International Journal of Precision Engineering and Manufacturing
https://doi.org/10.1007/s12541-021-00586-y

Online ISSN 2005-4602
Print ISSN 2234-7593

REGULAR PAPER

Classification of blade's leading edge based on neural networks in adaptive machining of near-net-shaped blade

Zikai Yin[1] · Junxue Ren[1] · Yongshou Liang[1]

Received: 5 February 2021 / Revised: 24 August 2021 / Accepted: 5 September 2021

Abstract

The near-net-shaped blade is adopted in the aero engine as it's material-saving and efficient. However, the leading edge shape's curvature is sharply changed in its machining process and the deformation trend of each cross section has slight differences. Using the traditional machining method is exhausting and time-consuming. Furthermore, it brings more errors during the whole machining process. Therefore, adaptive machining is imported in the machining of the near-net-shaped blade and the leading edge is to be reconstructed during this process. Besides, it is necessary to know whether the reconstructed leading edge is qualified. To address these two issues, a novel approach is proposed to discriminate and classify leading edges. In this paper, we trained a style transform model of generative adversarial networks with theoretical leading edges and used its discriminator network to evaluate the similarity of reconstructed leading edges we had accomplished in our previous work to establish a standard for the qualified reconstructed leading edge. Then, as the curvature of the near-net-shaped blade changes sharply and has complex features, which requires high accuracy of classification, different DenseNet models were adopted to classify whether these reconstructed images are qualified. We experimented on our LDEG dataset and the highest accuracy on the test set was 88.7%. The experiment results demonstrated that the proposed method is effective in evaluating and classifying leading edges in the machining process.

Keywords Near-net-shape blade · Adaptive machining · Discrimination · Style transformation · DenseNet · Classification

1 Introduction

The modern commercial airplane has required better aero engine performance. Near-net-shaped blades are applied to the blades of the aero engine. The geometric parameters of the near-net-shape blade's suction/pressure surface have met the design requirement after being forged, which means that it needs no further machining, while the leading/trailing edge cannot be forged precisely due to the sharply changing curvature. Therefore, the redundant blank of this area needs to be removed in the machining process. Complex deformation, however, exists in each blank, such as changes in a twist, thickness, position, and curvature [1]. If the leading/trailing edge was machined according to pressure/suction surface, the uncertainty of the blank's geometric shape and spatial position will lead to that the leading/trailing edge cannot connect with the blade body smoothly. Besides, errors in the forming of suction/pressure surfaces are unavoidable. Using a theoretical leading/trailing edge to plan the machining tool path will result in flaws. In this case, it is essential to adjust the machining program according to different blades, which means the leading/trailing edge model adjusted to the forged blade needs to be reconstructed. An example of the spatial relationship of the blank, theoretical leading edge of a cross section is shown in Fig. 1. To solve the problem mentioned above, adaptive machining technology [2] is introduced. Adaptive machining adapts manufacturing data to suit changed conditions and it is suitable for the manufacturing of complex surfaces, which is the key component of advanced manufacturing.

✉ Junxue Ren
rjx1968@nwpu.edu.cn

Zikai Yin
yinzikai1997@mail.nwpu.edu.cn

Yongshou Liang
liangyongshou@nwpu.edu.cn

[1] Key Laboratory of High-Performance Manufacturing for Aero Engines, School of Mechanical Engineering, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi'an 710072, People's Republic of China

Published online: 24 September 2021

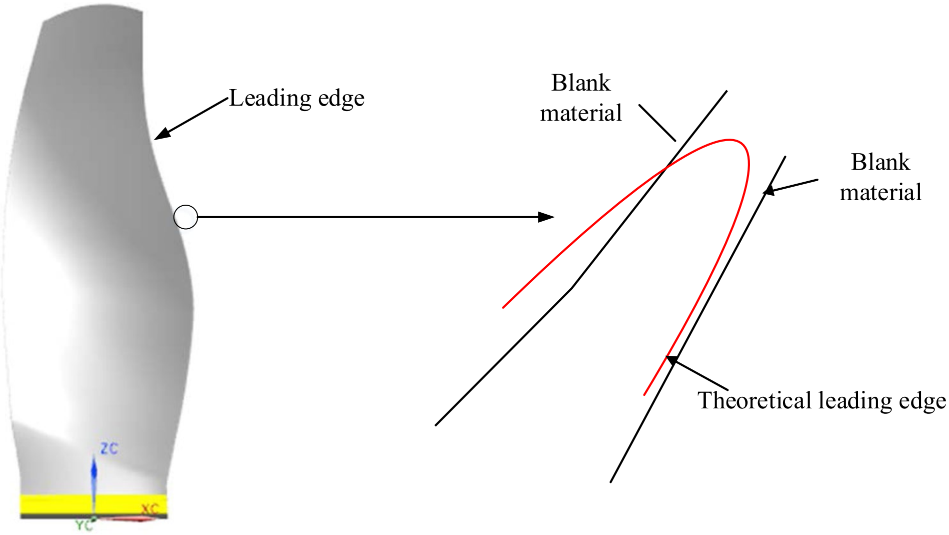


Fig. 1 Theoretical leading edge and blank

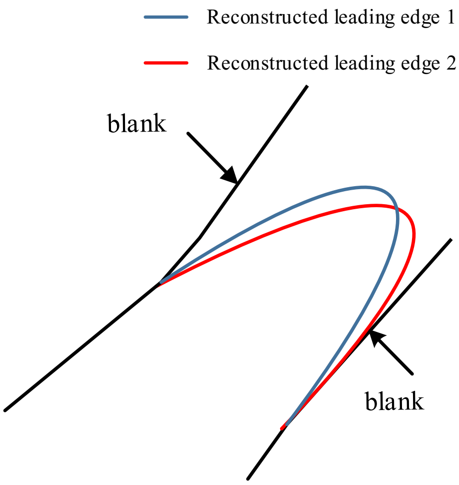


Fig. 2 Different shapes of reconstructed leading edges

Adaptive machining includes digital measurement, data matching, and model reconstruction. Firstly, digital data of blades is obtained by measurement, which is followed by data registration (matching). After that, model reconstruction technology is applied to each blade's specific cross section, by which, deformation trend information and design intend are taken into consideration to reconstruct its model.

