INTRODUCTION

In recent years, the world has been suffering from climate warming and energy crisis, forcing many countries to develop environmentally friendly new energy. Wind energy is one of them. Its acquisition also drives the development of wind power industry. The wind power generation technology is becoming more and more mature, and the utilization rate of wind energy has reached 95%. Blade as a key component of wind turbines, its cost accounts for about 20% of the installation cost of WT, and it is also a wind energy capture device. Hence, the detection and maintenance of WT blade is particularly important. Since the WT blade is exposed to the outside all years round, it is easy to be affected by harsh environment and complex working conditions. This leads to many defects in the WT blades, such as oil stains, paint peeling, corrosion, sand holes and so on, which greatly reduces the efficiency and performance of wind turbine. Consequently, the detection of WT blade surface defects is pretty important.

At present, there has been a lot of research on the detection of WT blades. The traditional blade defect detection methods are mainly based on nondestructive testing technology and vibration signal. There are also abundant sensors installed to detect blade defects, such as dynamic sensor, acoustic emission sensor (AES), multi-dimensional sensor monitoring network, etc. Inspired by the concept of transitivity of frequency response function, W. Yang studied a new method of WT blade condition monitoring (CM). The results show that the method can effectively detect and locate the defects of WT blades. Chin Shun Tsai proposed a new method of blade damage detection based on continuous wavelet transform. This method can be applied to the detection of WT blades.
identification of blade structure damage in many cases. Z. Fu proposed a reliability analysis method of wind turbine blade condition monitoring network based on wireless sensor network. The research provides theoretical and technical guidance for the establishment of high-performance wind turbine blade condition monitoring and control system.

However, some problems generally exist in these methods, the accuracy and safety of traditional detection methods cannot be guaranteed. In recent years, the development of machine learning also promotes the development of feature processing and gradually derived algorithms.

In order to explore a fast and accurate method to detect the WT blade defects, we improved the residual network with feature fusion, transfer learning method is also used to further improve the detection efficiency. The contributions of this paper are summarized as follows:

1. Deep learning is applied to the task of WT blade defect detection, and data preprocessing work is carried out.
2. A multi-feature fusion residual network structure is proposed.
3. Through experiments, the improved network is compared with 5 models, and its effectiveness is verified.
4. Transfer learning is carried out to speed up the process of training.

This paper is organized as follows: Sec II introduces acquisition and optimization of WT blade defect data. The description of algorithm in detail is given in Sec III. Sec IV focuses on the training and comparison with different algorithms. Finally, the conclusion and future work are summarized and prospected.

## 2 ACQUISITION AND OPTIMIZATION OF WT BLADE DEFECT DATA SET

Data are the basis of deep learning and the optimization of data also determines the optimization of deep network model. About 10,000 blade images were taken by UAV in a wind farm in East China under stable state. We used Dajiang mavic2 Pro UAV equipment to shoot WT blades, and the pitch angle of PTZ is controlled by the left wave wheel of the remote control. By deleting some invalid images that do not contain the WT blade parts and a large number of overlaps, the images of 7546 wind turbine blades were collected. The data set is divided into training set and test set. The former contains 6035 images, accounting for about 80%, and the latter contains 1511 images, accounting for about 20%. At the same time, the blade images are divided into five categories: normal situation, paint falling off, gelcoat falling off, surface erosion and oil stain.

Due to the uneven distribution of data between classes in the WT blade defect data set. Some data enhancement operations such as rotation, brightness, flip, color dithering are performed which is more suitable for training of deep network. Finally, the number of images was expanded to 9785, including 7819 training images and 1966 testing images. The resolution of each blade image taken by the UAV is 5472 × 3684, because the image size required for deep convolutional network input is 224 × 224, we adjust the WT blade data set to corresponding size to suit the network input. Figures 1 and 2 show the defect images of the WT blade and a series of WT blade fault enhanced images. The proportion and quantity comparison of each category of image data are shown in Figure 3A,B.
FIGURE 1 Example of defect images of the WT blade. (A) Normal, (B) Erosion, (C) Gelcoat off, (D) Oil Stain (E) Paint Off

FIGURE 2 Example of WT blade fault enhanced images

(A) Normal 18%
(B) Paint Off 17%
(C) Gelcoat Off 20%
(D) Erosion 15%
(E) Oil Stain 15%

