International Journal of Innovative
Computing, Information and Control
Volume **20**, Number **6**, December **2024**

ICIC International ©2024 ISSN 1349-4198
pp. 1775–1788

DESIGN AND RESEARCH OF A MULTI-VIEW GRAPH DEEP LEARNING 3D MODEL RETRIEVAL SYSTEM BASED ON FUSION VISION-TRANSFORMER

Rong Liang* and Fangping Li

Department of Art and Design
Taiyuan University
No. 7, Fendong Street, Tanghuai Industrial Park, Taiyuan 030000, P. R. China
2012050019@tyu.edu.cn
*Corresponding author: 2014010017@tyu.edu.cn

Received December 2023; revised April 2024

Abstract. *The development of computer vision has made three-dimensional models play a crucial role in the field of image processing. However, compared to 2D models, 3D models have more features, making it difficult to extract features and mine correlation information between features. Based on this, this study is based on a multi-perspective graph convolutional neural network, which uses an image entropy weight pooling layer to improve the original view pooling layer. It assigns a weight based on image entropy to each perspective image, and then performs view pooling operations. The Vision-Transformer module is embedded into a multi-perspective graph convolutional neural network to mine information associations between multi-view graphs. The results show that the multi-perspective graph convolutional neural network model fused with Vision-Transformer is more concentrated in classifying features of the same category in the view graph, and there is a significant distance difference between different features. The multi-perspective graph convolutional neural network model fused with Vision-Transformer achieves accuracy of 89.0%, 92.0%, 94.0%, and mean average precision values of 80.0%, 85.0%, and 88.0% when the number of view images is 6, 10, and 14. This study improves the retrieval accuracy of 3D models and has certain reference value in the field of computer vision.*
Keywords: Vision-Transformer, Multi-perspective graph convolutional neural network, 3D, Perspective image, Image entropy

1. **Introduction.** 3D model retrieval refers to the retrieval of 3D models that are similar or related to the query model from a given 3D model database through computer algorithms and techniques [1]. 3D model retrieval has important application value in fields such as computer vision and computer-aided design. The traditional 3D model retrieval methods are mainly based on manually designed feature descriptors, such as shape descriptors, statistical features, and local features [2]. However, these methods have poor retrieval performance when faced with complex model shapes and affine transformations. In recent years, researchers have begun to use deep learning algorithms for 3D model retrieval to address this issue. By constructing a deep neural network model, richer feature representations can be learned directly from the original data of the model [3]. A commonly used method is to represent a 3D model as a point cloud or voxel form, and extract and learn its features through convolutional neural network (CNN) [4]. In addition, there are also some methods that attempt to represent 3D models as graphical structures and use graph convolutional networks for feature extraction and learning [5]. The current algorithms focus more on the feature extraction of a single view graph, and it is difficult to

DOI: 10.24507/ijicic.20.06.1775

mine the correlation between multi-view graphs. To solve this problem, this study designs a retrieval algorithm by combining multi-view CNN and Vision-Transformer (MVVCNN) to deeply excavate the information correlation between multi-view images, so as to make the retrieval of 3D models more accurate.

The contributions of this paper are as follows. 1) It improves the 3D model retrieval algorithm based on multi-view CNN (MVCNN), improves the original maximum perspective pooling to a weight pooling method based on image entropy, and fuses the information of multi-view graphs more efficiently. 2) A novel 3D model retrieval algorithm based on Vision-Transformer (ViT) is proposed. The two-level ViT is embedded in the MVCNN graphs to mine the information correlation between multi-view graphs.

The innovations of this study are as follows. A 3D model retrieval system with B/S architecture is designed based on MVVCNN. A multi-view graph generated from the 3D model is reduced by virtual camera to construct 2D image data for representing the 3D model, which is then sent into the multi-view graph retrieval model for training and similarity measurement. Finally, the search result is obtained.

This study is divided into five parts. The first part introduces the background and significance of this study. The second part discusses the existing research on CNN and 3D model retrieval. The third part is based on MVCNN, using Entropy Weighted View (EWV) pooling layer to improve the original view pooling layer. ViT module is introduced to construct a 3D model retrieval system. In the fourth part, the model is tested and analyzed. The fifth part is a summary of the full text, and points out the shortcomings of this paper.

2. **Related Works.** CNNs are broadly applied in computer vision, and some experts and scholars have conducted relevant research based on this. Wang et al. believed that CNN, as a mainstream deep learning model, has achieved great success in the field of image recognition. However, the neurons used in CNN were too simplified. To raise the learning ability of the model, a new dendritic CNN was raised. This network model considered the nonlinear information processing function of dendrites in a single neuron, and its superiority was demonstrated through experiments [6]. Vives-Boix and Ruiz-Fernández believed that synaptic plasticity affected the performance of models in memory and learning in neural networks. Therefore, a method was provided to incorporate partial plasticity into CNN to strengthen learning in image classification issues. This method involved using partial plasticity as a weight update function during the backpropagation phase of the convolutional layer. The outcomes denoted that this method could effectively enhance the image classification learning of the model [7]. Shi et al. found that existing physics based methods could not control the focus of each pixel in 3D projection. Therefore, a CGH pipeline based on deep learning was proposed, which could synthesize realistic color 3D holograms in real time from a single RGB depth image. Moreover, CNN has extremely high memory efficiency, running at a resolution of 60 Hz on a single consumer level graphics processing unit, with a resolution of 1920×1080 pixels. The experiment findings expressed that this method could present the projection effect of holograms [8]. Joshi et al. proposed an efficient deep learning architecture-based peripheral blood cell image recognition and classification using interruption-based salp swarm and cat-based optimized CNN algorithms. At the same time, a binary encoding technique was developed to transform the parameter adjustment problem into an optimization problem. This method increased the diversity of the search space by providing higher classification accuracy. The experimental results showed that the global classification accuracy of this method was 97% [9].

