Efficient and Accurate Damage Detector for Wind Turbine Blade Images

LIANG LV1, ZHONGYUAN YAO2, ENMING WANG1, XIN REN1, RAN PANG2, HUA WANG1, YU ZHANG2, AND HAO WU1

1China Huaneng Clean Energy Research Institute, Beijing 102209, China
2Huaneng Yancheng Dafeng New Energy Power Generation Company Ltd., Nanjing, Jiangsu 210008, China

Corresponding author: Hao Wu (h_wu@qny.chng.com.cn)

This work was supported by the China Huaneng Group Company Ltd. Technology, Research and Application of Offshore Wind Power Intelligent Detection and Inspection Technology, under Project HNJK20-H44.

ABSTRACT The damage of wind turbine blades is one of the main problems restricting wind power development. Object detection can identify the damaged regions and diagnose the damage types. To handle the high-resolution wind turbine blade images, this article presents a novel efficient, and accurate damage detector (EADD) for wind turbine blade images. The proposed method adopts Single Shot MultiBox Detector (SSD) as the detection framework and offers an improved ResNet as the backbone. Firstly, the improved ResNet backbone uses dense connection blocks consisting of factorized depth-wise separable bottleneck (FDSB) and feature aggregation module (FAM), which makes the damage detection model more lightweight and has a faster detection speed. Secondly, the bidirectional cross-scale feature pyramid (BiFPN) is introduced into the proposed method to use multi-scale features fully and have more feature expression. In addition, data pre-processing, exponential moving average (EMA) and label smooth methods are utilized to improve the accuracy and robustness of the model. The experimental results on the wind turbine blade damage detection dataset show that our proposed method can achieve the best trade-off between detection accuracy and computation time compared with other competitive methods.

INDEX TERMS Wind turbine blade, damage detection, SSD, dense connection, BiFPN.

I. INTRODUCTION

Wind power is one of the fastest-growing energy sources with the development of clean energy [1]. The long-running wind turbines have some blade damages, which makes operators conduct regular inspections for the wind turbines. As the wind turbine’s vital component, the blades’ damage will reduce the power generation efficiency and cause hazardous accidents [2]. Because the wind farms are usually in remote areas and the wind turbines are huge and kinetic, manual inspection is time-consuming and expensive. Besides, the inconspicuous damage of the blade surface makes it difficult for human eyes to identify. Developing advanced technologies such as mobile devices and artificial intelligence [3] brings more intelligent inspection methods. Unmanned aerial vehicles with computer vision methods are used to detect the damage of the wind turbine blades [4], which greatly improves the efficiency and accuracy of the inspection.

Research on wind turbine blade damage detection can be divided into two categories: 1) damage detection methods based on machine vision, e.g. LBP [5] + SVM [6]; 2) damage detection methods based on deep learning. The machine vision-based methods have two shortcomings: 1) manually extracted features may be influenced by the environment, lighting and other factors, which cannot obtain a high signal-to-noise ratio and sufficient useful information. This causes low accuracy of the blade damage detection, which is difficult to meet the requirement of industrial application; 2) selecting suitable features consumes much time and fails to satisfy the requirement of real-time object detection.

Deep learning has made tremendous progress since the convolutional neural network was used in the ImageNet
competition [7], especially benefiting from large-scale, high-quality open source datasets. A large amount of object detection methods have been proposed based on ImageNet [7], PASCAL VOC [8] and MS COCO [9]. Several influential object detection methods include Faster R-CNN [10], Single Shot MultiBox Detector (SSD) [11] and YOLO [12].

All of the above methods are significant, but none of them can be directly applied to wind turbine blade images because of the following problems:

1. Multi-scale wind turbine blade damage is challenging to detect. For some categories, the damaged regions are tiny, sparse and vague. For some categories, the damaged regions are large;
2. Large size and high resolution of wind turbine blade images cause the computation to be time-consuming;
3. Since the camera angle is random, the damaged regions in the blade images can appear at any angle between 0° and 180°.

