The Visual Computer
https://doi.org/10.1007/s00371-023-03123-2

ORIGINAL ARTICLE

A rotation robust shape transformer for cartoon character recognition

Qi Jia[1] · Xinyu Chen[2] · Yi Wang[1] · Xin Fan[1] · Haibin Ling[3] · Longin Jan Latecki[4]

Accepted: 23 September 2023

Abstract
Recognizing cartoon characters accurately is important for animators to design and create cartoon scenarios by utilizing existing cartoon materials. Current deep learning approaches are sensitive to image rotation and heavily rely on rich textures that rarely exist in cartoon figures. In order to address this problem, the focus of our work is on the distinct nature of shapes, which mostly encodes the geometric structure of contours, rendering more discriminative and robust features than textures. We propose a rotation robust shape transformer for cartoon character recognition. As the filters in deep learning hardly detect discriminative gradient information in cartoon figures, we leverage multi-scale shape context (SC) to obtain the geometry of contour sampling points other than differences in gray level. Further, we propose a rotation-invariant positional encoding to depict the geometric relations of local shape features. The contributions of the different scales of SC templates are learned by attention-based transformer encoder. The obtained network is able to learn shape information effectively from cartoon contours only. The simplistic design attains surprisingly nearly 100% recognition accuracy, which beats both handcrafted and deep learning methods on the proposed challenging Cartoon dataset and traditional datasets. In particular, we gain 86.19% recognition accuracy on rotation test set, rendering an overwhelming superiority of 58.30 percentage higher than the state-of-the-art methods. Moreover, we develop an online cartoon character recognition application for animation scenarios.

Keywords Shape representation and learning · Transformer · Positional encoding · Cartoon character recognition

1 Introduction

Cartoon plays important roles in entertainment, education, and advertisement, attracting much research attention in the field of multimedia and computer graphics. As cartoon or animation creation is usually of high cost and labor-intensive, it is crucial for animators to effectively create new animation scenarios by recognizing and reusing existing cartoon characters [1].

Recognizing cartoon figures in animation or greeting cards accurately is quite challenging, as they lack distinctive texture information, and the same character may be represented in different colors or various dramatic gestures. Typical features of cartoon characters are mainly outlined by their structure or shape which consists of sharp and clear edges. Therefore, the key to distinguish different cartoon characters is to precisely spot the sparse and critical shape structures [2].

Shape, which provides geometric structure of objects, plays a crucial role in human visual perception, in particular, in object recognition. Recent neuropsychological studies reveal that humans have a specific brain area to process shape information [4]. Since shapes do not have brightness, color and texture information, shapes are stable to the variations in object color, texture, and light conditions. Due to these advantages, recognizing cartoon character by their shapes belongs to the oldest problems in computer vision. Shape recognition is usually considered as a classification problem. A large intra-class variation is one of the main challenges in cartoon character recognition. It is induced by deformation, articulation, occlusion, and a view point change. However, deep learning computer vision algorithms routinely focus on object textures and are heavily sensitive to image rotation [5]. In order to bridge this gap, we propose a deep archi-

✉ Xin Fan
xin.fan@dlut.edu.cn

1 International School of Information Science and Engineering, Dalian University of Technology, Dalian, China

2 School of Software Engineering, Dalian University of Technology, Dalian, China

3 Department of Computer Science, Stony Brook University, Stony Brook, USA

4 Department of Computer and Information Sciences, Temple University, Philadelphia, USA

Published online: 27 October 2023

tecture that learns cartoon feature representation from object contours only.

Early studies resorted to handcrafted features for shape representation. Most of them compute one or several geometric quantities, e.g., position, distance, and angle, of sample points along a shape and then apply pooling or coding techniques[1] to generate a shape descriptor. Shape context (SC) [6] and its variants [7] are among the most popular shape descriptors. Wang et al. [8] apply the clustering to contour segments as the coding strategy, yielding a powerful descriptor. Yu et al. [1] combine color histogram, edge, and skeleton features for cartoon character retrieval. These handcrafted features are good at representing the geometric information of shapes. However, most of them use the concatenation vector of local features as the whole shape descriptor, neglecting the geometric relations between these features. Moreover, feature engineering by manually setting various parameters highly relies on experience, lacking the flexibility to accommodate significant shape variations, such as rotation and dramatic changes of cartoon character's gesture.

Recently, deep learning-based features have gained great success in visual recognition [9]. Their strong learning ability made it possible to go beyond the limits of handcrafted features in cartoon recognition. Li et al. construct jigsaw puzzles to enhance shape features in the cartoon face classification network [10]. However, the ImageNet-trained CNNs are strongly biased toward recognizing textures rather than shapes. A cat with elephant texture inside is recognized as an elephant, which is in stark contrast to human behavioral evidence and reveals fundamentally different classification strategies [11]. The reason is that most existing learning models neglect geometric information in feature detection and representation. Most learning-based feature detectors are based on convolutional filters, and the convolution between filters and pixel values of images is taken as feature map [5], while geometric information should characterize the location distribution of contour points, which is not directly related to the pixel values. Moreover, most existing models only focus on detecting and learning features but neglecting explicitly learning the relations among them [3, 12]. Therefore, the existing deep networks are designed for detecting texture induced features in gray level, and little attention has been paid to the special nature of shapes, which mostly encode the geometric structure of contours.

We demonstrate some successful classification cases of our method in the first row of Fig. 1, which Vision Transformer (ViT) [3] failed to recognize. The classes wrongly assigned by ViT are illustrated in second row with red labels. As ViT mainly relies on texture feature for classification, ViT recognized Smurf as pigeon for their similar colors. In contrast, our method is able to distinguish different cartoon characters and capture even fine shape differences like between dolphin and plane.

Specifically, we design a shape transformer to learn attention-based multi-scale shape features and employ a rotation-invariant positional encoding for shape representation, as shown in Fig. 2. Since the input to our system are only outer shape contours, we need a local filter that can deal with this sparse contour information. We select shape context (SC), which can characterize the local differences of contour points, e.g., straight, curved left or right, and the degree of turning. SC templates of different scales are set to detect the geometric features of each sample point along the shape contour. Their contributions to the final shape features are learned and balanced by attention-based encoder of transformer. Furthermore, global geometric features should

[1] The histogram is one of the most commonly used.