Any design industry cannot do without 3D design, and 3D model data is increasing year by year, making it difficult to search. Some experts and scholars have conducted relevant research on the retrieval methods of 3D models. Hamza et al. proposed a 3D model shape retrieval system based on triangular meshes. This method first extracted the features of the 3D model to calculate its descriptor, and then divided the model into clusters based on descriptors that remain unchanged in scale and direction. It clustered the model using the fuzzy C-means clustering method. The results demonstrated the superior performance of this method [10]. Cheng et al. proposed a rapid design method for process equipment driven by classification retrieval based on 3D model definition. Firstly, a 3D model of information integration was established. A classification machine learning algorithm was constructed based on the definition of a 3D model using an extreme learning machine. Finally, the processing equipment for retrieving and mapping the 3D model definition was called, and the existing process equipment model was adjusted and modified. The outcomes denoted that this method could improve the efficiency of design [11]. Nie et al. proposed a novel multi-modal fusion network for 3D shape recognition, which utilized the correlation between different modalities to generate more robust fusion descriptors. Moreover, two new loss functions were designed to help the model learn relevant information during the training. The experimental results showed that this method had significant advantages compared to the most advanced methods [12]. Starly et al. believed that in the field of computer graphics, different datasets were used for 3D shape classification and retrieval, resulting in different categories of model descriptions. Based on this, an algorithm based on the MVCNN algorithm was put forward. This algorithm used the camera angle of the original angle to capture feature details for classification and retrieval, to reduce the amount of data and processing time required for training shape recognition algorithms. Through simulation experiments, it is shown that the algorithm was improved by nearly 6% compared to the original version [13].

In summary, existing research on image recognition using CNN mainly focuses on 2D images, lacking applications in 3D image recognition. In addition, most existing 3D model search methods are limited to image classification and lack research on information association and feature fusion of perspective maps. Therefore, this study is based on an MVCNN and uses an EWV pooling layer to improve the original view pooling layer and achieve information fusion of multi-view graphs. Then, it embeds the ViT module into the MVCNN to explore the information association between multi-view graphs. The method proposed by this research has high application value in computer vision.

3. Construction of a 3D Model for Deep Learning from Multi-View Fused with ViTs.

To achieve the retrieval of 3D models, this chapter is divided into two parts to construct the model. The first part is based on an MVCNN, which uses the EWV pooling layer to improve and achieve the fusion of multi-view graph information. The second part embeds the ViT module into an MVCNN based on EWV, achieving the mining of information associations between multi-view graphs.

3.1. The construction of improved MVCNN models.

A 3D model is a virtual representation form used to present the appearance and structure of objects in 3D space, consisting of a set of geometric data [14]. To batch generate multi-view graphs of 3D models in the Blender mapping system, it is necessary to first generate a blank model file in the Blender system. In this model file, environmental features such as background, texture, and lighting are constructed in advance. At the same time, to avoid the impact of light and shadow changes on the rendering of the perspective map when importing a 3D model, a Phong style lighting model is used to complete the rendering of the perspective

map. Phong style lighting model [15] is a commonly used algorithm in computer graphics for simulating lighting effects. The Phong style lighting model mainly considers ambient light, diffuse light, and specular light. It determines the color of each point on the model surface by calculating the direction and intensity of light. The calculation for the Phong type lighting model is shown in Equation (1).

$$I = I_{pa}k_a + \sum (I_{pd}k_d \cos i + I_{ps}k_s \cos^n \theta) \tag{1}$$

In Equation (1), I represents the intensity of lighting in the 3D model. k_a represents the environmental reflection coefficient. k_d represents the diffuse reflection coefficient. k_s represents the specular reflection coefficient. I_{pa} represents the intensity of ambient light. I_{pd} represents the intensity of diffuse reflection. I_{ps} represents the intensity of specular reflection. Due to different 3D models having different perspectives, the generated perspective images contain different information. Therefore, when choosing a perspective, the omnidirectional nature of the perspective image should be considered. Therefore, using the center of the 3D model as the origin of the Cartesian coordinate system, it constructs a cube that is larger in length, width, and height than the 3D model to surround the model. The length, width, and height of the cube can be adjusted according to the size of different models, and the coordinate axis divides the space into 8 quadrants. It sets up a virtual camera at the center of all faces of the constructed cube, as well as at the vertices of the 8 quadrants. The perspective of the virtual camera is aligned with the origin of the Cartesian coordinate system. Therefore, each 3D model can obtain 14 different perspective images, which can contain comprehensive information of the 3D model. After obtaining the perspective image, the MVCNN is used to extract features [16]. The architecture diagram of MVCNN is shown in Figure 1.

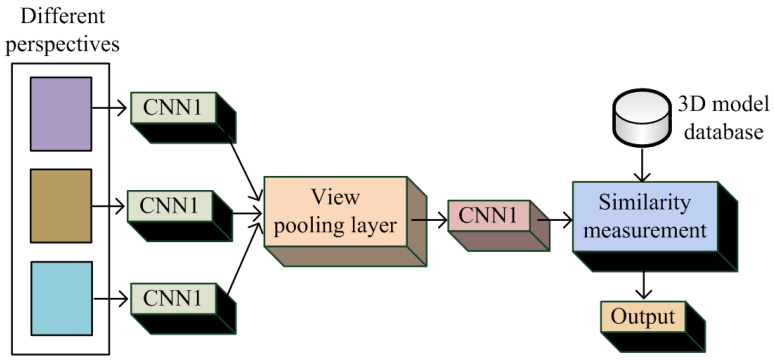
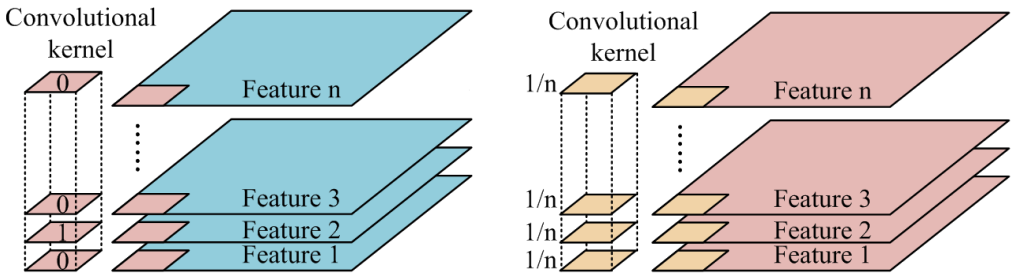


