Boundary-Aware Shape Recognition Using Dynamic Graph Convolutional Networks

Jinming Zhao[a], Junyu Dong[a], Huiyu Zhou[b], Xinghui Dong[a,*]

[a] *State Key Laboratory of Physical Oceanography and the Faculty of Information Science and Engineering, Ocean University of China, Qingdao, 266100, 238 Songling Road, Qingdao, 266100, Shandong, China*
[b] *School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, U.K.*

Abstract

Shape recognition, which often involves topology in mathematics, is a fundamental subfield of image recognition. Although deep learning techniques have been widely applied to image recognition and have achieved great success, this is not the case for 2D shape recognition. Inspired by the powerful spatial representation ability of Graph Convolutional Networks (GCNs), we leverage this technique to address the shape recognition problem. To this end, we propose a Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN). To be specific, we first extract the maximum boundary of the object depicted in an image and sample this boundary into a set of key points. Given a key point, a set of features is then extracted as its representation. Furthermore, we construct a series of graphs from the key points and use the BASR-GCN to learn the spatial layout of these points. In addition, we introduce a multi-scale BASR-GCN (BASR-GCN-MS) in order to exploit the shape features extracted at different scales. To our knowledge, GCNs have not been applied to 2D shape recognition before. The proposed method is tested using four publicly available shape data sets. Experimental results show that our method outperforms the baselines. We believe that these promising results should be due to the fact that the BASR-GCN captures the spatial layout and semantic information of the shape fulfilled by graph convolutions.[1]

Keywords: Shape Representation, Shape Recognition, Boundary, Skeleton, Graph Convolutional Networks.

[*] Corresponding author

Email addresses: zhaojinming@stu.ouc.edu.cn (Jinming Zhao), dongjunyu@ouc.edu.cn (Junyu Dong), hz143@leicester.ac.uk (Huiyu Zhou), xinghui.dong@ouc.edu.cn (Xinghui Dong)

[1] The models and code will be published on the acceptance of the paper.

1. Introduction

Shape, normally manifested in the form of a boundary, silhouette or skeleton, serves as a crucial cue for object recognition within an image [1, 2, 3, 4, 5]. Despite the absence of other characteristics, such as color/intensity, brightness and texture, objects can still be identified by humans via the shape cue. This phenomenon implies that shape is insensitive to variations in lighting, color and texture [6]. Therefore, shape can be used as a robust characteristic for shape retrieval and object recognition. In the literature, shape recognition has been widely applied to many fields, including medical image analysis[7], plant leaf identification [8], motion detection [9] and shape modeling [10].

However, the challenges to shape recognition mainly stem from the diversity changes induced by deformations and occlusions. As a result, the research of shape recognition has been focused on building reliable shape descriptors with sufficient discriminative power [11, 12]. Ideally, these descriptors should be capable of capturing the shape characteristics of a category under deformations and twists, while maintaining the distinctiveness across different categories.

Traditional shape recognition methods are normally built on top of low-level shape descriptors for the sake of addressing the challenges. These descriptors can be divided into three categories: region-based [13, 14, 15], boundary-based [16, 12] and skeleton-based [17]. The region-based shape descriptors, e.g., Zernike moments [13], are computed across the entire region of a shape, which are normally robust against shape deformations and occlusions. Nevertheless, they are susceptible to noise and struggle with capturing the intricate internal structure of shapes. The boundary-based shape descriptors, for example, Bag of Contour Fragments (BoCF) [16], are normally designed based on the spatial distribution of the boundary of an object. Thus, they are relatively stable to affine transformations. Since those descriptors overlook the internal information of the shape, they are sensitive to the non-rigid deformation and articulation. Although the skeleton-based shape descriptors, e.g., Bag of Skeleton-Related Contour Parts (BoSCP) [17], are able to encode the topological structure, geometric information and width variation of an object, they are sensitive to noise and deformation.

On the other hand, deep learning techniques have also been applied to shape recognition [18, 19]. They are normally developed on top of the boundary or silhouette images rather than the pure boundary, silhouette or skeleton data. In this case, the shape cue is not explicitly utilized. Thus, the shape characteristics may not be adequately exploited and the background may also impair the accuracy of shape recognition. Since Graph Convolutional Networks (GCNs) directly operate on non-Euclidean graph structures, they are able to capture the topological relationships and long-range dependencies between the nodes of a graph. In contrast, Convolutional Neural Networks (CNNs) rely on regular grid convolutions and are often biased toward texture,

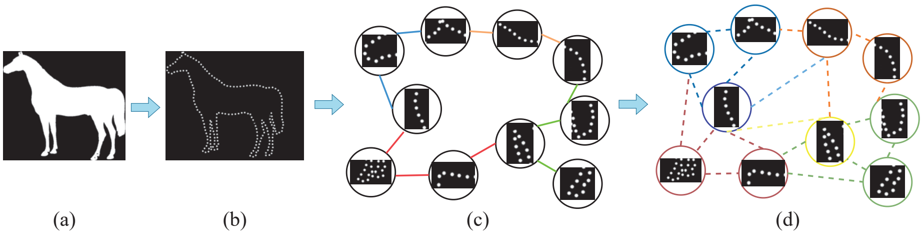


Figure 1: The illustration of the operation of the proposed BASR-GCN. Here, (a) and (b) show the silhouette of an object and the key points sampled from the maximum boundary of it, respectively. In (c), the initial graph is built by extracting features from these key points, in which the solid line denotes the spatial connection between two neighboring nodes and different colors suggest that the characteristics of neighboring nodes are aggregated from different parts of the boundary. The result obtained after applying the graph convolution to the graph is presented in (d), where the aggregated key points contain both the spatial and the semantic information. The dashed line indicates that the selection of neighboring nodes is influenced by the semantic information.

which makes them less effective at modeling the geometric continuity and global structure. Therefore, GCNs are better suited for shape recognition than CNNs.

To better exploit the shape characteristics, we are motivated to introduce a Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN) by explicitly exploiting the boundary cue. Specifically, the maximum boundary of the silhouette (see Fig. 1(a)) of an object is first extracted. According to the contour integration mechanism used by the Human Vision System (HVS), humans can still perceive the shape of a contour even if it has been discretized into a set of fragments [20]. Therefore, we discretize the boundary into a set of equidistant key points (see Fig. 1(b)) in order to reduce the computational cost. A set of graphs (see Fig. 1(c)) is further constructed on top of the key points. The BASR-GCN learns the spatial layout of keypoints through dynamic graph learning (see Fig. 1(d)). For the purpose of dynamically selecting neighboring nodes, pairwise distances between keypoints at each convolutional layer are computed in the feature space. As a result, the graph structure can be built adaptively. To further leverage the shape features extracted at different scales, we also propose a multi-scale BASR-GCN (BASR-GCN-MS), in which a shape is represented by the keypoints sampled at multiple scales. In terms of each scale, feature representations are learned individually. All the features are concatenated into a single feature vector, which encodes both fine and coarse shape characteristics.

To the best of our knowledge, GCNs have not been used in 2D shape recognition before. Our contributions can be summarized as threefold.

- We make the first effort on applying graph convolutions to 2D shape recognition.
- We propose a Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN) which integrates both the boundary and skeleton cues, en-

abling the effective learning of both the spatial layout of a shape and the semantic relationships between its different parts.

- To extract shape features at different scales, we also introduce a multi-scale BASR-GCN, which is able to derive the complementary information across scales.

The remainder of this paper will be organized as follows. We review the related work in Section 2. Our approach is introduced in Section 3. We describe the experimental setup in Section 4. The experimental results and ablation study are reported in Sections 5 and 6, respectively. Finally, we draw our conclusion in Section 7.

2. Related Work

2.1. Shape Recognition

Traditionally, shape recognition methods were designed based on hand-crafted shape descriptors. These descriptors can be categorized into three classes: region-based, boundary-based and skeleton-based. Region-based shape descriptors are normally developed by representing the shape of an object based on the silhouette information. Zernike moments [13] leveraged a set of complex orthogonal polynomials to derive the compact representation of shape geometries. Despite Zernike moments were robust to deformations and occlusions, they were sensitive to noise. The Angular Radial Transformation (ART) method [14] represented a shape by measuring the distribution of intensity values in the polar coordinates. This method may be affected when it deals with complex or noisy shapes even though it is independent of rotation and scaling. In [15], a convex hull was defined as the smallest convex polygon which enclosed a set of points. Although convex hull simplifies complex shapes and identifies essential features, it cannot capture the internal details and structure well.

