**ORIGINAL RESEARCH**

# Wind turbine blade structural state evaluation by hybrid object detector relying on deep learning models

Dipu Sarkar[1] · Sravan Kumar Gunturi[2]

## Abstract

Surveillance drones are remarkable devices for monitoring, as they have strong spatial and remote sensing capabilities. The prompt detection of peripheral damage to the blades of wind turbines is necessary to reduce downtime and prevent the potential failure of wind farms. Computer vision breakthroughs with deep learning have developed and been refined over time, mainly using convolution neural networks. From this perspective, we suggest a deep learning model for monitoring and diagnosing the blade health of wind turbines based on images captured by surveillance drones. The main limitations of standard monitoring devices are their poor detection accuracy and lack of real-time performance, making it complex to obtain the attributes of blades from aerial images. Based on the foregoing, this study introduces a method for increasing detection accuracy when carrying out operations in real time using You Only Look at Once version 3 (YOLOv3). We train and evaluate three deep learning models on the wind turbine image dataset. We find that many aerial images are unclear because of blurred motion. As avoiding such low-resolution images for training can affect accuracy, we use a super-resolution convolution neural network to reconstruct a blurred picture as a high-resolution one. The computational results demonstrate that YOLOv3 outperforms traditional models in terms of both accuracy and handling time.

**Keywords** Wind turbine · Surveillance drones · Renewable energy · YOLOv3

## 1 Introduction

Wind energy has become a preferred method of generating renewable energy, since 1 MW of wind energy compensates for roughly 2600 tons of annual $CO_2$ emission (Al-Khudairi and Ghasemnejad 2015). A Wind Turbine (WT) comprises rotary blades, core, gear unit, generator unit, tower, and base structures. Sufficient wind forces the WT blades to spin, and then the generator produces electricity from that motion. Blades are now commonly made of glass fiber-hardened composites, which offer enhanced resilience, increased resistance to corrosion, and lower weight, as well as enabling larger dimensions for greater harvesting of wind energy (Light-Marquez et al. 2011). WT blades are subject to various hazards over their 20-year lifecycle, including fatigue-induced deterioration from shifting loads, heavy wind, rain, thunderstorms, and bird strikes (Mandell et al. 2008). Since blades account for 15–20% of overall system cost, their damage may result in significant capital loss, unplanned downtime, and potential hazards. For this reason, greater focus has been placed on monitoring the condition of WT blades. Data collected from surveillance images may be used in condition-based maintenance schemes to effectively manage survey times, eliminate potentially unnecessary replacements, and provide realistic guidance for future proposals (Pandit and Infield 2019).

Earlier studies considered monitoring the condition of WT blades by analyzing signals from installed sensors. Sutherland et al. (1994) reported two methods for monitoring blade deterioration during testing: (1) coherent optical and (2) acoustic emissions (AE). Sørensen et al. (2002) suggested using micro-bending fiber optic sensors to identify damage to blade joints. AG (2004) used AE sensors to track dynamics induced by damage to WT blades. Raišutis et al. (2008) introduced an ultrasonic, air-coupled approach with

✉ Dipu Sarkar
  dipusarkar5@rediffmail.com

1 Department of Electrical and Electronics Engineering, National Institute of Technology Nagaland, Dimapur 797103, India

2 Department of Electronics and Instrumentation Engineering, National Institute of Technology Nagaland, Dimapur 797103, India

a wave-guide for the same purpose. Jasinien et al. (2009) suggested a new method of inspection to recognize the shape and size of blade cracks that incorporates pulse-echo contact and immersion methods. Häckell and Rolfes (2013), Prowell et al. (2009) installed accelerometers to track blade condition, experimentally evaluating the efficacy of this method. More specialized sensors to monitor blade state include large-fiber composite sensing devices (Pitchford et al. 2007), Fiber-Bragg sensing devices (Lee et al. 2015), and Doppler-Vibrometer beam scanning sensors (Ozbek et al. 2013). Besides advanced sensor implementations, signal processing techniques have been introduced to improve control features. Using Fiber-Bragg sensors, (Lee et al. 2015) designed an innovative signal translation algorithm to analyze three blade moment signals. Ozbek et al. (2013) suggested and simulated a concept for signal processing Doppler radars. An alternative technique to detect damage uses microwaves (Hosoi et al. 2015; Li et al. 2016a, b), which spread in dielectric materials with minimal attenuation.

Wang et al. (2016) introduced an information-driven method for detecting potential blade breakage using SCADA results. Wang and Zhang (2017) used surveillance drones to examine the state of WT blades, enhancing efficiency in wind farms. They used Haar-like attributes to describe rift regions and trained a multi-level system to classify damage. Processing aerial images effectively, however, requires new, more powerful methods to accurately identify crack details. Deep learning in object detection has been broadly embraced with the use of efficient computing devices, such as Graphical Processing Units. Some potential deep-learning methods for recognizing objects include Faster R-CNN (Ren et al. 2015; Shihavuddin et al. 2019), Single Shot Detection (SSD) (Liu et al. 2016), and You-Only-Look-Once (YOLO) (Redmon and Farhadi 2017; Chen et al. 2019). Tu et al. (2019) described diagnostic techniques for diagnosing damage to turbine blades using the sound emitted by the blade as it passes the tower, namely the time-domain average sound signal and the short-term Fourier. Another acoustic emission approach to damage diagnostics was introduced by (Krause and Ostermann 2020), with the main benefit that it utilizes sound below 35 kHz and reduces atmospheric noise, which is often present in this frequency range. Reddy et al. (2019) suggested a system for the structural analysis of turbine blades that involves classifying and identifying damage from images collected by drone, with a CNN model and annotated image dataset developed using python packages. Yu et al. (2020) suggested a fault-recognition system for WT blades in which hierarchical features of training images are derived from Deep CNN and provided to a classifier. Current approaches to detect damage to WT blades are summarized in Table 1. Their major drawbacks include the following:

– Manual examination involves much expense and risk.

**Table 1** Traditional monitoring methods to detect blade damages

| Researcher(s) | Proposed method |
| --- | --- |
| Sørensen et al. (2002) | Fiber optic sensor |
| AG (2004) | Acoustic emission sensor |
| Raišutis et al. (2008) | Ultrasonic sensor with waveguide |
| Jasinien et al. (2009) | Pulse-echo contact and immersion |
| Häckell and Rolfes (2013) | Accelerometers |
| Prowell et al. (2009) | Accelerometers |
| Pitchford et al. (2007) | Large-fiber composite sensor |
| Lee et al. (2015) | Fiber Bragg sensor |
| Ozbek et al. (2013) | Doppler radars |
| Hosoi et al. (2015); Li et al. (2016a, 2016b) | Microwave technique |
| Wang et al. (2016) | SCADA system |
| Wang and Zhang (2017) | Object detection using Haar-like attributes |
| Tu et al. (2019) | Acoustic emission method |
| Krause and Ostermann (2020) | Acoustic emission method |
| Reddy et al. (2019) | Convolutional neural network |
| Yu et al. (2020) | Deep CNN + Classifier |

– Installing sensors on active WTs significantly increases complexity and entails major capital expenditure.
– Likely sensor malfunctions reduce the expected quality of the signal and thus the performance of the monitoring methods.
– The low-resolution images taken by drone hinder detection performance in Deep-Learning assisted methods.
– The speed and accuracy of the predictions made by existing deep convolutional neural networks are poor.

The main objective of the proposed method is to address these drawbacks by using a novel, hybrid object detector. Surveillance drones are used to take aerial photographs of turbines as part of the planned review. The resulting low-resolution images are used as input to Super-Resolution CNN, which convert them to super-resolution images. After a training process, the proposed YOLOv3 object detector is used to detect damage to WT blades. The main contributions of this work are summarized as follows:

– We suggest a novel damage detection method to identify different types of surface damage to WT blades.
– The proposed damage-detection system, based on YOLOv3, uses a three-layer network that can train damage features automatically and adaptively.
– Laplace variance is used to classify images as blurry or clear.
– A SRCNN model is implemented to transform poor-resolution aerial images to high-resolution aerial images.