FIGURE 1. Architecture diagram of MVCNN

The MVCNN consists of a first segment CNN, a view pooling layer, and a second segment CNN. When each perspective image of a 3D model independently passes through the first CNN, a feature is generated [17]. All features generated from perspective images are fused into one feature after passing through the view pooling layer. The fused features are input into the second CNN to generate the final 3D model features [18]. The view pooling layer includes maximum pooling and mean pooling methods, but unlike the original pooling layer, the original pooling layer is a planar pooling method, with the convolutional kernel moving along the X and Y axes. The view pooling layer is a 3D pooling method, with its convolutional kernel moving along the Z-axis, which can stack the feature matrices of N perspective images [19]. The maximum pooling and average pooling operations in the view pooling layer are shown in Figure 2.

It assumes that the output of the view pooling layer is VP, and the total number of view graph features is N. The formulas for maximum and average view pooling operations



(a) Maximum view pooling (b) Mean view pooling

FIGURE 2. Maximum pooling and mean pooling operations

are shown in Equation (2).

$$\begin{cases} VP(j) = \max\limits_{1 \le i \le N} (f_i(j)) \\ VP(j) = \dfrac{1}{N} \sum\limits_{i=1}^{N} f_i(j) \end{cases} \tag{2}$$

In Equation (2), the first and second formulas represent maximum and mean view pooling, respectively. j represents the dimension of the feature in the perspective image, while f_i represents the feature corresponding to the ith perspective image. Due to the different angles of the perspective image during shooting, each perspective image contains different amounts of information, resulting in varying degrees of importance in feature extraction. Therefore, the maximum view pooling operation has been improved to perform weight pooling based on image entropy. Image entropy is an indicator used to measure the richness of image information, representing the distribution of pixels in an image [20]. The 1D entropy of an image refers to the amount of information contained in the aggregated features of the grayscale distribution in the image, as defined in Equation (3).

$$H = \sum_{i=0}^{255} p_i \log p_i \tag{3}$$

In Equation (3), p_i means the proportion of pixels with a grayscale value of i in the image. The grayscale value ranges from 0 to 255. Due to the inability of 1D entropy to represent the spatial characteristics of image grayscale distribution, the 2D entropy method of images is adopted. Assuming the scale of the image is N, it defines a binary (i, j). Among them, i stands for the grayscale value of the pixel, and j represents the domain grayscale mean of the image. The frequency at which a binary group appears is defined as $f(i, j)$, and the comprehensive feature expression combining the aggregation features of the image grayscale distribution and spatial features is shown in Equation (4).

$$P_{ij} = \frac{f(i, j)}{N^2} \tag{4}$$

In Equation (4), P_{ij} represents the comprehensive feature. The 2D entropy calculation for images is shown in Equation (5).

$$H = \sum_{i=0}^{255} P_{ij} \log P_{ij} \tag{5}$$

In Equation (5), H represents the 2D entropy of the image. The higher the image entropy, the more evenly distributed the pixels in the image, and the more information the image has. On the contrary, the lower the image entropy, the more concentrated the pixel

distribution in the image, and the less information image has, which may be uniform areas or edges. Therefore, it is necessary to assign higher weights to perspective images with high image entropy, and lower weights to perspective images with low image entropy. The calculated weights form a convolutional kernel and perform view pooling fusion for each perspective image. The improved view pooling module is the EWV pooling layer [21]. The schematic diagram is shown in Figure 3.

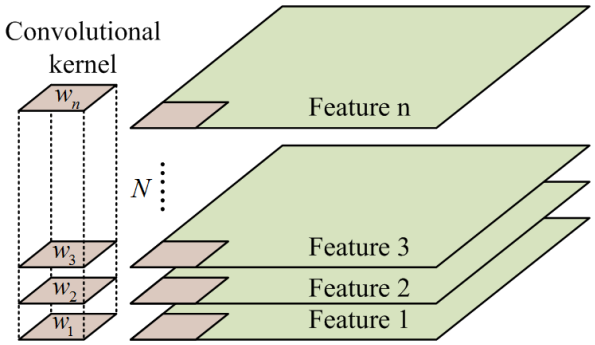


FIGURE 3. Schematic diagram of EWV pooling layer

The output of the EWV pooling layer is shown in Equation (6).

$$VP(j) = \sum_{i=1}^{N} w_i f_i(j) = \sum_{i=1}^{N} \frac{H_i}{\sum_{i=1}^{N} H_i} f_i(j) \tag{6}$$

In Equation (6), w_i represents the weight assigned to the i viewpoint map, and H_i represents the image entropy of the i viewpoint map.