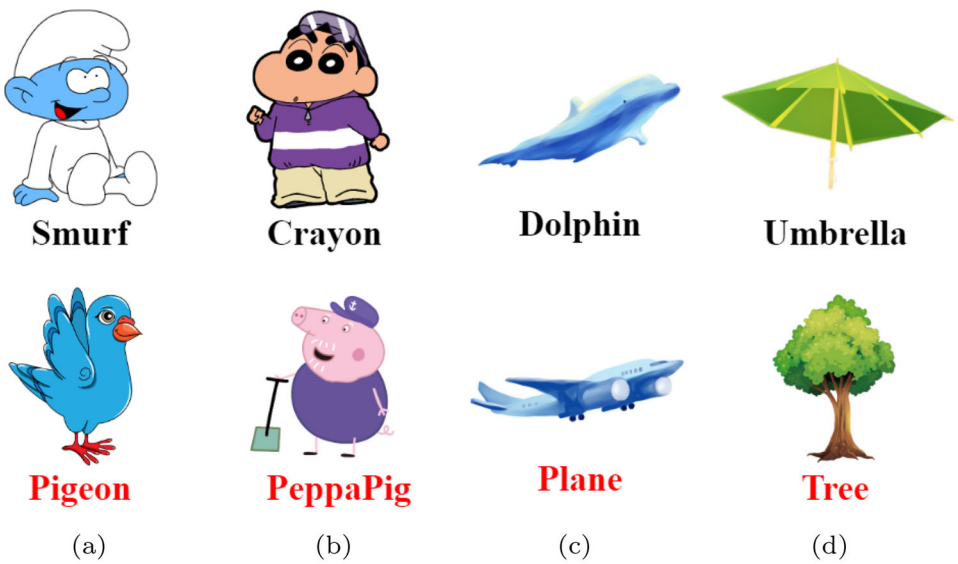

Fig. 1 Some successful classification cases of our method in the first row, which Vision Transformer (ViT) [3] failed to recognize. The classes wrongly assigned by ViT are illustrated in the second row

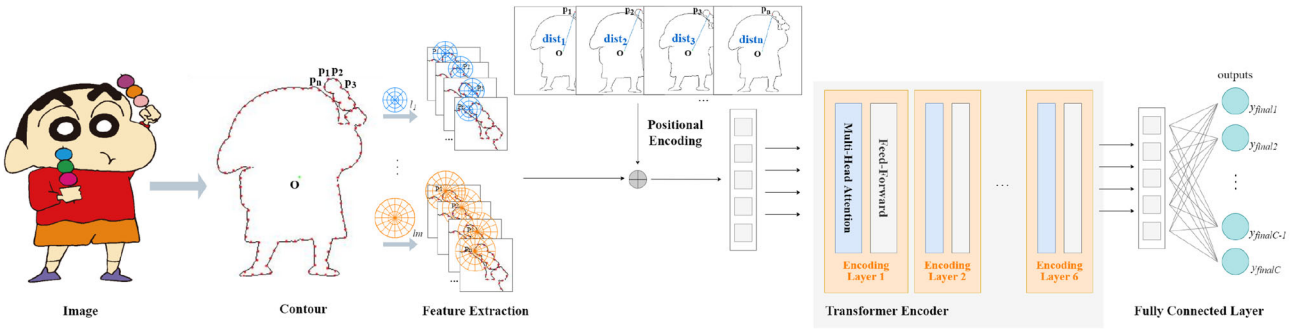

Fig. 2 Overview of the proposed framework for cartoon character recognition. Shape context (SC) templates are applied to extract feature vectors representing the geometry of the contour at different scales. For each shape, the SC features are collected clockwise at n sample points and sequentially concatenated with positional encoding. The contributions of SC features are then learned by attention-based encoding. A fully connected layer outputs the final probabilities of C categories

describe correlations of local features. We propose a rotation- and scale-invariant positional encoding to depict the geometric relation of adjacent features. Finally, the feature and positional embeddings are combined for the final results. Our contributions are summarized as follows:

- A novel shape transformer framework on explicit geometric features is constructed to learn pure geometric features other than textures in a deep fashion.
- Shape context (SC) templates in different scales are used to detect local shape features instead of traditional convolution filters to obtain the distribution of contour sample points other than differences in gray level.
- We propose a rotation-invariant position encoding method that embeds the relative distance between the local shape and the centroid, rather than introducing rotation instances in the training set, resulting in significant gains.
- We propose a new challenging Cartoon dataset and an online application based on it, assisting users to design cartoon scenarios for greeting cards or cartoon animations.[2]

Experiments on both Cartoon datasets and various shape recognition benchmarks show that our method significantly outperforms both handcrafted shape features and recent deep learning-based algorithms [3, 13]. In particular, the proposed method achieves 95.16% recognition rate, while ResNet 50 [13] only gains 25.52% on the challenging Cartoon dataset. We also gain overwhelming superiority of 58.30 percentage higher over Vision Transformer [3] on rotation test images. Moreover, an online application is developed based on cartoon character recognition of the proposed method.

[2] We provide the introduction video of the application in the supplement material.

2 Related works

This section reviews three lines of related works, i.e., handcrafted shape features, deep learning for images of few textures, and sequential learning strategies.

Cartoon or animation is popular and successful media in our life, and cartoon character recognition has been studied in computer vision for a long time [1]. The frequently used low-level features such as texture, intensity, and color cannot provide a comprehensive representation for cartoon recognition. Therefore, traditional methods tend to employ shape features for character representation [14].

Traditionally, shape is considered the contour information of the object. Given a set of finite sample points on the contour, the geometric relationship among these points can be used as a shape feature [15]. Wang et al. [16] explore shape feature by the distance between the tangent of each point and the other points. Researchers also develop descriptors to accommodate a wide range of geometric transformations [17]. There are also descriptors for some special shapes, such as lines and ellipses [18, 19]. The classic method Shape Context (SC) and its variants [6, 7] are among the most popular shape descriptors, which leverage the distribution of sample points other than their exact positions to make them more stable to noise and deformations. The SC feature is easy to obtain and widely used as basic feature for middle-level descriptors. Wang et al. develop the bag of contour fragments (BCF) [8] based on SC which achieves good performance for shape classification. However, the parameters are fixed for all categories neglecting the differences between them. Here, we explore our learnable descriptors based on SC, while utilizing neural networks to obtain adaptive parameters.

Deep learning framework explores robust features learned from extremely large datasets, which cover various object transformations and illumination changes [9]. Thereafter, numerous deep learning architectures have been designed

deliberately for low-texture targets. For cartoonlike images, such as sketches [20, 21] and Chinese handwritten characters, researchers enlarge the size of the filters to adopt networks to their sparse and structural nature [22, 23]. Lee et al. [24] classify the leaves by extracting the features of veins with CNN. Unfortunately, traditional handcrafted shape features [6, 16] are abandoned, since deep learning framework detects features upon pixel differences in gray level other than their geometry structure. Hilton et al. devise a capsule network to get the relations between features for handwritten number recognition, while neglecting the geometric nature of numbers [12]. PointNet [25] is able to learn shape information directly from sample points on 3D surfaces. We applied it to 2D sample points on contours, but its performance is significantly below the proposed approach (see Sect. 4).

In order to learn the relation of features, many works resort to memorable network architectures. Classic recurrent neural network (RNN) [26, 27] is designed for processing input sequence, which can deliver the outputs from the former sequence to the latter ones. However, it has a notorious limitation called "gradient vanishing." In order to overcome the limitation of RNN, LSTM [28–30] networks have been proposed. However, recurrent network can only learn the order of the feature sequence, neglecting geometric positional relations. In contrast, Vision Transformer (ViT) [3] uses a sequence of embedded image patches as input [3, 31–33], which has explicit representations of positional information for features. There are mainly two classes of methods to encode positional representations for transformer. One is absolute, while the other is relative. Absolute methods [34, 35] encode the absolute orders of input image patches, while relative position methods [36, 37] encode the relative distance between input elements and learn the pairwise relations of features [3, 38, 39]. However, existing encoding methods have no invariance to sequence order changes, which are undesirable for modeling geometry data structures with translation, rotation or reflection.

