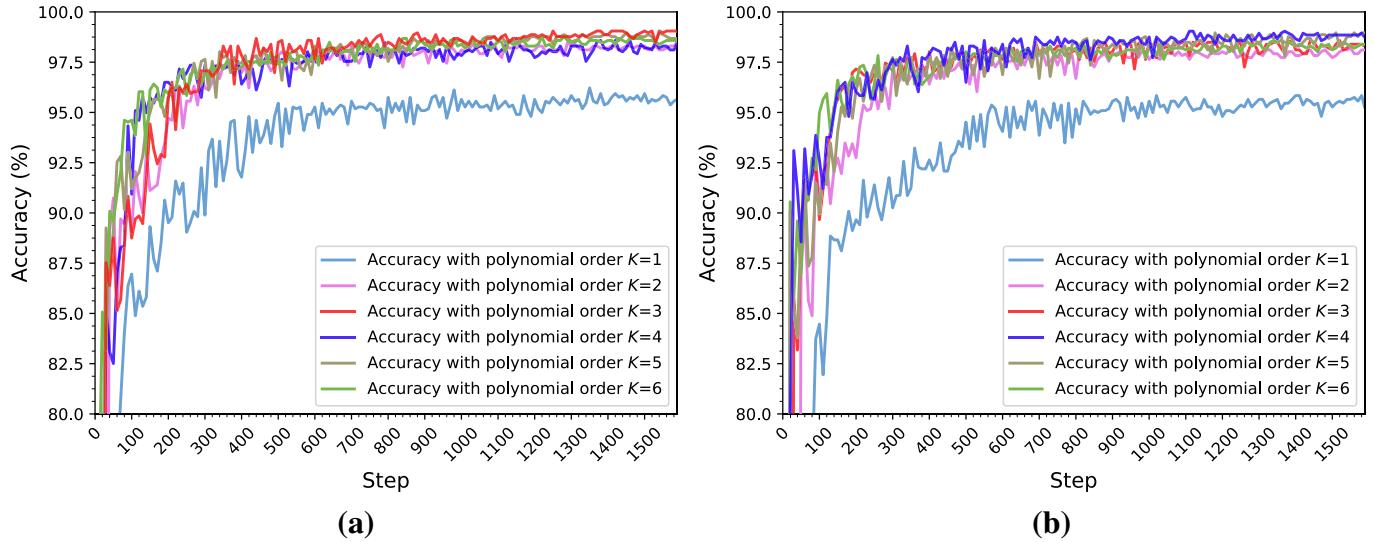


Fig. 14. Model performances: (a) and (b) respectively show the performances with difference polynomial orders K using DT and MST to construct the graph.

