The International Journal of Advanced Manufacturing Technology (2022) 121:7451–7462
https://doi.org/10.1007/s00170-022-09839-0

ORIGINAL ARTICLE

Development of a deep learning machining feature recognition network for recognition of four pilot machining features

Naser Mohammadi[1] · Mohammad Javad Nategh[1]

Received: 16 May 2022 / Accepted: 20 July 2022 / Published online: 30 July 2022

Abstract

Over the past few decades, several methods have been introduced for the recognition of machining features from design files. These methods entail considerable effort to infer the characteristics of machining features from the design and manufacturing information commonly expressed in different and incompatible formats on the geometry, topography, dimensions, tolerances, and surface finishes. The competence of these traditional methods in implementing their assignment is still challenging. The deep learning approach as an advanced facility in artificial intelligence is considered as a promising substitute. This approach has already been successfully and efficiently employed in different facets of social life. Its use in manufacturing engineering and industries is considered as a breakthrough, especially in some disciplines such as process planning, and machining feature recognition is in its embryonic phase. The input data in the present methods is the output information of a computer-aided design system. This is commonly faced with some limitations, the most serious ones of which are the loss of data from the design file occurring due to the geometric interference of features and slow extraction of features due to the extensive and excessive information existing in the design files. In the present study, a deep learning-based system named MFR-Net has been developed for the recognition of machining features from the images of workpieces. In addition to CAD systems, other tools such as the camera images of workpieces can also be employed to enter the input. The MFR-Net can also identify other relevant information needed for machining, including the position of the features, dimensions, and various symbols such as numbers, decimal points, and positive and negative signs employed in tolerances, parentheses, etc.

Keywords Computer-aided process planning · Machining feature · Deep learning · Artificial intelligence · Image processing

1 Introduction

The history of using computers in the machining industry dates back to the introduction and use of CNC machines in the late 1970s [1]. In workshops equipped with stand-alone CNC machines, the machining time comprises about 20% of the operating time, whereas, in automotive shops, this time reaches about 60% [2]. Big data is a characteristic feature of today's technology. Artificial intelligence and machine learning are efficient tools for processing big data. They provide new opportunities for manufacturing industries to enhance their competence and efficiency [3].

The recognition of machining features is a key step in computer-aided process planning. There are different definitions for machining features in the literature. According to Prabhakar and Henderson [4], a machining feature can be inferred as a mathematical representation of the topological and/or geometrical data defining that feature. These data can be derived from the CAD model of the workpiece to be machined.

Each machining feature undergoes a specific machining operation by using specific cutting tools, and it can be said that machining features convert CAD output data into computer-aided manufacturing (CAM) input data. [5]. The machining features are recognized by using the geometric and topological information from design files represented in different CAD schemes. Several machining rules are then used to plan the sequences of machining operations [6]. The

✉ Mohammad Javad Nategh
nategh@modares.ac.ir

Naser Mohammadi
n_mohammadi@modares.ac.ir

[1] Mechanical Engineering Department, Tarbiat Modares University, Tehran, Iran

geometric and topological information is then processed in specialized systems with the objective of automatic recognition of the machining features [7]. The CAD representation schemes include the wire frame model, surface representation model, and volumetric model. The traditional methods of machining feature recognition use one of these three representation models to identify the machining features [8]. The purpose of using a machining feature identification system is to convert the geometric and topological information of a workpiece designed in the design system into a set of features that can be used in the computer-aided process planning stage [9]. The machining feature recognition methods can be broadly divided into three categories including conventional methods, methods based on artificial neural networks, and the recently developed deep learning systems. Artificial neural networks are feature-based machine learning methods. They are less accurate as compared with deep learning because the recognized features are limited to knowledge of the mathematics developed by humans. For instance, a scratch considered as a defect cannot be properly distinguished from a line that is intentionally drawn on the surface of a component [10].

Among the common methods of machining feature recognition are the graph-based [11], the hint-based [12], the rule-based [13], the volume decomposition [14], and the syntactic rules pattern methods [15]. Extracting the manufacturing information of machining features such as dimensional and geometric tolerances has been the subject of other research efforts [16]. Extracting the characteristics of machining features from two-dimensional data with the help of programming languages such as Prolog is another field of research that has a long history [17]. The multilayer artificial neural network training for the identification of machining features began more than two decades ago [18] and had some successes in some areas, such as evaluating the machinability of freeform parts [19]. These methods based on artificial neural networks could alleviate some of the shortcomings of the conventional methods, such as the inability to detect defective machining features and low execution speed [20]. The application of artificial intelligence techniques in some areas such as surface defect detection [21] or tool wear prediction [22] is considered as a breakthrough in intelligent production, and in some disciplines such as process planning, and machining feature recognition is in its embryonic phase [23]. Among the few research work conducted in this field can be named the work done by Zhang et al. [24], who proposed a system for identifying machining features using a deep learning method called FeatureNet, in which a three-dimensional deep learning network is used to recognize the machining features from the output data of a design file. Shi et al. [25] developed a deep learning network called MsvNet for identifying the machining features by using multiple cut-out views of the features. Peng et al. [26] used a deep learning network to analyze the diagnoses of faults in rotating machinery parts. Moreno-Garcia et al. [27] employed two-dimensional convolutional networks to identify electrical components in an engineering drawing image. A similar work was done by Zhao et al. [28] to identify building components from scanned drawings of a building structure. In this research, a deep learning network has been taught by using 1500 scanned 2D building plans.

It should be noted that the development of a totally automated intelligent machining system, comparable with the decision-making faculty of expert operators, is a very difficult task [29]. It was tried in the present research to solve this difficulty by generating an intelligent system for the recognition of machining feature parameters existing in an image.

In the present work, a deep learning machining feature recognition network, named MFR-Net, has been developed by using the YOLO architecture [30]. This network has been trained to develop the required capability for the detection of machining features. MFR-Net can extract the machining features directly from the pictures taken from the workpiece. In addition to distinguishing the machining features, MFR-Net can identify other relevant information needed for machining, including the position of the features, dimensions, dimensional tolerances, and various symbols such as numbers, decimal points, and positive and negative signs employed in tolerances, parentheses, etc. in 34 identifiable classes. The introduced method could circumvent some of the limitations of the previous methods. The loss of some design data occurring due to the geometric interference of machining features and the dependence of identification capability on the design data formats are among these limitations. In addition, the developed method can deal with a sketch of the workpiece whose design file is not available. An additional innovative feature of MFR-Net is its capability to assign the dimensions and symbols recognized in an image to the relevant features.