3 Shape transformer framework

The proposed shape transformer framework consists of shape feature extracting, rotation-invariant positional encoding, attention-based feature encoder, and classification output layer. As shown in Fig. 2, feature extraction is set to explore the local geometric feature of each sample point under different SC template scales. Then, the space relations of local features are represented by positional encoding. Thereafter, the features of the sample points incorporated with positional encoding are fed into feature encoding layer. Attention-based encoder is used to balance the contribution of different template scales and acquire the optimal SC parameter combination. Finally, a fully connected layer outputs the final classification probability. In the following, we describe in detail each of the steps involved in this process.

3.1 Local shape feature extraction

As we mentioned in previous sections, shape feature should be represented and learned according to its geometric nature. Shapes are usually expressed by contours and composed of lines, points, and curves, which are sparse structures. Thus, instead of sliding convolution kernel on the whole image in deep learning framework, we describe local shape features along its contour by shape context [6], which is easy to compute and widely used as a basic shape feature.

For each cartoon character, we sample n points $P = \{p_1, p_2, \dots p_n\}$ on the shape contour clockwise equidistantly. At each sample point $p_i \in P$, we center a shape context (SC) descriptor, i.e., a circle divided into several cells. Two examples are shown in Fig. 3. A SC template is divided into n_{dist} regions according to the radius in log-polar, and into n_θ angular regions. The number of sample points in each cell obtained by the intersection of these regions is accumulated and represented by histogram. The histogram shows the distribution of sample points surrounding p_i, which is used as the local feature at point p_i. The features for n sample points $\{p_1, p_2, \dots, p_n\}$ are denoted as $\{x_1, x_2, \dots, x_n\}$.

In the traditional method, the parameters should be selected deliberately, as they can affect the robustness and distinctiveness of the features. Take parameter n_{dist} as an example, which reflects the range of local feature covered. As shown in Fig. 3a, if it is set too small, local features tend to be similar and lower down the distinctiveness. While in Fig. 3b, the larger template may cover non-local parts of the shape, and the features are totally different for the shapes in the same category.

Generally speaking, templates with more cells are better at detecting details than the ones with fewer cells, and they can work well on complicated shapes. Templates with fewer

Fig. 3 Influence of template sizes: **a** small templates at different points produce similar features and **b** large ones at the same point on shapes of the same category produce different features

cells can be used for simple shapes to reduce redundancy. In order to balance the effect of different parameters, we design m different templates $M = \{l_1, l_2, \ldots l_m\}$ to describe local features at each point coarse to fine. Thus, the templates are used to detect features at different scales, and the local features produced by templates $l_i \in M$ are noted as $\{x_{1_{l_i}}, x_{2_{l_i}}, \ldots x_{n_{l_i}}\}$ for $1 \leq i \leq m$. We concatenate the features of all templates as the feature for each sample point. The sequential concatenation features of all sample points are the original features of the whole shape.

The shape features are invariant under scaling and translation, and robust enough to resist geometric distortions. Completely rotation-invariant features could be provided by rotating the SC templates, but this seems not to be necessary as rotation mainly affects the order of the feature vectors. For example, the features $\{x_1, x_2, \ldots, x_n\}$ for n sample points are converted to $\{x_j, x_{j+1}, \ldots, x_n, x_1, \ldots, x_{j-1}\}(1 \leq j \leq n)$ after rotation. As cartoon characters usually exhibit rotation, even reflection, it is crucial to explore rotation-invariant spatial relations among the features.

3.2 Geometric positional encoding and self-attention

Even though relations between features are crucial for recognition, they have been seldom considered. This makes the learned shape features geometrically less discriminative and causes severe matching ambiguity, especially when inter-class differences are relatively minor in comparison with intra-class variability. A straightforward recipe is to explicitly inject positional embeddings of 2D sample point coordinates. However, cartoon recognition requires scale and rotation robustness as the input figure can be in arbitrary poses.

We design a novel geometric structure embedding to encode both the sequential property and rotation-invariant geometric relations of the sample points. The core idea is to leverage the relative distances computed by each sample point and the shape centroid. As shown in Fig. 4, O denotes the centroid of the shape, and $dist_1$ is the distance between sample point p_1 and O. Obviously, the distance keeps unchanged after shape rotation. We also normalize the distances for each shape to make the encoding scale-invariant.

The concatenation feature sequence produced by all templates keeps the sequential character of sample points, while the proposed positional encoding improves the rotation invariance of the overall features, which is calculated as Eq. 1:

$$PE_{(k,2q)} = \sin\left(\frac{\text{dist}_k}{10000^{\frac{2q}{d}}}\right), PE_{(k,2q+1)} = \cos\left(\frac{\text{dist}_k}{10000^{\frac{2q+1}{d}}}\right), \tag{1}$$

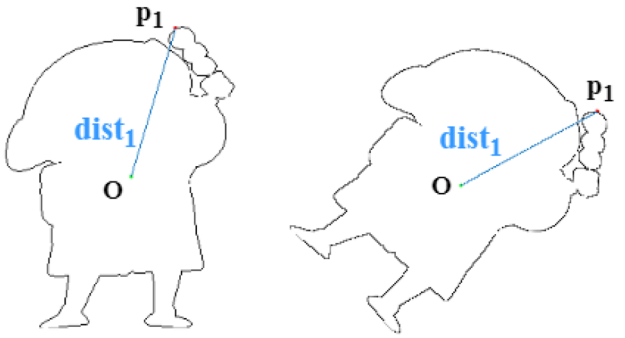

Fig. 4 The distance from each sample point to the centroid is rotation-invariant

where k is the order of the sample point, d is the feature dimension of each sample point, and q is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid. $dist_k$ is the normalized distance from point k to shape centroid. Meanwhile, we have a learnable position embedding E_{cls} for $[class]$ token.

We employ a geometric self-attention to learn the global correlations in both feature and geometric spaces among the sample points and multi-scale templates. Given the input feature coupled with positional embedding matrix z_0, the attention function is defined by scaled dot-product attention [35]:

$$\begin{aligned} Attention(Q, K, V) &= softmax(\frac{QK^T}{\sqrt{d_z}})V, \\ Q = z_0 W^Q,\ K &= z_0 W^K,\ V = z_0 W^V, \end{aligned} \tag{2}$$

where W^Q, W^K, and $W^V \in \mathbb{R}^{d \times d_z}$ are parameter matrices which are unique per layer and attention head, and d_z is the dimension of queries and keys. These matrices not only learn different weights for different scales of templates, but also give different attention to local features.

3.3 Sequential shape feature learning in transformer

As shown in Fig. 2, the proposed shape transformer includes a stack of identical layers as encoder, while each layer has two sub-layers. The first sub-layer employs a multi-head self-attention (MSA) to learn and balance the features of different templates. The second layer is a simple, position-wise fully connected feed-forward network (FFN), which is used to combine the geometric attention-based shape features of all templates and sample points. We employ a residual connection around each of the two sub-layers, followed by layer normalization (LN). Finally, we employ a fully connected layer to output the class label.