Notably, SSD shows excellent detection speed and high accuracy on the VOC dataset among the above algorithms, and the framework of SSD is highly transferable. To solve problems (1) and (2), we propose a novel, efficient and accurate damage detector (EADD) for wind turbine blade images. Considering the model’s speed, accuracy and flexibility, our proposed method is based on SSD. We develop an improved ResNet [13] as the backbone, which uses dense connection blocks consisting of the factorized depth-wise separable bottleneck (FDSB) and feature aggregation module (FAM). The improved ResNet can make the damage detection model more lightweight and have a faster detection speed. Furthermore, the bidirectional cross-scale feature pyramid (BiFPN) [14] is introduced into the proposed method to enhance multi-scale damage detection and take full advantage of multi-scale feature maps. To solve the problem (3), we pre-process the blade images through blade segmentation and rotation. The proposed method is evaluated on the wind turbine blade damage detection dataset we constructed. Experimental results verify the superiority of our proposed method compared with seven state-of-the-art methods.

The rest of this paper is described as follows: Section 2 introduces the previous research related to object detection and wind turbine blade damage detection. Section 3 elaborates the proposed damage detector for wind turbine blade images. In Section 4, we introduce the wind turbine blade damage detection dataset and conduct the damage detection experiments, then compare the results of our proposed method with seven other state-of-the-art methods. Finally, Section 5 summarizes this paper.

II. RELATED WORK

A. OBJECT DETECTION

Object detection based on deep learning can identify and locate the regions of interest in an image, which has the characteristics of fast speed, strong accuracy and high robustness. It can be divided into one-stage and two-stage detection algorithms. The two-stage detection algorithm was initially proposed in R-CNN [15]. The algorithm generates a series of regions of interest (ROIs) and then classifies them with a convolutional neural network. R-CNN [15] utilizes a traditional selective search strategy to generate the regions of interest (ROIs), each of which is extracted from the image and computed by the convolutional neural network separately. Therefore, R-CNN [15] has a lot of redundant computations. Faster R-CNN [10] introduces a region proposal network (RPN) as a replacement for the selective search strategy. The input of RPN is a set of prior bounding boxes (i.e., anchors). RPN makes the model more efficient and enables the network to be trained end-to-end. YOLO [12] and SSD [11] as the one-stage detection algorithms, removed the ROI pooling stage and detected objects directly by one neural network. The one-stage algorithms are usually faster than the two-stage algorithms and achieve satisfactory accuracy. SSD [11] sets up prior bounding boxes of different scales at different feature layers, and then performs classification and regression based on the prior bounding boxes. YOLO [12] predicts the coordinates and classes of objects directly without the use of anchors.

B. WIND TURBINE BLADE DAMAGE DETECTION

With the development of deep learning technology, object detection has been widely used in many fields, e.g., Kumar et al. [16] used the edge computing principle to propose a real-time multi-drone object damage detection system based on yolo-v3 for high-rise civil structures. Zhao Haoning et al. [17] designed an autonomous robot navigation system via motion planning and object detection. In order to deal with the restricted feature extraction ability of CNNs in low-resolution infrared imagery, Zhang Ruiheng et al. proposed a target detector in infrared imagery named Deep-IRTarget [18]. Zhang Ruiheng et al. also proposed a robust multi-player tracker incorporating with deep player identification [19] for basketball players in the multi-camera sports video.

Before unmanned aerial vehicles were widely used, most research on wind turbine blade damage detection mainly focused on focused on acquiring and processing sensor signals. The commonly used methods include acoustic emission technology [20], vibration detection [21], strain detection [22], infrared thermography [23], ultrasonic flaw detection [24], etc. However, there are some issues with sensor installation, data storage and transmission. The change in the environment easily disturbs the signals obtained by the sensors. Furthermore, installing a large number of sensors on the wind turbine blades will affect the performance of the blades to capture wind energy.

