






A Visual-Cognition-Inspired Model for Machining Feature Recognition

Yenan Shi¹ , Jingchen Hu²  and Guolei Zheng³ 

¹Beihang University, shiyenan@buaa.edu.cn

²Beihang University, hujingchen@buaa.edu.cn

³Beihang University, zhengguolei@buaa.edu.cn

Corresponding author: Guolei Zheng, zhengguolei@buaa.edu.cn

Abstract. Automatic feature recognition technology is the key support of intelligent design and manufacturing. Using artificial neural networks (ANNs) to identify machining features is a significant interdisciplinary research direction. Although the ANNs have the ability to learn and generalize, and process faster, they can only process numerical input and perform arithmetic operations, not logical operations, thus restricting the application of ANNs in the field of CAD machining feature recognition. This paper establishes a new Visual-cognition-inspired Model (VCIM) for machining feature recognition by imitating the visual cognition process of the human brain and related neural mechanisms. The VCIM has a structure closer to the cerebral cortex than ANNs, uses a 3D CAD model as a direct input, and has three new activation functions that can perform logical operations. The VCIM have been tested and verified to identify four different types of machining features, and new features can be recognized by changing the activation function definition.

Keywords: visual processing system, neural network, machining feature, feature recognition, aircraft structural parts

DOI: <https://doi.org/10.14733/cadaps.2020.429-446>

1 INTRODUCTION

Automatic machining feature recognition is the key support of intelligent design and manufacturing. Feature recognition is an essential component in all CAD/CAM systems that rely on features for analysis and decision making. A feature is a set of geometric elements with semantics. Feature recognition is the mapping of geometric models to feature models of specific applications and manufacturing processes using various recognition rules or methods. On the one hand, the machining features are closely related to the machining process, involving machining methods, tool types, machining tool path, fixtures, etc. On the other hand, the features themselves are diverse, especially in the field of aircraft manufacturing, where aircraft structural parts contain a large number of complex structures. These matters make machining feature recognition an important issue in CAD/CAM.

In recent decades, feature recognition technology has developed rapidly and achieved rich research results. Among the many different types of recognition methods, the more classical methods are graph-based method [3], volume-based method [20], rule-based method [18] and hint-based method [6]. Although these methods have their own advantages and disadvantages, they generally have the following common problems:

- No ability to learn and abstract;
- Poor noise resistance for the CAD input models;
- Focusing on the representations of specific types of CAD models and cannot be generalized to feature representations of other different representation methods;
- Poor ability to handle variable features and intersecting features.

These issues constrain the development of feature recognition methods, but they are compatible with ANN methods. First, ANNs have the ability to learn. Second, the ANN methods are data-driven methods. As long as a general mathematical expression method is established, the ANN can be used to identify different types of features. Again, ANNs have the ability to identify similar features without the need to predefine all feature instances. Therefore, since the end of the 1990s, scholars at home and abroad have begun to study the use of ANNs to recognize machining features, and have achieved a series of remarkable results [16]. At present, the ANNs for machining feature recognition include Multilayer Perceptron (MLP), Back-propagation Neural Network (BPNN), Self-organizing Feature Map (SOFM), Adaptive Resonance Theory (ART) network, Hopfield Neural Network, Convolutional Neural Network (CNN), etc. These ANN methods have a common point: they need to preprocess and encode CAD models or features, convert them into numerical vectors (also called Representation Vectors, denoted as RVs) and input them into the ANNs for calculation. Through the literatures, the commonly used feature preprocessing and encoding methods are Attributed Adjacency Graph (AAG) encoding, Face Adjacency Matrix (FAM) encoding, Face Score Vector (FSV) encoding and voxelization methods, etc.

For AAG encoding, the AAG is first decomposed into AAG subgraphs, then the AAG subgraphs are transformed into Adjacency Matrixes (AMs), and finally the elements of AM are selected to generate RV [5] [11-12]. Since AAG only defines the topology information of the model, the encoding method does not contain the geometric information of the model. FAM-based encoding is an improved version of AAG encoding. Compared with the AM, the FAM increases the encoding information of the surfaces, so that the adjacency relationship between the surfaces can be defined more accurately. Prabhakar and Henderson [15] used FAM to encode features, but failed to separate features with the same topology but different shapes. Considering that the AAG can only recognize plane and simple surface features, Ding and Yue [2] established the F-adjacency matrix and the V-adjacency matrix, which can identify the intersection features with common bottom surface. FSV encoding is the most commonly used feature preprocessing method. The method first defines the calculation formula of FSV and the score table of each component, and then defines the number of elements of RV and the corresponding face order, and then calculates the value of each feature surface according to the formula. Finally, each face value is used as an element value to generate an RV. Among them, the face value calculation formula is a function related to face, edge, vertex and their adjacency relationship [8]. The literatures [7], [9], [13-14] and [17] all use the FSV method to process and encode features. The voxelization method is a feature preprocessing and encoding method unique to 3D CNN. By dividing the CAD model into voxels, the voxels are assigned with binary numbers, these values constituting the input data of CNN [1] [4] [19]. However, the amount of parameters of the voxelization method is huge, which greatly increases the calculation cost of the network. Taking the literature [1] as an example, a simple hole feature model can create 1 billion voxels.

In summary, before recognizing machining features, ANNs must perform the preprocessing and encoding operations to convert the features from CAD models to numerical representation information. However, the conversion of information may have the following problems:

- Cannot accurately and completely express the geometric and topological information of the machining features;
- May overlap when representing different types of features;

- May cause the loss of certain information of features and affect the recognition accuracy.

Therefore, creating a reasonable and effective feature preprocessing and encoding method or designing a neural network that does not need to perform feature preprocessing are two methods to solve the above problems. This paper takes the latter as the starting point, gets inspiration from the visual cognitive process and neural mechanism of human brain, and establishes a new neural network structure VCIM for machining feature recognition. The VCIM uses the CAD 3D model as a direct input and does not require feature preprocessing operations, thus avoiding the three problems mentioned above. Since the objects processed by VCIM are face features, not numerical information, the new neural network performs logical operations rather than numerical calculations. This paper also establishes three different types of activation functions for the VCIM.

2 RELATED WORKS

The inspiration of VCIM comes from two aspects, one is the structure and function of the brain's primary visual cortex (the V1 cortex), and the other is the brain's visual processing flow. Through anatomical studies, it has been found that the V1 cortex is a tissue that contains both horizontal layer structures and vertical column structures.

(1) Layer structures

The V1 cortex consists of six horizontal layers, followed by the Molecular Layer (the I layer), the External Granular Layer (the II layer), the External Pyramidal Layer (the III layer), the Internal Granular Layer (the IV layer), the Internal Pyramidal Layer (the V layer), and the Pleomorphic Layer (the VI layer). There is a difference in the type and amount of cells in each layer, so the functions of each layer are different. Usually, after the nerve impulse is transmitted to the IV layer, it is spread to the layers based on the II layer and the IV layer, and finally outputted from the III layer, the V layer, and the VI layer.

(2) Column structures

In addition, cells in the V1 cortex with similar functional properties are located close together in columns that extend from the cortical surface to the white matter. The columns are concerned with functional properties and thus reflect the functional role of that area in vision. For example, some columns contain cells with similar selectivity for the orientation of stimuli.