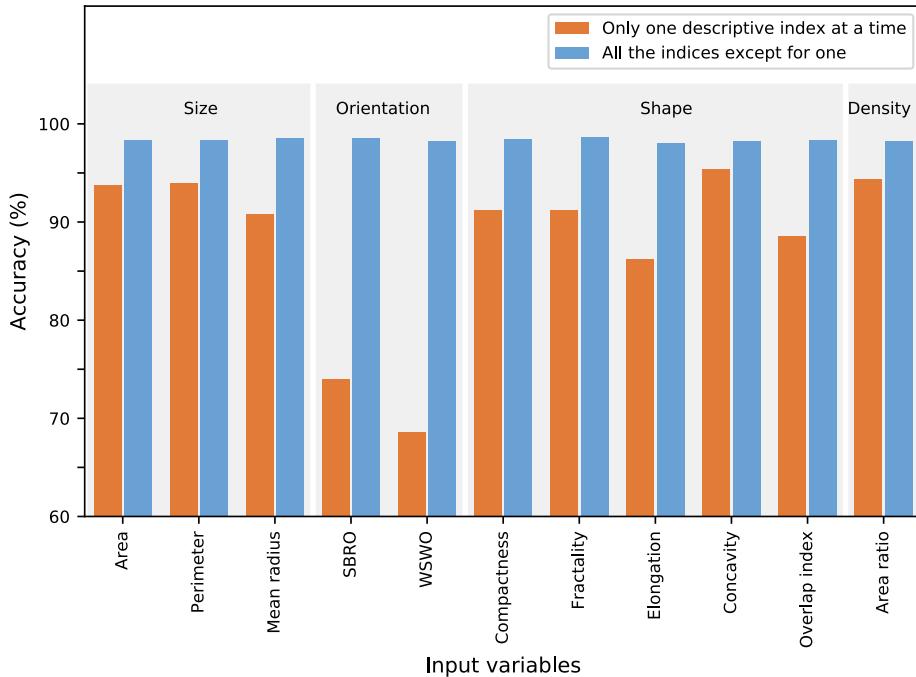


Fig. 15. Model performances taking different indices as input variables.

Table 4

Accuracies of the proposed GCNN model and other algorithms.

Method	Test accuracy of the Guangzhou dataset	Test accuracy of the Shanghai dataset
SVM	92.93%	75.59%
RF	95.14%	87.75%
GCNN	98.02%	92.60%

accuracies of the proposed method and the RF method were relatively high, which implies that the generalization ability of these methods is stronger than that of the SVM method. When the geographic region or scale of data changes, the features extracted are still applicable. By summarizing the classification performance and generalization ability, it can be concluded that the proposed method outperforms the other methods in building pattern classification.

5. Discussion

5.1. Rule-based method vs. machine learning

The basic ideas and processes of the rule-based method, traditional machine learning method, and proposed method for building group pattern classification are summarized in Fig. 16.

The rule-based method defines features and design rules manually to extract specific patterns. One obvious disadvantage of this method is that the rules are too rigid and the parameters must be set and calibrated manually, which not only is time consuming but also cannot guarantee the classification accuracy. Moreover, the usable range of a pattern is limited, and a set of well-defined rules may be ineffective when the scales or geographical characteristics of the data change. For example, rules used to identify patterns for urban areas might not be suitable for rural areas.

