

©2020, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International <http://creativecommons.org/about/downloads>



Manufacturing Cost Estimation Based on a Deep-learning Method

Fangwei Ning¹, Yan Shi ^{1*}, Maolin Cai ¹, Weiqing Xu ¹, Xianzhi Zhang ^{2*}

1 School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;

2 Department of Mechanical Engineering, Kingston University, London, SW15 3DW, UK

nfangwei@163.com ; caimaolin@buaa.com.cn; weiqing.xu@buaa.edu.cn;

* Correspondence: yesoyou@gmail.com ; Tel.: 86-10-82335821 (Y.S.); X.Zhang@kingston.ac.uk.

Abstract:

In the era of the mass customisation, rapid and accurate estimation of the manufacturing cost of different parts can improve the competitiveness of a product. Owing to the ever-changing functions, complex structure, and unusual complex processing links of the parts, the regression-model cost estimation method has difficulty establishing a complex mapping relationship in manufacturing. As a newly emerging technology, deep-learning methods have the ability to learn complex mapping relationships and high-level data features from a large number of data automatically. In this paper, two-dimensional (2D) and three-dimensional (3D) convolutional neural network (CNN) training images and voxel data methods for a cost estimation of a manufacturing process are proposed. Furthermore, the effects of different voxel resolutions, fine-tuning methods, and data volumes of the training CNN are investigated. It was found that compared to 2D CNN, 3D CNN exhibits excellent performance regarding the regression problem of a cost estimation and achieves a high application value.

Keywords: manufacturing, price quotation, cost estimation, deep learning, CNN

1. Introduction

In the context of customised mass production, the costs of the parts need to be estimated quickly and accurately. Responding to customer inquiries has always been a concern of mechanical part processing enterprises, which must ensure a relatively low product cost to survive under extreme competition. Although the cost of the part design stage only accounts for approximately 5% of the total cost of product development, it determines 70%–80% of the total cost of the mechanical parts [1]. Therefore, an

accurate estimation of the cost of product parts during the design stage can optimise the design and improve the competitiveness of the product.

A cost estimation is a quantitative estimation of the resource costs required for part processing. Differing from a cost calculation, a cost estimation is based on the premise that the company does not obtain the production schedule or data of the manufacturing process, and can be divided into direct and indirect costs. Direct costs are identified as the costs of parts used in production, such as design costs, material costs, and processing costs. Indirect costs refers to costs related to the product that are not directly identified, such as the plant lease, depreciation rate of the equipment, and sales and administrative expenses. A cost estimation is a typical mathematical regression problem, meaning a process of predicting events by building a complex model of the complex relationships in the sample. Since the concept of a regression model was first proposed, different regression models have emerged, including linear regression, ridge regression, and logical regression [2]. The existing cost estimation methods based on a regression model can be divided into two categories: parameter fitting and analysis estimation methods.

The principle of a parameter fitting estimation method is to compensate the data information of the products to be sold according to the structural information and cost information of similar products, and to obtain the cost statistics information for estimating the mathematical expression of the quotation of the products through the fitting relationship between the parameters. This method requires analysing the various links of the commodity cost in detail. Under a low estimation workload, a cost quotation estimation can be obtained; therefore, the method is widely used in the early stage of a product design and can solve the problem of an early cost estimation. The disadvantage of this method is that the accuracy of the cost estimation is low. Mature algorithms include regression analysis estimation, functional cost estimation, and learning curve estimation.

A regression analysis estimation is a statistical analysis method for determining the quantitative relationship between two or more variables. Rickenbacher et al. [3] proposed a statistical approach for the time estimation to complete a building job. In this

approach, the model was based on a linear regression analysis. The regression coefficients were estimated according to previous building jobs. The calculated building costs were split according to the volume and building height of the parts to obtain the costs for a single part. Mileham and Currie [4] studied a parametric model of a production cost estimation during the product design stage. The key was to transform the design parameters into cost characteristic parameters through a multiple regression analysis, thereby estimating the cost using the mapping function of such an analysis. This method is well suited for a cost estimation of an injection model.

A functional cost estimation [5] is a method for estimating the value of a product according to its function. French and Folley [6] studied various cost estimation methods for pressure vessels and rolling bearings. They believe that the cost of a product is determined by its functional characteristics. The functional characteristics of a product were determined according to the design parameters. According to the principle of a functional cost method, the cost of a product was estimated using such parameters.

As the principle of the learning curve estimation method [7], the work efficiency increases according to a certain ratio, and the working time per unit task exhibits a decreasing curve. The product cost can be estimated from the law of the output and product cost. Azzouz et al. [8] developed a classification scheme that characterises the different scheduling problems under learning effects, and compared different modelling approaches and solution algorithms through a literature review.

Another category is the analysis estimation method. The principle this is to analyse the activity of the product lifecycle in detail, evaluate the “motivation” of the cost during the process of this activity, and apply a product cost estimation according to the “motivation” of the cost. This method can achieve a higher estimation accuracy. However, it is based on complete information of the product eigenvalues and is suitable for the late stage of a product design or after design completion. Common cost estimation algorithms include feature-based and process-based estimation methods, non-parametric cost estimation methods, and activity-based estimation methods.

As the principle of a cost estimation method based on feature and processing technologies, the process of a quotation involves adding features that influence the

quotation. Each process of a product is equivalent to a feature, such as the tooling, labour costs, transportation costs, packaging costs, the manufacturing parameters, and the processing equipment. By estimating the corresponding manufacturing costs, an accurate estimation of the products can be quoted. According to Ji [9], a case-based reasoning (CBR) revision model was developed to predict the construction costs through feature counting. The formula applied in the model is a mathematical equation that improves the prediction accuracy by applying in advance the error value of the cost. The prediction effect of this method is extremely high. Rudolph and Emmelmann [10] introduced a cloud-based platform for additive manufacturing that analyses the geometry of a part, including a determination of the volume, surface area, and dimensions. These characteristic factors are used for the cost quotation, which is implemented based on the Standard Triangulation Language format.

