



# Part machining feature recognition based on a deep learning method

Fangwei Ning<sup>1</sup> · Yan Shi<sup>1</sup> · Maolin Cai<sup>1</sup> · Weiqing Xu<sup>1</sup>

Received: 9 February 2021 / Accepted: 2 August 2021 / Published online: 30 September 2021  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Machining feature recognition is a key step in computer-aided process planning to improve the level of design and manufacturing, production efficiency, and competitiveness. Although the traditional feature recognition method using a graph-based approach has advantages in feature logic expression, the calculation process is inefficient. Deep learning is a new technology that can automatically learn complex mapping relationships and high-level data features from a large amount of data. Therefore, this classification technology has been successfully and widely used in various fields. This study examined a three-dimensional convolutional neural network combined with a graph-based approach, taking advantage of deep learning technology and traditional feature recognition methods. First, the convex and concave machining features of a part were determined using an attributed adjacency graph. Then, the machining features were separated using the bounding box method and voxelized. Subsequently, a stretching and zooming method was proposed to obtain the training data. After training, the test and comparison results demonstrated the high accuracy rate of the proposed method and the improvement in recognition efficiency. The proposed method could also identify convex features, which further improved the recognition range.

**Keywords** Deep learning · Feature recognition · STEP · Convolution neural network · CAPP

## Introduction

The rapid progress of economic globalization and information integration has intensified the competition among enterprises. Part manufacturers must now improve their modes of production and technological levels as well as shorten their product manufacturing cycle to enhance their competitiveness. Computer-aided process planning (CAPP) (Krot & Czajka, 2018) bridges the gap between computer-aided design (CAD) and computer-aided manufacturing (CAM) to improve the levels of design and manufacturing, production efficiency, and competitiveness. The Standard for the Exchange of Product (STEP) model realizes a complete product model data describing the whole product cycle, including

design, manufacturing, utilization, maintenance, and scrap. It covers aspects such as geometry, topology information, behavior tolerance, surface roughness, material characteristics, process characteristics, design characteristics, and assembly characteristics (ISO 10303-1, 1994). However, part modeling by CAD requires pure geometric and topological information in steps, which cannot be directly applied to CAPP–CAM systems. Therefore, it is necessary to start from the CAD model of the part and transform the geometric and topological information for the machining of meaningful shapes, that is, machining features; in other words, machining feature recognition is essential. A CAPP system functions as an interpreter between CAM and CAD systems, regardless of whether the CAD output is pure geometric information or design features generated by design feature modeling technology. Feature recognition technology has received increasing attention in academic and industrial circles. New feature recognition methods are constantly emerging, while the scope of feature recognition has also expanded from the initial machining features to detection features, analysis features, etc. After years of development, common part-machining-feature recognition methods include graph-based, logic rules and expert systems, cell-based decomposition, convex hull volumetric decompo-

✉ Yan Shi  
yesoyou@163.com

Fangwei Ning  
nfangwei@163.com

Maolin Cai  
caimaolin@buaa.com.cn

Weiqing Xu  
weiqing.xu@buaa.edu.cn

<sup>1</sup> School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

sition, hint-based, and syntactic pattern approaches. The next section reviews these methods.

### Graph-based approach

Graph-based recognition is one of the most studied methods. This approach was proposed by Joshi in 1987, to implement feature recognition based on the geometric and topological information of parts (Joshi & Chang, 1988; Malyshev, Slyadnev, & Turlapov, 2017; Weise, Benkhardt, & Mostaghim, 2018). It converts a boundary representation (B-rep) model used in entity modelling into an attributed adjacency graph composed of an arc and a node. It uses an attributed adjacency graph (AAG) to represent the boundary model (Joshi & Chang, 1988). In the AAG, the node corresponds to the face of the model, while the arc represents the connection between the faces. An attribute is attached to the arc; if the attribute value on the arc is 0, then a concave connection exists between the two adjacent faces. If the attribute value is 1, then there is a convex connection. This method mainly defines an AAG feature in advance and then searches the AAG to match the subgraph. If the corresponding subgraph is found, then it is identified as the feature. With this method, it is easy to add new feature types. The method also can be combined with feature design and supports feature recognition in various applications, such as machining, design, and analysis (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2020). This approach recognizes independent features very well, and has a high accuracy. However, it is difficult to apply to a negative polyhedron. Venuvinod et al. (Venuvinod & Wong, 1994) made improvements to the approach and proposed a middle axle-attributed adjacency graph (MAAM) for assigning attributes that more precisely describe adjacency relations. For example, if a plane and a curved face form a convex angle (270°), the attribute is '2'. Subsequently, MAAM (Yuen & Venuvinod, 1999) was considered as a 'less expert system and more algorithmic' method for form pattern recognition. However, this approach cannot solve a feature-intersection problem.

Because of the high computational requirements of pattern matching, it is difficult to build a feature template library. Only polyhedral parts can be processed, while interactive features cannot (Hashemi et al., 2020). Moreover, the method of establishing the attribute adjacency matrix through the AAG to match the machining feature subgraph is a nondeterministic polynomial hard (NP-hard) computer problem, which makes the method time-consuming and inefficient (Rai & Vairaktarakis, 2019).

