Precision slicing for enhanced defect detection in high-resolution wind turbine blade imagery

Serhii Svystun[1,*,†], Oleksandr Melnychenko[1,†] Pavlo Radiuk[1,†], Oleg Savenko[1,†] and Anatoliy Sachenko[2,3,†]

[1] *Khmelnytskyi National University, 11, Institutes str., Khmelnytskyi, 29016, Ukraine*
[2] *Kazimierz Pulaski University of Technology and Humanities, Department of Informatics, Radom, Poland*
[3] *Research Institute for Intelligent Computer Systems, West Ukrainian National University, Ternopil, Ukraine*

Abstract

The analysis of high-resolution aerial imagery captured by unmanned aerial vehicles (UAVs) presents significant analytical challenges, primarily due to the minuscule size of observable objects and the variability in object scale influenced by UAV altitude and positioning. These factors often lead to diminished data fidelity and complicate the detection of smaller objects, which are critical in applications such as infrastructure monitoring. Traditional image processing techniques, which typically segment images into smaller, randomly cropped sections before analysis, must sufficiently address these challenges. In this work, we propose a novel defect detection framework for identifying minor to medium-sized damages on wind turbine blades (WTBs), a critical component in renewable energy production. The proposed framework, termed 'slice-aided inference,' enhances the existing methodologies by incorporating both traditional patch division and a novel, more advanced technique known as slice-aided hyper-inference. These techniques are rigorously assessed with various advanced deep learning models, emphasizing their efficiency in identifying surface defects. The empirical testing conducted as part of this study demonstrates significant enhancements in detection capabilities, leveraging a dataset of high-resolution UAV images to highlight the practical applications and effectiveness of the proposed framework in real-world scenarios.

Keywords

Aerial imagery, drone imaging, defect detection, WTBs, slice-aided inference, hyper-inference, deep neural networks, image segmentation, high-resolution imaging, and object detection

1. Introduction

Renewable energy sources are increasingly recognized for their substantial benefits and are not just solutions for individual nations but global imperatives in the face of climate change. Unlike fossil fuels, they significantly mitigate CO2 emissions, which is crucial in the international strategy to curb anthropogenic global warming [1, 2]. Deploying renewable energy bolsters national energy security and diminishes dependency on imported fossil fuels [3]. The localized

ICyberPhyS-2024: 1st International Workshop on Intelligent & CyberPhysical Systems, June 28, 2024, Khmelnytskyi, Ukraine

[*] Corresponding author.
[†] These authors contributed equally.

svystuns@khmnu.edu.ua (S. Svystun); melnychenko@khmnu.edu.ua (O. Melnychenko); radiukp@khmnu.edu.ua (P. Radiuk); savenko_oleg_st@ukr.net (O. Savenko); as@wunu.edu.ua (A. Sachenko)
0009-0009-8210-6450 (S. Svystun); 0000-0001-8565-7092 (O. Melnychenko); 0000-0003-3609-112X (P. Radiuk); 0000-0002-4104-745X (O. Savenko); 0000-0002-0907-3682 (A. Sachenko)

availability of renewable resources minimizes the risks associated with supply disruptions and price volatility, promoting energy independence [1]. As a result, there is an escalating trend among countries to invest in renewable energy infrastructures, including wind turbines and solar panels.

Wind turbines convert kinetic wind energy into electrical power, reducing greenhouse gas emissions and enhancing energy supply diversity, security, and sustainability. These turbines operate by harnessing the motion of wind to generate electricity without the need for combustion processes [4]. The efficiency and output of wind turbines are significantly influenced by the condition of their blades, which are integral to maximizing energy capture [5]. Research suggests that the blades are accountable for up to 25% of a turbine's total energy production. Maintaining these blades in optimal condition is imperative for maximizing energy generation and minimizing operational downtime and associated costs [6]. Historically, the inspection of wind turbines, especially offshore ones, has been predominantly manual and labor-intensive, contributing to elevated maintenance costs and extended operational delays. Recent advancements in sensor technologies, including those for acoustic, vibration, ultrasonic, and strain measurements, have substantially improved wind turbines' maintenance and condition monitoring [7–9]. The integration of visual sensors, which facilitate detailed examinations of turbine surfaces, is expected to yield significant benefits in the maintenance regimes of these critical energy assets.

There is a growing need for safer and more efficient methods of inspecting WTBs to enhance cost-effectiveness and efficiency [10]. The ideal solution would balance sensors' reliability, accuracy, and affordability. By implementing an effective and economical approach, wind farm operators can optimize energy production while cutting maintenance expenses. Technological advancements, especially in UAVs, are at the forefront in developed countries for various applications [11, 12].

UAVs have proven to be highly effective tools for aerial inspections, particularly in assessing WTBs [13]. By utilizing UAV-based inspection methodologies, practitioners have achieved an impressive throughput of 10–12 daily turbines, which could increase to 15–20 with complete automation [13]. These advancements surpass conventional inspection methods and hold significant promise for improving inspection efficacy, reducing expenses, and enhancing energy generation with minimal operational disruption, all of which are crucial in the energy sector.

The automated scrutiny of energy infrastructure, notably WTBs, can gain substantially from the progression in drone technology and remote image surveillance, engendering cost efficiencies and fostering climate change mitigation endeavors and safety protocols. Despite strides made in deep learning paradigms, persistent challenges in object detection persist, stemming from factors such as diminished image resolution, occlusions, intricate backgrounds, and the diminutive dimensions of target objects. The essence of deep learning methodologies lies in the iterative processes of training and inference, whereby models are honed to discern anomalies through optimization methodologies applied to datasets replete with instances of defects [14]. Subsequently, once trained, these models can identify deviations in new imagery by extrapolating learned patterns. The overarching significance of meticulous dataset curation, judicious selection of architectural frameworks, and optimization of these facets underscores the imperative of achieving precise outcomes.

Introducing high-resolution technology in aerial imaging has significantly enhanced our ability to capture intricate details from a high vantage point [15, 16]. However, this advancement also brings its own set of challenges. Depending on the UAV's proximity to the subject, the varying scales of objects within these images can make it difficult to discern small entities as the UAV moves away from its focal point. Additionally, the large amount of background information in high-resolution imagery can challenge effective computational processing.

Within the purview of deep learning detection methodologies, exemplified by convolutional neural networks (CNNs), these challenges manifest as impediments to optimal classifier training, frequently culminating in compromised detection accuracy [17]. The emergence of high-resolution data streams, typified by high-definition (HD) 4K, necessitates the innovation of novel analytical techniques to navigate these intricacies.

Figure 1 visually represents the multifaceted challenges encountered in inspecting surface defects on WTBs using drone technology.