A piece of visual cortex tissue can be called a visual cortex module. There are a number of columns in the module. These columns have the following characteristics: the neurons in a column can respond to a specific functional characteristic, and columns capable of responding to similar characteristics are spatially adjacent and partially overlapped. Thus, a stimulus can activate a plurality of contiguous columns within the cortex that form a hypercolumn. A columnar module can contain a number of hypercolumns that enable remote control between the hypercolumns by means of horizontal connections.

In summary, the tissue in the cortex has a hierarchical relationship of cortex → module → hypercolumn → column → cell, and each column contains six horizontal layers with different functions.

The brain's visual processing is a parallel and phased process, and Figure 1 shows the parallel paths of visual system. In the figure, the pink pathway is the ventral pathway, also called what pathway, carrying information about shape and color, mainly related to object recognition, and the storage of long-term memory. The blue pathway is the dorsal pathway, also known as the where/how pathway, which is dedicated to visually guided motion, as well as to the representation of the position of the object and the control of the eyes and arms, since it contains cells that are selective for the direction of motion. The two pathways can meet in the same cortex; at the same time, one cortex can contain many different types of visual characteristics, such as color, direction and shape. Therefore, the same cerebral cortex can recognize multiple visual characteristics at the same time.

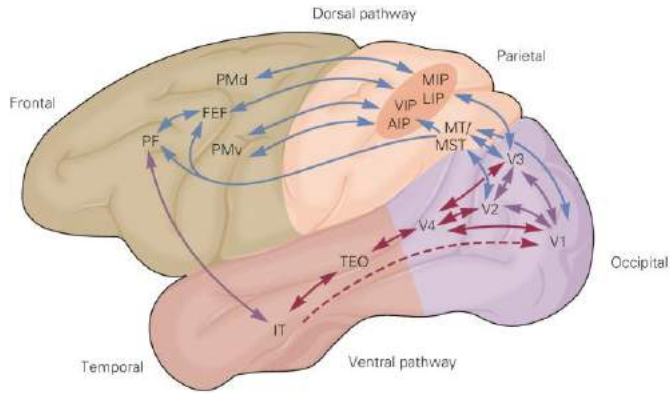


Figure 1: Parallel paths of visual processing system [10].

3 STRUCTURES

According to the above analysis of the visual processing mechanism, two rules can be summarized:

- The structure of the cortex is in order of cortex → module → hypercolumn → column → cell, and the cells in the column are arranged in layers according to function.
- The same cortical module can simultaneously recognize multiple visual characteristics.

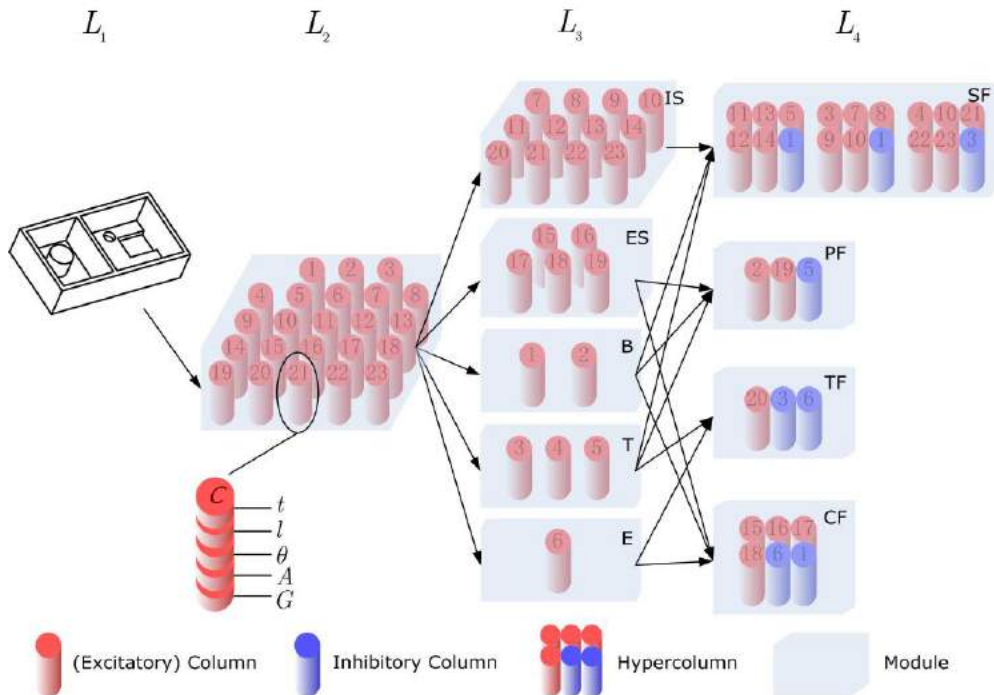


Figure 2: VCIM network structure.

Based on the above rules, this paper designs the VCIM network model shown in Figure 2. The VCIM is a four-layer neural network, consisting of L_1 layer, L_2 layer, L_3 layer and L_4 layer in sequence. Each layer contains several modules, each of which owns several columns. For some modules, there may also be hypercolumn structures composed of multiple columns. The relevant terms are explained below, and the variables and their definitions are given first.

<i>Variable</i>	<i>Definition</i>
L_i	i -th layer of the VCIM
f	A face
C	A column
θ_d	Direction angle of a column
t	Type of a column
l	Outer loop of a column
θ_e	External normal angle of a column
G	Adjacent face set of a column
H	A hypercolumn
M	A module
$g()$	Distribution function
$M_i \ i = 1,2,3,4,5$	IS-module, ES-module, B-module, T-module or E-module
n_i	Number of columns in M_i
$\varphi_i()$	Judgment function of M_i
α_m	Maximum inclination angle of the ISide
β_m	Maximum inclination angle of the BFace
γ_m	Maximum inclination angle of the TFace
E	A set of excitatory columns
I	A set of inhibitory columns
$M'_i \ i = 1,2,3,4$	SF-module, PF-module, TF-module or CF-module
m_i	Number of columns in M'_i
$h_i()$	Selection function of M'_i

Table 1: Variables and definitions.

3.1 Column

A column is used to store one face (f) of the CAD model. The column contains five components, each of which defines a category attribute of the face. The five attributes are the direction angle θ_d , the type t , the outer loop l , the external normal angle θ_e , and the adjacent face set G . If C is used to indicate a column, there is

$$C = \theta_d, t, l, \theta_e, G \quad (3.1)$$

In Figure 2, each (red or blue) cylinder represents a column, and every column has a hierarchical structure, indicating the five attribute of a face. In this paper, the machining features in aircraft structural parts are mainly studied. Therefore, the types of faces are defined as three

types, plane, ruled face (including cylindrical face) and other types of faces, which are represented by $t \in \text{Plane}$, $t \in \text{RuledFace}$ and $t \in \text{ElseFace}$, respectively.