(A) Proportion of different defect types of data set

(B) Number of training images and testing images in each classification

FIGURE 3 Comparison between different types of datasets. (A) Proportion of different defect types of data set, and (B) Number of training images and testing images in each classification
3 | DESCRIPTION OF ALGORITHM

3.1 | The network structure

In this section, a multi-feature fusion residual structure is proposed, including two feature fusion and three feature fusion residual structure. Two kinds of network structures are compared with residual structure and GoogLeNet’s Inception structure. The characteristics of the four network structures are introduced respectively, and the advantages of multi-feature fusion residual structure are described. The structure of the four networks is shown in Figure 4.

The residual network (ResNet) is mainly formed by stacking multiple residual structures. The network can train up to 152 layers. In the shallow residual structure, feature extraction is carried out by two 3x3 convolution kernels, and the input x and output F(x) are superimposed to obtain the final output H(x) = x + F(x), where F(x) is the residual of x and H(x). The deep residual structure consists of two 1x1 convolution kernels and a 3x3 convolution kernel. The former is used to raise and lower dimensions, while the latter is used to extract features. Because the residual network can effectively solve the degradation problem of deep network, the residual structure is selected as the research object. Of course, there are many improvements on the residual network. For example, M. Zhao proposed a deep residual shrinkage network to improve the accuracy of fault diagnosis to improve the feature learning ability of high noise vibration signals.23 Y. Zhang combined the residual network with some adaptive shortcuts to repair high-quality images, and achieved good results.24

GoogLeNet proposes the Inception structure that incorporates multi-scale features to enhance network performance, this connection structure can extract multiple feature information and effectively solve the overfitting problem caused by too deep network hierarchy. The feature is extracted by three filters with the size of 1 x 1, 3 x 3 and 5 x 5, which is stacked by a series of sparse structures. In addition, two auxiliary classifiers are added to solve the problem of gradient disappearance.

The deep network model adopted in this paper is improved on the basis of the ResNet network structure.
Inspired by the inception structure of GoogLeNet, the robustness of the model is improved by increasing the width of network and combining with various features. In addition, the proposed fusion feature is used to replace the single feature in the residual structure. Through comparing the performance of two-feature and three-feature fusion residual structure, the latter is better. The residual structure of three-feature fusion is selected to form the final residual structure of network framework. Two-feature fusion residual structure was extracted by feature fusion with convolution kernel with size of 3x3 and 5x5. Feature extraction of three-feature fusion residual structure was completed by convolution kernel with size of 1 × 1, 3 × 3 and 5 × 5 respectively. In both fusion methods, 1 × 1 convolution kernel is used for dimensionality reduction and increase the depth of the network. Finally, a Dropout layer with a probability of 0.2 is connected to prevent overfitting.\(^{25}\)

### 3.2 Loss function and optimization algorithm

After preprocessing the picture of WT defect by data enhancement method, the definition of loss function and gradient descent algorithm is a crucial part of the whole deep network which can directly affecting the accuracy of the model. The correlation between predicted value and real value can be compared by defining the loss function, and the weight of the model can be continuously updated by defining the gradient descent function. The loss function adopted in this paper is the cross entropy loss function,\(^{26}\) and the Adam optimization algorithm is adopted to complete the gradient descent.\(^{27}\) At the same time, Dropout is also used to solve the overfitting problem.\(^{28}\)

The loss function is defined as follows:

\[
J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log \left( h\theta \left( x^{(i)} \right) \right) + \left( 1 - y^{(i)} \right) \log \left( 1 - h\theta \left( x^{(i)} \right) \right) \tag{1}
\]

where \(m\) is the total number of samples, \(h\theta(x^{(i)})\) is the predicted value, \(y^{(i)}\) is the true value. The cross entropy function describes the distance between the actual output (probability) and the expected output (probability). The smaller the cross entropy is, the closer the two probability distributions will be. It compensates for the gradient dispersion of the traditional square deviation loss function and is more suitable as a loss function.

