A Swin Transformer based Framework for Shape Recognition

TIANYANG GU

Faculty of science, Queensland University of Technology, Australia, tianyanggu1@126.com

RUIPENG MIN

School of Rail Transportation, Soochow University, Suzhou, Jiangsu, China, minruipeng@126.com

ABSTRACT

Shape recognition is a fundamental problem in the field of computer vision, which aims to classify various shapes. The current mainstream network architecture is convolutional neural network (CNN), however, CNN offers limited ability to extract valuable information from simple shapes for shape classification. To address this problem, this paper proposes a deep learning model based on self-attention and Vision Transformers structure (ViT) to achieve shape recognition. Compared with the traditional CNN structure, ViT considers the long-distance relationship and reduces the loss of information between layers. The model utilizes a shifted-window hierarchical vision transformer (Swin Transformer) structure and an all-scale shape representation to improve the performance of the model. Experimental results show that the proposed model achieves superior accuracy compared to other methods, achieving an accuracy of 93.82% on the animal dataset, while the performance of state-of-the-art VGG-based method is only 90.02%.

CCS CONCEPTS • Computing methodologies → Machine learning; Machine learning approaches; Neural networks; **• Computing methodologies** → Artificial intelligence; Computer vision; Computer vision problems; Object recognition

Keywords-component; Shape recognition; Swin Transformer; Vision Transformer; Object recognition

1 INTRODUCTION

Shape is an important basis for many object recognition problems. In the real world, two-dimensional (2D) shapes are widely used in various applications, such as X-ray image classification. In recent years, many methods on shape classification [1]-[13] have been proposed. The two main components of these shape classification algorithms are shape representation and shape matching, respectively. In this paper, we solve the traditional shape matching problem with a deep learning model.

Earlier studies on shape representation were based on hand-crafted features. These hand-crafted descriptors mainly extract and represent the global or local features of a shape. The shape context (SC) [1] and its improved variants [2] are the most classical approaches of such descriptors, which represent a shape using a statistical histogram based on the spatial distribution of each sample point. The height function [3] of one shape is represented as an ordered sequence of height values for each sample point. The triangular area representation [14] applies positive and negative triangular areas to represent each contour shape. Such descriptors can highly represent shape specific information and are invariant to some geometric transformations. However, most of the shape features obtained by these methods are a set of discrete feature sequences, which are difficult to visualize directly and are prone to redundancy. To solve the above problems, a visualized all-scale shape representation method has been proposed [4]. All shape features are extracted at all scales and then represented as an image to visualize all features compactly.

In the shape matching phase, traditional methods, including dynamic programming [15], are mostly employed to obtain the minimum Euclidean distance. As a result, the computational efficiency is low when a large amount of feature data is available. Recently, deep learning models such as convolutional neural networks (CNNs) and their variants have high learning efficiency and are suitable for big data applications. However, experiments have shown that direct input of shape images [5] or shape representations [4] into CNN-based models for classification is unsatisfactory. Transformer [16] was first proposed in the field of natural language processing (NLP). Subsequently, the Vision Transformer (ViT) [17] and other transformer-based models [18]-[22] successfully adapted the transformer from the language area to the vision area. Compared to CNN structures, ViT considers long-range relations to reduce the loss of information between layers and has fewer parameters in the training phase. Currently, Swin Transformer [19] outperforms convolution-based models in a wide range of vision tasks, but we have not yet found the application in shape classification. Therefore, inspired by the superb performance of Swin Transformer, we would like to explore whether Transformer-based models can produce better results and higher efficiency.

In this paper, we propose a neural network model based on the Swin Transformer to classify 2D shapes. In addition, an all-scale representation of shape features is introduced as a preprocessing method to assist 2D shape classification. By utilizing the Swin Transformer, the model can simultaneously extract further global information from an all-scale representation and the original shape. Our approach shows excellent performance in shape classification experiments on the animal [6] dataset and the MPEG-7 [23] dataset.