The non-parametric cost estimation method [11] is based on previous operations, experience of the decision makers, and the planning, calculation, demonstration, data analysis, and processing. Further, the manufacturing costs can be obtained through a regression analysis of the products and by using a statistical algorithm. Juszczuk et al. [12] investigated issues of a cost estimation based on statistical methods using artificial neural networks (ANNs). They presented a concise comparison of the parametric and nonparametric approaches that require neither assumptions regarding the functional relationships nor an investigation into the rules.

Activity-based costing (ABC) [13] takes the basic principle of a “product consumes activity, an activity consumes resources.” It identifies and measures all activities through which an enterprise consumes resources, calculates the costs of the resources consumed for each activity, and estimates the cost of the products. ABC has a high estimation accuracy, although the estimation must be conducted after product design is completed. Time-driven ABC was developed to overcome the problem in which ABC models either become overly complex or are unrealistically simple, which are the reasons why ABC is no longer universally used and why it has been abandoned by some companies [14]. Wouters and Stecher [15] developed a real-time product cost measurement. This approach is used for calculating the rates and non-productive time,

as well as cases involving a mix of labour and machine times. The cost per unit of time combines both the cost per labour hour and the cost per machine hour, according to the operational relationships between the machines and operators.

Traditional regression models describe complex mapping relationships with a small number of samples. Table 1 provides a comparison between different models. In an application, the diversification of the part designs, the processing and manufacturing links, and an extraction of high-level data characteristics in each link present significant challenges to a traditional cost estimation method based on a regression model, mainly through the following two aspects. ① It is difficult to model complex mapping relations. The increases in the quantity and dimensions of the data make the mapping relationship complex. ② The robustness of the data feature expression is poor. Because of the complexity of the actual model structure used in an application, the artificial design features applied in traditional regression models can only deal with changes to a single condition.

Table 1 Comparison of various estimation methods

Model	Method type	Method name	Applicable stage	Estimation accuracy	Method difficulty	Product information needed
Regression model	Analysis estimation methods [9-15]	Activity-based estimation method [13-15]	Completion phase	High	Difficult	A large amount of information, including parts design and manufacturing information.
		Feature-based and process-based estimation method [9, 10]				
		Non-parametric cost estimation method [11, 12]				
	Parameter fitting estimation methods [3-8]	Regression analysis estimation [3, 4]	Early stage of design	Low	General difficulties	More part manufacturing information.
		Functional cost estimation [5, 6]				
		Learning curve estimation [7, 8]				

As a newly emerging technology, the deep-learning method has the ability to learn complex mapping relationships and high-level data features from a large number of

data automatically. Therefore, a regression model based on a deep neural network can effectively reduce the impact of the aforementioned problems on data prediction and improve the accuracy of the regression.

In this study, a part manufacturing cost estimation was investigated through a deep-learning technology. The rest of this paper is constructed as follows. Firstly, CNN methods are reviewed in Section 2. Next, the generation of image data, voxel data, and a data enhancement method are shown Section 3. In Section 4, methods for training images and voxel data of the parts in a 2D CNN and a 3D CNN are presented. For a 2D CNN, the effect of freezing different convolution layers on the training network is studied by fine-tuning the network method. For a 3D CNN, the influence of training a CNN by different resolutions of the voxel data is investigated. In Section 5, cost estimation methods based on a 2D CNN and a 3D CNN are discussed, and their advantages and disadvantages are compared.

2. CNN

The regression models of deep neural networks are divided into two types according to the model structure. The first is based on deep neural networks, which establish the regression relationship between the input data and the output data and realise a direct regression prediction [16]. The second involves recognising the model features using a deep neural network followed by the construction of a regression model [17]. According to the different methods of regression, regression models can be divided into ensemble and other types of learning [18]. The integrated learning model is a method of training multiple learners and combining them, which usually achieves better prediction results in practice than a single learner. The latter models can diversify the multi-objective task realisation models and flexibly set up learners for different learning objectives. This method also achieves good results. In addition, multiple-source regression, sequential regression, adaptive regression, and other regression models are widely used.

The application range of a deep neural network has been extended from a 2D model to a 3D model, i.e., from the original 2D image classification to the object detection and

visual search of the 3D object model, which introduces a wider range of applications. Compared with 2D models, 3D models have more complex representation rules. Most of the methods used to generate geometric data depend on the intermediate representations of the 3D shapes, such as point clouds, voxels, depth maps, and RGB-D.

Polygonal meshes [19] and point clouds [20] in a 3D model have difficulty describing the internal features, and they only describe the external shape. In addition, there are difficulties in data processing. RGB-D [21] is an image or image channel containing information related to the distance from the surface of the object to the vantage point. This method needs to determine the existing area of the object in advance, which consumes computer resources. Depth maps [22] do not directly represent 3D maps, and it is difficult to express the structure of a 3D object directly. Moreover, information is lost owing to occlusion problems. Voxels [23, 24] describe the space occupancy of objects in a very simple data form, which is extremely suitable for existing learning methods [25, 26]. Wu et al. [27] proposed a 3D ShapeNet, which inputs 30^3 resolution voxel data into a simple five-layer convolution depth confidence network. A total of 150,000 3D models are classified into 660 categories. As a beginning stage of the deep learning of 3D models, researchers are paying increasing attention to this method, despite the simple structure and low accuracy of 3D ShapeNet. Later, Maturana et al. [28] proposed VoxNet, which uses binary voxel grids and a corresponding 3D CNN architecture for a digital geometric analysis of 3D models. Compared with 3D ShapeNet, VoxNet achieves better recognition results. As the advantages of these methods, they can process 3D data from different sources, including LiDAR point clouds, RGB-D, depth maps, and polygonal grid data models. More importantly, these studies prove that CNNs can extract objects, similar to 2D data. The 3D structural features of the volume widen the application range of a CNN. Thus, more forms of 3D data have begun to use CNN learning.