The final loss function is shown in Equation (4). \(J(\theta)1\) represents the loss of the auxiliary classifier, and \(J(\theta)2\) represents the loss of the network. We refer to the final loss weight distribution of GoogLeNet. The final loss is obtained by multiplying by the weights of 0.2 and 0.8 and adding them.

\[
J(\theta)1 = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log \left( h\theta \left( x^{(i)} \right) \right) + \left( 1 - y^{(i)} \right) \log \left( 1 - h\theta \left( x^{(i)} \right) \right) \tag{2}
\]

\[
J(\theta)2 = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log \left( h\theta \left( x^{(i)} \right) \right) + \left( 1 - y^{(i)} \right) \log \left( 1 - h\theta \left( x^{(i)} \right) \right) \tag{3}
\]

\[
J(\theta) = 0.2 \times J(\theta)1 + 0.8 \times J(\theta)2 \tag{4}
\]

The gradient descent algorithm is defined as follows:

\[
v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{5}
\]

\[
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{6}
\]

\[
\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{7}
\]

\[
\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{8}
\]

\[
\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_{t+T}}} \hat{m}_t \tag{9}
\]

Adam (adaptive moment estimation) optimization algorithm is used to complete gradient descent. This algorithm can maintain the previous square gradient and the exponential decay average of the gradient and calculate the adaptive learning rate of each parameter to update the weight.

In Equations (5) and (6), \(g_t\) represents the gradient, it averages the gradient and the square of the gradient, so that each updated value is relative to the previous value, \(m_t\) and \(v_t\) represent the first time mean of the gradient and the second time non-central variance of the gradient respectively. \(\beta_1\) and \(\beta_2\) are constants, which are used to control the exponential decay. The purpose of Equations (7) and (8) is to correct the large sliding average deviation in the initial stage. When \(\tau\) becomes larger and larger, the denominator approaches 0, and the correction is completed. \(\varepsilon\) is the learning rate, in order to make the model tend to converge. In Adam, the learning rate attenuation is realized through the self-adaptive learning rate method. Equation (9) is the final parameter update formula. It can be seen that the learning rate is not fixed, and the learning rate is different for each iteration. Compared with other
adaptive learning rate algorithms, Adam algorithm has a faster convergence speed and requires less memory, also it can effectively prevent the learning rate from disappearing and the fluctuation of loss function caused by high variance parameter.

3.3 | Transfer learning

The purpose of transfer learning is to apply the knowledge or model learned in one field or task to different but related fields or problems.\textsuperscript{29} In deep learning, transfer learning is mainly used in performing new convolution tasks. In the process of training, the input data can be retrained by making full use of the weight learned from previous data to speed up the process of training, some weights of convolution network are no longer updated gradiently, so as to reduce the training time of the model. Yu extracted the feature of the blade image through the convolution neural network trained on the ImageNet data set, and completed the defect recognition of the wind turbine blade by combining transfer learning and convolution network.\textsuperscript{30} Yang combined transfer learning and ensemble learning in blade defect detection task. The results show that the use of transfer learning not only improves the ability of feature extraction but also accelerates the training of network.\textsuperscript{31}

In this paper, the weights trained in Pascal VOC data set are used for transfer learning task. Due to the difference of final output categories between the data set in this paper and Pascal VOC data set, in the convolution network of this paper, all the parameter layers used for feature extraction are frozen and not updated, only the parameters of the fully connected layer used for classification are updated.

In the multi-feature fusion residual network, the weights trained before are saved. In transfer learning, the parameters of feature extraction are not updated, only the parameters of classification layer are updated.

3.4 | The network of the multi-feature fusion residual

Residual structure of the multi-feature fusion is shown in Figure 5. In a traditional ResNet34 residual network, each residual structure is cycled multiple times (3, 4, 6, 3). In this paper, our network structure is carried out only once, which greatly reduces the depth of the structure. The final network structure is connected by 4 multi-feature fusion residuals. After completing a multi-feature fusion residual structure, a convolution kernel with size of $3 \times 3$ is connected to adjust the size of feature map into the next multi-feature fusion residual structure. In addition, before the penultimate residual structure, an auxiliary classifier is added to solve the gradient disappearance problem. Finally, transfer learning is carried out to accelerate the training process.

The feature extraction module is shown in Table 1. Among them, conv1-conv4 is the module of size adjustment and feature extraction, while the module of inception1 - inception4 is the module of feature fusion. Filter size is the size of the convolution kernel. Filter number is the number of convolution kernels.