To be specific, we take the local feature x_i for sample point $p_i \in P$ as an example. We concatenate the shape context

features of all templates as the feature x_i. Then the features for n sample points are represented by $x \in \mathbb{R}^{n\times d}$ in Eq. 3, where d is the dimension of SC feature of each sample point and n is the number of sample points. We refer to the x as a feature embedding of the whole shape.

$$x = [x_1; x_2; \cdots; x_i; \cdots; x_n], \quad 1 \le i \le n. \tag{3}$$

Similar to ViT's $[class]$ token, we prepend a learnable embedding to the sequence of the embedded features. As illustrated in Eq. 4, the class token x_{class} is added to the positional embedding E_{cls}, while the original shape feature vector x is incorporated with positional encoding (PE). The concatenation of two parts is denoted as z_0, which is the input of transformer encoder, containing $n+1$ vectors. $z_0^0 = x_{class}$ represents the first vector of z_0.

$$z_0 = [x_{class} + E_{cls}; x + PE], \quad E_{cls} \in \mathbb{R}^{1\times d}, PE \in \mathbb{R}^{n\times d} \tag{4}$$

After L identical layers, we apply layer normalization (LN) to z_L^0 as the final shape representation. Finally, we employ a fully connected layer to obtain the classification probability. The classification loss is the cross-entropy.

4 Experiment results and analysis

In order to evaluate our method, we create a challenging Cartoon dataset, and employ 3 widely used datasets for the evaluation of shape descriptors, i.e., Swedish Leaf dataset [40], Animal dataset [41], and Caltech101 dataset [42]. We compare our results with both handcrafted and learning-based state-of-the-art classification approaches in order to validate the effectiveness of our shape transformer. The ablation study on multi-scale templates, feature learning strategies, and positional encoding demonstrates high classification accuracy of the proposed method, especially on rotation shapes.

4.1 Data preparation and compared algorithms

Our Cartoon dataset contains 25 categories, including car, Doraemon, Snowman, MashiMaro, Mickey, PeppaPig, etc. We show some instances and the corresponding silhouettes per category in Fig. 5.[3] Each category has about 100 images with huge intra-class variations on both shape and texture.

Swedish Leaf dataset [40] contains isolated leaves from 15 different Swedish tree species, with 75 leaves per category. As shown in Fig. 6, the inter-class difference is quite small. Animal dataset [41] contains 2000 shapes of 20 categories, with 100 animal images per category. Shapes in each category have huge intra-class variations, e.g., as shown in Fig. 7, the shape varies a lot in cat category. Therefore, the Leaf and Animal datasets are used to evaluate the distinctiveness and robustness of the proposed method. Caltech101 dataset [42] is one of the largest shape datasets, which contains 9146 binary shapes of 101 categories, each category having 40 to 800 images. As shown in Fig. 8, the inter-class difference is quite small, and the large number of categories makes it challenging to evaluate our method on this dataset.

It is obvious that the amount of shapes in any dataset is not sufficient for data training. We apply several data augmentation strategies over these datasets, including horizontal reflection and stretching. Taking the intra-class variation into consideration, we also apply elastic distortion upon the original shapes [43].

The data augmentation procedure results in 36 times of the original images per category in Cartoon dataset, 12600 images per category in Swedish Leaf dataset, 8400 images per category in Animal dataset, and 84 times of the original images per category in Caltech101 dataset. We notice that the Swedish Leaf dataset consists of color images, so we convert them into binary images.

We compare the following algorithms with our method in order to demonstrate the performance.

Handcrafted shape features: Both classical and newly developed methods are introduced to compare with the proposed method. Inner Distance Shape Context (IDSC+DP) [7] and IDSC+ Morphological [44] are classical shape descriptors, which use the distribution of sample points as shape features. Skeleton path [41] and Skeleton-based [45] methods classify shapes based on dissimilarity statistics between shortest skeleton paths. Bag of Contour Fragments (BCF) [8] and Bag of Shape Feature (BoSCP-LP) divide a contour or skeleton into coded fragments. A multi-class SVM is used to classify the shape. We keep the same training and testing ratio with [8, 46].

Learned features: In order to show the advantage of the proposed method over deep learning framework on cartoon character recognition, Lee's method is introduced [24], which is also designed for shape learning in leaf recognition. Their method uses two branches of deep learning framework to encode both shape feature and texture feature of veins. As our evaluation is based on the datasets with almost no texture information inside, we take their learning branch for shapes to keep the same protocol. We also take the classical ResNet 50 [13], PointNet [25], and newly developed ViT [3] for comparison, which perform well on various classification problems. The ratio of the training set to the test set is 1:1.

[3] We provide more instances in supplement material.