However, the existed model reconstruction theories mainly focus on the smooth connection between model and blank material, instead of the similarity of theoretical shape. Only when the above requirements are taken into account in the process of model reconstruction and a model reconstruction theory with multiple constraints is established, can the design intention of high-performance free-form surface be complied with to the maximum extent and the accurate construction of machining process geometric model be carried out. Figure 2 gives two different reconstructed leading edges. These two reconstructed leading edges all satisfy the requirement of smooth connection, while the shapes vary. As design intent is to be retained to the maximum extent, which means that reconstructed leading edges are supposed to be similar to theoretical leading edges, we selected the similarity between reconstructed and theoretical leading edges as our principle. Unfortunately, the similarity between reconstructed and theoretical leading edges is hard to illustrate and quantified. In this paper, we tried to solve this problem by applying generative adversarial networks (GAN) as it defines the similarity between two different images according to their feature maps extracted by convolutional neural networks. In this method, details of reconstructed leading edges are reserved and the whole evaluation is more precise than the traditional method as we retained the original design intent as possible. Additionally, our data contains less semantic information, which results in the phenomenon that the key information could be lost after multiple layers of convolution. Hence, a suitable classification algorithm is desired for our forthcoming research. Our main works are demonstrated as follows:

(1) We proposed a novel method used to evaluate the similarity between a theoretical leading edge and a reconstructed leading edge based on generative adversarial networks.
(2) We trained a style transform model and took its discriminator to evaluate the similarity value above mentioned based on previous manual works, by which we built a standard for qualified reconstructed leading edges.
(3) A classification approach using DenseNet neural network was applied to automatically classify whether the reconstructed leading edge is qualified.

2 Related Works

Our works are based on the following three domains: adaptive machining, generative adversarial networks, and classification networks.

2.1 Adaptive Machining

Two key steps are included in the adaptive machining process: data registration and model reconstruction. As for data registration, Khameneifar et al. [3] proposed a method to evaluate the tolerance of blade position and torsion by determining the position of the overlapping points in the section line. Huang et al. [4] adopted the optimal fitting algorithm to complete the global accurate registration. Shi et al. [5] used the combined registration algorithm of normal distribution transformation and nearest point iteration (NDT-ICP) to carry out point cloud registration. Yu et al. [6] applied the deviation distribution on evaluating the point-to-point deviation to distinguish manufacturing errors from measurement errors. Chen et al. [7] proposed a feature-based registration method mainly focused on 3D eddy current pulsed thermography. In this paper, our main focus is the reconstruction of models, and data registration won't be further discussed.

Based on the different measurement methods, model reconstruction approaches can be divided into two types [8]: the model reconstruction based on scattered point cloud data and the model reconstruction based on regular point sets. Brujic et al. [9] proposed a modification method of the NURBS surface model based on scattered point clouds in three-dimensional space. This method overcomes the basic problem of singular or ill-conditioned matrices caused by incomplete data sets. However, the blade design process is based on a complex three-dimensional flow field, which relies heavily on multi-disciplinary design optimization. Because it takes a long time and is inefficient to calculate the three-dimensional flow field, this process is usually simplified to the superposition of two-dimensional cross sections. In terms of blade's reconstruction, considering the parameterized design process of the blade, the reconstruction method based on regular point sets is more used. Generally speaking, there are two types of this method. The first one is to fit or interpolate the measured data to construct a new curve or surface, which requires more data to make the new curves or surfaces match the measured points. Ke et al. [10, 11] proposed a constrained fitting method, which improved numerical stability. Khameneifar et al. [12] proposed an airfoil generation method according to measurement uncertainty value. The fitting or interpolating process requires a large number of data to make the constructed curves/surfaces match the measured data. The second method is to modify curves or surface parameters, by which object properties with fewer measured data are maintained. Piegl et al. [13] modified the shape of a non-uniform rational basis spline curve/surface, which was based on control-point and weight. Hu et al. extended Piegl's method to a constrained optimization problem, where the surface energy was optimized to a minimum under geometric constraints. These two methods adopted a common algorithm to modify the curve/surface mathematically. In terms of blade's machining, industrial requirement and conditions, geometric constraints, and data missing are supposed to be included, which means the blade's reconstruction algorithm should be specific. To bring design intent to reconstruction, Mohaghegh et al. [14, 15] used multi-segment arcs instead of spline curve fitting to realize the model reconstruction of blades. After that, they searched for some design parameters and introduced external geometric features to the reconstruction of the actual profile. Dong et al. [16] calculated circles whose center was located on the axial curve to reconstruct the profile curves and improved the continuity of curves. Yun et al. [17] took geometric constraint into consideration and import upper and lower tolerance to reconstruction. Li et al. [18] improved smoothness by minimizing strain energy. These methods were used to reconstruct the curves with complete data. Nevertheless, a part of the data of the curve might be inaccurate or missing. To solve this problem, Yilmaz et al. [19] interpolated the absence of blade by adjacent measured data. Zhao et al. [20] used an iterable algorithm to improve the accuracy of the reconstructed absent part by inserting knots. The abovementioned methods are not suitable for parts with sharply-changed curvature such as precision-forged blade and near-net-shaped blade's leading trailing edge anymore. Besides, the forging process itself brings additional deformation to the blade. To address this issue, Feng et al. [21] proposed a method based on deformation to predict the profile of blades. Wu et al. [22] used distance relationship, angle relationship, and radius relationship to cull measuring bad point. These previous works did not fully consider the design intention of the blade and the similarity between the actual blade surface and the theoretical surface and the similarity discrimination principle are to be further and deeply researched.

2.2 Generative Adversarial Networks

In the field of generative adversarial networks, Goodfellow et al. [23] proposed the generative adversarial networks for the first time. After that, Mehdi Mirza et al. [24] proposed a conditional generative adversarial model to lead generative adversarial networks to generate needed images. Phillip Isola et al. [25] proposed that an image generation task is to perform a pixel-to-pixel transformation. Jun-Yan Zhu et al. [26] applied cycle-consistent loss on generative adversarial networks to preserve the content of input images. These models have difficulty generating images in multi-domains as models are to be built separately for every pair of domains. Therefore, Yunjey Choi et al. [27] proposed starGAN model, for realizing the one-to-multi transformation task.