### Convolutional neural network (CNN) recognition approach

With the rapid development of deep learning technology, the idea of solving the complex and varying features of manufactured parts using machine learning has been proposed. Shi, Zhang, Baek, De Backer and Harik (2018) developed a novel feature representation scheme wherein the heat kernel signature (HKS), which is a concise and efficient pointwise shape descriptor, was input into a 2D CNN. The recognition results for interacting features indicated that the HKS feature representation scheme is effective in resolving the boundary loss caused by feature interactions. Zhang, Yang, Zhang and Zhu (2018) proposed FeatureNet, which showed that a 3D CNN could effectively be used to classify the voxel data of features. The watershed method was used to realize feature decomposition; however, there was no definite result for convex features. Zhou, Yang, Zhang, Li and Xiao (2019) and Ghadai, Balu, Sarkar and Krishnamurthy (2018) developed a method of recognizing special-shaped machining features, each of which is represented using multiple drawing views that contain rich information for differentiating each of these features. With these views as a training set, a deep residual network (ResNet) is trained successfully for feature recognition, recognizing manufacturing features from low-level geometric data such as voxels with very high accuracy. An approach and data structure for the automatic recognition of machining features using CNN was proposed by Ma, Zhang and Luo (2018), and a sample library for learning 3D point cloud data was constructed through CAD model transformation and feature sampling. The developed CNN recognition system could recognize 24 types of machined features using sample training and recognition experiments, and the recognition accuracy rate was higher than 95%. Cao, Robinson, Hua, Boussuge, Colligan and Pan (2019) investigated the application of deep learning methods to machining feature recognition in CAD models and presented a concise and informative graph representation for 3D CAD models. Experiments were also performed to evaluate the effectiveness of graph-based deep learning for interacting feature recognition. Shi, Qi, Qin, Scott and Jiang (2020) established a deep learning framework based on a multiple sectional view (MSV) representation named MsvNet for feature recognition. In MsvNet, the MSVs of a 3D model are collected as the input of the deep network, and the information obtained from different views is combined using the neural network for recognition.

In the design of parts, the influence of the mechanical analysis, structural analysis, and part processing modes should be considered, which increases the complexity of the structural form of the parts (Shi et al., 2020). It is difficult to establish a complex reflection of the geometry and part machining features. Because of the limited amount of known geomet-

ric information, calculations must be repeated to obtain more information. As a self-learning technology, deep learning can establish complex mapping relationships and high-level data features from a large amount of data. Therefore, classification models based on deep learning can effectively solve the abovementioned problems and improve the accuracy of part feature recognition.

The contributions of this study are shown in terms of the following. (1) To distinguish the convex and concave machining features of a part from the 3D model, the proposed method for finding the minimum subgraph in an AAG was studied to determine machining features. (2) The machining features were separated using the bounding box method. The remainder of this paper is organized as follows. Section 2 proposes a feature recognition method based on a three-dimensional (3D) CNN and introduces the related progress. Section 3 describes the determination of the machining feature surface of a part, and the machining features of convex and concave bodies are split in the part using an AAG and a bounding box. Section 4 introduces the part voxelization method and shows how training data are obtained using a method for drawing and scaling the part feature data. Section 5 describes how a 3D CNN can be used to recognize features, and how the effectiveness of the feature recognition method is verified using real parts. Section 6 presents the conclusions.

## CNN

### Related work

Deep learning is a method of machine learning based on the representation of data, which can simulate the neural structure of the human brain. The concept of deep learning comes from research on artificial neural networks. An artificial neural network (ANN) abstracts the neural network of the human brain from the perspective of information processing, establishes a simple model, and forms different networks according to different connection modes, which is referred to as neural networks. Therefore, deep learning, also known as deep neural networks, is developed from the previous ANN model of artificial neural networks. As a deep learning application technology, CNNs can automatically extract object features, eliminating manual feature extraction steps (Malhan, Kabir, Shah, & Gupta, 2019). CNNs are a type of neural network that is used for processing data with similar mesh structures. Their main feature is the use of convolution operators. A large number of local features can be extracted and mapped with interest features. Thus, they demonstrate excellent performance in many application fields (Zhang, Yang, Zhang, & Zhu, 2016). The applications of CNNs extend from 2 to 3D models. They have been widely used in object detection and vision research with 3D object models (Gong,

Zhong, Yu, Hu, & Li, 2019). A 3D model is usually expressed by an irregular polygon mesh or a point cloud, and the representation rules are more complex than those of a 2D model. Every pixel in a 2D image is represented by a position coordinate and color value. It is difficult for the polygon mesh or point cloud to describe the internal features. Thus, only the external shape can be expressed.

A voxel (Zhao, Zhang, Zhu, You, Kuang, & Sun, 2019) is the smallest unit of digital data in 3D space segmentation. Wu, Song, Khosla, Yu, Zhang and Tang (2019) proposed a 3D ShapeNet. Through a simple five-layer convolution network, they input  $30^3$  resolution voxel data, with 150,000 3D models divided into 660 categories. Although 3D ShapeNet has a simple structure and low accuracy, many researchers have begun to pay attention to it.

Since then, Maturana and Scherer (2015) used Voxnet to analyze the binary voxel mesh. In contrast to 3D ShapeNet, VoxNet can process different 3D data, including polygon mesh data, depth maps, RGB-D, and point clouds. However, when the resolution of the processed data is improved, the computational overhead will increase, and the low-resolution model recognition and classification accuracy is not very high. However, more importantly, these studies fully proved that CNNs can extract the 3D structural features of an object just as they can process 2D data, which further expands the application scope of CNNs. Based on this, in subsequent research, CNN learning was applied to more forms of 3D data.

### 3D CNN

CNNs are the first successful deep learning algorithms to train multilayer network structures, which are widely employed to solve the problem of learning and extracting deep features from image data. Their basic concept involves adopting the local receptive field of an image as the input of the network, transmitting the information to different layers, and obtaining the significant features of invariance through a digital filter for translation, rotation, and scale transformation. Weight sharing and pooling can significantly reduce the number of model parameters; therefore, a 2D CNN method for extracting deep image features can be extended to become a 3D method, which can be used to extract effective 3D data features. Figure 1 shows a CNN structure consisting of the input, four convolutional layers, two fully connected layers, and the output. The convolution and pooling layers are combined to extract a large number of features layer-by-layer, and the classification is completed in the fully connected layer.