In the previous methods for identifying machining features, the input data is directly obtained from B_Rep CAD or obtained through transforming the data to mesh, voxel, or cloud points. In the present work, however, the image of a workpiece consisting of one or more machining features is used as input to the feature recognition network. The image of a mechanical part can be created by different tools, including the imaging tool provided by the computer-aided design system. In previous methods, the design data can easily be lost due to the geometric interference of features or repeated conversion of data. In the present method, this problem has been avoided. The network developed by the authors is able to deal with the images of workpieces that are still in the initial steps of conceptual design.

2 Machining feature recognition network: MFR-Net

2.1 The architecture of MFR-Net

The exhaustive identification of a machining feature (MF) entails the extraction of three levels of design and manufacturing characteristics including basic, intermediate, and primary characteristics. The basic characteristics (BCs) represent lines and edges with their specific attributes. The intermediate characteristics (ICs) represent faces, bends, corners, etc. Each IC may consist of several BCs. The primary characteristics (PCs) embody the topological and geometrical attributes, dimensions, position, and of MF. The basic, intermediate, and primary characteristics are categorized into two groups. The first group is responsible for the recognition of some information such as the name of MFs and the location of specified dimensions in the MFs image. The second group includes the text, numbers, etc. Each of these characteristics in the two groups has its special relative impact coefficients that are used in the MFR-Net design and training phases. Twenty machining features introduced in STEP AP224 can be recognized as the basic machining features by MFR-Net. The composite machining features consisting of the basic features can also be identified. Four of these twenty machining features, including round edge, fillet, chamfer, and v-slot, were selected as pilots for designing the network.

The identification of MFs is carried out in three parts of MFR-Net, as three independent subnetworks. One subnetwork is developed for the recognition of MFs' shapes and the locations of dimensions. Another subnetwork is implemented for the recognition of tolerances, symbols, and texts. The other subnetwork is developed as the inference module. The schematic architecture of MFR-Net consisting of these three subnetworks is presented in Fig. 1. Each of these subnetworks is designed on the basis of YOLO v4 architecture. All BCs and ICs of machining features are extracted by the two first subnetworks. These two subnetworks are known as CSPDarknet53 and are the backbone structure of YOLO v4 architecture. Each CSPDarknet53 subnetwork consists of more than 17,800 convolutional filters and 2.76 million parameters. The 3rd subnetwork is the object detection block part of Yolo v4. In this subnetwork, all ICs of the two prior subnetworks are integrated on the basis of their common labels.

In the first subnetwork, the characteristics of eighteen PCs, including four names of the four pilot features, four positions of the origins of these features, and ten dimensions observed in the image, are extracted. In the second subnetwork, the characteristics of the next sixteen PCs (dimensions, texts, and symbols) are extracted. These include ten digits from zero to 9 and six symbols consisting of parentheses, decimal points, positive and negative signs, the word feature, comma signs, and radii denoted by p, d, t, f, c, and r, respectively. It should be noted that in each of these two independent subnetworks of MFR-Net, the BCs are first extracted from the input image data. The ICs are then identified by using the recognized BCs. Finally, 18 primary characteristics in the first subnetwork and 16 primary characteristics in the second subnetwork are identified. Each of these 34 PCs is called a class as an independent identifiable object. In the third part of MFR-Net, the two mentioned subnetworks are combined to conclude the complete specifications of machining features. The output is presented in a table that can be used in other steps of the computer-aided process planning.

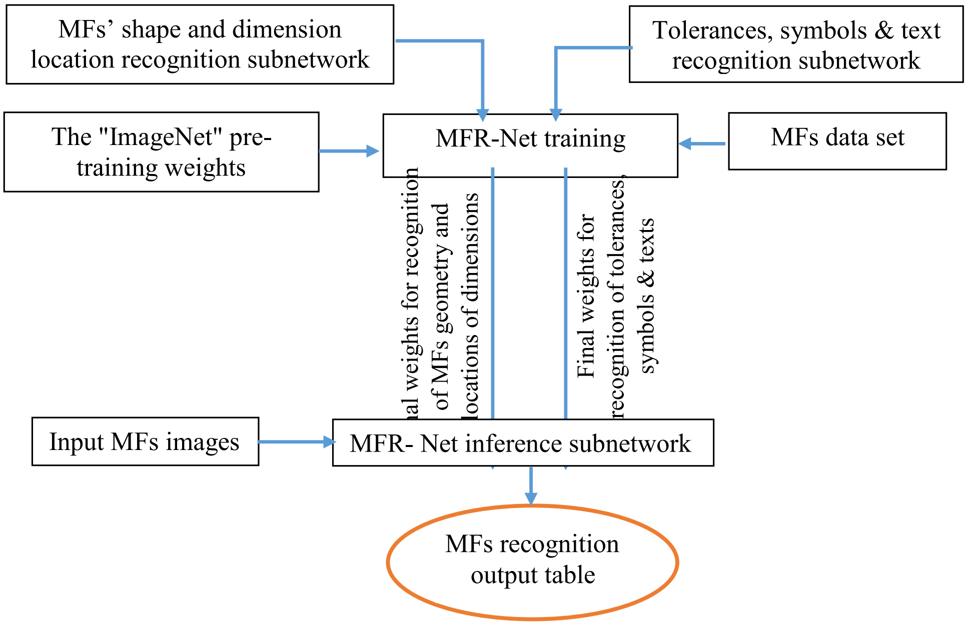

MFs' shape and dimension location recognition subnetwork
Tolerances, symbols & text recognition subnetwork
The "ImageNet" pre-training weights
MFR-Net training
MFs data set
al weights for recognition of MFs geometry and locations of dimensions
Final weights for recognition of tolerances, symbols & texts
Input MFs images
MFR- Net inference subnetwork
MFs recognition output table

Fig. 1 The schematic architecture of MFR-Net

The overall architecture of MFR-Net is schematically illustrated in Fig. 1.

For training MFR_Net, the shape and location of 18 primary characteristics are first defined and labeled. For example, the location and shape of the fillet machining feature is labeled as "Fillet." All information related to the fillet MF is specified by a name and a "_" sign below it. For example, fillet radius is labeled as r_fillet, and its location in the image is determined. While programing MFR_Net, the input data to the 2nd subnetwork are just labels with the "_" sign, which states this region of the image contains some texts, numbers, and signs. For example, r_fillet and d_fillet denote the radius and depth of fillet MF.