Fig. 5 Some instances and the corresponding silhouettes in Cartoon dataset

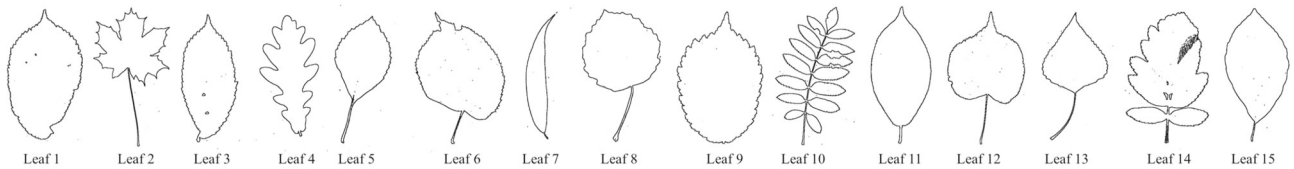

Fig. 6 Shapes examples in Leaf dataset

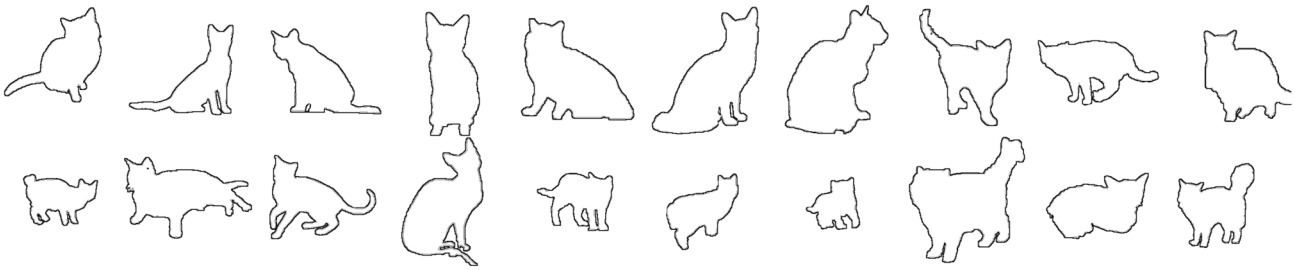

Fig. 7 Shapes examples of category Cat in Animal dataset

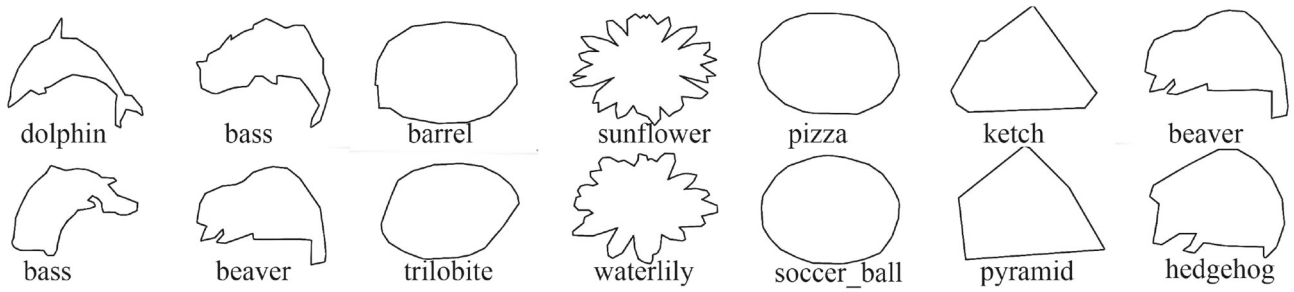

Fig. 8 Similar categories in Caltech101 dataset

Table 1 Feature dimensions by different parameters of SC templates

n_{dist}	5	7	9	9	11	25
n_θ	12	14	14	16	14	6
Feature dimension	60	98	126	144	154	150

4.2 Experimental settings

Shape context template: We select six groups of parameters widely used in handcrafted shape descriptors [8, 47]. Table 1 lists the feature dimensions produced by six combinations of parameters in the SC template. Therefore, six feature vectors are produced for each sample point. For each sample point, the input feature embeddings are the concatenation of feature vectors produced by six templates, and the dimension is $60 + 98 + 126 + 144 + 154 + 150 = 732$. We combine feature embeddings and position embeddings to get the input of transformer encoder. The output of the encoder keeps the same dimension as the input.

Network details: For each shape, we sample 150 points on the shape contour clockwise equidistantly. We take the sample point with the smallest y-coordinate as the start point for each shape. With the help of proposed positional encoding, our method is robust to rotation with no need of pre-alignment and careful selection of the starting point that are common for traditional shape descriptors.

Training details: We set a initial learning rate 4×10^{-4} for our network. The model uses cross-entropy as the loss function, and is trained for 100 epochs, in which we decay the learning rate 10 times at the 30th epoch and the 40th epoch. All the experiments are conducted on a single NVIDIA RTX 3090 GPU. We use the ADAM optimizer. The weight decay is set as 1×10^{-4}. We use a batch size that maximizes the occupancy of available GPU memory. In this paper, the batch size is set as 256 in each epoch. In order to keep the same protocol with the baselines, the ratio of the training to test is 1:1, which is the lowest training ratio among the competitive methods.

4.3 Experimental comparisons on benchmarks

For Cartoon dataset, our method only works on silhouettes, while the other deep learning methods take the original images as input. For Swedish Leaf dataset, Animal dataset, and Caltech101 dataset, all the methods take the silhouettes as input.

4.3.1 Cartoon dataset

The proposed Cartoon dataset is one of the most challenging datasets with large intra-class differences in shapes. Since the proposed method is vastly superior to handcrafted methods, we compare it with deep learning-based methods, as images all have inner color and texture in Cartoon dataset. The second column of Table 2 shows the accuracy of the proposed method is 95.16%, which is 41.13% and 69.64% higher than Lee's [24] method and ResNet 50 [13]. The reason is that each category in Cartoon dataset is designed in various colors and shapes, and some of them are highly abstract and personified. They lack distinctive texture but have typical shapes in common. Take the elephant category in Fig. 5 as an example, cartoon elephants have different colors and textures, but they all have the typical trunk. This is the reason that the proposed method outperforms the other deep learning-based methods. The gap between ViT and our results is 1.7%, which is remarkable since our method uses significantly less information.

We gather some cases with low classification rates in ViT but succeed in our method. The results of ViT are shown in Fig. 1, the first row of Fig. 1 shows the query cartoon images, including Smurf, Crayon, Dolphin, and Umbrella, and the second row shows the misclassified results. We can see the query and misclassified images have lots of similar local texture pattern and color, such as the purple body of Crayon and PeppaPig. However, the major difference between them is the body shape, which can be characterized by our attention structure according to the sequential features along the contour.

4.3.2 Swedish Leaf dataset

Swedish Leaf dataset is a challenging dataset with large inter-class similarities. The experimental results are shown in the third column of Table2. Our method obtains the highest accuracy of 99.70% in average. All the other methods have the accuracy over 85%, and the performance of BCF [8] is 96.56%, which is about 3 percentage lower than our method. However, ResNet 50 obtains the lowest accuracy of 85.46%, as the texture of leaves is not discriminative enough [13]. ViT gets the second highest result of 98.78%, which is about 1% lower than ours.

4.3.3 Animal dataset

Animal dataset includes severe intra-class variability since the same kind of animals may have various postures. As shown in Fig. 7, the category of cat includes the silhouettes of various postures, some of which are totally different. The experimental results are listed in the fourth column of Table 2. The performance of testing methods drop a lot compared with the other datasets. The basic handcrafted method IDSC [47] and Skeleton paths [41] obtain the accuracy less than 74%. The accuracy of mid-level features BCF [8] and BoSCP-LP [46] is less than 90%. PointNet [25] is unable to perform well

Table 2 Classification accuracy on the whole datasets

Methods	Classification accuracy (%)			
	Cartoon	Swedish Leaf	Animal	Caltech101
BCF [8]	–	96.56	83.40	69.04
BoSCP-LP [46]	–	–	89.77	–
IDSC+DP [7]	–	94.13	73.60	–
IDSC+Morphological [44]	–	94.80	–	–
PointNet [25]	–	–	10.00	–
Lee [24]	54.03	95.79	66.91	80.48
ResNet 50 [13]	25.52	85.46	62.70	86.79
Skeleton paths [41]	–	–	67.90	–
Skeleton-based [45]	–	94.43	–	–
ViT [3]	93.48	98.78	99.16	91.24
Multi-scale shape feature+SVM	–	75.85	49.96	18.05
Multi-scale shape feature+LSTM	69.20	97.78	98.63	**93.27**
Ours	**95.16**	**99.70**	**99.76**	92.11

Bold indicates the best performance

on this dataset. Its accuracy of only 10% is extremely low, which is most likely due to the fact that it cannot describe the neighborhood relations among contour points. Again, due to the lack of texture information in the binary animal images, ResNet 50 [13] obtains the accuracy of only 62.70%. However, ViT [3] gets 99.16%, which is only 0.6% lower than ours. It shows the amazing performance of transformer. Our method achieves an amazing performance of 99.76%, which is at least 10% higher than the state-of-the-art except ViT. In order to demonstrate the comparison results in an intuitive way, we gather some cases with top four error rates in BCF but succeed in our method. The results of BCF are shown in Fig. 9, and the top four categories with the lowest classification rates are dog, crocodile, dolphin, and leopard. The first row of Fig. 9 shows the query shapes and the second row shows the misclassified results. Take dolphin shape as an example, we can see the query and misclassified shapes have lots of similar local parts, such as the fins and fishtails. However, the major difference between them is their body shape, which can be memorized and stored by our encoder according to the sequential features along the contour. The reason is that BCF detects features by shape structure, but lacks the relation between local features.

