scientific reports

Check for updates

OPEN

Algorithm for detecting surface defects in wind turbines based on a lightweight YOLO model

Zhenjie Wu, Yulu Zhang, Xiang Wang, Haofei Li, Yuyang Sun & Gang Wang✉

Improving wind power generation efficiency and lowering maintenance and operational costs are possible through the early and efficient diagnosis and repair of surface defects in wind turbines. To solve the lightweight deployment difficulty and insufficient accuracy issues of the traditional detection methods, this paper proposes a high-precision PC-EMA block based on YOLOv8 using partial convolution (PConv) combined with an efficient multiscale attention (EMA) channel attention mechanism, which replaces the bottleneck layer of the YOLOv8 backbone network to improve the extraction of target feature information from each layer of the network. In the feature fusion phase, GSConv, which can retain more channel information, is introduced to balance the model's complexity and accuracy. Finally, by merging two branches and designing the PConv head with a low-latency PConv rather than a regular convolution, we are able to effectively reduce the complexity of the model while maintaining accuracy in the detection head. We use the WIoUv3 as the regression loss for the improved model, which improves the average accuracy by 5.07% and compresses the model size by 32.5% compared to the original YOLOv8 model. Deployed on Jetson Nano, the FPS increased by 11 frames/s after a TensorRT acceleration.

Keywords Wind turbines, Partial convolution, Channel attention mechanism, GSConv, YOLOv8

One of the major renewable energy sources for powering homes and businesses is wind energy[1,2]. The advancement of wind power technology has resulted in the installation of numerous wind turbines. These generators frequently work under challenging conditions and endure lengthy periods of complicated, fluctuating loads, which can cause many failures. In addition to increasing operating expenses, fixing and maintaining these malfunctions shortens the wind turbine's lifespan[3]. Based on statistical data[4], maintenance expenses make up approximately 25–30% of the total wind power expenditure. Reducing operational expenses and increasing power generation efficiency are two benefits of early defect detection and resolution[5].

During operation, wind turbines encounter a variety of complex force conditions, such as torsion loads, inertia forces, and bending loads[6]. In addition, wind turbine towers may experience fracturing, ice, and spalling, which can cause various degrees of surface imperfections and damage[7]. It is essential to identify and address all the forms of turbine problems as soon as possible. Process or human error is a frequent cause of manufacturing problems in the production of wind turbine blades. For instance, interlayer folding and the introduction of minute impurities are likely to occur during the delamination process; the absence of glue and air bubbles during vacuum infusion molding may occur, resulting in the formation of fundamental flaws in the blades. The operational height of large-scale wind turbine blades often exceeds 90 m, while their lengths range from 50 to 80 m. Consequently, relying solely on periodic manual maintenance for fault detection is not feasible, especially in the case of offshore wind farms.

There are now two categories of wind turbine surface flaw detection devices in use both domestically and internationally. The first uses nondestructive testing techniques such as ultrasonic and infrared thermal imaging, while the second uses unmanned aerial vehicle (UAV) technology to image visible light defects. Because deep learning detection targets do not need as many algorithms to support them, they are easier to use and are faster than the normal detection techniques. Convolutional neural networks (CNNs) have many applications in the industrial inspection field. There are two-stage or one-stage detectors, the representative two-stage detectors are R-CNN[8], Faster-RCNN[9], and Mask R-CNN[10], which first generate the candidate frames of the potential targets and then determine the target category and correct the candidate frames. While there are certain benefits to two-stage detectors in terms of detection accuracy, there are drawbacks as well, including training challenges, sluggish detection speeds, and optimization. One-stage algorithms are single-stage target detection algorithms,

School of Electrical and Information Engineering, Beihua University, Jilin 132021, China. ✉email: bhwanggang@163.com

which are represented by the Single Shot MultiBox Detector (SSD)[11] and the You Only Look Once (YOLO)[12–17] series, which can directly predict the confidence of the category and locate the target position on the image. The single-segment YOLO series of algorithms simultaneously performs target identification and boundary regression, and is widely used in industry because of its fast detection speed, small model size, and flexible deployment.

Wu et al.[18] proposed the deployment of a large flexible strain gauge network with a strain derivation algorithm. The algorithm can be used to detect crack expansions in real wind turbine blades. Wang et al.[19] proposed a data-driven framework to automatically detect surface cracks on wind turbine blades photographed by UAVs. The crack region is described by Haar features, and a cascade classifier is trained to detect cracks. Yu et al.[20] presented a method for merging infrared images of wind turbine blades using a U-shaped neural network (U-Net) and flight data from unmanned aerial vehicles (UAVs). The panoramic infrared image of the wind turbine blade is obtained by combining the U-Net network with the parameters after the intricate background information is eliminated and the splicing zone of the blade is retained. Sahir Moreno et al.[21] introduced a CNN-based deep learning vision approach to detect certain common blade damages and proposed a concept for the automatic detection of wind turbine blade damage with robots. Martin Stokkeland et al.[22] proposed a vision module that enables a drone to independently calculate the distance to a wind turbine blade and detect the blade by means of a Hough transform algorithm. Mao et al.[23] designed a model called a context aligned-deformable cascade R-CNN for automatic detection of wind turbine defects using a cascade R-CNN as a framework. A categorized and labeled defect dataset was produced to incorporate transfer learning ideas to enhance the convergence speed and model robustness. Qiu et al.[24] proposed an autonomous visual inspection system for WTBs that combines the YOLO and CNN architectures. Their approach leverages the intermediate layers of the CNN to collect more precise information about small objects. Additionally, they introduce a multi-scale feature pyramid to merge complementary features, thereby enhancing the model's extraction capabilities. To evaluate the effectiveness of their model in detecting small object defects, they created a WTB dataset comprising 23,807 images. Yang et al.[25] proposed a image recognition model of wind turbine blade damage that combines transfer learning and integrated learning with a random forest-based classifier. Initially, the wind turbine blade images undergo preprocessing using the Otsu method to eliminate complex backgrounds. Subsequently, transfer learning and integrated learning techniques are utilized in the detection task to improve the convergence speed and detection accuracy of the model. Zhang et al.[26] proposed a surface defect detection model for the WTB, which adds a microscale detection layer on top of YOLOv5 and utilizes the K-means algorithm to recluster the anchor points, and in the CBAM, an attention mechanism is used in each feature fusion layer, and a channel pruning algorithm is used to reduce the size of the model.

Currently, infrared and ultrasonic inspection methods have high accuracy advantages and the ability to detect deep and internal defects in the wind turbines. However, these methods have numerous problems; real-time outdoor inspections cannot be implemented, the inspection process is complicated and costly, and the detection of surface defects in wind turbines is poor. In contrast, the wind turbine surface defect detection method based on sample images taken by UAVs flying remotely transmits and utilizes background target detection algorithms for detection, localization, and identification. However, there are several problems with the current deep learning-based wind turbine defect detection methods; the high computational complexity of most high-precision algorithms leads to slow detection speeds when embedded in UAV equipment, thus reducing detection efficiency. Therefore, there are still shortcomings in the various current detection methods.

In complex environments, the existing algorithms often have difficulty effectively extracting the defective features of wind turbines, and fail to synthesize the multiscale target features, resulting in inaccurate detections. Erroneous detections may lead to the misuse of materials by maintenance personnel to repair the problem, and defects reappear and expand in the short term, resulting in a waste of resources and manpower. There are few studies on the use of YOLOv8n for UAVs in the field of wind turbine defect detection, and there is still room for improvement of this model in terms of real-time performance, accuracy, and detection speed. To solve these problems, we propose a lightweight, real-time and efficient YOLOv8n wind turbine defect detection model.