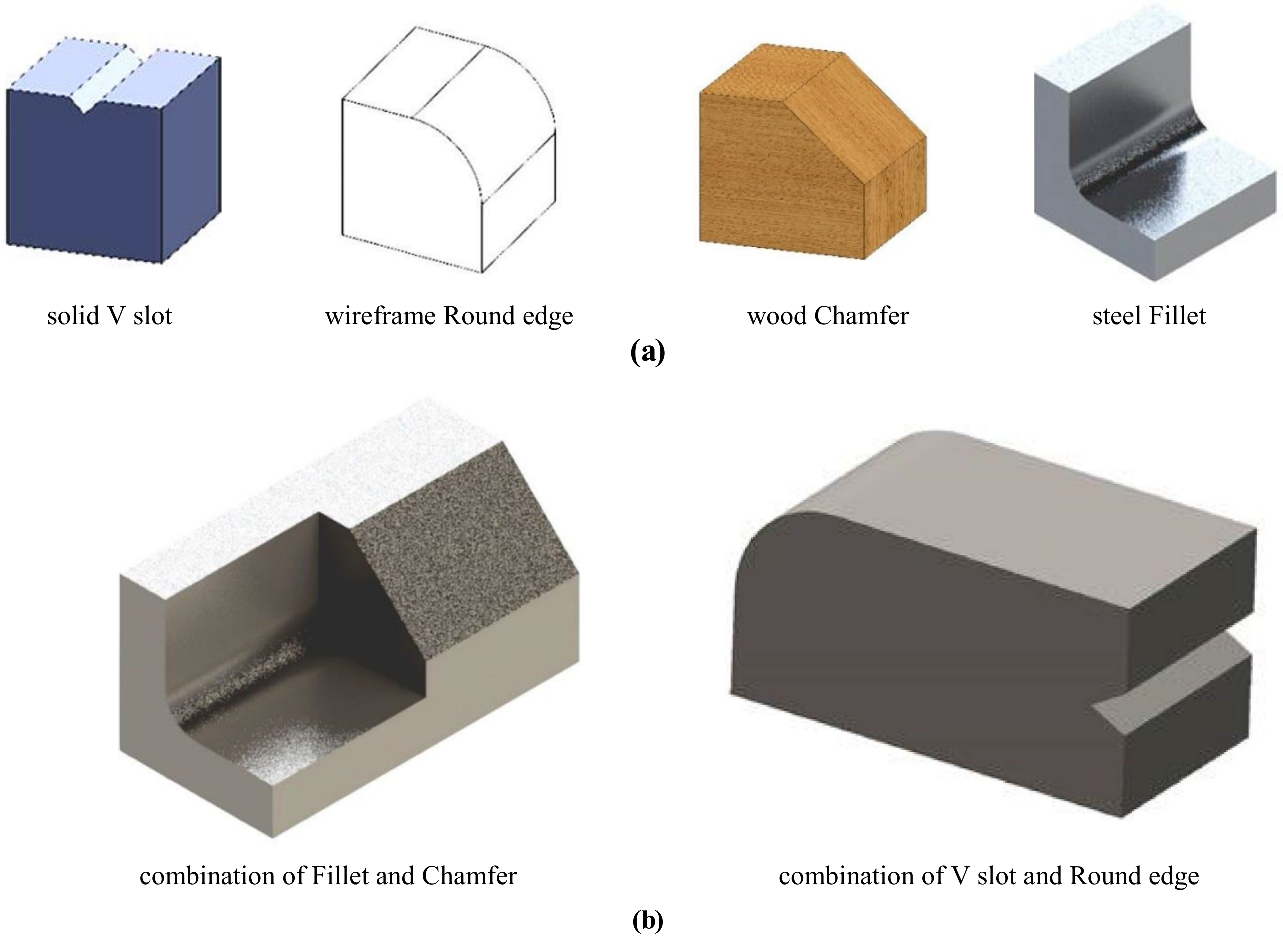
2.2 Data preparation

2.2.1 Preparing a database of machining feature images

Data preparation is one of the main steps of developing deep learning networks. To prepare the required database in the present study, 50,000 images in wireframe, isometric, diametric, and ordinary image formats (photos) of machining features and different combinations of them were prepared from different view angles and with different materials by SolidWorks software.

More than 1000 images of the database were selected for preparing the data required for training the subnetwork to identify 4 pilot MFs' shapes and locations of their dimensions. More than 1000 images of numbers, symbols, and letters were selected for teaching the second subnetwork (tolerances, symbols, and text recognition subnetwork). Figure 2 shows the images of the four pilot and two composite machining features with different materials, which are represented in different formats.

Since the sizes of numbers, dimensions, and texts were very small compared to the sizes of the shapes of machining features, the development of two parallel subnetworks for simultaneous recognition of shapes, numbers, dimensions, and texts in a single image encountered problems. A new technique has been used to solve this problem. In this technique, the regions of images that include numbers,



solid V slot

wireframe Round edge

wood Chamfer

steel Fillet

(a)

combination of Fillet and Chamfer

combination of V slot and Round edge

(b)

Fig. 2 Machining features: **a** 4 pilot MFs and **b** combination features

dimensions, and texts are first identified and then separated from the image. These regions are cropped, magnified, and labeled in the network's database of numbers, letters, and symbols.

2.2.2 Labeling images of MFs

The images in the database were divided into two groups; 80% of the images were used for network training and 20% for evaluating the accuracy of the network after training.

To train "MFR-Net," the images of training available in the database should be converted into the information needed to train the network. For this purpose, the training images in the database including machining features, numbers, and symbols were labeled with the help of Python programming language. In labeling, a predetermined rectangle enclosing a specific feature is first defined. All characteristics that fall within this rectangle and are typically used for recognition of machining features are labeled as belonging to that rectangle and its enclosed machining feature. This is repeated separately for all features stored in the database. The properties of each machining feature are determined by using the pixels within the labeled box.

Since the complete identification of machining features is done by using two independent parallel subnetworks simultaneously, the required information of each of these two subnetworks has been prepared independently, so the dataset preparation network has been developed in two parts. In the first part, the machining feature, its dimensions, and positions are identified by using eighteen classes for four pilot MFs' primary characteristics and labeled. Four of these eighteen primary characteristics are used for recognition of the feature type, four characteristics for identification of the position of each feature, and ten characteristics for identification of the symbols and locations of the dimensions of these features. The ten dimensions include three for chamfer's slope's angle, length, and depth; two for fillet's depth and inner radius; two for round edge's depth and radius; and three for v-slot's angle, length, and depth. Figure 3 shows the labeling along with the identifying characteristics for four pilot MFs.