![Figure 5](image_url)
TABLE 2 Hyperparameters of classification model

| Hyperparameters | The numerical |
|-----------------|---------------|
| Learning rate   | 0.0008        |
| Iterations      | 300           |
| Exponential decay rate | 0.9, 0.999 |
| epsilon coefficient | 1e−8      |
| Batch Size      | 32            |

4 | TRAINING AND COMPARISON

This chapter mainly introduces the performance of seven neural network models in different situations, including 21 experimental results under the original data set, enhanced data set and transfer learning. This experiment is based on the Windows 10 operating system. GPU acceleration is performed in three configurations: RTX 2080 Ti, Titan XP and GTX 1080ti. Finally, the performance of models are verified by time, test accuracy, training loss, confusion matrix and F1 score. The experiment is carried out under the learning framework of pytorch. The hyperparameters involved in the experiment are shown in Table 2.

4.1 | The process of training

We implemented seven classification models and conducted 21 comparative experiments based on the original WT blade data set, enhanced data set and transfer learning. The training loss and test accuracy of different networks are shown in Figure 6.

Figure 6 shows the training loss and test accuracy of 7 classification models under the original WT data set, enhanced data set and transfer learning. In the picture, enhanced data (ED) represents the input of enhanced data into convolution network, and transfer learning (TL) represents the combination of convolutional network and transfer learning. All networks are stable after 300 iterations. Figure 6D shows the test accuracy comparison of 7 models, and Figure 6P is the comparison of test accuracy under transfer learning.

We can see from the performance of AlexNet that there is obvious over fitting phenomenon in the test accuracy due to the uneven distribution of types in the original WT data set. After adding data by means of data enhancement, the test accuracy not only increases but also tends to be stable. At the same time, after applying the weight of Pascal VOC data set to transfer learning, the training loss and test accuracy have been significantly improved. The final test accuracy of AlexNet on the data set of WT blade defects can reach about 75%. Similarly, the advantages of data enhancement and transfer learning are further confirmed in VGG, GoogLeNet and ResNet. The increase of data improves the test accuracy of each model, solves the over fitting phenomenon, and makes the curve region stable. The introduction of transfer learning significantly improve the iteration speed of each model, and the training loss and test accuracy tend to be stable after about 50 iterations. It is worth mentioning that the introduction of transfer learning does not reduce the accuracy of the model. In VGG and GoogLeNet networks, the addition of transfer learning further excavates the accuracy of the model. Finally, the test accuracy of VGG and GoogLeNet on the WT blade defect data set is about 90% and 88%, the accuracy of ResNet18 and ResNet34 is 91% and 92%.

Two kinds of multi-feature fusion residual networks are tested, among them, ResNer_2F represents two feature fusion residual network, ResNet_3F is called three feature fusion residual network. It can be seen from Figure 6L–M that the accuracy of both networks can meet the requirements of accurate detection of WT blade defects, and the final curve tends to be stable. It can be seen from Figure 6O that the test accuracy of the proposed multi-feature fusion residual network is slightly higher than that of other networks. Most importantly, from the enlargement trend of Figure 6O, it can be seen that the iterative speed of multi-feature fusion residual network is significantly faster than other networks. Therefore, the multi-feature fusion residual network has the characteristics of fast detection speed and meet the requirements of accurate detection of WT blade defects. Which is a method of rapid detection of blade defects in line with our needs. At the same time, the last figure shows that the introduction of transfer learning does not affect the final detection accuracy.
Table 3 shows the running time of each model under transfer learning and different configurations for one iteration. It can be seen that the time required for one iteration of multi-feature fusion residual network is relatively less. The iteration time of ResNet_2F and ResNet_3F is 12.1s and 14.2s respectively under RTX2080, because the complete training needs 300 iterations, the multi-feature fusion residual network can reduce 30–40 min compared with other traditional convolution neural networks.