2.1 2D-CNN

A CNN is a feedforward neural network with convolutional computations and a deep structure, and is one of the representative algorithms of deep learning [29,30]. Its artificial neurons can respond to the surrounding area of a portion of the coverage and achieve an excellent performance for large 2D images and 3D model processing. Fig. 1 shows a typical CNN structure consisting of three convolution layers, three pooling layers, and two fully connected layers. The convolutional layer cooperates with the pooling layer to form multiple convolution groups, extracting features layer-by-layer. Regression is achieved through several fully connected layers.

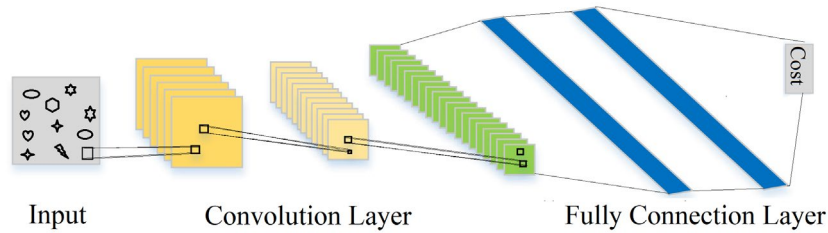


Fig. 1 Structure of the 2D CNN

For the regression problem, the CNN output is a continuous value, and the activation function is added to the linear regression. The activation function used in this network is the rectified linear unit (ReLU) [31], as shown in Fig. 2. Compared with other activation functions, such as sigmoid, tanh, and softplus, the ReLU has the following advantages: For linear functions, the ReLU is more expressive, particularly in deep networks; the gradient of the nonnegative interval is constant and thus there is no vanishing gradient problem; and the convergence speed of the model is maintained in a stable state.

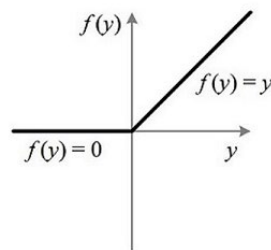


Fig.2 Activation function of ReLU

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1)$$

As the number of layers of the CNN model and the complexity of the model increase,

the error rate of the model decreases. However, training a complex CNN requires a large amount of input information and can take several days. Transfer learning can solve the problem of a long training time [32]. VGG16 [33] was used to investigate the relationship between the depth of the CNN and its performance. By stacking small 3×3 convolution cores and maximum 2×2 pooling layers repeatedly, a CNN of 16–19 layers was successfully constructed. The network structure is shown in Fig. 3.

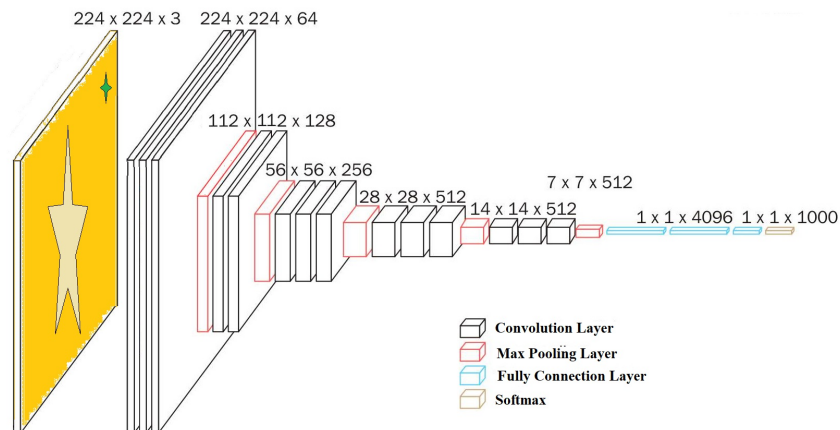


Fig. 3 The network structure of VGG16

2.2 3D CNN

The earliest application of a 3D CNN was to recognise the actions of people in a video. A one-dimensional feature was added to the 2D convolution core to extract its temporal characteristics. For the 3D model shown in Fig. 4, the dimensions of the model are similar to those of the 2D model plus the time series; thus, a 3D CNN has a better learning ability for a 3D model. The key problem of 3D model recognition and classification is extracting the feature description of the model and achieving model recognition and regression by comparing the similarity of the features. Therefore, the 2D CNN used to extract the deep features of the images is extended to extract the features of the 3D data.

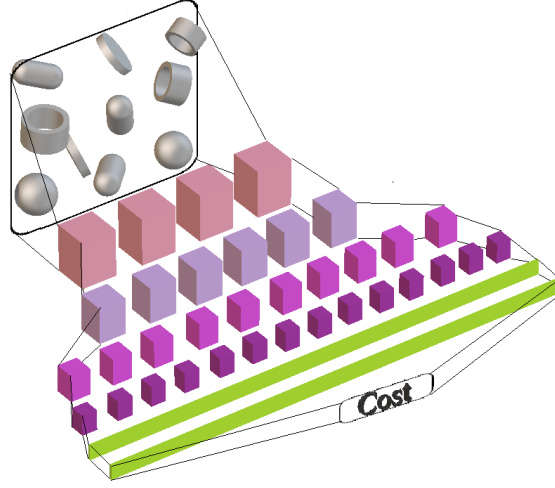


Fig. 4 Structure of the 3D CNN

Adam optimisation is used to adjust the learning rate in this study. The Adam optimisation algorithm is an extension of the stochastic gradient descent algorithm. It has recently been widely used in deep-learning applications, particularly in computer vision and natural language processing [34]. The Adam algorithm adaptively adjusts the learning rate of each parameter through a first-order data estimation and second-order moment estimation of the gradient.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (4)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (5)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (6)$$

m_t and v_t represent the first and second moment estimates of the gradient, respectively. m_{t-1} and v_{t-1} represent the first and second moments at time $t-1$. g_t represents the parameter gradient θ at time t , and \hat{m}_t and \hat{v}_t represent the corrections of m_t and v_t , respectively. $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