– The proposed system achieves high detection accuracy and real-time operation.

The remainder of the article is organised as: In Sect. 2, the theoretical background of deep learning methods are discussed. Then, Sect. 3 explains about the implementation of proposed system. In Sect. 4, the experimental results are discussed. Finally, Sect. 5 concludes the paper with future directions

## 2 Theoretical background

Developments in deep learning specifically, in convolutional neural networks (CNN) for computer-vision applications have greatly increased the performance of feature classification (Khalilpourazari and Khalilpourazary 2018) and object detection. Graphic Processing Units have also made a major contribution to CNN implementations by using parallelization to solve the problems of real-time execution of data-intensive activities (Khalilpourazari and Mohammadi 2018; Shehab et al. 2019). In fact, emerging patterns of heavy processing, such as video stream analysis, have now been allowed in cloud computing (Khalilpourazari et al. 2019; Abualigah et al. 2019), enabling analysis of video in real time by sophisticated deep-learning algorithms, as applied, for example, to surveillance (Khalilpourazari and Pasandideh 2019; Abualigah et al. 2018). Deep-learning techniques are representation learning approaches at various levels of abstraction. They are acquired by writing simple but highly nonlinear frameworks that transform representation at one stage, starting with raw input, into representation at greater, much more complex stages (Shehab et al. 2020).

For classification problems, higher layers of representation amplify those input characteristics that are important for segregation and suppress insignificant variations. YOLO recognizes objects by splitting an image into grid blocks, as opposed to the region approach used by two-stage detectors. The function map of the YOLO output layer is intended to display bounding box coordinates, object scores, and class scores. YOLO (Redmon et al. 2016) also recognizes several objects with one inference, with consequently much higher speed of detection compared to traditional methods. Localization errors are nevertheless high due to grid unit processing, and the precision of recognition is poor, making YOLO unsuitable for object-recognition applications. Redmon and Farhadi (2017) suggested YOLOv2 to address these issues, with detection efficiency improved by adopting a batch normalization process for the convolution layers. YOLOv2 also incorporates an anchor box, multi-level training, and fine-grained characteristics. Still, the detection accuracy for small objects remains low. Therefore, (Redmon and Farhadi 2018) introduced YOLOv3. YOLOv3 comprises convolution layers and a deep network for improved accuracy. It uses a residual skip relation to address the vanishing gradient problem in deep networks and a process of up-sampling and concatenation to hold fine-grained features and identify small objects. The most notable feature is its identification at three different levels, as used in a pyramid network feature (Lin et al. 2017). This lets YOLOv3 track objects of different sizes. When an image with three channels is given as input (i.e., R, B, and G) into the YOLOv3 system, bounding box coordinates and scores of objects and classes are obtained. The outcomes from the three levels are mixed and analyzed using non-maximum suppression. Next, the results of the final detection are determined. Hence, YOLOv3 is suitable for object-recognition applications requiring precision and speed.

## 3 Proposed blade damage detector model

Generally, image classification is used to assign a picture under a certain category or class, whereas the purpose of object detection is to classify the location of target items in an image using bounding boxes, along with the object classification. The condition assessment of the WT blades can be performed by using YOLOv3 object detection model through the following steps. As shown in Fig. 1, pre-processing of images from the surveillance drone can be performed as follows. First, the original drone inspection image collection is divided into two groups: (1) high-quality images that can be processed directly and (2) low-resolution, blurry images. The second, blurry images are transformed into a high-resolution image set through a super-resolution reconstruction process described briefly in Sect. 3.2. The transformed images are later combined with the original, high-resolution images and may be applied as a new image set for further processing. The original input images were also resized to $416 \times 416$ to accelerate the learning process. To extract WT blade damage features, the resized images are tested as input to the Darknet-53. The feature-pyramid-network (FPN) method yields predictions over three distinct stages. Overall, YOLOv3 predictions contain the bounding box variables, object score, and class prediction. The process of eliminating low confidence boxes is regarded as the non-maximum suppression process as shown in Fig. 2. Next, YOLOv3 extracts the anchors that overlay the ground truth subject under multiple thresholds and then give the classification of each bounding box and related positioning.

### 3.1 Image pre-processing unit

We used the variance of the Laplace distribution to differentiate blurry from clear aerial images. The Laplace operator is chosen to calculate the image's second-degree
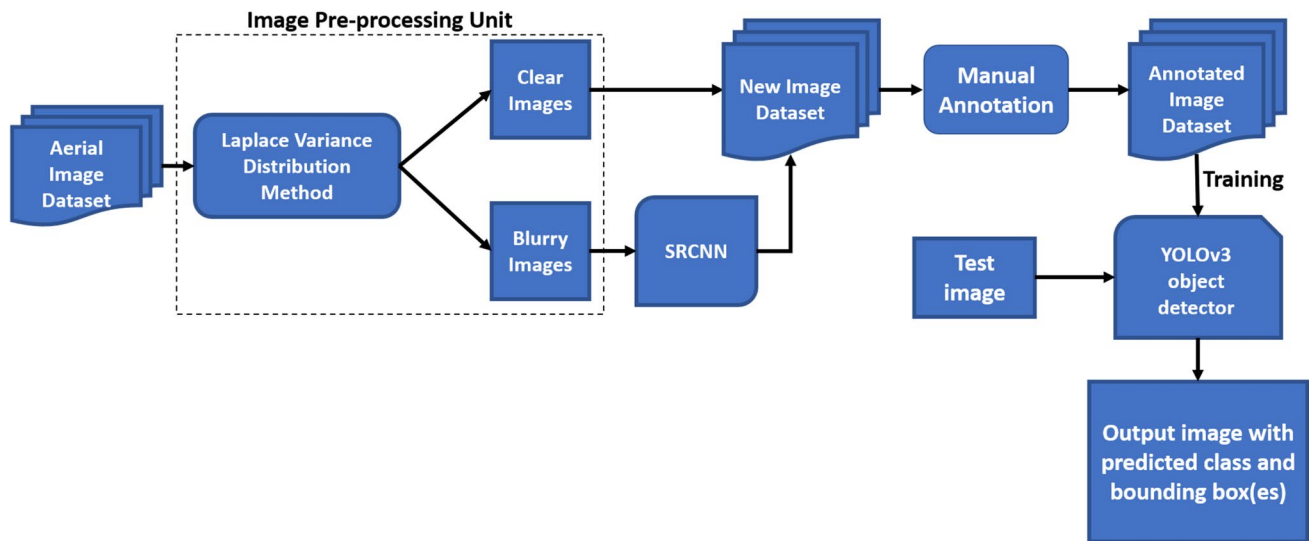
**Fig. 1** Damage detection model using YOLOv3

**Fig. 2** Non-maximum suppression process



Before Non-Max Suppression

After Non-Max Suppression

differential in order to increase the contrast between neighboring picture elements. Essentially, the Laplace operator is first used to convolute the image and then to determine the deviation. In sharp images, the border seems stronger, so the divergence significantly increases; in blurry images, by contrast, the edge detail is comparatively less, and so is the difference. Therefore, if the deviation is less than the specified threshold, the picture is considered blurred, while the picture is labeled clear if the divergence is greater than the stated threshold value (Fig. 3). The initial surveillance images are thus subdivided into blurry and clear image groups. Next, the blurry images are used as inputs to the SRCNN model, which transforms them. Finally, the transformed images are merged with the clear images from the initial dataset to form a new image dataset as shown in Fig. 1.

### 3.2 SRCNN reconstruction

Some aerial images are blurry as a result of drone body shuddering and imaging exposure issues, severely impeding the effective recognition of WT blade damage. Pre-processing of images is a necessary step in the formation of the deep-learning training set. Dong et al. (2014) proposed the SRCNN model, which is a network training algorithm for pre-processing image data. The SRCNN algorithm effectively addresses the above-described issues by changing poor-resolution pictures to super-resolution ones. Implementation of the SRCNN model typically comprises three stages: extraction of image patches, non-linear projection, and reconstruction of super-resolution image frames.