Before determining the face direction with the normal direction, the direction of the CAD model is stipulated. The aircraft structural parts are generally flat, and the shape features are mostly concentrated on the front and back of the structural parts, as shown in Figure 3. The bottom surface is referred to as a horizontal plane (xy plane). The direction perpendicular to the horizontal plane is referred to as the vertical direction (z direction), and the forward direction (+z direction) is defined from the bottom surface of the model to the front side of the model. Therefore, the +z direction in Figure 3(a) is perpendicular to the bottom surface and points upwards, and the +z direction in Figure 3(b) is perpendicular to the bottom surface and points downwards. According to the above agreement, the direction component and the normal component of the face can be defined.

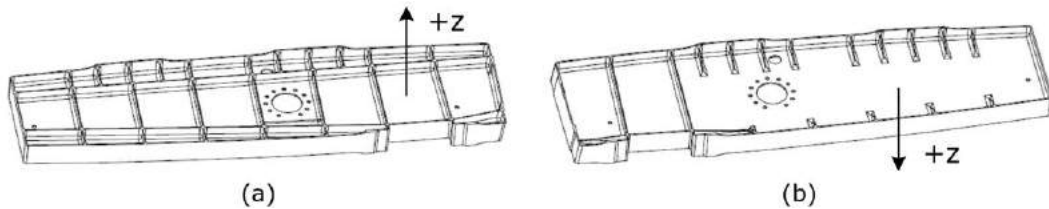


Figure 3: An aircraft structural part: (a) the front side and (b) the back side.

The direction angle θ_d of the face refers to the angle between the face and the z-axis. If the face is a plane, then θ_d is the angle between the plane and the z-axis. If the face is a ruled face, and then θ_d is the angle between the generatrix of the ruled face and the z-axis. θ_d is set as $\theta_d \in [0, 90^\circ]$. The external normal angle θ_e refers to the angle between the external normal of the face and the +z direction, and $\theta_e \in [0, 180^\circ]$.

The outer loop of the face is divided into three types according to the concavity and convexity. When the edges of the outer loop are all concave, the outer loop belongs to a concave loop, which is recorded as $l \in cLoop$. When the edges of the outer loop are all convex, the outer loop belongs to a convex loop, which is recorded as $l \in vLoop$. If the outer loop is neither a concave loop nor a convex loop, and then the outer loop is called a mixed loop, which is recorded as $l \in mLoop$.

The adjacent face set refers to the set of faces adjacent to the outer loop of f . If there is a face f_1 in which the outer loop of f is the inner loop of f_1 , then it is expressed as $G = +f_1$. If the outer loop of f is also the outer loop of f_1 , it means $G = -f_1$. If f has multiple outer loops, and the above two cases exist at the same time, it is expressed as $G = -f_1 + f_2$.

3.2 Hypercolumn

For a set of columns $H = C_i | i=1, 2, \dots, n$, where n is the number of columns, if H can constitute a machining feature, such as a slot, a protrusion, etc., and then these columns are called a hypercolumn. In Figure 2, multiple columns clustered together represent a hypercolumn. The hypercolumn is a specific structure in the layer L_4 , and each hypercolumn in a module is a machining feature.

3.3 Module

A module is a more advanced set of columns/hypercolumns. Each module stores a class of columns with the same definition or similarity. If the module is marked as M , and then there is

$$M = C_i \cup H_j \mid i = 1, 2, \dots, m, j = 1, 2, \dots, n \quad (3.2)$$

Where C_i is the i -th column in the module, H_j is the j -th hypercolumn in the module, and m and n are the number of columns and hypercolumns. For L_2 and L_3 , the module does only contain columns, so $H_j = \emptyset$. For L_4 , the module contains hypercolumns, there is $C_i = \emptyset$. In Figure 2, a rectangular block represents a module.

3.4 Layer

A layer is a more advanced form of a module. One layer contains one or more different modules. If the layer is marked as L , then there is

$$L = M_i \mid i = 1, 2, \dots, m \quad (3.3)$$

Where m is the number of modules in the layer.

The VICM is a new type of neural network that can recognize machining features in different fields. The number of layers, modules and columns are optional. When towards the machining features of the aircraft structural parts, the VICM is set to a four-layer network described above. For better understanding the layer structure, the model in Figure 2 is described here as an example in detail.

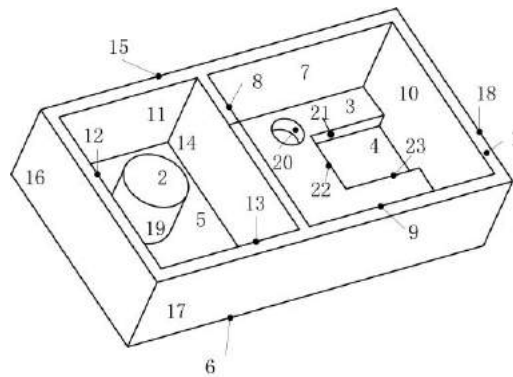


Figure 4: CAD model to be recognized.

(1) L_1 is the CAD model to be recognized.

(2) L_2 has a module that stores all the 23 faces of the CAD model, each column representing a face with a number, and the faces marked by serial numbers can be seen in Figure 4.

(3) L_3 has five modules. IS-module stores all the internal sides (ISides), which contains twelve faces (or columns) from 7 to 23. ES-module stores all the external sides (ESides), which contains five faces from 15 to 19. B-module represents the bottom face (BFace) module, storing Face 1 and Face 2. T-module represents the top face (TFace) module and stores Faces 3, 4 and 5. E-module stores all the "else faces (EFaces)" that do not belong to other modules, namely Face 6.

(4) L_4 has five modules, which store slots, protrusions, through holes/opening, outer contour and other features, labeled SF-module, PF-module, TF-module, CF-module, and EF-module respectively. These modules can also be seen in Figure 2.

The VICM performs dynamic structure configuration. For different CAD models, the number of modules and the number of columns in each layer may not be the same.

4 ACTIVATION FUNCTIONS

The activation functions of the ANNs are used to calculate the weight of neurons, and introduce nonlinear characteristics to the network. Similarly, the VICM activation functions are used for column calculations and are responsible for mapping the input of the column to the output. However, the activation functions of the ANNs process numerical calculations, but the activation functions of the VICM are to perform logical operations on the faces of the CAD model, and different kinds of activation functions are used between different layers.

4.1 Distribution Function

STEP files express the geometric model with B-Rep, B-Rep defines a model as a closed boundary surface enclosing a limited space such that a part can be represented by its boundaries (a subset of surfaces), each face defined by edges [9]. The distribution function takes the topological data of CAD 3D model as input, assigns a column to each face, and stores the attribute information of the face. The distribution function acts between L_1 layer and L_2 layer. The output is the columns of L_2 . The distribution function is defined as

$$C_i = g(f_i) \quad (4.1)$$

Where f_i is the i -th face of the CAD model, and $g()$ represents the distribution function.

The algorithm flow of the distribution function is shown in Figure 5. The mark with "unvisited/visited" is for the purpose of distinguishing and identifying faces.