Boundary-based shape descriptors were designed on top of the boundary of an object. The shape context descriptor [12] encoded the relative spatial distribution of the points sampled on the boundary. Inspired by the Bag of Features [21] methods, Wang et al. [16] proposed the Bag of Contour Fragments (BoCF) descriptor, which used Locally-Lonstrained Linear Coding (LLC) [22] to encode the features of contour fragments. Ribas and Bruno [1] modeled shape boundaries as a directed complex network, which extracted topological features, and leveraged randomized neural networks to efficiently learn discriminative representations. To extract features from various perspectives, Blandon et al. [2] introduced a framework that combined the contour information with a multi-view learning strategy, which enhanced the ability of the model to recognize and classify complex shapes. In contrast, Giveki et al. [23] designed a shape descriptor that captured diverse features from boundary pixels. A multi-view learning strategy was also used to improve classification accuracy across different perspectives.

Although the boundary-based shape descriptors are insensitive to affine transformations, they are sensitive to the non-rigid deformation because they do not take into consideration the internal characteristics of the shape. Since the skeleton is able to preserve the geometric information and topological structure of a shape, shape descriptors have also been introduced based on this cue. Shen et al. [17] proposed the Bag of Skeleton Paths (BoSP) descriptor, which exploited both the boundary and skeleton cues. However, skeleton-based descriptors still struggle with noise, angle transformations and significant non-rigid deformations.

In the past decade, deep learning techniques have been widely used in computer vision [24, 25, 26, 27, 28]. Motivated by the success achieved in image recognition and segmentation, Convolutional Neural Networks (CNNs) were applied to shape recognition, including the Shape Boltzmann Machine approach [18] and Shape Classification Network (SCN) [19]. However, these methods normally use silhouette or boundary images as the input rather than the pure silhouette, boundary or skeleton data. In other words, the shape information is not explicitly exploited. In this situation, the shape characteristics may not be sufficiently utilized while the background may impair the discriminatory power of the model trained. Recently, Hossain et al. [29] developed a framework, referred to as Invariant Shape Representation Learning (ISRL), to enhance the robustness of image classifiers. Nevertheless, this framework used images rather than the shape data.

In contrast, we explicitly utilize the boundary cue for shape representation, while the skeleton cue is also used as an additional characteristic.

2.2. Graph Convolutional Networks

Graph Convolutional Networks (GCNs) have been extensively utilized in different fields, including the point cloud data [30], social networks [31] and recommendation engines [32]. Micheli [33] proposed an early form of spatial GCNs with composite non-recursive layers. Spectral GCNs [34] used the spectrum of the graph Laplacian to represent graphs. In [34], Kipf et al. introduced a semi-supervised classification method based on GCNs. In particular, a GCN layer propagation rule was designed through a local first-order approximation of spectral graph convolution, which realized the linear scalability to the large-scale graph data. Inductive representation learning was employed on large-scale graphs in the GraphSAGE method [35]. This method efficiently generated embeddings for unseen data using node feature information and optimized the full graph sampling to the partial neighbor sampling centered around the node.

Traditional GCNs typically operate on a static adjacency matrix, which is built based on inter-node connections and cannot be unaltered thereafter. In this situation, the adjacency matrix has to be rebuilt once new nodes are introduced. To address this issue, Wang et al. [30] introduced a dynamic model which updated the graph structure between layers. Within each round of update, a new set of K nearest points were determined using K-Nearest Neighbors (KNN). To address the limitations of the existing methods which relied on graph convolution or its approximations, the Graph Attention Network (GAT) [36] employed a masked self-attention layer. Nodes were able to focus on the features within their vicinity by stacking layers. Since GAT implicitly assigns varying weights to different nodes in the neighborhood without requiring costly matrix operations or the prior knowledge of the graph structure, it is suitable for both the inductive and transductive problems.

Within the proposed BASR-GCN, a set of graphs are built from the key points sampled from a boundary. GCNs are used to capture the spatial layout of the boundary. Considering the key point data is similar to the point cloud data that the EdgeConv [30] method processed, the BASR-GCN is built on top of the convolutional layer of the EdgeConv. In addition, we introduce a multi-scale BASR-GCN which is able to learn the spatial layout information of the shape at different scales. To our knowledge, GCNs have not been used in 2D shape recognition before.

3. The Boundary-Aware Shape Recognition Graph Convolutional Network

For the sake of explicitly exploiting the boundary cue, we propose a Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN). Specifically, the maximum boundary of an object is first extracted. The boundary is discretized into a series of key points. Regarding each key point, both the x and y coordinates are used as its location features. The angle between this point and its preceding key point is also calculated, which is utilized as the angle feature of the point. The nearest skeleton point to each key point is then located and the x and y coordinates of this point are employed as two additional features of the related key point. A set of graphs are further built from the key points. The BASR-GCN is used to learn the spatial layout of these points. To exploit the shape features extracted at different scales, we also design a multi-scale BASR-GCN (BASR-GCN-MS). Figures 2 and 3 exhibit the architectures of the proposed BASR-GCN and BASR-GCN-MS, respectively.

3.1. Overall Network Architecture

As illustrated in Fig. 2, we first extract the maximum boundary of the silhouette of an object. A set of key points are derived which can be used to represent the shape approximately. In terms of each key point, the nearest skeleton point is located. Feature extraction is then performed in order to obtain a compact representation of each key point. The key points are then fed into the Stem block, which consists of three one-dimensional convolutional layers and two GELU activation layers. This process

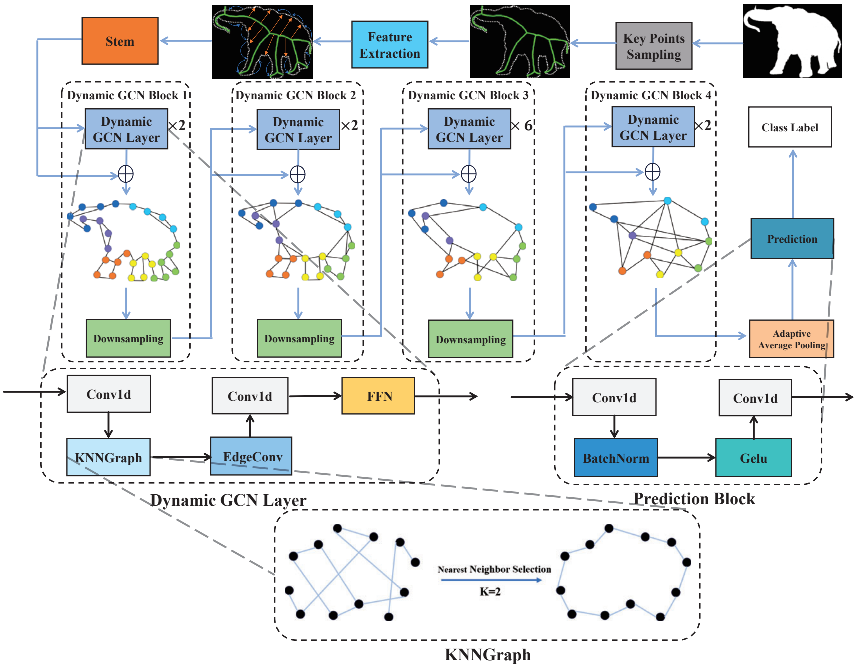


Figure 2: The architecture of the proposed Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN).

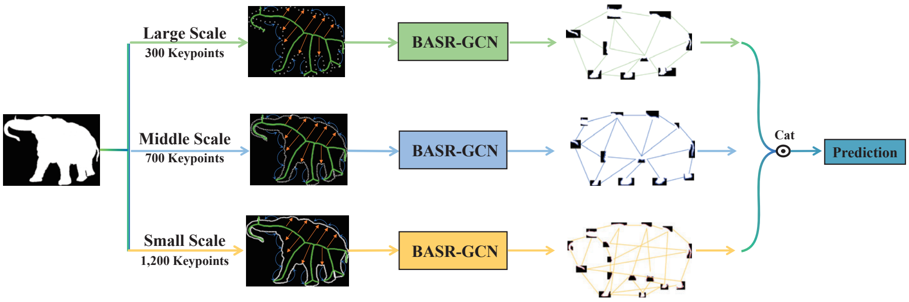


Figure 3: The illustration of the proposed Multi-scale Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN-MS).

introduces the diversity of features and adjusts the dimensionality of the features to the number of channels of the features sent to the first GCN block. Furthermore, a series of graphs are constructed on top of the key points. To capture both the spatial layout and the semantic information of the shape, GCNs are applied to these graphs.