3.2. **Construction of MVVCNN.** In the Transformer module, self attention mechanism, multi start attention (MSA), and position encoding are the main three parts. The attention mechanism can focus on key information and remove secondary information [22]. The self attention mechanism is more suitable for analyzing the internal correlations between input features, reducing dependence on external information [23]. The self attention mechanism first multiplies the lowest level input sequence with their respective weight matrices to obtain the query vector Q, key vector K, and value vector V. The calculation for the three vectors is shown in Equation (7).

$$\begin{cases} Q = W_q I \\ K = W_k I \\ V = W_v I \end{cases} \tag{7}$$

In Equation (7), I represents the input sequence, and W refers to the corresponding weight matrix. It calculates the similarity between the query vector Q and the key vector K. It multiplies each Q with all K points to obtain the correlation coefficient. It uses Softmax to calculate an attention weight matrix composed of weight values between 0 and 1. Then, the weight matrix is weighted to sum the corresponding V values, and the final output weighted by the self attention mechanism is obtained as shown in Equation (8).

$$O(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{8}$$

In Equation (8), O represents the weighted output, and d_k represents the correlation coefficient. To better integrate the information correlation between various perspective

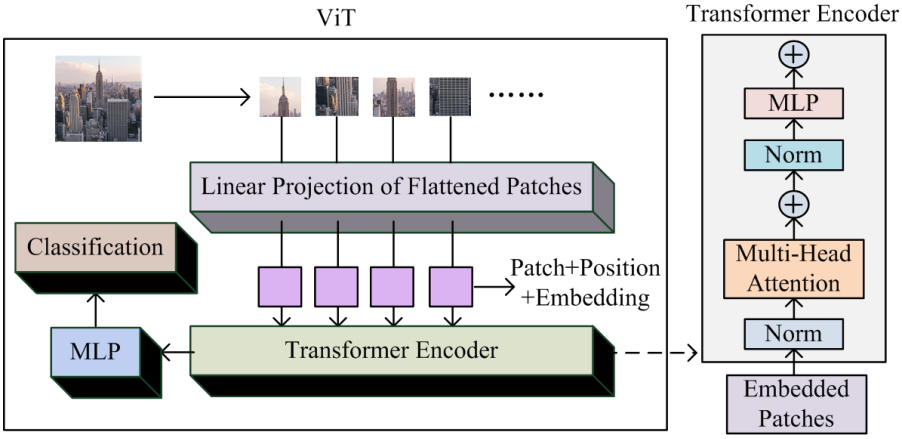


FIGURE 4. ViT framework diagram

images and fuse the features of perspective images, the ViT module [24] is used to integrate the correlation information between multi-view graphs. The architecture diagram of the ViT module is denoted in Figure 4.

When ViT processes 2D image information, it first divides the image into fixed sized small blocks, treating each block as a patch. That is to flatten the input image $x \in R^{H\times W\times C}$ into $x \in R^{N\times\left(P^2\times C\right)}$. P denotes the size of the patch. N expresses the length of the input sequence. Then it needs to flatten the image information, position information, and image category information of the patch into a single vector and use it as input. Assuming that each patch is a tensor of $d_1 \times d_2 \times d_3$, the vectorized result vector is $d_1 d_2 d_3 \times 1$. If the image is divided into n blocks, the vector is expressed as $[x_1, x_2, x_3, \ldots, x_n]$. A fully connected layer is utilized to linearly transform the vector $[x_1, x_2, x_3, \ldots, x_n]$, and then a new vector $[z_1, z_2, z_3, \ldots, z_n]$ is obtained. It encodes each position in the image into a position vector with the same size as the z vector. It adds the position vector to the z vector, where the z vector contains the content and position information of the patch. In addition, feature classification is represented by CLS and an embedding operation is used to obtain a vector z_0 of the same size as z. The above operation is shown in Equation (9).

$$z = [CLS; x_1E; x_2E; \ldots; x_nE] + E_{pos},\ E \in R^{D\times\left(P^2\cdot C\right)},\ E_{pos} \in R^{D\times(N+1)} \tag{9}$$

MSA layers and fully connected layers are alternately stacked to form an encoder, and a total of $n+1$ vectors from z_0 to z_n are input into the encoder to obtain the output vectors c_0 to c_n. The output vector c_0 to c_n is represented as the extracted feature vector from the image, and finally input c_0 into the Softmax classifier. The output result of the classifier represents the classification result vector, represented by p, and the size of p is represented by the number of categories. The schematic diagram of the ViT workflow is denoted in Figure 5.

In traditional Transformers, both encoders and decoders are included, but the decoder has been removed from the ViT framework, leaving only the encoder to classify and retrieve images [25,26]. In ViT, it consists of alternating MSA modules and multi-layer perceptron (MLP). The MSA expression is shown in Equation (10).

$$z'_l = MSA(LN(z_l)) + z_{l-1}, \quad l \in 1, 2, \ldots, L \tag{10}$$

In the above equation, L represents the number of layers of the encoder, and LN represents the layer standardization function. The expression of MLP is shown in Equation (11).

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l \in 1, 2, \ldots, L \tag{11}$$

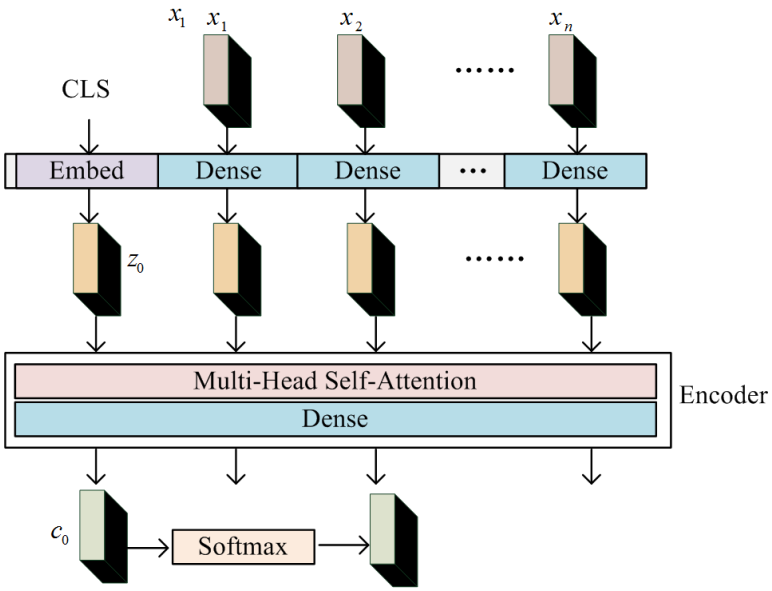


FIGURE 5. Schematic diagram of ViT workflow

In addition to MSA and MLP, layer standardization and skip connections are also added at the beginning and end of each encoder, as shown in Equation (12).

$$y = LN\left(z_L^0\right) \tag{12}$$

To achieve the fusion of associated information from multi-view images, improvements were made on the Transformer module and MVCNN network to construct an MVVCNN model. The MVVCNN architecture diagram is shown in Figure 6.