4.3.4 Caltech101 dataset

Caltech101 dataset is one of the largest datasets with large inter-class similarities and intra-class variability. The last column of Table 2 demonstrates that the accuracy of our method is 92.11%, over 23 and 11 percentage points higher than BCF [8] and Lee's method [24], respectively. The performance of our method also outperforms ResNet 50 [13] over 5 percentage points and ViT [3] about 0.8%.

4.4 Ablation study

4.4.1 Effectiveness of multi-scale features

We take an ablation study to demonstrate the effectiveness of our multi-scale feature balance strategy. Table 3 provides the accuracy of each single SC template and the proposed method includes six templates. The first column is the list of alternative designs. The six different templates are labeled with the corresponding SC parameters. The first column under each dataset shows the overall results, and the second column lists the performance of a random category. The best performances are shown in bold, showing significant advantage of the multi-scale feature over individual template. The only exception is the dog category of Animal dataset, where a single encoder performs merely 0.07% higher than the proposed method. We also label the best performance among six templates in bold, which shows that the best performance cannot be attributed to a single template. In particular, the performance varies a lot for individual categories. Thus, we cannot fix the parameters for all datasets and categories, and our multi-scale feature balance strategy makes it possible to gain the optimal results.

4.4.2 Effectiveness of sequential feature learning

In order to explore geometric relations of shape features, we also try another two feature relation learning ways. As shown in last three rows of Table 2. Firstly, we directly concatenate multi-scale shape features and feed them to linear multi-class SVM for shape classification. The accuracy is 75.85% on Swedish Leaf dataset and 18.05% on Caltech101 dataset, which are the lowest among all the methods.

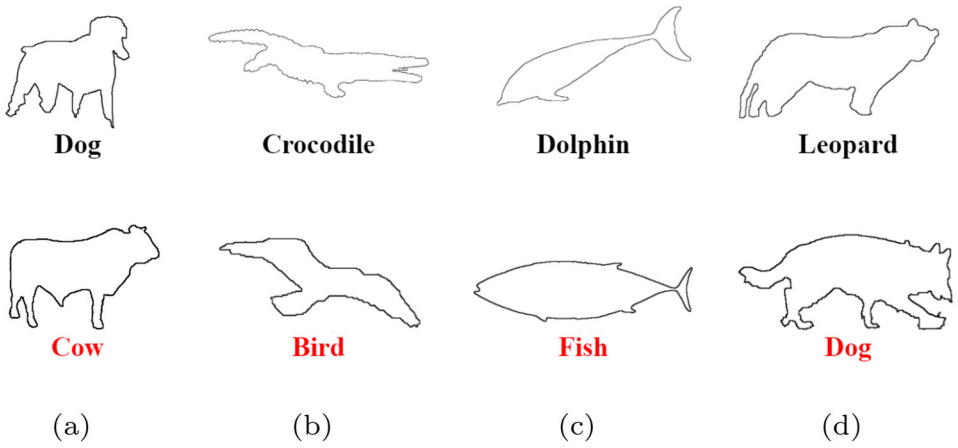

Fig. 9 Some successful classification cases of our method in the first row, which BCF [8] failed to recognize. The classes wrongly assigned by BCF are illustrated in the second row

Table 3 Comparison between single SC template and geometric attention-based methods on three datasets and on specific categories

Alternative designs	Swedish leaf		Animal		Caltech101	
	Overall	Category-l9nr	Overall	Category dog	Overall	Buddha
$l_1 : 5 \times 12$	98.76	97.14	99.22	99.40	90.31	**98.43**
$l_2 : 7 \times 14$	99.26	**97.71**	99.29	99.24	90.17	98.35
$l_3 : 9 \times 14$	**99.31**	97.52	99.39	99.60	90.05	98.10
$l_4 : 9 \times 16$	99.27	97.37	**99.68**	99.69	**91.35**	98.35
$l_5 : 25 \times 6$	99.25	97.13	99.60	99.21	89.63	96.41
$l_6 : 11 \times 14$	98.78	97.10	**99.68**	**99.86**	88.15	97.17
Ours	**99.70**	**99.11**	**99.76**	99.79	**92.11**	**98.80**

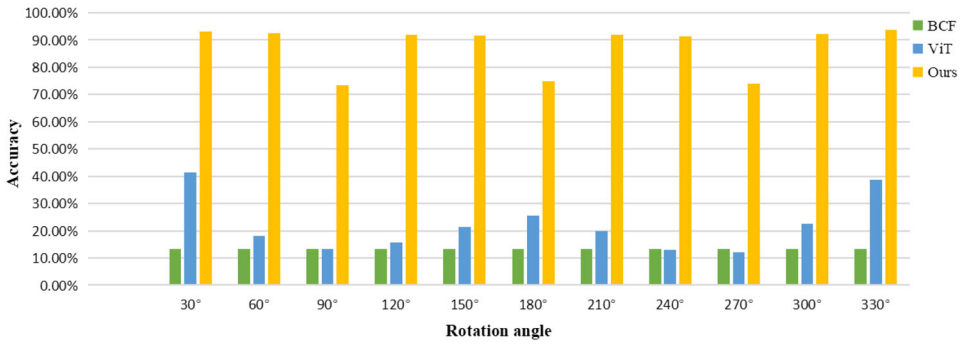

Fig. 10 Comparison on rotated shapes of Leaf dataset

The reason is that SVM takes the features as a whole and does not learn the correlation between features of different templates or sample points. Then we take long short-term memory (LSTM) framework as basic blocks to exploit the sequential feature of contours. This try obtains great gains compared with SVM, which achieves over 90% accuracy on three traditional datasets. However, the accuracy on the challenging Cartoon dataset is lower than 70%, as LSTM can only learn the order of features without geometric information. Consequently, our shape transformer with rotation-invariant positional encoding achieves almost the highest accuracy among all the datasets. An exception of 1% lower than LSTM framework appears on Caltech101 dataset. In fact, our method beats LSTM framework on 99 categories out of 101. The little gap is generated by two categories of pizza and soccer ball. As shown in the fifth column of Fig. 8, the two categories have no distinguishable geometric features, resulting little advantage of the positional encoding.