The 2D CNN performs a 2D convolution operation on an image and outputs a 2D image. The 3D CNN performs 3D convolution on 3D data and outputs 3D data. 3D convolution is the stacking multiple consecutive frames to form a cube, and then using the 3D convolution kernel in the cube. As

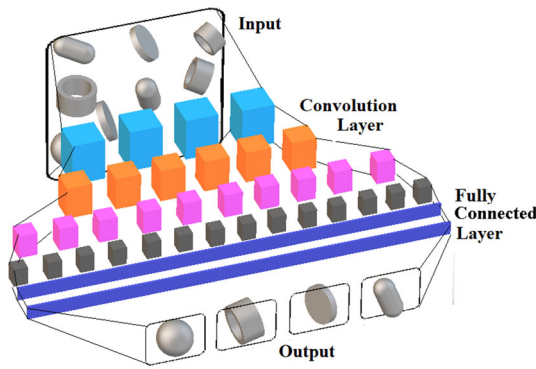


Fig. 1 Structure of 3D CNN

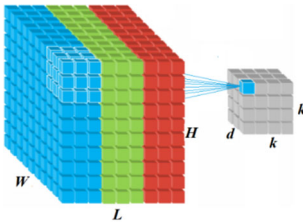


Fig. 2 Diagram of 3D convolution

shown in Fig. 2,  $H$ ,  $W$ , and  $L$  represent the 3D height, width, and length, respectively. The size of the convolution core is  $k \times k \times d (d < L)$ .

Here,  $x^l \in R^{H^l \times W^l \times L^l}$  represents the input layer of the CNN, and  $(i^l, j^l, k^l)$  represents the  $i^l$  th row,  $j^l$  th column, and  $k^l$  th layer in layer  $l$ . The ranges of these values are  $0 \leq i^l \leq H^l$ ,  $0 \leq j^l \leq W^l$ , and  $0 \leq k^l \leq L^l$ , respectively. In addition,  $y$  is a short description of the convolution result of layer  $l$  as  $x^{l+1}$ , that is,  $y = x^{l+1} \in R^{H^{l+1} \times W^{l+1} \times L^{l+1}}$ .

In the convolutional layer, the  $l$ th layer is convoluted with the self-learning convolutional layer  $k_{i,j,k}$ . The convolution result is generated in the form of a characteristic graph  $y$  of this layer by the activation function  $g(\cdot)$ .

$$y = g \left( \sum_{i=0}^H \sum_{j=0}^W \sum_{k=0}^L K_{i,j,k} \otimes x_{i^l,j^l,k^l}^l + b_{i,j,k} \right), \tag{1}$$

where  $\otimes$  represents the convolution operation,  $b_{i,j,k}$  represents the bias, and the convolution kernel  $k_{i,j,k}$  can be convoluted with one or more feature graphs of the previous layer.

$$F_{m_{l+1}} = \frac{F_{m_l} + 2 \times P_l - K_l}{\lambda} + 1, \tag{2}$$

where  $F_{m_{l+1}}$  is the size of the layer  $l + 1$  feature map,  $F_{m_l}$  is the size of the layer  $l$  feature map,  $K_l$  is the size of the layer  $l$  convolution core,  $\lambda$  is the step size of the convolution-core movement, and  $P_l$  is the number of columns with zero values

for the edge filling (padding) of the previous feature map in the convolution operation.

Softmax is widely used in machine learning and deep learning, especially in dealing with multiclassification problems. The softmax function performs numerical processing on the final output of the classifier and presents it in the form of relative probability. The softmax function is defined as follows:

$$S_i = \frac{\exp(v_i)}{\sum_{j=1}^c \exp(v_j)}, \tag{3}$$

where  $v$  is the output unit of the classifier, and  $S_i$  is the ratio of the guidance of the classifier output elements and the sum of the indices of all elements.

By using the cross-entropy function as a classification objective function, the softmax loss function is defined as follows:

$$f = -\frac{1}{n} \sum_{i=1}^n (p_i \log S_i + (1 - p_i) \log(1 - S_i)), \tag{4}$$

where  $n$  is the number of training set samples and  $p_i$  is the label distribution. After the objective function is obtained, the parameter weight is optimized using an error back-propagation algorithm.

## Machining feature splitting

### Feature splitting method

The 3D geometric information in STEP AP203 includes the normal vector of the vertex, the edge co-edge surface plane, and the edge. According to the information in the STEP file, the topological relationship between the machining features is obtained. First, all geometric models of the part are obtained. Then, the faces are represented as nodes, and the faces with intersecting edges are connected by straight lines (according to the concavity and convexity of edges). The connection mode between nodes is determined, and the AAG is obtained (Maturana & Scherer, 2015). The minimum subgraph is interpreted as an indicator of the existence of potential features in element construction. The minimum subgraph is generated by removing the convex connection while keeping the concave connection. These minimal subgraphs are the machining feature surfaces. Figure 3 shows a 3D model that includes the machining features of the holes, grooves, and columns. Figure 4 shows an AAG of the 3D part model, where the solid lines represent the convex edges, while the dotted lines represent the concave edges. The minimum subgraph can be obtained from the concavity or convexity

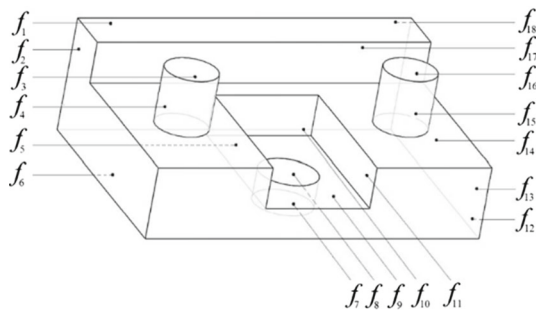


Fig. 3 3D part

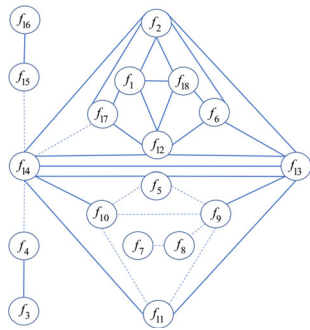


Fig. 4 AAG of part

of the edge. Figure 5 shows the separation of the machining features, which are the machining features of the holes and grooves in the part.

