2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW) | 979-8-3503-1315-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMEW59549.2023.00056

Automatic Defect Detection in Wind Turbine Blade Images: Model Benchmarks and Re-Annotations

Imad Gohar
School of Engineering and Physical Sciences
Heriot-Watt University
Putrajaya, Malaysia
ig2010@hw.ac.uk

John See
School of Mathematical and Computer Sciences
Heriot-Watt University
Putrajaya, Malaysia
j.see@hw.ac.uk

Abderrahim Halimi
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, United Kingdom
a.halimi@hw.ac.uk

Weng Kean Yew
School of Engineering and Physical Sciences
Heriot-Watt University
Putrajaya, Malaysia
w.yew@hw.ac.uk

***Abstract*—Automatic detection of defects from wind turbine blade images has shown tremendous progress in recent years. However, there are not many annotated datasets feasible for benchmarking purposes, and a lack of consistency in annotation procedures across existing works. In this paper, we investigate the data annotation process for wind turbine blade images to reduce inaccuracies in defect detection and to benchmark the performance of the patch-based detection framework on recent deep learning architectures. In this study, we identify challenges in the detection task that are incurred by the presence of extreme bounding box aspect ratios among the annotations. Experiments on two additional annotation sets show that the sets with altered box aspect ratios are able to improve the overall defect detection accuracy, particularly for classes containing boxes with very small aspect ratios. We also provide extensive class-wise results with visual examples of the highlighted problem.**

***Index Terms*—Defect detection, data annotation, aspect ratio, deep learning, wind turbine blade images**

I. Introduction

Monitoring the health of wind turbine blade (WTB) is important because the turbine's blade contributes up to 25% in energy production of the wind turbine [1]. As such, special techniques are required to control the operation and maintenance (O & M) costs of these projects to ensure their profitability. Effective O & M practices can help reduce costs associated with energy generation, such as maintenance, repair, and replacement. Due to the advances in computer vision and artificial intelligence, there has been a growing adoption of technology from the energy sector toward automatic monitoring of energy assets in a remote setting.

While automatic detection of defects in WTB images has moved on from the early use of image descriptors [2], deep learning models particularly convolutional neural network (CNN)-based architectures, have been increasingly popular [3] [4] [5] [6]. With the recent advances in deep learning, CNN architectures have become more matured and efficient, and newer architectures such as Vision Transformers (ViT) [7] have also emerged as powerful and robust models for object detection and classification tasks. However, there remains work to be done to develop accurate and reliable machine learning models where high-quality annotations (which are shared to the community) and proper performance benchmarking are established. This is especially true in the field of object detection, where precise localization and identification of objects are crucial to measure success. However, the current state of literature for WTB defect detection does not provide proper benchmarking among the widely known object detection models. Moreover, drone-based imagery are often captured at high resolutions necessitating the use of a patch-based detection framework motivated by the work of Foster et al. [8]. It is worth investigating how various parameters contribute to the effectiveness of such a framework.

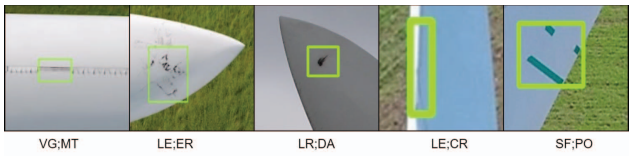


Fig. 1: Common defects found in wind turbine blade images. These defect classes are identified in the database used in our experiments. Classes with extreme aspect ratios such as LE;CR can be challenging to detect in high-resolution images.

To enable effective supervised learning of models, the data annotation process for defects on the WTB images is an important and complex task that requires careful attention to detail and a deep understanding of the underlying domain knowledge (*i.e.* types of defective blade conditions, what can be considered or not considered as a defect). To accurately annotate these images, one needs to have a clear understanding of the types of defects that are most common and the specific features that distinguish between them. However, an annota-

979-8-3503-1315-4/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMEW59549.2023.00056

tion process is often tedious in nature, with many challenging decisions to make. For instance, some defects may occupy a large area of blade surface causing bounding boxes to be unwieldy large. Meanwhile, long and narrow components of the WTB can result in extreme aspect ratios, *i.e.* boxes with a much larger horizontal side than the vertical side and vice versa, as exemplified in Figure 1. Annotators could either choose to keep the entire defect intact or break the defect into smaller boxes. Such decisions are not without consequences; small boxes are notoriously difficult to detect, as many findings have acknowledged [9], [10], while boxes with extremely large or small aspect ratios faced data scarcity for training purposes. Therefore, it is important to explore and determine optimal aspect ratios by assessing their class-wise performances on benchmark datasets.