2.3 Classification Networks

In terms of images classification, neural network [28] has become a popular field in recent years, resulting in applications related to object detection and classification [29]. In recent years, Simonyan et al. [30] found that increasing the number of layers of convolutional neural networks plays an important role in improving the performance of those networks. Therefore, they proposed a VGGNet with nineteen layers. After that, more and more deep convolutional network models have been used. As the depth of the convolutional neural network increases, the image information will gradually disappear as the convolutional layer deepens. To solve this problem, Kaiming He et al. [31] proposed ResNet, using skip connection to train a model with high performance. Gao H et al. [32] proposed the DenseNet and obtained better results than ResNet.

This paper is structured as follows: Sect. 3 presents the proposed method including discrimination and classification. Section 4 gives the training detail. Section 5 reports the experimental result and discussion. Finial conclusion is depicted in Sect. 6.

3 Proposed Method

3.1 Revisiting GAN

A generative adversarial network (GAN) consists of two independent neural networks: a discriminator and a generator. The main job of the generator is to generate images while the main job of the discriminator is to classify whether these fake images are real. The goal of a generative adversarial model can be written as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_{z}(z)}\left[\log\left(1 - D(G(z))\right)\right] \tag{1}$$

where x is sampled from the training dataset and z represents a group of random noises. D and G denote discriminator and generator respectively.

The generator's loss function can be written as:

$$L_G = E_{z\sim p_z(z)}\left[\log(1 - D(G(z))\right] \tag{2}$$

And the loss function of the discriminator is:

$$L_D = E_{x\sim p_{data}(x)}\left[\log D(x)\right] \tag{3}$$

For a single input image x_i, the loss function of the discriminator can be written as:

$$L_D^i = E_{x_i\sim p_{data}(x)}\left[\log D\left(x_i\right)\right] \tag{4}$$

L_D evaluates the probability of that the input image x belongs to the target image's domain. In our task, the value of L_D represents the probability of assigning the label 'theoretical leading edge' to the reconstructed leading edge. In other words, the higher the probability is, the more similar the theoretical leading edge the reconstructed leading edge is.

3.2 Similarity Discrimination

In this paper, we annotated reconstructed leading edge images with two domains: thin and thick according to blank material's thickness information. Unfortunately, previous generative adversarial network models focus on a specific domain and are unable to generate images in multi-domains. Therefore, we used a style transform model of generative adversarial network to evaluate its similarity.

Considering the instability of the generative adversarial model itself and the difference of the input dataset, we implemented normalization to the loss function of the discriminator, denoted as L. The final similarity value is as follows:

$$s_i = \frac{L_i}{\sum_j^N L_j} \tag{5}$$

where N is the number of images.

If we use S_t to represent the standard similarity value, S_t can be calculated by:

$$s_t = \frac{\left(\max\limits_{i=1,2\ldots,N}\left(s_i\right) - \min\limits_{i=1,2\ldots,N}\left(s_i\right)\right)}{2} \tag{6}$$

After the process mentioned above is finished, the reconstructed leading edge images were divided into two groups (qualified and unqualified) based on the following rules:

$$s_i = \begin{cases} > s_t\ qualified \\ < s_t\ unqualified \end{cases} \tag{7}$$

By importing similarity value S_i, we transfer the differences of reconstructed leading edge and theoretical leading edge into the distance between their images in the spatial domain, in which S_i can be interpreted as:

$$s_i = \left\| u_{rec} - u_{theo} \right\| \tag{8}$$

where u_{rec} and u_{theo} denote the feature maps of reconstructed leading edge and theoretical leading edge after convolutions. This distance represents the reconstructed leading edge's deviation from its corresponding theoretical leading edge.

In this work, we use theoretical leading edge images to train the discriminator of GAN and applied it to the evaluation of the similarity value of reconstructed leading edge images. The overview of the method is shown in Fig. 3.

We adopted and modified the architecture of StarGAN [27], the generator has a convolutional layer with a kernel size of seven and a stride size of one for down-sampling and two convolutional layers with kernel sizes of four and stride size of two for up-sampling. For the discriminator network, we used two convolutional layers with different kernel sizes for computing adversarial loss and domain classification loss. The detail of our model is shown in Fig. 4.

3.3 Classification

Unlike the general dataset in the complex scenario such as COCO, our dataset is very simple and contains less semantic information. The leading edge's information could be lost after multiple convolutional operations. In this case, traditional convolutional neural network models are not suitable anymore. Aiming to address this kind of question, a typical solution is to import residual connection and the model is named ResNet.

Gao H et al. proposed the DenseNet in 2017, which had an excellent performance on CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets than ResNet. DenseNet has some similarities with ResNet. One big difference between the two is that DenseNet builds fully connected layers, that is, the output of every layer is connected with the later layers. If we use $X0, X1, X2, ..., Xl-1$ to represent the output of previous $(l-1)$ layers, the output of l^{th} can be expressed as:

$$Xl = Hl([X0, X1, X2, ..., Xl-1]) \tag{9}$$

$Hl(.)$ is defined as three continuous operations: batch normalization [33], ReLU function, and convolution with a 3×3 kernel.

DenseNet consists of four DenseBlocks and three Transition layers. DenseBlock mainly focuses on feature extraction and Transition compresses the dimension of the feature map to reduce the network's parameter. Figure 5 illustrates its architecture of connection.

What DenseBlock uses is the dense connection, which is expressed as $Xl = Hl([X0, X1, X2, ..., Xl-1])$. Its inner connection is illustrated as Fig. 6.

where different colors represent feature maps. Transition layers aim to connect two adjacent layers and reduce the size of the DenseBlock output image. It includes a convolution operation with a convolution kernel size size of 1×1 and a 2×2 average pooling. DenseNet can be divided into different models due to their different numbers of respective layers. According to [32], DenseNet121, DenseNet161, DenseNet169, and DenseNet201 got

