EXPRESS PAPER **Open Access**

Unsupervised anomaly detection with compact deep features for wind turbine blade images taken by a drone

Yinan Wang[1*], Ryota Yoshihashi[1], Rei Kawakami[1], Shaodi You[2], Tohru Harano[3], Masahiko Ito[3], Katsura Komagome[3], Makoto Iida[1] and Takeshi Naemura[1]

Abstract

Detecting anomalies in wind turbine blades from aerial images taken by drones can reduce the costs of periodic inspections. Deep learning is useful for image recognition, but it requires large amounts of data to be collected on rare abnormalities. In this paper, we propose a method to distinguish normal and abnormal parts of a blade by combining one-class support vector machine, an unsupervised learning method, with deep features learned from a generic image dataset. The images taken by a drone are subsampled, projected to the feature space, and compressed by using principle component analysis (PCA) to make them learnable. Experiments show that features in the lower layers of deep nets are useful for detecting anomalies in blade images.

Keywords: Wind turbine blade, Drone, Anomaly detection, Unsupervised learning

1 Introduction

Wind power is a widespread renewable form of energy, and its scale has gradually enlarged for the sake of power generation efficiency. However, accidents involving the blades of the turbines, the causes of which are abrasion, metal fatigue, and lightning strikes, are a serious concern after the installation [1]. While thorough periodic safety checks and maintenance are effective, these measures require many experts and the loss of power caused by shutting down the wind turbine is burdensome [2]. Compared with high-place work such as rope work, blade inspection by an automated drone requires less labor and the inspection time can be dramatically reduced. The recorded images are also useful in the long term for various purposes. However, the judgment of whether an image shows evidence of damage still depends on visual inspection of the images by trained experts. This involves not only labor but also variation in the judgment criteria depending on the skill levels of the experts and other personal factors. The shortage of such experts for maintenance is also a serious problem.

Here, deep learning would be a promising means of automated inspection owing to its high performance in image classification, detection, segmentation, etc. However, applying it to anomaly detection poses two problems. First, a large amount of data is required for training a deep neural network. Since blades usually work normally, images of blades with abnormalities are difficult to collect. This leads to an imbalance in the training data, making the accuracy of classes with insufficient data unstable. Second, anomalies in blade images such as hairline cracks are faint, and finding good features to represent them is not a trivial problem.

To address these issues, we propose a method that utilizes one-class support vector machines (OCSVM) [3] and features from the middle layer of convolutional neural networks (CNNs) [4], compressed via principal component analysis (PCA). OCSVM can fit a hypersurface to normal data without supervision, and thus, it is a popular method in unsupervised anomaly detection. A CNN [4] trained on a large-scale general image dataset was used to extract the feature spaces, since a CNN that is properly trained on a large-scale dataset such as ILSVRC can be used as a feature extractor for various tasks even if it is not fine-tuned on the target domain data [5].

*Correspondence: wangyn@hc.ic.i.u-tokyo.ac.jp
[1]The University of Tokyo, 7 Chome-3-1 Hongo, Bunkyo, Tokyo, Japan
Full list of author information is available at the end of the article

Furthermore, to make high-dimensional deep features learnable, the dimensionality has to be reduced; we chose PCA for this purpose. Figure 1 shows an overview of the method. Since OCSVM can perform training with only positive samples, it greatly alleviates the cost of data annotation, while generalized image features are available through CNNs.

Our experiments compared the classifications using a deep-feature space against those acquired using a popular hand-crafted feature, namely the histogram of oriented gradients (HOG) [6], and the feature extracted by the convolutional autoencoders (CAEs) [7], both implemented with OCSVM. The deep-feature-based classification significantly improves on the one made with HOG and CAEs, indicating that generic image features are also useful for detecting anomalies in blade images. We also compared the performance of features extracted using different layers of the CNN and found that those from lower layers perform better than those from higher ones, since lower layers are good at expressing anomalies such as cracks.