With the development of unmanned aerial vehicle technology equipped with high-resolution cameras and the wide application of deep learning technology in the field of object detection, a new method has been developed for wind turbine blade damage detection. Yang Yunxi et al. [25] employed transfer learning and ensemble classifiers to detect wind...
turbine blade damage. Long Wang et al. [26] proposed a data-driven framework for the automatic detection of wind turbine blade surface cracks by using unmanned aerial vehicles. Dipu Sarkar et al. [27] proposed a YOLOv3-based UAV image recognition model for wind turbine blade damage. ASM Shihavuddin et al. [28] used the Inception-ResNet-V2 architecture for Faster R-CNN to propose an efficient automatic detection method for wind turbine blade damage.

III. PROPOSED METHOD
A. FRAMEWORK OF OUR PROPOSED METHOD
The framework of our proposed damage detection method for wind turbine blade images is shown in Figure 1, which is based on the SSD structure, consisting of backbone, neck and detection head. The backbone (blue dotted line) of the proposed method takes advantage of the ResNet-50 [13] (Conv1->Conv5) with dense connection blocks (blue cube) consisting of several factorized depth-wise separable bottlenecks (FDSB) and a feature aggregation module (FAM). The neck (grey rectangle box) of the proposed method employs the bidirectional cross-scale feature pyramid (BiFPN) [14]. Then three additional dense connection blocks are introduced as the input (D3, D4, D5) of the neck. Two feature maps from Conv4 (D1) and Conv5 (D2) are also as the input of the neck. The feature maps are aggregated through BiFPN and input to the detection head (green rectangle box) to obtain the detection results.

B. DENSE CONNECTION BLOCKS
The proposed dense connection block (see Figure 2) consists of N factorized depth-wise separable bottlenecks (FDSB) and a feature aggregation module (FAM). The feature maps extracted by FDSB are input into the FAM and aggregated through dense connections. The channel attention of FAM makes the network focus on the important channels and suppress useless information.

C. FACTORIZED DEPTH-WISE SEPARABLE BOTTLENECK
Figure 3 shows the structure of different types of residual blocks. The basic residual block [13] can effectively improve the performance of the neural network, but with the deepening of the number of network layers, the use of standard convolution kernels leads to a huge amount of parameters and calculation costs, so some efficient convolution kernels
FIGURE 3. The structure of different types of residual blocks. From left to right are (a) basic residual block [13], (b) bottleneck [13], (c) MobileNet [29], (d) non-bottleneck-1D [30], and (e) our FDSB. “Conv” is a standard convolution kernel, “DSConv” denotes depth-wise separable convolution, “FConv” denotes 1D-factorized convolution kernel, and “FDSConv” denotes factorized depth-wise separable convolution kernel.

and residual structures are proposed. Bottleneck [13] replaces two $3 \times 3$ convolution kernels with “$1 \times 1$ convolution kernel -> $3 \times 3$ convolution kernel -> $1 \times 1$ convolution kernel” to compress or increase the number of feature map channels flexibly. MobileNet [29] and non-bottleneck-1D [30] use the depthwise separable convolution and 1D-factorized convolution kernels to reduce the number of parameters and computation cost in place of the standard convolution kernel respectively. The proposed FDSB is a novel residual block, it retains the structure of the bottleneck, and combines depth-wise separable convolution (DSConv) and the 1D-factorized convolution (FConv) kernels to propose the factorized depth-wise separable convolution (FDSConv), FDSConv can further reduce the number of parameters. Channel shuffle [31] is utilized in FDSB to enhance channel information interaction among channels and maintain great representation capacity. We compare different types of convolutional kernels in Table 1, comparison results show our FDSConv has the fewest parameters.

TABLE 1. Comparison of different types of convolutional kernels. $n$ is the size of the convolutional kernel, $c_{in}$ and $c_{out}$ represent the input and output channel numbers respectively.

| Convolutional type | Parameters |
|--------------------|------------|
| Standard           | $n^2 \times c_{in} \times c_{out}$ |
| DSConv             | $n^2 \times c_{in} + c_{in} \times c_{out}$ |
| FConv              | $2 \times n \times c_{in} \times c_{out}$ |
| FDSConv            | $2 \times n \times c_{in} + c_{in} \times c_{out}$ |

D. FEATURE AGGREGATION MODULE

As shown in Figure 4, we first concatenate the output feature maps from each FDSB, then we implement the convolution, batch normalization (BN) and Relu activation to refine the concatenated feature maps from dense connections. Next, a weight vector of each channel is computed through global average pooling (gap) [32], convolution and sigmoid activation. The weight vectors can re-weight the channels to retain the correlation information among channels.