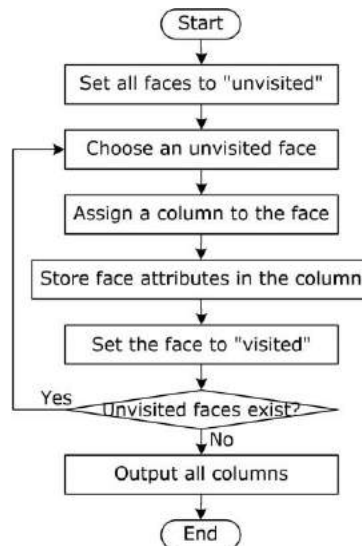


Figure 5: Algorithm flow of the distribution function.

4.2 Judgment Function

The judgment function judges each column of the layer L_2 , and calculates its relative position type to determine which module it belongs to. The result is stored in the corresponding module of the layer L_3 . The related expression is defined as:

$$M_i = C_j | \varphi_i(C_j) == T, i = 1, 2, \dots, 5, j = 1, 2, \dots, n_i \quad (4.2)$$

Where M_1, M_2, M_3, M_4 and M_5 are IS-module, ES-module, B-module, T-module and E-module, n_1, n_2, n_3, n_4 and n_5 indicate the number of columns in the corresponding module, and $\varphi_1(), \varphi_2(), \varphi_3(), \varphi_4()$ and $\varphi_5()$ are the judgment functions of the five modules separately. If and only if the judgment function of a certain column is true, and then the column is the element of the module corresponding to the current judgment function.

The five judgment functions are introduced below.

(1) ISide judgment function

The ISide is generally a plane or a ruled face. Usually, the plane is perpendicular to the horizontal plane, and the generatrix of the ruled surface is along the z-axis. However, there is also a case where the ISide is inclined, and generally the inclination angle is small. Therefore, assuming that the maximum inclination angle is α_m , $\alpha_m \leq 25^\circ$, the external normal angle of the ISide should be satisfied $\theta_e \in 90^\circ - \alpha_m, 90^\circ + \alpha_m$.

The judgment function $\varphi_1()$ of M_1 can be expressed as:

$$\varphi_1(C) = \left(\begin{array}{l} \theta_d \in [0, \alpha_m] \wedge t \in \text{Plane} \vee t \in \text{RuledFace} \wedge \\ l \in mLoop \wedge \theta_e \in [90^\circ - \alpha_m, 90^\circ + \alpha_m] \end{array} \right) \quad (4.3)$$

Specifically, there may be two external loops on the ISide, and the two external loops are both the inner loops of two faces in G , and the judgment function can be expressed as

$$\varphi_1(C) = G == +f_1, +f_2 \quad (4.4)$$

Equations (4.3) and (4.4) together serve as the judgment function of the ISide.

(2) ESide judgment function

The ESide is similar to the ISide. The only difference is that the outer loop is a convex loop. Therefore, the judgment function $\varphi_2()$ of M_2 can be expressed as:

$$\varphi_2(C) = \left(\begin{array}{l} \theta_d \in [0, \alpha_m] \wedge t \in \text{Plane} \vee t \in \text{RuledFace} \wedge \\ l \in vLoop \wedge \theta_e \in [90^\circ - \alpha_m, 90^\circ + \alpha_m] \end{array} \right) \quad (4.5)$$

Specifically, there may be two external loops on the ESide, and the two external loops are the inner loop of one face and the outer loop of the other face in G respectively, and the judgment function can be expressed as

$$\varphi_2(C) = G == +f_1, -f_2 \quad (4.6)$$

Equations (4.5) and (4.6) together serve as the judgment function of the ESide.

(3) BFace judgment function

The type of a BFace is a plane, and the outer loop is a concave loop or a mixed loop. In general, the BFace is parallel to the horizontal plane. However, there is also a case where the bottom face is inclined, and generally the inclination angle is small. Therefore, assuming that the maximum inclination angle is β_m , the external normal angle of the BFace should be no greater than β_m , i.e. $\theta_e \in [0, \beta_m]$. The judgment function $\varphi_3()$ of M_3 can be expressed as:

$$\varphi_3(C) = \left(\begin{array}{l} \theta_d \in [0, \beta_m] \wedge t \in \text{Plane} \wedge \\ l \in cLoop \vee l \in mLoop \wedge \theta_e \in [0, \beta_m] \end{array} \right) \quad (4.7)$$

(4) TFace judgment function

The TFace is a plane or a ruled face, and the outer loop is a convex loop. Typically, the generatrix of the ruled face or the plane is parallel to the horizontal plane. However, there is also a case where the top face is inclined, and generally the inclination angle is small. Therefore, assuming that the maximum inclination angle is γ_m , the external normal angle of the TFace should be satisfied $\theta_e \in [180^\circ - \gamma_m, 180^\circ]$. The judgment function $\varphi_4()$ of M_4 can be expressed as:

$$\varphi_4(C) = \left(\begin{array}{l} \theta_d \in [0, \gamma_m] \wedge t \in \text{Plane} \vee t \in \text{RuledFace} \wedge \\ l \in v\text{Loop} \wedge \theta_e \in [180^\circ - \gamma_m, 180^\circ] \end{array} \right) \quad (4.8)$$

(5) EFace judgment function

When a face does not belong to the above four types, it is stored in the E-module. The judgment function $\varphi_5()$ can be expressed as:

$$\varphi_5(C) = \left(\begin{array}{l} \varphi_1(C) \neq T \wedge \varphi_2(C) \neq T \wedge \\ \varphi_3(C) \neq T \wedge \varphi_4(C) \neq T \end{array} \right) \quad (4.9)$$

Through the above judgment functions, the columns of the layer L_2 can be identified into the corresponding modules in the layer L_3 .

4.3 Selection Function

The selection function selects the appropriate module and its columns from the layer L_3 to form different types of machining features, which are stored in the layer L_4 . The faces associated with the machining features can be divided into two types, one is element faces directly forming the machining features, and the other is limit faces adjacent to the element faces and playing a limiting role. Both the two types of faces collectively define the machining features (see Table 2).

<i>Machining feature</i>	<i>Element face</i>	<i>Limit face</i>
Slot	ISide, BFace, EFace	TFace, BFace
Protrusion	ESide, TFace, EFace	BFace
Through hole/opening	ISide, EFace	BFace, ESide
Contour	ESide	TFace, ESide

Table 2: Machining features with element faces and limit faces.

By mimicking the excitatory stimulation and inhibitory stimulation of neurons, the columns that store the element faces in the VCIM are called excitatory columns, which are represented by a set E . The columns that store the limit faces are called inhibitory columns and are represented by a set I . Figure 2 shows excitatory columns and inhibitory columns. For the convenience of the view, inhibitory columns are shown in blue and excitatory columns are in red. As mentioned earlier, a feature is stored in a hypercolumn H , therefore H can be represented as $H(E,I)$. The construction expressions for various features of the layer L_4 are:

$$M'_i = H_j | H_j = h_i E_j, I_j, i = 1,2,3,4, j = 1,2,\dots,m_i \quad (4.10)$$

Where M'_1, M'_2, M'_3 and M'_4 indicate SF-module, PF-module, TF-module and CF-module. m_i indicates the number of columns in the corresponding module. $h_i()$ is the selection function of the corresponding module.