Inspired by the structure of CNNs, the proposed BASR-GCN is designed as a pyramidal architecture. This network contains four consecutive GCN blocks. Within each block, a series of GCN layers are contained and the number of the channels of features and the number of nodes are set to the same for different layers. The number of nodes reduces from a block to the deeper block because the downsampling operation is applied to the output of the prior block.

As the depth of the block increases, each node aggregates the information of the adjacent nodes contained in the prior block. Consequently, selection of adjacent nodes inclines towards encoding the semantic information of the shape. In other words, the nodes from the other parts of the shape are selected as neighbors, which establishes the connection between two different parts.

The node-level features learned by the GCN are fed into the adaptive average pooling module. As a result, the graph-level embedding features are generated, which integrate the topological spatial layout of the previously learned shape and the semantic connections in the feature space. Finally, a set of key features which determine the shape of an object are learned.

Besides, the structure of the proposed Multi-scale BASR-GCN (BASR-GCN-MS) is illustrated in Fig. 3. As can be seen, the BASR-GCN is applied to three sets of key points sampled at the small, middle and large scales, separately. The small-scale, middle-scale and large-scale features learned at the three branches are fused and sent to the prediction layer for shape recognition.

3.2. Key Point Sampling

The boundary of a shape, which embodies the global characteristics of the shape and is insensitive to affine transformations, has been successfully applied to shape recognition [16, 17]. Inspired by these studies, we first extracted the maximum boundary of the shape of an object. Specifically, we adopted the method that Wang et al. [16] proposed, which employed a contour tracing strategy based on the isocontour analysis to extract the outer boundary of binary shapes. The process began with cropping and filling operations to remove background regions near the image edges and to reduce the influence of internal holes when identifying the maximum boundary.

The binary image was then treated as a scalar field, from which all isocontours corresponding to a fixed threshold slightly below 1 (e.g., 0.8) were extracted. This threshold captured the transition between the foreground and background regions. Among the resulting closed contours, the contour with the longest arc length was selected as the primary boundary of the shape. This selection helped to eliminate small internal structures, noise-induced fragments, and holes, thus focusing the analysis on the overall contour of the object.

To mitigate the influence of redundant or overly dense points, the contour was simplified and orientation standardized. The uniform arc-length interpolation was further used to resample the contour into a fixed number of points, ensuring consistency and comparability in the subsequent feature extraction process. These points were referred to as key points.

3.3. Feature Extraction

With regard to each key point, the x and y coordinates are used as its location features. We also calculate the arctangent value of the angle between each key point and its preceding key point. This value is utilized as the angle feature of the key point. Considering that the features extracted on the boundary are incapable of adequately representing the internal characteristics of the shape, such as thickness and curvature, we first extract the shape skeleton using the Adaptive Linear Span Network (Ada-LSN) that Liu et al. [37] proposed and sample a set of skeleton points. The Ada-LSN automatically fuses multi-scale deep features using a Neural Architecture Search (NAS)–driven pyramid structure, namely, Linear Span Pyramid (LSP), by stacking multiple Linear Span Units (LSUs) across different feature layers. Then the nearest skeleton point is located by means of the nearest-neighbor search for each key point. Both the x and y coordinates of this point are employed as two additional features of the associated key point.

After feature extraction is complete, each key point is represented by a five-dimensional feature vector. This vector is fed into the Stem block, which consists of three consecutive one-dimensional convolutional layers. The output of the Stem block will be sent to the first GCN block.

Multi-scale image representation has been extensively applied and normally produces the better result, compared to the single-scale representation. To learn the features of a shape at different scales, we sample three sets of key points at different scales and apply the proposed BASR-GCN to each set individually. Correspondingly, three sets of features are generated, which represent the characteristics of the shape at the small, middle and large scales and are denoted as F_{Small}, F_{Mid} and F_{Large}, respectively. The three sets of features are further fused by concatenating as:

$$F_{MS} = Cat(F_{Small}, F_{Mid}, F_{Large}). \tag{1}$$

In essence, F_{MS} encodes the multi-scale characteristics of the shape and normally owns the stronger discriminatory power than the features extracted at a single scale.

3.4. Dynamic GCN Block

The static graph convolution is normally used in the existing graph convolution studies [34]. During the convolution computation, the adjacency matrix of the graph does not change. Although this design can save the computational cost and speed up the computation, the original spatial adjacency relationship may not be enough for shape recognition when the complex intra-class and inter-class changes occur, due to the high similarity of the spatial distributions of different objects. To address this issue, we use the dynamic graph convolution instead, which takes into account both the spatial layout of the shape and the semantic relationship between different parts by continuously updating the adjacency matrix and downsampling the nodes.

The proposed BASR-GCN contains four dynamic GCN blocks, which comprise 2, 2, 6 and 2 dynamic GCN layers, respectively. Within each block, the input is first passed through different GCN layers. Then the output is processed by a downsampling operation, in which the number of nodes is halved and the number of feature channels is doubled, before they are sent to the next block. Specifically, the number of feature channels is set to 80, 160, 320 and 640 in terms of the four blocks, respectively.

3.4.1. Dynamic GCN Layer

The dynamic GCN layer is adopted based on the graph convolutional layer that Han et al. [38] proposed. In contrast to the original design, the adopted version has three differences. First, we select a fixed number of node neighbors rather than increasing the number of neighbors with the progress of the block because either excessive or insufficient neighbors will impair the extraction of the semantic information of the shape from the nodes. Second, we remove the position encoding component that Han et al. [38] used to represent the positional adjacency relationship between nodes but employ both the x and y coordinates of the key point instead. Third, we replace the two-dimensional convolution used in [38] by the one-dimensional convolution in order to process the key points.

Given a dynamic GCN layer, a one-dimensional convolutional layer is first used to increase the diversity of features. Inspired by the existing work [39], we then calculate the difference between the features of a node v_i and the features of the other nodes and select K nearest neighbors $N(v_i)$ for the node v_i according to

$$N(v_i) = KNN_k(v_i, V), \forall v_i \in V. \tag{2}$$

Subsequently, we add an edge e_{ij} from v_i to every neighboring node v_j contained in $N(v_i)$, which can be expressed as

$$e_{ij} = \{v_i, v_j\}, \forall v_j \in N(v_i). \tag{3}$$

The result is a sparse graph $G_d = (V, E)$, where $E = \{e_{ij}\}$.

The current dynamic GCN block updates the selection of neighboring nodes using the KNN algorithm at each dynamic GCN layer. Thereby, the structure of the graph is updated. In this case, the information exchange between different nodes gradually varies from the nodes at the closest distance in the *Euclidean* space to the nodes with the most similar features in the semantic space.

Furthermore, the graph convolution operation [38] is performed on the graph constructed. The key to graph convolution operations is the aggregation and update of the information of adjacent nodes, which can be formulated as

$$G' = F(G, W) = Update(Aggregate(G, W_{agg}), W_{update}), \tag{4}$$

where G' is the graph after the graph convolution operations have been conducted, $F(\cdot)$ is a set of graph convolution operations, G is the graph before the graph convolution operations are performed, W is the weighting coefficient, $Update(\cdot)$ and $Aggregate(\cdot)$ denote the aggregation and update operations, respectively, and W_{agg} and W_{update} stand for the weights used for the aggregation and update operations.

Given a node, the features of the neighboring nodes are aggregated in order to compute the representation for the next layer. The update operation merges and updates the features aggregated. These operations can be expressed as

$$x_i' = u(x_i, a(x_i, K(x_i), W_{agg}), W_{update}), \tag{5}$$

where x_i' denotes the node features derived after the aggregation and update operations have been performed, x_i represents node features before these operations are conducted, $K(x_i)$ stands for the neighboring nodes and $a(\cdot)$ is the aggregation operation.

To fulfil the above operations, we adopt the EdgeConv [30], which was originally introduced for the point cloud data, because it can effectively extract local shape features of the point cloud data and maintain the permutation invariance of these data. Within the scenario of EdgeConv, the aggregation and update operations can be defined as

$$x_a = a(x_i) = concat[x_i, (x_j - x_i)], \forall x_j \in K(x_i), \tag{6}$$

$$x_i' = u(x_a) = x_a W_{update}, \tag{7}$$

where x_a and x_i' represent the node features derived after the aggregation and update operations, respectively. $a(\cdot)$ is the aggregation operation, which simply calculates the difference between the features of x_i and its neighboring node x_j, and then concatenate it with the features of x_i. $u(\cdot)$ is the update operation in which the weight coefficient after the aggregation is updated to obtain the node x_i'.