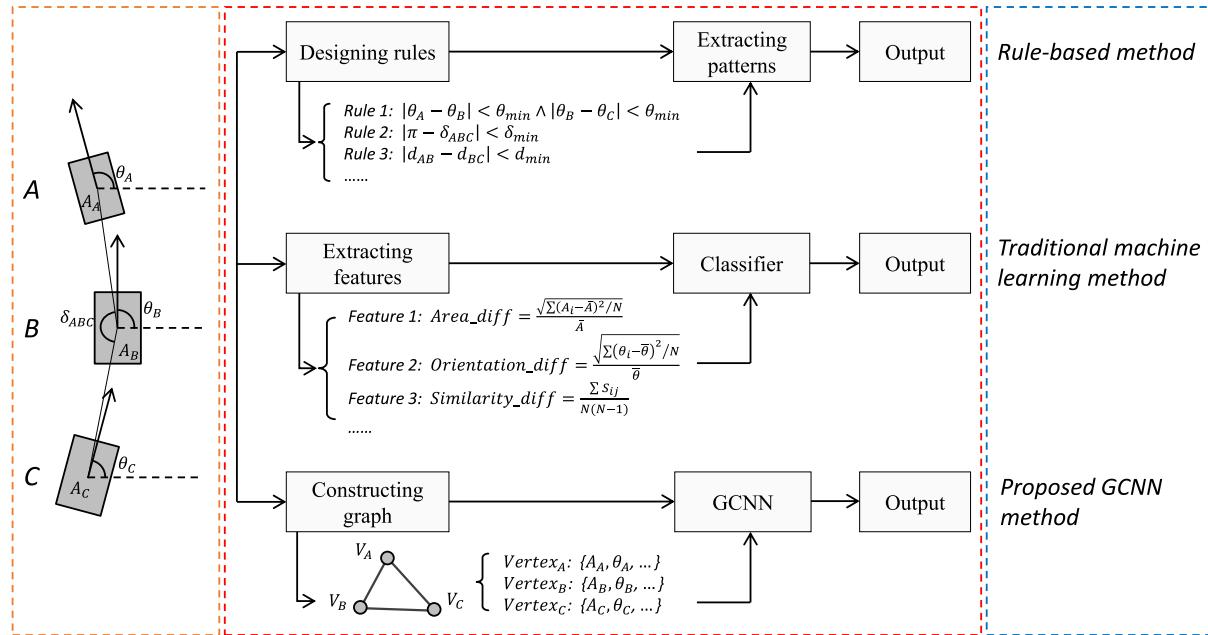


Fig. 16. Basic ideas and processes of three types of methods for building pattern classification. θ_i and A_i respectively denote the orientation and area of building $i \in \{A, B, C\}$, δ_{ABC} denotes the azimuth angle of the three buildings, and S_{ij} denotes the similarity between building i and building j .

A good machine learning algorithm can overcome these drawbacks. As the algorithm is data-driven, it is adaptive to data when finding the parameters that can produce better results, and it iterates faster. On the other hand, we are not limited to defining specific patterns within a certain range. In other words, the machine learning algorithm can be used to recognize various patterns at different scales or in different regions as long as examples are available.

However, existing machine learning methods for classifying building group patterns must still manually define some features to describe the group units. These handcrafted features have a significant impact on the final classification results, and rich experience and extensive knowledge are required to design them. This process becomes increasingly difficult and uncertain as the data volume increases.

The proposed GCNN is learned and trained directly on the basis of the input variables and connection modes. The convolution kernels used are essentially feature extractors, and their parameters are obtained by the learning and training of labeled examples. The entire process belongs to an end-to-end solution. Another advantage of the proposed method is that its processing unit is a single building rather than the entire group, and the extracted features are local, which is more consistent with human cognition.

5.2. Remaining issues

One of the challenges in constructing a machine learning system is that there are many aspects to be considered. Here, we highlight three considerations for perfecting the GCNN model. The first is the pooling technology, which not only reduces the complexity of the model as the number of parameters decreases but also benefits the extraction of features at different scales. Hence, how to combine the multi-scale representation of geographical data with pooling remains an interesting and promising research topic. The second consideration is polynomial