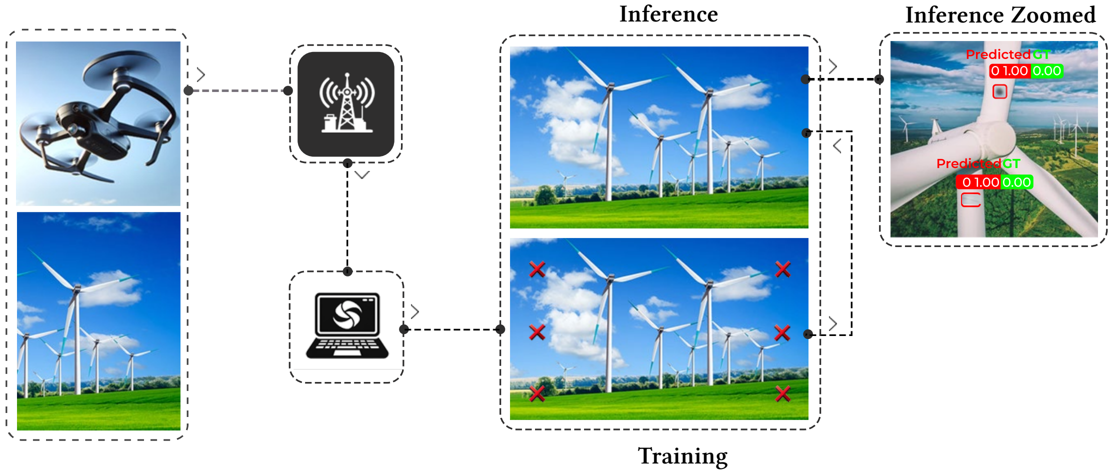

Figure 1: This scheme illustrates the utilization of drones for the surveillance of WTB surfaces. In transmitting image data to servers facilitated by 4G/5G technology, defect detection models frequently encounter the challenge of processing high-resolution images. In our proposed framework, training is conducted on image patches, with a meticulous pre-processing step that involves discarding patches lacking relevant content. Subsequently, during the inference stage, implementing a robust strategy becomes imperative to detect minor defects.

The scheme in Figure 1 underscores the intricacies inherent in high-resolution images, accentuating the complexities inherent in this domain.

The rest of the paper is structured as follows. Related works review existing methodologies and similar approaches. The suggested architecture section details the proposed defect detection framework, covers the DTU-Drones dataset and annotation process, and outlines training and inference strategies, emphasizing slice-aided hyper-inference. The results section presents experimental outcomes and performance analysis. The discussion highlights comparative analysis and practical implications. Finally, the conclusion summarizes findings, improvements in defect detection, and future research directions, followed by references.

2. Related works

The analysis of defects on WTBs encompasses a spectrum of methodologies, ranging from conventional techniques rooted in image processing to contemporary approaches leveraging hand-crafted features. Wang and Zhang, for instance, deployed Haar-like features in tandem with cascaded classifiers to discern surface cracks on WTBs, with a principal emphasis on distinguishing cracked regions from non-cracked ones [18]. Similarly, Huang and Wang extended the utility of Haar-like features, integrating them with the parallel Jaya K-means algorithm to achieve enhanced precision in surface crack detection [19]. Deng and Guo devised a novel strategy by amalgamating an optimized Lévy flight strategy combined with the log-Gabor filter for defect identification [20].

To efficiently detect large-scale cracks on WTBs, Peng et al. [21] proposed an analytical framework harnessing UAV-captured imagery. Alternatively, Ruiz and Magda [22] adopted a distinct approach by converting operational signals from wind turbines into grayscale representations, subsequently leveraging multichannel texture features for pattern recognition.

The advent of deep learning methodologies has catalyzed a significant paradigm shift within the realm of WTB defect detection. Shihavuddin et al. [23] spearheaded the introduction of a feature pyramid network augmented with offline data augmentation techniques tailored specifically for processing higher-resolution imagery. Their methodology involved training diverse Faster-RCNN detectors on heterogeneous private datasets, yielding promising outcomes. Subsequent explorations delved into the efficacy of YOLO models and EfficientDet, further substantiating the potential of deep learning frameworks in this domain.

The superiority of CNNs over traditional descriptors has been underscored, particularly with the added advantage of ensemble classifiers. Foster et al. [17] contributed to this discourse by categorizing WTB defects through the utilization of image patches for both training and inference tasks. Yu et al. [24] addressed challenges associated with blurry imagery by deploying a super-resolution CNN model complemented by Laplacian variance pre-processing techniques. Remarkable advancements in detection performance for WTB defects have also been ascribed to deploying other deep learning architectures.

Collectively, these studies illuminate the pivotal role of deep learning methodologies in augmenting defect detection capabilities in WTB inspection.

The processing of high-resolution drone-captured images poses substantial challenges due to the variability in object scales and the requisite computational resources [25]. To mitigate these challenges, a prevalent approach involves partitioning these images into smaller patches, a practice endorsed by several studies in the field [13, 19]. This strategy alleviates computational burdens, enhances object clarity, and augments dataset dimensions, fortifying model performance.

Despite the substantial body of literature dedicated to detecting WTB defects, a notable discrepancy endures in the methodologies and classification schemes employed across these studies, a gap that our research aims to address. Benchmarking the performance of WTB surface defect detectors proves particularly challenging due to the confidentiality of data [24] and the absence of annotations, even in cases where data accessibility is feasible [17, 23]. Our work strives to establish a unified approach and set a benchmark for future studies in this area.

Utilizing drones for detecting surface defects on wind turbine blades (WTB) has proven to be a cost-effective and efficient method, supported by prior research findings. However, this inspection technique comes with its own set of challenges, such as processing high-resolution images, detecting small objects, and adjusting for changes in object scale due to variations in the drone's position.

This study aims to improve accuracy of defect detection in renewable energy assets using imagery captured by UAVs. To accomplish the stated objective, this paper intends to make the following distinctive contributions:

- We present a defect detection framework that integrates a realistic slice-based inference strategy for object detection in high-resolution images.
- We conduct a benchmark comparison of our framework against several state-of-the-art deep learning detection baselines and slicing strategies, tailored specifically for inspecting wind turbine blades.
- We perform an extensive evaluation using a high-resolution drone image dataset, showcasing significant improvements in detecting minor and medium-sized defects on wind turbine blades.

3. Suggested architecture

Figure 2 depicts our proposed framework for WTB surface defect detection. During the pre-processing step, high-resolution images undergo partitioning into patches, which are subsequently incorporated into the training process. This framework aims to optimize defect detection performance by leveraging patch-based and full-resolution inference strategies, thereby addressing the challenges associated with high-resolution imagery in WTB inspection.

3.1. Dataset

This study uses the DTU-Drones inspection dataset of WTB images sourced from the Technical University of Denmark (DTU). This publicly available dataset, accessible at [26], comprises 589 high-resolution images captured between 2017 and 2018 across diverse environmental conditions. Notably, while high-resolution images typically possess dimensions of 1920 × 1080 pixels, the images within this dataset boast 5280 × 2890 pixels, unequivocally classifying them as high resolution.