According to the composition information of various features in Table 1, the definition of the selection functions are

$$\begin{aligned}
 \hat{h}_1 E, I &= E \subset M_1 \wedge E \subset M_3 \wedge E \subset M_5, I \subset M_4 \\
 \hat{h}_2 E, I &= E \subset M_2 \wedge E \subset M_4 \wedge E \subset M_5, I \subset M_3 \\
 \hat{h}_3 E, I &= E \subset M_1 \wedge E \subset M_5, I \subset M_3 \wedge I \subset M_5 \\
 \hat{h}_4 E, I &= E \subset M_2, I \subset M_4 \wedge I \subset M_5
 \end{aligned}
 \tag{4.11}$$

As can be seen from Table 2 and Figure 2, certain features such as slots, protrusions, and holes all contain EFaces, which is due to the widespread presence of chamfered faces or fillets in machining features in aircraft structural parts. These faces are located in the E-module because they do not meet the other module definitions in the layer L_3 . For some stepped slots or slots having a nested relationship, the limit face of the lower slot is the BFace of the upper slot, so the limit face of the slot also includes the BFace.

5 IMPLEMENT

In this section we demonstrate how the VCIM works for machining feature recognition, and validate and analyze the performance. As CATIA® is a multiplatform CAD/CAM/CAE commercial software suite which provides processing solution to a wide variety of industries, we adopt it for testing and analysis.

5.1 Procedure

Based on the structure of the VCIM, the process of VCIM computing is divided into three phases.

- Phase 1: With a CAD 3D model as input, distribute a column to each face according to distribution function in 4.1, with each component storing a face attribute.
- Phase 2: Judge the individual columns of L_2 based on the judgment functions in 4.2, and place the columns that meet the conditions in the appropriate modules.
- Phase 3: Select the appropriate columns in L_3 to form the E and I in hypercolumns, and finally identify all features.

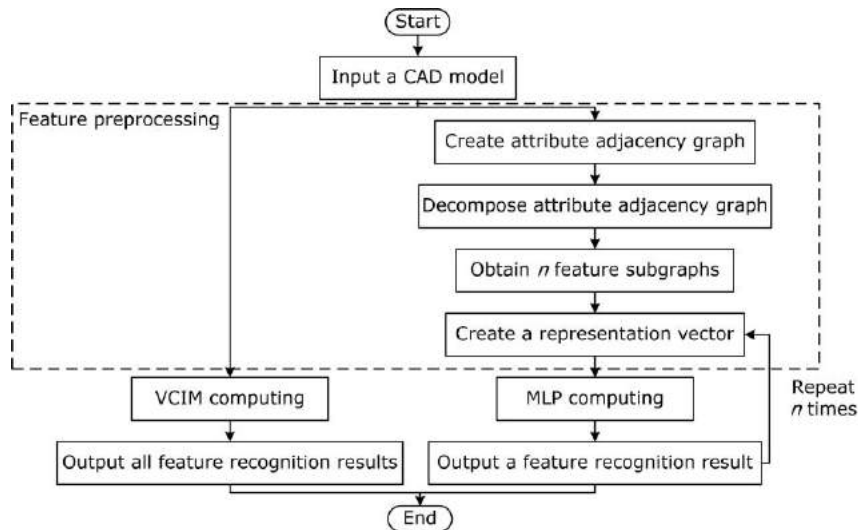


Figure 6: Algorithm flow of the two methods.

To compare the difference between the proposed method and other approaches, the literature [17] is used for discussion. The flow chart of the two algorithms is shown in Figure 6. After a CAD model is input into the system, the VCIM algorithm is carried out along the left process and the literature method is carried out along the right process. Three main differences can be found between the two approaches. One is the VCIM method does not need feature preprocessing like the MLP method. The second is that the MLP method can only recognize one feature at a time, so it is necessary to separate the features of the CAD model first, and then send the features to the MLP network for calculation. If the CAD model has n features, the neural network needs to be executed n times. Another thing to notice is that the MLP in Figure 6 is trained by default, but the VCIM does not need to be trained, because the activation function defines the topological relationship of a class of features, which is independent of the geometry and shape of the features, so it can accurately identify the type of feature. This is essentially different from MLP adjusting weights through errors.

5.2 Examples

In this section, three different experiments are conducted. Example 1 tests the results and validates the method accuracy. Example 2 and Example 3 use a simple model and a complex model to compare and analyze the proposed method and the literature [17] method.

5.2.1 Example 1

The first example is the CAD 3D model shown in Figure 2 and Figure 4. According to the agreement mentioned above, the model is positively placed and +z direction points upwards. Every cylinder in Figure 2 represents a column. In L_2 , the faces are distributed and numbered (the numbering results can see Figure 4). In L_3 , the faces are judged and placed in five modules. In L_4 , the faces are selected to find their correct machining features with E and I . For the convenience of the view, inhibitory columns are shown in blue and excitatory columns are in red. Multiple columns clustered together represent a hypercolumn, namely a feature.

In the experiment, six features are finally recognized. They are 3 slots, 1 protrusion, 1 through hole and 1 contour. The recognition results are shown in Table 3. The number before the parenthesis represents the face serial number, and the parenthesis shows the face positions in the modules L_3 and L_4 . Table 3 does not show the limit faces of each machining feature.

<i>Feature result</i>	<i>Face composition</i>
Slot 1	$7(M_{31}, M_{41}), 8(M_{31}, M_{41}), 9(M_{31}, M_{41}), 10(M_{31}, M_{41}), 3(M_{34}, M_{41})$
Slot 2	$11(M_{31}, M_{41}), 12(M_{31}, M_{41}), 13(M_{31}, M_{41}), 14(M_{31}, M_{41}), 5(M_{34}, M_{41})$
Slot 3	$21(M_{31}, M_{41}), 22(M_{31}, M_{41}), 23(M_{31}, M_{41}), 10(M_{31}, M_{41}), 4(M_{34}, M_{41})$
Protrusion 1	$2(M_{33}, M_{42}), 19(M_{32}, M_{42})$
Through hole 1	$20(M_{31}, M_{43})$
Contour 1	$15(M_{32}, M_{44}), 16(M_{32}, M_{44}), 17(M_{32}, M_{44}), 18(M_{32}, M_{44})$

Table 3: Recognition results of Example 1.

5.2.2 Example 2

The second example is a simple part in Figure 7(a). According to the agreement mentioned above, the +z direction of the model is obliquely upward. There are seven features in this model, which are 5 slots, 1 blind hole and 1 external contour.

The VCIM used in the part is a four-layer network, but the architecture is different from that of the first example. From Figure 7(b), the VCIM has 62 columns in total. In L_3 , the part has 45 ISides, 9 EFaces, 1 TFace, 6 BFaces and 1 EFace. In L_4 , the faces constitute 6 machining features. 49 columns form five slot features, and 9 columns form an external contour features. These numbers do not include the limit face numbers. The recognition result for one slot is given in Figure 7(c).