*Image patches extraction* This operation selects (overlapping) maps from blurry image Y, representing each map as a high-dimensional vector. These vectors comprise a series

**Fig. 3** Calculation of blurriness in the input images



Image with blurness       Image with no blurness
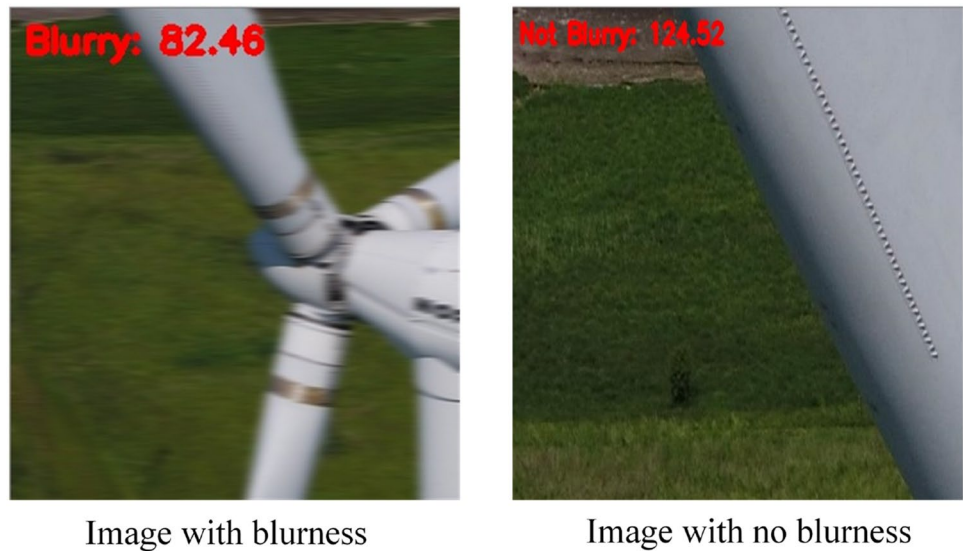
of function maps, the number of which is proportional to the vector dimension. The standard procedure is to obtain patches from the initial surveillance pictures and employ a sequence of convolution filters. Every convolution filter can be considered a basis, and simple development can be included in the optimization network. The first layer operation is denoted as:

$$F_1(Y) = maximum(0, W_1 * Y + B_1) \tag{1}$$

where $W_1$ represents $n_1$ filters of size (e x $f_1$ x $f_1$), 'e' is the number of raster bands in the input, $f_1$ is the map coverage of the filter, $B_1$ is an $n_1$-spatial segment, and '$*$' represents convolution. The product of the convolution process contains $n_1$ attribute features, and the final image from the primary convolution level is achieved over the Rectified Linear Unit ReLU (max(0, x)) activation method (Nair and Hinton 2010).

*Non-linear projection* An $n_1$-dimensional attribute vector is obtained from each frame of images in the primary convolution section layer. The resulting $n_1$-dimensional attribute vector in the second convolution layer is projected non-linearly to the $n_2$-dimensional feature segment. Second layer processing is defined as:

$$F_2(Y) = maximum(0, W_2 * F_1(Y) + B_2) \tag{2}$$

where $W_2$ denotes $n_2$ filters of size $n_1 \times f_2 \times f_2$, and $B_2$ represents a $n_2$-spatial vector. The convolution product contains $n_2$ feature maps.

*Reconstruction of super resolution image* The last convolution section layer integrates all the super-resolution frames created with the help of the upper layers to produce a super-resolution model, the SRCNN network's final image output. Therefore, the third layer process is:

$$F_3(Y) = W_3 * F_2(Y) + B_3 \tag{3}$$

where $W_3$ denotes e filters of scale $n_2 \times f_3 \times f_3$ and $B_3$ represents a e-sized vector.

*SRCNN structure* The SRCNN network uses the bicubic interpolation model to enlarge the collected, smeared surveillance picture to the desired size and register the interlaced image as Y. Super-resolution restoration aims to rebuild Y to an appropriately high-resolution rasterized image H, analogous to the actual image X. Through training, we will obtain the respective end-to-end mapping feature F(Y).

The fundamental arrangement of the SRCNN model is illustrated in Fig. 4, where it may be observed that the total structure contains a three-stage CNN. The primary convolution stage extracts the frames of images from Y and then identifies certain features with low resolution. The next convolutional layer produces features of high resolution using non-linear mapping. Lastly, the reconstruction of super-resolution images is accomplished by the third convolution layer, which is analogous to creating images near to the actual resolution. To achieve the mapping feature F between super-resolution images, it is necessary to train the system variable $\theta = (W_1, W_2, W_3, B_1, B_2, B_3)$ in the specific process. To learn these variables, the difference between the replicated photograph F(Y;$\theta$)and the actual image X with high-resolution must be minimized. The mean squared error (MSE) is taken as a function of loss, and expression for MSE is expressed as:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} ||F(Y_i;\theta) - X_i||^2 \tag{4}$$
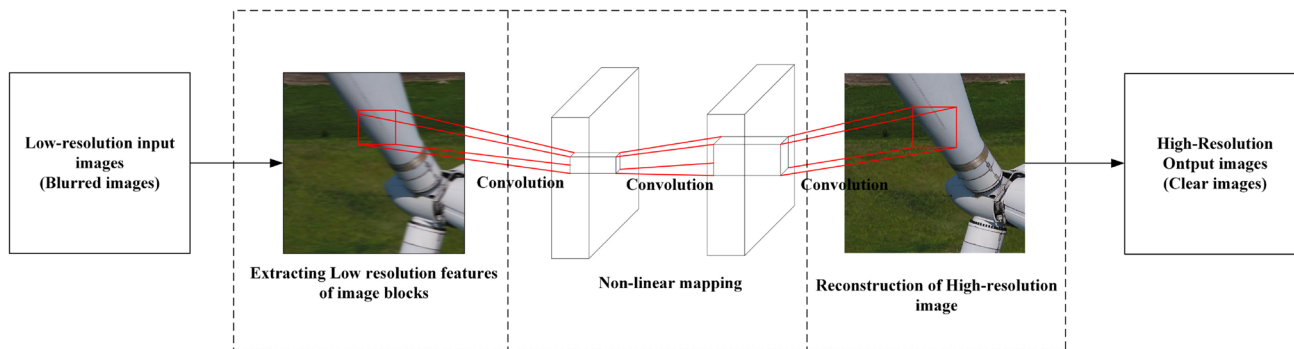
## SRCNN 3-stage Network structure



**Fig. 4** SRCNN network structure

where $n$ denotes the number of training examples, $X_i$ is the actual image, $Y_i$ represents the poor-resolution reference image, and $F(Y_i;\theta)$ is the clear image produced by the SRCNN model.