Given the absence of surface defect bounding boxes in the DTU dataset, we embarked on an annotation process after an exhaustive analysis of various WTB surface defects prevalent in the dataset. It's worth noting that several recent studies have utilized the current dataset for similar purposes [17, 23]. However, these studies could have increased their value by making their annotations publicly accessible for broader utilization and by providing more consistent interpretations of surface defect types.

3.2. Surface defects on wind turbine blades

Identifying defects required a comprehensive review of the existing literature concerning the detection of surface defects in wind turbine blades WTBs. Utilizing publicly available datasets (Section 3.1) and conducting a comprehensive literature search, we systematically categorized various surface defects for our study.

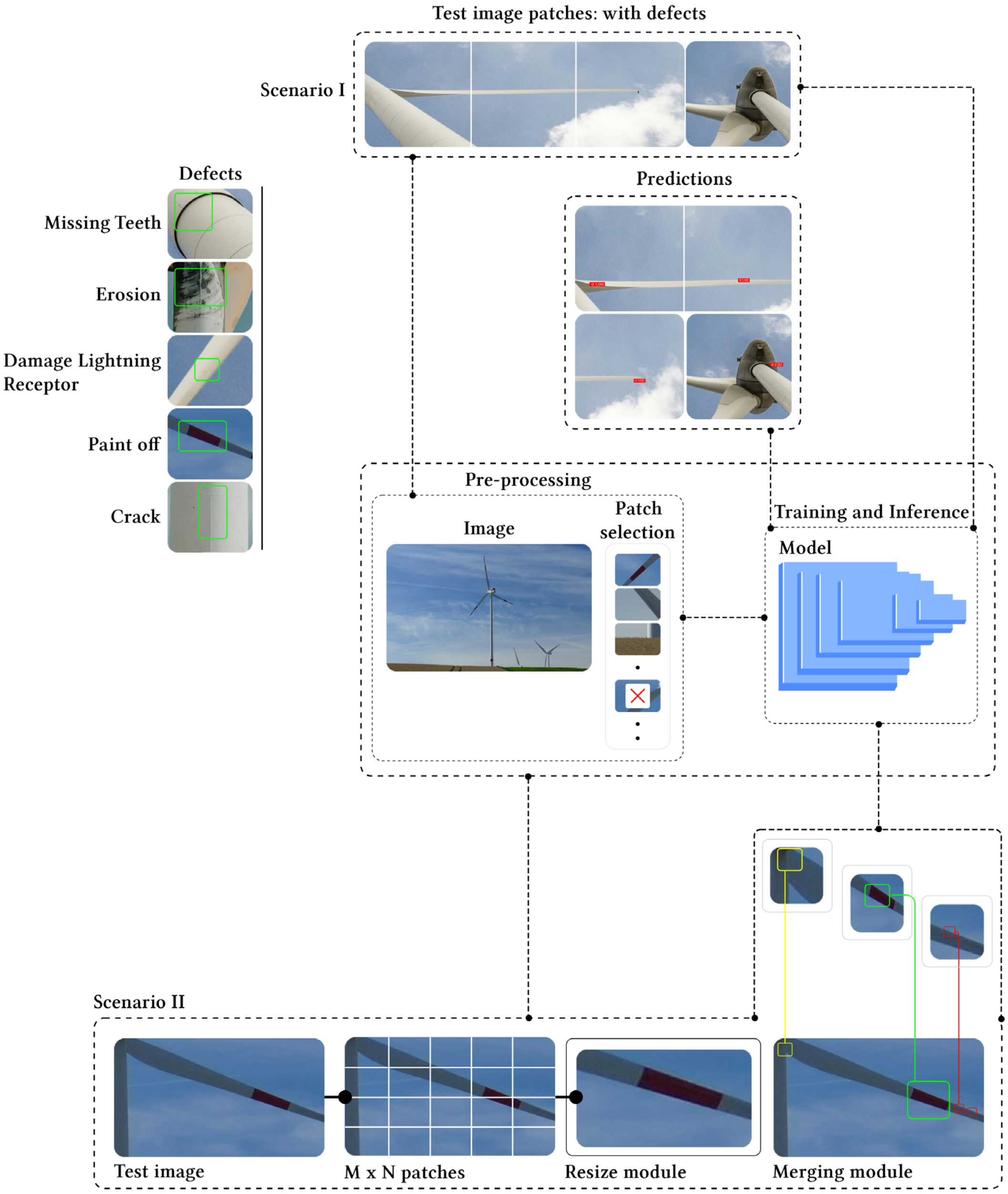

Figure 2: The left portion of the figure showcases different surface defect types from the DTU dataset. On the right, the workflow is depicted, beginning with dataset pre-processing and training, as shown by the black arrows. Two distinct inference scenarios are demonstrated: Scenario I, marked by red arrows, employs image patches for inference, while Scenario II, indicated by blue arrows, uses the original resolution of the test images for inference.

We carefully selected a total of 314 images, each representing one of five distinct types of surface defects, as detailed in this investigation. These images collectively encompass 879 instances, with multiple defects observed per image. Specifically, we identified the following five categories of defects, as illustrated on the left side of Figure 2.

- Missing Teeth. This defect category involves missing teeth in the vertex-generating panel, a critical component of WTBs. Identifying the presence or absence of these teeth is essential for ensuring optimal blade performance.
- Paint-Off. Paint-off describes the loss or peeling of the protective paint layer on the surface of WTBs. Although not directly harmful, paint-off signals the necessity for maintenance to maintain the blade's structural integrity and extend its lifespan.
- Erosion. Erosion is a form of surface degradation affecting WTBs that experiences gradual wear and tear caused by environmental conditions or extended exposure to natural elements. While erosion may not pose immediate threats, it necessitates regular maintenance to mitigate potential issues.
- Crack. Surface cracks in WTBs are considered critical defects owing to their potential to induce structural instability, ultimately leading to catastrophic failure. Detecting and accurately localizing surface cracks are paramount for facilitating prompt maintenance and averting further structural deterioration.
- Damage Lightning Receptor. The lightning receptor plays a crucial role in protecting the wind turbine blade from lightning strikes. Identifying any surface damage to the lightning receptor is essential for evaluating its effectiveness and ensuring it provides sufficient protection against lightning-related damage.

3.3. Data annotation

The dataset annotation process was conducted with meticulous attention to detail, explicitly targeting the defective segments of WTBs. Our annotation methodology entailed precisely localizing regions of interest corresponding to each defect type.

During the annotation process, significant focus was placed on precisely identifying and delineating specific areas of the wind turbine blades WTBs that exhibited defects such as missing teeth in the vertex-generating panel, erosion, damage to the lightning receptor, cracks, and paint-off. These surface defects were carefully localized within their respective regions, providing a thorough and detailed annotation of the defective segments.