The main contributions of this paper are as follows:

(1) The backbone network uses our designed PC-EMA module instead of the normal C2f layer, which improves the multiscale feature extraction capability and the accuracy of the network to maximize the extraction of effective feature information with less computing power. This approach also enhances the model's ability to detect small targets.
(2) The neck network reduces the number of model parameters and the computational complexity while ensuring detection accuracy by integrating the GSConv and VoVGSCSP modules.
(3) The decoupled head of YOLOv8n separates classification from regression, which makes the network more expressive but makes the detection head too complex. For this reason, we design the PConv head to merge the two branches so that the classification and regression tasks are implemented together in a 1×1 convolution. Moreover, we use the PConv convolution, which can extract spatial features more efficiently, to replace the ordinary convolution in the original detection head to maximize the accuracy while ensuring a lightweight image.
(4) Since the horizontal-to-vertical ratio described by the CIoU is relative and has some ambiguity, we introduce the WIoUv3 as the bounding box regression loss. The WIoUv3 can more accurately evaluate the performance of the target detection algorithms when dealing with targets of different sizes by taking into account the difference in the size of the target box, and it can improve the detection accuracy, especially when more than one object part is involved.
(5) We collected defect images of each area of the wind turbine using drones on-site, and created a comprehensive dataset of surface defects for wind turbines.

Related algorithms

Review of YOLOv8

The YOLOv8n model consists mainly of a backbone, neck, and head network. YOLOv8n adopts the architecture of CSPDarknet-53 in the backbone network but removes one layer of ConvModule, reducing the number of convolutional kernels by 1024. The neck part of YOLOv8 uses the idea of a path aggregation network (PAN[27]). Unlike the Feature Pyramid Network (FPN) that operates in a top-down manner, the Path Aggregation Network (PAN) introduces a bottom-up pathway in addition to the traditional top-down pathway, allowing for easier propagation of low-level information to the higher-level top. This enables the model to capture fine-grained details and contextual information of the target at different scales, and further process and fuse features from the backbone network. Finally, the head is the predictive part of the network and is designed using a decoupled-head structure. The decoupled-head is used to process classifications and detections separately, and a different loss function is used for each task; for the classification task, the Binary Cross Entropy loss (BCE loss) is used to measure the accuracy of classification; for the detection task, the distributed focal loss[28] and CIoU[29] are used. This structure can improve the flexibility and detection accuracy of the model to some extent.

EMA attention mechanism

The EMA[30] attention mechanism is a new efficient multiscale attention mechanism that operates without the need for a dimensionality reduction, which is designed with the idea of efficiently capturing multiscale features while maintaining high efficiency, thus improving the accuracy of the model. In Fig. 1, "g" indicates the divided groups, "X Avg Pool" represents the 1D horizontal global pooling, and "Y Avg Pool" indicates the 1D vertical global pooling.

The EMA attention mechanism not only encodes the interchannel information to adjust the importance of the different channels but also retains the precise spatial structure information in the channels. After the convolutional layers, the EMA employs multiscale branches responsible for extracting features at different scales. By introducing different sized convolutional kernels or pooling operations in the network, feature representations at different scales can be obtained at different layers. These features of different scales are fed to their respective attention branches for processing. Each attention branch uses its own attention mechanism to dynamically adjust the importance of the features based on specific contextual information. In this way, features at different scales can be assigned different weights according to their relative importance, thus better capturing the small target in the image.

The EMA has the following advantages:

(1) The channel information is considered, and the spatial information is preserved.
(2) It better preserves the channel information and reduces the computational overhead, with fewer parameters and fewer computations.
(3) When facing targets at different scales, the information can be processed dynamically to improve the recognition of small targets in complex environments, thus improving the performance of the computer vision tasks.
(4) It is flexible enough to be simply plugged into the core modules of a lightweight network without retraining the entire model.

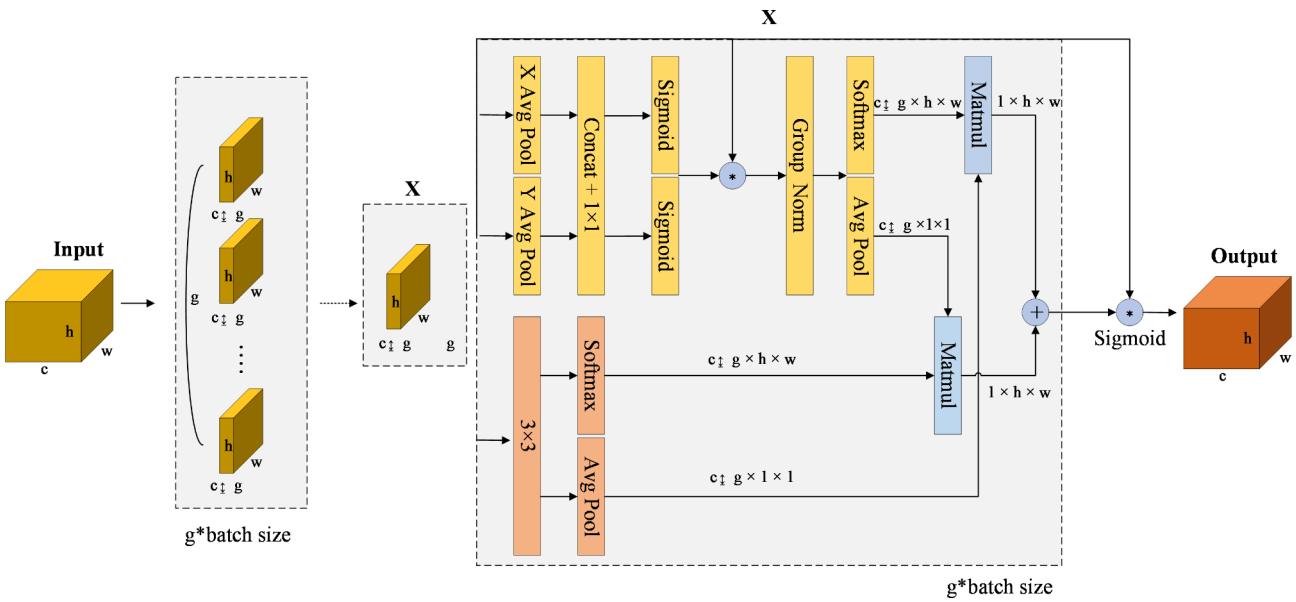

Fig. 1. EMA model structure.

In the WTB detection task, the introduction of the EMA mechanism can enhance the feature learning power of the model for defects, thus improving the detection accuracy and robustness. Moreover, the EMA is efficient and lightweight and has less impact on the number of parameters in our lightweight model. In wind turbine defect detection, there may be many different types of defects with different features, such as shape, size, and color. At this time, EMA attention may assist the model in extracting important information from images at various scales, hence enhancing the detection performance.

Improved network modeling

A lightweight YOLOv8 wind turbine defect detection algorithm is proposed whose structure is shown in Fig. 2. The model complexity of this algorithm is greatly reduced. In this paper, we first improve the basic block in YOLOv8n by integrating the PConv and EMA modules and then design a PC-EMA module for multiscale feature extraction. We employ the GSConv and VoVGSCSP in the neck network to minimize the inference time and balance the model's accuracy and complexity. In the detection head part, the classification and detection heads are combined, and the PConv head detection head is designed using PConv[31], significantly reducing the model's parameter count and increasing the model's detection speed. Finally, the bounding box regression loss is Wise-IoU (WIoU)v3, which forces the model to concentrate more on the common quality samples, enhancing the model's accuracy and localization capabilities.