Table 4 Comparison on rotated shapes of the Leaf dataset

Methods	Classification accuracy(%)
BCF [8]	13.33
ViT [3]	27.89
Ours with SA PE [35]	45.47
Ours	86.19

4.4.3 Effectiveness of PE on rotation test set

In order to verify the robustness of the proposed positional encoding (PE) under rotation, we keep the training set unchanged, and rotate the original Swedish Leaf dataset every 30 degrees as the test set. The rotation test set has $15 \times 75 \times 12 = 13500$ shapes. We compare our method with both handcrafted method BCF [8] and learning-based method ViT [3]. We also replace our positional encoding by sinusoid absolute PE (SA PE) [35] with other part unchanged. Table 4 shows that the accuracy of the proposed method remains around 86%, which is at least 58% higher than the state-of-the-art methods. The accuracy of BCF [8] and ViT [3] drops a lot, which are both lower than 30%, as rotation heavily affects the spatial relations of local features. Sinusoid absolute PE (SA PE) [35] assigns a fixed order to the sequential sample points, which only encodes the sequential relation of features. Our framework with SA PE only gains 45.47% accuracy. However, the accuracy of our PE is almost twice of that with sinusoid absolute PE [35], as we have both sequential and geometric relations in our positional encoding strategy. Figure 10 shows the results on rotated shapes of different degrees. Our method is far ahead of the others, showing the effectiveness of the proposed rotation-invariant positional encoding strategy.

4.5 Online application

Based on the proposed Cartoon dataset, we design a novel app to help users to design cartoon scenarios by the cartoon characters. After a user first creates a shape contour, the app suggests most similar cartoon figures. The user can then select one of them and complete a scene map. The system overview is shown in Fig. 11. The app runs in real time and submits the shape contour to the server. Both geometric features and positional encoding are extracted and fed into the proposed network on the server. The recognition result and its corresponding instances are sent back to the app. Users can choose any cartoon picture they like. Our results show that the proposed network architecture and online application generalize well to real user input and enable high-quality recognition results without additional post-processing.

5 Conclusion

Since binary silhouette images as well as object contours exhibit little to no texture information, which is essential for convolutional neural networks (CNNs) for cartoon feature learning, we propose a novel way of feature learning. The features of different scales are combined and their contributions are integrated and weighted with a self-attention layer. The novel positional encoding retains the geometric position relation of sample points. All of these yield a novel network architecture for shape recognition. The presented experimental results demonstrate that our geometry-based learning network is robust to intra-class variations and has high discriminative ability to inter-class similarity for cartoon character recognition. Particularly, we achieve a breakthrough accuracy of 86.19% on rotation Leaf dataset, over 58 percentage points higher than the state of the art. As we only employ the silhouette of objects for classification, it is hard for our method to distinguish minor difference inside the shape. We will try more challenging datasets in our future

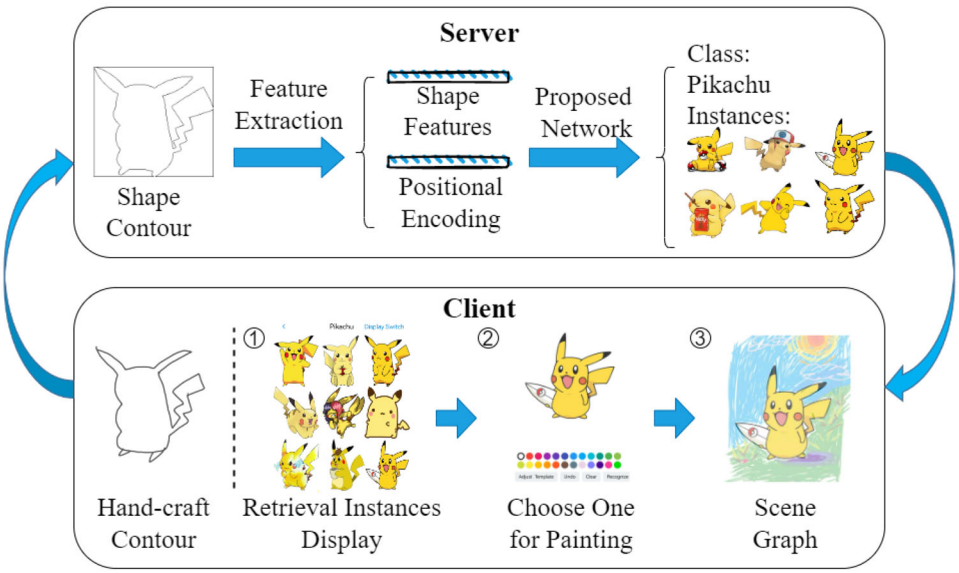

Fig. 11 System overview of the proposed online application

work, such as corrupted inputs and shapes with inner structures.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00371-023-03123-2.

Acknowledgements This work was supported in part by the Natural Science Foundation of China under Grant 62272083 and Grant 61876030, in part by the Liaoning Provincial Natural Science Foundation under Grant 2022-MS-128, in part by the Fundamental Research Funds for the Central Universities DUT23YG109, and in part by the U.S. National Science Foundation under Grant I IS-1814745.

Data availability The datasets generated or analyzed during the current study are available on Google drive (https://drive.google.com/drive/folders/1vhw907BYVosw7wMKmhD7CAe4x0NbenIG?usp=sharing).

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "A Rotation Robust Shape Transformer for Cartoon Character Recognition."