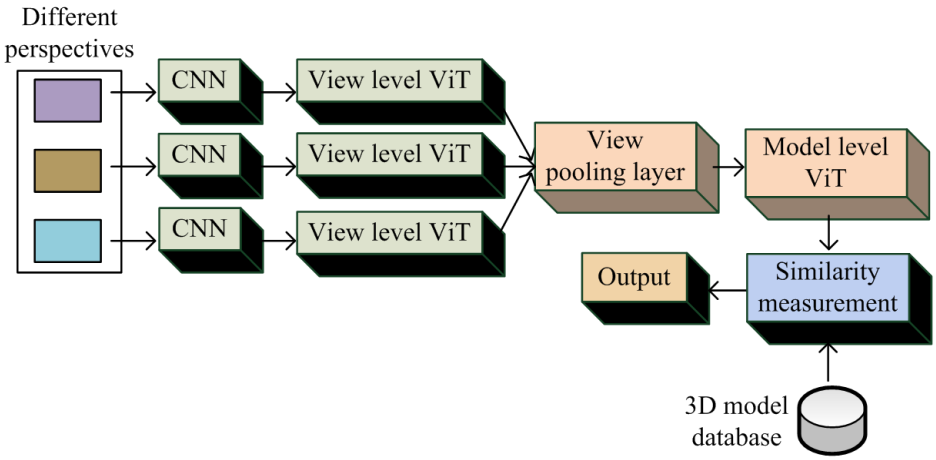


FIGURE 6. MVVCNN architecture diagram

In the MVVCNN, after passing through the CNN module, the feature matrix of each perspective image is independently extracted through the view level ViT layer. In this module, the [6,6, 256] matrix is flattened to [36, 256] and the position of each patch is decoded. It adds the position vector to the feature vector, and the synthesized new vector contains the content and position information of the patch. After inputting the new feature vectors into the view level ViT module, 14 256-dimensional view features are generated. These 14 features are input into the model level ViT after passing through the EWV module. In the original ViT, the position vector is simply encoded. Based on the

characteristics of Transformer position encoding, a position encoding method based on virtual camera position was constructed and applied to model level ViT. This encoding method can obtain the rendered perspective map of the virtual camera based on its absolute position in the coordinate system, and encode the position based on the perspective images. A 3D Cartesian coordinate system is established based on the origin of the 3D model, with the number of cameras represented by C. The position of the camera is shown in Equation (13).

$$X = R^{C\times 3} \tag{13}$$

In Equation (13), X represents the 3D vector of each camera. The position of each perspective image is encoded into a vector embedded in Equation (14).

$$E_{pos} = R^{3\times D} \tag{14}$$

By using XE_{pos}, the feature encoding vector for each position can be obtained, and the size of this vector is consistent with the weight pooling output of the image entropy. Finally, the feature vector and position encoding vector are added to obtain the final feature vector. It measures the similarity between the feature vector and the features in the feature library using Euclidean distance one by one, and the calculation is shown in Equation (15).

$$d(x,y) = \sqrt{\sum_{i=1}^{256} (x_i - y_i)^2} \tag{15}$$

The calculated Euclidean distance values are sorted from smallest to largest, and the smaller the value, the higher the similarity between the two features.

4. Model Performance Testing and Analysis.

To study the retrieval performance of the constructed model, this chapter is divided into two parts for testing. The first part will compare and analyze the models before and after improvement, and the second part will compare and test the MVVCNN model with other existing models.

4.1. Performance testing of MVVCNN.

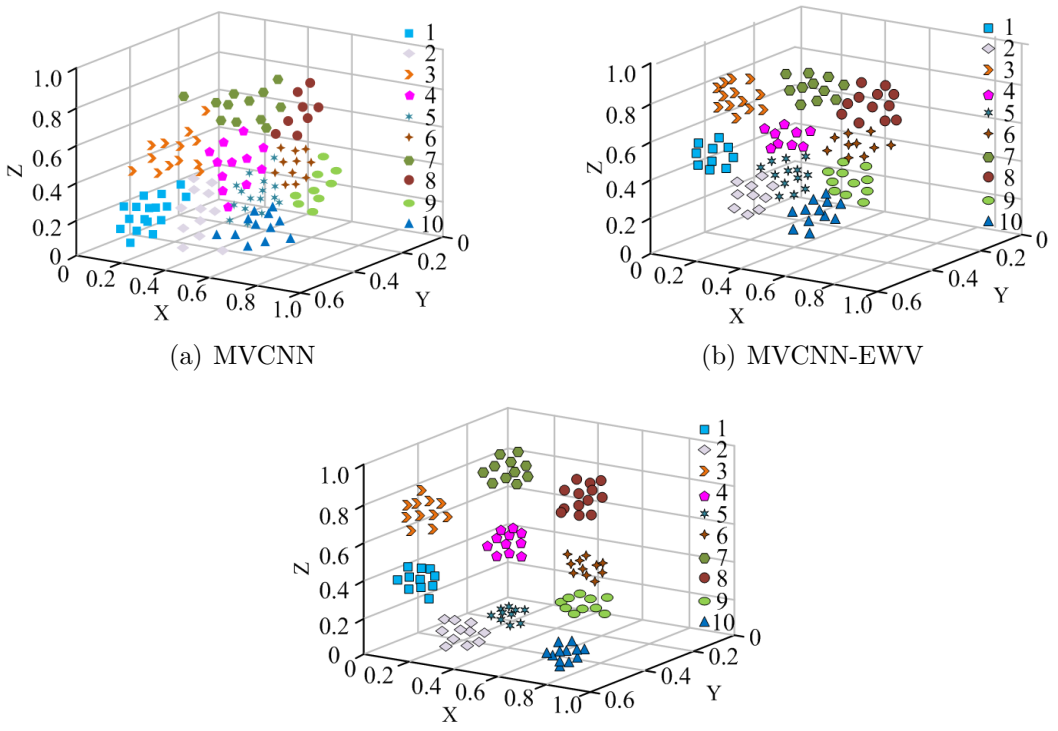
Princeton University's ModelNet dataset was selected for the study. There were 4899 models in ModelNet10, including 10 categories, 3991 of which were training sets and 908 of which were data sets. ModelNet40 had a total of 12291 3D models, including 40 categories, of which 9843 models were training sets and 2448 models were test sets. The experiments in this study were conducted on a computer with 32GB memory, an Intel(R) Xeon processor and an NVIDA Quadro P620 graphics card under Windows operating system. The code of the experiment was based on Tensorflow framework, and Adam optimization method was used for network training. The initial learning rate of 0.001 was set first, the exponential attenuation was maintained, and the learning rate was changed to 0.0001 when the loss was not significantly reduced. The other parameter settings and experimental environment settings are shown in Table 1.