**Bounding box of machining features**

The minimum subgraph of the AAG is the only machining feature surface of the part. The question then becomes how to create feature entities for the part without generating new machining features during part substantiation. In this study, the bounding box method was adopted. The basic idea of this method is to replace complex geometric objects with a geometry (i.e., a bounding box) that has a slightly larger

Fig. 5 Minimum subgraph of machining features

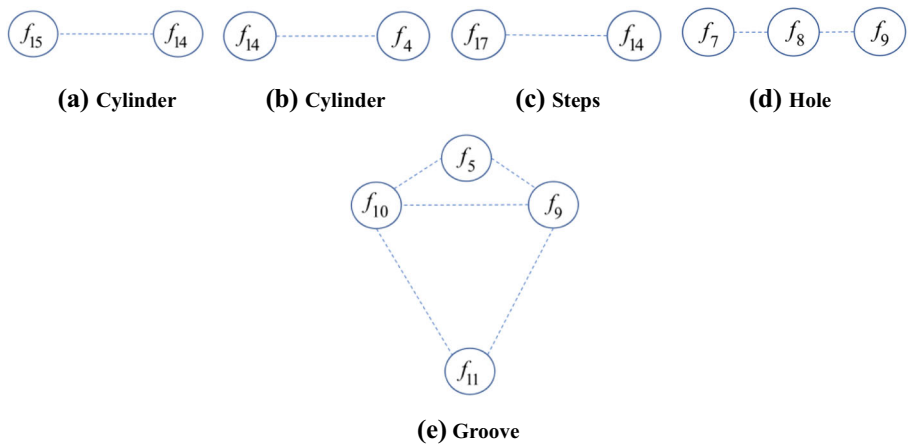
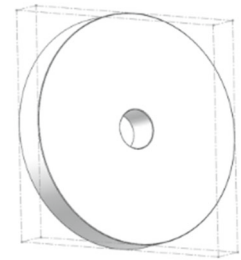


Fig. 6 Bounding box of a feature



volume and simple characteristics, as shown in Fig. 6. The axis-aligned bounding box (AABB) was the first bounding box used. It is defined as the smallest hexahedron containing the object with its edges parallel to the axis. Therefore, only six scalars are needed to describe an AABB.

In this study, the feature surface of a part is judged by minimum subgraph of AAG. Then, the processing feature surface of the part is bounding-boxized. bounding-boxization refers to the intersection of the six bounding box planes of the minimal subgraph and the processing features of the part. First, judging the normal direction of the feature surface. If the normal direction of the part diverges, the feature is a convex body. Using the bounding box of the feature surface and the feature surface to intersect and materialize the feature surface. If the normal direction of the feature surface converges, it means that the feature is a concave body. By extending the non-intersecting edge of the feature surface, the feature surface intersects the 1.1-fold bounding box, and the part containing the smallest subgraph of the processing feature is the feature entity of the part split. The split process is shown in Fig. 7. Figure 8 is the processing feature entity of the part split out in Fig. 3.

**Data preparation**

**Database creation**

There are many types of machining features because of the various functions and complex structures of parts. Thus, it is

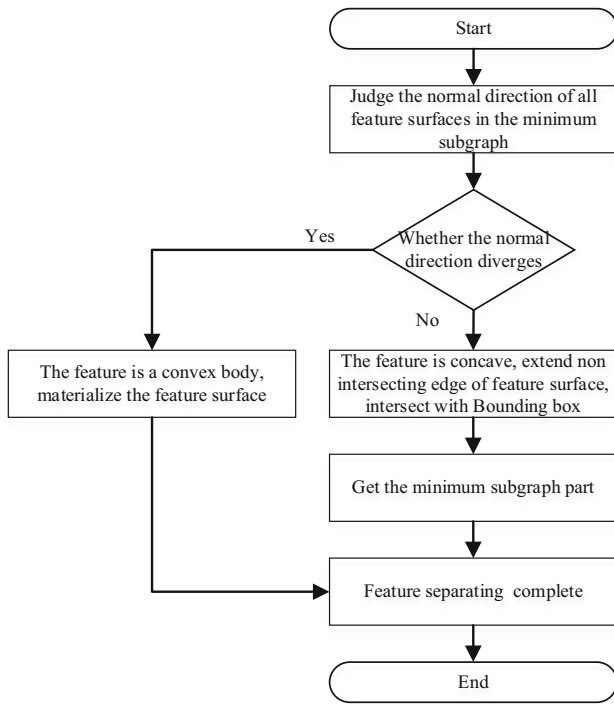


Fig. 7 Logic diagram of part separation

difficult to systematically classify the parts. There are currently few methods to classify the machining features of parts. To facilitate research on processing part-feature recognition methods, 14 types of common machining features are listed in Table 1, including convex and concave features, and the names and 3D models of the parts. The proposed feature recognition method can recognize any type of feature and

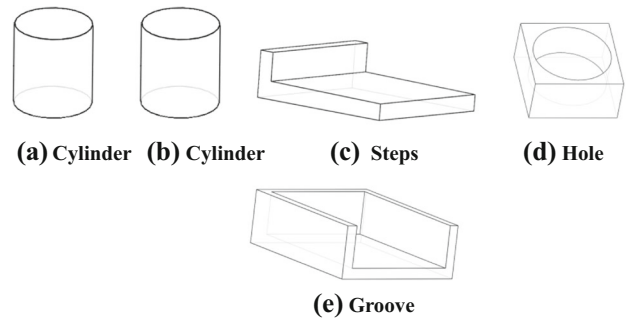


Fig. 8 Machining features of separated parts

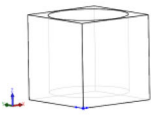
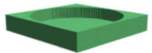

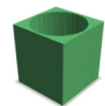
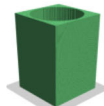


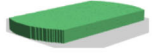


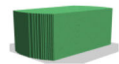


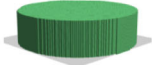


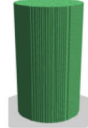
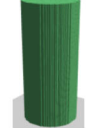
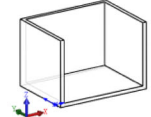
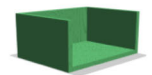
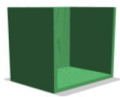
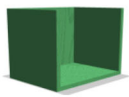


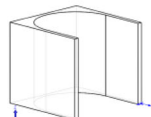
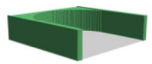
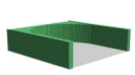
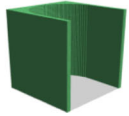


only needs to establish the corresponding feature training set.