### 3.3 Damage detection system using YOLOv3

Object recognition is a computer vision task that requires one or more targets to be located inside an image, in which each object can be labelled. A recent deep learning approach, Selective Search, has been employed to minimize the number of bounding boxes the algorithm needs to check, such as R-CNN and Fast R-CNN. Another technique, called Overfeat, consists of multi-stage picture scanning utilizing sliding window-like techniques. At the other side, YOLO approaches the problem of target recognition in a different way. This only transfers the entire image across the network once. SSD is another object detection algorithm that generates a deep-learning network for a picture once, but it is much slower than YOLOv3, although the latter retains relatively similar accuracy. The YOLO model family is a series of in-depth learning models intended to identify objects quickly. A single, functioning, neural network (formerly a GoogleNet-edition, later revised and recognized as DarkNet based on VGG) divides the input into a cell grid and each cell explicitly predicts a bounding box and the classification of objects. The outcome is a significant number of bounding-rectangles, consolidated by a post-processing step into a final prediction as shown in Fig. 5.
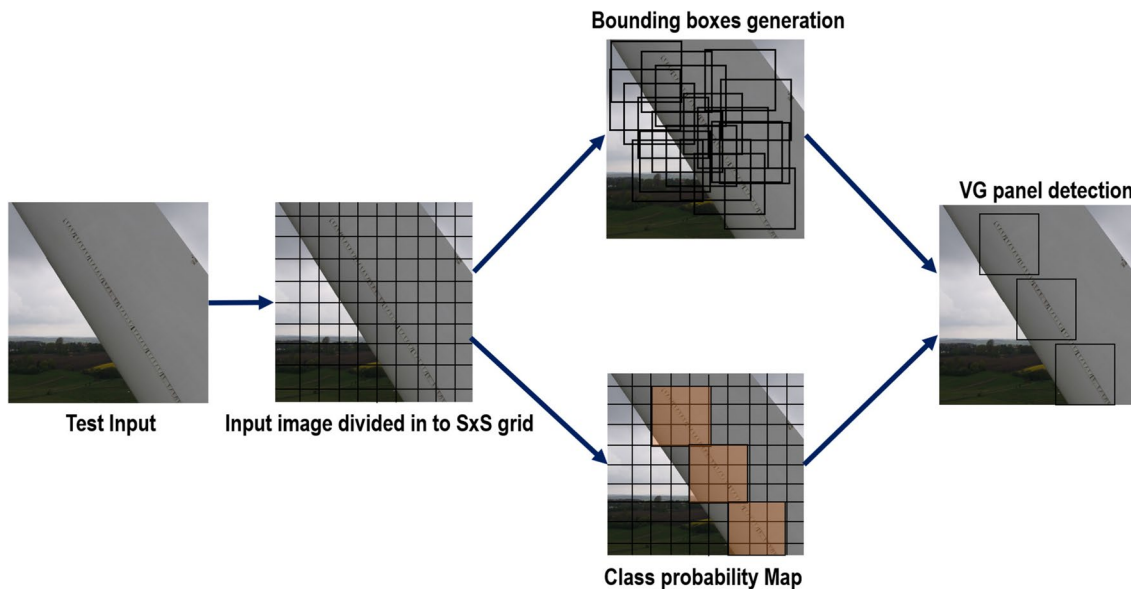


**Fig. 5** YOLOv3 architecture

YOLOv3 uses the FPN concept to predict boxes at different scales. It uses a certain number of convolution layers and extra layers (residual-layers) to finish the procedure of detection and manages the attributes of the whole picture to predict every minimum bounding-rectangle. Meanwhile, it predicts the realization of top-to-bottom training across all classes of all bounding rectangles, those that hold significant average accuracy and high efficiency in real-time. YOLOv3 begins the process by splitting the input surveillance image into N × N blocks and attaches a bounding rectangular anchor for every ground truth on the map. The network finds four parameters $(t_x, t_y, t_w, t_h)$ for each bounding box, as shown in Fig. 6, and then implements a method to predict four associated coordinates: the two midst coordinates $(b_x, b_y)$ of the bounding rectangle,$(C_x, C_y)$ denotes the location of the grid, and the height $b_h$ and width $b_w$. The minimum bounding rectangle prediction and the Intersection-over-Union (IOU) equations are given as follows:

$$b_x = \sigma(t_x) + C_x \tag{5}$$

$$b_y = \sigma(t_y) + C_y \tag{6}$$

$$b_w = p_w e^{t_w} \tag{7}$$

$$b_h = p_h e^{t_h} \tag{8}$$

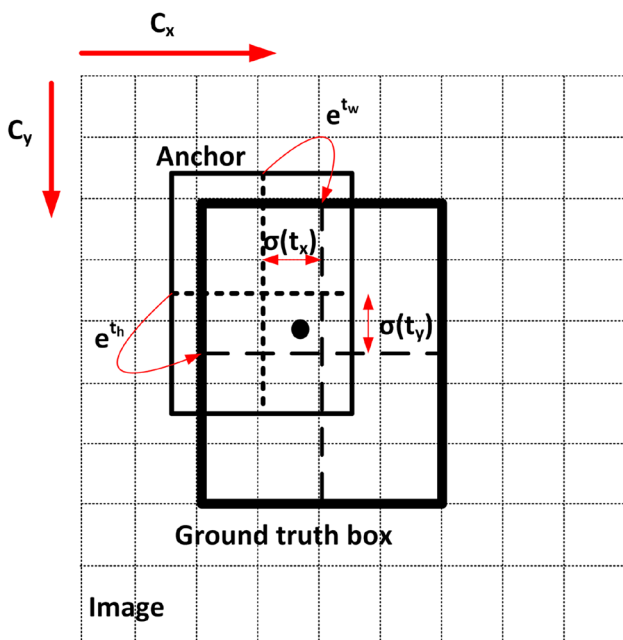$$IOU = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} \tag{9}$$



**Fig. 6** Bounding box detection

where the IOU shown in expression 9 is the amount of variance between the device detected bounding rectangle and the ground truth rectangle. BBgt is the learning label-based ground truth rectangle, BBdt is the bounding rectangle for detection, and area (.) shows the region.

*Network structure of YOLOv3* The main configuration of the YOLOv3 system is shown in Fig. 7, which inherits its framework from Darknet-53. This network is a fusion of YOLOv2 (Zhao and Qu 2018), Darknet-19 (Ghenescu et al. 2018), and ResNet (Szegedy et al. 2017). So, YOLOv3 mainly uses convolution kernels 1 × 1 and 3 × 3, and some related structures with shortcuts. The input surveillance image is processed first, after which its size is changed to 416 × 416, and then it is processed using YOLOv3.