The feature extraction effects of MVCNN, MVCNN-EWV, and MVVCNN models were compared and ModelNet10 was selected as the dataset. The dataset included 10 features, and the visualization results obtained after PCA dimensionality reduction are shown in Figure 7.

From Figure 7, the features extracted by the original MVCNN model could achieve clustering, but there was no significant distance between feature categories. The MVCNN-EWV model had a more significant clustering effect for features of the same category, and the distance between categories also increased. The MVVCNN model had more concentrated features of the same category, and there were obvious differences between different

TABLE 1. Experimental environment and parameter settings

Parameter and environment settings	Parameters and environment	Setting
Experimental environment	Operating system	Windows10 Enterprise
	Processor	Intel(R)Xeon E-2244G CPU@3.80GHz 3.79GHz
	Memory	32GB
	Graphics card	NVIDA Quadro P620
	Programming language	Python 3.6.7
	Development editor	PyChrm 2019.3.5
	Deep learning framework	TensorFlow1.14.0
	Image processing library	OpenCV4.4.0
Parameter	Initial learning rate	0.001
	Exponential decay rate	0.9
	Fuzzy factor	10-8
	Batch training volume	16
	Epoch	20

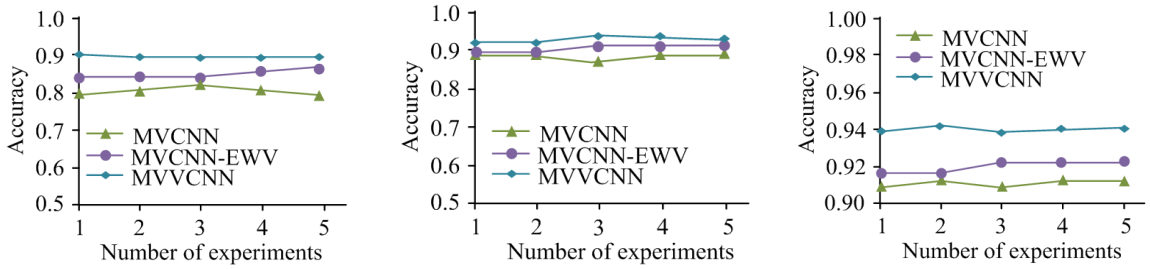


(a) MVCNN (b) MVCNN-EWV

(c) MVVCNN

FIGURE 7. Visualization of feature extraction

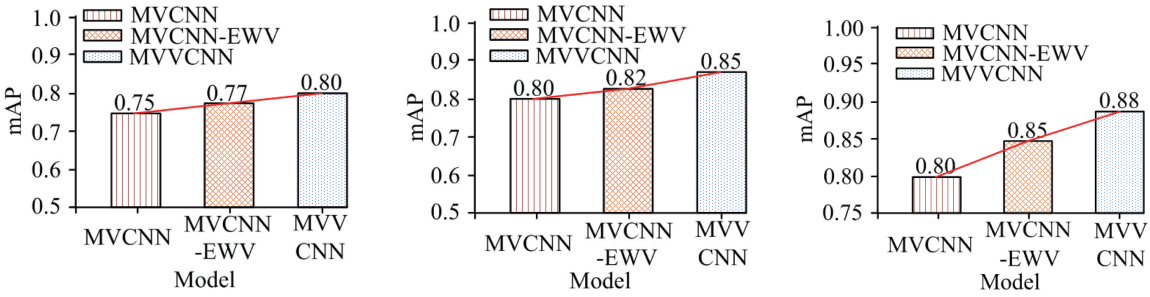
features. Therefore, the MVVCNN model could effectively distinguish the features of 3D models and had better feature extraction performance. The accuracy of classification was set as the testing indicator, and the number of perspective images was set to 6, 10, and 14. After conducting 5 experiments, the accuracy comparison results of the three models are shown in Figure 8.



(a) The perspective image is 6. (b) The perspective image is 10. (c) The perspective image is 14.

FIGURE 8. Accuracy comparison chart

From Figure 8, when the number of perspective images was 6, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 82.0%, 85.0%, and 89.0%, respectively. When the number of perspective images was 10, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 88.0%, 90.0%, and 92.0%, respectively. When the number of perspective images was 14, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 91.0%, 92.0%, and 94.0%, respectively. Therefore, as the number of perspective images increased, the accuracy would improve. The MVVCNN model had higher accuracy values compared to the other two models. The retrieval accuracy mean average precision (mAP) value was used as an indicator, and the outcomes are indicated in Figure 9.



(a) The perspective image is 6. (b) The perspective image is 10. (c) The perspective image is 14.