3.4. Pre-processing

Before embarking on the learning phase, a series of pre-processing steps were diligently executed to ensure the manageability of processing and model training for high-resolution images, all while preserving the intricate details inherent in the imagery. Building upon the insights gleaned from previous research, notably [17], we undertook an empirical analysis to systematically assess the impact of various patch sizes on our study. The chosen patch sizes proved compatible with our object detection models and demonstrated a discernible performance advantage over alternative patch sizes.

An automated approach was employed to select image patches containing at least one defect, excluding patches exhibiting only background or devoid of defects from the dataset. Following the patching process, the dataset was partitioned into distinct subsets for training, testing, and validation purposes, facilitating subsequent experimental inquiries. Notably, both online (on-the-fly) and offline augmentation techniques were applied exclusively after utilizing data samples during the model training and inference processes.

3.5. Detection system framework

This section provides a comprehensive exposition of the detection framework tailored to address the nuances of high-resolution images. Illustrated in Figure 2 is the entirety of the proposed training and inference pipeline devised for the detection of (WTB) surface defects from high-resolution images. The delineation of the proposed detection framework unfolds across two distinct phases, namely, training and inference.

Let's $I \in D_r$ represents a high-resolution image in the training partition database $D = \{D_r, D_t\}$, where preprocessing produces non-overlapping image patches.

$$p = \{(p, b) \ \mid \ p \subseteq I, \ p \in \mathrm{R}, K \times K \times 3, \ b \neq \emptyset\}. \tag{1}$$

where b is the set of bounding boxed associated with each image patch p and K = 1024, as established in Section 4.1.

A model $\mathcal{M}$ is trained using image patches that make up the training set.

$$P = \bigcup_{i \in |D_r|} p_i, \tag{2}$$

such that $\mathcal{M} \leftarrow \ \mathrm{train}(\boldsymbol{P})$.

We trained three baseline neural network architectures for evaluation and benchmarking purposes: YOLOv5 and RetinaNet, both known for their efficiency and Faster-RCNN, renowned for their accuracy but requiring additional computational resources. YOLOv5 employs a compact yet practical architecture, featuring a deep CNN backbone with 21 convolutional layers (CSPDarknet21), supplemented with a feature pyramid network (PANet) and multiple detection heads for efficient object detection. The system utilizes anchor boxes and a composite loss function play pivotal roles in its training process, with non-maximum suppression refining results during inference. On the other hand, RetinaNet adopts a one-stage design that emphasizes efficiency and employs anchor boxes for region proposals. It utilizes a backbone CNN (ResNet50) and a Feature Pyramid Network (FPN) to capture multi-scale features crucial for detecting objects of various sizes. Lastly, Faster-RCNN operates as a two-stage model, featuring a Region Proposal Network (RPN) for generating proposals and RoI Align layers for feature extraction from these proposals, utilizing ResNet50 as its backbone. Despite its accuracy, Faster-RCNN may demand more computational resources. These neural network architectures were employed to train surface defect detection models, with images pre-processed as outlined in Section 3.4. All models underwent training using the standard multi-class cross-entropy loss function.

$$\mathcal{L}_{\mathcal{CE}} = -\sum_{i=1}^{C} t \log(s_i). \tag{3}$$

In this formula, t_i and s_i represent the ground truth label and the SoftMax probabilities for the i-th class of C total classes, respectively. The function captures the concept of cross-entropy loss, a common loss function used in classification problems to measure the difference between the true distribution t and the predicted distribution s.

3.6. Inference strategies

In the subsequent phase, the inference process unfolds through two distinct strategies: Scenario I and Scenario II. Each plan is meticulously crafted to assess the proposed method's performance under unique conditions. Figure 3 visually delineates both scenarios, offering a graphical representation of their respective methodologies.

Evaluating the method's performance under these contrasting conditions yields valuable insights into the proposed framework’s practical viability and efficacy.

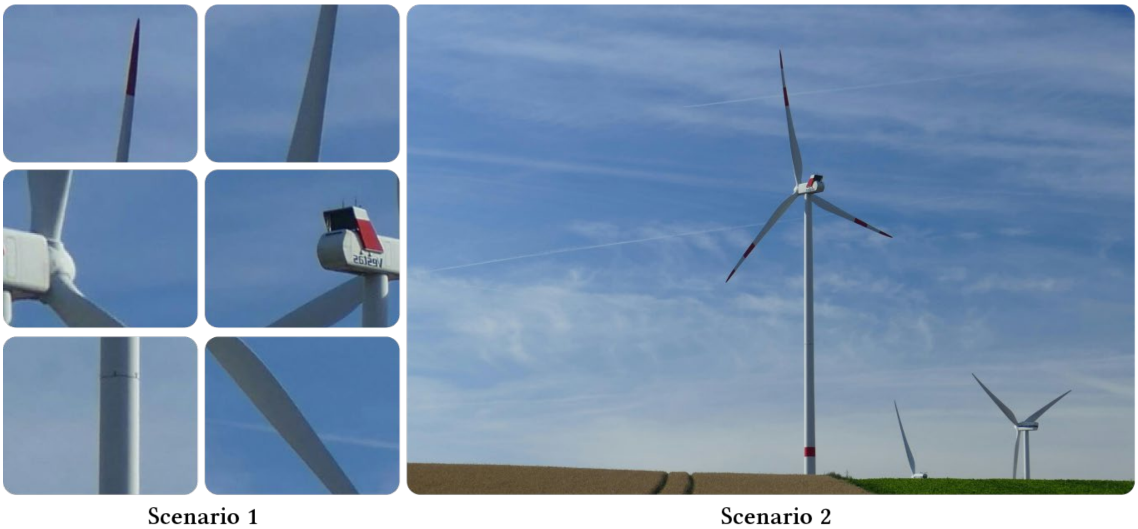

Figure 3: A visual example that showcases the disparity between Scenario I and II, elucidating two distinct evaluation scenarios for the proposed method. In Scenario I, the model undergoes training and testing on pre-processed image patches, whereas in Scenario II, the model is directly tested on raw high-resolution images.

3.6.1. Scenario I: Patch-based inference

Scenario I revolves around constructing the test set by leveraging image patches, as expounded in Section 3.4. Within this framework, individual patches are fed into the trained model for inference. Essentially, Scenario I illustrates a configuration where the model is trained with image patches and evaluated on test patches that, while not identical, maintain the same patch size as those used in training. This setup adheres to the conventional paradigm of machine learning model training. However, it's crucial to acknowledge that Scenario I may segment extended defects, creating separate bounding boxes within distinct image patches, which may not align with practical feasibility. While the patch-based inference process facilitates swift processing, it necessitates additional postprocessing steps for consolidating and identifying corresponding image patches.

3.6.2. Scenario II: Slice-aided inference

In Scenario II, heightened realism is achieved by employing unprocessed high-resolution images for the test set. This circumvents the need for manual pre-processing, instead employing an internal pre-processing mechanism intrinsic to the testing process, as demonstrated in

Equation (4). This configuration offers notable advantages, notably enabling the direct utilization of high-resolution images for prediction and amalgamating multiple detected defects in the original image rather than treating them individually, thereby better reflecting the challenges encountered in real-world scenarios.