In the second part of MFR-Net's database (for identification of numbers, letters, and symbols), sixteen classes are employed including ten classes for recognition of numbers zero to 9 and six classes for recognition of prefix f, implying that the next figure is the feature's number, decimal point, comma, ± sign, radius R, and parentheses, which are labeled by f, d, c, t, r, and p, respectively. Figure 4 shows examples of tagging the images of numbers and symbols.

In these images, the background color and image size of numbers and letters vary in the database.

All the images in the database were manually tagged with the mentioned formats and turned into information in a format that can be used by the deep learning network for network training. The 34 classes illustrated in Figs. 3 and 4 are described in Table 1.

The format employed to describe the coordinates of the feature is fa (x, y, z). In this format, f is employed to indicate the direction and starting location of reading the digits and symbols. This letter is followed by the feature number, a. The coordinate of the feature is expressed by the distances of its starting point from the workpiece's zero along X, Y, Z coordinates. The workpiece zero (0, 0, 0) is coincident with the reference point of the machine tool.

After labeling all the image data in the database, this labeled image data was converted to text data by the Labelimg software and a text file in xml format was prepared for each tagged image. Then all the text data of the database were saved in two separate files in CSV format: one for the shape, format, and position of the dimensions of the machining features and the other for the dimensions, sizes, and signs in the supplementary data; data files related to each of the two networks used in the MFR-Net are presented.

In the last step of data preparation, each of the two mentioned CSV files is converted to a TF record file, which converts the database data into a binary format that can be understood by the computer.

2.3 MFR-Net training

The main functions of network training are concentrated on updating the values of network weights. In MFR-Net, such as other deep learning networks, there are two kinds of unknown parameters: hyper parameters that are determined by the network designer in the design phase and parameters that are determined automatically by the network through network training. The hyper parameters include the number of network layers, learning rate coefficient, activation function, error function, and some other ones that are determined based on the problem. In the present study, these parameters are selected based on YOLO deep learning network [30] with some changes made, for instance, in the number of primary characteristics, dimension of the input image to the network, learning rate, and some other parameters. The other hyper parameters such as activation function, number and dimensions, and filter tensors were adopted in this research after examining them in the Google virtual laboratory or the virtual graphics card of the Google website called Google Collaboratory.

The most important parameters automatically decided by the network are the weight coefficients. In this research, Google Collaboratory has been used to perform the calculations of the network training stage. The initial values of MFR-Net's weights have been adopted from the Image-Net deep learning network [31] as the default weights. Some data manipulation techniques such as image rotation and

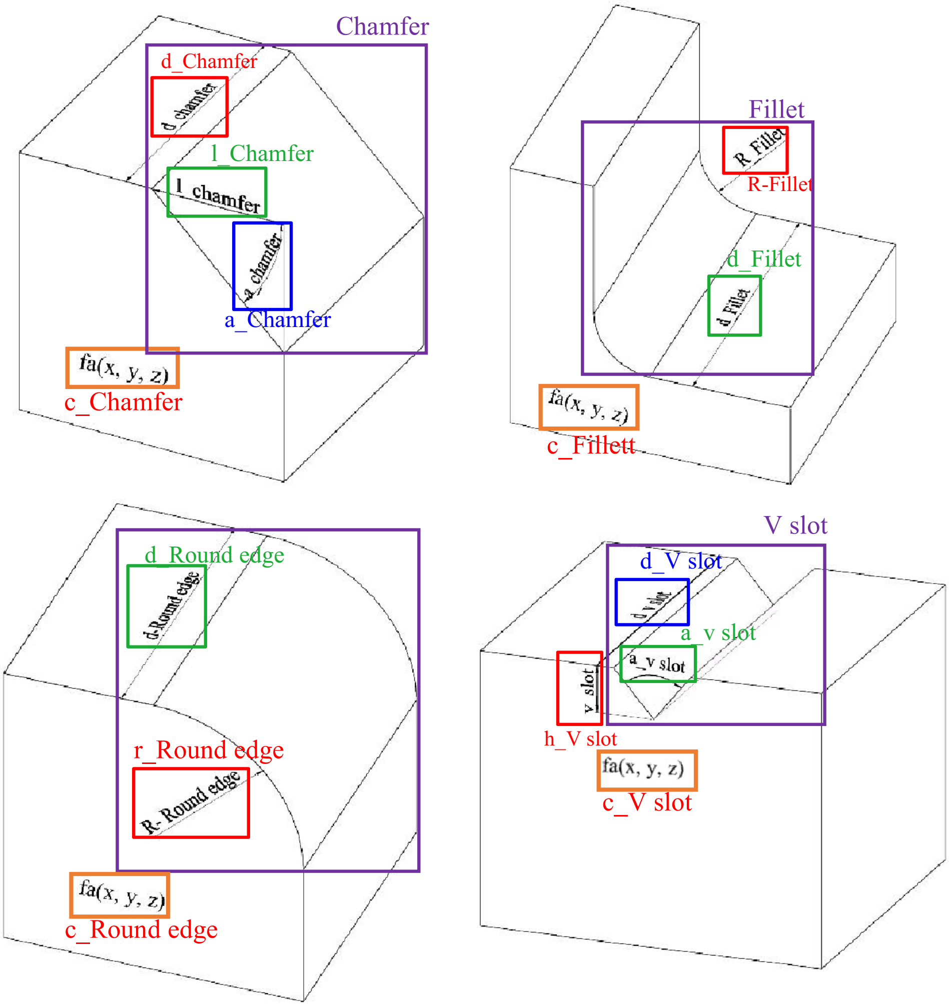


Fig. 3 Labeling of four pilot MFs and the identifying characters

random drawing of non-parallel lines were used to solve the challenging problems of network performance in recognition and increase its detection capability.

The dimensions of the PCs' search areas in this research are determined by considering the labeled training images and the nearest neighbor algorithm. Based on the nearest neighbor algorithm, the dimensions of all images labeled in the database were first divided into nine clusters. Then, the average length and width of each cluster are selected as the length and width of one of the nine search boxes being employed to search for the presence of PCs in different areas of the image. Table 2 shows the dimensions obtained for the nine search boxes used in this research.

The optimal values of hyper parameters were determined by examining different functions and observing how to reduce the error and increase the accuracy of the model in identifying the machining features.