FIGURE 9. Comparison of mAP

From Figure 9, when the number of perspective images was 6, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 75.0%, 77.0%, and 80.0%, respectively. When the number of perspective images was 10, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 80.0%, 82.0%, and 85.0%, respectively. When the number of perspective images was 14, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 80.0%, 85.0%, and 88.0%, respectively. Therefore, MVVCNN had higher retrieval accuracy values and better performance in retrieving 3D models.

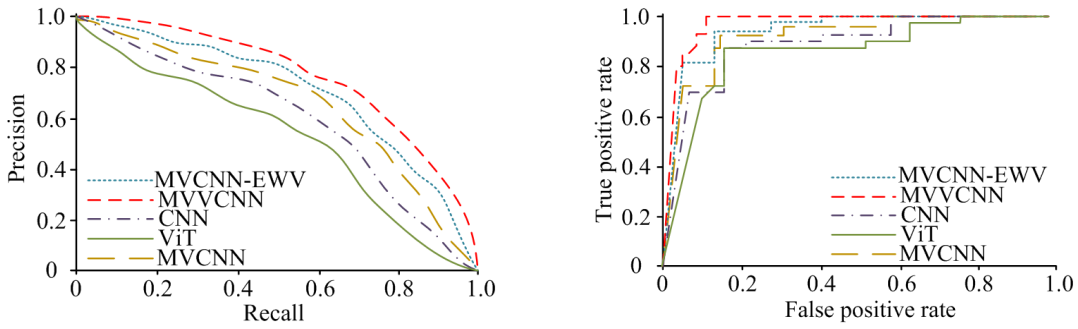
4.2. **Model comparison testing and analysis.** To study the performance differences between the MVVCNN model and existing advanced models, CNN, Spherical Harmonic Function Descriptor (SPH), LightField Function Descriptor (LFD), ViT, and MVVCNN were selected for comparison. CNN can be trained to recognize shapes independently in a rendered view. SPH is the angle part of the spherical coordinate solution of Laplace's equation. It is a famous function in modern mathematics. It is widely used in quantum mechanics, computer graphics, rendering light processing and spherical mapping. Another example that is particularly popular in computer graphics settings is LFD, which extracts

a set of geometric and Fourier descriptors from the contours of objects rendered from several different viewpoints. ViT treats images as sequence data and each pixel or region of the image as a position in the sequence, so that spatial and temporal information in the image can be captured using the self attention mechanism. The results of testing on the ModelNet40 dataset are denoted in Table 2.

TABLE 2. Comparison of mAP values of different models

3D retrieval model	Number of perspective images	mAP(%)
CNN	10	62.5
SPH		34.5
LFD		41.3
ViT		78.5
MVVCNN		80.0
CNN	14	65.9
SPH		40.5
LFD		47.6
ViT		82.9
MVVCNN		88.2

From Table 2, when the perspective maps were 10 and 14, the MVVCNN model had mAP values of 80.0% and 88.2%, respectively. The SPH and LFD models were the best existing retrieval models based on shape descriptors, but the results obtained were average. Therefore, deep learning could mine deeper correlation information in perspective images and more accurately represent the features of perspective images. The P-R curves and ROC curves of CNN, ViT, MVCNN, MVCNN-EWV, and MVVCNN were compared, as shown in Figure 10.



(a) P-R curve (b) ROC curve

FIGURE 10. Comparison of P-R curves

From Figure 10, the P-R curve of the MVVCNN model had the largest area enclosed by the coordinate axis. Although the performance of MVCNN-EWV and MVCNN models was not better than that of MVVCNN model, they also achieved better performance compared to CNN and ViT models.

5. **Conclusion.** To improve the accuracy of 3D model retrieval, this study was based on an MVCNN, which uses the EWV pooling layer for improvement. It embedded the ViT module into the improved MVCNN-EWV to construct a 3D image retrieval model. The research findings denoted that the MVCNN model could only realize clustering, and there was no significant distance between feature categories. The MVCNN-EWV model

had a significant clustering effect on features of the same category, and the distance between categories also increased. The MVVCNN model had more concentrated features of the same category, and there were obvious differences between different features. When the number of perspective images was 6, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 82.0%, 85.0%, and 89.0%, respectively. When the number of perspective images was 10, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 88.0%, 90.0%, and 92.0%, respectively. When the number of perspective images was 14, the accuracy of MVCNN, MVCNN-EWV, and MVVCNN was 91.0%, 92.0%, and 94.0%, respectively. When the number of perspective images was 6, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 75.0%, 77.0%, and 80.0%, respectively. When the number of perspective maps was 10, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 80.0%, 82.0%, and 85.0%, respectively. When the number of perspective images was 14, the mAP value of MVCNN, MVCNN-EWV, and MVVCNN was 80.0%, 85.0%, and 88.0%, respectively. The MVVCNN model had mAP values of 80.0% and 88.2%, respectively, when the perspective maps were 10 and 14. Moreover, the P-R curve of the MVVCNN model had the largest area enclosed by the coordinate axis, which could still achieve the best results compared to other models. There were shortcomings in this study, such as a lack of optimization of model retrieval time. Therefore, in future research, the minimum search time can be used as one of the optimization indicators.

REFERENCES

[1] B. Chen, H. Li and W. Luo, Image processing operations identification via convolutional neural network, *Science China (Information Sciences)*, vol.63, no.3, pp.275-281, 2020.

[2] T. Stomaci, F. Buonamici and G. Gelati, 3D-printed models for left atrial appendage occlusion planning: A detailed workflow, *Rapid Prototyping Journal*, vol.29, no.11, pp.74-81, 2023.

[3] M. V. Gendt, T. Besard and S. Vandenberghe, Productively accelerating positron emission tomography image reconstruction on graphics processing units with Julia, *The International Journal of High Performance Computing Applications*, vol.36, no.3, pp.320-336, 2022.

[4] J. Hu, W. Deng and Q. Liu, Constructing an efficient and adaptive learning model for 3D object generation, *IET Image Processing*, vol.15, no.8, pp.1745-1758, 2021.