The machining features are separated in the part model, and the feature surface is materialized in the bounding box. For a concave feature surface, the split features will be fixed on the bounding box of the part, the position of the feature surface is fixed relative to the box, and the length and width of the box may be different. The split convex features have the same shape, but the size of the faces may be different. The training of a 3D CNN is greatly influenced by the amount of training data. A large amount of data can make the 3D CNN converge within a period of short time. To obtain sufficient data, the STEP file is first transformed into a standard triangle language (STL) file. An STL file can describe the surface geometry of 3D objects and realize the representation of a logical model using a triangle mesh. Then, triangle meshes of the feature model with different scales [0.1–10] along the Z-direction are randomly selected, and 1000 different models are generated by scaling or stretching each

Table 1 Some features of common parts

Feature	Nonthrough hole	Square cube	Cylinder	Groove	Circular groove	Ring	Keyway
Type	Concave	Convex	Convex	Concave	Concave	Concave	Concave
Class	1	2	3	4	5	6	7
3D models							
Feature	Four-sided non-through hole	Five-sided non-through hole	Through hole	Straight groove	Four-sided through hole	Decahedron	Three-facet hole
Type	Concave	Concave	Concave	Concave	Concave	Convex	Concave
Class	8	9	10	11	12	13	14
3D models							

**Table 2** Some training datasets of machining features of some parts

Feature	STEP model	Voxel model				
Nonthrough hole						
Square cube						
Cylinder						
Groove						
Circular groove						

machining feature. In this way, the feature models of parts of different sizes and the same part of the machining feature are obtained. Some features are listed in Table 2. This feature-scaling method can quickly and efficiently yield a large amount of training data.

## Voxelization

Based on the expression of the plane information, the 3D information carrier adds depth information to the spatial information, which is unique in the space. This increases the 3D information. Voxelization transforms the geometric representation of the model into a voxel representation closest to the model, and generates voxel datasets. Therefore, it is necessary to transform the 3D geometric model of a part into voxel data via voxelization. The separated features are expressed as voxel data, and subsequently used as input data for the 3D CNN method.

A larger voxel resolution for a part results in a greater amount of details for the voxel data contained in the model, which induces the 3D CNN to learn the machining features contained in the voxel data. However, this will also cause greater difficulties in training. Therefore, a voxel resolution of  $128^3$  was selected. The models of some parts after the machining feature voxelization are given in Table 2.

To reduce the amount of model training calculations, data enhancement was included in the voxelization. New model data was generated after the model data was rotated around the  $X$ -,  $Y$ -, or  $Z$ -axis, as shown in Table 3.

## Experiments

### Experimental introduction

A 3D CNN was used to train the machining feature data of the parts to realize machining feature recognition. The dataset contained voxelated data split into three subsets: 84,000 training datasets, 28,000 validation datasets, and 28,000 testing datasets. The network structural and training parameters used in the experiment are listed in Table 4.

### Training process and analysis of results

The stochastic gradient descent (SGD) optimizer and root mean square prop (RMSProp) algorithm were used. The RMSProp algorithm was used to calculate the differential squared weighted average for the gradient. The advantages of this method include eliminating the swing amplitude direction and correcting the swing amplitude so that the swing

**Table 3** Implementing results of some data enhancements

Source model	Rotate 90° around the X-axis	Rotate 180° around the X-axis	Rotate 270° around the X-axis	Rotate 90° around the Y-axis	Rotate 180° around the Y-axis	Rotate 270° around the Y-axis	Rotate 90° around the Z-axis	Rotate 180° around the Z-axis	Rotate 270° around the Z-axis

**Table 4** Network structural parameters and training hyperparameters

Structure	Layer (type)	Output shape
Convolution	Con3d_1	1,64,64,32
	Con3d_2	1,64,64,32
	Con3d_3	1,32,32,32
	Con3d_4	1,32,32,64
	Con3d_5	1,16,16,64
Fully connected	dense	124
	dense	124
	dense	14

amplitude of each dimension was small (Ning, Shi, Cai, Xu, & Zhang, 2020). The activation function used the rectified linear unit function, and the initial learning rate was set as 0.001.

Training with the SGD optimizer ended after 37 epochs, which took 2 h. An i9 graphics processor, GeForce 2080Ti graphic card (11 GB), and a 1.2-TB hard disk were required. As shown in Fig. 8, the loss value of the network decreased with further iterations, and the classification accuracy of the network improved. The training and validation accuracies of the learning model were 93.57% and 92.89%, respectively. The accuracy of the test dataset obtained using the learned model was 90.73%. Training with the RMSProp optimizer ended after 33 epochs. As shown in Fig. 9, the loss value of

the network decreased with further iterations, and the classification accuracy of the network improved. The learned model achieved training and validation accuracies of 98.62% and 98.51%, respectively. The accuracy of the test dataset obtained using the learned model was 95.85%.

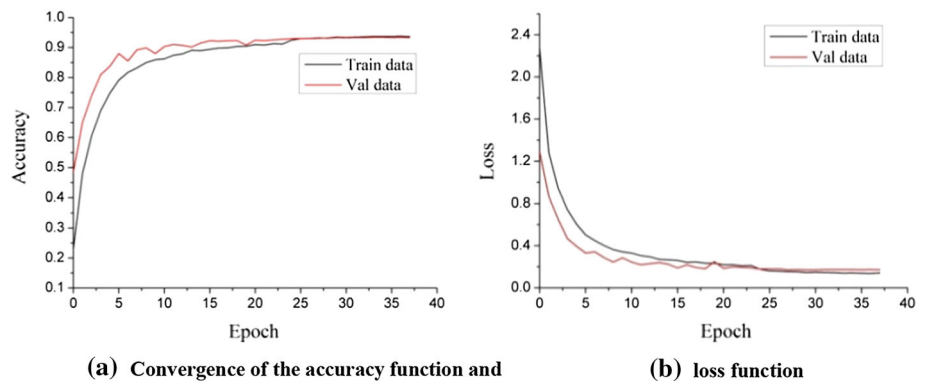
Table 5 summarizes the voxel data of 14 randomly features from the testing data used as the input of the learned model of the RMSProp optimizer. It also presents the predicted labels as the predicted features of the learned model and the true labels as the actual features. The probability of the extracted feature information was calculated using the classification probability classifier output by the softmax function.