To this end, this paper intends to bring to the attention of the community several needs that should be investigated for effective automatic detection of WTB defects. Our focus is related to three distinct areas: (1) to define specific challenges in the annotation process and to obtain optimal aspect ratios that can reduce the inaccuracy of the detection task; (2) to benchmark the performance of several deep learning architectures on a patch-based defect detection framework; and (3) to provide in-depth analysis to better understand the interacting parameters of the framework.

II. Related Work

While a number of works on defect detection in wind turbine blade (WTB) images are using the popular DTU-Drones dataset [11], their annotations and labels used are far from consistent. There are works that consider both components and defects as among the labeled classes [12] while other works [8] tend to be less precise in how they categorise the defects.

More recent works have started introducing a variety of deep learning methods and frameworks to build defect detection models and systems. Kabir et al. [13] proposed a reliability analysis framework to overcome the limitations of traditional offline analysis methods. The work of Carnero et al. [14] devise an autonomous way of capturing a complete sweep of the surface of the WTB using a portable motorised telescope system. Their system uses standard deep learning models to perform defect detection on the images. Another recent work [15] demonstrated the viability of using Vision Transformer (ViT) networks to detect defects from WTB images. Several other works focused on improving specific mechanisms in the existing detection process. Wu et al. [16] focuses on proposal generation to improve the recall of small objects in high-resolution WTB images. A detection and instance segmentation approach in [17] utilises depthwise separable convolutions instead of standard convolutions to reduce the computational cost of processing WTB images.

To ensure the reliability and reproducibility of research results, it is important to make the dataset and the associated annotations publicly available, so that other researchers can validate and build upon the work. This can also facilitate the development of benchmark datasets and standard evaluation metrics, which can help to compare and assess the performance of different machine learning models and techniques.

III. Data Preparation

This section outlines the entire process of data preparation for WTB defect detection. We provide further elaboration of the process according to the following scopes: (i) Description of the dataset, (ii) Component-based labelling of defects, (iii) Patch-wise processing employed in our framework, and (iv) Challenges in bounding box aspect ratios leading towards the re-consideration of existing annotations.

A. Dataset

To properly establish a benchmark that is consistent and reproducible, we use the publicly available DTU-Drones inspection images of wind turbine blades dataset [11] which comprises of temporal inspection images for the years 2017 and 2018 of the Nordtank wind turbine at DTU wind facilities in Roskilde, Denmark. However, the dataset, which can be accessed from this website[1], only contains the raw image data without component or defect annotations. To the best of our knowledge, all existing works in literature did not share their annotations. Therefore, we manually performed the annotation process by hand and they are publicly available on GitHub[2] .

B. Component-based labelling of defects

In current literature, some existing works [12] do not disambiguate between component detection and defect detection (both components and defects are considered together as separate classes) while other works [8] simplify the task to only detect 'damage' and 'dirt' without specific categories. For instance, the leading edge of the WTB can have various kinds of defective issues, *e.g.* erosion of the edge, or cracks.

In this work, we introduce a component-based labeling approach to enable more precise and targeted defect analysis. This is achieved by including information on the localization and type of defect, but also about the specific WTB component that is affected. While this labeling approach may be more time-consuming, it can have a far-reaching impact on the development of other related tasks such as specific WTB component detection and overall component counting for inspection and maintenance purposes.

As shown in Figure 1, our component-based labeling provides two labels per defect sample. The first part represents the labeled component of the WTB, and the second part (after the semi-colon separator) represents the type or status of the defect. The annotation process is performed by a human expert using an open-source tool [18], which facilitated the exporting of the labels in the format required by different code libraries used.

The scope of this study involves only the wind turbine blade (WTB), with no consideration given to other parts of the wind turbine structure such as the rotor and tower. Overall,

[1]https://data.mendeley.com/datasets/hd96prn3nc/2
[2]https://github.com/imadgohar/DTU-annotations

TABLE I: Total number of annotations of each class used in our experiments.