The random gradient descent method randomly extracts a group of samples and updates the parameters of the samples.

$$\theta = \theta - \eta \nabla_{\theta} \mathcal{J}[\theta, x^{(i)}, y^{(i)}] \quad (7)$$










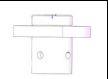
$\mathcal{J}(\theta)$ represents the minimization objective function, η represents the learning rate, and $x^{(i)}$ and $y^{(i)}$ correspond to the label.

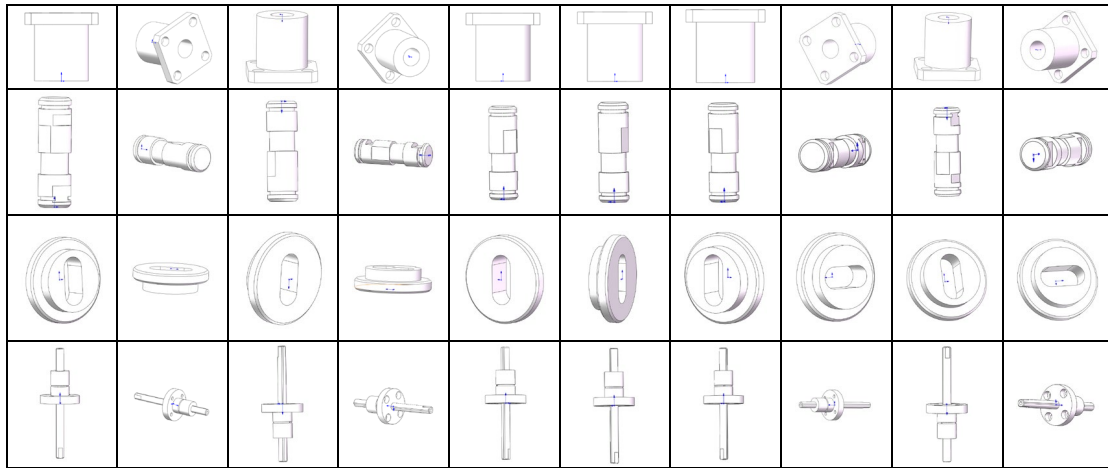
3. Data Preparation

3.1 Image data

The input of the 2D CNN is a picture of a mechanical part. The size of the image is $224\text{PX} \times 224\text{PX}$ in the format of JPG, and the pixel depth is 24. It uses a single white background, highlighting the outline of the model with the edge line color, with automatic completion of the photographic function. To obtain more training data with changes and improve the learning efficiency of the network, the data of the image are enhanced. Relevant data-enhancement methods are as follows. ① Picture inversion: pictures are rotated 90° , 180° , and 270° along the X , Y , and Z axes, respectively, to generate pictures and complete the inversion at the same time. Some part pictures are presented in Table 2. ② Scale: the resolution of RGB and depth maps is scaled in equal proportion to the scale factor $s \in [1, 1.5]$, and the depth stored in the depth maps is divided by s . ③ Color jittery: scaling factors $k \in [0.6, 1.4]$ for the brightness, contrast, and saturation of color images. ④ Color normalization: after the mean value is subtracted from the color image and the result is divided by the standard deviation, the normalized data obey the standard normal distribution. Scaling and rotation use nearest-neighbor interpolation to ensure that no new depth values are generated.

Table 2 Data Enhancement Implementation of image Reversal

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



3.2 Voxel data

(1) Data voxelisation

Voxelisation involves transforming the geometric representation of the model into the voxel representation closest to the model and generating voxel datasets. The voxel datasets not only contain the surface information of the model, they also describe the internal attributes of the model. The spatial voxels of the model are similar to the 2D pixels of the image, except that they are 3D cubic units rather than 2D points.

First, the initial 3D model is divided into an octree structure until the specified resolution is reached. The voxel data corresponding to the surface elements of the model are obtained under this precision and are marked as 1, and the other non-surface elements are marked as 0. Then, along the positive and negative axes of the model (X , Y , and Z), each line of the model elements is scanned. If the value of the element is 0, the element is outside the model, and the voxel value is changed to 1. If the value of the element is 1, the line scanning is stopped, and the next line scanning is started. After positive and negative X -, Y -, and Z -axis scanning, the surface and external element values of the model are marked as 1. The internal element is marked as 0, and the voxelisation of the model is completed. A flowchart of the voxelisation is shown in Fig. 5.