4.2 Comparison of detection time of each model

FIGURE 6 Performance of 7 convolutional networks. (A) Training loss of AlexNet, (B) Test accuracy of AlexNet, (C) Training loss of VGG, (D) Test accuracy of VGG, (E) Training loss of GoogLeNet, (F) Test accuracy of GoogLeNet, (G) Training loss of ResNet18, (H) Test accuracy of ResNet18, (I) Training loss of ResNet34, (J) Test accuracy of ResNet34, (K) Training loss of ResNet_2F, (L) Test accuracy of ResNet_2F, (M) Training loss of ResNet_3F, (N) Test accuracy of ResNet_3F, (O) Test accuracy 7 models, (P) Test accuracy 7 models using transfer learning.
which improves the efficiency of the computer to a certain extent. In addition, the introduction of transfer learning also greatly reduces the detection time and the pressure on the device. To sum up, the combination of transfer learning and the proposed network is a new method we are looking for to quickly detect the defects of WT blades.

### 4.3 Comparison of F1 score of each model

F1 score is used to further evaluate the performance of the model. The F1 score of the five networks are shown in Table 4. F1 score is a measure of classification problem, which considers both the accuracy and recall of the model. It is also a harmonic average of accuracy and recall.

The calculation formula of F1 score:

\[
F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

The value range of F1 score is from 0 to 1. 1 represents the best output of the model, and 0 represents the worst output of the model. As shown in equation (10), precision is for our prediction results, which indicates how many positive examples in the sample are predicted correctly. It can be seen in Table 4 that the performance of the proposed network is relatively excellent. In addition, the performance of F1 score before and after transfer learning is well.

### 4.4 Comparison of confusion matrix of each model

Confusion matrix is the most intuitive and simple method to measure the accuracy of classification models. We select 149 WT blade pictures to evaluate the confusion matrix of two feature fusion networks. As shown in Figure 7.

In Figure 7, the diagonal of the confusion matrix represents the number of correct detections in each category. It can be seen that the performance of ResNet_3F is the better which it detects three more WT blade images than ResNet_2F.

### 5 Conclusion

In this paper, deep learning is applied to the identification of WT blade defects. In order to quickly detect the defects of WT blades, a residual network structure based on multi-feature fusion is proposed. Firstly, the UAV is used to shoot
the WT blade image. In view of the uneven distribution of classes, the data set is expanded by data enhancement to form a complete data set. At the same time, transfer learning is used to further compress the parameters of the model and reduce the detection time. Finally, the results show that transfer learning can greatly accelerate the training process of network while ensuring accurate detection, the accuracy of the proposed network can reach 93%, and the detection time is shorter than other networks, at the same time, the comparison of confusion matrix and F1-score further confirm the superiority of ResNet_3F.

Since the defect types of the WT blade are relatively less in this experiment, more blade defect types need to be labeled and input into the multi-feature fusion residual network to verify its performance repeatedly. At the same time, it is necessary to further increase the number of multiple features to improve the detection accuracy and discuss the number of multiple features in optimal network. In addition, the design of WT blade defect detection system interface which can accurately determine the types of defects is also a critical work in the future.

ACKNOWLEDGMENTS
This work was supported in part by the National Natural Science Foundation of China under Grant 61973209 and Grant 61703103, and in part by the Natural Science Foundation of Shanghai under Grant 20ZR1421200.

ORCID
Jiawei Zhu https://orcid.org/0000-0001-7248-3468

REFERENCES
1. Shanmugam L, Mani P, Joo YH. Stabilisation of eventtriggered-based neural network control system and its application to wind power generation systems. IET Control Theory Appl. 2020;14(10):1321-1333.
2. Chen Z, Guerrero JM, Blaabjerg F. A review of the state of the art of power electronics for wind turbines. IEEE Trans Power Electron. 2009;24(8):1859-1875.
3. Bakka T, Karimi H-R, Christiansen S. Linear parameter-varying modelling and control of an offshore wind turbine with constrained information. IET Control Theory Appl. 2014;8(1):22-29.
4. Besnard F, Bertil l L. An Approach for Condition-Based Maintenance Optimization Applied to Wind Turbine Blades. IEEE Trans Sustain Energy. 2010;1(2):77-83.
5. Canturk R, Slavkovsky E, Niezrecki C, Inalpolat M. Wind turbine blade damage detection using various machine learning algorithms. ASME International Design Engineering Technical Conference/Computer and Information in Engineering Conference (IDETC/CIE), Charlotte, NC, AUG, 21-24, 2016.
6. Yang B, Sun DB. Testing, inspecting and monitoring technologies for wind turbine blades: A survey. Renew Sustain Energy Rev. 2013;22:515-526.
7. González, Gómez A, Fassois SD. A supervised vibration-based statistical methodology for damage detection under varying environmental conditions & its laboratory assessment with a scale wind turbine blade [J]. J Sound Vib. 2016;366:484-500.
8. Tang J, Soua S, Mares C, et al. An experimental study of acoustic emission methodology for in service condition monitoring of wind turbine blades [J]. Renew Energy. 2016;99:170-179.
9. Yang K, Rongrong JA, Worden K. Damage detection in a laboratory wind turbine blade using techniques of ultrasonic NDT and SHM [J]. Strain. 2018;54(6):1-16.
10. Yang W, Lang Z, Tian W. Condition monitoring and damage location of wind turbine blades by frequency response transmissibility analysis. *IEEE Trans Indust Electr.* 2015;62(10):6558-6564.