The output of the Edgeconv is further fed into a one-dimensional convolutional layer in order to increase the diversity of the features again. The features processed are sent to a Feedforward Neural Network (FFN). This network includes two one-dimensional convolutional layers and a GELU activation function, which alleviates the problem of over-smoothing. The output of the FFN is fed into the next layer or the downsampling operation.

3.4.2. Mitigating Over-Smoothing

Within convolution operations, it is a common practice to stack a set of convolutional layers for the sake of enhancing the performance of the model trained. However, a rapid decrease in the performance, known as the over-smoothing phenomenon [40], usually occurs when many convolutional layers are stacked in the scenario of GCNs. This phenomenon should be attributed to the fact that graph convolutions aggregate the information contained in neighboring nodes. With the number of layers increases, the receptive field of the nodes will continuously expand. The features aggregated will extend to all the nodes in the graph when the number of layers reaches a certain level. As a result, the discrimination between nodes is impaired.

The depth of graph convolutions is extended by adding residual connections into the ResGCN [39]. In this study, each node carries both the spatial and semantic information related to its neighbors. When the graph convolution is performed at a deep level, the neighbors of the node will spread across the full graph, due to the downsampling and feature exchange operations. In this case, each key point of the boundary becomes the neighbor of each of the other points. Therefore, the unique structure of different shapes cannot be well represented. To address the over-smoothing issue, the core is to make the nodes at the deep level more distinctive.

Motivated by the ResGCN [39], we introduce the residual connection into each dynamic GCN layer for the purpose of alleviating the issue of over-smoothing. Considering that the linear transformation can be used to increase the diversity of features, which also decreases the probability of over-smoothing, we further add a one-dimensional convolutional layer before and after the graph convolutional layer and add a Feedforward Neural Network (FFN) at the end of the dynamic GCN layer.

4. Experimental Setup

In this section, we first elaborate the four publicly available data sets that we used in the shape recognition experiments. Then we introduce the performance metric used in these experiments. Finally, the implementation details of the proposed networks are described.

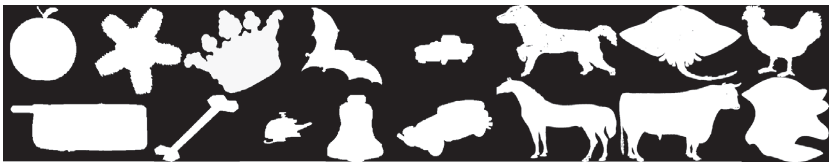


Figure 4: Sixteen examples of the MPEG-7 [41] data set.

4.1. Data Sets

We used four publicly available data sets in total, including MPEG-7 [41], Animal [42], Swedish Leaf [43] and Flavia [44]. We will introduce each data set as follows.

4.1.1. MPEG-7

The MPEG-7 [41] data set, provided by the Image Processing and Pattern Recognition Research Group of Bielefeld University in Germany, has been widely utilized in the field of shape recognition. This data set comprises 1,400 two-dimensional shapes (refer to Fig. 4 for examples), which can be divided into 70 categories. Each category contains 20 shape instances. Each shape instance is saved as a binary image. The MPEG-7 data set is characterized by its diversity of shape categories, which covers a broad range of objects, such as animals, people, plants, tools, vehicles, etc. This data set also exhibits significant variations in shapes, for example, rotation, scaling, translation, noise and occlusion. The data set has served as a useful tool for evaluating the performance of different shape descriptors and shape matching algorithms. In this study, we randomly selected 10 images from each category as training images and utilized the remaining 10 images of each category as testing images.

4.1.2. Swedish Leaf

The Swedish Leaf [43] data set originated from a leaf classification project, which was conducted at Linköping University and the Swedish Museum of Natural History. This data set encompasses 1,125 leaf images (refer to Fig. 5 for examples). These images can be categorized into 15 distinct Swedish tree species. Each species comprises 75 leaves. Since the Swedish Leaf data set only contains color images, we first binarized these images by simplistically applying the threshold of 127 to them. As a result, those images were converted into silhouette images. Then the maximum boundary of each silhouette image was extracted by following the BOCF [16] approach. We randomly selected 25 images from each species as training images and utilized the remaining 50 images in each species as testing images.

4.1.3. Flavia

As one of the most frequently utilized data sets in the domain of leaf recognition, the Flavia [44] data set comprises 1,907 leaf images (refer to Fig. 6 for examples),

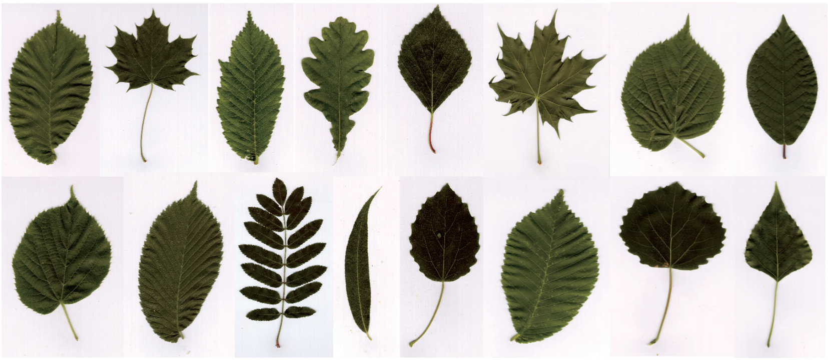


Figure 5: Sixteen examples of the Swedish Leaf [43] data set.

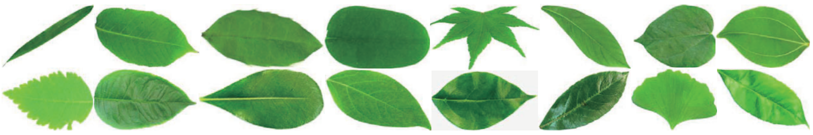


Figure 6: Sixteen examples leaf images of the Flavia [44] data set.

which can be grouped into 32 species. The majority of the leaves included in the Flavia data set are the common plants, which can be found in the Yangtze River Delta region of China. Each species consists of at least 50 leaf images. Same as the Swedish Leaf [43] data set, only color images are contained in the Flavia data set. We utilized the same method to binarize these images into silhouette images and extract the maximum boundary from each silhouette image. In terms of each species, 945 images were randomly selected as training images and the remaining images were used as testing images.

4.1.4. Animal

The Animal [42] data set consists of 2,000 silhouette images (refer to Fig. 7 for examples). These images were divided into 20 categories and each category comprises 100 images. Since the animal contained in the same category may show different poses and different animals may manifest the resemblance in certain poses, the Animal data set exhibits substantial intra-class variation, which makes the data set particularly challenging for the shape recognition task. We randomly selected 50 images from each category as training images and used the remaining 50 images in each category as testing images.

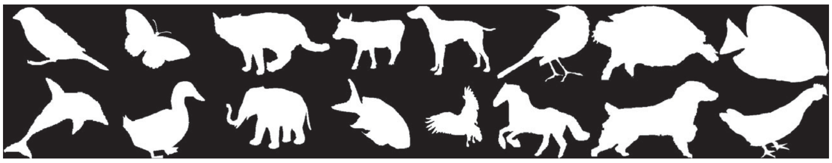


Figure 7: Sixteen examples of the Animal [42] data set.

4.2. Performance Metric

To assess the performance of the proposed network or the existing baselines in the shape recognition task, the recognition accuracy was utilized as the performance metric, which can be defined as follows:

$$\text{Accuracy} = \frac{\textit{Number of Correctly Recognized Samples}}{\textit{Number of All Samples}} \times 100\%. \tag{8}$$

Since this metric is able to reflect the capability of a shape recognition method in representing and identifying shapes, it provides an intuitive measure for comparing our approach against other methods.

4.3. Implementation Details

Table 1: Details of the parameters used during the training process.