approximation of the convolution kernel. The present study tested the sensitivity of the model to the polynomial order. However, the polynomial bases have scope for further discussion and investigation, e.g., employing orthogonal Chebyshev polynomials as bases instead of non-orthogonal monomials. The third consideration is the network structure. Only a feedforward neural network was used to construct the learning model. However, there are many sophisticated network

structures that may help to intensify the convergence and improve the performance, such as DenseNet (Huang et al., 2017) and PEN Net (Sidiike et al., 2018), or extend the GCNN to other tasks, such as FCNs (Long et al., 2015).

In addition, the present study only discussed two building perceptual patterns: regular and irregular. However, building groups can also distinguish more detailed patterns visually or classify them into new pattern types in other application scenarios (such as urban development analysis). Improving the GCNN such that it has the ability to classify other building pattern types requires further investigation. A large number of high-quality training examples are required. Volunteered geographic information (VGI) may be a potential source of data for such a task, although some of the data are not well labeled and there are some quality issues to be addressed.

Finally, it should be noted that as with CNNs, the GCNN also faces difficulty in explaining the process of feature extraction; in fact, the difficulty is greater in the GCNN case. On the one hand, the color variable of an image can be visually shaped; thus, visualization can be used to interpret and understand this process (Zeiler and Fergus, 2014). However, the orientation variable or shape variable is more abstruse and elusive. On the other hand, the graph convolution is manipulated in the Fourier domain, and characteristics of the Fourier-transformed values, i.e. increasing stability and sparsity of their distributions as shown in Fig. 10, were observed in this experiment. However, these characteristics are difficult to understand intuitively after re-conversion to the vertex domain; to be more specific, it is difficult to discern some typical learned structures on graphs, which is comparable to the edges or higher-level structures in images.