After examining some error/loss functions such as the square mean error function and the hinge loss function, the "cross entropy" loss function was selected as the most suitable one for the present research, as follows:

$$E = -\sum_{i=1}^{q} y_i \log\left(\hat{y}_i\right) \tag{1}$$

where E is the error/loss value, y_i is the ith MF characteristic value in the input image, $\hat{y}_i$ is the ith MF characteristic

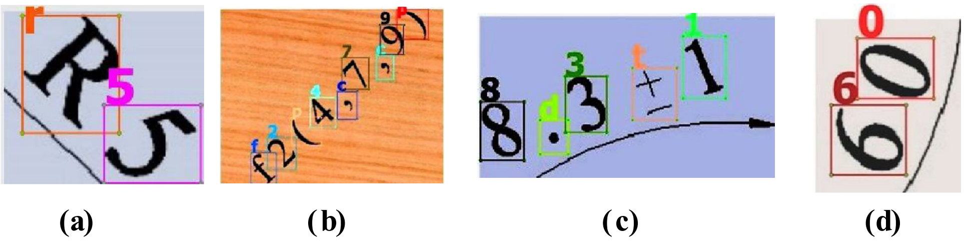


Fig. 4 Examples of labeling the numbers and symbols in the 2nd part of MFR-Net's dataset: **a** number 5 and letter R, **b** numbers 2, 4, 7, 9, and comma and parentheses, **c** numbers 1, 8, 3, dot, and ± sign, and **d** 6 and zero numbers

Table 1 Specifications of the defined classes of MFR-Net

Image	Machining feature classes	Class label	Image	Machining feature classes	Class label
Fillet (Fig. 3)	Shape	Fillet	Round edge (Fig. 3)	Shape	Round edge
	Radius	r_Fillet		Radius	r_Round edge
	Depth	d_Fillet		Depth	d_Round edge
	Coordinates	c_Fillet		Coordinates	c_Round edge
Chamfer (Fig. 3)	Shape	Chamfer	V-slot (Fig. 3)	Shape	V slot
	Angle	a_Chamfer		Angle	a_V slot
	Length	l_Chamfer		Height	h_V slot
	Depth	d_Chamfer		Depth	d_V slot
	Coordinates	c_Chamfer		Coordinates	c_V slot
Figure 4a	Number 5 and letter R	5, r	Figure 4c	Numbers 1,8, 3, decimal point, and ± sign	1, 3, 8, t, d
Figure 4b	Numbers 2, 4, 7, 9, and comma and parentheses sign	2,4,7,9, c, p	Figure 4d	6 and zero numbers	6, 0

value recognized by the network as the output result, and q is the number of defined machining feature characteristics (34 classes in this research).

The value of E is a network detection error and depends on y_i and $\hat{y}_i$. It can have infinite positive values down to zero. The closer the detection error value is to zero, the more accurate the network detection. The precision of the detection procedure is measured as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

where TP and FP denote the number of correct and false recognition of each characteristic, respectively.

The average precision of the network is obtained as follows:

$$AP = \frac{1}{34}\sum_{i=1}^{34}(Precision)_i \tag{3}$$

where 34 is the number of defined MF characteristics. Several optimization functions were subsequently selected and examined to achieve the best results to train MFR_Net. Some of them are presented in this section. On the first try, the SGD optimization function was studied as follows:

$$w_{\text{new}} = w_{\text{old}} - \beta\frac{\partial E}{\partial w} \tag{4}$$

where w_{new} is the new weights after each optimization iteration; w_{old} is the old weights before each optimization iteration; w is weights; and β is the learning rate.

The change in the learning rate decay, $\Delta\beta$, is constant in each iteration. The training curves, in this case, are illustrated in Fig. 5.

As can be seen in Fig. 5, the detection error has reached 0.95 after 3000 iterations (Fig. 5b) by adopting the "SGD" optimizer function (Eq. (4)) with fixed training rate decay (Fig. 5a). Due to zero average precision (Fig. 5c), further training of the network would not lead to any degree of acceptable success in detection of machining feature and the related parameters.

The next optimization function examined in this study was the momentum optimizer with cosine decay of the learning rate in each iterative step, expressed as follows:

$$w_{\text{new}} = w_{\text{old}} - (\frac{\beta}{1-\gamma})\frac{\partial E}{\partial w} \tag{5}$$

$$\Delta\beta = \cos(No.\ of\ iterations) \tag{6}$$

where γ is a coefficient specifying the number of iterations in the previous gradients, varying from 0.5 to 0.9 in this research. Although this optimizer has had better results than before, but the results were not sufficiently satisfactory. Figure 6 shows the network training curves by adopting Eqs. (5) and (6).

It is evident from Fig. 6 that the results are not satisfactory as far as the detection error and.

the network accuracy are concerned. The average detection accuracy has reached 0.12 (or 12%) after 3000 iterations that is not acceptable.

After examining some other optimizers, the best performance was achieved by applying the Adam optimizer with exponential learning rate decay, expressed as follows:

Table 2 Dimensions of the search boxes in pixels

1st box	2nd box	3rd box	4th box	5th box	6th box	7th box	8th box	9th box
[15×16]	[18×32]	[29×20]	[46×29]	[32×52]	[88×21]	[72×56]	[136×149]	[232×284]

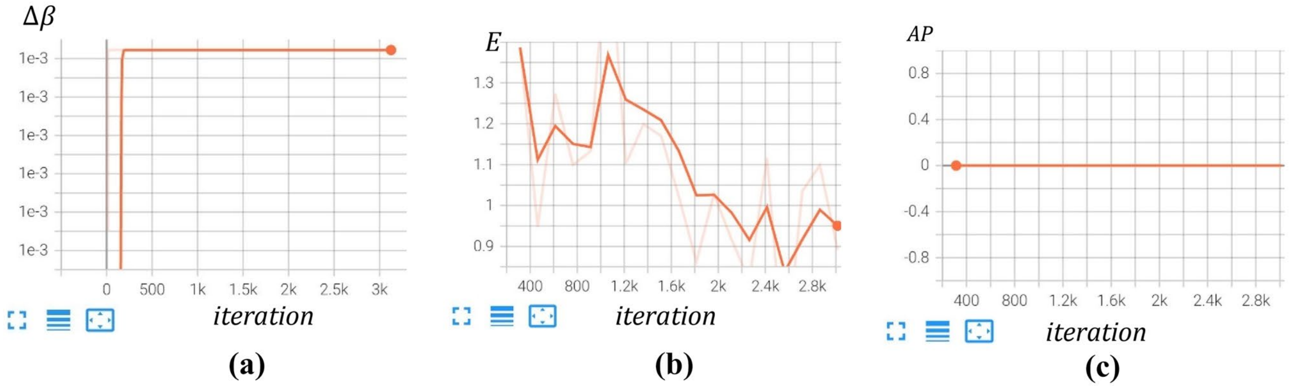


Fig. 5 Training curves in SGD optimizer for each iteration: **a** constant learning rate decay, **b** error value changes, and **c** average precision

$$w_{\text{new}} = \beta\, w_{\text{old}} + (1-\beta)\frac{\partial E}{\partial w} \tag{7}$$

$$\Delta\beta = \exp(No.\ of\ iteration) \tag{8}$$

The learning rate curves for the final hyperparameters are shown in Fig. 7. The error decreased to 0.02 after 10,000 iterations. The average precision has increased to about 85%. These are quite acceptable.