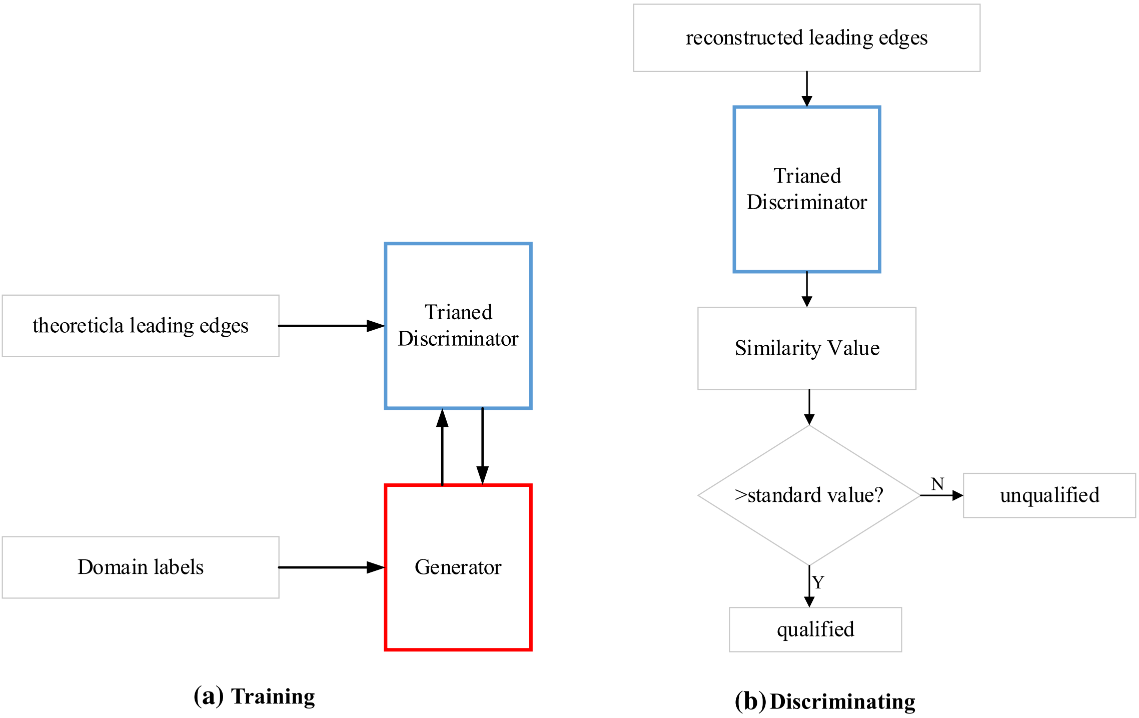
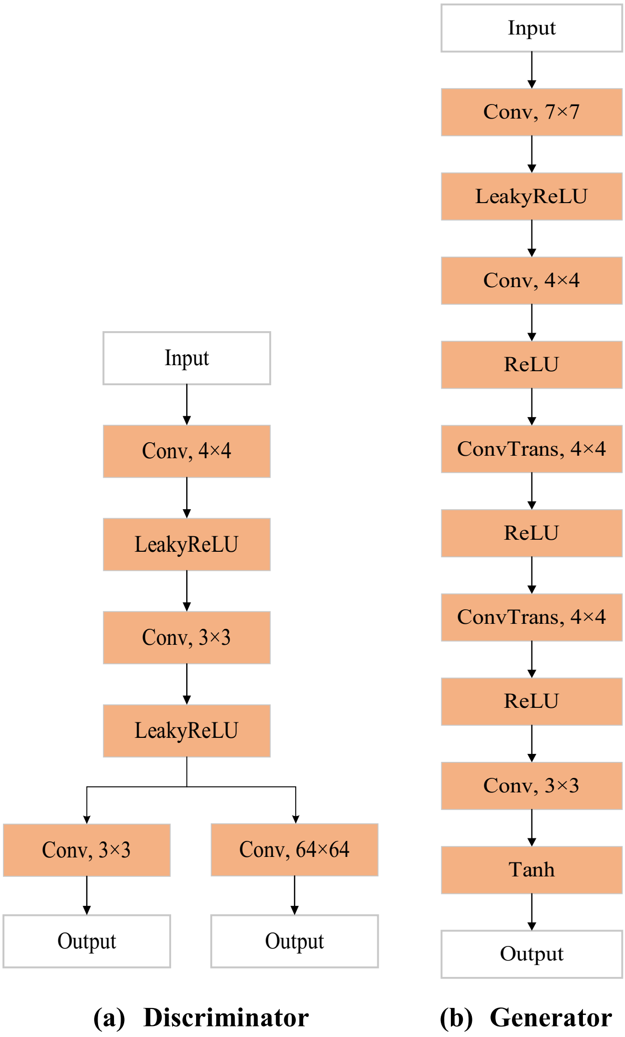


Fig. 3 Overview of proposed discrimination method



(a) Discriminator **(b) Generator**

Fig. 4 Structures of discriminator and generator

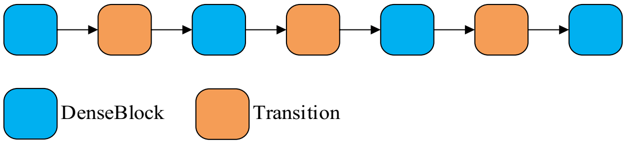


Fig. 5 DenseNet's connection

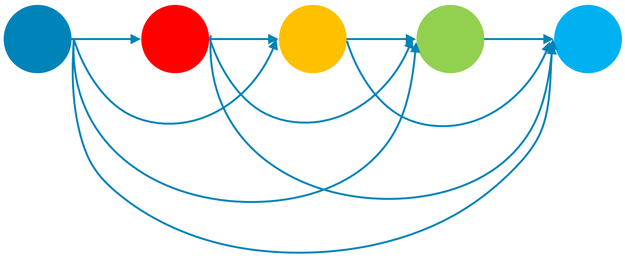


Fig. 6 The inner connection of a DenseBlock

Table. 1 Statistic of LDEG dataset

Type	Number	Domain	Label
Theoretical leading edge	44	Theoretical	None
Reconstructed leading edge	310	Thin, thick, theoretical	Qualified, unqualified

extraordinary performance in ImageNet competition. Hence, we selected these four models as our classifiers.

DenseNet directly connects different layers and makes use of uniform-size feature maps and corresponding feature channels for feature reuse, which can effectively reduce the number of parameters, so that the gradient value will not be too large or too small in the training process, and the gradient explosion can be prevented. What's more, it allows us to train the network at a deeper level without compromising the effectiveness of the training because of the deeper network layers.

4 Experiments

4.1 Dataset

The dataset we applied was created by ourselves and was named by LDEG. LDEG dataset contains 44 theoretical leading edge images of different cross sections and 143 reconstructed leading edge images which are annotated with two binary attributes. As theoretical leading edge images are only to train a discriminator for similarity discrimination, the annotation is set to be all theoretical. The detail of LDEG dataset is shown in Table. 1.