References

1. Yu, J., Liu, D., Tao, D., Seah, H.S.: On combining multiple features for cartoon character retrieval and clip synthesis. IEEE Trans. Syst. Man Cybern. B (Cybern.) **42**(5), 1413–1427 (2012)
2. Rios, E.A., Cheng, W.-H., Lai, B.-C.: Daf: re: A challenging, crowd-sourced, large-scale, long-tailed dataset for anime character recognition. arXiv preprint arXiv:2101.08674 (2021)
3. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
4. Wang, X., et al.: Domain selectivity in the parahippocampal gyrus is predicted by the same structural connectivity patterns in blind and sighted individuals. J. Neurosci. **37**(18), 4705–4716 (2017)
5. Geirhos, R., et al.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
6. Belongie, S., Malik, J., Puzicha, J.: Shape context: a new descriptor for shape matching and object recognition. Adv. Neural Inf. Process. Syst. **13**, 831–837 (2001)
7. Shekar, B., Pilar, B., Kittler, J.: An unification of inner distance shape context and local binary pattern for shape representation and classification. In: Proceedings of the 2nd International Conference on Perception and Machine Intelligence, pp. 46–55 (2015)
8. Wang, X., Feng, B., Bai, X., Liu, W., Latecki, L.J.: Bag of contour fragments for robust shape classification. Pattern Recognit. **47**, 2116–2125 (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1106–1114 (2012)
10. Li, Y., Lao, L., Cui, Z., Shan, S., Yang, J.: Graph jigsaw learning for cartoon face recognition. arXiv:2107.06532 (2021)
11. Ritter, S., Barrett, D.G., Santoro, A., Botvinick, M.M.: Cognitive psychology for deep neural networks: a shape bias case study. In: International conference on machine learning, pp. 2940–2949 (2017)
12. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. Adv. Neural Inf. Process. Syst. **30**, 3856–3866 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
14. Haseyama, M., Matsumura, A.: A cartoon character retrieval system including trainable scheme. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), vol. 3, pp. III–37 (2003)
15. Hu, R., Jia, W., Ling, H., Zhao, Y., Gui, J.: Angular pattern and binary angular pattern for shape retrieval. IEEE Trans. Image Process. **23**, 1118–1127 (2014)
16. Wang, J., Bai, X., You, X., Liu, W., Latecki, L.J.: Shape matching and classification using height functions. Pattern Recognit. Lett. **33**, 134–143 (2012)
17. Jia, Q., et al.: Hierarchical projective invariant contexts for shape recognition. Pattern Recognit. **52**, 358–374 (2016)
18. Chen, S., Xia, R., Zhao, J., Chen, Y., Hu, M.: A hybrid method for ellipse detection in industrial images. Pattern Recognit. **68**, 82–98 (2017)
19. Micusik, B., Wildenauer, H.: Structure from motion with line segments under relaxed endpoint constraints. Int. J. Comput. Vis. **124**, 65–79 (2017)
20. Yu, Q., et al.: Sketch me that shoe. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 799–807 (2016)
21. Sarvadevabhatla, R.K., Kundu, J., Babu, R.V.: Enabling my robot to play pictionary: recurrent neural networks for sketch recognition. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 247–251 (2016)
22. Yu, Q., et al.: Sketch-a-net: a deep neural network that beats humans. Int. J. Comput. Vis. **122**, 411–425 (2017)
23. Wang, T.-Q., Liu, C.-L.: Fully convolutional network based skeletonization for handwritten Chinese characters. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2540–2547 (2018)
24. Lee, S.H., Chan, C.S., Mayo, S.J., Remagnino, P.: How deep learning extracts and learns leaf features for plant classification. Pattern Recognit. **71**, 1–13 (2017)
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In: CVPR, pp. 652–660 (2017)
26. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1017–1024 (2011)
27. Xu, P., et al.: SketchMate: deep hashing for million-scale human sketch retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8090–8098 (2018)
28. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
29. Miyagi, R., Aono, M.: Sliced voxel representations with LSTM and CNN for 3D shape recognition. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 320–323. IEEE (2017)
30. Dai, G., Xie, J., Fang, Y.: Siamese CNN-BiLSTM architecture for 3D shape representation learning. In: IJCAI, pp. 670–676 (2018)
31. Carion, N., et al.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer (2020)
32. Touvron, H., et al.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357. PMLR (2021)
33. Wang, X., Girshick, R., Gupta, A. He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)

34. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning, pp. 1243–1252. PMLR (2017)
35. Vaswani, A., Guyon, I., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) NIPS, vol. 30. Curran Associates Inc., Red Hook (2017)
36. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: NAACL (2018)
37. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.: Attentive language models beyond a fixed-length context, Transformer-xl. arXiv:1901.02860 (2019)
38. Chu, X., Zhang, B., Tian, Z., Wei, X., Xia, H.: Do we really need explicit position encodings for vision transformers. arXiv preprint arXiv:2102.10882 (2021)
39. Srinivas, A., et al.: Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16519–16529 (2021)
40. Söderkvist, O.: Computer vision classification of leaves from Swedish trees. Master's thesis (2001)
41. Bai, X., Liu, W., Tu, Z.: Integrating contour and skeleton for shape classification. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pp. 360–367 (2009)
42. Li, F.-F., Andreetto, M., Ranzato, M.A.: Caltech101 image dataset. http://www.vision.caltech.edu/Image_Datasets/Caltech101/ (2003)
43. Simard, P.Y., Steinkraus, D., Platt, J.C., et al.: Best practices for convolutional neural networks applied to visual document analysis. In: Icdar, vol. 3 (2003)
44. Hu, R.-X., Jia, W., Zhao, Y., Gui, J.: Perceptually motivated morphological strategies for shape retrieval. Pattern Recognit. **45**, 3222–3230 (2012)
45. Sirin, Y., Demirci, M.F.: 2D and 3D shape retrieval using skeleton filling rate. Multimed. Tools Appl. **76**, 7823–7848 (2017)
46. Shen, W., Du, C., Jiang, Y., Zeng, D., Zhang, Z.: Bag of shape features with a learned pooling function for shape recognition. Pattern Recognit. Lett. **106**, 33–40 (2018)
47. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 286–299 (2007)

Qi Jia received the B.E. and Ph.D. degrees in computer science and technology from Dalian University of Technology , Dalian China, in 2005 and 2014, respectively. She joined the School of Software, Dalian University of Technology in 2008, where she is currently an associate professor. Her current research interests include computational geometry, image processing, and computer vision.

Xinyu Chen received the B E degree from the Dalian University of Technology, Dalian, China, in 2020. She is currently working toward the master's degree in software engineering at the Dalian University of Technology, Dalian, China. Her research interests include computer vision and deep learning.

Yi Wang received the BE and PhD degrees in computer science and technology from Jilin University, Jilin, China, in 2002 and 2009, respectively. Since 2009, she has been with the Dalian University of Technology, China. She is currently an associate professor. Her research interests include machine learning, image processing, and computer vision.

Xin Fan (Senior Member, IEEE) received the B.E. and Ph.D. degrees from Xi an Jiaotong University, Xi an, China, in 1998 and 2004, respectively. He was a Post Doctoral Research Fellow with Oklahoma State University, Stillwater, OK, USA, and the University of Texas Southwestern Medical Center, Dallas, TX, USA, from 2006 to 2009. He joined the Dalian University of Technology, Dalian, China, in 2009, where he is currently a Full Professor. His current research interes ts include image processing and machine vision. Dr. Fan received the 2015 IEEE ICME Best Student Award as the Corresponding Author and two articles were selected as the Finalist of the Best Paper Award at ICME 2017.

Haibin Ling received B.S. and M.S. fr om Peking University in 1997 and 2000, respectively, and Ph.D. from University of Maryland in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia; from 2006 to 2007, he worked as a postdoctoral scientist at UCLA; from 2007 to 2008, he worked for Siemens Corporate Research as a research scientist; and from 2008 to 2019, he was a faculty member of the Department of Computer Sciences for Temple University. In fall 2019, he joined the Department of Computer Science of Stony Brook University, where he is now a SUNY Empire Innovation Professor. His research interests include computer vision, augmented reality, medical image analysis, visual privacy protection, and human computer interaction. He received Best Student Paper Award of AC M UIST (2003), Best Journal Paper Award at IEEE VR (2021), NSF CAREER Award (2014), Yahoo Faculty Research and Engagement Award (2019), and Amazon Machine Learning Research Award (2019). He serves or served as associate editors for IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), IEEE Trans. on Visualization and Computer Graphics (TVCG), Pattern Recognition (PR), and Computer Vision and Image Understanding (CVIU). He has served as Area Chairs various times for CVPR and ECCV.

Longin Jan Latecki (Senior Member, IEEE) is currently a Professor with Temple University. He has published over 300 research papers and books. His main research interests include computer vision and machine learning. He received the Annual Pattern Recognition Society Award together with Azriel Rosenfeld for the best article published in the journal Pattern Recognition in 1998. He was a recipient of the 2018 Amazon Research Awards. He is also the Associate Editor-in-Chief of Pattern Recognition, an Editorial Board Member of Computer Vision and Image Understanding, and on the Advisory Board of Journal of Imaging.