From Table 5, it can be concluded that the accuracy of the test results is high, particularly for the features of the square cube and cylinder, which have significantly different shapes. Therefore, the proposed 3D CNN method has a high degree of shape recognition.

Figure 10 is the confusion matrix of 14 features in the test dataset classification using the learned model. It can be concluded from Fig. 10 that the 3D CNN had low accuracy in the classification of circular grooves (class 5) and straight grooves (class 11). The shapes of these two features are similar, and the voxel data weakens the difference when representing these two shapes. Therefore, improving voxel resolution can solve this problem.



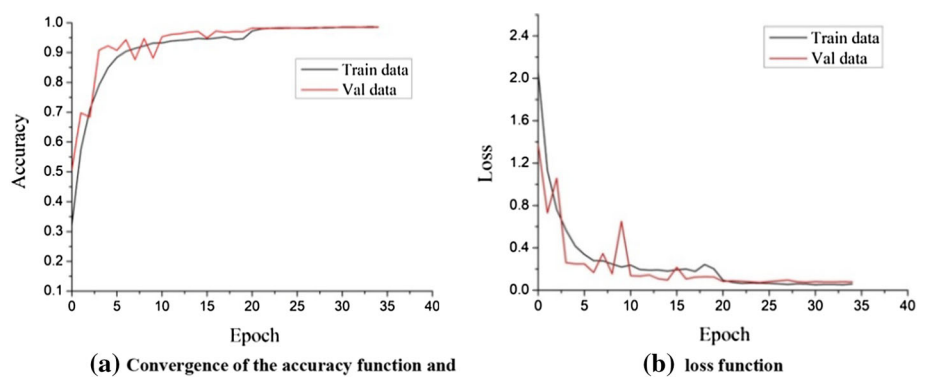
**Fig. 9** Training process using the SGD optimizer



**Table 5** Recognition and classification for 3D models

3D models							
Predicted label	Nonthrough hole	Square cube	Cylinder	Groove	Circular groove	Ring	Keyway
True label	Nonthrough hole	Square cube	Cylinder	Groove	Circular groove	Ring	Keyway
Probability	0.9523	0.9811	0.9714	0.9421	0.9511	0.9639	0.9514
3D models							
Predicted label	Four-sided nonthrough hole	Five-sided nonthrough hole	Through hole	Straight groove	Four-sided through hole	Octahedron	three-facet hole
True label	Four-sided nonthrough hole	Five-sided nonthrough hole	Through hole	Straight groove	Four-sided through hole	Octahedron	three-facet hole
Probability	0.9611	0.9423	0.9438	0.9558	0.9678	0.9614	0.9637

**Fig. 10** Training process using the RMSProp optimizer

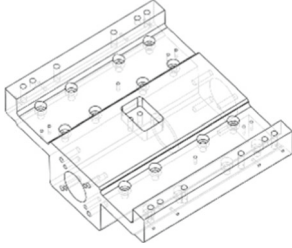
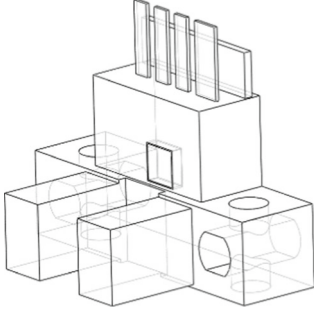
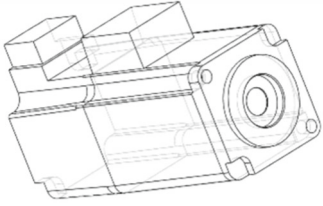
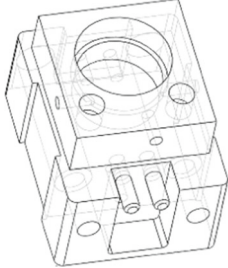
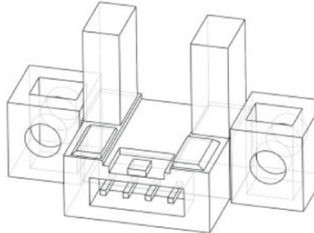
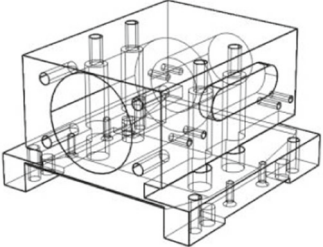
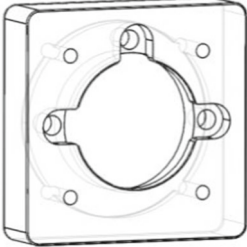
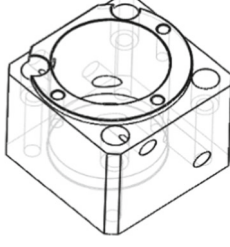
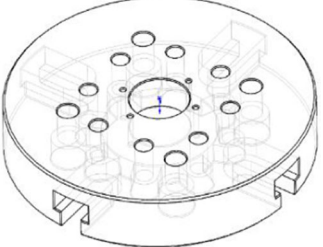


**Verification of results**

Some parts of a linear electric cylinder were selected to test the proposed part machining feature recognition method. The part structure was complex and included machining features

such as holes, through holes, and grooves, as listed in Table 6. These parts also included a transition surface, which was identified and deleted in this study. For example, for part 1, the number of actual machining features was 77; the number of features split by this method was 77; the number of accu-

**Table 6** Machining feature identification of actual parts

	Part 1	Part 2	Part 3
3D model			
Characteristic quantity	77	14	14
Number of separation features	77	13	14
Number of correct convex features identified	0	8	5
Number of correct concave features identified	71	5	8
Accuracy	100%	92.86%	92.86%
	Part 4	Part 5	Part 6
3D model			
Characteristic quantity	22	20	44
Number of separation features	21	20	43
Number of correct convex features identified	3	11	1
Number of correct concave features identified	18	9	42
Accuracy	95.45%	100%	97.73%
	Part 7	Part 8	Part 9
3D model			
Characteristic quantity	14	20	39
Number of separation features	13	19	38
Number of correct features identified	13	19	38
Accuracy	92.86%	95.00%	97.44%