The contributions of the paper are summarized as follows. First, we show a novel approach towards practical automatic blade inspection using images taken by a drone. Second, we show a method to utilize features acquired through a generic image dataset (crawled from the web) for unsupervised anomaly detection in a substantially different image domain such as the blade surface. We particularly show that the features acquired in lower layers of CNNs trained with ImageNet are also useful for detecting blade anomalies, even when the data are compressed by PCA.

2 Related work

Crack detection is often required when inspecting roads and infrastructure. For this subject, some supervised approaches exist. Cha et al. [8] used CNNs to detect concrete cracks, although their method had to use 20,000 normal and 20,000 abnormal training samples. Although supervised approaches are powerful, they are often not suitable for anomaly detection, because it is difficult to collect labeled data of rare abnormalities.

A few theoretical studies have used deep learning for unsupervised anomaly detection. Autoencoders are one of the popular approaches among them. Autoencoders combined with Gaussian mixtures [9], and the generative adversarial networks [10] are used with minimization of reconstruction error. An optimization that minimizes the volume of hypersphere of the outputs [11] has also been proposed. Those theoretical studies are mostly tested only with simulative anomaly detection settings (e.g., MNIST with one-class-vs-others setting).

Erfani et al. [12] combined a linear one-class SVM with deep belief nets, which produced comparable performance to that of deep autoencoders. Bendale et al. [13] performed novelty detection with networks trained with ImageNet, but the domain of the novelty images was still limited in the web images, and the method's transferability was not discussed.

OCSVM [3] While the regular SVM finds a hyper-plane that separates two classes with the largest margin from support vectors in a feature space, OCSVM is trained only with normal data of one class so that it finds a hyper-surface having the maximum margin differentiating the region having a high density of normal data from that of the origin, as shown in Fig. 1. A kernel trick is usually used so that a non-linear hypersurface can fit the data.

CNN [4] Below, we use VGG-16 [4], which is pre-trained on the ILSVRC2014 ImageNet dataset, as a feature extractor. VGG-16 is a CNN architecture that won the ILSVRC in 2014 [4]. It has 13 convolution layers and three fully
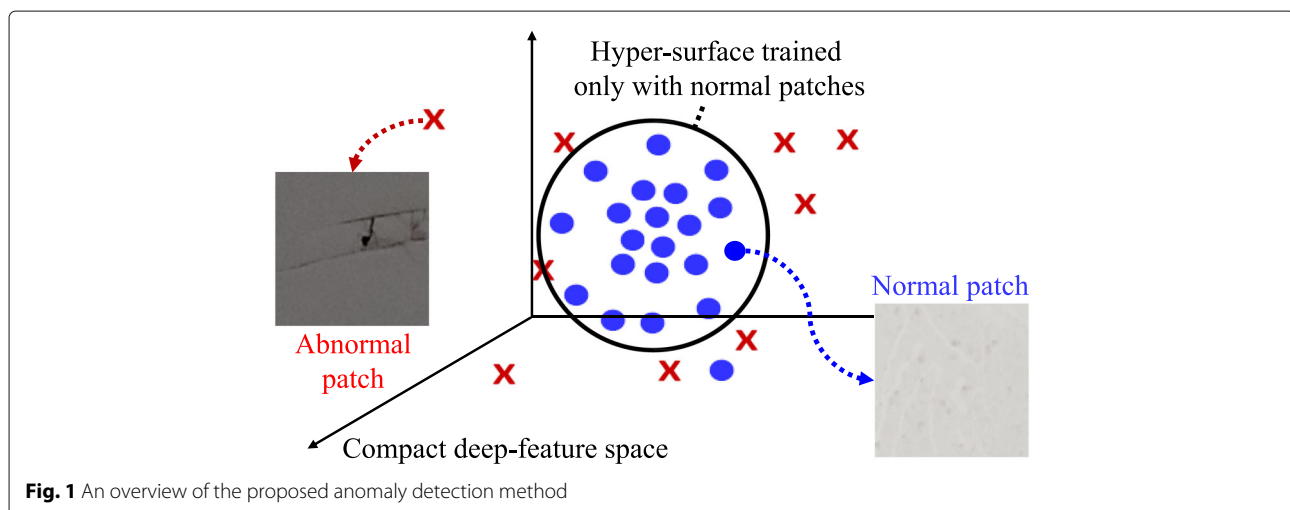

Fig. 1 An overview of the proposed anomaly detection method

connected layers, in total 16 weight layers. All convolutional layers use 3 by 3 kernels, and the pooling layer uses a maximum pooling of 2 by 2. The ReLU activation function is used after each hidden layer.