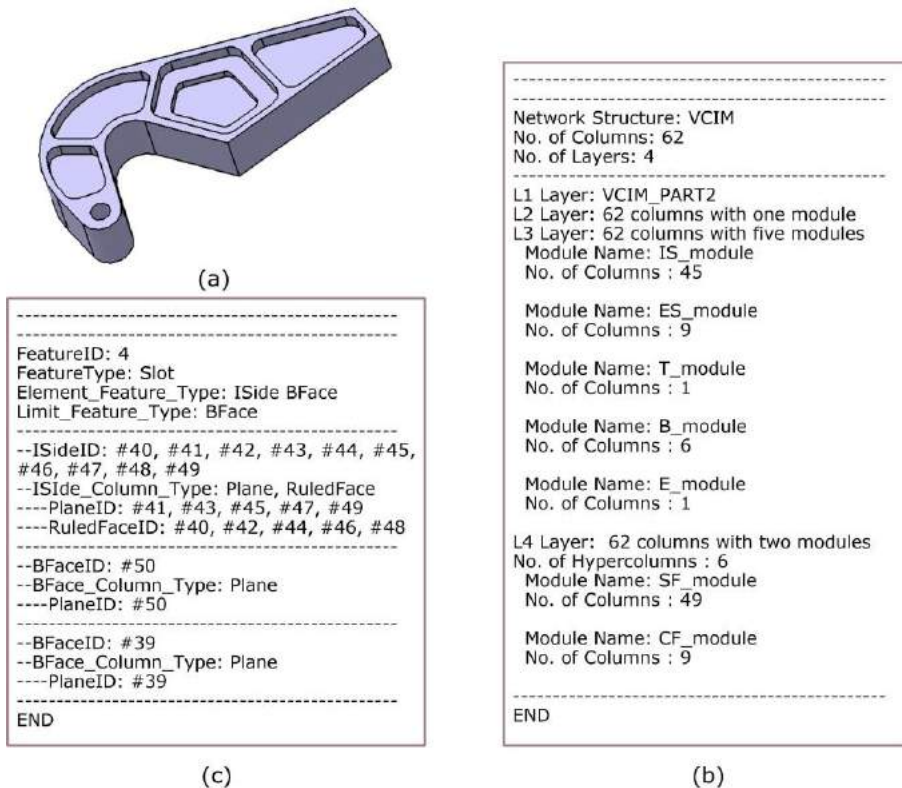


Figure 7: Recognition results of Example 2: (a) the test part, (b) the network structure for the part, and (c) a slot recognition result.

When comparing and analyzing the two methods, recognition accuracy and parameters are two important reference indicators. The recognition results are shown in Figure 8(a). It can be seen that compared with the features in the model (shown in green), both methods can recognize all the slots, but the VCIM method fails to recognize the blind hole, and the MLP does not recognize the contour. The reason for the failure of the VCIM is that the selection function does not define the blind hole. Equation (4.11) can add a selection function for blind holes, that is

$$h_5 E, I = E \subset M_1 \wedge E \subset M_5 \wedge t \in \text{RuledFace}, I \subset M_3 \wedge I \subset M_5 \quad (5.1)$$

Where $t \in \text{RuledFace}$ is used to define that the ISide of the blind hole consists of a ruled surface so as to distinguish it from a slot.

Parameters are another item to discuss. For a trained MLP, the parameters are related to the number of neurons of each layer. The MLP in [17] is a four-layer model with the number of neurons of 12-24-24-1. Therefore, if each feature is calculated only once, the calculation amount is $12 \times 24 + 24 \times 24 + 24 \times 1 = 888$. For the VCIM, the parameters are mainly concentrated in L_2 , L_3 , and

L_4 . A face needs to be calculated at most five times to determine the module it belongs to in L_3 , and each face needs at most four times to calculate its feature in L_4 . Therefore, the calculation amount of a face is $1 \times 5 + 1 \times 4 = 9$, and the calculation amount of a feature is $9 \times n$, where n is the number of faces in a feature. Based on the above discussion, the parameters of the two methods are shown in Figure 8(b). It can be seen that compared with the MLP, the VCIM method has an absolute advantage in terms of parameters.

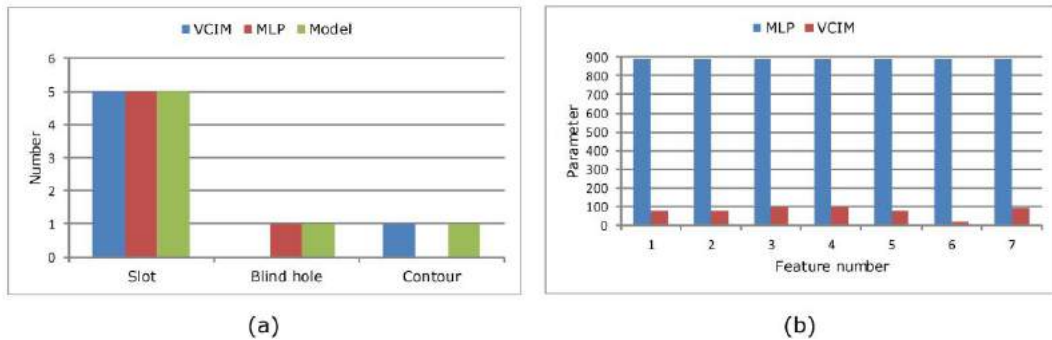


Figure 8: Comparison of two methods: (a) the number of features recognized, and (b) the parameters of two methods.

5.2.3 Example 3

The third example is a complex aircraft frame structure in Figure 9(a). According to the agreement mentioned above, the +z direction of the model is upward. There are 68 slot features, 14 through hole/opening features, 4 protrusion features and 1 external contour features in this model. Both the VCIM method and the literature [17] method can recognize all the features.

The VCIM used in the part is a four-layer network, but the architecture is different from those of the first two examples. From Figure 9(b), the VCIM here has about 772 columns in total. The faces are numbered in L_2 . In L_3 , the part has 574 ISides, 18 ESides, 68 TFaces, 111 BFaces and 1 EFace. In L_4 , the faces are assembled to get machining features. There are four kinds of machining features in the model. 614 columns form the slot features, 8 column forms the protrusion features, 28 columns form the through hole/opening features and 14 columns form the external contour features. These numbers do not include the limit face numbers. The recognition result for one slot marked with red ellipse in Figure 9(a) is given in Figure 9(c).

The features in third example are more complex and irregular. For example, the number of sides constituting slots is large and the side shapes are quite different, and Figure 10(a) is a typical instance. The feature is a slot and its sides and bottom faces have irregular shapes. However, it still satisfies the activation functions defined about slot, so that the VCIM can recognize it. According to the recognition results in Figure 10(b), this feature is identified as two slots. The green faces represent sides of slots.

Also, there are some intersection features in the frame structure, such as intersection slots, which share some sides. This article takes the area enclosed by the red rectangle in Figure 9(a) as an example to illustrate. The area can be divided into 6 subareas, each of which is marked by the number 1 to 6. The two methods both decompose the intersection feature into single sub-features for recognition. As the activation functions and feature definitions are different, the recognition results are not the same, which can be seen in Figure 11(a). The VCIM identifies all features as slots, and the literature method identifies the same features as pockets/slots and steps respectively. The total number of features recognized by the two methods is the same.