1. The first section comprises two layers of convolution form. The size of input image is 416 × 416 × 3 and the kernel size is 3 × 3 × 64 and 3 × 3 × 32. The size of the output match function is reduced to 208 × 208 × 64 after completion of the convolution process.
2. The second section comprises three levels of convolution, followed by a residual layer. The convolution kernel size is 3 × 3 × 128, 3 × 3 × 64, and 1 × 1 × 32, and the output map is reduced to 104 × 104 × 128 after completion of the convolution process.
3. The third section comprises five convolution layers with two layers of residual form. The scale of the convolution kernel is 3 × 3 × 256, 3 × 3 × 128, and 1 × 1 × 64, and the output function map is reduced to 52 × 52 × 256 after the process of convolution.
4. The fourth section comprises 17 convolution layers with eight residual layers. The convolution kernel size is 1 × 1 × 128, 3 × 3 × 256, and 3 × 3 × 512, and the output function map is reduced to 26 × 26 × 512 after completion of the convolution process.
5. The fifth section contains 17 convolution layers and eight residual layers. The dimensions of the convolution kernel are 1 × 1 × 256, 3 × 3 × 1024, and 3 × 3 × 512. The output function map is reduced to 13 × 13 × 1024 after completion of the convolution process.
6. The sixth section comprises eight convolution and four residual form layers. The convolution kernel size is 1 × 1 × 512 and 3 × 3 × 1024, and the output function map stays the same after completion of the convolution process.
7. The last section comprises three prediction networks. YOLOv3 predicts rectangular boxes on three various levels, and then extracts the attributes of these scales. The network prediction output is a 10 × 10 × (3 × (4 + 1 + 6))tensor for four minimum rectangular box corrections, one objectness projection, and six classifiers as shown in Table 2.
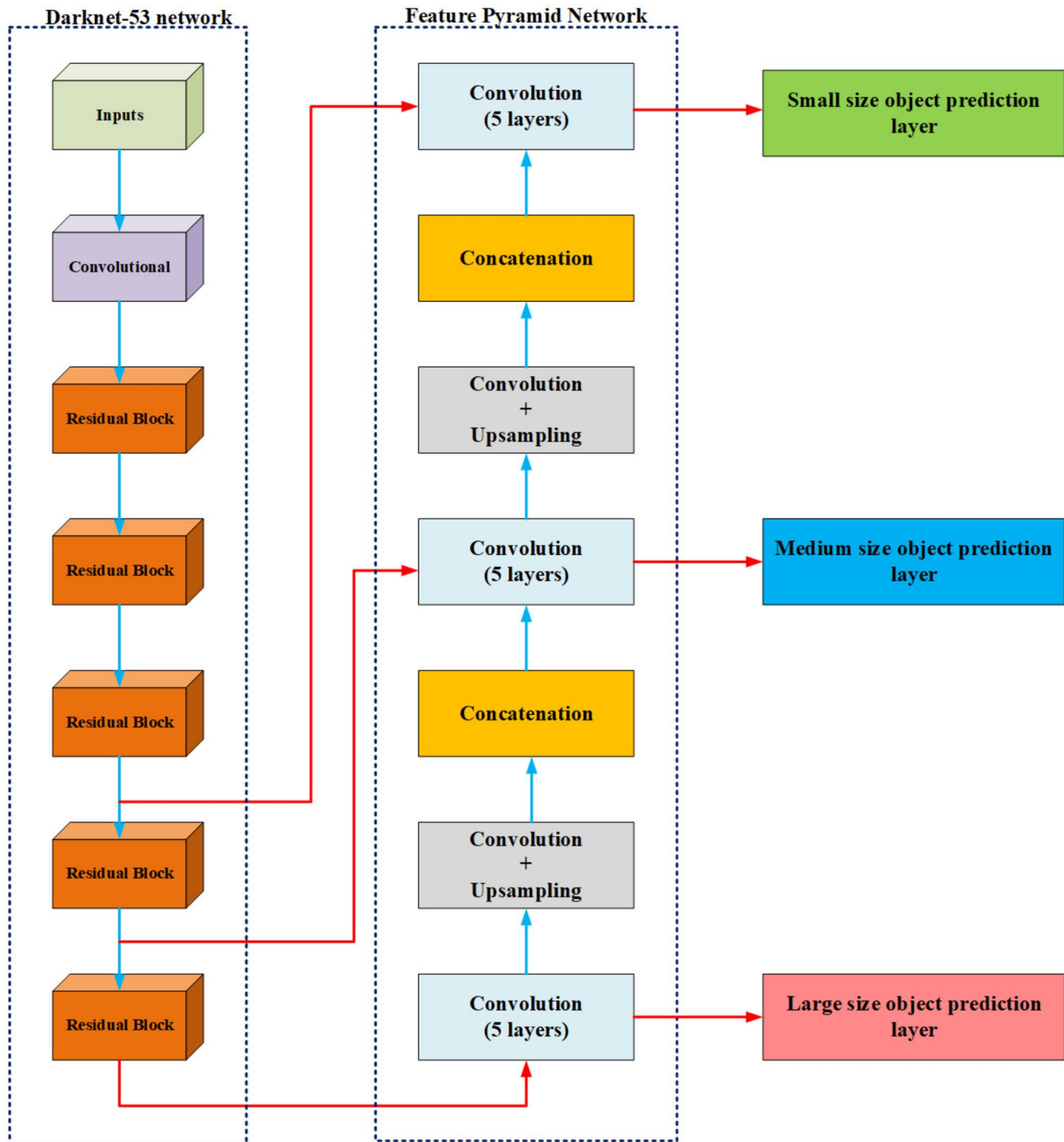
**Fig. 7** Overview of YOLOv3 network architecture

*Training* The training of the YOLOv3 network is split into three activities, as follows. In Step 1, the size of the aerial image captured by surveillance drone, $5280 \times 2970$, is too large to be the network input. As such, the actual image size is changed to $416 \times 416$ in order to accelerate the training procedure. In Step 2, the VOC2007 (Rao et al. 2019) dataset pattern is used to label the outer form of the leading edge (LE) erosion, the vortex generator (VG) Panel, the VG with vanished teeth, and the lightning receptor to show up in a particular image. Lastly, in Step 3, the system variables of the YOLOv3 algorithm are initiated, and the network is trained to get variables for the identification of specified objects.

**Table 2** Dataset classification

| Classes | Training annotations | Testing annotations |
|---|---|---|
| VG panel | 120 | 24 |
| VG panel with missing teeth | 80 | 24 |
| LE corrosion | 120 | 24 |
| Crack | 80 | 26 |
| Lightning detector | 140 | 28 |
| Damaged lightning detector | 60 | 20 |

*Essential parameters* This paper presents an additional study of the choice of three essential variables. Batch size: Theoretically, the greater the volume, the easier the preparation. Nonetheless, owing to hardware constraints, we cannot increase the value forever, so the authors attempted four different batch sizes of 8, 16, 64, and 128, respectively. When the batch size was chosen as 64, 16, 8 during preparation, there would be no loss of power, so we chose 64 as the batch size based on the above argument. Weight decay: To avoid overfitting, we first set the correct rate of learning and then modified the decay measure from the constant value (0.01) to the final measure (0.0005). Ignore thresh: The value of the IOU threshold defines the number of IOUs used in the loss estimation. If the pre-defined threshold is less, it will cause underfitting, whereas if the threshold limit is high, it obviously causes overfitting. Thus, the ignore thresh value is set as 0.65 on the basis of the above argument and the case at hand.

*Selection of parameters* The choice of the three variables listed above will influence the detection accuracy. It is therefore important to change these parameters to prevent underfitting and overfitting while training. To boost the precision of the typical detection, YOLOv3 adopts a multi-label categorization that is distinctly against the old interpretations that use a contradictory label. The logistic classifier is used to determine the objectness value for each bounding rectangle. For classification loss while testing, YOLOv3 uses a discrete cross-entropy loss for each number, which eliminates the MSE commonly exploited in past versions. The loss factor used in the YOLOv3 feature training is shown as follows:

$$Loss(s_m) = \begin{cases} -\log_2(s_m) & \text{if } g_m = 1 \\ -\log_2(1 - s_m) & \text{if } g_m = 0 \end{cases} \quad (10)$$

where m represents the sample count, $s_m \in [0, 1]$ reflects the objectness value anticipated by the system, which calculates the expected likelihood that the *m*th sample is WT turbine blade damage. Additionally, $g_m$ shows the ground truth. It needs to be observed that $g_m \in (0, 1)$ is implied when the *m*th observation relates to the object class. Network variables are trained by reducing the loss to all samples, i.e. $\sum_m Loss(s_m)$.

In this article, the Adaptive Moment Estimation, shortly referred to as the Adam Optimization procedure (Kingma and Ba 2019), is used to change network parameters. Adam would be the first order optimization mechanism that can substitute the standard Stochastic gradient descent method and adjust network weights recursively, depending on training data. The model calculates the appropriate adaptive training score for various values by determining the gradient estimate of primary and secondary moments. It integrates two optimization algorithms, including the advantage of a root-mean-square propagation and adaptive gradient algorithm (Ruder 2016), which are useful for improving the effectiveness of scattered gradients and efficiency of training.

## 4 Results and discussion

The general specifications of the personal computer engaged in this study are as follows: Intel(R) Xeon(R) CPU E5-2780, NVIDIA 1080 RTX as GPU, main frequency 2.80 GHz. The operating software is open source type Ubuntu 16.04, and Tensorflow is the platform for algorithm development.

*Dataset* In this study, the authors used surveillance drone images of the WT, a publicly available dataset. It is the only available WT image-set comprising a collection of 700 images. This collection includes the 2017 and 2018 period inspection photos of the Nordtank WT installed at the DTU Wind Power test site in Roskilde, Denmark (Shihavuddin and Chen 2018). To increase the dataset, collected 300 more images from the internet from different sources. Those pictures without part of the blade or a little part of the blade have been manually inspected and removed. The images in the dataset are classified as (a) VG panel, (b) VG panel with missing teeth, (c) LE corrosion, (d) Crack, (e)Lightning detector, (f) Damaged lightning detector. These classes are the static indicators of WT blades' health. The reason for choosing these classes for analysis is that all of these damages provide improved visual characteristics and can be detected by the inspection team, and would be helpful if the program had to learn to do the same thing. For every class, Table 2 lists the number of annotations which are manually annotated and viewed as ground truths. Examples of manually annotated damage of WT blades are shown in Fig. 8.