To effectively manage the processing of a high-resolution image during inference, particularly within Scenario II, standard resizing or fixed cropping methods are suboptimal for two primary reasons: (1) small objects may become nearly invisible following such transformations, potentially eluding detection; and (2) the precision between overlapping objects may be significantly compromised after resizing the original image.

The text describes the implementation of a computational technique called "slice-aided hyper-inference" for processing high-resolution images, particularly for detecting small and medium-sized objects. The image I' from the dataset Dt, which is part of the test partition, is segmented into $M \times N$ patches, denoted as $p'mn.$ This approach is designed to facilitate the analysis of each segment individually, improving the inference accuracy and efficiency in handling high-resolution data.

To effectively manage the detection of surface defects, mainly to avoid issues with disjointed defects across the boundaries of patches, the method incorporates overlapping sampling of patches using a sliding window technique. This approach ensures that each window samples patches with a defined overlap percentage v between adjacent windows. This overlapping strategy aids in maintaining continuity in the areas of interest across patches, thereby reducing the risk of missing or misidentifying defects that span multiple patches.

After the patches are extracted and overlapped, they are resized to a uniform size of $W \times W$ pixels. This standardization is crucial for consistent processing and analysis, ensuring that all patches are subjected to the same scale regardless of their original dimensions. Finally, the resized patches undergo patch-level model inference. This step involves analyzing each patch using a trained model to predict or identify defects. The inference process on each patch is done independently, allowing for detailed and localized defect detection within the high-resolution image. The inference is carried out as follows:

$$\hat{b}_{mn} \leftarrow \inf\Big(M\big(Z(p'_{mn}, W)\big)\Big) \quad \forall m \in [1, M], n \in [1, N]. \tag{4}$$

The process described involves the detection of defects within images through a series of computational steps. Initially, $\hat{b}_{mn}$ represents the set of output bounding boxes determined by the analysis. The function $Z(\cdot)$ is a resizing module that standardizes the size of each image patch p' to a uniform width W. Which is crucial for consistent processing across different patches.

Once the patches are processed, the bounding boxes with confidence scores exceeding the detection threshold T_{det} are retained for further analysis. These selected boxes are then subjected to Non-Maximum Suppression (NMS), a method used to eliminate redundant bounding boxes. NMS is applied based on the Intersection over Union (IoU) metric, where boxes overlapping more than a predefined threshold T_{mns} are consolidated. This step ensures that the final set of bounding boxes is non-redundant, optimizing the clarity and accuracy of the defect detection process.

4. Results

Experiments utilized the DTU Images dataset for evaluation. Subsequent sections detail the outcomes corresponding to the two scenarios outlined in Section 3.6. These results include the class-wise mean average precision and comparisons among small, medium, and large objects across both scenarios.

4.1. Evaluation details

To assess the efficacy of two distinct methodologies, the dataset was divided into three parts: training, validation, and testing, with proportions of 70%, 15%, and 15%. In the first scenario (Scenario I), we divided the original high-resolution images into patches measuring 1024 pixels on each side, as detailed in Table 1.

Table 1
The image depicts a comparison of patch sizes employed for training and inference in experiments where YOLOv5 serves as the foundational model, highlighting the variations and effects on model performance. In Scenario I, the table shows the results from the validation set, where 'K' denotes the patch resolution, and the count of samples correlates with the training dataset used. Scenario II details 'W' as the window size with the validation set outcomes influenced by parameters V = 0.1, T_{mns} = 0.5, and T_{det} = 0.001.

K	Images	Labels	mAP@0.5	mAP @0.5-0.95	W	Patches	mAP @0.5	mAP @0.5-0.95
640	878	1045	78.7	44.2	511	83	81.4	44.4
800	756	943	79.4	42.5	799	39	82.1	44.7
1024	588	786	84.5	45.9	1023	23	81.9	44.6
2048	352	656	84.4	47.3	2047	5	80.6	43.8

This patch size was chosen to facilitate manageable training under resource constraints while preserving the image's intricate details, as discussed in Section 3.4. In the second scenario (Scenario II), through detailed experimentation, it was found that a patch width of 800 pixels (W) yielded the best results, as indicated in Table 1.

4.2. Performance metric

To assess the effectiveness of our models, we evaluated their performance on the test partition by calculating the mean average precision (mAP). This metric was examined at the commonly used 0.5 IoU threshold (expressed as mAP@.50) typical in object detection analyses and through a more detailed measure spanning a range from 0.5 to 0.95 IoU thresholds, in increments of 0.05 (expressed as mAP@.5-.95). In alignment with the criteria set by the COCO challenge, we analyzed performance specifically for small, medium, and large objects. For this purpose, mAP@.5-.95s was calculated for small objects (area < 322), mAP@.5-.95m for medium objects (322 < area < 962), and mAP@.5-.95l for large objects (area > 962).

4.3. Training configurations

In our experiments, training was performed with a batch size of 8, using the Stochastic Gradient Descent (SGD) optimizer. Learning rates were set according to standard practices: for Faster-RCNN and RetinaNet, and for YOLOv5. Baseline models were sourced from the Detectron2 and Ultralytics repositories. All experiments were conducted on a system equipped with an Intel i5 processor and a single NVIDIA RTX 4060 GPU.

4.4. Experimental results

4.4.1. Overall results

In Table 2, the YOLOv5 model demonstrates stable performance, with a maximum variation of 1.7 points observed in the "large" object size category.

Table 2
mAP@.5-.95 for Scenario I and Scenario II for Small, Medium, and Large objects using the DTU test set.

Models	Scenario I – Small	Scenario I Medium	Scenario I – Large	Scenario II – Small	Scenario II-Medium	Scenario II – Large	Δ(Small)	Δ(Medium)	Δ(Large)
YOLOv5	28.3	48.5	52.4	26.2	49.1	54.1	-2.1	0.7	1.8
Faster-RCNN	17	43.3	64.8	30.2	49.5	51.5	14.2	7.2	-13.3
RetinaNet	14.3	37.7	66.6	19.1	44.5	51.1	5.8	7.8	-16.5

In contrast, Faster-RCNN displays more significant fluctuations, especially achieving notable improvements for "small" and "medium" objects by 14.2 and 7.2 points, respectively. However, it also shows a marked decrease of 14.3 points for "large" objects, indicating its sensitivity to scenario changes and significant variability across different object sizes. RetinaNet also shows improved performance for "small" and "medium" objects, with increases of 5.8 and 7.8 points, but experiences a drop of 15.5 points for "large" objects, mirroring the trend seen in Faster-RCNN. Overall, it is clear from the comparison of detection performance across small, medium, and large objects in Scenarios I and II that the proposed framework in Scenario II significantly enhances the performance of smaller objects when using the Faster-RCNN and RetinaNet models.

Table 3 delineates the comparative effectiveness of baseline models across Scenarios I and II using two key metrics.