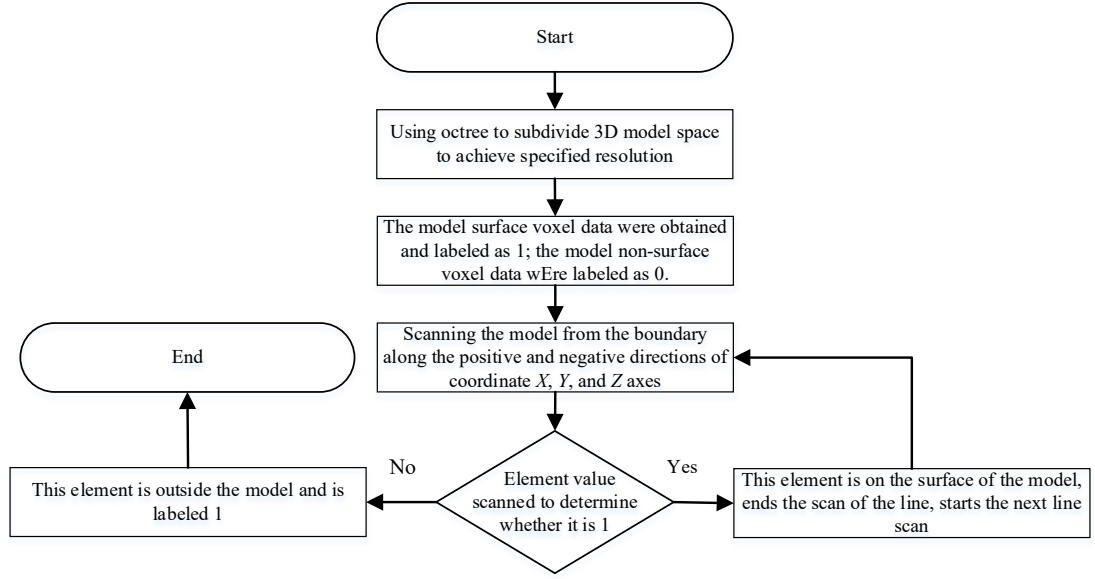


Fig. 5 Voxelisation flowchart

(2) Data enhancement

To reduce the computational complexity of the model training, data enhancement is completed during the voxelisation process. The number of rotating θ copies of the model is determined based on the rotation angle n . Each model is then rotated around the X -, Y -, and Z -axes to generate a new model copy through the rotation transformation matrix R . The rotation formula is as follows:

$$n = \frac{360^\circ}{\theta} \quad (8)$$

$$M' = R \cdot M \quad (9)$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Here, M represents the source model, and M' represents the replica of the target model.

Different voxel resolutions can describe parts with different levels of detail. A higher resolution yields a better part description effect. Voxel resolutions of 64^3 , 128^3 , and 256^3 were employed in this study. The effects of the voxel data enhancement and the different voxel resolutions are shown in Tables 3–5.

Table 3. Implementing results of data enhancement with voxel resolution of 64^3

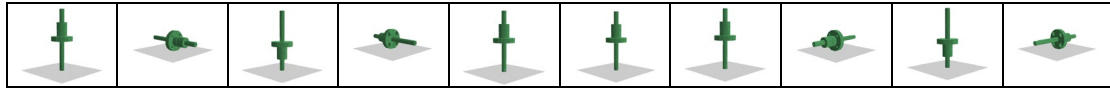
Source model	Rotate 90° around the X-axis	Rotate 180° around the X-axis	Rotate 270° around the X-axis	Rotate 90° around the Y-axis	Rotate 180° around the Y-axis	Rotate 270° around the Y-axis	Rotate 90° around the Z-axis	Rotate 180° around the Z-axis	Rotate 270° around the Z-axis

Table 4. Implementing results of data enhancement with voxel resolution of 128^3

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

Table 5. Implementing results of data enhancement with voxel resolution of 256^3

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



4. Experiments

In this study, a CNN was used to estimate the manufacturing costs of the parts. There were more than 12 types of parts, including a guide shaft, guide-shaft bearing, positioning guide shaft, fixing ring, roller, insertion pin, and metal gasket. Some of the parts are shown in Table 6. The dataset was split into three subsets: training, validation, and testing datasets (60%, 20%, and 20%, respectively). The model was trained using the training dataset and validated using the validation dataset. The specific amount of data is shown in Table 7.

Table 6 Some of the parts in this work



Table 7 Training data

	Image files	Voxel data 64 ³	Voxel data 128 ³	Voxel data 256 ³
Training data	72594	72594	72594	72594
Validation data	24198	24198	24198	24198
Testing data	24198	24198	24198	24198

In this study, Google's deep-learning framework TensorFlow was applied. The experiment was conducted using the Ubuntu 18.04 operating system, and the computer had the following specifications: an i9 graphics processor, GeForce 2080Ti memory (11 GB), and a 1.2-TB hard drive. The CNN data flow was designed in TensorFlow using Python. The image files and voxel data were input into the 2D CNN and 3D CNN as training data for estimating the part manufacturing costs. The convolution kernel was initialised by a random decimal. Table 8 shows the parameters of the network structure and the training hyperparameters applied for each experiment.

Table 8 Network structure parameters and training hyperparameters

	64		128	256	First 24 Layers freezing	
Structure	Layer (type)	Output shape	Output shape	Output shape	Layer (type)	Output shape
Convolution	Con2d_1	1,64,64,32	1,128,128,32	1,256,256,32	Con2d_1	224,224,64
	Con2d_2	1,32,64,64	1,64,128,64	1,128,256,64	Con2d_2	112,112,128
	Con2d_3	1,16,32,128	1,32,64,128	1,64,128,128	Con2d_3	112,112,128
					Con2d_4	56,56,256
					Con2d_5	56,56,256
					Con2d_7	28,28,512
					Con2d_8	28,28,512
					Con2d_9	28,28,512
					Con2d_10	14,14,512
					Con2d_11	14,14,512
					Con2d_12	14,14,512
					Con2d_13	25088
	Fully connected	dense	512	512	512	dense
dense		512	512	512	dense	4096
dense		1	1	1	dense	1

4.1 Training Pictures

Image files were input into the 2D CNN, as shown in Table 2. Before training, the data were normalised. In this study, the trained model VGG16 was used, and the last fully connected layer was modified as having a density of 1. The ReLu activation function was used to fine-tune the network. The effects of freezing different VGG16 convolution layers, namely the first 12, 18, and 24 layers, and using the original model on the training of the model were examined. These network layers were used for feature extractors without training, whereas the other network layer parameters were fine-tuned during training. The initial learning rate of the 2D CNN was 10^{-4} , and the minimum learning rate was 10^{-6} . When the model was not iterated five consecutive times, the multiplier coefficient of the learning rate was 0.05. When the model was not convergent 40 consecutive times, the network training was considered completely stopped. The convergence of the loss functions for each training is shown in Figs. 6–9.