[5] W. Wu, M. Xu and Q. Liang, Multi-camera 3D ball tracking framework for sports video, *IET Image Processing*, vol.14, no.15, pp.3751-3761, 2020.

[6] R. Wang, Z. Lei, Z. Zhang and S. Gao, Dendritic convolutional neural network, *IEEJ Transactions on Electrical and Electronic Engineering*, vol.17, no.2, pp.302-304, 2022.

[7] V. Vives-Boix and D. Ruiz-Fernández, Synaptic metaplasticity for image processing enhancement in convolutional neural networks, *Neurocomputing*, vol.462, no.4, pp.534-543, 2021.

[8] L. Shi, B. Li, C. Kim and P. Kelllnhofer, Towards real-time photorealistic 3D holography with deep neural networks, *Nature*, vol.591, no.7849, pp.234-239, 2021.

[9] S. Joshi, R. Kumar and A. Dwivedi, Hybrid DSSCS and convolutional neural network for peripheral blood cell recognition system, *IET Image Processing*, vol.14, no.17, pp.4450-4460, 2020.

[10] N. A. Hamza, S. H. Jafer and R. M. Hadi, 3D model retrieval using MeshSIFT descriptor and fuzzy C-means clustering, *Indonesian Journal of Electrical Engineering and Computer Science*, vol.19, no.3, pp.1452-1460, 2020.

[11] Q. Cheng, S. Wang and X. Fang, Intelligent design technology of automobile inspection tool based on 3D MBD model intelligent retrieval, *Proc. of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol.235, nos.10-11, pp.2917-2927, 2021.

[12] W. Nie, Q. Liang, Y. Wang, X. Wei and Y. Su, MMFN: Multimodal information fusion networks for 3D model classification and retrieval, *ACM Transactions on Multimedia Computing Communications and Applications*, vol.16, no.4, pp.1-22, 2020.

[13] B. Starly, A. Angrish and A. Bharadwaj, MVCNN++: CAD model shape classification and retrieval using multi-view convolutional neural networks, *Journal of Computing and Information Science in Engineering*, vol.21, no.1, pp.1-27, 2020.

[14] B. Vivekanandam, Speedy image crowd counting by light weight convolutional neural network, *Journal of Innovative Image Processing*, vol.3, no.3, pp.208-222, 2021.

[15] A. K. Sharadhi, V. Gururaj and S. P. Shankar, Face mask recogniser using image processing and computer vision approach, *Global Transitions Proceedings*, vol.3, no.1, pp.67-73, 2022.
[16] H. Qu, M. Sako and A. Mller, SCONE: Supernova classification with a convolutional neural network, *The Astronomical Journal*, vol.162, no.2, pp.67-75, 2021.
[17] S. Y. Tan, A. Kuganesan and K. Buchan, Iterative model reconstruction in lumbar spine image retrieval from computed tomography of the abdomen and pelvis, *Hong Kong Journal of Radiology*, vol.24, no.1, pp.15-22, 2021.
[18] L. Han, Y. Tong and J. Piao, Non rigid 3D shape partial matching based on deep feature fusion, *Journal of Computer-Aided Design & Computer Graphics*, vol.33, no.3, pp.475-486, 2021.
[19] X. Wang, Application of network protocol improvement and image content search in mathematical calculus 3D modeling video analysis, *AEJ-Alexandria Engineering Journal*, vol.60, no.5, pp.4473-4482, 2021.
[20] A. A. Liu, H. Zhou and W. Nie, Hierarchical multi-view context modelling for 3D object classification and retrieval, *Information Sciences*, vol.547, no.8, pp.984-995, 2021.
[21] M. Kiefer, T. V. Clarmann and B. Funke, IMK/IAA MIPAS temperature retrieval version 8: Nominal measurements, *Atmospheric Measurement Techniques*, vol.14, no.6, pp.4111-4138, 2021.
[22] Z. Y. Zhao, W. Z. Huang and J. Pan, A sparse feature extraction method with elastic net for drug-target interaction identification, *Scientific Programming*, vol.2021, no.44, pp.1-10, 2021.
[23] Y. Fang, B. Luo and T. Zhao, ST-SIGMA: Spatio-temporal semantics and interaction graph aggregation for multi-agent perception and trajectory forecasting, *CAAI Transactions on Intelligence Technology*, vol.7, no.4, pp.744-757, 2022.
[24] Y. Yang and X. Song, Research on face intelligent perception technology integrating deep learning under different illumination intensities, *Journal of Computational and Cognitive Engineering*, vol.1, no.1, pp.32-36, 2022.
[25] Y. Lei, Research on microvideo character perception and recognition based on target detection technology, *Journal of Computational and Cognitive Engineering*, vol.1, no.2, pp.83-87, 2022.
[26] K. R. Ummah, T. Karlita, R. Sigit, E. M. Yuniarno, I K. E. Purnama and M. H. Purnomo, Effect of image pre-processing method on convolutional neural network classification of COVID-19 CT scan images, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1895-1912, 2022.

Author Biography

Rong Liang was graduated with a bachelor's degree of Art Design from Hainan Normal University in 2011 and a master's degree of Art Design from Inner Mongolia Normal University in 2014, mainly engaged in visual communication and theoretical research. She worked at Department of Art and Design of Taiyuan University since 2014, publishing 6 academic papers, 1 academic works, presiding over 1 project, participating 3 projects and gaining 1 appearance patent.

Fangping Li got her bachelor's degree of English from Hengyang Normal University in 2006 and her master's degree of Design and Art from Shanxi University in 2012, mainly engaged in visual communication and theoretical research. She worked at Department of Art and Design of Taiyuan University since 2012, publishing 2 papers indexed by Chinese core database, participating in provincial-level projects, hosting 1 provincial-level first-class course, winning multiple domestic and international awards for personal works, and leading students to win national and provincial-level awards multiple times.