The primary contributions of this work can be summarized as follows: 1) We propose a neural model for 2D shape classification based on the Swin transformer and obtain a robust performance on the test dataset; 2) our model not only utilizes the all-scale representation of the target shape, but also takes into account the raw data; 3) compared with the parameters of previous neural networks in shape classification, the proposed method has fewer parameters, which improves the efficiency of big data applications.

The rest of this paper is organized as follows. Section 2 reviews the relevant works. Section 3 describes the details of the model formulation and architecture. Section 4 explains the experimental details. Section 5 concludes the paper.

2 RELATED WORK

The existing researches on shape representation and recognition can be divided into methods based on hand-crafted features and methods based on learning models.

2.1 Handcrafted methods

The hand-crafted descriptors mainly calculate and extract the global information or local features of the target shape. Shape context (SC) [1] is the most classic method among such descriptors, in which the shape context descriptor uniformly samples the contour of the shape and utilizes the spatial distribution of each sample point to replace their precise location. This kind of descriptor has rotation, expansion, and translation invariance, but it cannot solve the problem of internal deformation and has poor noise robustness. Based on the shape context, a series of improved description methods have been successively proposed. The most prominent one among them is the inner distance shape context descriptor (IDSC) [2]. IDSC replaces the Euclidean distance in the shape context with the inner distance, where the inner distance is the shortest path between two related contour points within a shape contour. This method is less sensitive to shapes with articulated transformation. Triangular

area representation [14] performs shape matching by calculating the approximate curvature at each contour point. The triangular area representation uses the positive and negative triangular area formed by boundary points on different scales to represent each contour point. The curvature of the corresponding point of the contour is measured by the area of the triangle, and the concavity and convexity of the contour are reflected by the sign of the area value. As a global shape descriptor, this method is computationally efficient. Similarly, for each sample point on the contour, the height of each sample point is expressed as the distance between the tangent where the sample point is located and the rest of the points. Thus, the height function [3] of a shape is represented as an ordered sequence of the height values of each sample point. This method has a fast calculation speed and extracts the obvious geometric relationship between the shape sample points, but the shape representation is easily affected by noise and irregular transformation.

2.2 Learning-based methods

In response to the above problems, in recent years, some shape descriptors and classification frameworks based on learning models have been proposed to automatically learn shape features. For example, Bag of Contour Fragments (BCF) [11] and Shape Vocabulary [12] utilize some shape descriptors like SC and match these features based on learning models. Currently, deep learning models such as convolutional neural networks (CNNs) and other methods have high learning efficiency and are widely used for image classification. However, due to the lack of surface information such as color and texture, the effect of inputting this binary shape image into CNN directly is not ideal. For example, Lee [5] utilized CNN to extract leaf vein features and classify leaf shapes. This method uses a convolutional neural network (CNN) to directly learn useful leaf features from the original representation of the input data. However, this method neglects the traditional hand-crafted shape features. The visualized all-scale shape representation [4] proposed an efficient shape feature representation method that can be applied to deep learning models. Thus, it is urgent to improve the corresponding shape classification method based on cutting-edge neural networks to adapt to the wide range of applications.