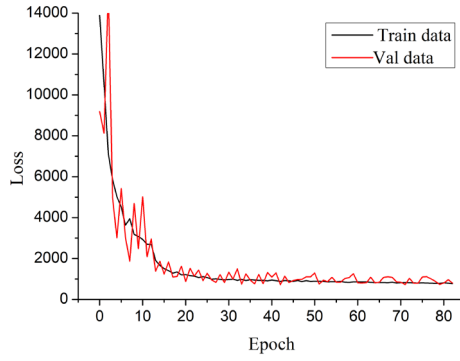


Fig. 6 Convergence of the loss functions for the original model

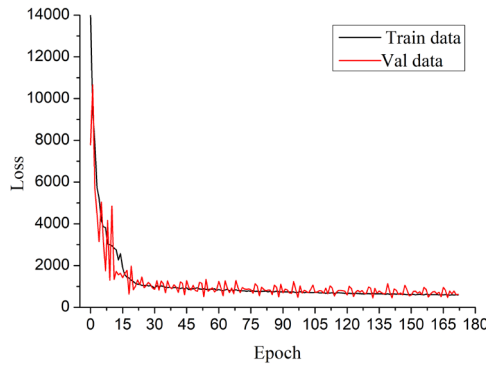


Fig. 7 Convergence of the loss functions for freezing the first 12 layers

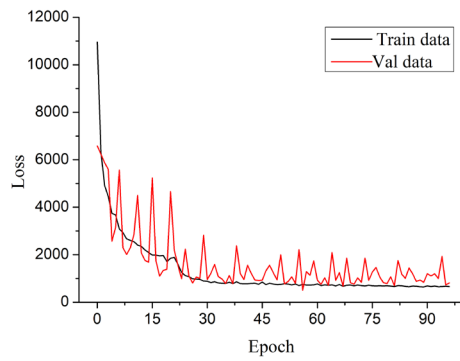


Fig. 8 Convergence of the loss functions for freezing the first 18 layers

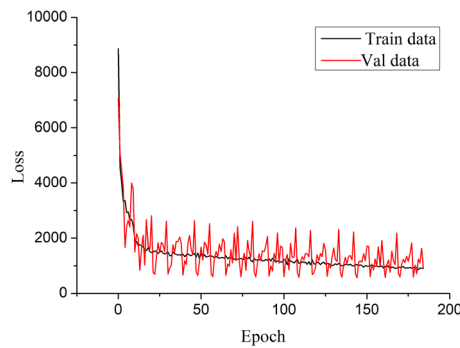


Fig. 9 Convergence of the loss functions for freezing the first 24 layers

From Figs. 6–9, it can be concluded that the losses in the training and verification sets decreased and tended to be stable with increasing iterations, indicating that the 2D

CNN has a better learning ability for the part images. The original model of the VGG16 training took the shortest amount of time, indicating that a larger number of convolutional layers involved in the feature extraction in the network yielded better network learning and a smaller range of loss for the verification set.

4.2 Training voxel data

During this experiment, voxel data with resolutions of 64^3 , 128^3 , and 256^3 were input into the same 3D CNN. Some voxel data are shown in Tables 3–5. The data were normalised prior to training. The ReLu activation function was also used in the network. The initial learning rate of the 3D CNN was 10^{-4} , and the minimum learning rate was 10^{-6} . When the model was not iterated five consecutive times, the learning rate multiplier coefficient was 0.05. If the model was not convergent ten consecutive times, the network training was considered to be completely stopped. The training results for convergence of the loss functions are shown in Figs. 10–12. Fig. 13 shows the convergence of the loss functions when the number of data were 400,000 and the resolution was 128^3 .

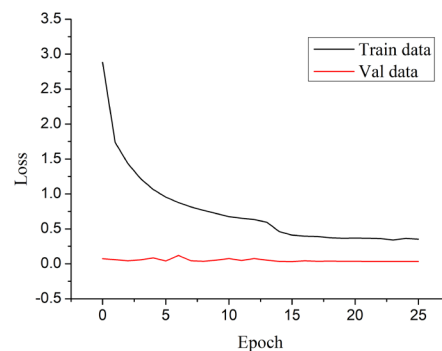


Fig. 10 Convergence of the loss functions for training with voxel data (resolution of 64^3)

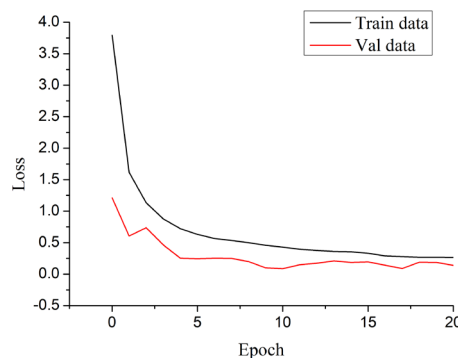


Fig. 11 Convergence of the loss functions for training with voxel data (resolution of 128^3)