Classes	No. of labels
VG;MT	264
LE;ER	338
LR;DA	20
LE;CR	82
SF;PO	92

TABLE II: Average aspect ratios (A/R) (width over height) for each defect type in the training set partition of DTU-drones.

Type	VG;MT	LE;ER	LR;DA	LE;CR	SF;PO
mean A/R	2.026	1.180	1.050	0.768	1.102
# samples	196	229	12	57	65

we annotated 324 high resolution WTB images[3] (each of size 5280 $\times$ 2890 pixels) from the DTU-Drones dataset [11] with five distinct defect types from four different components of the WTB: missing teeth in vertex generating panel (VG;MT), leading edge erosion (LE;ER); lightning receptor damage (LR;DA), leading edge crack (LE;CR) and surface paint-off (SF;PO). The distribution of classes is given in Table I.

C. Patch-wise processing

Motivated by the work of Foster et al. [8], we simplify high-resolution images into patches to ensure that the training procedure is tractable under limited computational resources.

The patch-wise division of a high-resolution image I produces a set of non-overlapping image patches, which can be succinctly defined as:

$$\mathbf{p} = \{(p, b) \mid p \subseteq I, p \ \in \mathbb{R}^{K \times K}, b \neq \emptyset\} \tag{1}$$

where b is the set of bounding boxes associated with each image patch p, and K denotes the patch size. We show later in the experimental results (Sec. V) that $K = 1024$ gives the best results at a reasonable computational cost.

Special consideration is given to ensure that the original bounding boxes are also partitioned if they fall into different (adjacent) patches. Thereafter, patches are selected based on the criteria that each image patch should contain at least one bounding box, and patches that do not meet this criterion are discarded from the database.

D. Re-considering the annotations

There exists defects that have very small or large aspect ratios primarily due to the fact that some structures of the WTB are commonly found with such characteristics. This can be problematic for the learning of detection models. In this study, we made a preliminary attempt at investigating the impact of altering the aspect ratio of certain annotations so as to uncover desirable annotation procedures in the face of such challenges. To do so, we selected bounding boxes from the *LE;CR* class to undergo a re-annotation process. By examining the average aspect ratios of bounding boxes across all defect classes (see Table II), we find that the *LE;CR* class contained boxes that have significantly smaller aspect ratios than that of other classes. Further investigation found that many cases in this class were of extremely low aspect ratio or boxes that almost cover the complete blade vertically, as shown in Figure 2. To systematically alter the aspect ratios of these selected boxes, we derive two further sets of annotations based on the original annotations. These two sets of annotations were obtained by dividing the default annotations into two (D2) and three (D3), respectively, along the longer side of the box. Figure 2 shows an example of the re-annotation of a *LE;CR* class box with a larger bounding box that covers almost the entire length of the WTB. By allowing alternative annotations (D2 and D3), the extreme aspect ratios can be mitigated. This study provides insights into the feasibility of annotating with different bounding box sizes for the automatic detection of defects in WTB images.

IV. Defect Detection Framework

A. Network architectures

To properly benchmark the performance of different popular network architectures for object detection, we evaluate the proposed dataset on several competitive Convolutional Neural Network (CNN) YOLO-based architectures and a recent Transformer-based neural network architecture. The models that were selected for training include, YOLOv5 small(s) and medium(m)[4], YOLOv7 [19], YOLOv8 small(s)[5], and Detection Transformer (Detr) [20]. The YOLO-based architectures are initialized with weights pre-trained on the COCO dataset [21] from the *ultralytics* library while the ResNet-50 Detr model is pre-initialized with the weights from the ImageNet dataset provided by the *detectron2* library [22].

B. Training configuration

In this study, we utilised the standard transfer learning (fine-tuning) approach to develop WTB defect detection models for both the YOLO- and transformer-based networks. Different batch sizes were applied depending on the availability of computational resources for performing the training on the WTB image data. The fine-tuning process involved updating the weights of the last layer of the network while keeping the other layers frozen. We used the stochastic gradient descent optimizer with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0001, and the number of epochs is set to 300. Resource constraints mean that the learning rate for Detr model is set to 1e-4 with a batch size of 2 and YOLOv7 uses a batch size of 6 while YOLOv5 and YOLOv8 both use a batch size of 8. For all model training, the learning rate is reduced by a factor of 10 after 100 epochs and once again after 200 epochs. The training is carried out on an NVIDIA RTX 3060 GPU. The performance of the evaluated models is measured on the test set using mean average precision (mAP). We report the metric at the standard 0.5 IOU threshold (mAP@.5) and

[3]There is a total of 589 high-resolution WTB images in the DTU-Drones dataset but only 324 images contained visible defects.