*Analysis of the SRCNN results* The blurred image found by classification process was used as a test object for reconstruction of high-resolution images. The analysis of the SRCNN indicates that the restoration effect is influenced by the count and the filter size, as well as the considerable layers of the model. The authors referred to the configuration of the SRCNN algorithm in (Dong et al. 2016; Zhou et al. 2018; Elsaid and Wu 2019). Simultaneously, standard models such as bicubic convolution interpolation
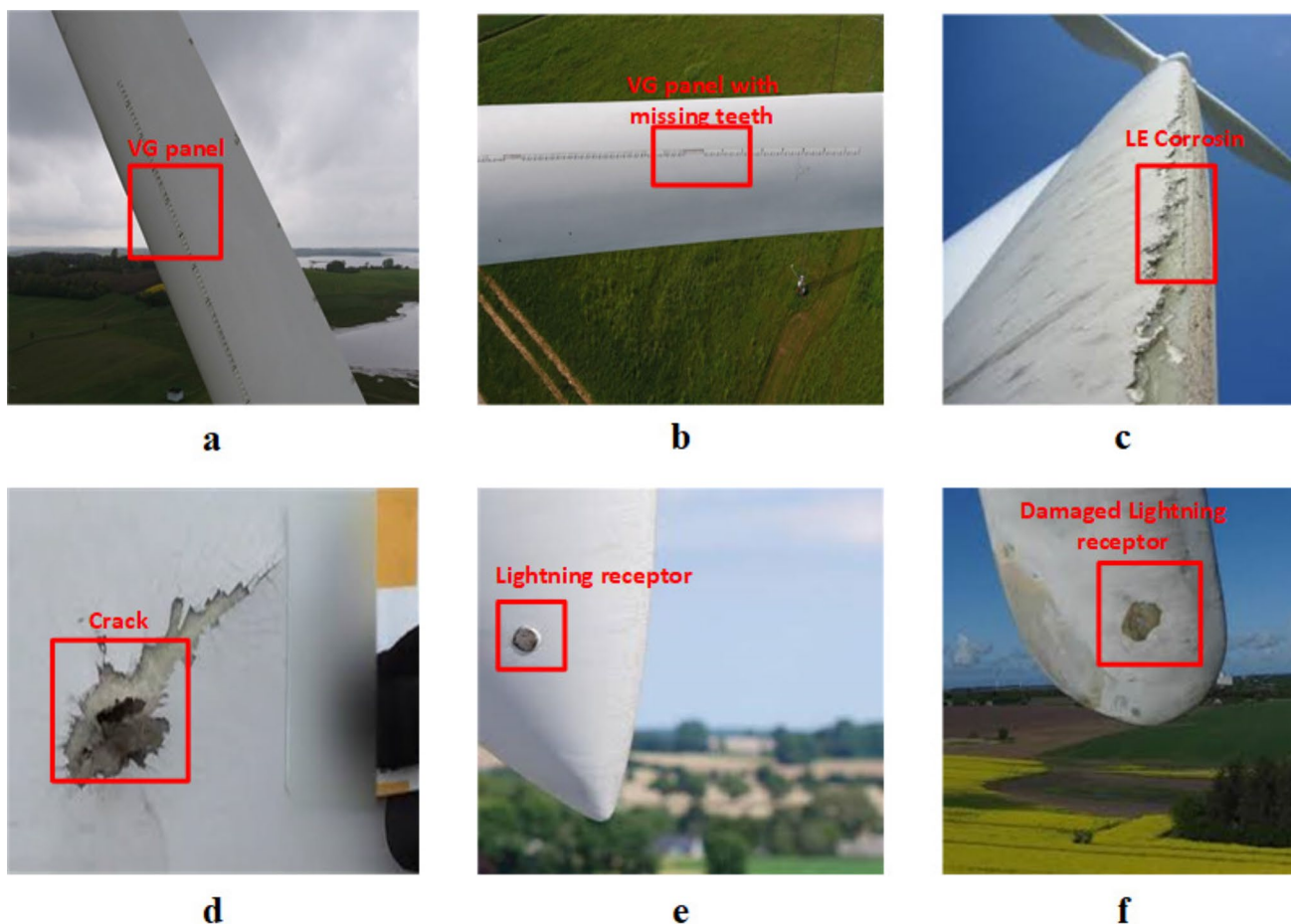
**Fig. 8** **a** VG panel, **b** VG panel with missing teeth, **c** LE corrosion, **d** cracks, **e** lightning detector, **f** damaged lightning receiver

(Tian et al. 2012) and sparse coding (Druzhkov and Kustikova 2016) are compared for test analysis. Figure 9 is the plot of peak signal to noise ratio (PSNR) in decibels and backprops. It is evident from the plot that the SRCNN has the best PSNR values for various backprops when contrasting them with other models.

*Analysis of the YOLOv3 results* We base this research on the YOLOv3 model and configure the network. Several key variables are selected, in which the kernel function is set to $1 \times 1$ or $3 \times 3$, the step-size being 1, and the size of the batch set to 64. Variables are revised once per batch of testing instances and the coefficient of decay weight is 0.0005. The adaptive rate of learning is calculated with a
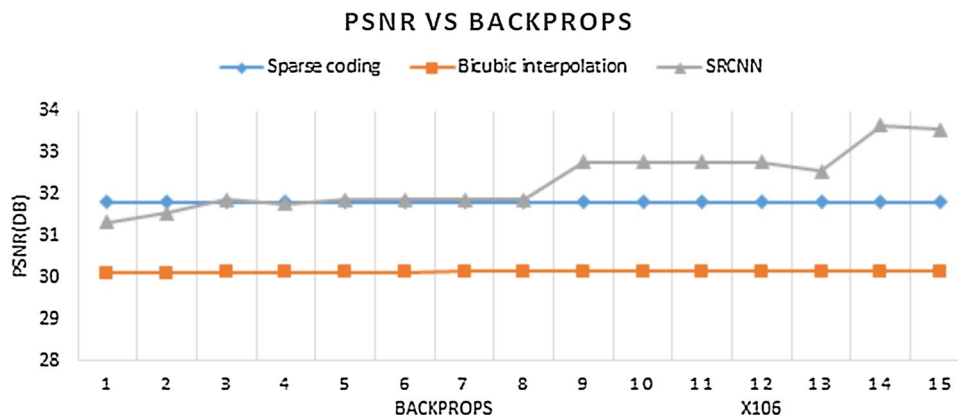
**Fig. 9** PSNR vs back propagations

**Table 3** Comparison of state-of-the-art deep learning models with respect to average accuracy

| Iterations | Average Accuracy (%) | | |
|---|---|---|---|
| | Faster R-CNN (Shihavuddin et al. 2019) | YOLOv2 (Zhao and Qu 2018) | YOLOv3 (Proposed method) |
| 100 | 67 | 66 | 67 |
| 1000 | 71 | 70 | 72 |
| 2000 | 85 | 83 | 84 |
| 3000 | 91 | 91 | 92 |
| 4000 | 94 | 92 | 94 |
| 5000 | 95 | 93.5 | **96** |

**Table 4** Comparison of training time and average processing time per image for different models

| Model | Training time (in Min) | Average processing time per image (in s) |
|---|---|---|
| Faster R-CNN | 220 | 0.652 |
| YOLOv2 | 120 | 0.522 |
| YOLOv3 | 90 | 0.20 |



**Fig. 10** Comparison of Loss curves for three deep learning models

**Table 5** Comparison of mean average precision

| Model | Mean average precision |
|---|---|
| Faster R-CNN | 0.87 |
| YOLOv2 | 0.90 |
| YOLOv3 | 0.96 |

rate of 0.01 and a highest iteration count of 5000. During the training process, the accuracy of state-of-the-art deep learning algorithms, Faster R-CNN, YOLOv2 and YOLOv3, in recognizing the training objects is measured, while the number of iterations is 500, 1000, 2000, 3000, 4000, and 5000, respectively. The findings are tabulated in Table 3, where it can be perceived that the average accuracy of detection of the three algorithms also improves steadily with the rise in iteration cycles, with the maximum accuracy of recognition of the YOLOv3 algorithm approaching 95.6%. After the training process, the model has been assessed with a test set consisting of 200 images.