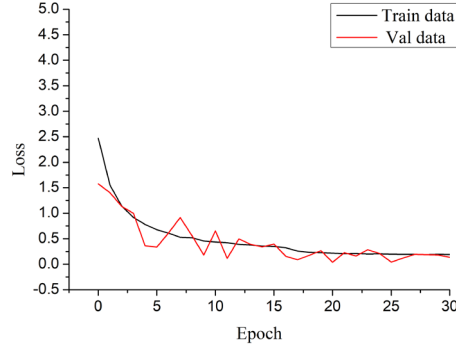


Fig. 12 Convergence of the loss functions for training with voxel data (resolution of 256^3)

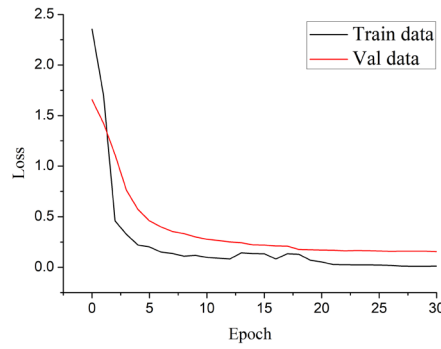


Fig. 13 Convergence of the loss functions for training with voxel data (resolution of 128^3 ; 400,000 training data)

As shown in Figs. 10–13, the loss of the training and verification sets decreased with increasing iterations and tended to be stable. Compared with the training using images of the parts, the 3D CNN converged more easily and took less time to learn the voxel data. When the voxel resolution reaches 256^3 , the voxel data consume more memory and require a longer training time. As shown in Figs. 11 and 13, as the amount of training data increased, the 3D CNN converged easily, and the loss value of the verification set maintained stability. This indicates that increasing the amount of training data can enhance the learning ability of the network.

To measure the accuracy, the mean absolute percentage error (MAPE) was adopted, which avoids the problem of a mutual cancellation of errors. Thus, it can accurately reflect the magnitude of the actual prediction errors, as shown in Fig. 14. The errors shown in the figure were calculated from the testing dataset.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (11)$$

Here, y_i represents the real price of the parts, and \hat{y}_i indicates the predicted price of the trained model.

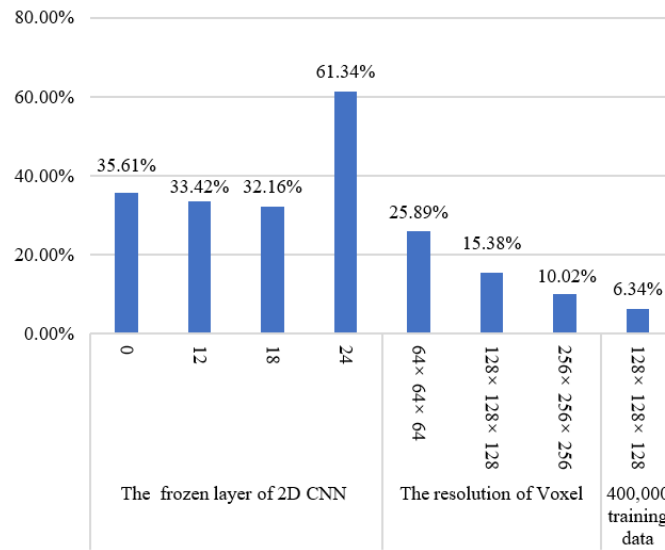


Fig. 14 MAPEs for different training data

Fig. 14 shows the MAPE values for training using the part images and voxel data, with a freezing of the different numbers of VGG16 layers, and with different voxel resolutions and training data amounts. The following results were obtained. When the VGG16 was migrated, the first 18 layers of the network were frozen. The model had the smallest MAPE and the most accurate cost estimate for the part. In addition, the 3D CNN had different learning effects for the different voxel resolutions. As the voxel resolution increased, the MAPE decreased. When the voxel resolution reached 256^3 , the MAPE value was 10.02%, indicating that a higher resolution of the voxel data yielded richer feature information of the part, which made it easier for the 3D CNN to learn the features of the part. However, an increase in the voxel resolution made the data processing and 3D CNN network training more difficult. In addition, a larger training data volume yielded a smaller MAPE. When the number of training data reached 400,000, the MAPE of the model was 6.34%. The model was less accurate when the amount of training data was 72,594.

In general, the accuracy of the method used for estimating the manufacturing cost of the parts when applying a 2D CNN was far lower than that of the method used for learning the voxel data of the part with a 3D CNN. The images only contained 2D

surface information of the part, and the internal information of the part was not applied; thus, the 2D CNN was unable to fully understand all feature information of the part. The 3D CNN performed well for the part price estimation regression problem. The voxel data contained all geometric features of the part, and a higher voxel resolution corresponded to a larger amount of part information, which was conducive to 3D CNN learning.

5. Conclusions

The problem of achieving a part processing evaluation was studied using a deep-learning approach. A CNN learning method applying part data for an estimation of the manufacturing cost was proposed. First, by generating a 2D image of the part as the input of the 2D CNN, a fine-tuned VGG16 model was used to study the effects of freezing the first 12, 18, and 24 layers of the network, as well as the performance of the original model for estimating the cost of the parts. When the first 18 layers of the freezing network were used, the accuracy was the highest, and the cost of the part manufacturing was estimated. Moreover, a method for training a 3D CNN with voxel data of the parts was presented. The voxel resolutions of the training data used in the 3D model were 64^3 , 128^3 , and 256^3 , respectively. After training, with the improvement in the voxel resolution, the MAPE of the cost estimation of the parts obtained using the model was small. When the number of training data reached 400,000, the accuracy of part estimation of the 3D CNN model was higher than when 72,594 training data were applied. This method achieves an accurate part manufacturing cost estimation. Finally, a comparison of the two schemes presented in this paper revealed that the accuracy of the 2D CNN learning of the part images used to estimate the manufacturing cost was significantly lower than that of the 3D CNN learning using voxel data of the part. Because the images only contain 2D surface information of the parts and a lack internal information, 2D CNN cannot learn all feature information of the parts. Voxel data contain all types of feature information of the parts; thus, the features can be easily learned using a 3D CNN. In this paper, the method for training the voxel data of a part to estimate the cost through a 3D CNN has potential application in the current

machining industry. This method can be used in a part quotation, which can greatly reduce the current quotation time and improve the quotation accuracy.