Parameter	Value
Epochs	500
Optimizer	AdamW
Batch Size	16
Start Learning Rate (LR)	1×10^{-3}
Number of KNN Neighbors	6
Dropout Probability	0.1
Use Stochastic	True
Times of Data Augmentation	20
Small Scale	1,200
Middle Scale	700
Large Scale	300
Multi-scale	[300, 700, 1200]

GCNs normally select neighboring nodes during the convolution computation. We set the number of neighboring nodes selected, k, to 7 by taking into account the composition quantity of the semantic parts of a shape. GELU [45] was employed as the non-linear activation function. For the purpose of mitigating overfitting, we set the dropout probability to 0.1. To reduce the computational load, prevent overfitting and enhance the robustness of the graph, random GCN was utilized. Regarding the randomization factor commonly used in the graph convolutions, we set it to 0 for the sake of preventing the interference with the connections between different parts when GCNs reach the deeper layers with the fewer nodes. Only the image rotation operation was utilized for data augmentation prior to maximum boundary extraction, in which each image was rotated at fixed angles starting from 30° and increasing by 15° up to 315°. As a result, 20 additional images were obtained in terms of each shape. The details of the parameters used during the training process are reported in Table 1. We implemented the proposed BASR-GCN using PyTorch 1.13. The network was trained and tested on a single NVIDIA 3090 GPU.

In this study, we sampled key points from a boundary at a fixed interval, which indicates the scale where we extract shape representation. In total, three single scales were used in our experiments, including small scale, middle scale and large scale, at which 1,200, 700 and 300 key points were sampled from the boundary, respectively. Regarding the BASR-GCN-MS, all the three scales were utilized in order to learn the multi-scale representation. In terms of a data set, the experiment was performed on a random split of it only once, unless otherwise specified.

5. Experimental Results

In this section, we report the experimental results obtained using the proposed BASR-GCN along with three single scales and multiple scales on the four data sets. Our results are also compared with those produced by the existing methods.

5.1. MPEG-7

The results derived using the proposed BASR-GCN and 15 baselines on the MPEG-7 [41] data set are reported in Table 2. As can be seen, the multi-scale BASR-GCN outperformed all its counterparts and achieved the accuracy of 99.14%. Eight misclassified shape images together with eight shape images of the corresponding categories classified are shown in Fig. 8. It can be observed that the silhouette images of the ground-truth and classified categories manifest similar boundaries and thicknesses, which pose the challenge to the shape recognition task, even though there is a significant semantic difference between both the categories, such as guitars and spoons.

5.2. Swedish Leaf

As reported in Table 3, the BASR-GCN outperformed all its counterparts on the Swedish Leaf [43] data set, no matter what scales were used. In particular, the multi-scale BASR-GCN produced the accuracy of 100%, which was higher than the best result 98.74% achieved among the 15 baselines.

Table 2: Comparison of the accuracy values obtained using the proposed BASR-GCN together with different scales and 15 baselines on the MPEG-7 [41] data set.

Method	Accuracy (%)
Contour Segments [42]	91.10
Class Segment Set [46]	90.90
Skeleton Paths [42]	86.70
CNN-SCN [19]	90.99
EIFR-LR [2]	96.76
EIFR-LC [2]	96.77
FFNET [25]	95.57*
EfficientViT-m4 [24]	95.86*
ViG [38]	95.14*
ICS [42]	96.60
GHOSM [47]	97.40
BOCF [16]	97.16 ± 0.79
BoShUDL [48]	98.75
LCMR [49]	98.75
BoSCP [17]	98.41
BASR-GCN-Small (Ours)	98.14
BASR-GCN-Middle (Ours)	98.43
BASR-GCN-Large (Ours)	98.14
BASR-GCN-MS (Ours)	**99.14**

* The results of FFNET, EfficientViT and ViG were obtained by running the original source code on the MPEG-7 [41] data set.

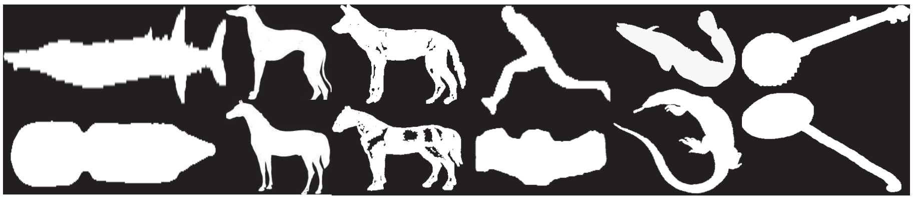


Figure 8: Examples of the shape images misclassified by our BASR-GCN-MS on the MPEG-7 [41] data set. The first row shows eight shape images which belong to different categories, including Fish, Dog, Dog, Stef, Sea Snake, Guitar, respectively, while the second row displays eight shape images of the corresponding misclassified categories which comprises Pencil, Horse, Horse, Watch, Lizzard, Spoon, respectively.

5.3. Flavia

As shown in Table 4, the multi-scale BASR-GCN yielded the better result than that produced by a single-scale BASR-GCN and outperformed the 15 baseline methods on the Flavia [44] data set. In Fig. 9, eight misclassified shape images together with eight shape images of the corresponding categories classified are displayed. It can be seen that the boundaries, thicknesses and skeletal topological structures of the leaves contained in different categories exhibit the obvious similarity. This finding should be

Table 3: Comparison of the accuracy values obtained using the proposed BASR-GCN together with different scales and 15 baselines on the Swedish Leaf [43] data set.

Method	Accuracy (%)
EIFR-LR [2]	96.34
EIFR-LC [2]	94.42
VGG16 (Pre-trained)-RF	96.13*
MTD+1-NN [50]	97.62
MTD + LBP-HF [50]	98.15
FFNET [25]	98.13*
EfficientViT-m4 [24]	97.87*
ViG [38]	98.53*
Deep-Plant [51]	97.54
LCMR [49]	98.33
HGO-CNN [50]	96.83
BOCF [16]	96.56
CBoW [52]	97.23
CRSA [1]	96.62
BoShUDL [48]	98.74
BASR-GCN-Small (Ours)	99.73
BASR-GCN-Middle (Ours)	99.73
BASR-GCN-Large (Ours)	99.60
BASR-GCN-MS (Ours)	**100**

* The results of FFNET, EfficientViT, ViG and VGG16 (Pre-trained)-RF were obtained by running the original source code on the Swedish Leaf [43] data set.

responsible for the misclassification that our method produced.

Fig. 10 presents the confusion matrix plotted using the results of our BASR-GCN-MS method on the Flavia [44] data set. As can be seen, the proposed method achieved a high recognition accuracy on most categories, covering not only smooth and regular leaf shapes but also noisy and uneven leaf shapes. (Note that the number of images contained in different categories varies). In total, only seven test samples were misclassified. It is demonstrated that our method achieved strong robustness across different leaf categories.

5.4. Animal

The BASR-GCN was also tested on the Animal [42] data set together with 15 baselines. As reported in Table 5, the proposed BASR-GCN-MS outperformed all its counterparts. Figure 11 displays eight misclassified shape images and eight shape images of the corresponding categories classified. Significant intra-class variation can be observed, due to the different postures of the same animal species. This variation, coupled

Table 4: Comparison of the accuracy values obtained using the proposed BASR-GCN along with different scales and 15 baselines on the Flavia [44] data set.

Method	Accuracy (%)
MLBP [53]	97.55
RM-LBP [54]	97.94
OM-LBP [54]	97.89
RIWD [55]	97.50
Deep-Plant [51]	98.22
HGO-CNN [50]	97.53
VGG16 [56]	95.00
VGG16 (Pre-trained)-RF	95.53*
VGG19 [56]	96.25
FFNET [25]	98.34*
EfficientViT-m4 [24]	98.54*
ViG [38]	98.75*
MTD+1-NN [50]	92.66
MTD + LBP-HF [50]	99.16
SSV [57]	98.78
BASR-GCN-Small (Ours)	98.86
BASR-GCN-Middle (Ours)	99.06
BASR-GCN-Large (Ours)	98.02
BASR-GCN-MS (Ours)	**99.27**

* The results of FFNET, EfficientViT, ViG and VGG16 (Pre-trained)-RF were obtained by running the original source code on the Flavia [44] data set.

with the resemblance between different species shown in some specific postures, lead to the outcome that the Animal [42] data set is challenging.

Figure 12 presents the confusion matrix produced using our BASR-GCN-MS method on the Animal [42] data set. It can be seen that our method was able to derive a high recognition accuracy for relatively small and irregular shape categories, such as Bird and Rat. In particular, confusion occurred when the boundaries and skeleton topology structures between two categories showed a high similarity, such as Cat and Dog.

5.5. Other Performance Metrics

To augment the evaluation of the proposed method, we further calculated the precision, recall and F1-score values obtained using our method across the four data sets. These metrics aim to measure the classification performance of the model from different perspectives. The results in terms of those metrics are presented in Table 6. It can be seen that our method produced high values with regard to the three metrics.