E. BiFPN

In order to obtain more sufficient feature representations, we adopt BiFPN [14] to replace the multi-scale features in SSD. As shown in Figure 5, $D_1^m$, $D_2^m$, $D_3^m$, $D_4^m$, and $D_5$ are the input of the BiFPN. $D_i^{out}(i = 1, 2, 3, 4, 5)$ is the output of BiFPN and $D_i^{out}(i = 2, 3, 4)$ represents the intermediate node of BiFPN. The deep semantic features are transmitted from $D_5$ to $D_1$ from top to bottom. The low-level position information is transmitted from $D_1$ to $D_5$ from bottom to top. The detailed computation of BiFPN is described as follows:

$$D_1^{out} = \text{Conv}(w_{11}^{out}D_1 + w_{12}^{out}D_2^{m} + w_{13}^{out}D_3^{m} + \varepsilon_1^{out})$$

$$D_i^{m} = \text{Conv}(w_{i1}^{m}D_i + w_{i2}^{m}\text{Resize}(D_{i+1}^{m}), i = 2, 3, 4)$$

$$D_i^{out} = \text{Conv}(w_{i1}^{out}D_i + w_{i2}^{out}D_i^{m} + w_{i3}^{out}\text{Resize}(D_{i-1}^{out})^{out}, i = 2, 3, 4)$$
\[ D_5^{out} = \text{Conv}(\frac{w_5^{out} D_5 + w_2^{out} D_2^{out}}{w_5^{out} + w_2^{out} + \varepsilon_5^{out}}) \]  

(4)

where \( w_5^{out} \) and \( w_2^{out} \) are the learnable weights of each node participating in the operation, \( \text{Resize()} \) is the size scaling function, \( \text{Conv()} \) is the convolution operation and \( \varepsilon \) is a small value to ensure the stability of the calculation.

**IV. DATA PRE-PROCESSING AND MODEL SETTING**

In this section, we introduce the training and testing experiments of our proposed method.

F. DETECTION HEAD

After feature extraction and enhancement, the network takes the features as the input of the detection head and implements the detection computation. Specifically, the detection head is similar to SSD and consists of two convolution layers: 3 × 3 Conv layer and 1 × 1 Conv layer. The shape of the output of each layer is \( N \times (C + 4) \), where \( N \) is the number of anchor boxes in each layer, \( C \) is the number of categories and four values are the predicted offsets of \( \text{xywh} \).

A. DATA PRE-PROCESSING

1) BLADE SEGMENTATION AND ROTATION

Considering the particular shape of wind turbine blades, some of the blade damage is long and narrow, which is along the blade length, and the bounding boxes of object detection methods are regular rectangles rather than rotated rectangles, there will be a proportion of the background in the bounding boxes of damage targets, this is not conducive to labeling the damaged area (see Figure 6 (a)), and also interferes the detection performance of the model.

We pre-process the blade images through blade segmentation and rotation to help the object detection model detect the damage better. Image segmentation can accurately extract the contour of the target in an image via labeling each pixel with a category [33], [34]. Firstly, we train a segmentation model to extract the blade area (see Figure 6 (b)), then we present a blade rotation method to rotate the blade in the vertical direction, as Figure 6 (c) shows, the new manually annotations are closely fit the damaged area of the wind turbine blades without much background area. The procedure of blade rotation process has two steps:
1) Calculate the minimum enclosing rectangle of the blade.
2) Calculate the angle of the blade according to the long and short sides of the rectangle and rotate the picture.

The damage bounding box is rotated synchronously.