3   Anomaly detection for blade images

The process starting from acquiring images and ending with outputting a classification score is described below. In the training, images without anomalies are sampled and projected into the deep-feature space, and OCSVM constructs a discriminant hypersurface, the circle shown in Fig. 1, by using these normal data points. In the testing, OCSVM regards data points outside the hypersurface as anomalies.

3.1   Preprocessing of acquired images

Figure 2 shows the procedure of data acquisition and preprocessing of images. The details of the drone and the captured data are provided in Section 4.1. Next, the backgrounds are removed from the captured images. Because the wind turbines are usually situated in windy locations and their blades are fairly high above the ground, it is difficult for the drone to take clear images when it is near the blade; typically, some of the background remains in the image. The gradient between the background and the blade is large, and although it is reflected in the features, such changes in features are inappropriate for detecting anomalies. Furthermore, the background of the test data and training data are likely to have different distributions which can be easily detected as anomalies. Although there are many semantic segmentation algorithms that can automatically remove the background, most of them require a large number of training data with pixel-level labels. In this paper, we manually chose regions without any background.

Second, we need to divide the image into small patches. Here, a high-resolution image is needed to detect small

defects in the blade. However, a high-resolution image would produce feature vectors of higher dimension, making the classifier difficult to train. Thus, we divided images into small patches covering 128 by 128 pixels, to be fed into a feature extractor. This is the smallest size in which the anomalies such as cracks are still recognizable to the human eye.

3.2   Feature extraction

The feature extraction procedure is shown in Fig. 3. We used VGG-16 as the feature extractor. While the output of each hidden layer can be used as a feature vector, the earlier layers tend to extract low-level features such as edges and corners. The later layers tend to extract more complex features, and the features extracted by the last few layers are apt to extract information on each class of the training dataset. Therefore, when using pre-trained models without fine-tuning, it is presumably better to use layers closer to the input because of their generalization ability.

The feature vector of each 128-by-128 pixel patch extracted by the first layer of VGG-16 is still as high as 262,144 dimensions. The higher the dimension of the feature vector is, the larger the model of the classifier will be, and a sparse distribution of data in the feature space will make the training of the classifier more difficult. Therefore, it is necessary to compress the feature vector further.