From Table 4, it is also noted that the training time of YOLOv3 (90 Min) is comparatively shorter than other models such as Faster RCNN (220 Min) and YOLOv2 (120 Min). The loss curves of all models were evaluated in the training process as shown in Fig. 10.

From Table 5, we can see that YOLOv3 has the best value of mean average precision and the least prediction time when compared with Faster R-CNN and YOLOv2 models. In conclusion, compared with Faster R-CNN and YOLOv2, the suggested model has better improvement in the detection accuracy of WT blade damage. The recognition results of such damage in the surveillance image are displayed in Fig. 11, from where it can be observed that the suggested algorithm in this paper will precisely locate the blade damage. Meanwhile, it also verifies the validity of the WT blade damage detection algorithm implemented in this paper.
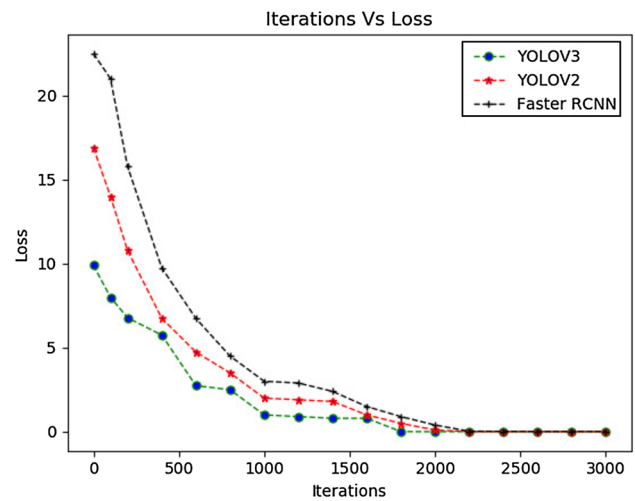
## 5 Conclusion

This paper proposes a YOLOv3-based model for the recognition of WT blade damage from surveillance drone images in order to resolve the poor detection accuracy problem associated with conventional methods. The presented scheme includes two major steps: during the first step, low-resolution drone images are identified and transformed to a super-resolution images using the SRCNN method. In the second step, the advanced deep learning method YOLOv3 is used to detect damage to turbine blades in the aerial images. The YOLOv3 model, together with the SRCNN model, can precisely detect the location and condition of damage to WT blades under various directions, illumination intensities, and backgrounds. SRCNN's peak signal-to-noise ratio was compared with other standard methods such as bi-cubic convolution interpolation and sparse coding. The test results showed that the proposed SRCNN had the highest peak signal-to-noise ratio compared to other models. Experiments were conducted using state-of-the-art, deep-learning, object-detection techniques, and the findings show that the proposed YOLOv3 solution has the fastest average detection speed
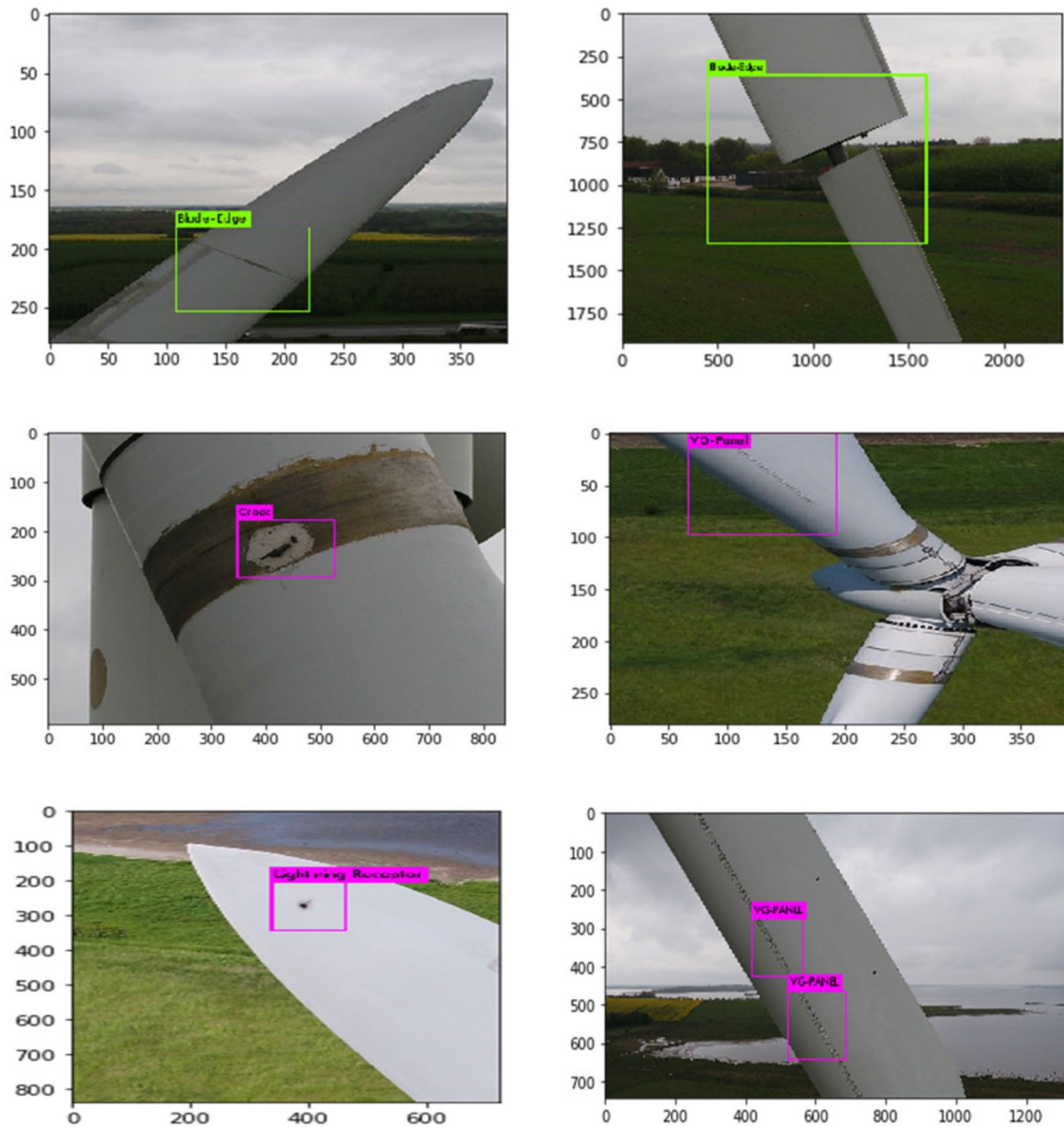
**Fig. 11** The detection of WT blade damages on test images using proposed model

of 0.20 s and the maximum average precision of 0.96. In future work, we will use own surveillance images to validate the proposed method. Also, we will use advanced image augmentation methods such as Generative adversarial networks to achieve the best performance from a proposed system.