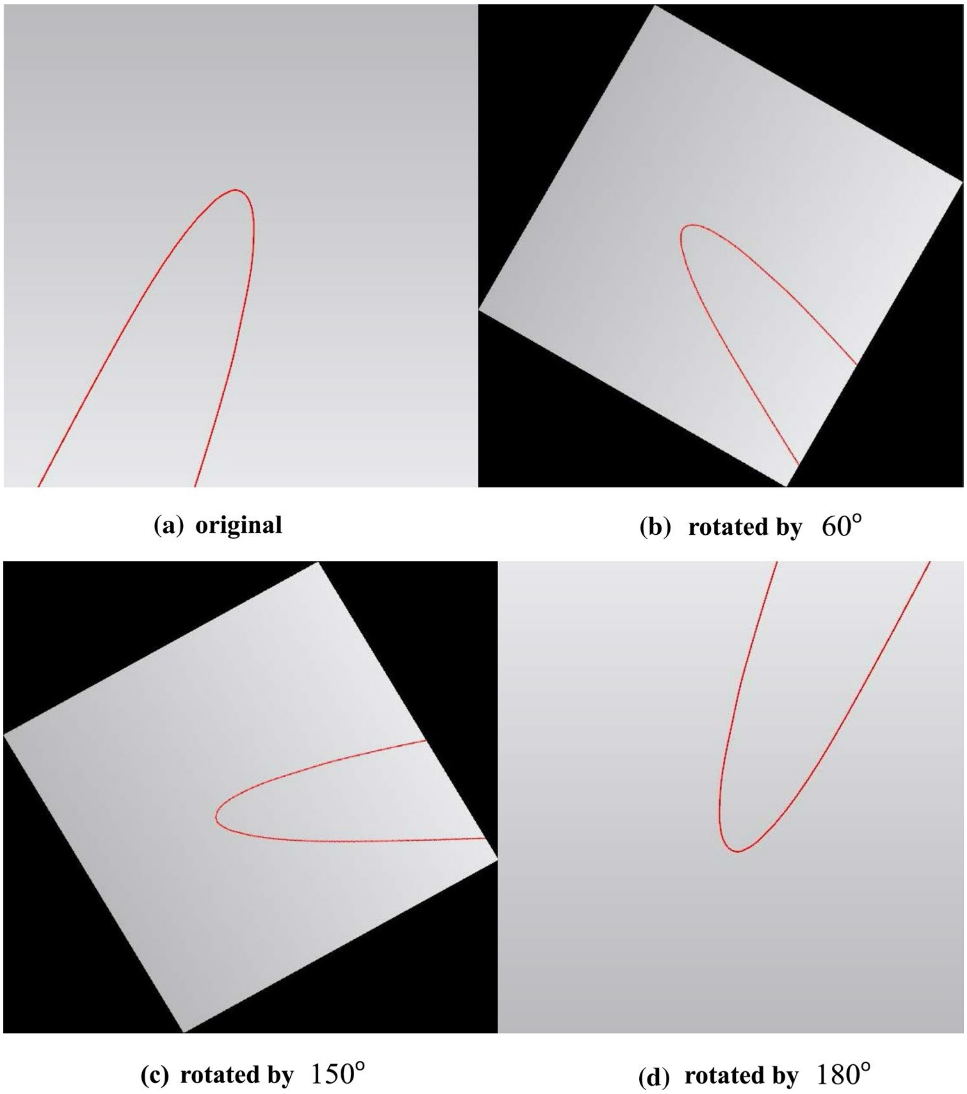
Generally speaking, neural network obtains better performance based on more data. However, theoretical leading edge data in LDEG dataset are limited. A solution for this issue is to use data augmentation to enlarge the dataset. We performed appropriate flips horizontally for the training data to expand the dataset. Figure 7 is the images before and after the geometric flip of the same training set picture.

Assuming that (i,j) is the coordinate of the original image, the new image's coordinate can be computed as:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \tag{10}$$

where $\left(i', j'\right)$ represents the coordinates of the new image and θ represents the angle of rotation. After the flipping operation, we can expand the number of inputs fourfold.

The training set contains 80 qualified images and 110 unqualified images. The test set contains 60 images of qualified



(a) original

(b) rotated by 60°

(c) rotated by 150°

(d) rotated by 180°

Fig. 7 Results of rotating an image by different angle

leading edges and 60 images of unqualified leading edges. It should be noticed that all training data of DenseNet had been classified through the similarity principle above mentioned. To make the model more stable, we standardized the training data. Assuming x is the input picture, the standardization method is written as follows:

$$z = (x/255 - 0.5)/0.5 \tag{11}$$

4.2 Training Process

4.2.1 Training of GAN

In this paper, the full loss function is divided into three parts: traditional adversarial loss, domain classification loss, and reconstruction loss. Traditional adversarial loss is written as:

$$L_{adv} = E_x\left[\log D_{ori}(x)\right] + E_{x,c}\left[\log\left(1 - D_{ori}(G(x,c))\right)\right] \tag{12}$$

where x is the input real image and c is the target domain label. Equation (4) was replaced with Wasserstein distance [34, 35] to secure the diversity of generated images. So, the adversarial loss is rewritten as:

$$L_{adv} = E_x[D_{ori}(x)] - E_{x,c}[D_{ori}(G(x,c))] - \lambda_{gp}E_{x'}[(||\nabla D_{ori}(x^{'})||_2 - 1)^2] \tag{13}$$

where x' is sampled uniformly along a straight line between a pair of real and generated images and λ_{gp} is the weight of Wasserstein distance.

In addition, a domain classification loss is defined as an output image is supposed to be classified to its corresponding domain. Domain classification loss has 2 parts and is written as follows:

$$L^{r}_{cls} = E_{x,c'}[-\log D_{cls}(c'|x)] \tag{14}$$

$$L^{f}_{cls} = E_{x,c}[-\log D_{cls}(c|G(x,c))] \tag{15}$$

where c' is the original domain of input real image. L^{r}_{cls} is used to optimize the discriminator and L^{f}_{cls} is used to optimize the generator. By minimizing loss functions, the discriminator learns to classify input leading edge image to original domain and the generator learns to generate reconstructed leading edge images that the discriminator can classify correctly.