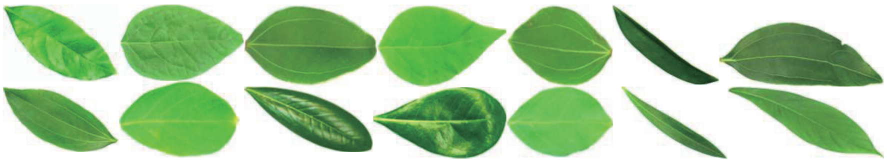


Figure 9: Examples of the shape images misclassified by our BASR-GCN-MS on the Flavia [44] data set. The first row shows seven shape images which belong to different categories, including Sweet Osmanthus, Wintersweet, Chinese Cinnamon, Wintersweet, Chinese Cinnamon, Ford Woodlotus, Chinese Cinnamon, respectively, while the second row displays seven shape images of the corresponding misclassified categories which comprises Chinese Cinnamon, Crape Myrtle, Southern Magnolia, Japanese Cheesewood, Crape myrtle, Oleander, Nanmu, respectively.

6. Ablation Study

To examine the effectiveness of different components of the proposed network, we conducted a series of ablation experiments. For simplicity, only the proposed BASR-GCN-MS and the Swedish Leaf [43] data set were used in the ablation study. In this section, we report the results derived in the ablation study.

6.1. Effect of the Number of Neighboring Nodes

Given a node, each GCN layer in the dynamic GCN block selects K neighboring nodes. As a result, the adjacency matrix is reconstructed. Thus, the value of K affects the feature exchange between neighboring nodes and the connections established between different parts of the boundary. If the number of neighboring nodes is too small, insufficient neighbors might be used for feature exchange, which impairs the accuracy of shape representation. On the contrary, a large number of nodes may result in the redundant information and a lack of distinctive features between nodes, which lead to the over-smoothing issue. We tested four different K values. The results obtained using the proposed BASR-GCN-MS are reported in Table 7. As can be seen, the best performance was obtained when the value of K was set to 6.

6.2. Effect of the Scale of Key Points

In this study, we discretized the boundary into a predefined number of key points, which have a fixed interval. Hence, the number of key points reflects the scale at which shape representation is learned. To evaluate the effect of different scales on the proposed BASR-GCN, we conducted an experiment using three scales: small, middle and large, at which 1,200, 700 and 300 key points are sampled, respectively. In addition, we tested four different combinations of scales. The results derived using seven scale schemes are shown in Table 8. It can be observed that the combination of the large and middle scales or the combination of the large, middle and small scales produced

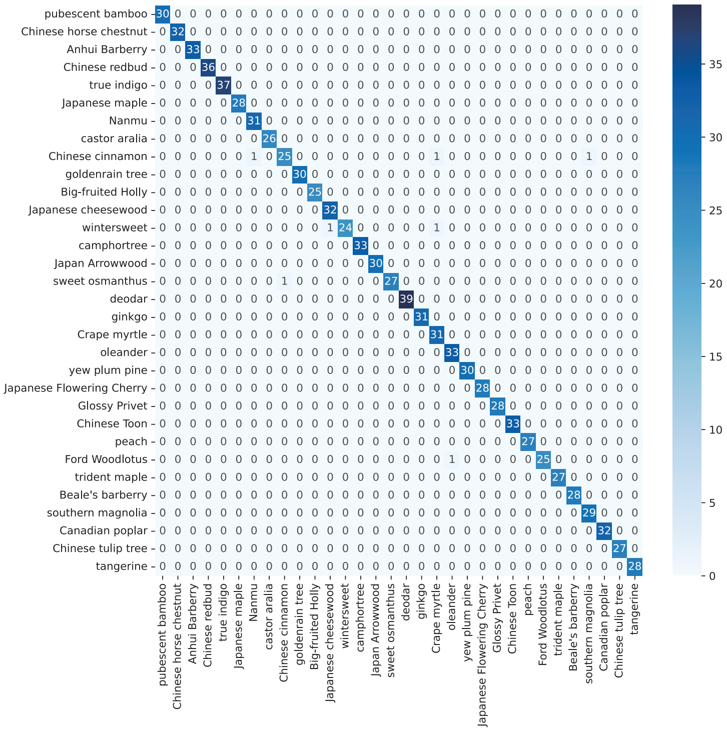


Figure 10: The confusion matrix produced by our BASR-GCN-MS method on the Flavia [44] data set. The rows represent the actual categories and the columns denote the categories that our method classified.

the best result. However, the utilization of only the small or middle scale also generated the comparable result. It should be noted that the combination of the three scales consistently outperformed the combinations of two scales or a single scale on the other data sets.

In addition, the performance gain obtained using multi-scale feature schemes was relatively small on the Swedish Leaf [43] data set. We attribute this observation to the high inter-class separability and low intra-class variance inherent in the data set. Many leaf categories exhibit distinctive global shapes with minimal deformation or topological complexity. As a result, the performance of our BASR-GCN reached the saturation point and there was not much room for improvement. In this situation, the use of multiple scales only brought a marginal benefit because most of the discriminative information had already been captured at a single scale.

6.3. *Effect of Different Graph Convolutions*

To examine the effect of different graph convolutions on the performance of the proposed method, we compared three different graph convolutions. The results are

Table 5: Comparison of the accuracy values obtained using the proposed BASR-GCN along with different scales and 15 baselines on the Animal [42] data set.

Method	Accuracy (%)
Class Segment Set [46]	69.70
ICS [42]	78.40
IDSC [58]	73.60
Contour Segments [42]	71.70
Bag of SIFT [59]	74.90
Skeleton Paths [42]	67.90
FFNET [25]	79.40*
EfficientViT-m4 [24]	77.40*
ViG [38]	80.30*
BOCF [16]	83.40
BoCF-LP [60]	86.30
ConBOW [61]	86.00
EIFR-LR [2]	82.69 ± 0.58
BoShUDL [48]	89.01
BoSCP-LP [60]	89.70
BASR-GCN-Small (Ours)	90.40
BASR-GCN-Middle (Ours)	90.90
BASR-GCN-Large (Ours)	87.90
BASR-GCN-MS (Ours)	**91.90**

* The results of FFNET, EfficientViT and ViG were obtained by running the original source code on the Animal [42] data set.

displayed in Table 9. It can be seen that the EdgeConv [30] outperformed its two counterparts. This result should be due to the fact that the EdgeConv recomputes the difference between the features of two nodes and updates the neighboring nodes of each node, which enables it capture the features at different scales and owns permutation invariance. Therefore, the EdgeConv is particularly useful for the irregular point data.

6.4. Effect of the Features of Key Points

To represent a key point sampled from the boundary of an object, three types of features are extracted. The first type of features contain the x and y coordinates of a key point. The second type of feature is the angle between a key point and its preceding key point. The third type of features consist of the x and y coordinates of the nearest skeleton point to a key point. The three types of features are referred to as Boundary (Location), Boundary (Angle) and Skeleton (Location), respectively. The results produced by the proposed BASR-GCN-MS with different types of features of key points are reported in Table 10. As can be seen, the best result was achieved using the three types of features together. In particular, the angle feature boosted the performance of the BASR-GCN-MS better, compared to the skeleton features.