**B. MODELING SETTING**

In order to guide the network training, we propose the following two strategies:

1) **EXPONENTIAL MOVING AVERAGE (EMA)** [35]

EMA uses exponential decay to calculate the average weights in the model training process, which can make the model more robust on test data. The underlying mechanism is that using the moving average of the weights usually works better than using the ending weights at the end of training. The formula is as follows:

\[
W_{EMA} = \lambda \ast W_{EMA} + (1 - \lambda) \ast W,
\]

where \(W_{EMA}\) is the moving average weight, \(W\) is the weight updated in the current period and \(\lambda\) is the weighting factor.

In practical applications, \(\lambda\) is generally set to a number very close to 1 [0.9, 1], we set \(\lambda = 0.998\) in our experiments.

2) **LABEL SMOOTH** [36]

The traditional hard-coded label is either 0 or 1 and the predicted value utilizes the sigmoid activation function. The gradient of the sigmoid function tends to be very small or very large if it equals 0 or 1. This causes the model to overfit easily.

To overcome this problem, the original hard-coded label is used to smooth the processing method with the following equation:

\[
y' = (1 - \varepsilon) \ast y + \varepsilon \ast \mu,
\]

where \(y'\) is the sample label after label smoothing, \(\varepsilon\) is the smoothing parameter (=1/number of categories) and \(\mu\) is a hyperparameter to introduce noise with fixed distribution to the probability distribution

**C. LOSS FUNCTION**

As equation 7 shows, the loss function includes focal loss [37] and iou loss [38], which are employed for classification and regression, respectively.

\[
L_{total} = L_{focal} + L_{iou}
\]

The detail of focal loss and iou loss are shown in equation 8 and 9:

\[
L_{focal} = \alpha_t (1 - p_t)^\gamma \log(p_t)
\]

where \(p_t\) is the predicted probability, \(\alpha_t\) and \(\gamma\) are hyper-parameters.

\[
L_{iou} = 1 - \left( \frac{\hat{B} \cap B}{\frac{\hat{B} \cup B}{\hat{B} \cup B} - \frac{\rho^2(\hat{b}, b)}{c^2}} \right)
\]

where \(\rho^2(\hat{b}, b)\) means the Euclidean distance between two center points, \(c\) is the diagonal distance of the smallest rectangle that can enclose \(B\) and \(\hat{B}\).

**V. EXPERIMENTAL VALIDATION AND EVALUATION**

**A. DATASET AND EVALUATION METRICS**

An unmanned aerial vehicle (UAV) is used to collect more than 10,000 high-resolution wind turbine blade images (4000 × 3000 pixels in size) in a wind farm in China, then construct the training, validation, and testing datasets with the ratio of 7:2:1. For this work, three common damage are defined: Gelcoat peeling off (GPO), Surface cracking (SCR) and Surface corrosion (SCO). The evaluation indexes include FPS (Frames Per Second), the size of network parameters (Param.), and floating point operations (FLOPs) are used for speed evaluation, mAP is used for performance evaluation, which is the mean of average precision of all the classes. The precision is computed as:

\[
Precision = \frac{TruePositive}{TruePositive + FalsePositive},
\]

where if the IOU is more than the threshold (IOU = 0.5), it is classified as True Positive. Otherwise, it is classified as False Positive.

**B. IMPLEMENTATION DETAILS**

In this paper, EADD is compared with seven state-of-art object detection methods, and all experiments are implemented with the same training strategy via Pytorch on a Tesla V100 GPU. An improved ResNet-50 is utilized as the backbone network. BiFPN is used as the feature fusion layer (Neck) and the detection head. First, the backbone network is pre-trained based on ImageNet. For the BiFPN and detection head, the Xavier initialization strategy is applied. The Adam optimizer is used for training with 120k iterations. The initial learning rate is set to 0.01, \(\alpha_t\) and \(\gamma\) are set to 0.25 and 2, respectively. The learning rate will decrease by five times at 40k and 80k. Due to the large size of the original images, for the comparison of time cost, we conduct model prediction on an input with the size of 320×320 pixels, and the batch size is set to 16. We also implement data augmentation to address the issue of data imbalance [39], including random flips, rotates, and random changes in brightness, contrast and saturation in the range of [0.5, 2.0].