PC-EMA model

Defective targets on the surface of wind turbines vary in shape, size, and color. Often, there are multiple scales of defective targets present in the same image, which can easily lead to missed detections, especially for relatively small targets. The YOLOv8 backbone network consists of a large number of 3×3 standard convolutional and

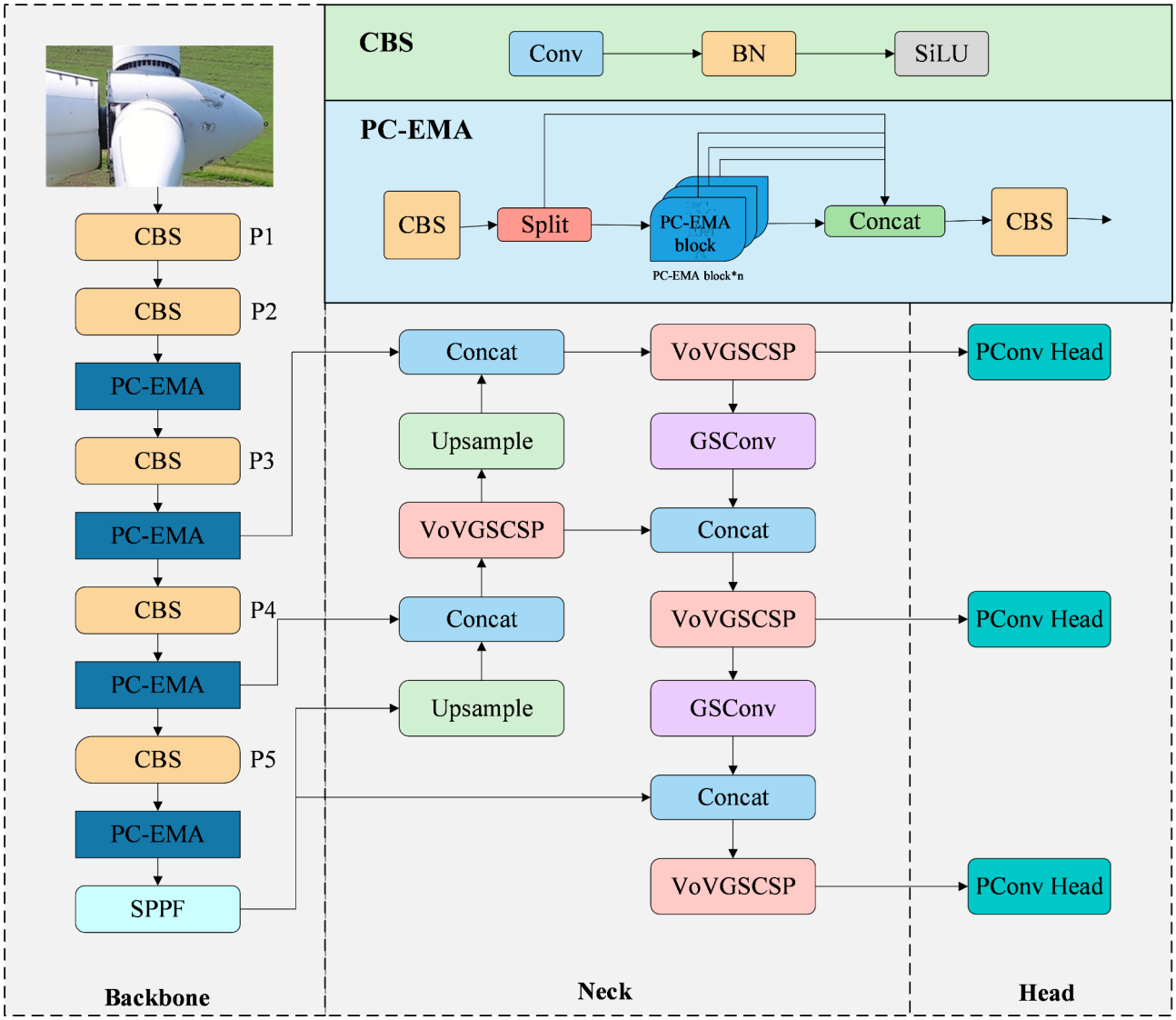

Fig. 2. Structure of the improved YOLOv8n network.

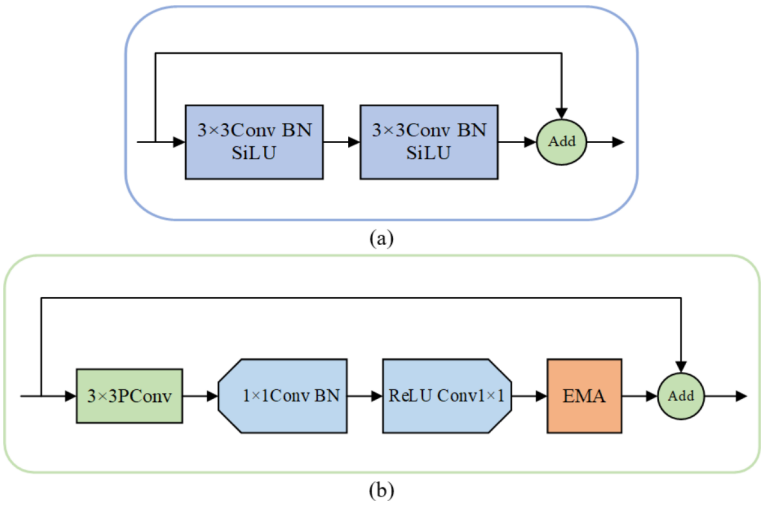

Fig. 3. Convolution structure: (a) basic block. (b) PC-EMA block.

C2f layer tandem modules for feature extraction. This simple module stack is not able to effectively distinguish between the target and background. In the bottleneck part of YOLOv8 shown in Fig. 3(a), there is an excessive use of 3 × 3 standard convolutions. As a result, the network parameters increase significantly since the bottleneck structure is stacked multiple times in the network. This architecture limitation makes it challenging for YOLOv8 to handle the diverse characteristics of defective targets in wind turbine images. Consequently, it may fail to accurately detect small targets or differentiate them from the background.

We design a PC-EMA block based on the PConv and EMA attention mechanisms to address the aforementioned issues. This module increases the accuracy of the network and preserves more data about the small targets. It also enhances the extraction of the multiscale characteristics. The structure is shown in Fig. 3(b). The PC-EMA block consists of a 3 × 3 PConv, two 1 × 1 Convs and an EMA attention module. PConv applies convolution for spatial feature extractions on one part and keeps it unchanged on the other channels. This operation can extract spatial features efficiently and reduce redundant computations. After that, to make full use of all the channel information, two 1 × 1 Conv layers, which are shown together as inverted residual blocks, are introduced. The design idea of the inverted residual blocks is to improve the expressive power of the network while keeping the computational complexity low. By expanding the dimensionality of the feature map, the network can better capture and represent the diversity information of the input data, thus improving the performance of the model. The inverted residual block is a design module that optimizes computational efficiency, allowing the model to perform better after a hardware deployment. Meanwhile, to improve the model's detection performance for tiny targets in complex backgrounds, we introduce an efficient multiscale EMA attention mechanism without reduction, which has a small parameter count and can process feature information dynamically. In addition, we also use Batch Normalization (BN)[32] between two 1 × 1 Convs. One of the advantages of Batch Normalization (BN) is that when the model hardware is deployed, the BN layer is treated as a simple scale layer by TensorRT, and through optimization, it is merged with the Conv layer in front of it, thus improving the inference speed of the model. Therefore, using BN through TensorRT processing can improve the speed of the model without affecting the normal inference of the model. For the activation layer, we chose ReLU[33], considering the balance between runtime and effectiveness. In designing the PC-EMA block, we opted for fewer BN and ReLU layers compared to the Basic block in the original YOLOv8n. This decision was made to avoid excessive use of BN and ReLU layers, which has the potential to limit feature diversity and hinder overall computational speed, thereby potentially impacting performance.