2.4 Some technical points in MFR-Net

Some technics specific to MFR-Net have been devised to assist the network in order to better identify the MFs' characteristics with reduced error in recognizing machining features and reading the symbols and numbers. These are described as follows.

As already evidenced, the letter f is employed as an indicator to assist the network in correctly reading the digits and identifying the symbols. It should be noted that reading more than one digit is naturally done from left to right, horizontally. However, a multi-digit number in an image may have been placed vertically. Therefore, to help the network in recognizing multi-digit numbers, tolerances, and symbols, the letter f is designated to signify the direction and the starting location of the reading.

In some images, one dimension may be placed on top of another. In this case, it would be difficult for the network to detect the right dimension. As the solution to this problem, the network keeps the dimension with a higher detection percentage and removes the second one when the overlap between the images of the two dimensions is more than 60%. In addition, when the number of dimensions specified in the image is more than the number of machining features in the image, the network decides to remove additional dimensions according to the type of machining feature. In case the number of dimensions of an MF is less than the required number, the network sends an inadequate-dimension error message.

Another important problem is how to assign dimensions to the relevant machining feature when there are more than one machining feature in the image. Given that the network

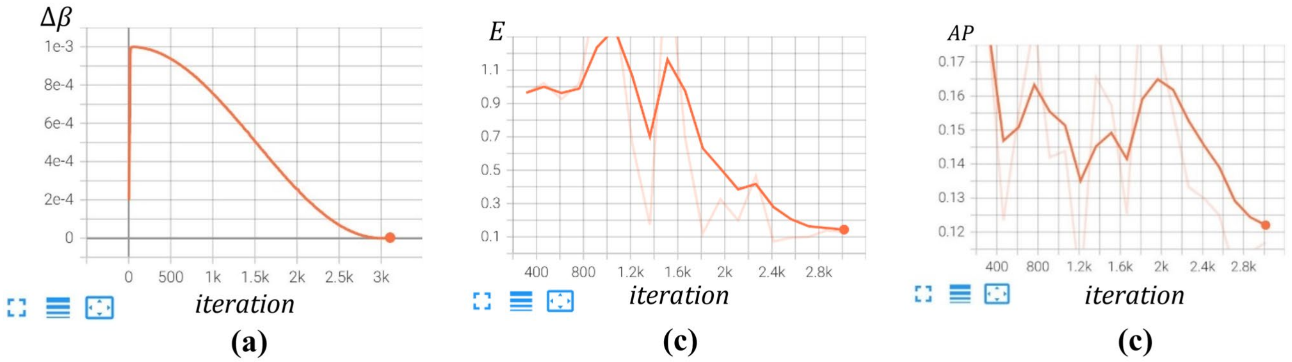


Fig. 6 Training curves in momentum optimizer for each iteration: **a** cosine learning rate decay, **b** error value changes, and **c** average precision

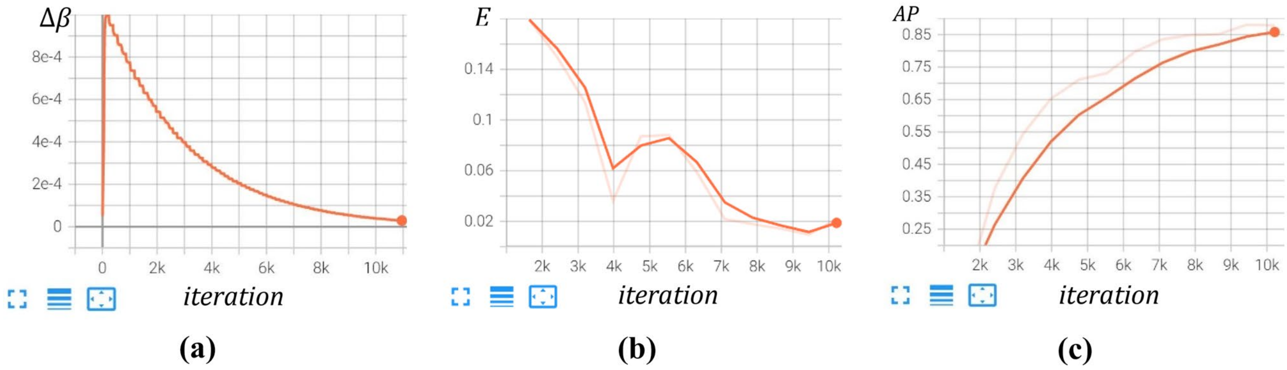


Fig. 7 Training curves in Adam optimizer for each iteration: **a** exponential learning rate decay, **b** error value changes, and **c** average precision

recognizes the shape and placement of dimensions as a separate subject (PC), it is a challenge for the network to determine which dimensions belong to each MF. The network detects the locations of all dimensions and machining features in the image and assigns the nearest dimensions to each machining feature by using the nearest neighbor algorithm.

In some images, different multi-digit numbers may be close to each other. In these cases, the size of the digits is first identified, and by drawing a hypothetical line in the average dimensions of the digits of a number, if the center of the size of a digit is 75% farther from the line, that digit of the mentioned number is not considered in the set of digits. If in a picture the letter assigned to a machining feature (f) does not appear in front of the coordinate parentheses, the network intelligently detects this and adds a letter f to the beginning of the parentheses. If two letters f are consecutively repeated, the extra letter f is deleted. When a parenthesis in the image is open, the network automatically closes the parenthesis in the output text.

3 Case study

The proposed method has been examined for quite a few cases including images with only one machining feature and complicated workpieces composed of several machining features in a single image. Some examples of these cases are presented in this section. The information in the images, based on the description in “Labeling images of MFs” (Figs. 3 and 4 and Table 1), includes such items as the type of machining feature, the dimensional and geometric information of each workpiece, and the location of these information in the image. The four different cases examined by the developed network are illustrated in Fig. 8. In Fig. 8a,b, two single machining features (fillet and chamfer) with the relevant dimensions have been recognized. The workpiece shown in Fig. 8c consists of four machining features in 3D views. Another workpiece examined by the network is illustrated in Fig. 8d that shows a 2D image of a workpiece consisting of five machining features. The MFR_Net could successfully recognize all the machining features and their relevant information. The feature information shown in Fig. 8 is summarized in Table 3.