[4]https://github.com/ultralytics/yolov5

[5]https://github.com/ultralytics/ultralytics

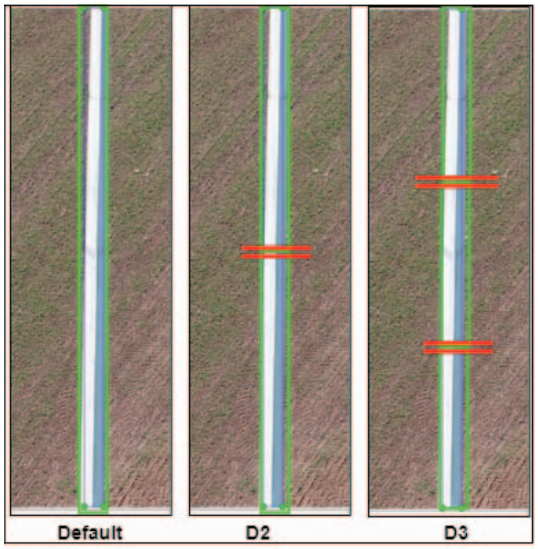


Fig. 2: The crack class with a large label was split into two parts and labeled as D2, and then further divided into three parts and labeled as D3.

TABLE III: Results using different patch sizes (K) on the DTU-Drones validation set with YOLOv5s as the base model.

K	Images	Labels	mAP50	mAP50-95
640	888	1,055	79.7	45.2
800	766	953	80.4	43.5
1024	598	796	85.5	46.9
2048	362	666	85.4	48.3

the average mAP over the range of 0.5-0.95 IOU thresholds at intervals of 0.05 (mAP@.5-.95).

V. Experimental Results & Discussion

In this section, we present a series of results from our experiments on the DTU-Drones dataset, starting with an examination of feasible patch sizes for the framework to the overall benchmark results on several recent models. We also discuss the impact of bounding box re-annotations, showing some visual results on why it matters.

A. Patch-wise results

Before benchmarking the performances of different models, we first investigate feasible patch sizes (K) to be fixed for the rest of our experimentation. Table III compares the detection performance based on different patch sizes. Results indicate that the mAP scores saturate at around $K = 1024$. Any further increase in K will increase computational cost without significantly improving the mAP scores. Using smaller patch sizes (*i.e.* 640, 800) resulted in poorer detection performances, which suggest that the defect boxes might have become too fragmented (into different patches), thus leading to more incorrect detection.

TABLE IV: mAP@.5 and mAP.5-.95 results of five object detection models on our default annotated dataset. The batch size of YOLOv7 is 6 and for Detr is 2, the remaining models use the batch size of 8.

Model	Val		Test	
	mAP50	mAP50-95	mAP50	mAP50-95
YOLOv5s	**85.5**	46.9	81.8	38.5
YOLOv5m	83.7	44.6	**83.1**	**42.5**
YOLOv8s	84.8	**47.2**	75.6	39
YOLOv7	73.9	42.3	57.1	31.2
DETR	75.9	32.3	69.54	27.66

TABLE V: mAP@.5 and mAP@.5-.95 results with YOLOv5s as baseline on three types of annotations, Default, D2 and D3.

Method	Val		Test	
	mAP50	mAP50-95	mAP50	mAP50-95
Default	85.5	**46.9**	81.8	38.5
D2	87.1	45.8	70.6	35.7
D3	**90.6**	**46.9**	**83.4**	**41.5**

B. Overall benchmark results

The default annotated dataset is used to evaluate the performance of several recent object detection models, namely YOLOv5s, YOLOv5m, YOLOv8s, YOLOv7 and Detr, as shown in Table IV. The evaluation results show that YOLOv5s achieved the highest score on the validation set while YOLOv5m clearly performed the best on the test set. Meanwhile, YOLOv8 is marginally better than the YOLOv5 models on the validation set but surprisingly did not perform well on the test set. We also experimented with two other models at smaller batch sizes which obviously reported less optimal results. The Detr model is more computationally expensive and hence it probably requires more data to realise its potential for this task. Overall, the results on a 5-class detection task indicate that these models are reasonably effective in detecting and localising defects in WTB images.