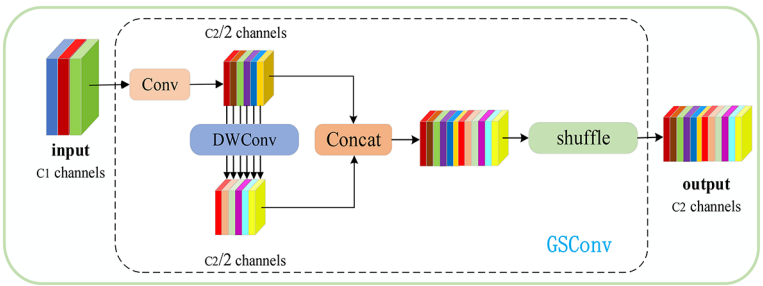

Fig. 4. Network structure of GSConv.

Neck network based on the GSConv and VoVGSCSP

Wind turbine detection scenarios usually involve complex environments, which greatly interferes with the target capture in images. Moreover, if the network employs depth-separable convolution in the feature extraction stage, a large amount of channel information may be lost. To address these issues, we introduce a lightweight convolution called GSConv[34] in the neck network. It uses a shuffle strategy and combines normal convolution and depth-separable convolution. This mechanism allows the convolution operation to be flexible enough to adapt to different image features. As shown in Fig. 4 below, the input feature map is first passed through a convolutional layer and then deeply convolved using DWConv to stitch the results of the two convolutions, one is a SC (channel-dense convolution) and the other is a DSC (channel-sparse convolution). Finally, a shuffling operation is carried out to rearrange the feature channels and improve the flow of information between the features so that the SC information is fully mixed into the DSC output. In this way, richer channel information can be extracted while keeping the network lightweight to cope with target detection tasks in complex environments.

Typically, the time complexity is usually defined by floating point operations (FLOPs). Therefore, the time complexities of SC, DSC, and GSConv are shown in Eqs. (1)-(3), respectively:

$$Times_{SC} = O(W \times H \times K_1 \times K_2 \times C_1 \times C_2) \tag{1}$$

$$Times_{DSC} = O(W \times H \times K_1 \times K_2 \times 1 \times C_2) \tag{2}$$

$$Times_{GSConv} = O\left[W \times H \times K_1 \times K_2 \times \frac{C_2}{2} \times (C_1 + 1)\right] \tag{3}$$

Where W represents the width of the output feature map, H represents the height, and K1 and K2 represent the size of the convolution kernel. C1 is the number of channels per convolution kernel and the number of channels of the input feature map, and C2 represents the number of channels of the output feature map. According to the above equation, the time complexity of GSConv is approximately 50% of that of SC (0.5 + 0.5C1; the larger the value of C1 is, the closer the ratio is to 50%), and the time complexity of DSC is also greater than that of GSConv (C2 > C1). The GSConv module in wind turbine defect detection improves the localization accuracy and defect detector performance by allowing the network to concentrate more on the defect location and extract more precise information through adaptive convolutions.

While GSConv lowers the computational complexity, a new module is still required to maintain accuracy and shorten the inference time. Therefore, the network module VoVGSCSP—which is related to VoVNet and CSPNet—is created using the GSConv one-shot aggregation technique, as illustrated in Fig. 5. Its function is similar to that of C2f, but it requires less computing power. This study presents a method that substitutes the VoVGSCSP module for the C2f module of the neck network, and GSConv for ordinary convolutions in the feature fusion network. This improves the expressiveness of the model in the face of complex graph structures and resolves the issue of decreasing model localization ability as a result of network deepening. Introducing the GSConv and VoVGSCSP modules into our necking network reduces the memory requirements of the model. This is important for resource-constrained embedded devices or edge computing environments to make the model more adaptable to hardware constraints.

PConv head

In YOLOv8, we adopt a decoupled head structure, as shown in Fig. 6. This structure first inputs three effective feature layers, P3, P4, and P5, into the head network and then separates the classification and regression tasks by two parallel 3 × 3 convolutional layers, thus realizing the independent execution of the classification and regression tasks. Next, the classification, localization and confidence detection tasks are again processed through 1 × 1 convolutional layers. The target's class may be predicted using the classification header, and its position and size can be predicted using the regression header. The primary goal of this decoupled structure is to resolve conflicts between the classification and regression tasks while also somewhat increasing the detection accuracy. However, decoupling the detection header while accelerating convergence simultaneously, increases the complexity of the operation and drastically increases the network parameters due to the stacking of channels resulting from the use of multiple 3 × 3 convolutions in series.

In practical wind turbine defect detections, UAV devices with limited computational resources are often used. However, the decoupled head of YOLOv8 has a complex structure and typically utilizes two independent branches for classification and localization. This design significantly increases model parameters, which contradicts our original goal of developing a lightweight real-time object detection model. To address this issue, we have made improvements to the decoupled head by introducing the PConv Head, as illustrated in Fig. 7. Instead of using two separate 3 × 3 convolutions, we replaced them with Pconv convolution, which merges the

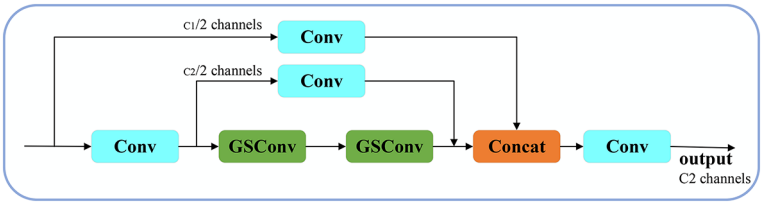

Fig. 5. Network structure of the VoVGSCSP.

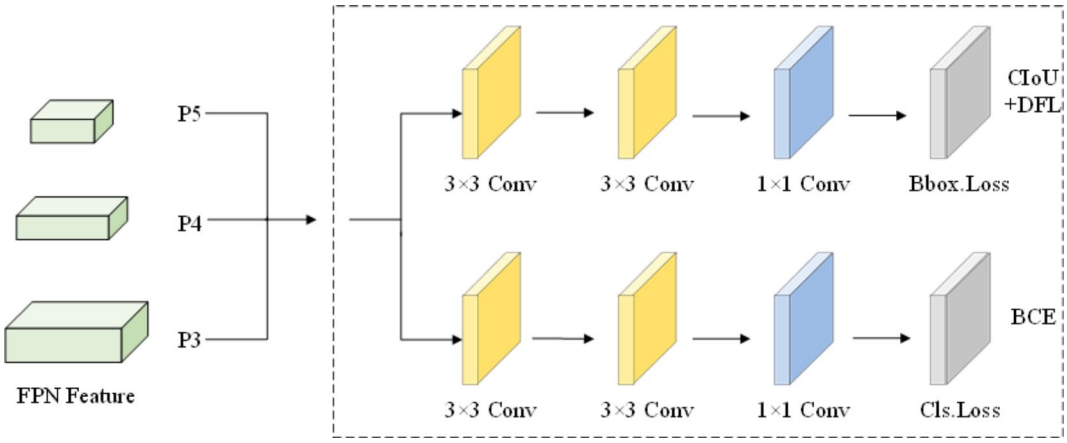

Fig. 6. YOLOv8 detection head structure diagram.