4 Results and discussion

As shown in the case studies, MFR_Net was able to identify machining features with all the dimensions and sizes listed on the workpiece’s image. The method introduced in this research does not require repeated conversion of design data, as is the case in the traditional methods. The specifications of machining features are directly extracted from the image of the workpiece. In the methods based on a deep learning approach such as FeatureNet and MSVnet, the input data is still in B-Rep format, as in the traditional methods. In these methods, in addition to converting design file data into B_Rep format, the database models are converted from B-rep to spatial occupancy enumeration, also known as voxels, which adds one more conversion of input data to the system. The recognition process in these methods is, thus, more complicated. In previous methods including both the traditional methods and those based on deep learning approaches, some data can be lost due to the geometric interference of features. Whereas the method presented in this research is able to identify all features even with partial views of the feature’s image, in 3D or 2D representation. This can be seen in Fig. 8c,d.

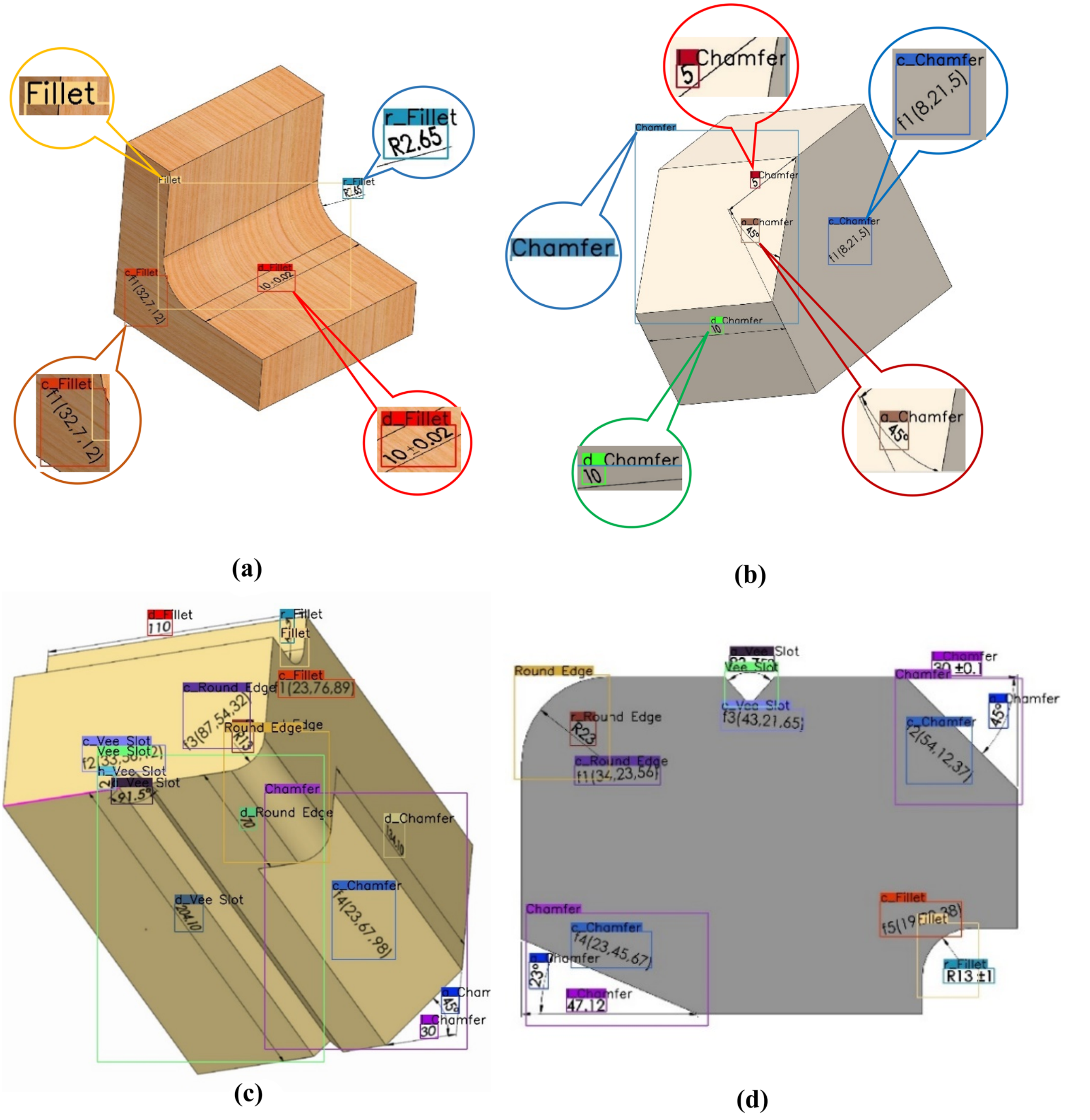


Fig. 8 MFR_Net output images: **a** fillet, **b** chamfer, **c** 3D workpiece's image with 4 MFs, and **d** 2D workpiece's image with 5 MFs

As can be seen in the case studies, all information such as the types of machining features, their dimensions, coordinate centers, and their locations in the images are recognized by MFR_Net, whereas the previous methods such as FeatureNet and MSVnet, classified as deep learning networks, are just able to identify the types of machining features.

Although both FeatureNet and MSVnet are able to identify 24 machining features, and the MFR_Net network is currently developed for the identification of 4 independent machining features, the architecture of all three subnetworks is independent of the number of identifiable features. By increasing the number of samples in the databases and retraining them, it is possible to develop networks to identify any number of machining features.