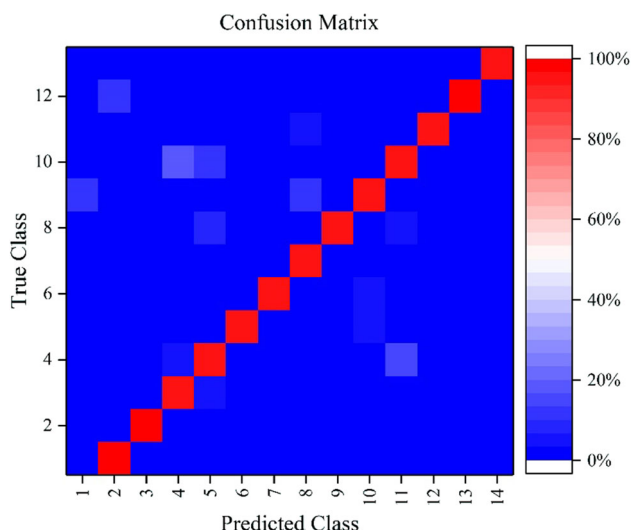


Fig. 11 Confusion matrix of 14 features in the test dataset classification using the learned model

rate recognitions of the split features by the trained 3D CNN was 77; and the feature recognition accuracy was 100%. The identification accuracies for the rest of the parts are listed in Table 6.

Fig. 12 3D model of a complex part

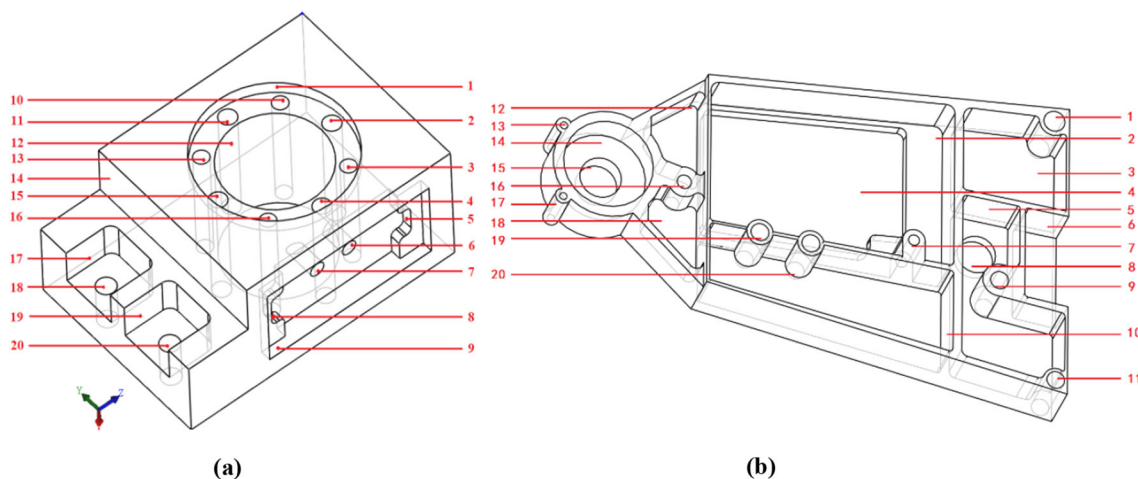
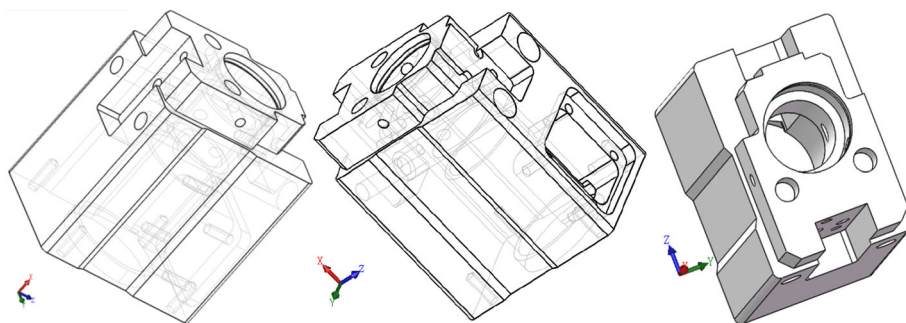


Fig. 13 3D model from reference (Sebastian, 2016; Zhang et al., 2018)

Thus, it can be concluded that the proposed separation method can realize accurate separation of the machining features listed in Table 7. The recognition results include convex, concave, and polyhedral features. Part processing starts from a blank, and the shape features of the blank are often convex. Therefore, the initial features of the part could not be determined by the minimum subgraph of the AAG. However, for the machining features of the parts, the proposed method performed well for the split features, which was recognized by the 3D CNN with high accuracy.

Figure 11 shows a 3D model of a complex part containing numerous holes and grooves. When the proposed method was used to identify the features of this part, the number of features identified was 34, the number of machining features of the split part was 37, and the feature recognition accuracy was 89.48%. There were 38 features in this part model (including one shape feature) among which three machining features were untrained, making it impossible to recognize them.

**Comparison of feature recognition results**

Compared with the method in Sebastian (2016), a new hybrid feature recognition method based on the graph + rule is proposed. As presented in Fig. 12, both parts (a) and (b) contain

21 machining features. The proposed method recognized them in less than 1 s, and all the features were correctly identified. The method presented in Sebastian (2016) required 2 s to identify parts (a) and (b), and features 2, 4, 11, and 15 in part (a) were identified as stepped holes. (Sebastian, 2016) defines as a series of coaxial holes, one inside another, such as a counterbore. The features are defined by graphics and rules that require complex definitions and take a long time to find specific definitions. In Zhang et al. (2018), 19 machining features of part (a) were separated by watershed method, and 18 machining features were identified correctly.