11. Tsai C-S, Hsieh C-T, Huang S-J. Enhancement of damage-detection of wind turbine blades via cwt-based approaches. *IEEE Trans Energy Convers.* 2006;21(3):776-781.

12. Fu Z, Luo Y, Gu C, Li F, Yue Y. Reliability analysis of condition monitoring network of wind turbine blade based on wireless sensor networks. *IEEE Trans Sustain Energ.* 2019;10(2):549-557.

13. Chen P, Lin C, Scholkopf B. A tutorial on v-support vector machines. *Appl Stoch Models Bus Ind.* 2005;21(2):111-136.

14. Wang L, Zhang Z. Automatic detection of wind turbine blade surface cracks based on uav-taken images. *IEEE Trans Indust Electr.* 2017;64(9):7293-7303.

15. Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Trans Industr Inf.* 2015;11(3):812-820.

16. Krizhevsky IS, Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012:1097-1105.

17. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Computer Vision and Pattern Recognition. 2014.

18. Jia Y, Szegedy C, Liu W, Sermanet P. Going deeper with convolutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

19. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016;2016:770-778.

20. Xu D, Wen C, Liu J. Wind turbine blade surface inspection based on deep learning and UAV-taken images. *J Renew Sustain Energ.* 2019;11(5):53305.

21. Zhang C, Wen C, Liu J. Mask-MRNet: A deep neural network for wind turbine blade fault detection [J]. *J Renew Sustain Energ.* 2020;12(5):53302.

22. Chen X, Zhang B, Zhao J, Wu J. Ecg data compression using a neural network model based on multi-objective optimization. *PLoS One.* 2017;12:1-16.

23. Zhao M, Zhong S, Fu X, Tang B, Pecht M. Deep residual shrinkage networks for fault diagnosis. *IEEE Trans Indust Inf.* 2020;16(4):4681-4690.

24. Zhang Y, Sun L, Yan C, Ji X, Dai Q. Adaptive residual networks for high-quality image restoration. *IEEE Trans Image Process.* 2018;27(7):3150-3163.

25. Hinton G, Srivastava N, Krizhevsky A. Improving neural networks by preventing co-adaptation of feature detectors [J]. *Comp Sci.* 2012;3(4):212-223.

26. Shannon CE. A mathematical theory of communication. *Bell Syst Tech J.* 1948;27(3):379-423.

27. Kingma D, Ba J. Adam: A method for stochastic optimization. *Computerence.* 2014.

28. Srivastava N, Hinton G, Krizhevsky A. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research.* 2014;15:1929-1958.

29. Lin H, Hu J, Xiao Ding W, Alhamid MF, Jalil Piran M. Towards secure data fusion in industrial iot using transfer learning. *IEEE Transactions on Industrial Informatics,* pages1–1, 2020.

30. Yu Y, Cao H, Yan X, et al. Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor[J]. *Neurocomputing.* 2019;376:1-9.

31. Yang X, Zhang Y, Lv W, Wang D. Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier–ScienceDirect [J]. *Renewable Energy.* 2021;163:386-397.

32. Vakili M, Ghamarsi M, Rezaei M. Performance analysis and comparison of machine and deep learning algorithms for IoT data classification [J]. 2020.

**How to cite this article:** Zhu J, Wen C, Liu J. Defect identification of wind turbine blade based on multi-feature fusion residual network and transfer learning. *Energy Sci Eng.* 2022;10:219–229. doi:10.1002/ese3.1024