We imported cycle consistency loss [25] as the reconstruction loss in this paper, which was defined as follows to make sure that input images would only change their domain-related part:

$$L_{rec} = E_{x,c,c'}[||x - G(G(x,c),c^{'})||_1] \tag{16}$$

Overall, the full loss function can be written as:

$$L_D = -L_{adv} + \lambda_{cls} L^{r}_{cls} \tag{17}$$

$$L_G = -L_{adv} + \lambda_{cls} L^{r}_{cls} + \lambda_{rec} L_{rerc} \tag{18}$$

where λ_{cls} and λ_{rec} determine the domain classification and reconstruction losses' relative importance and for the whole training process.

We trained the GAN model with a batch size of sixteen and used Adam [36] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. λ_{gp} was set to ten and finally we set $\lambda_{cls} = 1$ and $\lambda_{rec} = 10$. The whole model was trained for 2000 iterations.

4.2.2 Training of DenseNet

We define the loss function in the training process as cross entropy, and its expression is as follows:

$$L = -\sum_{i=1}^{N} y^{(i)} \log(\hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right) \tag{19}$$

where $y^{(i)}$ is the predicted value of the i^{th} sample, and $\hat{y}^{(i)}$ is the label value of the i^{th} sample.

All the networks are trained using stochastic gradient descent. In training, we use eight neural network models: AlexNet[37], ResNet18, DenseNet121, and Dense161, DenseNet169, and DenseNet201. To avoid over-fitting, we set the learning rate to 0.001 and the number of training epochs to 30. All training procedures were carried out on a single GPU of Nvidia GTX 1080Ti.

5 Results and Discussions

5.1 Results of the Discrimination

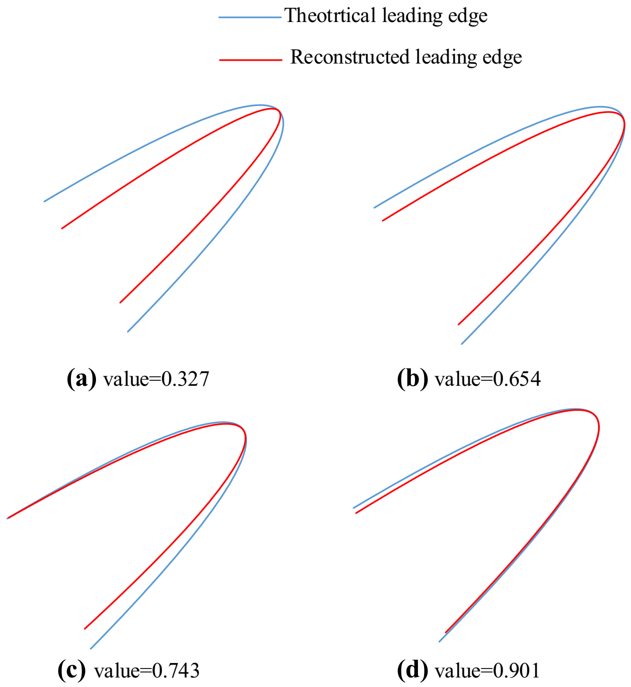
Figure 8 demonstrates evaluation results of four different reconstructions of the same leading edge given by the discriminator, in which similarity values vary from 0 to 1.

Our discussion of the abovementioned results is divided into two parts: the remained area of the theoretical leading edge and the reconstructed curve of the reconstructed leading edge.

By comparing remained areas of four leading edges, (d) remains the most part of the theoretical leading edge while (a) only remains a little part of the theoretical leading edge. As the remained part gets larger, the similarity value rises, which conforms to our previous manual standard for discrimination.

Like our discussion about remained area's effect on similarity discrimination, the same conclusion can be inferred by comparing reconstructed curves. The reconstructed curves of (d) fit more with the theoretical leading edge. On the contrary, the reconstructed curves of (a) differ greatly from their corresponding theoretical leading edge. Hence, the reconstructed curves' influence on similarity identifies with our manual standard.

The result above illustrates that our discriminator is capable of evaluating the similarity of reconstructed leading edges. It should be mentioned that the higher value a reconstructed leading edge gets if it is more similar to theoretical leading edges. Accordingly, the better aerodynamic performance it has. Through this approach, we can obtain these qualified and unqualified reconstructed leading edges for the training of the next classification.



(a) value=0.327

(b) value=0.654

(c) value=0.743

(d) value=0.901

Fig. 8 Evaluation results of the discriminator

5.2 Result of Classification

5.2.1 Evaluation Metrics

True positive (TP), false positive (FP), false negative (FN), True negative (TN) are usually needed in the evaluation of a model's performance. We imported a confusion matrix for a binary classification task to define them. See Table. 2

In this paper, we utilized accuracy(Acc), precision, and recall to evaluate the robustness of models on test sets. They are formulated as follows:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{20}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{21}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{22}$$

Acc evaluates the performance of classification. Generally speaking, the higher acc a model gets, the better its classification capability is. Precision represents the ratio that correctly predicted samples account for all predicted samples and indicates the model's ability to predict the target class. On the other side, recall is the percentage of the targets correctly predicted and supposed to be predicted. It illustrates the ability that the model searches for the correct targets. In the leading edge classification task, we mainly focus on precision as our goal is to classify the qualified leading edges as possible.

Besides those abovementioned metrics, we introduced floating point operations (FLOPs), inference time, and the number of parameters (denoted as #Params), which measures the computing efficiency and complexity of a neural network model.

Table 2 Confusion matrix

Ground-truth	Prediction	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

6 Results

Table. 3 shows the recognition accuracy, precision, and recall of the models of AlexNet, ResNet18, DenseNet121, DenseNet161, DenseNet169, and DenseNet201:

DenseNet169 had the highest accuracy and precision of classification. For the specific problem of the leading edge's reconstruction, as the data set itself contains relatively simple features and there is no interference from other noises, the recognition accuracy of the DenseNet network could reach 88.7%. In the leading edge classification task, we mainly focus on the Acc and Precision. Based on the experiment results above, we inferred that DenseNet169 is the most suitable for classifying leading edges. The classification error is acceptable during the machining process. However, DenseNet169 and DenseNet201 did not perform well on the test set as we expected. Some factors contribute to this result. The first reason is that DenseNet169 and DenseNet201 contain more parameters, which means that more time is needed for training. Besides, the number of images of the LDEG dataset limits the final performance of DenseNet169 and DenseNet201. In future work, the set of training parameters of the DenseNet model and the augmentation method of the LDEG dataset can be further investigated.