Principal component analysis is a method of dimensional compression under the premise of a minimal loss of information by mapping feature vectors from the original feature space to the low-dimensional space with the largest contribution of the variance. In this paper, we compress the 262,144-dimension feature vector to 8,000 dimensions. In addition, we re-scale the feature vector by using min-max normalization, so that it is easier for OCSVM to fit the training data. Each feature value is re-scaled using the following equation.
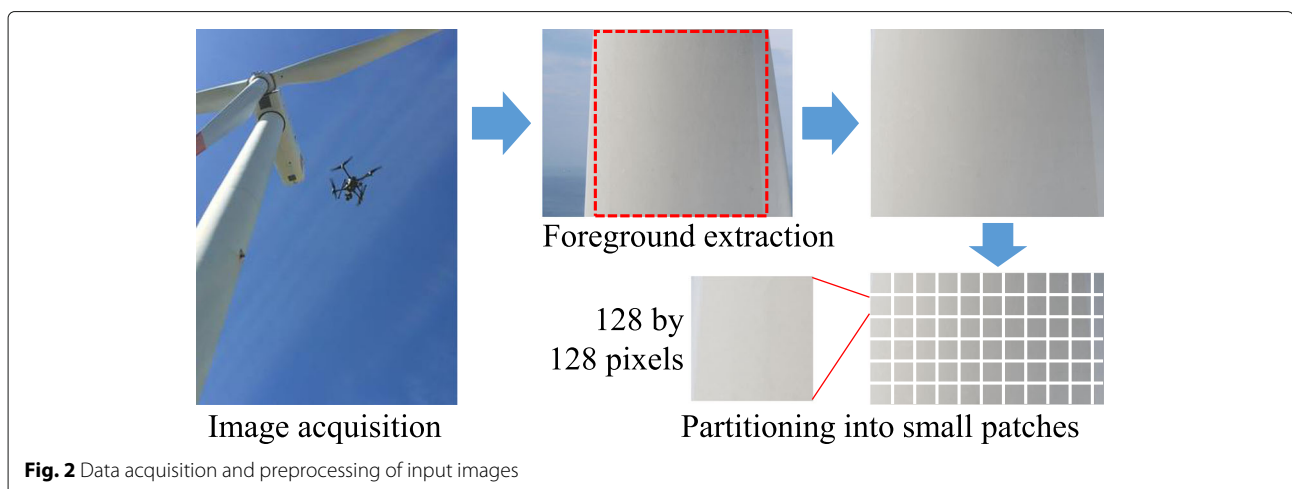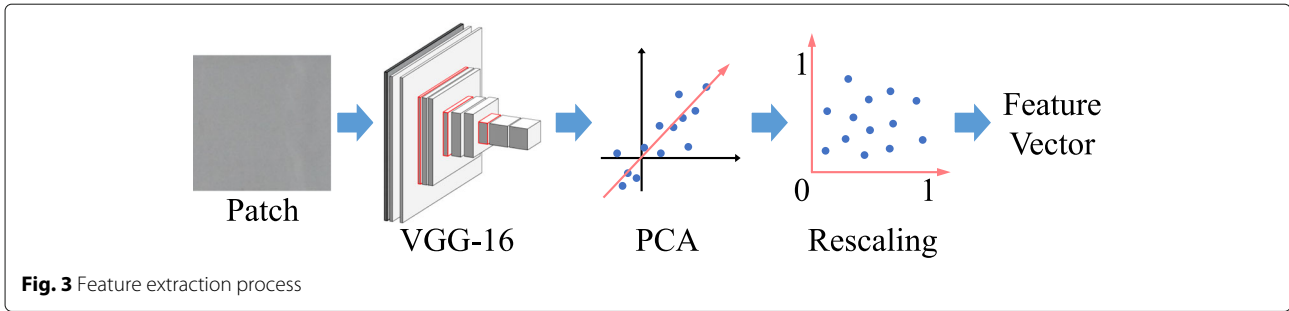

Fig. 2 Data acquisition and preprocessing of input images

Fig. 3 Feature extraction process

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}. \tag{1}$$

3.3 Parameters of OCSVM
We used the Gaussian RBF (radial basis function) kernel for the SVM kernel trick. The test data closer to the origin were classified as abnormal, and those on the opposite side (corresponding to the region inside the circle in Fig. 1) were classified as normal.

The hyper-parameters of OCSVM include ν (nu) and γ (gamma). ν is the ratio of outliers to reduce the influence of outliers in the training data. In the learning, OCSVM will ignore the $100 \times \nu$ percent of the data that is farthest from the center of the training data. γ is a parameter proportional to the reciprocal of the standard deviation of the Gaussian distribution. As γ increases, the discriminant hypersurface fits more snugly to the learning data and the enclosed hypervolume becomes smaller. When γ becomes smaller, the hypersurface becomes smoother and the region of normal data becomes larger.

4 Experiment
4.1 Data preparation and evaluation metrics
The images of the pressure side and suction side of the wind turbine blades were captured using a MATRICE 210, a drone manufactured by DJI Corporation and equipped with a 45-mm lens. We manually operated the drone during the data acquisition and took images after focusing on the blade in a static state. To ensure no anomalies to be missed during image capture, we made consecutive images have large overlaps. The resolution of the images was 5280 × 2970.

The training data contained 130 images of blades with no damage, and it was divided into 73,918 patches. The test data contained 30 blade images with known damage, and it was divided into 21,085 patches. We manually screened the patches and labeled 244 patches of test data with damage as abnormal for the evaluation. As a result, the anomalous patches amounted to 1.16% of the total.