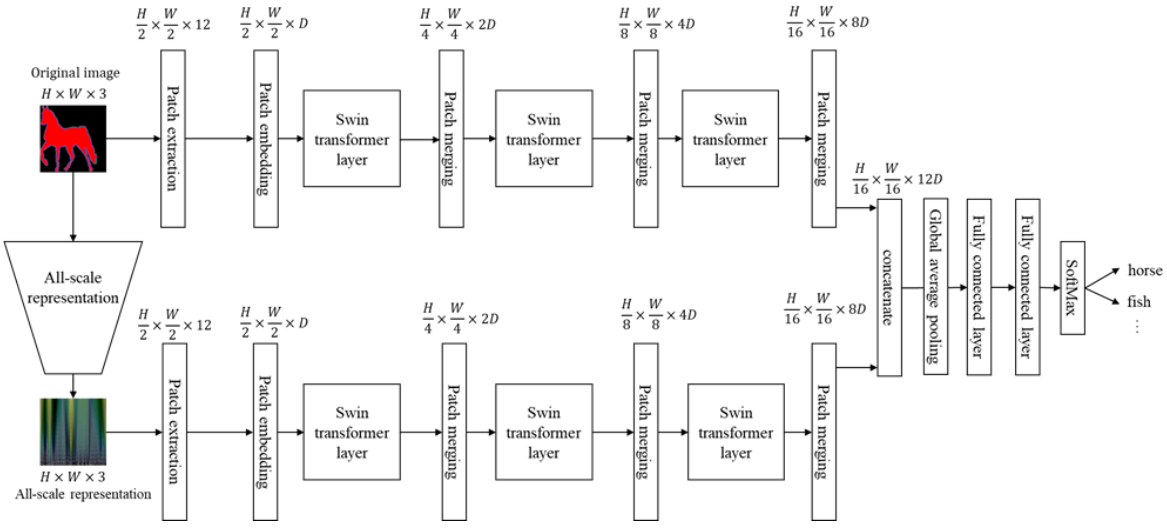

Figure 1: Overall architecture of the proposed framework

3 MODEL FORMULATION

3.1 Overall Architecture

Figure 1 shows the overall system architecture, which consists of two streams. On the one hand, for the stream that processes the original shape image, firstly, we subsample the image with heights H, width W, and RGB channels to $\frac{H}{2} \times \frac{W}{2} \times D$ using a patch extraction layer and a patch embedding layer, where D is the dimension. The output of these two layers is then fed into the first Swin Transformer layer. A patch merging layer that transforms the size to $\frac{H}{4} \times \frac{W}{4} \times 2D$ is then applied after the first Swin Transformer layer, which is followed by the second Swin Transformer. Then, the second patch merging layers outputs the data with a size of $\frac{H}{8} \times \frac{W}{8} \times 4D$, which will be processed by the last Swin Transformer layer. The last patch merging subsamples the output of the last Swin Transformer layer again to $\frac{H}{16} \times \frac{W}{16} \times 8D$. On the other hand, for the other stream, the structure is much similar, with three Swin Transformer layers and the residual connections to process the all-scale representation of the original image. The output size of two streams is the same so that the concatenation layers following these two streams double the channels from 8D to 16D. After the global average pooling, two fully connected layers with GELU as activation function and the layer with SoftMax as the activation function are utilized to generate the final output.

3.2 All-scale representation

As mentioned in the introduction, the features calculated and extracted by handcrafted descriptors are relatively complex. Thus, the shape features represented cannot be visualized intuitively and it cannot be directly plugged into the deep learning models. Therefore, this article introduces a compact color image representation method [4], which utilizes a color image to represent the invariant characteristics of the target shape at all scales. First, three kinds of invariant shape features are extracted from the shape contour, including area feature, arc length feature, and central distance feature. The three kinds of shape features are invariant features in different aspects of shape at different dimensions, which are normalized to the size of the shape in the image. Since these three shape features are extracted at different scales respect to the shape, the features at all scales in the scale space can be extracted to obtain enough shape information and fully represent the shape. After that, all the features in the scale space are compactly represented by a color image. In this image representation, the R, G, B channels are used to represent the three kinds of invariant shape features, respectively. The value of the feature is represented as the value of color. In each channel, the x axis of the image is regarding to the sequence of contour points, while the y axis is regarding to all the scales.

In the all-scale representation, the shape feature relationships between different scales and adjacent contour points are all embodied in adjacent pixels (the x-axis is the adjacent contour points, and the y-axis is the adjacent scale) in Figure 1. Therefore, neural network can effectively learn the relationship between neighboring pixels.