## Compliance with ethical standards

## References

Abualigah LM, Khader AT, Hanandeh ES (2018) A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering. Intell Dec Technol 12(1):3–14

Abualigah L, Shehab M, Alshinwan M, Alabool H (2019) Salp swarm algorithm: a comprehensive survey. Neural Comput Appl 2019:1–21

Al-Khudairi O, Ghasemnejad H (2015) To improve failure resistance in joint design of composite wind turbine blade materials. Renew Energy 81:936–951

Chen H, He Z, Shi B, Zhong T (2019) Research on recognition method of electrical components based on yolo v3. IEEE Access 7:157818–157829

Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. Eur Conf Comput Vis 2014:184–199

Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. Eur Conf Comput Vis 2016:391–407

Druzhkov P, Kustikova V (2016) A survey of deep learning methods and software tools for image classification and object detection. Pattern Recogn Image Anal 26(1):9–15

Dutton AG (2004) Thermoelastic stress measurement and acoustic emission monitoring in wind turbine blade testing. Eur Wind Energy Conf Lond 2004:22–25

Elsaid NM, Wu Y-C (2019) Super-resolution diffusion tensor imaging using srcnn: a feasibility study. In: 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, pp 2830–2834

Ghenescu V, Mihaescu RE, Carata S-V, Ghenescu MT, Barnoviciu E, Chindea M (2018) Face detection and recognition based on general purpose dnn object detector. In: 2018 international symposium on electronics and telecommunications (ISETC), IEEE, pp 1–4

Häckell MW, Rolfes R (2013) Monitoring a 5 mw offshore wind energy converter—condition parameters and triangulation based extraction of modal parameters. Mech Syst Signal Process 40(1):322–343

Hosoi A, Yamaguchi Y, Ju Y, Sato Y, Kitayama T (2015) Detection and quantitative evaluation of defects in glass fiber reinforced plastic laminates by microwaves. Compos Struct 128:134–144

Jasinien E, Raiutis R, Voleiis A, Vladiauskas A, Mitchard D, Amos M et al (2009) Ndt of wind turbine blades using adapted ultrasonic and radiographic techniques. Insight-Non-Destruct Test Condition Monitor 51(9):477–483

Khalilpourazari S, Khalilpourazary S (2018) Optimization of time, cost and surface roughness in grinding process using a robust multi-objective dragonfly algorithm. Neural Comput Appl 2018:1–12

Khalilpourazari S, Pasandideh SHR (2019) Sine-cosine crow search algorithm: theory and applications. Neural Comput Appl 2019:1–18

Khalilpourazari S, Soltanzadeh S, Weber G-W, Roy SK (2019) Designing an efficient blood supply chain network in crisis: neural learning, optimization and case study. Ann Oper Res 2019:1–30

Khalilpourazari S, Mohammadi M (2018) A new exact algorithm for solving single machine scheduling problems with learning effects and deteriorating jobs. arXiv:1809.03795

Kingma DP, Ba JA (2019) A method for stochastic optimization. arXiv:1412.6980

Krause T, Ostermann J (2020) Damage detection for wind turbine rotor blades using airborne sound. Struct Control Health Monitor 27(5):e2520

Lee J-K, Park J-Y, Oh K-Y, Ju S-H, Lee J-S (2015) Transformation algorithm of wind turbine blade moment signals for blade condition monitoring. Renew Energy 79:209–218

Li Z, Haigh A, Soutis C, Gibson A, Sloan R, Karimian N (2016a) Delamination detection in composite t-joints of wind turbine blades using microwaves. Adv Compos Lett 25(4):096369351602500401

Light-Marquez A, Sobin A, Park G, Farinholt K (2011) Structural damage identification in wind turbine blades using piezoelectric active sensing. Struct Dyn Renew Energy 1:55–65

Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125

Li Z, Soutis C, Haigh A, Sloan R, Gibson A, Karimian N (2016b) Microwave imaging for delamination detection in t-joints of wind turbine composite blades. In: 2016 46th European Microwave Conference (EuMC), IEEE, pp 1235–1238

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single shot multibox detector. Eur Conf Comput Vis 2016:21–37

Mandell JF, Samborsky DD, Agastra P (2008) Composite materials fatigue issues in wind turbine blade construction. In: SAMPE 2008

Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814

Ozbek M, Meng F, Rixen DJ (2013) Challenges in testing and monitoring the in-operation vibration characteristics of wind turbines. Mech Syst Signal Process 41(1–2):649–666

Pandit RK, Infield D (2019) Comparative assessments of binned and support vector regression-based blade pitch curve of a wind turbine for the purpose of condition monitoring. Int J Energy Environ Eng 10(2):181–188

Pitchford C, Grisso BL, Inman DJ (2007) Impedance-based structural health monitoring of wind turbine blades. In: Health monitoring of structural and biological systems 2007, vol 6532, pp 65321I. International Society for Optics and Photonics

Prowell I, Veletzos M, Elgamal A, Restrepo J (2009) Experimental and numerical seismic response of a 65 kw wind turbine. J Earthq Eng 13(8):1172–1190

Raišutis R, Jasiūnienė E, Žukauskas E (2008) Ultrasonic ndt of wind turbine blades using guided waves. Ultrasound 63(1):7–11

Rao C, Fan Y, Su K, Latecki LJ (2019) Common object discovery as local search for maximum weight cliques in a global object similarity graph. In: International conference on discrete geometry for computer imagery, Springer, pp 219–233

Reddy A, Indragandhi V, Ravi L, Subramaniyaswamy V (2019) Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics. Measurement 147:106823

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271

Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv:1804.02767

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 2015:91–99

Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747

Shehab M, Abualigah L, Al Hamad H, Alabool H, Alshinwan M, Khasawneh AM (2019) Moth-flame optimization algorithm: variants and applications. Neural Comput Appl 2019:1–26

Shehab M, Alshawabkah H, Abualigah L, Nagham A-M (2020) Enhanced a hybrid moth-flame optimization algorithm using new selection schemes. Eng Comput 2020:1–26

Shihavuddin ASM, Chen X (2018) DTU - Drone inspection images of wind turbine. Mendeley Data, V2. https://doi.org/10.17632/hd96prn3nc.2

Shihavuddin A, Chen X, Fedorov V, Christensen AN, Riis NAB, Branner K, Dahl AB, Paulsen RR (2019) Wind turbine surface damage detection by deep learning aided drone inspection analysis. Energies 12(4):676

Sørensen BF, Lading L, Sendrup P, McGugan M, Debel CP, Kristensen OJD, Larsen GC, Hansen AM, Rheinländer J, Rusborg J, Vestergaard JD (2002) Fundamentals for remote structural health monitoring of wind turbine blades-a preproject. Risø National Laboratory, Denmark. Forskningscenter Risoe. Risoe-R, 1336 p

Sutherland H, Beattie A, Hansche B, Musial W, Allread J, Johnson J, Summers M (1994) The application of non-destructive techniques to the testing of a wind turbine blade. In: Technical report, Sandia National Labs., Albuquerque, NM (United States)

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI

Tian Y, Yap K-H, He Y (2012) Vehicle license plate super-resolution using soft learning prior. Multimedia Tools Appl 60(3):519–535

Tu T-H, Lo F-C, Liao C-C, Chung C-F, Chen R-C (2019) Using wind turbine noise to inspect blade damage through portable device. Inst Noise Control Eng 259–8:1787–1791

Wang L, Zhang Z (2017) Automatic detection of wind turbine blade surface cracks based on UAV-taken images. IEEE Trans Industr Electron 64(9):7293–7303

Wang L, Zhang Z, Xu J, Liu R (2016) Wind turbine blade breakage monitoring with deep autoencoders. IEEE Trans Smart Grid 9(4):2824–2833

Yu Y, Cao H, Yan X, Wang T, Ge SS (2020) Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor. Neurocomputing 376:1–9

Zhao J, Qu J (2018) Healthy and diseased tomatoes detection based on yolov2. Int Conf Hum Center Comput 2018:347–353

Zhou R, Achanta R, Süsstrunk S (2018) Deep residual network for joint demosaicing and super-resolution. In: Color and imaging conference, vol 2018, pp 75–80. Society for imaging science and technology