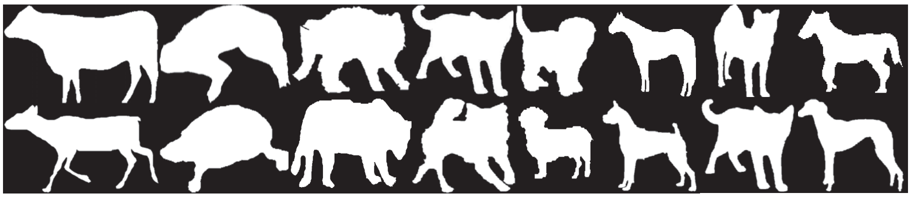


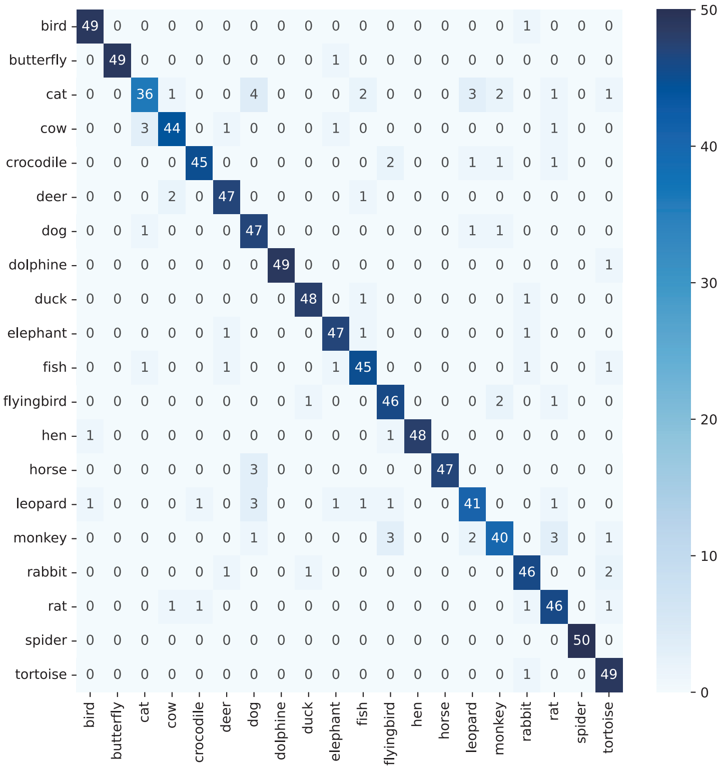
Figure 11: Examples of the shape images misclassified by our BASR-GCN-MS on the Animal [42] data set. The first row shows eight shape images which belong to different categories, including Cow, Monkey, Cat, Cat, Cat, Horse, Dog and Horse. respectively, while the second row displays eight shape images of the corresponding misclassified categories which comprises Deer, Tortoise, Leopard, Dog, Dog, Dog, Cat and Dog.

Table 6: The Precision, Recall and F1-Score values derived using our BASR-GCN-MS method on four data sets.

Data Set	Precision	Recall	F1-Score (%)
MPEG-7 [41]	99.2424	99.1429	99.1385
Swedish Leaf [43]	100.0000	100.0000	100.0000
Flavia [44]	99.2899	99.2723	99.2651
Animals [42]	92.0595	91.9000	91.8649

7. Conclusion

In this study, we proposed a Boundary-Aware Shape Recognition Graph Convolutional Network (BASR-GCN), which exploits the boundary cue explicitly. To be specific, we first extracted the maximum boundary of an object. To reduce the computational load, the boundary was then discretized into a set of equidistant key points, which still retained the shape information according to the contour integration mechanism that the Human Vision System (HVS) utilized. Regarding each key point, a five-dimensional feature vector was calculated in order to represent its location, angle and skeleton characteristics. Furthermore, a set of graphs were built on top of the key points using these features. The BASR-GCN was used to learn the spatial layout of the key points. Since the BASR-GCN dynamically generated new adjacency matrices during the convolution operation, it was able to establish the semantic connection between different parts of the boundary. We also introduced a multi-scale BASR-GCN (BASR-GCN-MS) for the purpose of exploiting the shape features extracted at different scales. To the authors' knowledge, GCNs have not been explored in the previous 2D shape recognition studies. The proposed network was tested together with four publicly available shape data sets. The results demonstrated that our network performed better than the baselines. We believe that these promising results should be due to the fact that the BASR-GCN captures the global spatial layout of the boundary and the semantic information learned by a series of dynamic GCN blocks.



	bird	butterfly	cat	cow	crocodile	deer	dog	dolphine	duck	elephant	fish	flyingbird	hen	horse	leopard	monkey	rabbit	rat	spider	tortoise
bird	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
butterfly	0	49	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
cat	0	0	36	1	0	0	4	0	0	0	2	0	0	0	3	2	0	1	0	1
cow	0	0	3	44	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
crocodile	0	0	0	0	45	0	0	0	0	0	0	2	0	0	1	1	0	1	0	0
deer	0	0	0	2	0	47	0	0	0	0	1	0	0	0	0	0	0	0	0	0
dog	0	0	1	0	0	0	47	0	0	0	0	0	0	0	1	1	0	0	0	0
dolphine	0	0	0	0	0	0	0	49	0	0	0	0	0	0	0	0	0	0	0	1
duck	0	0	0	0	0	0	0	0	48	0	1	0	0	0	0	0	1	0	0	0
elephant	0	0	0	0	0	1	0	0	0	47	1	0	0	0	0	0	1	0	0	0
fish	0	0	1	0	0	1	0	0	0	1	45	0	0	0	0	0	1	0	0	1
flyingbird	0	0	0	0	0	0	0	0	1	0	0	46	0	0	0	2	0	1	0	0
hen	1	0	0	0	0	0	0	0	0	0	0	1	48	0	0	0	0	0	0	0
horse	0	0	0	0	0	0	3	0	0	0	0	0	0	47	0	0	0	0	0	0
leopard	1	0	0	0	1	0	3	0	0	1	1	1	0	0	41	0	0	1	0	0
monkey	0	0	0	0	0	0	1	0	0	0	0	3	0	0	2	40	0	3	0	1
rabbit	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	46	0	0	2
rat	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	46	0	1
spider	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
tortoise	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	49

Figure 12: The confusion matrix produced by our BASR-GCN-MS method on the Animal [42] data set. The rows represent the actual categories and the columns denote the categories that our method classified.

Although promising results have been derived using our method, it still has at least three limitations. First, its performance is highly dependent on the quality of boundary extraction. Noisy or incomplete contours may lead to inferior recognition accuracy. Second, it employs a fixed keypoint sampling strategy, which may not be optimal for shapes of varying complexities. Third, it has not been evaluated under occlusion scenarios and its robustness to occlusion remains unknown.

Acknowledgement

This study was supported by the National Natural Science Foundation of China (NSFC) (No. 42176196).

References

[1] L. C. Ribas, O. M. Bruno, Learning a complex network representation for shape classification, Pattern Recognition 154 (2024) 110566.

Table 7: The accuracy (%) derived using the proposed BASR-GCN-MS with different numbers of neighboring nodes on the Swedish [43] data set.

Number of Neighboring Nodes	**Accuracy (%)**
3	99.73
5	99.87
6	**100**
9	99.47

Table 8: The accuracy (%) derived using the proposed network with different combinations of scales on the Swedish [43] data set.

Scales	**Accuracy (%)**
Large	99.60
Middle	99.73
Small	99.73
[Large, Middle]	**100**
[Large, Small]	99.73
[Middle, Small]	99.87
[Large, Middle, Small]	**100**

[2] J. Blandon, A. Orozco-Gutierrez, A. M. Álvarez-Meza, An enhanced and interpretable feature representation approach to support shape classification from binary images, Pattern Recognition Letters 151 (2021) 348–354.

[3] Z. Jiang, C. Zhou, Comprehensive study on shape representation methods for shape-based object recognition, Journal of Optics 53 (2024) 1890–1896.

[4] Z. Karimi, S. P. Savant, A. Zeid, S. V. Kamarthi, Shape recognition and corner points detection in 2d drawings using a machine learning long short-term memory (lstm) approach, European Journal of Artificial Intelligence and Machine Learning 3 (2024) 1–9.

[5] A. Hemmat, A. Davies, T. A. Lamb, J. Yuan, P. Torr, A. Khakzar, F. Pinto, Hidden in plain sight: evaluating abstract shape recognition in vision-language models, in: Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24, Curran Associates Inc., Red Hook, NY, USA, 2024.

[6] Y. Zhang, M. A. Mazurowski, Convolutional neural networks rarely learn shape for semantic segmentation, Pattern Recognition 146 (2024) 110018.

[7] M. G., S. Elizabeth, S. Mathew Koshy, Circular mesh-based shape and margin descriptor for object detection, Pattern Recognition (2018) 97–111. doi:10.1016/j.patcog.2018.07.004.

Table 9: The accuracy (%) derived using the proposed BASR-GCN-MS with different graph convolutions on the Swedish [43] data set.

Graph Convolution	**Accuracy (%)**
EdgeConv [30]	**100**
GraphSAGE [35]	99.73
Max-Relative GraphConv [39]	99.87

Table 10: The accuracy (%) derived using the proposed BASR-GCN-MS with different types of features of key points on the Swedish [43] data set.

Features of Key Points	**Accuracy (%)**
Boundary (Location)	99.33
Boundary (Location)+Skeleton (Location)	99.47
Boundary (Location+Angle)	99.87
Boundary (Location+Angle)+Skeleton (Location)	**100**

[8] C. Yang, Plant leaf recognition by integrating shape and texture features, Pattern Recognition 112 (2021) 107809. doi:10.1016/j.patcog.2020.107809.