3.3 Swin transformer layer

A single Swin Transformer layer [19] consists of two successive modules: a window based self-attention module and a shift window based self-attention module, as shown in Figure 2. For window based self-attention module, the key is the window based self-attention layer (W-MSA). After which there is a MLP module, which has two fully connected layers with GELU as the activation function. A LayerNorm [25] is utilized before the W-

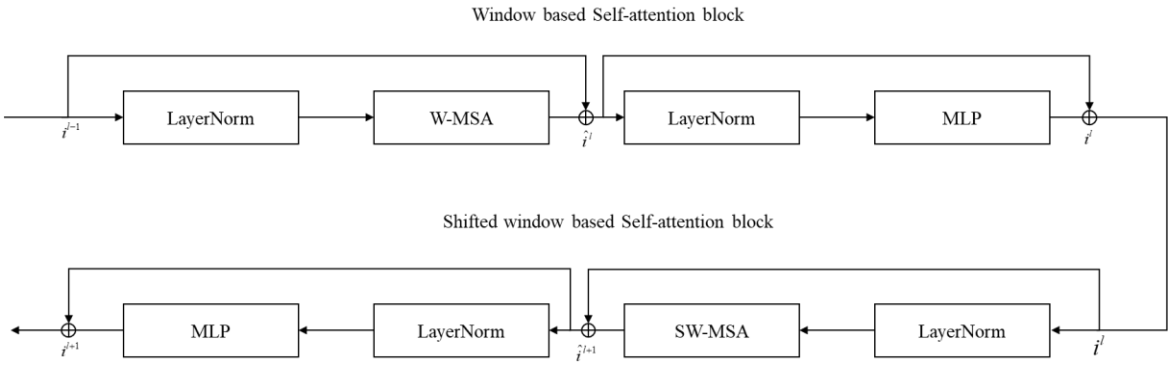

Figure 2: Swin transformer structure

MSA and the MLP, respectively. In addition, a residual connection is employed after W-MSA and MLP. The only difference between the shift-window based self-attention and the window based self-attention is that the former utilizes shifted window based multi-head self-attention (SW-MSA) while the latter utilizes window based multi-head self-attention.

LayerNorm performs normalization within each input across the last axes and is defined as:

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \gamma + \beta \qquad (1)$$

where x is the input and ϵ is a small number to prevent the divisor from being equal to zero. γ and β are learnable affine transform parameters.

W-MSA replaces global self-attention in the vision transformer [17] and can significantly reduce the computational complexity by dividing the image into patches using a window and calculating only the self-attention within each local window. Considering an image of size $h \times w$ divided into $N \times N$ patches by a window, the computational complexity of W-MSA and MSA is:

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2N^2hwC \qquad (2)$$

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C \qquad (3)$$

where C is the dimension. Eq. (2) and (3) clearly show that the complexity of W-MSA is linear to h and w while the MSA is quadratic to *hw.*

Although window-based multi-headed self-focus reduces the complexity, it cannot obtain the information of different windows. Therefore, shift-window-based multi-headed self-care (SW-MSA) is performed after the W-MSA block to solve this problem. SW-MSA shifts the windows in W-MSA by $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor)$ pixel to establish the connection between different windows. In summary, the Swin transform layer is executed in the following way:

$$\hat{i}^{l} = \text{W-MSA}(\text{LN}(i^{l-1})) + i^{l-1} \quad (4)$$

$$i^{l} = \text{MLP}(\text{LN}(\hat{i}^{l})) + \hat{i}^{l} \quad (5)$$

$$\hat{i}^{l+1} = \text{SW-MSA}(\text{LN}(i^{l})) + i^{l} \quad (6)$$

$$i^{l+1} = \text{MLP}(\text{LN}(\hat{i}^{l+1})) + \hat{i}^{l+1} \quad (7)$$

3.4 Patch extraction and embedding

Since the self-attentive structure receives a different sequence of tokens from the two-dimensional image, the patch extraction and embedding layer processes the original input image into a shape that is compatible with the self-attentive. Assuming that one token of the input self-attention contains information of $S \times S$ patches and the size of the original input image is $L \times L \times 3$, the patch extraction layer can resize the size of the input to $\frac{L}{S} \times \frac{L}{S} \times 3S$ by using a CNN layer with a size of $S \times S$, stride *s* and kernel number 3S, where S is a hyperparameter. The final output of the patch extraction layer is resized to $P \times C$, where the value of P is $\frac{L}{S} \times \frac{L}{S}$, and the value of C is 3S. Then, the linear embedding layer transforms the channel C into a given dimension D. Thus, after the patch extraction and patch embedding layers, the size of the image will be transformed to $P \times D$ and can be handled by self-attention.

3.5 Patch merging

Similar to the patch embedding layer, the patch merging layer subsamples the output of Swin transformer layer. It halves the input size from $h \times w$ to $\frac{h}{2} \times \frac{w}{2}$ and doubles the dimension from *D* to *2D.* This process is just like the pooling operation in a convolutional neural network, which increases the receptive filed in the deeper layers.

4 EXPERIMENTS

4.1 Dataset

Here, we evaluate the performance of the proposed framework on Animal dataset [6] and MPEG-7 [23] dataset. As shown in Figure 3, Animal dataset contains 20 kinds of different animal shapes, and with 100 images for each animal. We utilize some approaches such as affine transformation to augment the dataset. As a result, the final dataset after data augmentation has 10,000 images. As shown in Figure 4, MPEG-7 contains 70 common shapes with 20 images for each shape. We also perform data augmentation and it has 7,000 images after the augmentation. For both datasets, half of the data were used for training and the other half were used for testing.

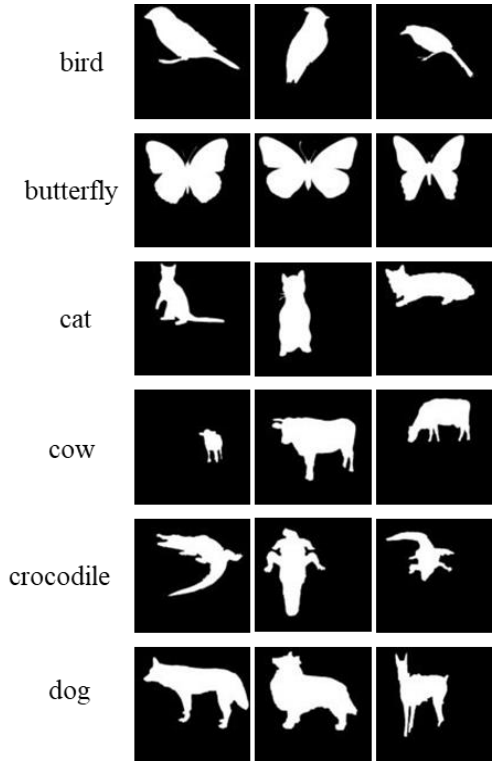

Figure 3: Examples of Animal dataset.