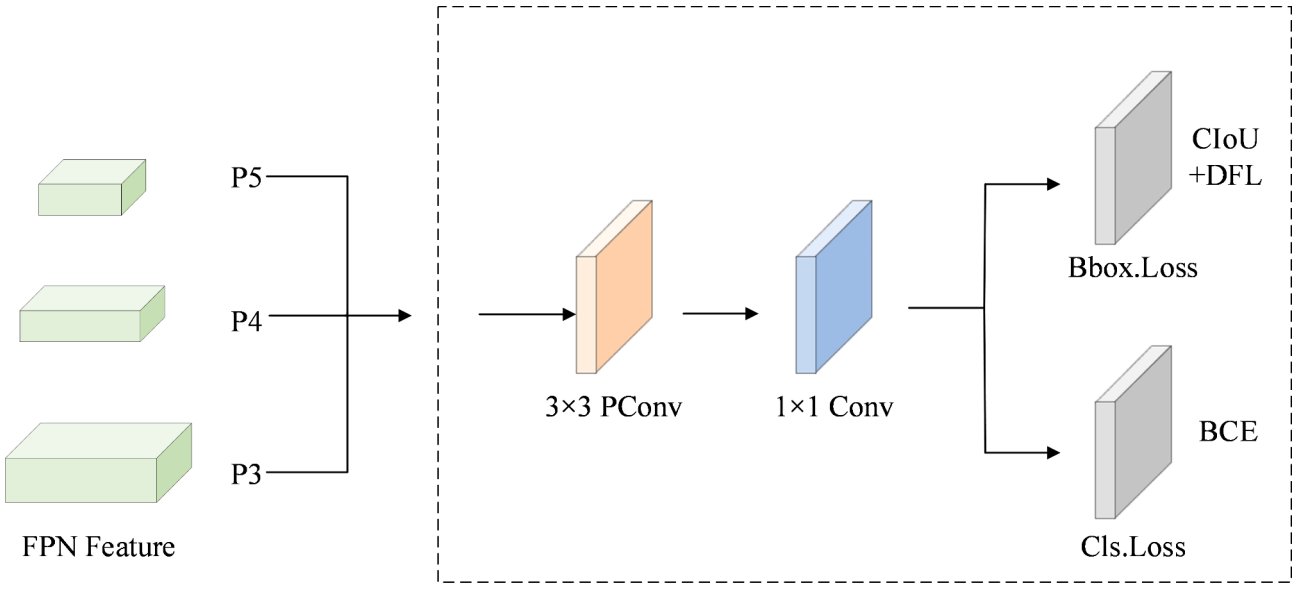

Fig. 7. The improved PConv head detection structure.

two branches. While this operation slightly affects the network's detection accuracy, we leverage the redundancy of feature maps in PConv to maximize channel information preservation while minimizing GFLOPs and memory accesses. This ensures the accuracy of the model. Furthermore, we employ 1×1 convolutions for dimensionality reduction to further reduce the computational load of the model. The PConv Head is designed to include a set of convolutional layers, with some dedicated to classification and others to regression. This design enhances computational efficiency and reduces the number of model parameters by sharing the feature extraction process. Compared to the original detection head, our modified detection head significantly reduces the number of network parameters and GFLOPs, while only experiencing a minor loss in accuracy.

Experiment

Datasets

In applied research, there are few publicly accessible datasets that can be used for deep learning training on wind turbine surface flaw identification. The images used in this paper consist of the images of wind turbine surface defects taken by UAVs from wind farms in Northwest China, and images of wind turbine surface defects taken by public UAVs from the Roboflow website. There are a total of 1437 images. After image enhancement and other preprocessing operations, the categories of dirt, leakage, erosion, cracks, and paint off were expanded to 775, 717, 704, 748, and 789 images respectively. The UAV photographed the surface of the wind turbine from multiple angles, as shown in Fig. 8, including the wind turbine hub, tower, wind vane, yaw system, blades, and other structures. The resolution of the image is 1280×1080, which basically meets the target detection requirements. In some experiments, we demonstrated some defective target images, as shown in Fig. 9. To simulate photos taken during bad weather, we introduced an image blurring algorithm in the dataset preprocessing operation.

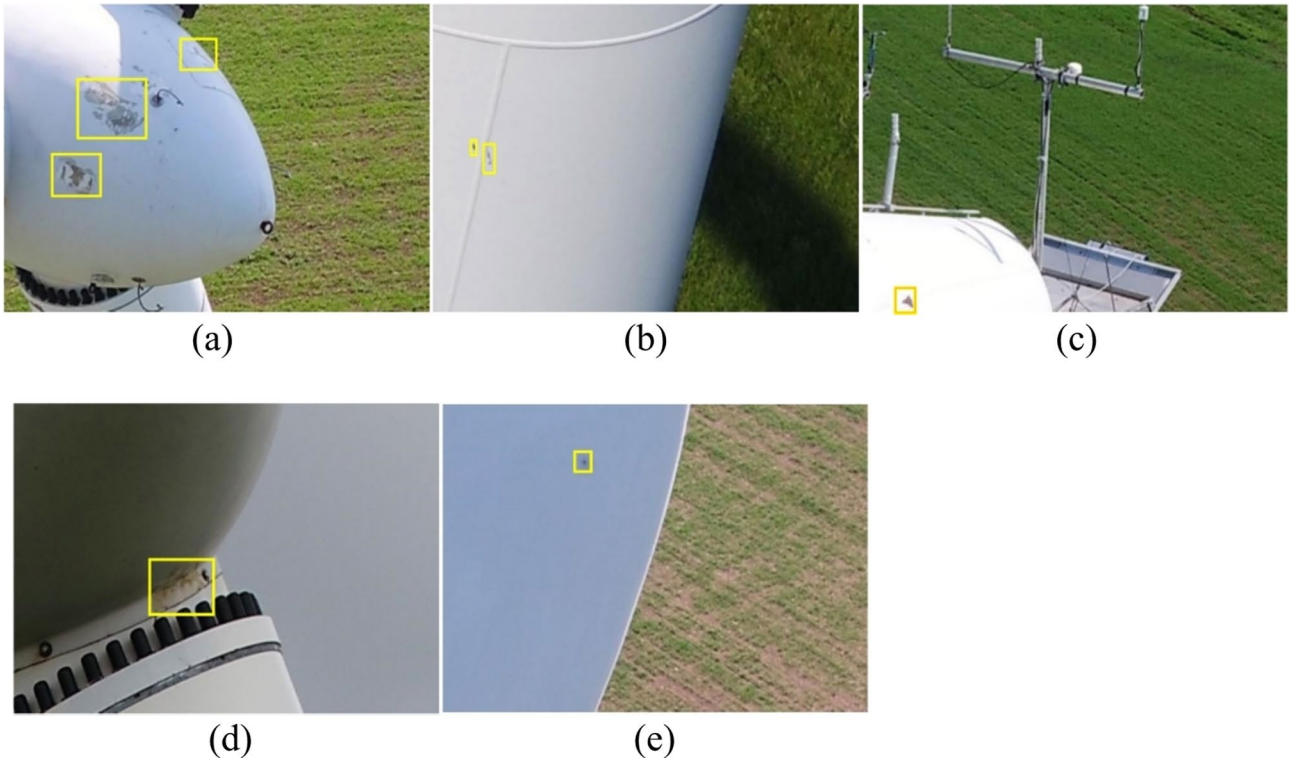

Fig. 8. Target images of defects at each location of the wind turbine: (a) wheel, (b) tower, (c) weathercock, (d) yaw system, and (e) blade.