[9] D. Giveki, Robust moving object detection based on fusing atanassov's intuitionistic 3d fuzzy histon roughness index and texture features, International Journal of Approximate Reasoning 135 (2021) 1–20.

[10] Y.-F. Feng, L.-Y. Shen, C.-M. Yuan, X. Li, Deep shape representation with sharp feature preservation, Computer-Aided Design 157 (2023) 103468.

[11] X. Bai, C. Rao, X. Wang, Shape vocabulary: A robust and efficient shape representation for shape matching, IEEE Transactions on Image Processing 23 (2014) 3935–3949.

[12] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE transactions on pattern analysis and machine intelligence 24 (2002) 509–522.

[13] Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, H. Yang, Reduced reference perceptual quality model with application to rate control for video-based point cloud compression, IEEE Transactions on Image Processing 30 (2021) 6623–6636.

[14] I. K. Kazmi, L. You, J. J. Zhang, A survey of 2d and 3d shape descriptors, in: 2013 10th International Conference Computer Graphics, Imaging and Visualization, IEEE, 2013, pp. 1–10.

[15] Z. Jiang, C. Zhou, Comprehensive study on shape representation methods for shape-based object recognition, Journal of Optics 53 (2024) 1890–1896.

[16] X. Wang, B. Feng, X. Bai, W. Liu, L. Jan Latecki, Bag of contour fragments for robust shape classification, Pattern Recognition 47 (2014) 2116–2125. doi:10.1016/j.patcog.2013.12.008.

[17] W. Shen, Y. Jiang, W. Gao, D. Zeng, X. Wang, Shape recognition by bag of skeleton-associated contour parts, Pattern Recognition Letters 83 (2016) 321–329.

[18] S. A. Eslami, N. Heess, C. K. Williams, J. Winn, The shape boltzmann machine: a strong model of object shape, International journal of computer vision 107 (2014) 155–176.

[19] C. Zhang, Y. Zheng, B. Guo, C. Li, N. Liao, Scn: a novel shape classification algorithm based on convolutional neural network, Symmetry 13 (2021) 499.

[20] X. Dong, M. J. Chantler, Perceptually motivated image features using contours, IEEE Transactions on Image Processing 25 (2016) 5050–5062. doi:10.1109/TIP.2016.2601263.

[21] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), volume 2, IEEE, 2006, pp. 2169–2178.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. doi:10.1109/cvpr.2010.5540018.

[23] D. Giveki, M. A. Soltanshahi, H. Rastegar, Shape classification using a new shape descriptor and multi-view learning, Displays 82 (2024) 102636.

[24] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, Y. Yuan, Efficientvit: Memory efficient vision transformer with cascaded group attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14420–14430.

[25] X. Gao, L. Chen, F. Chao, X. Chang, X. Gao, H. Jiang, L. Liu, H. Zhang, The effectiveness of a simplified model structure for crowd counting, IEEE Transactions on Instrumentation and Measurement (2025).

[26] C. P. Lee, K. M. Lim, Y. X. Song, A. Alqahtani, Plant-cnn-vit: plant classification with ensemble of convolutional neural networks and vision transformer, Plants 12 (2023) 2642.

[27] W. Zhou, X. Lin, J. Lei, L. Yu, J.-N. Hwang, Mffenet: Multiscale feature fusion and enhancement network for rgb–thermal urban road scene parsing, IEEE Transactions on Multimedia 24 (2021) 2526–2538.

[28] W. Zhou, H. Zhang, W. Yan, W. Lin, Mmsmcnet: Modal memory sharing and morphological complementary networks for rgb-t urban scene semantic segmentation, IEEE Transactions on Circuits and Systems for Video Technology 33 (2023) 7096–7108.

[29] T. Hossain, J. Ma, J. Li, M. Zhang, Invariant shape representation learning for image classification, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, 2025, pp. 4279–4289.

[30] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, J. M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Transactions on Graphics (2019) 1–12. doi:10.1145/3326362.

[31] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Neural Information Processing Systems,Neural Information Processing Systems (2017).

[32] F. Monti, M. Bronstein, X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks (2017).

[33] A. Micheli, Neural network for graphs: A contextual constructive approach, IEEE Transactions on Neural Networks 20 (2009) 498–511.

[34] T. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv: Learning,arXiv: Learning (2016).

[35] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Neural Information Processing Systems,Neural Information Processing Systems (2017).

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903 (2017).

[37] C. Liu, Y. Tian, Z. Chen, J. Jiao, Q. Ye, Adaptive linear span network for object skeleton detection, IEEE transactions on image processing 30 (2021) 5096–5108.

[38] K. Han, Y. Wang, J. Guo, Y. Tang, E. Wu, Vision gnn: an image is worth graph of nodes, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22, Curran Associates Inc., Red Hook, NY, USA, 2022.

[39] G. Li, M. Muller, A. Thabet, B. Ghanem, Deepgcns: Can gcns go as deep as cnns?, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9267–9276.

[40] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[41] L. J. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), volume 1, IEEE, 2000, pp. 424–429.

[42] X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: 2009 IEEE 12th international conference on computer vision workshops, ICCV workshops, IEEE, 2009, pp. 360–367.

[43] O. Söderkvist, Computer vision classification of leaves from swedish trees (2001).

[44] S. Wu, F. Bao, E. Xu, Y.-X. Wang, Y.-F. Chang, Q. Xiang, A leaf recognition algorithm for plant classification using probabilistic neural network, Cornell University - arXiv,Cornell University - arXiv (2007).

[45] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), Cornell University - arXiv,Cornell University - arXiv (2016).

[46] K. B. Sun, B. J. Super, Classification of contour shapes using class segment sets, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, IEEE, 2005, pp. 727–733.

[47] B. Alwaely, C. Abhayaratne, Ghosm: Graph-based hybrid outline and skeleton modelling for shape recognition, ACM Transactions on Multimedia Computing, Communications and Applications 19 (2023) 1–23.

[48] L. Yang, L. Wang, Y. Su, Y. Gao, Bag of shape descriptor using unsupervised deep learning for non-rigid shape recognition, Signal Processing: Image Communication 96 (2021) 116297.

[49] C. Yang, L. Fang, H. Wei, Learning contour-based mid-level representation for shape classification, IEEE Access 8 (2020) 157587–157601.

[50] C. Yang, Plant leaf recognition by integrating shape and texture features, Pattern Recognition 112 (2021) 107809.

[51] S. H. Lee, C. S. Chan, S. J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification, Pattern recognition 71 (2017) 1–13.

[52] J. Zeng, M. Liu, X. Fu, R. Gu, L. Leng, Curvature bag of words model for shape recognition, IEEE Access 7 (2019) 57163–57171.

[53] Y. Naresh, H. Nagendraswamy, Classification of medicinal plants: an approach using modified lbp with symbolic representation, Neurocomputing 173 (2016) 1789–1797.

[54] M. Turkoglu, D. Hanbay, Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine, Physica A: Statistical Mechanics and its Applications 527 (2019) 121297.

[55] E. Yousefi, Y. Baleghi, S. M. Sakhaei, Rotation invariant wavelet descriptors, a new set of features to enhance plant leaves classification, Computers and Electronics in Agriculture 140 (2017) 70–76. doi:10.1016/j.compag.2017.05.031.

[56] S. Anubha Pearline, V. Sathiesh Kumar, S. Harini, A study on plant recognition using conventional image processing and deep learning approaches, Journal of Intelligent & Fuzzy Systems 36 (2019) 1997–2004.

[57] G. Saleem, M. Akhtar, N. Ahmed, W. S. Qureshi, Automated analysis of visual leaf shape features for plant classification, Computers and Electronics in Agriculture 157 (2019) 270–280.

[58] H. Ling, D. W. Jacobs, Shape classification using the inner-distance, IEEE transactions on pattern analysis and machine intelligence 29 (2007) 286–299.

[59] K.-L. Lim, H. K. Galoogahi, Shape classification using local and global features, in: 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, IEEE, 2010, pp. 115–120.

[60] W. Shen, C. Du, Y. Jiang, D. Zeng, Z. Zhang, Bag of shape features with a learned pooling function for shape recognition, Pattern Recognition Letters 106 (2018) 33–40.

[61] M. Bicego, P. Lovato, A bioinformatics approach to 2d shape classification, Computer Vision and Image Understanding 145 (2016) 59–69.