Table 3
Overall, mAP@.5 and mAP@.5-.95 for Scenario I and II on the DTU test set.

Models	mAP@.50		mAP@.5-.95	
	Scenario-I	Scenario-II	Scenario-I	Scenario-II
YOLOv5	82.3	86.1	42.7	45.2
Faster-RCNN	74.2	84.4	38.8	44.1
RetinaNet	71.6	71.4	33.9	38.9

In Scenario I, YOLOv5 achieves a mAP@.50 of 81.3, which increases to 85.1 in Scenario II. Similarly, Faster-RCNN exhibits a rise from 73.2 to 83.4 when moving from Scenario I to Scenario II. In contrast, RetinaNet records a slight decrease in mAP@.50 from 70.6 to 70.4. For the mAP@.5-.95 metric, all models show enhancements; YOLOv5 improves from 41.7 to 44.2, Faster-RCNN from 37.8 to 43.1, and RetinaNet from 32.9 to 37.9 when transitioning from Scenario I to Scenario II. The results indicate that RetinaNet, along with the other models, benefits significantly under the optimized conditions of Scenario II, particularly in detecting small and medium objects within high-resolution imagery captured by drones. This suggests that Scenario II surpasses the typical configurations in previous studies, enhancing overall detection capabilities.

4.4.2. Class-wise results

Table 4 presents additional observations.

Table 4
Class-wise mAP@.5-.95 for Scenario I and Scenario II on the DTU test set.

Classes	YOLOv5 Scenario I	YOLOv5 Scenario II	Faster-RCNN Scenario I	Faster-RCNN Scenario II	RetinaNet Scenario I	RetinaNet Scenario II
ER	44.7	45	39.9	48.2	37.6	47.9
DA	39.7	31.5	26.7	31.2	21.7	21.4
CR	28	54.9	31.7	49.8	24.7	31.2
PO	58.8	57.4	56.6	54.2	53.3	59
MT	42.4	38.1	38.7	37.3	32	35.1

From Table 4, in the case of YOLOv5 under our proposed framework (Scenario II), there is a twofold improvement in the performance for the CR class, although decreases are observed in the MT and DA.

But for Faster-RCNN, significant enhancements are noted in the ER, DA, and CR classes with the adoption of our framework. Conversely, RetinaNet shows performance gains in all classes except for the DA class. It's noted that samples in the DA class usually represent minor defects on the WTB surface, which perform well under the slice-aided setup.

Figure 4 provides a graphical comparison that highlights class-specific performance differences between Scenario I and Scenario II for the evaluated models.

Notably, YOLOv5 consistently improves, particularly in the CR class within Scenario II. In contrast, Faster-RCNN demonstrates enhanced performance across three distinct classes under Scenario II conditions, indicating increased reliability. Furthermore, RetinaNet exhibits exceptional results in Scenario II for all classes, which can largely be attributed to the implementing of the focal loss function. This function effectively tackles class imbalances within the dataset.

In addition, Figure 5 presents the precision-recall curve for the DTU test set, highlighting the effectiveness of our method in detecting surface defects on WTBs.

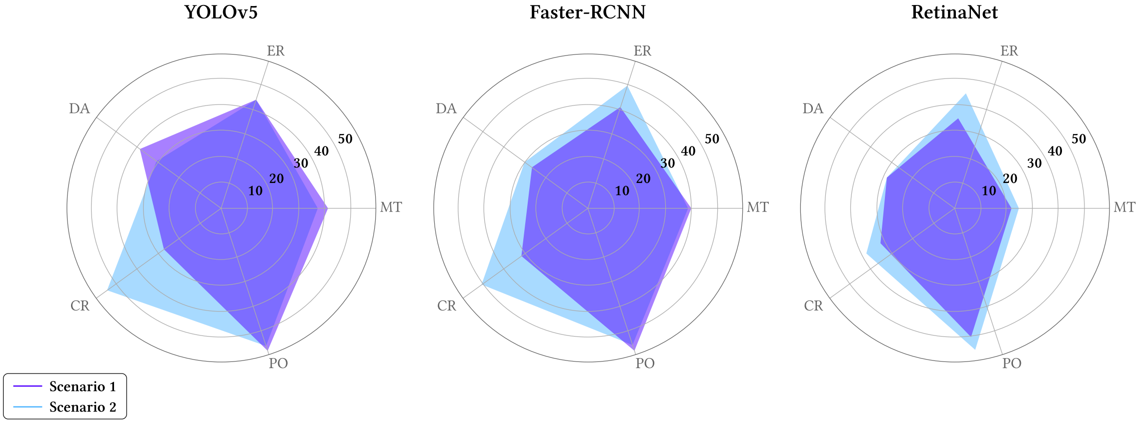

Figure 4: The overlapping area in purple illustrates the Intersection over Union (IoU) performance comparison at thresholds from 0.5 to 0.95 for YOLOv5, RetinaNet, and Faster R-CNN across different classes in the WTB dataset.

The curves in Figure 5 evaluate two key performance metrics: precision, which indicates the accuracy of the detections, and recall, which measures the method's ability to identify all relevant defects in the images.

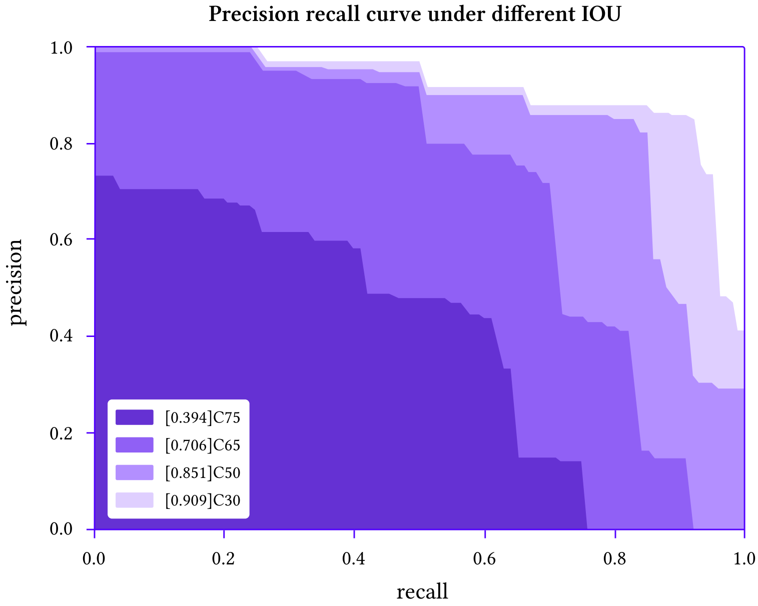

Figure 5: Precision-recall curve at various IoU thresholds.