The above results indicate that the DenseNet network can effectively identify the bridge of the leading edge. For the specific problem of the leading edge's bridging, because the training set itself contains relatively simple features and there is no interference from other noises, the recognition accuracy of the DenseNet network could reach 88.7% and the classification error is acceptable during the machining process. Compared with DenseNet, AlexNet has simpler layers and architecture, so it rates lower than DenseNet in terms of recognition accuracy.

Figure 9 gives an example of the final classification result on the same cross section. Although these two reconstructed leading edges meet the machining requirement, that is, having a smooth connection with blank material, the unqualified

Table 3 Models' performance on the test set

Model	FLOPs	#Params	Acc	Precision	Recall	Inference time
AlexNet	1.44G	61.10 M	80.0%	79.0%	81.6%	5 s
ResNet18	1.82G	11.69 M	81.3%	82.3%	85.0%	19 s
DenseNet121	5.76G	7.90 M	86.3%	83.1%	90.0%	18 s
DenseNet161	15.64G	28.69 M	81.0%	75.3%	91.7%	37 s
DenseNet169	6.84G	14.15 M	88.7%	85.1%	95.0%	20 s
DenseNet201	8.74G	20.01 M	83.3%	83.3%	83.3%	24 s

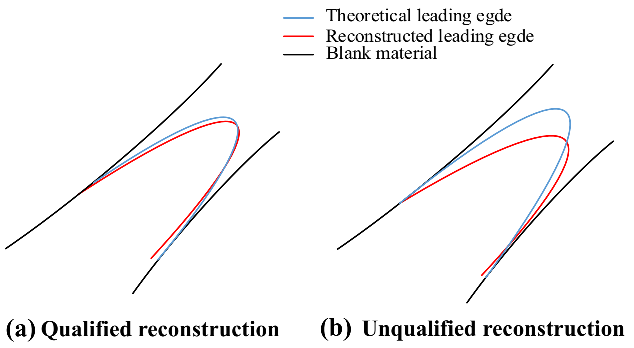

Theoretical leading egde
Reconstructed leading egde
Blank material
(a) Qualified reconstruction
(b) Unqualified reconstruction

Fig. 9 Qualified and unqualified reconstruction

reconstructed leading edge's shape differs greatly from that of the theoretical part. On the contrary, a qualified reconstructed leading edge fits more closely with the theoretical part.

7 Conclusion

Aiming at the similarity evaluation and classification problems of the reconstruction of leading edges in the adaptive processing of fan blades, we firstly trained a GAN model and took its discriminator to discriminate the similarity of reconstructed leading edges, by which, we built a standard of qualified reconstructed leading edges. After that, four DenseNet networks were built to automatically identify the leading edge after reconstructing and determine whether it meets the processing requirements. The testing result showed that the highest accuracy on the test set reached 88.7%. The result proves that the proposed method based on neural networks is applied to the classification of the leading edge in adaptive machining processing successfully. In future work, the following issues can be further investigated:

1. The images of the LDEG dataset contain fewer semantic information, which means that their spatial information could be lost after multiple convolutional operations. Considering that our next work is to recognize the reconstructed area as an object detection task, how to design the architecture and loss function should be deeply researched.
2. All images in this work are of their original sizes and we did not invest the result after resizing them. The effect on our model's robustness after image resizing is supposed to be further studied.
3. In this paper, we did not take the influence of errors during measuring and data registration into consideration such as viewpoint. The development of the error compensation algorithm is another focus.

Acknowledgements This work was supported by the National Science and Technology Major Project (Grant number: J2019-VII-0001-0141). We are also grateful to Dr. Hailiang Jin and Yapeng Duan for their technical support.

Author's Contribution Zikai Yin proposed the concept of this paper and carried out the experiment. Then he finished the draft for the manuscript. Yongshou Liang and Junxue Ren reviewed the draft of this paper and gave some suggestions. All the authors read and approved the final manuscript.

Funding This work was supported by the National Science and Technology Major Project, China (Grant No.: J2019-VII-0001–0141).

Data Availability The data of this work is included in the article.

Declarations

Conflicts of interest The authors declare that they have no conflict of interests.

References

1. Makem, J. E., Ou, H., & Armstrong, C. G. (2012). A virtual inspection framework for precision manufacturing of aerofoil components. *Computer-Aided Design, 44*(9), 858–874. https://doi.org/10.1016/j.cad.2012.04.002
2. Bremer, C. (2000). Adaptive Strategies for Manufacturing and Repair of Blades and Blisks. *Proceedings of the ASME Turbo Expo 2000: Power for Land, Sea, and Air*, V1T-V4T. https://doi.org/10.1115/2000-GT-0340
3. Khameneifar, F., & Feng, H. (2016). A new methodology for evaluating position and orientation errors of airfoil sections. *The International Journal of Advanced Manufacturing Technology, 83*(5–8), 1013–1023. https://doi.org/10.1007/s00170-015-7641-x
4. Huang, J., Yuan, Y., Wang, Z., Qi, Z., Xing, C., & Gao, J. (2018). A global-to-local registration and error evaluation method of blade profile lines based on parameter priority. *The International Journal of Advanced Manufacturing Technology, 94*(9–12), 3829–3839. https://doi.org/10.1007/s00170-017-1125-0
5. Shi, X., Peng, J., Li, J., Yan, P., & Gong, H. (2019). The iterative closest point registration algorithm based on the normal distribution transformation. *Procedia Computer Science, 147*, 181–190. https://doi.org/10.1016/j.procs.2019.01.219
6. Jin, Y., Liao, H., & Pierson, H. (2019). Automatic feature-based point cloud alignment and inspection. *Procedia Manufacturing, 39*, 484–492. https://doi.org/10.1016/j.promfg.2020.01.405
7. Chen, X., Tian, G., Wu, J., Tang, C., & Li, K. (2019). Feature-based registration for 3D eddy current pulsed thermography. *IEEE Sensors Journal, 19*(16), 6998–7004. https://doi.org/10.1109/JSEN.2019.2911699
8. Várady, T., Martin, R. R., & Cox, J. (1997). Reverse engineering of geometric models—an introduction. *Computer-Aided Design, 29*(4), 255–268. https://doi.org/10.1016/S0010-4485(96)00054-1
9. Brujic, D., Ristic, M., & Ainsworth, I. (2002). Measurement-based modification of NURBS surfaces. *Computer-Aided Design*. https://doi.org/10.1016/S0010-4485(01)00060-4
10. Ke, Y., Zhu, W., Liu, F., & Shi, X. (2006). Constrained fitting for 2D profile-based reverse modeling. *Computer-Aided Design, 38*(2), 101–114. https://doi.org/10.1016/j.cad.2005.07.004