We set ν as 0.01, and γ as 0.1, in the experimental implementation of OCSVM. The code was written using python and the scikit-learn library. We used a Xeon E5-2637 v4 (3.5 GHz, 8 core CPU) to train the OCSVM. The training took 3 to 4 h, depending on the data and parameters.

Since the anomalies accounted for only about 1% of the total data, the usual sort of evaluation metric for classification, such as *accuracy*, is not adequate (a classifier that answers *normal* to all data achieves 99% accuracy.) To evaluate the results properly, we defined the patches showing damage as *positive* and normal patches as *negative* and used the precision, recall, and F1 score as evaluation scores, defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{recall} = \frac{\text{TP}}{\text{TP+FN}}, \tag{2}$$

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{3}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

4.2 Compared method
We compared OCSVM in the deep-feature space against the most common conventional image feature, the histogram of oriented gradients (HOG) [6]. The HOG feature is a monochrome gradient-based feature that can represent shapes and textures in an image, and it is good at detecting cracks in planar objects [14]. HOG calculates image gradients for each pixel and makes them into a histogram in a unit of cells, and then normalize the gradients by using a block composed of numerous cells. It is robust to changes in illumination. In the experiment, output dimension of HOG was 8100, and we did not compress it by PCA.

We also compared the feature extracted by the encoders of CAEs with ours. The implemented encoder includes four convolutional layers and two max pooling layers, namely, the first two blocks of the VGG. We borrowed the structure from VGG so that the architectural difference between ours and CAEs becomes as small as possible. On top of them, we added an 8-channel 3 × 3 convolutional layer for dimensionality reduction; thus, the dimension of feature extracted by the encoder was 8192. The decoder mirrors the encoder part, and the autoencoder

was trained to reconstruct the blade images. Through training, it should ideally capture a compressed feature with details of blade images which is helpful for image reconstruction.

4.3 Results

The classification results of anomalies (as positives) using HOG, CAEs, and VGG-16 features with OCSVM are summarized in Table 1. With the first layer of VGG-16, OCSVM detected 192 of the 21,085 test patches as abnormal. Among them, 121 were true abnormal patches and 71 were normal patches. Namely, 49.6% of the total abnormal patches were correctly detected, and 63.0% were indeed abnormal among the detected patches. For the normal patches as positives, precision and recall were respectively 99.4% and 99.7%. The values for the VGG features were significantly better than those produced by the HOG features. This may be because the defects in the blade cannot be represented only by the magnitude and direction of the gradients and more complicated features such as deep features properly express the difference between abnormal and normal. Besides, the max pooling in the VGG networks reduced the effect of noise and deformation to features, thereby improving robustness. The CAE features produced the lowest F1 score due to low precision, because it classified some overexposure patches as abnormal and was not able to detect hairline cracks. Regarding the VGG features, the first layer outperformed the third layer. The difference was especially noticeable in the recall value. The low-level features seem to be better than higher level ones at reflecting defects in the blades.