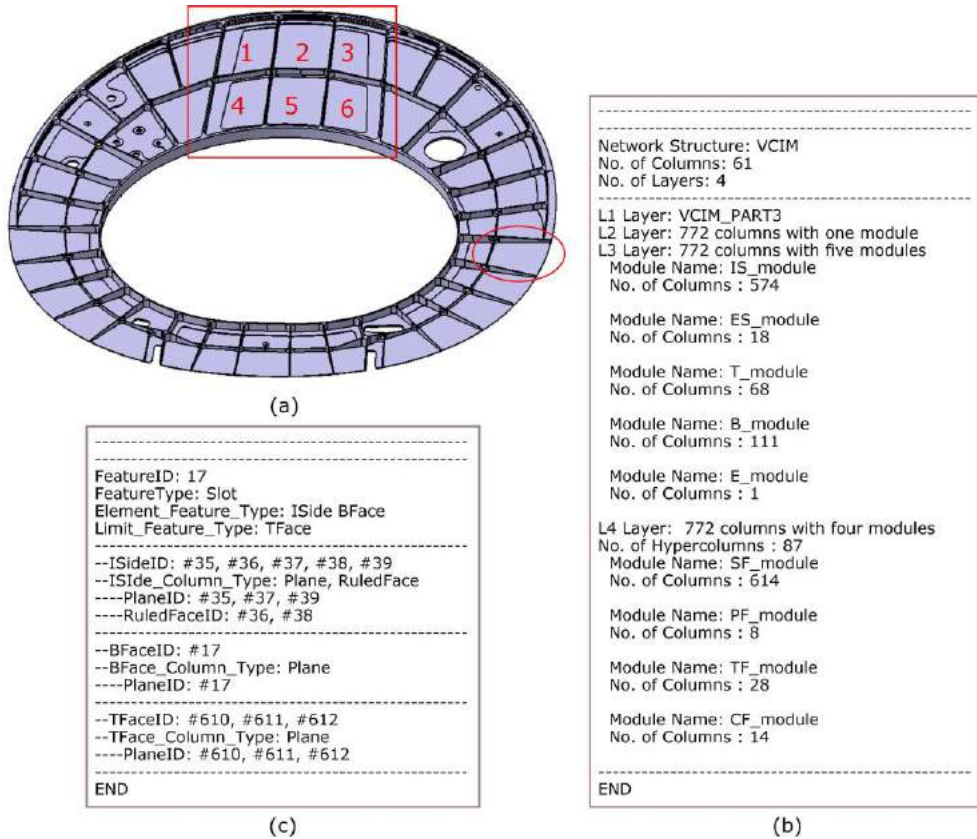


Figure 9: Recognition results of Example 3: (a) the test part, (b) the network structure for the part and (c) a slot recognition result.

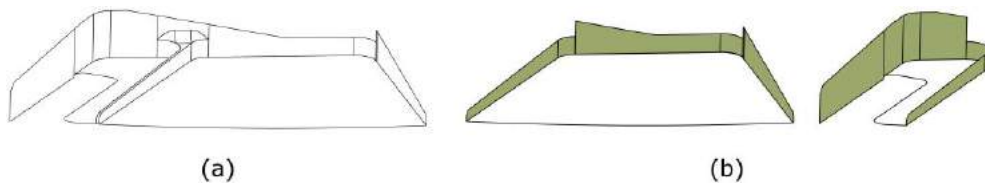


Figure 10: A slot feature of Example 3: (a) the slot, and (b) the recognition results of the VCIM.

Another thing to notice is the parameters of recognizing intersection features. As mentioned above, for a trained MLP, if a feature is calculated once in the MLP, and then the parameters are 888. For a subarea in Figure 9(a), the calculation amount is $888 \times n$, where n is the feature number of the subarea. For the VCIM, the calculation amount of a face is $1 \times 5 + 1 \times 4 = 9$, and the calculation amount of a feature is $9 \times m$, where m is the number of faces in a feature. Based on the above discussion, the parameters of the two methods are shown in Figure 11(b). It can be seen that the parameters of the VCIM method are much smaller than those of the MLP method, and the results are similar to those of Example 2.

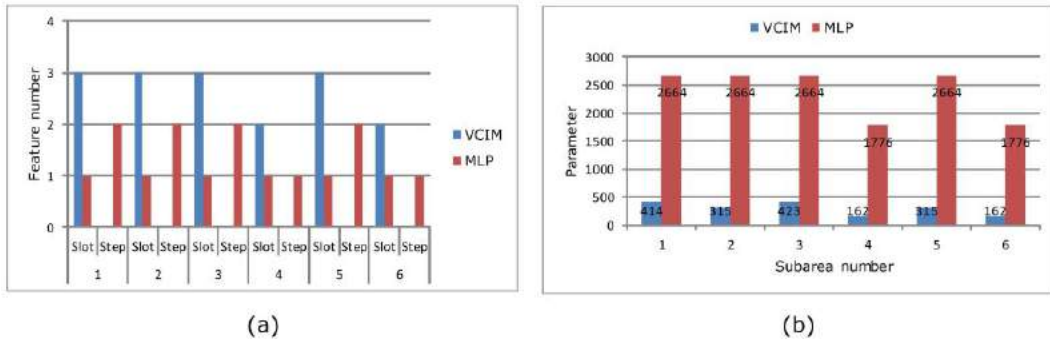


Figure 11: Comparison of two methods: (a) the number of features recognized, and (b) the parameters of two methods.

5.3 Analysis

Through the three examples, this paper analyzes the VCIM functionally.

(1) Feasibility

As the VCIM is a new method of calculating, the feasibility of the HBIM is the basic performance towards its applications. From Example 1, Example 2 and Example 3, the VCIM can accurately recognize machining features both in simple parts and more complex aircraft structural part. This proves the feasibility of the VCIM.

(2) Generality

As a goal and an important characteristic of the VCIM, the generality of the VCIM is also the concern of this paper. The application scope of this method is machining feature recognition in the field of CAD/CAM, especially for aircraft structural parts. Feature is a region of functional interest on a part [17], and machining feature is a group of adjacent faces which are machined out and have corresponding processing meaning, not any faces in a CAD model. Generally, a kind of machining feature has a kind of topology, and this is also the main idea of activation functions in this paper. Therefore, the VCIM built on this basis can recognize this kind of feature, regardless of the specific shape and number of faces of the feature.

In Example 1, the slot features only consist of planes, and the sides are surrounded by a standard rectangle. In Example 2, the slot features contain ruled faces, and the slot shapes are more natural, such as sector and pentagon. The features in Example 3 are more complex. There are many irregularities in the part, such as the slot in Figure 10(a) and the intersection slot features. Although these slots are different, they can be accurately identified. This is an important manifestation of the generality of the algorithm, that is, the feature recognition gets rid of the influence of shape and number of faces. However, if the system does not define the activation functions of a kind of feature, the VCIM will fail to recognize this feature, such as the blind hole in Example 2. Therefore, the activation functions are extremely significant for the VCIM.

(3) Unity

The unity of the VICM is that the network system can recognize several different types of machining features. Using the one system, all recognition results can be output, without the need for each feature to create a recognition method. When the activation functions are changed, the VCIM can be used to recognize new features in other fields. More importantly, the number of columns in the input layer is dynamic and not fixed, so models of any size are applicable. The size of the model can be changed from 23 in Example 1 to 62 Example 2, or even 772 Example 3.