**C. EXPERIMENTAL EVALUATION AND COMPARISON**

In Table 2, the ablation study on the test dataset demonstrates the improvements by introducing the proposed modules of EADD. We use the SSD network as the baseline, when ResNet replaces the backbone VGG (SSD + ResNet), both the performance and speed of the model are enhanced. After using DCB to improve ResNet (SSD + ResNet_DCB), the model’s accuracy decreases slightly. Still, the speed is greatly
TABLE 2. Ablation study on the test dataset, ResNet_DCB represents ResNet improved via dense connection blocks, BiFPN represents bidirectional cross-scale feature pyramid.

| Method            | GPO  | SCR  | SCO  | mAP  | FPS | Param.(M) | FLOPs (G) |
|-------------------|------|------|------|------|-----|-----------|-----------|
| SSD               | 74.9%| 76.7%| 73.4%| 75.0%| 40  | 23.8      | 35.2      |
| SSD + ResNet      | 77.7%| 80.3%| 76.6%| 78.2%| 44  | 21.7      | 27.5      |
| SSD + ResNet_DCB  | 77.5%| 80.1%| 75.9%| 77.8%| 62  | 7.2       | 11.2      |
| SSD + BiFPN       | 77.7%| 79.8%| 76.1%| 77.9%| 37  | 25.1      | 38.9      |
| EADD              | **80.8%**| **82.3%**| **80.5%**| **81.2%**| 56  | 8.3       | 12.8      |

TABLE 3. Comparisons between EADD with seven state-of-the-art models on the test dataset.

| Method           | GPO  | SCR  | SCO  | mAP  | FPS | Param.(M) | FLOPs (G) |
|------------------|------|------|------|------|-----|-----------|-----------|
| Faster R-CNN [10] | 78.5%| 79.4%| 78.2%| 78.7%| 12  | 60        | 246       |
| YOLO-V3 [40]     | 77.3%| 78.7%| 77.6%| 77.8%| 65  | 7.1       | 9.7       |
| SSD [11]         | 74.9%| 76.7%| 73.4%| 75.0%| 40  | 23.8      | 35.2      |
| YOLO-V4 [41]     | 77.5%| 79.2%| 78.1%| 78.2%| 68  | 6.1       | 8.6       |
| RetinaNet [37]   | 81.2%| 83.6%| 80.9%| 81.9%| 25  | 38        | 205       |
| EfficientDet [14]| 80.9%| 82.5%| 79.8%| 81.0%| 49  | 12        | 24.9      |
| DETR [42]        | **82.5 %**| **84.5%**| **81.6%**| **82.8%**| 10  | 62        | 253       |
| EADD             | 80.8%| 82.3%| 80.5%| 81.2%| 56  | 8.3       | 12.8      |

FIGURE 7. The damage detection results of the proposed method, where red boxes represent gelcoat peeling off, green boxes represent surface cracking, black boxes represent surface corrosion.

ehanced (FPS + 18), and the number of parameters (from 21.7M to 7.2M) and FLOPs (from 27.5G to 11.2G) are greatly reduced. mAP increased by 2.9% when BiFPN is introduced into SSD (SSD + BiFPN) with limited parameters (+1.3M) and FLOPs (+3.7G). When both modules are used (EADD), mAP (81.2%), FPS (56), Param. (8.3M) and FLOPs (12.8G) are notably improved. Therefore, compared to SSD, the proposed backbone network optimization and multi-scale feature fusion bring significant performance promotion.

To verify the performance of our proposed method, EADD is compared to the Faster R-CNN [10], YOLO-V3 [40], SSD [11], YOLO-V4 [41], RetinaNet [37], EfficientDet [14]
and DETR [42] with mAP, FPS, Param., and FLOPs as the metric (see Table 3) in the same experiment setting on the test dataset. Our proposed method performs much better than the two-stage detection method Faster R-CNN (+2.5% mAP, +44 FPS, −51.7M Param., and −233.2G FLOPs). Compare with four real-time object detection methods YOLOV3, SSD, YOLO-V4, and EfficientDet, the proposed EADD balance the model accuracy and speed better, although the YOLO-V4 has the fewest parameters and FLOPs, EADD achieves a higher mAP (+3%). It is worth noticing although the mAP of EADD is lower than those of RetinaNet and DETR, the parameter size and FLOPs of RetinaNet (38M, 205G) and DETR (62M, 253G) are several times than EADD (8.3M, 12.8G), their FPS (25, 10) indexes are far less than EADD (56). Compared with other competitive methods, the proposed EADD can achieve the best trade-off between detection accuracy and computation time. The model predictions are shown in the following figure (see Figure 7).