Figure 4 a, b, and c show examples of correctly detected, missed, and miss-detected images. Observing TP in (a), one can see that both deep cracks and shallow hairline cracks can be detected. This indicates that the VGG feature is effective for representing cracks in blade images, and it is preserved even compressed via the PCA. However, in (b), the hairline cracks and cracks with smaller contrast with background tend to be missed, especially near the boundary of patches. The training data included patches with painted edges and/or dirt on the blades, but since the edge direction and dirt patterns are not generally aligned, there would likely be a difference between the
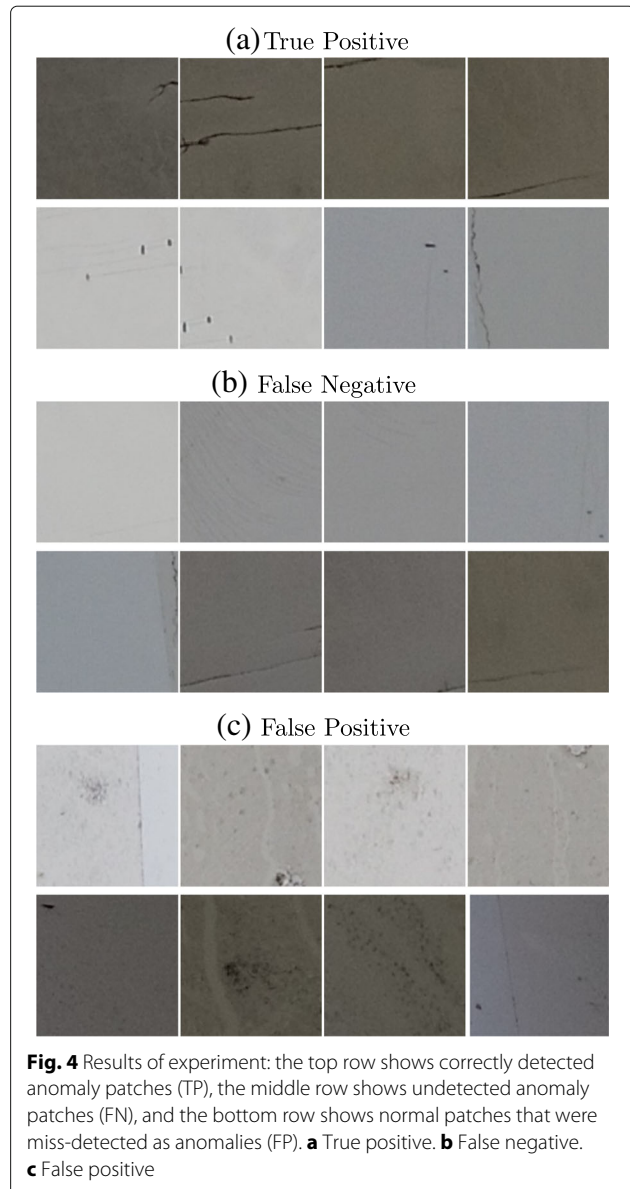

Fig. 4 Results of experiment: the top row shows correctly detected anomaly patches (TP), the middle row shows undetected anomaly patches (FN), and the bottom row shows normal patches that were miss-detected as anomalies (FP). **a** True positive. **b** False negative. **c** False positive

feature vectors. Looking closely at the distance of those patches to the discriminant surface of the learned data, we can see that they are ignored as outliers in OCSVM; thus, it is difficult to correctly detect them with the current approach.

The classifier discriminates on the basis of whether the distance between the data and the discriminant hypersurface is larger than the threshold value. A precision-recall curve can be drawn by changing this threshold. As shown in Fig. 5, when the recall is very low, the precision does not increase. This means that there are several normal patches that are distant from the discriminant hypersurface (ignored by *nu* during training) and mixed with abnormal patches. We show such examples in Fig. 4c. The
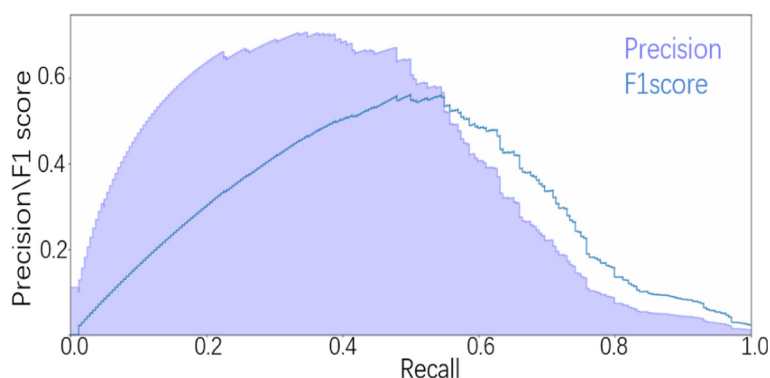
Table 1 Classification results of anomalies with CAEs, HOG, and VGG features

Feature	Precision	Recall	F1 score
CAEs	0.101	0.344	0.156
HOG	0.164	0.263	0.202
VGG16 layer3 +PCA	0.546	0.224	0.318
VGG16 layer1 +PCA	*0.630*	*0.496*	*0.555*

Fig. 5 Precision-recall curve of the classification result with anomaly as positive

images with the stain tend to be detected as abnormal more often than the images with the thin hairline crack.