(4) Comparison with MLP

Compared with MLP, the VCIM has several differences. First, the VCIM does not need training. Although it has a structure similar to neural network, it is not a real neural network because the

neurons in VCIM are not weights, nor are they corrected by error functions. In fact, the neurons are faces with their attributes. The activation functions define the topological relationship of a class of features, which is independent of the geometry and shape of the features, so it can accurately identify the type of feature. Second, the VCIM has no learning ability. As the activation functions are similar to logical expressions to judge various faces, they do not adjust weights to obtain expected output, so that the VCIM does not have learning process. Third, the parameters of the VCIM method are much smaller than those of the MLP method, which can be seen from Example 2 and Example 3. Fourth, the VCIM does not need feature preprocessing, and can recognize all features once, while the MLP separates the features of the CAD model first, and identify a feature at a time. At last, the structure of VCIM is variable, which reduces the amount of calculation to a certain extent, but the MLP is a network with fixed layers and neurons.

6 CONCLUSIONS

In this paper, a new network structure called VCIM is established for machining feature recognition in the field of such as aircraft structural parts, which is different from the traditional ANNs in the following innovations:

- The VCIM has column, hypercolumn and module structures similar to cerebral cortex.
- The VCIM creates three new activation functions for logical operations.
- The configuration structure of each layer can be dynamically configured as needed.
- New machining features can be identified once the definition of activation functions is changed.

Future work will further verify the feature recognition of more complex CAD models, and study the identification of intersecting features and composite features. Also, the activation functions with more precise classification is a major concern.

Yenan Shi, <https://orcid.org/0000-0003-1167-1240>
 Jingchen Hu, <https://orcid.org/0000-0001-5310-6341>
 Guolei Zheng, <https://orcid.org/0000-0002-3887-6265>

REFERENCES

- [1] Balu, A.; Lore, K. G.; Young, G.; Krishnamurthy, A.; Sarkar, S.: A deep 3D convolutional neural network based design for manufacturability framework, ArXiv e-prints, 1612.02141, 2016. <https://arxiv.org/abs/1612.02141v1>
- [2] Ding, L.; Yue, Y.: Novel ANN-based feature recognition incorporating design by features, *Computers in Industry*, 55(2), 2004, 197-222. <https://doi.org/10.1016/j.compind.2004.02.002>
- [3] Fougères, A.-J.; Ostrosi, E.: Intelligent agents for feature modelling in computer aided design, *Journal of Computational Design and Engineering*, 5(1), 2018, 19-40. <https://doi.org/10.1016/j.jcde.2017.11.001>
- [4] Ghadai, S.; Balu, A.; Sarkar, S.; Krishnamurthy, A.: Learning localized features in 3D CAD models for manufacturability analysis of drilled holes, *Computer Aided Geometric Design*, 62, 2018, 263-275. <https://doi.org/10.1016/j.cagd.2018.03.024>
- [5] Guan, X. S.; Meng, G. W.; Yuan, X. H.: Machining feature recognition of part from STEP file based on ANN, 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, IEEE, Changchun, China, 2010, 54-57. <http://doi.org/10.1109/CMCE.2010.5609638>
- [6] Han, J. H.; Requicha, A. A.: Integration of feature-based design and feature recognition, *Computer-Aided Design*, 29(5), 1997, 393-403. [https://doi.org/10.1016/S0010-4485\(96\)00079-6](https://doi.org/10.1016/S0010-4485(96)00079-6)

- [7] Hao, Y. T.; Chi, Y. M.: Research on ANN-based feature recognition and manufacturing behavior sequence, 2011 Second International Conference on Mechanic Automation and Control Engineering, IEEE, Hohhot, China, 2011, 7568-7574. <https://doi.org/10.1109/MACE.2011.5988802>
- [8] Hwang, J. L.; Henderson, M. R.: Applying the perceptron to three-dimensional feature recognition, *Journal of Design Manufacture*, 2(4), 1992, 187-198.
- [9] Jian, C. F.; Li, M.; Qiu, K. Y.; Zhang, M. Y.: An improved NBA-based STEP design intention feature recognition, *Future Generation Computer Systems*, 88, 2018, 357-362. <https://doi.org/10.1016/j.future.2018.05.033>
- [10] Kandel, E. R.; Schwartz, J. H.; Jessell, T. M.; Siegelbaum, S. A.; Hudspeth, A. J.: Principles of neural science 5th ed., McGraw-Hill Education / Medical, United States, 2012.
- [11] Li, W. D.; Ong, S. K.; Nee, A. Y. C.: Recognition of overlapping machining features based on hybrid artificial intelligent techniques, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 214(8), 2000, 739-744. <https://doi.org/10.1243/0954405001518107>
- [12] Nezis, K.; Vosniakos, G.: Recognizing 2½D shape features using a neural network and heuristics, *Computer-Aided Design*, 29(7), 1997, 523-539. [https://doi.org/10.1016/S0010-4485\(97\)00003-1](https://doi.org/10.1016/S0010-4485(97)00003-1)
- [13] Onwubolu, G. C.: Manufacturing features recognition using backpropagation neural networks, *Journal of Intelligent Manufacturing*, 10(3-4), 1999, 289-299. <https://doi.org/10.1023/A:1008904109029>
- [14] Öztürk, N.; Öztürk, F.: Hybrid neural network and genetic algorithm-based machining feature recognition, *Journal of Intelligent Manufacturing*, 15(3), 2004, 287-298. <https://doi.org/10.1023/B:JIMS.0000026567.63397.d5>
- [15] Prabhakar, S.; Henderson, M. R.: Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models, *Computer-Aided Design*, 24(7), 1992, 381-393. [https://doi.org/10.1016/0010-4485\(92\)90064-H](https://doi.org/10.1016/0010-4485(92)90064-H)
- [16] Qi, F.; Tan, J. R.; Zhang, S. Y.: Feature recognition based on RBF neural networks, *Journal of Computer-Aided Design & Computer Graphics*, 14(6), 2002, 562-565. <http://doi.org/10.3321/j.issn:1003-9775.2002.06.015>
- [17] Sunil, V. B.; Pande, S. S.: Automatic recognition of machining features using artificial neural networks, *The International Journal of Advanced Manufacturing Technology*, 41(9-10), 2009, 932-947. <https://doi.org/10.1007/s00170-008-1536-z>
- [18] Zehtaban, L.; Roller, D.: Automated rule-based system for Opitz feature recognition and code generation from STEP, *Computer-Aided Design and Applications*, 13(3), 2016, 309-319. <https://doi.org/10.1080/16864360.2015.1114388>
- [19] Zhang, Z. B.; Jaiswal, P.; Rai, R.: FeatureNet: Machining feature recognition based on 3D convolution neural network, *Computer-Aided Design*, 101, 2018, 12-22. <https://doi.org/10.1016/j.cad.2018.03.006>
- [20] Zhao, P.; Sheng, B. Y.: Recognition method of process feature based on delta-volume decomposition and combination strategy, *Journal of South China University of Technology (Natural Science Edition)*, 39(8), 2011, 30-35. <http://doi.org/10.3969/j.issn.1000-565X.2011.08.006>