C. Re-considered annotations results

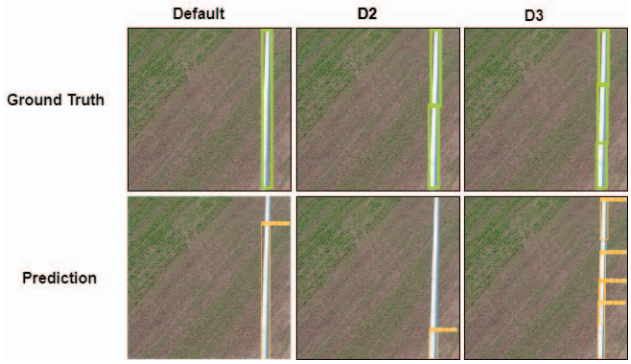
Table V compares the performance of the three sets of annotations (discussed in Sec. III-D), denoted as *Default*, *D2*, and *D3*, based on the YOLOv5s model. We report improvements in the mAP score for *D2* and *D3* annotation sets compared to that of the *Default* annotation set. These results indicate that reducing the impact of small aspect ratios in just one particular class can yield an overall improvement in the detection performance. The performance of D2 decreased dramatically in the test set due to the fact that most LE;CR samples in the test set (after partitioning from the dataset) were not the ones with the smallest aspect ratios..

Table VI shows the class-wise detection results across three annotation sets. The results showed that the mAP for the LE;CR class significantly increases from the Default set to the D3 set, from mAP@.5 of 57.9 to 80 on the validation set, and mAP@.5 of 58.5 to 71 on the test set. Specifically, improving the weakest-performing class can be beneficial to the overall performance of the detector.

TABLE VI: Class *LE;CR* result comparison for three types of annotations *i.e.* Default, D2 and D3 using YOLOv5s object detector and mAP@0.5.

Classes	Validation			Test		
	Default	D2	D3	Default	D2	D3
VG;MT	96.4	99.3	97.8	84.7	79	82.1
LE;ER	83.3	78.9	81.6	83.4	82.8	77.2
LR;DA	99.5	99.5	99.5	99.5	56.1	95.5
LE;CR	57.9	63.6	80	58.5	58	71
SF;PO	90.4	94.3	94	83.1	83.8	91.4

(a) Example 1

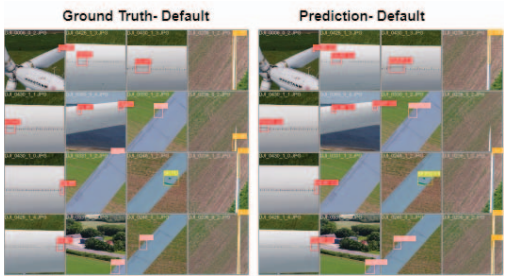


(b) Example 2

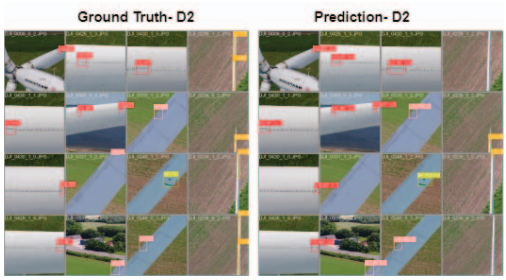
Fig. 3: Ground Truth with corresponding predicted results on Default, D2 and D3 annotations using YOLOv5s network.

D. Visual results

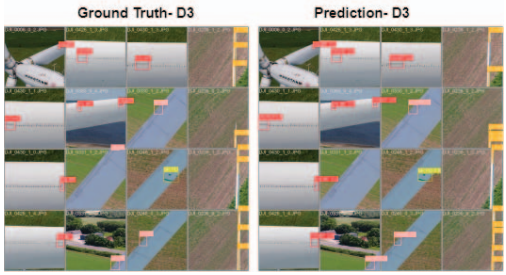
We show some examples of visual results from three annotation sets using the YOLOv5s model along with the actual ground truth labels in Figures 3a and 3b. The ground truth annotations (in light green boxes) reflect the actual presence of defects in the WTB images, while the detection (in orange boxes) show the model's ability to accurately identify and localize these defects. The visual results clearly show that the D3 annotations can facilitate better learning from the used model, and that boxes with extremely small aspect ratios (and likewise, for those with very large aspect ratios) can be notoriously difficult to detect. Fig. 4 shows that the detection performance for most other classes remains largely unchanged even after re-annotations were applied to the LE;CR class.