5 Conclusion

We presented a method to detect anomalies in blade images by running OCSVM in a compact deep-feature space via PCA. Despite the use of unsupervised learning, the proposed combination of CNN, PCA, and OCSVM achieved 0.55 in F1 score. Furthermore, we showed that the lower layer of deep features are more useful than the higher one at detecting anomalies in blade images. It is known that the deep features perform better than HOG because of the ability to extract context information; however, the deep features still defeated HOG when detecting the basic shape such as cracks even without fine-tuning. The limitation of the method is that it may wrongly detect conspicuous dirt, stains, and patterns of painted lines. In future work, we will collect and annotate more abnormal images of blades and examine methods combining anomaly detection and supervised learning based on CNNs.

Availability of data and materials
The data that support the findings of this study were provided from Eco Power Co., Ltd. Restrictions are applied to the availability of these data. The data are however available from the authors upon reasonable request and with permission of Eco Power Co., Ltd.

Authors' contributions
YW, RY, and RK designed and implemented the proposed detection method and drafted the manuscript. TH, MIto, and KK acquisite the data. TN, MIida, and SY edited the manuscript. RK, TN, MIida, and SY supervised the work. All authors reviewed and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

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

Author details
[1]The University of Tokyo, 7 Chome-3-1 Hongo, Bunkyo, Tokyo, Japan. [2]Data61-CSIRO, Canberra, Australia. [3]Eco Power Co., Ltd., TOC Osaki Bldg.1, 1-6-1 Osaki, Shinagawa-ku, Tokyo, Japan.

References

1. Chia CC, Lee J-R, Bang H-J (2008) Structural health monitoring for a wind turbine system: a review of damage detection method. Meas Sci Technol 19:122001. https://doi.org/10.1088/0957-0233/19/12/122001
2. Lu B, Li Y, Wu X, Yang Z (2009) A review of recent advances in wind turbine condition monitoring and fault diagnosis. pp 1–7. https://doi.org/10.1109/PEMWA.2009.5208325
3. Schölkopf B, Platt J, Shawe-Taylor J, Smola A, C. Williamson R (2001) Estimating support of a high-dimensional distribution. Neural Comput 13:1443–1471. https://doi.org/10.1162/089976601750264965
4. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556
5. Razavian A, Azizpour H, Sullivan J, Carlsson S (2014) CNN features off-the-shelf: an astounding baseline for recognition, vol. 1403. https://doi.org/10.1109/CVPRW.2014.131
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05. IEEE Computer Society, Washington. pp 886–893. https://doi.org/10.1109/CVPR.2005.177
7. Masci J, Meier U, Cireşan D, Schmidhuber J (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela T, Duch W, Girolami M, Kaski S (eds). Artificial neural networks and machine learning – ICANN 2011. Springer, Berlin. pp 52–59
8. Cha Y-J, Choi W, Buyukozturk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng 32:361–378. https://doi.org/10.1111/mice.12263
9. Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, Chen H (2018) Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations. https://openreview.net/forum?id=BJJLHbb0-. Accessed 09 May 2019
10. Schlegl T, Seeböck P, Waldstein S, Schmidt-Erfurth U, Langs G (2017) Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. pp 146–157. https://doi.org/10.1007/978-3-319-59050-9_12

11. Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: Dy J, Krause A (eds). Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80. PMLR, Stockholmsmässan. pp 4393–4402. http://proceedings.mlr.press/v80/ruff18a.html. Accessed 09 May 2019

12. Erfani S, Rajasegarar S, Karunasekera S, Leckie C (2016) High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognit 58. https://doi.org/10.1016/j.patcog.2016.03.028

13. Bendale A, Boult T (2016) Towards open set deep networks. pp 1563–1572. https://doi.org/10.1109/CVPR.2016.173

14. Meng L, Wang Z, Fujikawa Y, Oyanagi S (2015) Detecting cracks on a concrete surface using histogram of oriented gradients. pp 103–107. https://doi.org/10.1109/ICAMechS.2015.7287137