6. Conclusions and outlook

Previously, spatial vector data were not sufficient for performing patterns analysis and knowledge mining when using powerful convolutional networks because in most cases, the data can only be modeled as a graph structure that does not satisfy the requirements for regularity. To overcome this problem, the present study introduced the graph convolutional neural network (GCNN), whereby the convolution operation is converted from the vertex domain into a point-wise product in the Fourier domain. Building perceptual pattern classification

was considered as a case study, and experiments on large building datasets confirmed that the proposed method produces satisfactory results in identifying regular and irregular patterns, thereby achieving a significant improvement over existing methods.

This study highlighted the potential of the GCNN to analyze graph-structured spatial data. In the future, additional efforts should be devoted toward improving the GCNN with deep learning technologies and applying the GCNN to other analysis tasks, e.g., road network pattern recognition and point cloud identification.

Acknowledgments

The authors thank the editors and anonymous reviewers for their insightful comments and constructive suggestions. This study was supported by the National Key Research and Development Program of China under grant 2017YFB0503500, and the National Natural Science Foundation of China under grant 41531180.

References

- Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Deng, L., Penn, G., Yu, D., 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio, Speech, Language Process.* 22 (10), 1533–1545.
- Ai, T., Guo, R., 2007. Polygon cluster pattern mining based on Gestalt principles. *Acta Geod. Cartogr. Sin.* 36 (3), 302–308.
- Anders, K.H., 2006. Grid typification. In: Riedl, A., Kainz, W., Elmes, G.A. (Eds.), *Progress in Spatial Data Handling*. Springer, Berlin, Heidelberg, pp. 633–642.
- Angel, S., Parent, J., Civco, D.L., 2010. Ten compactness properties of circles: measuring shape in geography. *Can. Geogr./Le Géographe Can.* 54 (4), 441–461.
- Anselin, L., 1995. Local indicators of spatial association—LISA. *Geogr. Anal.* 27 (2), 93–115.
- Anwer, R.M., Khan, F.S., van de Weijer, J., Molinier, M., Laaksonen, J., 2018. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. *ISPRS J. Photogramm. Remote Sens.* 138, 74–85.
- Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S., 1991. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 209–216.
- Basaraner, M., Cetinkaya, S., 2017. Performance of shape indices and classification schemes for characterising perceptual shape complexity of building footprints in GIS. *Int. J. Geogr. Inf. Sci.* 31 (10), 1952–1977.
- Batty, M., 2008. The size, scale, and shape of cities. *Science* 319 (5864), 769–771.
- Batty, M., Longley, P.A., 1994. *Fractal Cities: A Geometry of Form and Function*. Academic Press, London.
- Beck, A., Long, G., Boyd, D.S., Rosser, J.F., Morley, J., Duffield, R., Sanderson, M., Robinson, D., 2018. Automated classification metrics for energy modelling of residential buildings in the UK with open algorithms. *Environ. Plan. B: Urban Anal. City Sci.*
- Bertin, J., 1983. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison.
- Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process. Mag.* 34 (4), 18–42.
- Bruna, J., Zaremba, W., Szlam, A., Lecun, Y., 2014. Spectral networks and locally connected networks on graphs. In: *Proceedings of International Conference on Learning Representations (ICLR)*.
- Burghardt, D., Steiniger, S., 2005. Usage of principal component analysis in the process of automated generalisation. In: *Proceedings of the 22th International Cartographic Conference (ICC)*, pp. 9–16.
- Cetinkaya, S., Basaraner, M., Burghardt, D., 2015. Proximity-based grouping of buildings in urban blocks: a comparison of four algorithms. *Geocarto Int.* 30 (6), 618–632.
- Christophe, S., Ruas, A., 2002. Detecting building alignments for generalization purposes. In: Richardson, D.E., van Oosterom, P. (Eds.), *Advances in Spatial Data Handling*. Springer, Berlin, pp. 419–432.
- Chung, F.R.K., 1997. *Spectral Graph Theory*. American Mathematical Society, Providence, Rhode Island.
- Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3844–3852.
- Deng, M., Tang, J., Liu, Q., Wu, F., 2018. Recognizing building groups for generalization: a comparative study. *Cartogr. Geogr. Inf. Sci.* 45 (3), 187–204.
- Dhillon, I.S., Guan, Y., Kulis, B., 2007. Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (11), 1944–1957.
- Ding, P., Zhang, Y., Deng, W.J., Jia, P., Kuijper, A., 2018. A light and faster regional convolutional neural network for object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* 141, 208–218.
- Du, S., Luo, L., Cao, K., 2016a. Extracting building patterns with multilevel graph partition and building grouping. *ISPRS J. Photogramm. Remote Sens.* 122, 81–96.