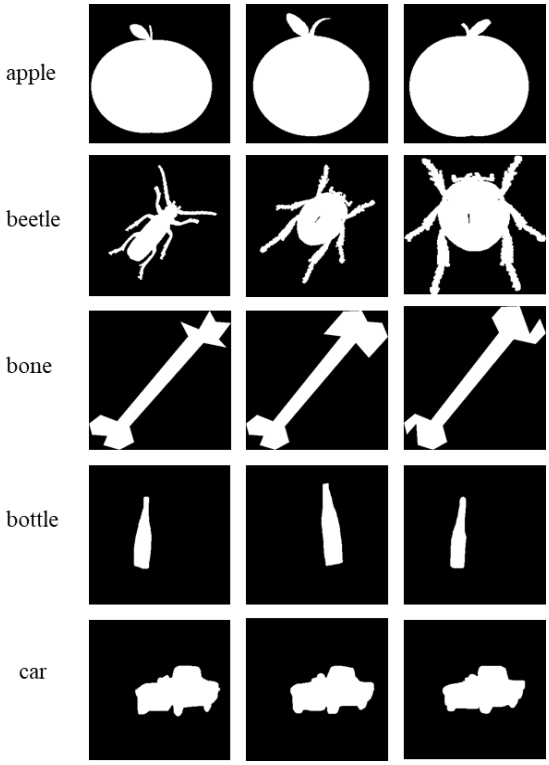

Figure 4: Examples of MPEG-7 dataset.

4.2 Model configuration

For the animal dataset, we set D in Figure 1 to 64 and set 1024 nodes in the last two fully connected layers. In this case, the total parameters of the model are 9 M. For the MPEG-7 dataset, we set D to 96 and set 2048 nodes in the last two layers. In this case, the total parameter is 19.8 M. Dropout is utilized before the last fully connected layers with a drop rate of 0.5, and an early stop is also used. All experiments are conducted on a computer with Intel i9-10900k processor and a RTX 3090 graphic card. The program codes are written by Python 3.7, which is available on https://github.com/wobenbi/Shape_Recg.

4.3 Result and discussion

Table 1 compares the results of the different methods on the animal dataset. Our structure achieves the most advanced results with an accuracy of 93.82%, which is the highest compared to other methods. Table 2 shows the results for the MPEG-7 dataset. Our structure obtained 98.02% accuracy, which is 1% lower than the highest result, which may be due to the fact that the dataset used for training was not large enough, with only 50 images per category used for training. Our models have 9M and 19.8M parameters, respectively, while the VGG-based approach has more than 138M parameters.

Table 1: Accuracy comparison on Animal dataset

Method	Half-training ANIMAL accuracy
Lee [5]	66.91%
IDSC [2]	73.6%
Skeleton Paths [6]	67.9%
Class segment set [7]	69.7%
Contour Segments [6]	71.7%
ICS [6]	78.4%
Hierarchical Shape Tree [9]	80%
Lim [10]	80.37%
BCF [11]	83.4%
Shape Vocabulary [12]	84.3%
BoSCP-LP [13]	89.77%
FV-based [24]	89.26%
VGG-based [4]	90.02%
Ours	**93.82%**

Table 2: Accuracy comparison on MPEG-7 dataset

Method	Half-training MPEG-7 accuracy
Lee [5]	84.22%
IDSC [2]	85.4%
Skeleton Paths [6]	86.7%
Class segment set [7]	90.9%
Contour Segments [6]	91.1%
Curvature Classification [8]	92.77%
ICS [6]	96.6%
BCF [11]	97.16%
BoSCP-LP [13]	98.72%
FV-based [24]	98.77%
VGG-based [4]	99.09%
Ours	**98.02%**

5 CONCLUSION

In this paper, a two-stream neural network model was developed to classify shapes using the original image and its all-scale representation. In addition, to improve the capability of the model, a shifted-window hierarchical vision transformer structure was utilized. We evaluate our model using both animal dataset and MPEG-7 dataset. Experimental results demonstrate that the proposed model has a superior performance on shape recognition tasks. It obtains a recognition rate of 93.82% on the animal dataset, which is the highest so far, and only uses

9M parameters. At the same time, we observed an overfitting problem on both datasets, so one of the future works is to reduce the overfitting. We utilize an all-scale representation to help the model and develop an end-to-end model that uses only the original images to improve performance. In such a model, the first few layers should do some transformations and extract more information, as the all-scale representation does.

6 REFERENCE

[1] Belongie, Serge, Jitendra Malik, and Jan Puzicha. "Shape matching and object recognition using shape contexts." IEEE transactions on pattern analysis and machine intelligence 24.4 (2002): 509-522.