(a) Default annotations



(b) D2 annotations



(c) D3 annotations

Fig. 4: Ground truth vs. corresponding predicted results on different annotation sets.

VI. Conclusion

This work demonstrates the importance of carefully considering the data annotation process for complex tasks such as defect detection in wind turbine blade images. By altering the aspect ratio of certain defects, we show by experiments that it is possible to improve the accuracy of defect detection in images where defects may structurally possess extreme aspect ratios. Overall, the findings of this work emphasise the need for domain knowledge and a close attention to detail during the data annotation process to reduce inaccuracies and improve model performance. Future directions toward more robust defect detection should strongly consider incorporating domain knowledge via an active learning setup where interaction with 'human-in-the-loop' can be beneficial to training accurate detection models.

Acknowledgment

This work is supported by HWUM JWS 2021 funding and the UK Royal Academy of Engineering under the Research Fellowship Scheme RF/201718/17128. .

REFERENCES

[1] K. A. Adeyeye, N. Ijumba, and J. Colton, "The effect of the number of blades on the efficiency of a wind turbine," in *IOP Conference Series: Earth and Environmental Science*, vol. 801, no. 1. IOP Publishing, 2021, p. 012020.

[2] L. Deng, Y. Guo, and B. Chai, "Defect detection on a wind turbine blade based on digital image processing," *Processes*, vol. 9, no. 8, p. 1452, 2021.

[3] S. Moreno, M. Peña, A. Toledo, R. Treviño, and H. Ponce, "A new vision-based method using deep learning for damage inspection in wind turbine blades," in *2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2018, pp. 1–5.

[4] N. Anantrasirichai and D. Bull, "Defectnet: Multi-class fault detection on highly-imbalanced datasets," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2481–2485.

[5] D. Xu, C. Wen, and J. Liu, "Wind turbine blade surface inspection based on deep learning and uav-taken images," *Journal of Renewable and Sustainable Energy*, vol. 11, no. 5, p. 053305, 2019.

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

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[8] A. Foster, O. Best, M. Gianni, A. Khan, K. Collins, and S. Sharma, "Drone footage wind turbine surface damage detection," in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 2022, pp. 1–5.

[9] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, "An evaluation of deep learning methods for small object detection," *Journal of electrical and computer engineering*, vol. 2020, pp. 1–18, 2020.

[10] K. Tong and Y. Wu, "Deep learning-based detection from the perspective of small or tiny objects: A survey," *Image and Vision Computing*, p. 104471, 2022.

[11] A. Shihavuddin and X. Chen, "Dtu - drone inspection images of wind turbine," 2018.

[12] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, and R. Reinhold Paulsen, "Wind turbine surface damage detection by deep learning aided drone inspection analysis," *Energies*, vol. 12, no. 4, p. 676, 2019.

[13] S. Kabir, K. Aslansefat, P. Gope, F. Campean, and Y. Papadopoulos, "Online dynamic reliability evaluation of wind turbines based on drone-assisted monitoring," *arXiv preprint arXiv:2211.13258*, 2022.

[14] A. Carnero, C. Martín, and M. Díaz, "Portable motorized telescope system for wind turbine blades damage detection," *Engineering Reports*, p. e12618, 2023.

[15] D. Dwivedi, K. Babu, P. K. Yemula, P. Chakraborty, and M. Pal, "Identification of surface defects on solar pv panels and wind turbine blades using attention based deep learning model," *arXiv preprint arXiv:2211.15374*, 2022.

[16] H. Wu, B. Li, L. Tian, J. Feng, and C. Dong, "An adaptive loss weighting multi-task network with attention-guide proposal generation for small size defect inspection," *The Visual Computer*, pp. 1–18, 2023.

[17] P. Diaz and P. Tittus, "Fast detection of wind turbine blade damage using cascade mask r-dscnn-aided drone inspection analysis," *Signal, Image and Video Processing*, pp. 1–9, 2023.

[18] Tzutalin, "Labelimg," Free Software: MIT License, 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[21] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[22] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.