The curves illustrate the method's performance across various conditions by employing different Intersections over Union (IoU) thresholds, namely C75 (IoU threshold of 0.75), C65 (IoU threshold of 0.65), C50 (IoU threshold of 0.5), and C30 (IoU threshold of 0.3). These IoU thresholds are instrumental in assessing the robustness of the method and the trade-offs between precision and recall at various levels. Such insights are vital for refining and optimizing defect detection methods to enhance accuracy and efficiency in real-world applications.

4.4.3. Visual comparisons

Figure 6 compares key results from Scenarios I and II, using a single trained baseline model focusing on small objects from the dataset.

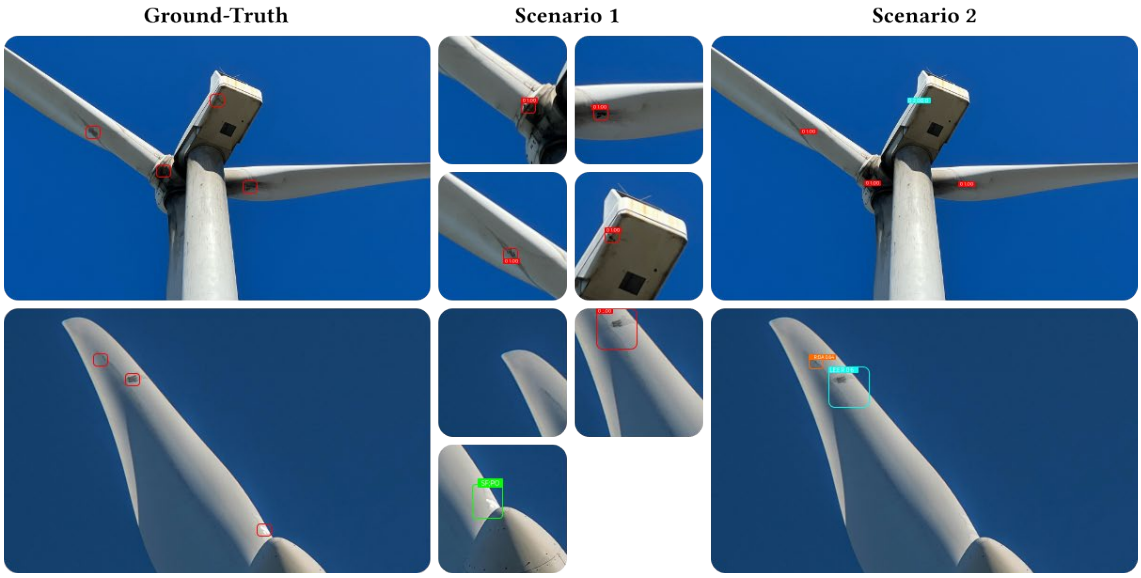

Figure 6: Visual comparison of inference strategies, showcasing prediction results across different scenarios.

Close analysis reveals that our proposed framework significantly enhances the ability to detect defects, particularly in cases where Scenario I might miss or inadequately detect them. This improvement is notably illustrated in Figure 6, particularly in the second row, the second column, where the model under Scenario I completely misses a DA defect. Conversely, this defect is successfully detected in Scenario II, shown in the second row, third column; it remains unnoticed in a 1024-pixel setting but becomes apparent in an 800-pixel context, as used in Scenario II. These findings underscore the efficacy of a multi-scale image processing strategy.

However, it is essential to recognize that both scenarios still encounter challenges, especially with certain defect classes that are difficult to localize. This issue is highlighted in Figure 6, where a PO defect at the edge of the image in the second row, third column, poses localization challenges for the model in Scenario II. This example demonstrates the complex challenges present in analyzing drone-captured imagery.

4.4.4. Efficiency

Given the complexities involved with slice-aided inference, it naturally leads to longer inference times when processing a full-size image. In Scenario II, we observed an average inference time of 0.418 seconds per patch using the YOLOv5 model, equating to approximately 27.6 seconds for an entire full-size image. However, it is worth noting that the inference duration could be shortened by selectively processing only specific patches.

A comparative analysis of inference speeds between the two scenarios reveals that Scenario I is more efficient. This efficiency advantage stems from Scenario I's method of patch-based processing, which contrasts with the slower performance observed in Scenario II. The latter's extended processing time is primarily due to its detailed analysis, which thoroughly considers predictions from the original high-resolution images.

The complexity of object detection models can be gauged by examining the number of parameters they incorporate. YOLOv5, known for its simplicity, utilizes approximately 7.2 million parameters. In contrast, RetinaNet features around 32 million parameters, and Faster-RCNN is even more complex, with about 38 million parameters.

5. Discussion

This study explored two distinct methodologies for evaluating our defect detection technique tailored for WTB inspections. Scenario I utilized segmented patches from images for both training and testing. This method proved fast but had the drawback of occasionally missing defects that span multiple patches, resulting in fragmented detections of a single defect. Additionally, detecting small objects is challenging due to the high-resolution aerial imagery and the variable scale of objects caused by the drone's varying distance from the target.

Common strategies to address this issue in high-resolution images involve randomly cropping or rescaling images before they are introduced to the model for training and testing. Nevertheless, these tactics may still result in poor representation of objects during the training phase. Alternatively, we considered segmenting the images into smaller patches for direct application in both the training and testing phases.

On the other hand, Scenario II evaluated our method on raw, high-resolution images. This approach successfully identified defects overlooked in Scenario I, especially those that were small or spanned multiple patches. Performance-wise, YOLOv5 demonstrated consistent results in both scenarios, with slight improvements for medium and large objects in Scenario II. Faster R-CNN showed substantial enhancements in detecting small and medium objects in Scenario II, though its efficiency declined for larger objects. Likewise, RetinaNet improved its detection of small and medium-sized objects in Scenario II but struggled with larger objects.

The comparative analysis summarized in Table 3 underscores that the proposed method in Scenario II consistently elevates the performance of YOLOv5, Faster-RCNN, and RetinaNet across various metrics. Our approach could significantly boost defect detection in practical applications, especially for smaller objects. This technique is versatile for both on-shore and off-shore operations, requiring only an image of the turbine blade.

6. Conclusions

This paper presents a robust framework specifically designed for detecting surface defects on WTBs using high-resolution imagery. Our proposed slice-aided inference method significantly improves the detection accuracy of small and medium-sized defects in high-resolution UAV-captured images. The experimental results show that our framework, particularly under Scenario II conditions, enhances the performance of deep learning models such as YOLOv5, Faster-RCNN, and RetinaNet. For instance, YOLOv5 achieved a mAP of 45.2% in Scenario II, compared to 42.7% in Scenario I. Similarly, Faster-RCNN improved from 38.8% in Scenario I to 44.1% in Scenario II, and RetinaNet showed an increase from 33.9% to 38.9%.

Despite these significant improvements, the proposed framework has some limitations. One of the main challenges is the increased computational cost associated with slice-aided inference, which leads to longer processing times. For example, the average inference time in Scenario II was 27.6 seconds per full-size image, which is substantially higher than in Scenario I. Additionally, the method still faces difficulties in detecting certain defect types, such as paint-off defects at the image edges, which can affect localization accuracy.