[2] Ling, Haibin, and David W. Jacobs. "Shape classification using the inner-distance." IEEE transactions on pattern analysis and machine intelligence 29.2 (2007): 286-299.

[3] Wang, Junwei, et al. "Shape matching and classification using height functions." Pattern Recognition Letters 33.2 (2012): 134-143.

[4] Min Ruipeng，Li Yifan，Huang Yao，Yang Jianyu，Zhong Baojiang，Visualized all-scale shape representation and recognition，Journal of image and Graphics，pp. 1-14, 2021.

[5] Lee, Sue Han, et al. "How deep learning extracts and learns leaf features for plant classification." Pattern Recognition 71 (2017): 1-13.

[6] Bai, Xiang, Wenyu Liu, and Zhuowen Tu. "Integrating contour and skeleton for shape classification." 2009 IEEE 12th international conference on computer vision workshops, ICCV workshops. IEEE, 2009.

[7] Latecki, Longin Jan, Rolf Lakamper, and T. Eckhardt. "Shape descriptors for non-rigid shapes with a single closed contour." Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). Vol. 1. IEEE, 2000.

[8] Jia Q, Yu M Y, Fan x, et. al. Shape coding and recognition method based on curvature classification. Chinese Journal of Computers, 2018, 41(11): 35 – 48.

[9] Li, Y., J. Zhu, and F. L. Li. "A hierarchical shape tree for shape classification." 2010 25th International Conference of Image and Vision Computing New Zealand. IEEE, 2010.

[10] Lim, Kart-Leong, and Hamed Kiani Galoogahi. "Shape classification using local and global features." 2010 Fourth Pacific-Rim Symposium on Image and Video Technology. IEEE, 2010.

[11] Wang, Xinggang, et al. "Bag of contour fragments for robust shape classification." Pattern Recognition 47.6 (2014): 2116-2125.

[12] Bai, Xiang, Cong Rao, and Xinggang Wang. "Shape vocabulary: A robust and efficient shape representation for shape matching." IEEE Transactions on Image Processing 23.9 (2014): 3935-3949.

[13] Shen, Wei, et al. "Bag of shape features with a learned pooling function for shape recognition." Pattern Recognition Letters 106 (2018): 33-40.

[14] Alajlan, Naif, et al. "Shape retrieval using triangle-area representation and dynamic space warping." Pattern recognition 40.7 (2007): 1911-1920.

[15] Müller, Meinard. Information retrieval for music and motion. Vol. 2. Heidelberg: Springer, 2007.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” arXiv preprint arXiv:1706.03762, 2017.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020.

[18] C.-F. Chen, Q. Fan, and R. Panda, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” arXiv preprint arXiv:2103.14899, 2021.

[19] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” arXiv preprint arXiv:2103.14030, 2021.

[20] C.-F. Chen, Q. Fan, and R. Panda, “Crossvit: Cross-attention multiscale vision transformer for image classification,” arXiv preprint arXiv:2103.14899, 2021.

[21] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation ´ through attention,” arXiv preprint arXiv:2012.12877, 2020.

[22] Touvron, Hugo, et al. "Going deeper with image transformers." arXiv preprint arXiv:2103.17239 (2021).

[23] Latecki, Longin Jan, Rolf Lakamper, and T. Eckhardt. "Shape descriptors for non-rigid shapes with a single closed contour." Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). Vol. 1. IEEE, 2000.

[24] Yang C, Fang L, Wei H. 2020. Learning contour-based mid-level representation for shape classification. Access, 99: 1-1 [DOI: 10.1109/ACCESS.2020.3019800]

[25] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).

Authors' background

Your Name	Title*	Research Field	Personal website
Tianyang Gu	**Bachelor student**	**Neural network and machine learning**	
Ruipeng Min	**Bachelor student**	**Partten recognition and image processing**	