11. Ke, Y., Fan, S., Zhu, W., Li, A., Liu, F., & Shi, X. (2006). Feature-based reverse modeling strategies. *Computer-Aided Design, 38*(5), 485–506. https://doi.org/10.1016/j.cad.2005.12.002
12. Khameneifar, F., & Feng, H. (2014). Airfoil profile reconstruction under the uncertainty of inspection data points. *The International Journal of Advanced Manufacturing Technology, 71*(1–4), 675–683. https://doi.org/10.1007/s00170-013-5527-3
13. Piegl, L., & Tiller, W. (1997). *The Nurbs Book*. Springer.
14. Mohaghegh, K., Sadeghi, M. H., & Abdullah, A. (2007). Reverse engineering of turbine blades based on design intent. *The International Journal of Advanced Manufacturing Technology, 32*(9–10), 1009–1020. https://doi.org/10.1007/s00170-006-0406-9
15. Mohaghegh, K., Sadeghi, M. H., Abdullah, A., & Boutorabi, R. (2010). Improvement of reverse-engineered turbine blades using construction geometry. *The International Journal of Advanced Manufacturing Technology, 49*(5–8), 675–687. https://doi.org/10.1007/s00170-009-2409-9
16. Dong, Y., Zhang, D., Bu, K., Dou, Y., & Wang, W. (2011). Geometric parameter-based optimization of the die profile for the investment casting of aerofoil-shaped turbine blades. *The International Journal of Advanced Manufacturing Technology, 57*(9–12), 1245–1258. https://doi.org/10.1007/s00170-011-3681-z
17. Yun, Z., Zhi-Tong, C., & Tao, N. (2015). Reverse modeling strategy of aero-engine blade based on design intent. *International journal of advanced manufacturing technology, 81*(9–12), 1781–1796. https://doi.org/10.1007/s00170-015-7232-x
18. Li, Y., & Ni, J. (2009). Constraints based nonrigid registration for 2d blade profile reconstruction in reverse engineering. *Journal of Computing and Information Science in Engineering, 9*(3), 31005.
19. Yilmaz, O., Gindy, N., & Gao, J. (2010). A repair and overhaul methodology for aeroengine components. *Robotics and Computer-Integrated Manufacturing, 26*(2), 190–201. https://doi.org/10.1016/j.rcim.2009.07.001
20. Zhao, Z., Fu, Y., Liu, X., Xu, J., Wang, J., & Mao, S. (2017). Measurement-based geometric reconstruction for milling turbine blade using free-form deformation. *Measurement, 101*, 19–27. https://doi.org/10.1016/j.measurement.2017.01.009
21. Yazhou, F., Junxue, R., & Yongshou, L. (2018). Prediction and reconstruction of edge shape in adaptive machining of precision forged blade. *The International Journal of Advanced Manufacturing Technology, 96*(5–8), 2355–2366. https://doi.org/10.1007/s00170-018-1771-x
22. Wu, D., Wang, H., Zhang, K., Zhao, B., & Lin, X. (2020). Research on adaptive CNC machining arithmetic and process for near-net-shaped jet engine blade. *Journal of Intelligent Manufacturing, 31*(3), 717–744. https://doi.org/10.1007/s10845-019-01474-z
23. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., et al (2014). Generative Adversarial Nets. *Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS)*, 2672–2682. https://doi.org/10.5555/2969033.2969125
24. Mirza, M., Osindero, S. (2014, November 6). *Conditional Generative Adversarial Nets*. arXiv:1411.1784. Retrieved August 13, 2021
25. Isola, P., Zhu, J., Zhou, T., Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632
26. Zhu, J. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV)*, 2242–2251. https://doi.org/10.1109/ICCV.2017.244
27. Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., Choo, J. (2018). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. *Proceedings of the 31th IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 8789–8797. https://doi.org/10.1109/CVPR.2018.00916
28. Hinton, G. H. G. E., & Salakhutdinov, R. S. R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.
29. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Jie, C., Liu, X., et al. (2020). Deep learning for generic object detection: a survey international. *Journal of Computer Vision*. https://doi.org/10.1007/s11263-019-01247-4
30. Simonyan, K., Zisserman, A. (2014, September 4). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556v6. Retrieved August 13, 2021, from https://arxiv.org/abs/1409.1556v6
31. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90
32. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243
33. Ioffe, S., Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning(ICML)*, 448–456.
34. Arjovsky, M., Chintala, S., Bottou, L. (2017, December 6). *Wasserstein GAN*. arXiv: 1701.07875. Retrieved August 13, 2021, from https://arxiv.org/abs/1701.07875
35. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. (2017). Improved Training of Wasserstein GANs. *Proceedings of the 31st International Conference on Neural Information Processing Systems(NIPS)*, 5769–5779.
36. Kingma, D., Ba, J. (2017, January 30). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980v8. Retrieved August 13, 2021, from https://arxiv.org/abs/1412.6980v8
37. Krizhevsky, A. K. A., Sutskever, I. S. I., Hinton, G. H. G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems(NIPS)*, 1097–1105. https://doi.org/10.1145/3065386

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

Zikai Yin is a master candidate at the school of Mechanical Engineering, Northwestern Polytechnical University. His research interests include intelligent manufacturing and computer vision.

Junxue Ren is a professor at the school of Mechanical Engineering, Northwestern Polytechnical University. His research interests include CNC machining, intelligent manufacturing and adaptive machining.

Yongshou Liang is a researcher at the school of Mechanical Engineering, Northwestern Polytechnical University. His research interests include CNC machining intelligent manufacturing and adaptive machining.