Future research should focus on addressing these limitations by optimizing the inference process to reduce computational overhead and improving the detection algorithms to better handle edge cases and complex backgrounds.

References

[1] F. R. Alharbi, D. Csala, Gulf cooperation council countries' climate change mitigation challenges and exploration of solar and wind energy resource potential, Appl. Sci. 11.6 (2021) 2648. doi:10.3390/app11062648.

[2] M. Ikram, R. Sroufe, Q. Zhang, M. Ferasso, Assessment and prediction of environmental sustainability: novel grey models comparative analysis of China vs. the USA, Environ. Sci. Pollut. Res. 28 (2021) 17891–17912. doi:10.1007/s11356-020-11418-3.

[3] J. Mamkhezri, M. Khezri, Assessing the spillover effects of research and development and renewable eFiskernergy on CO2 emissions: international evidence, Environ., Dev. Sustain. 26 (2023) 7657–7686. doi:10.1007/s10668-023-03026-1.

[4] E. Hernandez-Estrada, O. Lastres-Danguillecourt, J. B. Robles-Ocampo, A. Lopez-Lopez, P. Y. Sevilla-Camacho, B. Y. Perez-Sariñana, J. R. Dorrego-Portela, Considerations for the structural analysis and design of wind turbine towers: A review, Renew. Sustain. Energy Rev. 137 (2021) 110447. doi:10.1016/j.rser.2020.110447.

[5] K. A. Adeyeye, N. Ijumba, J. Colton, The effect of the number of blades on the efficiency of a wind turbine, IOP Conf. Ser. 801.1 (2021) 012020. doi:10.1088/1755-1315/801/1/012020.

[6] C. Cieslak, A. Shah, B. Clark, P. Childs, Wind-Turbine Inspection, Maintenance and Repair Robotic System, in: ASME Turbo Expo 2023: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers, New York, NY, USA, 2023, pp. 1–11. doi:10.1115/gt2023-101713.

[7] Y. Du, S. Zhou, X. Jing, Y. Peng, H. Wu, N. Kwok, Damage detection techniques for wind turbine blades: A review, Mech. Syst. Signal Process. 141 (2020) 106445. doi:10.1016/j.ymssp.2019.106445.

[8] O. Melnychenko, O. Savenko, A self-organized automated system to control unmanned aerial vehicles for object detection, in: Proceedings of the 4th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS'2023), CEUR-WS.org, Aachen, 2023, pp. 589–600.

[9] A. I. Panagiotopoulos, D. Tcherniak, S. D. Fassois, Damage detection on an operating wind turbine blade via a single vibration sensor: A feasibility study, in: Lecture Notes in Civil Engineering, Springer International Publishing, Cham, 2021, pp. 405–414. doi:10.1007/978-3-030-64908-1_38.

[10] S. Sun, T. Wang, F. Chu, In-situ condition monitoring of wind turbine blades: A critical and systematic review of techniques, challenges, and futures, Renew. Sustain. Energy Rev. 160 (2022) 112326. doi:10.1016/j.rser.2022.112326.

[11] Z. Liu, X. Liu, K. Wang, Z. Liang, J. A. F. O. Correia, A. De Jesus, GA-BP neural network-based strain prediction in full-scale static testing of wind turbine blades, Energies 12.6 (2019) 1026. doi:10.3390/en12061026.

[12] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama, D. Flynn, Unmanned aerial drones for inspection of offshore wind turbines: A mission-critical failure analysis, Robotics 10.1 (2021) 26. doi:10.3390/robotics10010026.

[13] W. Qi, Object detection in high resolution optical image based on deep learning technique, Nat. Hazards Res. 2.4 (2022) 384–392. doi:10.1016/j.nhres.2022.10.002.

[14] O. Melnychenko, L. Scislo, O. Savenko, A. Sachenko, P. Radiuk, Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning, Sensors 24.6 (2024) 1913. doi:10.3390/s24061913.

[15] R. Yang, R. Wang, Y. Deng, X. Jia, H. Zhang, Rethinking the random cropping data augmentation method used in the training of CNN-based SAR image ship detector, Remote Sens. 13.1 (2020) 34. doi:10.3390/rs13010034.

[16] O. Pavlova, O. Halytskyi, Video repeater design concept for UAV control, Comput. Syst. Inf. Technol. No. 1 (2024) 33–38. doi:10.31891/csit-2024-1-4.

[17] O. Melnychenko, O. Savenko, P. Radiuk, Apple detection with occlusions using modified YOLOv5-v1, in: 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IEEE, New York, NY, USA, 2023, pp. 107–112. doi:10.1109/idaacs58523.2023.10348779.

[18] L. Wang, Z. Zhang, Automatic detection of wind turbine blade surface cracks based on UAV-taken images, IEEE Trans. Ind. Electron. 64.9 (2017) 7293–7303. doi:10.1109/tie.2017.2682037.

[19] L. Wang, Z. Zhang, X. Luo, A two-stage data-driven approach for image-based wind turbine blade crack inspections, IEEE/ASME Trans. Mechatron. 24.3 (2019) 1271–1281. doi:10.1109/tmech.2019.2908233.

[20] L. Deng, Y. Guo, B. Chai, Defect detection on a wind turbine blade based on digital image processing, Processes 9.8 (2021) 1452. doi:10.3390/pr9081452.

[21] L. Peng, J. Liu, Detection and analysis of large-scale WT blade surface cracks based on UAV-taken images, IET Image Process. 12.11 (2018) 2059–2064. doi:10.1049/iet-ipr.2018.5542.

[22] M. Ruiz, L. E. Mujica, S. Alférez, L. Acho, C. Tutivén, Y. Vidal, J. Rodellar, F. Pozo, Wind turbine fault detection and classification by means of image texture analysis, Mech. Syst. Signal Process. 107 (2018) 149–167. doi:10.1016/j.ymssp.2017.12.035.

[23] A. Shihavuddin, M. R. A. Rashid, M. H. Maruf, M. A. Hasan, M. A. u. Haq, R. H. Ashique, A. A. Mansur, Image based surface damage detection of renewable energy installations using a unified deep learning approach, Energy Rep. 7 (2021) 4566–4576. doi:10.1016/j.egyr.2021.07.045.

[24] Y. Yu, H. Cao, X. Yan, T. Wang, S. S. Ge, Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor, Neurocomputing 376 (2020) 1–9. doi:10.1016/j.neucom.2019.09.071.

[25] V. V. Morozov, O. V. Kalnichenko, O. O. O. M. Mezentseva, The method of interaction modeling on basis of deep learning the neural networks in complex IT-projects, Int. J. Comput. (2020) 88–96. doi:10.47839/ijc.19.1.1697.

[26] A. Shihavuddin and X. Chen, DTU - Drone inspection images of wind turbine, Software, v. 2, Mendeley Data, 2018. doi:10.17632/hd96prn3nc.2.