For some aspects of the parts that have no processing requirements (such as surface roughness and accuracy) but complex machining features, this method has high application for cost estimations. Future research will address how to add part processing requirements to voxel data to improve the accuracy of a cost estimation. The machining requirements of the parts are generally determined through 2D drawings, and future research will be conducted to extract the machining requirements of the parts and add them to the 3D model. During the voxelisation process, information on the machining requirements will be added. By training the voxel data containing the machining requirements of the parts, a highly precise cost estimation of any part can be realised.

References

- [1] Dewhurst P, Boothroyd G. Early cost estimating in product design. *Journal of Manufacturing systems*, 1988, 7(3): 183-191.
- [2] Harrell Jr F E. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, 2015.
- [3] Rickenbacher L, Spierings A, Wegener K. An integrated cost-model for selective laser melting (SLM). *Rapid Prototyping Journal*, 2013, 19(3): 208-214.
- [4] Mileham A R, Currie G C, Miles A W, et al. A parametric approach to cost estimating at the conceptual stage of design. *Journal of engineering design*, 1993, 4(2): 117-125.
- [5] Ganorkar A B, Lakhe R R, Agrawal K N. Cost estimation techniques in manufacturing industry: concept, evolution and prospects. *International Journal of Economics and Accounting*, 2017, 8(3-4): 303-336.
- [6] Folley M S, French M J, Widden M B. Function Cost of Pressure Vessels and Rolling Element Bearings. *Journal of Engineering Design*, 1992, 3(4): 291-305.
- [7] Tyagi S, Cai X, Yang K. Product life-cycle cost estimation: a focus on the multi-generation manufacturing-based product. *Research in Engineering Design*, 2015, 26(3): 277-288.
- [8] Azzouz A, Ennigrou M, Ben Said L. Scheduling problems under learning effects: classification and cartography. *International Journal of Production Research*, 2018, 56(4): 1642-1661.
- [9] Ji C Y, Hong T H, Hyun C T. CBR revision model for improving cost prediction accuracy in multifamily housing projects. *Journal of Management in Engineering*, 2010, 26(4): 229-236.
- [10] Rudolph J P, Emmelmann C. A cloud-based platform for automated order processing in additive Manufacturing. *Procedia CIRP*, 2017, 63: 412-417.

- [11] Takezawa K. Introduction to nonparametric regression. John Wiley & Sons, 2005.
- [12] Juszczak M. The challenges of nonparametric cost estimation of construction works with the use of artificial intelligence tools. *Procedia engineering*, 2017, 196: 415-422.
- [13] Von Beck U, Nowak J W. The merger of discrete event simulation with activity based costing for cost estimation in manufacturing environments. *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*. IEEE, 2000, 2: 2048-2054.
- [14] Chiarini A, Vagnoni E. World-class manufacturing by Fiat. Comparison with Toyota production system from a strategic management, management accounting, operations management and performance measurement dimension. *International Journal of Production Research*, 2015, 53(2): 590-606.
- [15] Wouters M, Stecher J. Development of real-time product cost measurement: A case study in a medium-sized manufacturing company. *International Journal of Production Economics*, 2017, 183: 235-244.
- [16] Rothe R, Timofte R, Van Gool L. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 2018, 126(2-4): 144-157.
- [17] Wei S E, Ramakrishna V, Kanade T, et al. Convolutional pose machines. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016: 4724-4732.
- [18] Psuj G. Multi-sensor data integration using deep learning for characterization of defects in steel elements. *Sensors*, 2018, 18(1): 292.
- [19] Litany O, Bronstein A, Bronstein M, et al. Deformable shape completion with graph convolutional autoencoders. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018: 1886-1895.
- [20] Yang Y, Feng C, Shen Y, et al. Foldingnet: Point cloud auto-encoder via deep grid deformation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018: 206-215.
- [21] Huang S H, Pan Y C. Ergonomic job rotation strategy based on an automated RGB-D anthropometric measuring system. *Journal of Manufacturing Systems*, 2014, 33(4): 699-710.
- [22] Wang, Charlie C. L., et al. Depth cameras based techniques and applications in design, manufacturing and services. *Journal of Manufacturing Systems* 33.4(2014):675-676.
- [23] Zhi S, Liu Y, Li X, et al. Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning. *Computers & Graphics*, 2018, 71: 199-207.
- [24] Liu Z, Tang H, Lin Y, et al. Point-Voxel CNN for Efficient 3D Deep Learning. *arXiv preprint arXiv:1907.03739*, 2019.
- [25] Shin D, Fowlkes C C, Hoiem D. Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018: 3061-3069.

- [26] Rossi A, Tessmann O. From Voxels to Parts: Hierarchical Discrete Modeling for Design and Assembly. International Conference on Geometry and Graphics. Springer, Cham, 2018: 1001-1012.
- [27] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.
- [28] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 922-928.
- [29] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press, 2016.
- [30] Wang J, Fu P, Gao R X. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. Journal of Manufacturing Systems, 2019, 51: 52-60.
- [31] Shea-Brown E, Josic K, de la Rocha J, et al. Universal properties of correlation transfer in integrate-and-fire neurons. arXiv preprint q-bio/0703037, 2007.
- [32] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012: 1097-1105.
- [33] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [34] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.