- Du, S., Shu, M., Feng, C., 2016b. Representation and discovery of building patterns: a three-level relational approach. *Int. J. Geogr. Inf. Sci.* 30 (6), 1161–1186.
- Duchêne, C., Bard, S., Barillot, X., Ruas, A., Trévisan, J., Holzapfel, F., 2003. Quantitative and qualitative description of building orientation. In: *Proceedings of the 7th ICA Workshop on Progress in Automated Map Generalisation*.
- Duvenaud, D.K., Maclaurin, D., Iparragirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P., 2015. Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2224–2232.
- Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1263–1272.
- Gong, M., Yang, H., Zhang, P., 2017. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. *ISPRS J. Photogramm. Remote Sens.* 129, 212–225.
- Gong, X., Wu, F., 2018. A typification method for linear pattern in urban building generalization. *Geocarto Int.* 33 (2), 189–207.
- Gonzalez-Abraham, C.E., Radloff, V.C., Hammer, R.B., Hawbaker, T.J., Stewart, S.I., Clayton, M.K., 2007. Building patterns and landscape fragmentation in northern Wisconsin, USA. *Landscape Ecol.* 22 (2), 217–230.
- Hammond, D.K., Vandergheynst, P., Gribonval, R., 2011. Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.* 30 (2), 129–150.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
- He, X., Zhang, X., Xin, Q., 2018. Recognition of building group patterns in topographic maps based on graph partitioning and random forest. *ISPRS J. Photogramm. Remote Sens.* 136, 26–40.
- Hecht, R., Meinel, G., Buchroithner, M., 2015. Automatic identification of building types based on topographic databases – a comparison of different data sources. *Int. J. Cartogr.* 1 (1), 18–31.
- Henaff, M., Bruna, J., LeCun, Y., 2015. Deep Convolutional Networks on Graph-Structured Data. Available from: arXiv preprint arXiv:1506.05163.
- Henn, A., Römer, C., Gröger, G., Plömer, L., 2012. Automatic classification of building types in 3D city models. *GeoInformatica* 16, 281–306.
- Huang, B., Zhao, B., Song, Y., 2018. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Rem. Sens. Environ.* 214, 73–86.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708.
- Huang, H., Kieler, B., Sester, M., 2013. Urban building usage labeling by geometric and context analyses of the footprint data. In: *Proceedings of the 26th International Cartographic Conference (ICC)*.
- Jain, A., Zamir, A.R., Savarese, S., Saxena, A., 2016. Structural-RNN: deep learning on spatio-temporal graphs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5308–5317.
- Jochem, W.C., Bird, T.J., Tatem, A.J., 2018. Identifying residential neighbourhood types from settlement points in a machine learning approach. *Comput. Environ. Urban Syst.* 69, 104–113.
- Kim, Y., 2014. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751.
- Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: *Proceedings of International Conference on Learning Representations (ICLR)*.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105.
- Ktena, S.I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glover, B., Rueckert, D., 2018. Metric learning with spectral graph convolutions on brain connectivity networks. *NeuroImage* 169, 431–442.
- Kumar, B., Pandey, G., Lohani, B., Misra, S.C., 2019. A multi-faceted CNN architecture for automatic classification of mobile LiDAR data and an algorithm to reproduce point cloud samples for enhanced training. *ISPRS J. Photogramm. Remote Sens.* 147, 80–89.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86 (11), 2278–2324.
- Li, W., Goodchild, M.F., Church, R., 2013. An efficient measure of compactness for two-dimensional shapes and its application in regionalization problems. *Int. J. Geogr. Inf. Sci.* 27 (6), 1227–1250.
- Li, Y., Zemel, R., Brockschmidt, M., Tarlow, D., 2016. Gated graph sequence neural networks. *Proceedings of International Conference on Learning Representations (ICLR)*.
- Li, Z., Yan, H., Ai, T., Chen, J., 2004. Automated building generalization based on urban morphology and Gestalt theory. *Int. J. Geogr. Inf. Sci.* 18 (5), 513–534.
- Lombardo, U., 2014. Quantitative morphometric analysis of lakes using GIS: rectangularity R, ellipticity E, orientation O, and the rectangularity vs. ellipticity index, REi. *Cartogr. Geogr. Inf. Sci.* 41 (4), 340–347.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.
- Lüscher, P., Weibel, R., Burghardt, D., 2009. Integrating ontological modelling and Bayesian inference for pattern classification in topographic vector data. *Comput. Environ. Urban Syst.* 33, 363–374.
- Masci, J., Boscaigni, D., Bronstein, M.M., Vandergheynst, P., 2015. Geodesic convolutional neural networks on riemannian manifolds. In: *Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 832–840.
- Meinel, G., Hecht, R., Herold, H., 2009. Analyzing building stock using topographic maps and GIS. *Build. Res. Inf.* 37 (5–6), 468–482.

- Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5425–5434.
- Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML), pp. 2014–2023.
- Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., 2018. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* 145, 120–147.
- Peura, M., Iivarinen, J., 1997. Efficiency of simple shape descriptors. In: Proceedings of the Third International Workshop on Visual Form, pp. 443–451.
- Pilehrooosha, P., Karimi, M., 2018. An integrated framework for linear pattern extraction in the building group generalization process. *Geocarto Int.* 1–22.
- Rahimi, A., Cohn, T., Baldwin, T., 2018. Semi-supervised user geolocation via graph convolutional networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 2009–2019.
- Sandryhaila, A., Mouraj, J.M.F., 2013. Discrete signal processing on graphs. *IEEE Trans. Signal Process.* 61 (7), 1644–1656.
- Sandryhaila, A., Mouraj, J.M.F., 2014. Discrete signal processing on graphs: frequency analysis. *IEEE Trans. Signal Process.* 62 (12), 3042–3054.
- Shuman, D.I., Ricaud, B., Vandergheynst, P., 2016. Vertex-frequency analysis on graphs. *Appl. Comput. Harmon. Anal.* 40 (2), 260–291.
- Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* 30 (3), 83–98.
- Sidike, P., Asari, V.K., Sagan, V., 2018. Progressively Expanded Neural Network (PEN Net) for hyperspectral image classification: a new neural network paradigm for remote sensing image analysis. *ISPRS J. Photogramm. Remote Sens.* 146, 161–181.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., 2017. Mastering the game of Go without human knowledge. *Nature* 550 (7676), 354–359.
- Steiniger, S., Burghardt, D., Lange, T., Weibel, R., 2008. An approach for the classification of urban building structures based on discriminant analysis techniques. *Trans. GIS* 12 (1), 31–59.
- Such, F.P., Sah, S., Dominguez, M.A., Pillai, S., Zhang, C., Michael, A., Cahill, N.D., Ptucha, R., 2017. Robust spatial filtering with graph convolutional neural networks. *IEEE J. Sel. Top. Signal Process.* 11 (6), 884–896.
- Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region. *Econ. Geogr.* 46, 234–240.
- Vanderhaegen, S., Canters, F., 2017. Mapping urban form and function at city block level using spatial metrics. *Landscape Urban Plan.* 167, 399–409.
- Wang, P., Gan, Y., Shui, P., Yu, F., Zhang, Y., Chen, S., Sun, Z., 2018. 3d shape segmentation via shape fully convolutional networks. *Comput. Graph.* 70, 128–139.
- Wang, W., Du, S., Guo, Z., Luo, L., 2015. Polygonal clustering analysis using multilevel graph-partition. *Trans. GIS* 19 (5), 716–736.
- Wei, Z., Guo, Q., Wang, L., Yan, F., 2018. On the spatial distribution of buildings for map generalization. *Cartography Geogr. Inf. Sci.* 45 (6), 539–555.
- Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., Habib, A., 2019. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* 147, 178–192.
- Wu, B., Yu, B., Wu, Q., Chen, Z., Yao, S., Huang, Y., Wu, J., 2018. An extended minimum spanning tree method for characterizing local urban patterns. *Int. J. Geogr. Inform. Sci.* 32 (3), 450–475.
- Wurm, M., Schmitt, A., Taubenböck, H., 2016. Building types classification using shape-based features and linear discriminant functions. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 9 (5), 1901–1912.
- Yan, H., Weibel, R., Yang, B., 2008. A multi-parameter approach to automated building grouping and generalization. *Geoinformatica* 12, 73–89.
- Yan, X., Ai, T., 2018. Analysis of irregular spatial data with machine learning: classification of building patterns with a graph convolutional neural network. In: Winter, S., Griffin, A., Sester, M. (Eds.), 10th International Conference on Geographic Information Science (GIScience 2018). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. Dagstuhl, pp. 69:1–69:7.
- Yan, X., Ai, T., Zhang, X., 2017. Template matching and simplification method for building features based on shape cognition. *ISPRS Int. J. Geo-Inf.* 6 (8), 250.
- Yang, W., 2008. Identify building patterns. In: Chen, J., Jiang, J., Kainz, W. (Eds.), The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 391–398.
- Yousefhussien, M., Kelbe, D.J., Ientilucci, E.J., Salvaggio, C., 2018. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS J. Photogramm. Remote Sens.* 143, 191–204.
- Yu, B., Yin, H., Zhu, Z., 2018. Spatial-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), pp. 3634–3640.
- Yu, W., Ai, T., Liu, P., Cheng, X., 2017. The analysis and measurement of building patterns using texture co-occurrence matrices. *Int. J. Geogr. Inf. Sci.* 31 (6), 1079–1100.
- Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), European Conference on Computer Vision. Springer, Cham, pp. 818–833.
- Zhang, L., Deng, H., Chen, D., Wang, Z., 2013. A spatial cognition-based urban building clustering approach and its applications. *Int. J. Geogr. Inf. Sci.* 27 (4), 721–740.
- Zhang, X., Ai, T., Stoter, J., 2008. The evaluation of spatial distribution density in map generalization. In: The XXI Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS 2008), pp. 181–188.
- Zhang, X., Ai, T., Stoter, J., Kraak, M., Molenaar, M., 2013. Building pattern recognition in topographic data: examples on collinear and curvilinear alignments. *Geoinformatica* 17, 1–33.
- Zhu, D., Liu, Y., 2018. Modelling spatial patterns using graph convolutional networks. In: Winter, S., Griffin, A., Sester, M. (Eds.), 10th International Conference on Geographic Information Science (GIScience 2018). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. Dagstuhl, Germany, pp. 73:1–73: 7.
- Zhu, X., Michael, R., 2012. Approximating Signals Supported on Graphs. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 3921–3924.