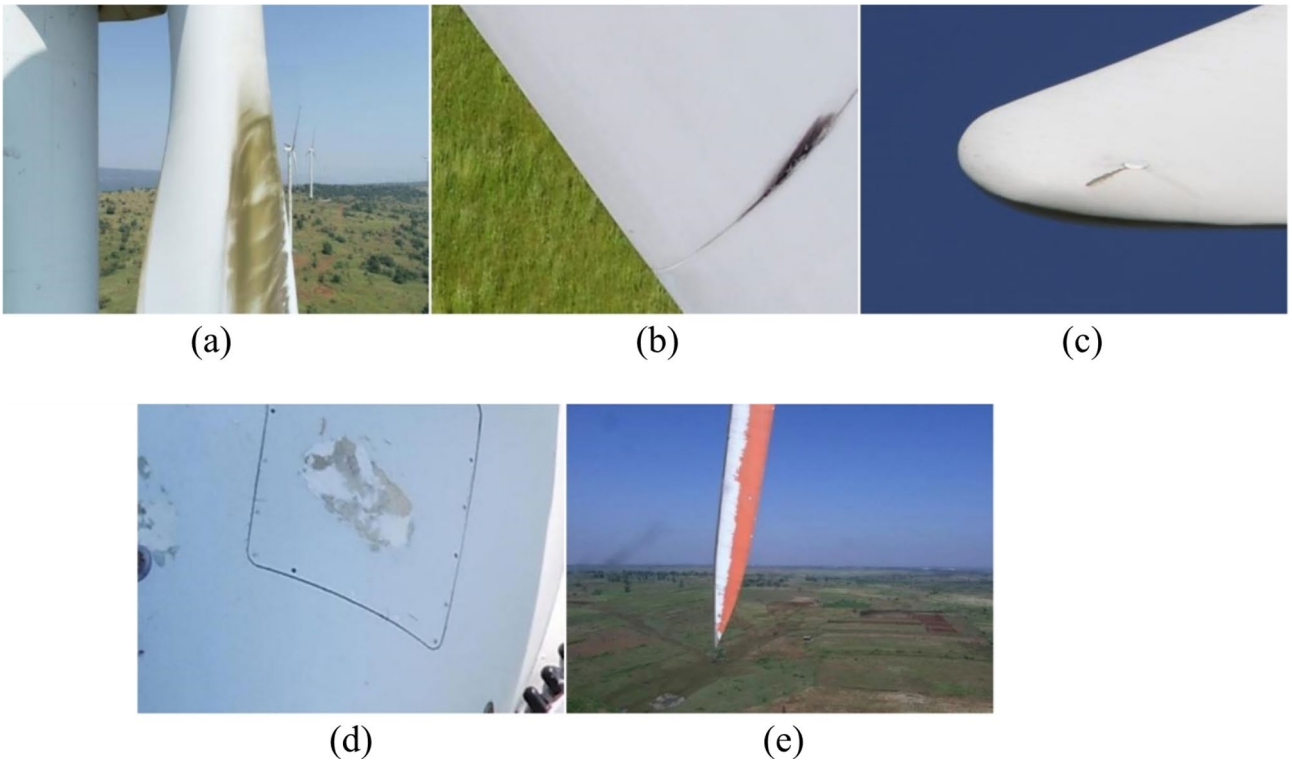

Fig. 9. Types of defects in wind turbines: (a) dirt, (b) leakage, (c) crack, (d) erosion, and (e) paint off.

Additionally, to simulate photos taken under dim light, we performed brightness and contrast transformations. All these operations are designed to improve the model's target detection ability for different environments. The 3733 obtained images are screened and labeled, and the dataset is divided into training, validation, and test sets at a ratio of 8:1:1, with 2986 for the training set, 374 for the test set, and 373 for the validation set.

Evaluation metric

We use the precision (P), recall (R), model size, frame rate (FPS), mean average precision (mAP), and mean average precision (IoU) = 0.5 (mAP@0.5) to measure the detection effectiveness. The number of predicted faults (TP), the number of misdetected defects (FP), and the number of nondetected defects (FN) are represented. P represents the accuracy of the improved model, or the percentage of defects that are properly classified. R stands for the recall of the model and refers to the ratio of the number of correct samples predicted to the number of correct samples. The P-R curve describes how a category varies in accuracy and recall at different thresholds, with each point on the curve representing the accuracy and recall at a different threshold. The AP value of the category is represented by the area contained in that curve. The AP values for each category are calculated and averaged to obtain the mAP. mAP@0.5 is the mAP at the intersection over union (IoU) threshold of 0.5. The formula for each metric is shown in Eqs. (4)-(7):

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$R = \frac{TP}{TP + FN} \tag{5}$$

$$P_{AP} = \int_0^1 P(R)\mathrm{d}_R \tag{6}$$

$$P_{mAP} = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{7}$$

Additionally, the model size and detection speed metrics are added to evaluate the performance of the lightweight models more accurately.

Experimental results

Experiments are designed using the WTB dataset and training parameters with a batch size of 16, a learning rate of 0.01, and an epoch number of 200. The improved model is compared with previous defect detection methods. The experimental data are shown in Table 1.

By focusing on more useful feature information and altering the original convolution approach, the PC-EMA module increases the accuracy of defect detection, as demonstrated by the comparison of Experiment B and Experiment C. The mAP@0.5 increased by 3.68%. Slim-neck is an improvement of the neck network by GSConv and VoVGSCSP, which are compared with Experiments B and D. GSConv and VoVGSCSP fusion increases the adaptivity of the neural network and improves the fusion of the target feature maps. The mAP@0.5 improved by 1.96%. Comparing Experiments B and E, our improved PConv head detection head has a small accuracy loss but drastically reduces the model's computation and size, where the GFLOPs are reduced by 3.2G and the model size is reduced by 1.13 M. The addition of the WIoU loss increases the detection accuracy, as demonstrated in Experiments B and F. The comparison of Experiments A and G shows that YOLOv8n incorporates the PC-EMA module and Slim-neck to maximize the model accuracy, but the lightweight effect is not obvious enough to

Method		mAP@0.5(%)	Model size(M)	GFLOPs(G)
A	Ours	95.13	4.01	3.9
B	YOLOv8n	90.06	5.94	8.9
C	+ PC-EMA	93.74	5.28	7.9
D	+ Slim-ncek	92.02	5.59	8.0
E	+ PConv Head	89.59	4.81	5.7
F	+ Wise-IOU	90.67	5.94	8.9
G	+ PC-EMA	95.48	4.93	7.1
	+ Slim-ncek			
H	+ PC-EMA	93.44	4.15	4.7
	+ PConv Head			
I	+ Slim-ncek	91.78	4.46	4.9
	+ PConv Head			
J	+ PC-EMA	94.86	4.01	3.9
	+ Slim-ncek			
	+ PConv Head			

Table 1. Comparisons between the models.

achieve the lightweight level we require. Comparing Experiments B and H, the model has a better lightweighting level, but the accuracy is 1.39% lower than that of our model. Our model improves the backbone, neck, head networks, and the loss function so that the detection speed and accuracy of the model meets real detection requirements. These designs enable the model to extract useful cross-spatial features while capturing useful information in complex real-world scenarios. In terms of model speed, these designs significantly reduce the model's latency. As a result, our model exhibits a higher level of detection accuracy and is more lightweight in various defect detection tasks.

Using the same dataset, Table 2 illustrates how our model's detection efficiency is higher. There is insufficient feature extraction from SSD, insufficiently high detection accuracy, and insufficiently strong model training generalization. Although the two-stage detection approach, which involves two steps of detection, results in too few model frames to accomplish real-time detection, the detection effect of Faster R-CNN in the table is reasonably strong. The YOLO series of algorithms offers great detection speeds compared to the two standard models mentioned above. Our algorithm detects 12 FPSs more quickly than YOLOv8n. Our model is 1.93 M smaller than YOLOv8n in terms of size. Compared with those of the other five models, the mAP of our model is improved by 5.07%.