In Table 4, ablation experiments show that exponential moving average (EMA) and label smooth (LS) can effectively improve the mAP of EADD.

### VI. CONCLUSION

In this paper, we present a fast and accurate damage detector (EADD) for wind turbine blade images. The proposed method uses SSD as the detection framework and proposes an improved ResNet as the backbone. The improved ResNet backbone utilizes dense connection blocks composed of FSDB and FAM, which makes the network more lightweight and have a faster detection speed. Meanwhile, BiFPN is used to aggregate multi-scale feature maps. Furthermore, data preprocessing, EMA and label smooth are used to reinforce the accuracy and generalization ability of the proposed blade damage detector. Compared with other competitive methods, the experimental results on the wind turbine blade damage detection dataset show that our proposed method can achieve the best trade-off between detection accuracy and computation time.

### REFERENCES

[1] B. Obama, “The irreversible momentum of clean energy,” Science, vol. 355, no. 6321, pp. 126–129, 2017.

[2] D. S. Li, S.-C. M. Ho, G. B. Song, L. Ren, and H. N. Li, “A review of damage detection methods for wind turbine blades,” Smart Mater. Struct., vol. 24, no. 3, 2015, Art. no. 033001.

[3] E. Rich, Artificial Intelligence. New York, NY, USA: McGraw-Hill, 1983.

[4] M. Stokkeland, K. Klausen, and T. A. Johansen, “Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection,” in Proc. Int. Conf. Unmanned Airc. Syst., 2015, pp. 998–1007.

[5] Z. Guo and D. Zhang, “A completed modeling of local binary operator for texture classification,” IEEE Trans. Image Process., vol. 19, no. 6, pp. 1657–1663, Jun. 2010.

[6] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Apr. 1995.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 1097–1105.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, pp. 303–338, Jul. 2010.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 740–755.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 91–99.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 21–37.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection.” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[14] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10781–10790.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[16] P. Kumar, S. Batchu, N. Swamy, and S. R. Kota, “Real-time concrete damage detection using deep learning for high rise structures.” IEEE Access, vol. 9, pp. 112312–112331, 2021.

[17] H. Zhao, C. Wang, R. Guo, X. Rong, J. Guo, Q. Yang, L. Yang, Y. Zhao, and Y. Li, “Autonomous live working robot navigation with real-time detection and motion planning system on distribution line.” High Voltage, Jun. 2022, doi: 10.1049/hve.2.12221.

[18] R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu, “Deep-IRTarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation,” IEEE Trans. Multimedia, vol. 24, pp. 1735–1749, 2021.

[19] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, “Multi-camera multi-player tracking with deep player identification in sports video,” Pattern Recognit., vol. 102, Jun. 2020, Art. no. 107260.

[20] B.-H. Han, D.-J. Yoon, Y.-H. Huh, and Y.-S. Lee, “Damage assessment of wind turbine blade under static loading test using acoustic emission,” J. Intell. Mater. Syst. Struct., vol. 25, no. 5, pp. 621–630, Mar. 2014.

[21] Y. Wang, M. Liang, and J. Xiang, “Damage detection method for wind turbine blades based on dynamics analysis and mode shape difference curvature information,” Mech. Syst. Signal Process., vol. 48, nos. 1–2, pp. 351–367, Oct. 2014.

[22] J. Sierra-Pérez, M. A. Torres-Arredondo, and A. Güemes, “Damage and nonlinearities detection in wind turbine blades based on strain field pattern recognition. FBGs, OBR and strain gauges comparison.” Compos. Struct., vol. 135, pp. 156–166, Jan. 2016.