Table 3 The parameter values of machining features recognized by the network and illustrated in Fig. 8

Image no.	Feature no.	Feature type	Coordinates (*x, y, z* in mm)	Length (mm)	Height (mm)	Depth (mm)	Radius (mm)	Angle (degree)
Figure 8a	f1	Fillet	32, 7, 11	*	*	10 ± 0.02	2.65	*
Figure 8b	f2	Chamfer	23, 76, 45	5	*	10	*	45
Figure 8c	f1	Fillet	23, 76, 89	*	*	110	13	*
Figure 8c	f2	Vee slot	33, 56, 12	*	2	204.10	*	91.5
Figure 8c	f3	Round edge	87, 54, 32	*	*	70	13	*
Figure 8c	f4	Chamfer	23, 67, 98	30	*	134.10	*	45
Figure 8d	f1	Round edge	34, 23, 56	*	*	**	23	*
Figure 8d	f2	Chamfer	54, 12, 37	30 ± 0.1	*	**	*	45
Figure 8d	f3	Vee slot	43, 21, 65	*	**	**	*	83.75
Figure 8d	f4	Chamfer	23, 45, 67	47.12	*	**	*	23
Figure 8d	f5	Fillet	19, 20, 38	*	*	**	13 ± 1	*

*Not applicable for this MF type; **Does not exist in the image

5 Conclusion

The deep learning network, MFR-Net, developed in the present study for the identification of 4 pilot machining features including round edges, fillets, chamfers, and v-slots is not limited to the use of CAD output files but can extract the features directly from the images of workpieces. In addition, MFR-Net adopts CAD files in any format, even hand drawings.

Various characteristics of four pilot machining features including their names, dimensions, tolerances, and locations are identified. The four pilot machining features include round edges, fillets, chamfers, and v-slots. Workpieces with a single or double feature are dealt with in MFR-Net. A distinguishing characteristic of the developed network is its capability to distinguish the pertinence between the primary characteristics and the machining features based on the assignment of each primary characteristic to its nearest machining feature existing in the image.

The network developed in this study can process the design of workpieces even in their initial steps of conceptual design. This provides the designer with significant assistance to develop the concept and improve the design.

The network is under development to include twenty machining features and workpieces with more than two machining features.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00170-022-09839-0.

Availability of data and material The data has been presented in the tables.

Declarations

Ethics approval The authors have respected the ethics.

Consent for publication The authors agree this paper be published in The International Journal of Advanced Manufacturing Technology.

Competing interests The authors declare no competing interests.

References

1. Cheng K (ed) (2008) Machining dynamics: theory, applications and practices. Springer, London
2. Paulo DJ (ed) (2022) Mechanical and industrial engineering: historical aspects and future directions (materials forming, machining and tribology). Springer, Heidelberg
3. Carou D, Sartal A, Davim JP (2022) Machine learning and artificial intelligence with industrial applications: from big data to small data. Springer
4. Prabhakar S, Henderson MR (1992) Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models. Comput Aided Des 24(7):381–393
5. Hao Y (2006) Research on auto-reasoning process planning using a knowledge based semantic net. Knowl-Based Syst 19(8):755–764
6. Li RK, Bedworth DD (1988) A framework for the integration of computer-aided design and computer-aided process planning. Comput Ind Eng 14(4):395–413
7. Verma AK, Rajotia S (2010) A review of machining feature recognition methodologies. Int J Comput Integr Manuf 23(4):353–368
8. Liu CH, Perng DB, Chen Z (1994) Automatic form feature recognition and 3D part reconstruction from 2D CAD data. Comput Ind Eng 26(4):689–707
9. Ismail N, Bakar NA, Juri AH (2005) Recognition of cylindrical and conical features using edge boundary classification. Int J Mach Tools Manuf 45(6):649–655
10. Datta S, Davim JP (2022) Machine learning in industry. Springer
11. Joshi S, Chang TC (1988) Graph-based heuristics for recognition of machined features from a 3D solid model. Comput Aided Des 20(2):58–66
12. Li H, Huang Y, Sun Y, Chen L (2015) Hint-based generic shape feature recognition from three-dimensional B-rep models. Adv Mech Eng 7(4):1687814015582082
13. Henderson MR, Anderson DC (1984) Computer recognition and extraction of form features: a CAD/CAM link. Comput Ind 5(4):329–339

14. Han J, Pratt M, Regli WC (2000) Manufacturing feature recognition from solid models: a status report. IEEE Trans Robot Autom 16(6):782–796
15. Staley SM, Henderson M, Anderson DC (1983) Using syntactic pattern recognition to extract feature information from a solid geometric data base. In Computers in Mechanical Engineering (pp 61–66)
16. Manafi D, Nategh MJ, Parvaz H (2017) Extracting the manufacturing information of machining features for computer-aided process planning systems. Proc Inst Mech Eng B J Eng Manuf 231(12):2072–2083
17. Meeran S, Pratt MJ, Kay JM (1993) The use of PROLOG in the automatic recognition of manufacturing features from 2-D drawings. Eng Appl Artif Intell 6(5):409–423
18. Pham DT, Sagiroglu S (2001) Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms. Int J Mach Tools Manuf 41(3):419–430
19. Korosec M, Balic J, Kopac J (2005) Neural network based manufacturability evaluation of free form machining. Int J Mach Tools Manuf 45(1):13–20
20. Yue Y, Ding L, Ahmet K, Painter J, Walters M (2002) Study of neural network techniques for computer integrated manufacturing. Eng Comput
21. Zheng X, Zheng S, Kong Y, Chen J (2021) Recent advances in surface defect inspection of industrial products using deep learning techniques. Int J Adv Manuf Technol 113(1):35–58
22. Lim ML, Derani MN, Ratnam MM, Yusoff AR (2022) Tool wear prediction in turning using workpiece surface profile images and deep learning neural networks. Int J Adv Manuf Technol 6:1–8
23. Kumar SL (2017) State of the art-intense review on artificial intelligence systems application in process planning and manufacturing. Eng Appl Artif Intell 65:294–329
24. Zhang Z, Jaiswal P, Rai R (2018) Featurenet: machining feature recognition based on 3D convolution neural network. Comput Aided Des 101:12–22
25. Shi P, Qi Q, Qin Y, Scott PJ, Jiang X (2020) A novel learning-based feature recognition method using multiple sectional view representation. J Intell Manuf pp 1–19
26. Peng B, Xia H, Lv X, Annor-Nyarko M, Zhu S, Liu Y, Zhang J (2021) An intelligent fault diagnosis method for rotating machinery based on data fusion and deep residual neural network. Appl Intell pp 1–15
27. Moreno-García CF, Elyan E, Jayne C (2019) New trends on digitisation of complex engineering drawings. Neural Comput Appl 31(6):1695–1712
28. Zhao Y, Deng X, Lai H (2020) A deep learning-based method to detect components from scanned structural drawings for reconstructing 3D models. Appl Sci 10(6):2066
29. Davim JP (2009) Machining: fundamentals and recent advances. Springer
30. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition pp 779–788
31. Sengupta S, Basak S, Saikia P, Paul S, Tsalavoutis V, Atiah F, Ravi V, Peters A (2020) A review of deep learning with special emphasis on architectures, applications and recent trends. Knowl-Based Syst 194:105596

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