We use a comparison test that is based on the UAV aerial photography test set to more thoroughly validate the improved model detection performance. When dirt and crack defects overlap in the first row of photos in Fig. 10, our model can identify the fracture defects concealed by dirt, proving the validity of our model. The Faster-RCNN model is able to recognize erosion targets in the second row of photos in Fig. 10 because these items typically have large forms and noticeable color features. Due to low light, in the dim environment shown in the third row of Fig. 10, both SSD and Faster R-CNN misidentify the leakage as a crack. In contrast, our model has great stability and makes no misdetections. In the fourth and fifth rows of Fig. 10, the first two models failed to detect defects due to tiny targets, and the original YOLOv8n model detected only the larger defects in the figure; however, our model detected all the defects, demonstrating the ability of our model to detect tiny targets. In addition, our model improves global feature extraction, retains more feature information about small targets after multiscale feature fusion, and outperforms the other three models in terms of detection accuracy. The model possesses a high degree of durability and can effectively detect surface problems on wind turbines in intricate situations.

We conducted a comparison experiment on YOLOv8 utilizing WIoUv3 and a few standard loss functions while maintaining consistency in other training situations to validate the efficacy of WIoUv3 in wind turbine fault diagnoses. Table 3 shows that the model performs best in terms of detection when the WIoUv3 is used as the bounding box regression loss. Furthermore, the mAP@0.5 of the WIoUv3 model is 0.61% greater than that of the CIoU model, demonstrating the value of implementing the WIoUv3 model.

Tests with the Jetson Nano were performed to confirm that the model was lightweight. The Jetson Nano is an affordable, low-power, and powerful AI embedded development board with a quad-core ARM CPU, a 128-core GPU, and 4 GB of LPDDR4 storage.

TensorRT is an inference engine released by NVIDIA that is optimized and accelerated for NVIDIA series hardware to achieve maximum utilization of the GPU resources and improve inference performance. To make hardware detection faster, we import TensorRT into Python for the most efficient utilization of the GPU and then export the trained model best.PT to the best.onnx format for a lightweight deployment. As shown in Table 4, with an image resolution of 640 × 640, the inference time of this improved model is 28% faster than that of the original model, the detection speed greatly improves compared to that of various YOLO algorithms and is 4 FPS faster compared to that of YOLO8n, which is accelerated by TensorRT to 11 FPS faster than that of the original model. As the input resolution decreases, the inference speed of the model gradually approaches, and the model frame rate also increases. However, the improved model is the least affected by the resolution, making it the fastest. The improved model shows an 8 FPS improvement compared to the original model. After deployment, the reliability and robustness of the model's lightweight design were verified, fully utilizing the device's performance.

Conclusion

Our proposal involves utilizing a lightweight network in conjunction with a YOLOv8n detection model to address the wind turbine fault detection problem in the real industry. The following features are the key ways in which the model enhances the YOLOv8n network structure. The developed PC-EMA module is incorporated into the backbone feature extraction network. It can efficiently capture multiscale feature information and improve the feature expression ability of the channel attention mechanism. To reduce the number of computations, GSConv and VoVGSCSP are introduced in the feature fusion section. The initial complex detection head is replaced in

Method	mAP@0.5(%)	Precision(%)	Recall(%)	Model size (M)	Frame rate(fps)
SSD	74.58	77.61	72.38	92.86	48
Faster-RCNN	82.11	84.27	80.42	114.37	5
YOLOv5	85.29	86.25	84.06	5.01	64
YOLOv7-tiny	87.35	88.34	86.89	14.32	59
YOLOv8n	90.06	91.39	89.95	5.94	71
Ours	95.13	95.71	91.73	4.01	83

Table 2. Performance comparison with classical models.

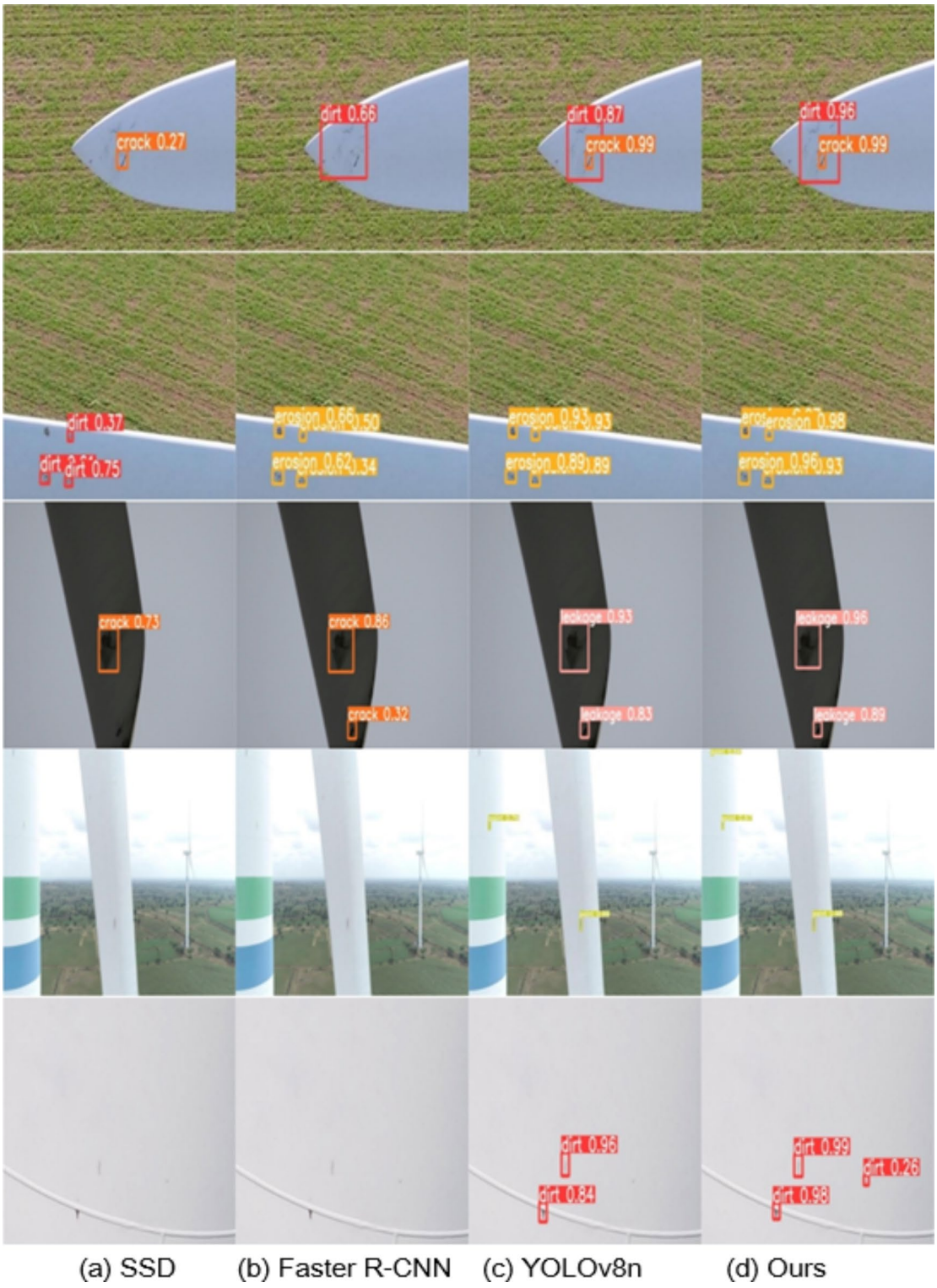

Fig. 10. Results of different detection models.

the detection head section with a lightweight, low-latency PConv head, which greatly decreases the number of model parameters and FLOPs, and increases the frame rate of the model. The WIoUv3 is added to the loss function to increase the recall rate for small targets, enhance the regression accuracy of the Bbox, and improve the localization of the detection frames.