[23] M. A. Runsey and W. Musial, “Application of infrared thermography nondestructive testing during wind turbine blade tests,” J. Sol. Energy Eng., vol. 123, no. 4, pp. 271, 2001.

[24] C. Yin, Z. Zhang, Z. Wang, and H. Guo, “Numerical simulation and experimental validation of ultrasonic de-icing system for wind turbine blade,” Appl. Acoust., vol. 114, pp. 19–26, Dec. 2016.

[25] X. Yang, Y. Zhang, W. Lv, and D. Wang, “Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier,” Renew. Energy, vol. 163, pp. 386–397, Jan. 2021.

[26] L. Wang and Z. Zhang, “Automatic detection of wind turbine blade surface cracks based on UAV taken images,” IEEE Ind. Electron. Mag., vol. 64, no. 9, pp. 7293–7303, Sep. 2017.
D. Sarkar and S. K. Gunturi, “Wind turbine blade structural state evaluation by hybrid object detector relying on deep learning models,” J. Ambient Intell. Humanized Comput., vol. 12, no. 8, pp. 8535–8548, Aug. 2021.

A. Shihavuddin, X. Chen, V. Fedorov, A. N. Christensen, N. A. B. Riis, K. Brauner, A. B. Dahl, and R. R. Paulsen, “Wind turbine surface damage detection by deep learning aided drone inspection analysis,” Energies, vol. 12, no. 4, p. 676, Feb. 2019.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

E. Romero, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, “ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 263–272, Jan. 2018.

X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6848–6856.

X. Lu, Q. Chen, and S. Yan, “Network in network,” 2013, arXiv:1312.4400.

E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, “ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 263–272, Jan. 2018.

X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 8960–8970.

A. Shihavuddin, X. Chen, V. Fedorov, A. N. Christensen, N. A. B. Riis, K. Brauner, A. B. Dahl, and R. R. Paulsen, “Wind turbine surface damage detection by deep learning aided drone inspection analysis,” Energies, vol. 12, no. 4, p. 676, Feb. 2019.

D. Haynes, S. Corss, and G. K. Venayagamoorthy, “An exponential moving average algorithm,” in Proc. IEEE Conge. Evol. Comput., Jun. 2012, pp. 1–8.

M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar, “Does label smoothing mitigate label noise?” in Proc. Int. Conf. Mach. Learn., 2020, pp. 6448–6458.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, “IoU loss for 2D/3D object detection,” in Proc. Int. Conf. 3D Vis. (DV), Sep. 2019, pp. 85–94.

X. Lu, C. Ma, J. Shen, X. Yang, I. Reid, and M.-H. Yang, “Deep object tracking with shrinkage loss,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 5, pp. 2386–2401, May 2020.

J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.

A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 213–229.

ZHONGYUAN YAO received the B.S. and M.S. degrees from Tsinghua University, in 2007 and 2011, respectively. He is currently a Senior Engineer with Huaneng Jiangsu Clean Energy Branch. His research interests include new energy power generation and new energy standardization.

HAO WU received the B.S. and Ph.D. degrees from Tsinghua University, in 2016 and 2021, respectively. He is currently an Engineer with the China Huaneng Clean Energy Research Institute, Beijing, China. His research interests include control and optimization of new energy, image recognition, and carbon dioxide emission management.

LIANG LV received the B.S. degree from the Wuhan Institute of Technology, in 2016, and the M.S. degree from Shanghai Jiaotong University, in 2021. He is currently an Assistant Engineer with the China Huaneng Clean Energy Research Institute, Beijing, China. His research interests include deep learning, image processing, object detection, and semantic segmentation.

HUA WANG received the B.S. and M.S. degrees from North China Electric Power University, in 2007 and 2010, respectively. He is currently a Senior Engineer with the China Huaneng Clean Energy Research Institute, Beijing, China. His research interests include new energy power generation and new energy standardization.

YU ZHANG received the B.S. degree from the Nanjing University of Science and Technology, in 2006. He is currently an Assistant Engineer with Huaneng Jiangsu Clean Energy Branch. His research interests include new energy power generation and new energy standardization.