## Conclusions

The feature recognition method using the graph-based approach can only recognize polyhedral and concave features. 3D CNNs have a strong ability to recognize large data features and features of interest for classification. Thus, a 3D CNN combined with a graph-based approach was proposed to take advantage of deep learning technology and traditional feature recognition methods to recognize the machining features of parts.

To distinguish the convex and concave machining features of a part from the 3D model, the proposed method for finding the minimum subgraph in an AAG was studied to determine machining features. Then, its separation was realized using the bounding box concept.

Furthermore, a stretching and zooming method was proposed to obtain the training data. Fourteen common machining features were designed, which realized stretching and scaling in the Z-axis direction. The data enhancement method was used to obtain the feature data to train the 3D CNN.

The test results demonstrate that the proposed method accurately identified convex, concave, and polyhedral features, and improved the recognition efficiency. The ability to identify convex features further improved the recognition range.

## References

- Cao, W., Robinson, T., Hua, Y., Boussuge, F., Colligan, A. R., & Pan, W. (2020). Graph representation of 3D CAD models for machining feature recognition with deep learning. In *Proceedings of the ASME 2020 international design engineering technical conferences and computers and information in engineering conference*, 46th design automation conference (DAC) (Vol. 11A). <https://doi.org/10.1115/DETC2020-22355>.
- Ghadai, S., Balu, A., Sarkar, S., & Krishnamurthy, A. (2018). Learning localized features in 3D cad models for manufacturability analysis of drilled holes. *Computer Aided Geometric Design*, 62(MAY), 263–275.
- Gong, Z., Zhong, P., Yu, Y., Hu, W., & Li, S. (2019). A CNN with multiscale convolution and diversified metric for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6), 3599–3618.
- Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). Mgf: A multi-label graph-based feature selection algorithm via pagerank centrality. *Expert Systems with Applications*, 142, 46–53.
- Joshi, S., & Chang, T. C. (1988). Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-Aided Design*, 20(2), 58–66.
- Krot, K., & Czajka, J. (2018). Processing of design and technological data due to requirements of computer aided process planning systems. In *International conference on intelligent systems in production engineering and maintenance* (pp. 267–274).
- Ma, Y., Zhang, Y., & Luo, X. (2019). Automatic recognition of machining features based on point cloud data using convolution neural networks. In *Proceedings of the 2019 international conference on artificial intelligence and computer science* (pp. 229–235).
- Malhan, R. K., Kabir, A. M., Shah, B., & Gupta, S. K. (2019). Identifying feasible workpiece placement with respect to redundant manipulator for complex manufacturing tasks. In *2019 international conference on robotics and automation (ICRA)* (pp. 5585–5591).
- Malyshev, A., Slyadnev, S., & Turlapov, V. (2017). Graph-based feature recognition and suppression on the solid models. In *GraphiCon* (Vol. 17, pp. 319–322).
- Maturana, D., & Scherer, S. (2015). Voxnet: A 3D convolutional neural network for real-time object recognition. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 922–928).
- Ning, F., Shi, Y., Cai, M., Xu, W., & Zhang, X. (2020). Manufacturing cost estimation based on the machining process and deep-learning method. *Journal of Manufacturing Systems*, 56, 11–22.
- Rai, S., & Vairaktarakis, G. (2019). NP-complete problems and proof methodology. *Encyclopedia of Optimization*. [https://doi.org/10.1007/978-0-387-74759-0\\_462](https://doi.org/10.1007/978-0-387-74759-0_462)
- Sebastian, R. (2016). An overview of gradient descent optimization algorithms. arXiv:1609.04747
- Shi, Y., Zhang, Y., Baek, S., De Backer, W., & Harik, R. (2018). Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications*, 15(6), 941–952.
- Shi, P., Qi, Q., Qin, Y., Scott, P. J., & Jiang, X. (2020). A novel learning-based feature recognition method using multiple sectional view representation. *Journal of Intelligent Manufacturing*, 31, 96.
- Sunil, V. B., Agarwal, R., & Pande, S. S. (2010). An approach to recognize interacting features from b-rep cad models of prismatic machined parts using a hybrid (graph and rule based) technique. *Computers in Industry*, 61(7), 686–701.
- Venuvinod, P. K., & Wong, S. Y. (1994). A graph-based expert system approach to geometric feature recognition. *Journal of Intelligent Manufacturing*, 6, 401–413.
- Weise, J., Benkhardt, S., & Mostaghim, S. (2018). A survey on graph-based systems in manufacturing processes. In *2018 IEEE symposium series on computational intelligence (SSCI)* (pp. 112–119). IEEE.
- Wu, N. Z., Song, S., Khosla, A., Yu, N. F., Zhang, N. L., & Tang, N. X., et al. (2015). 3D Shapenets: A deep representation for volumetric shape modeling. In *IEEE conference on computer vision and pattern recognition*. arXiv preprint [arXiv:1406.5670](https://arxiv.org/abs/1406.5670).
- Xu, X., Caulfield, S., Amaro, J., Falcao, G., & Moloney, D. (2019). 1.2 Watt classification of 3D voxel based point-clouds using a cnn on a neural compute stick. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2018.10.114>
- Yuen, F., & Venuvinod, P. K. (1999). Geometric feature recognition: Coping with the complexity and infinite variety of features. *International Journal of Computer Integrated Manufacturing*, 12(5), 439–452.

- Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3708–3712).
- Zhang, Z., Jaiswal, P., & Rai, R. (2018). FeatureNet: Machining feature recognition based on 3d convolution neural network. *Computer-Aided Design*, *101*, 12–22.
- Zhao, X., Zhang, H., Zhu, G., You, F., Kuang, S., & Sun, L. (2019). A multi-branch 3D convolutional neural network for eeg-based motor imagery classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*(10), 2164–2177.
- Zhou, G., Yang, X., Zhang, C., Li, Z., & Xiao, Z. (2019). Deep learning enabled cutting tool selection for special-shaped machining features of complex products. *Advances in Engineering Software*, *133*, 1–11. <https://doi.org/10.1016/j.advengsoft.2019.04.007>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.