The performance indices of the improved model, such as the mAP, frame rate, and model size, are improved to a certain extent for detecting defective targets, compared with those of the other models. After the deployment of Jetson Nano, it is proven that our model has certain application value in industrial deployments. Additionally,

Loss function	Box_Precision(%)	Recall(%)	mAP0.5 (%)
CIoU	92.67	89.57	90.06
SIoU	89.53	90.12	89.59
EIoU	92.62	90.58	90.14
WIoUv1	91.66	90.11	90.23
WIoUv2	92.39	90.18	90.32
WIoUv3	93.40	90.35	90.67

Table 3. Comparison of the detection results for different loss functions introduced by YOLOv8n.

Image size	Models	Preprocessing (ms)	Inference(ms)	FPS	TensorRT FPS
640×640	YOLOv5	7.5	128.2	8	17
	YOLOv7-tiny	3.4	90.7	11	22
	YOLOv8n	2.8	73.8	13	24
	Ours	1.7	53.1	17	35
512×512	YOLOv5	3.6	81.4	10	20
	YOLOv7-tiny	2.8	65.8	12	27
	YOLOv8n	2.5	52.1	14	29
	Ours	1.7	44.7	19	37
416×416	YOLOv5	1.3	71.9	12	24
	YOLOv7-tiny	1.3	67.3	14	27
	YOLOv8n	1.2	59.3	16	30
	Ours	1.0	41.8	19	37

Table 4. Comparison of the speeds for Jetson Nano.

our model addresses the problem that a large amount of network computations are not easily embedded in edge devices during the process of detecting defects.

Data availability

The datasets used and analyzed during the current study are available from the corresponding author upon request.

Received: 5 March 2024; Accepted: 30 September 2024
Published online: 19 October 2024

References

1. Chen, X., Yan, R. & Liu, Y. Wind turbine condition monitoring and fault diagnosis in China. *IEEE Instrum. Meas. Mag.* **19**, 22–28 (2016).
2. AFatehi, M. et al. Aerodynamic performance improvement of wind turbine blade by cavity shape opti-mization. *Renew. Energy*. **132**, 773–785 (2019).
3. Ribrant, J. & Bertling, L. Survey of failures in wind power systems with focus on Swedish wind pow-er plants during 1997–2005. In *IEEE Power Engineering Society General Meeting* 1–8 (2007).
4. García Márquez, F. P., Tobias, A. M., Pérez, P., Papaelias, M. & J. M. & Condition monitoring of w-ind turbines: Techniques and methods. *Renew. Energy*. **46**, 169–178 (2012).
5. Zhang, D. et al. A data-driven design for fault detection of wind turbines using random forests and XGboost. *IEEE Access.* **6**, 21020–21031 (2018).
6. Castellani, F., Astolfi, D. & Natili, F. SCADA data analysis methods for diagnosis of electrical faults to wind turbine generators. *Appl. Sci.* **11**(8) (2021).
7. Yang, Z., Zhang, H., Guan, P. & Dong, Y. Test of offshore wind generator pile foundation based on distributed Brillouin optical fiber sensing. in *Optics Frontiers Online 2020: Distributed Optical Fiber Sensing Technology*, Vol. 11607. 116070I (2021).
8. Shuang, F. et al. AFE-RCNN: Adaptive feature enhancement RCNN for 3D object detection. *Remote Sens.* **14**(5), 1176 (2022).
9. Ren, S., He, K., Girshick, R., Sun, J. & Faster, R-C-N-N. Towards real-time object detection with re-gion proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2017).
10. He, K., Gkioxari, G., Dollár, P., Girshick, R. & Mask, R-C-N-N. In. *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988 (2017).
11. Liu, W. *et al.* SSD: Single Shot MultiBox Detector. In vol. 9905, 21–37 (2016).
12. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788 (2016).
13. Redmon, J. & Farhadi, A. YOLO9000: Better, Faster, Stronger. Preprint at https://doi.org/10.48550/arXiv.1612.08242 (2016).
14. Redmon, J. & Farhadi, A. YOLOv3: An Incremental Improvement. Preprint at http://arxiv.org/abs/1804.02767 (2018).
15. Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. YOLOv4: optimal speed and accuracy of object D-etection. Preprint at https://doi.org/10.48550/arXiv.2004.10934 (2020).
16. Li, C. *et al.* yolov6: A single-stage object detection framework for industrial applications. Prepri-nt at (2022). https://doi.org/10.48550/arXiv.2209.02976

17. Wang, C., Bochkovskiy, A. & Liao, H. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Rec-ognition (CVPR)*, 7464–7475 (2023).
18. Takeda, N. Characterization of microscopic damage in composite laminates and real-time monitoring by embedded optical fiber sensors. *Int. J. Fatigue*. **24**, 281–289 (2002).
19. Wang, L. & Zhang, Z. Automatic detection of wind turbine blade surface cracks based on UAV-T-aken images. *IEEE Trans. Industr. Electron.* **64**, 7293–7303 (2017).
20. Yu, J. et al. An infrared image stitching method for wind turbine blade using UAV flight data and U-Net. *IEEE Sens. J.* **23**, 8727–8736 (2023).
21. Moreno, S., Peña, M., Toledo, A., Treviño, R. & Ponce, H. A. New vision-based method using deep learning for damage inspection in wind turbine blades. In *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1–5 (2018).
22. Stokkeland, M., Klausen, K. & Johansen, T. A. Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 998–1007 (2015).
23. Mao, Y., Wang, S., Yu, D. & Zhao, J. Automatic image detection of multi-type surface defects on wi-nd turbine blades based on cascade deep learning network. *Intell. Data Anal.* **25**, 463–482 (2021).
24. Qiu, Z., Wang, S., Zeng, Z. & Yu D. Automatic visual defects inspection of wind turbine blades via YOLO-based small object detection approach. *J. Electron. Imaging* **28**(4), 043023 (2019).
25. 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. *Renew. Energy*. **163**, 386–397 (2021).
26. Zhang, R., Wen, C. & SOD-YOLO: A small target defect detection algorithm for wind turbine blades based on improved YOLOv5. *Adv. Theory Simul.*. **5**, 2100631 (2022).
27. Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. Path Aggregation Network for Instance Segmentation. Preprint at http://arxiv.org/abs/1803.01534 (2018).
28. Li, X. *et al.* Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. Preprint at https://doi.org/10.48550/arXiv.2006.04388 (2020).
29. Zheng, Z. *et al.* Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. Preprint at https://doi.org/10.48550/arXiv.2005.03572 (2021).
30. Ouyang, D. *et al.* Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (2023).
31. Chen, J. et al. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. Preprint at https://doi.org/10.48550/arXiv.2303.03667 (2023).
32. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Preprint at https://doi.org/10.48550/arXiv.1502.03167 (2015).
33. Glorot, X., Bordes, A. & Bengio, Y. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 315–323 (2011).
34. Li, H. *et al.* Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. Preprint https://doi.org/10.48550/arXiv.2206.02424 (2022).

Author contributions

G.W. is responsible for the conceptual design of the project. Z.W. is responsible for the construction of the algorithm and the writing of the paper. Y.Z. is responsible for the collection and labeling of experimental data. X.W. experimental data testing preliminary analysis of experimental results. H.L. is responsible for the organization of the paper and the final review of the article. Y.S. is responsible for the validation of the article data.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.W.

Reprints and permissions information is available at www.nature.com/reprints.

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