Structural health and intelligent monitoring of wind turbine blades with a motorized telescope

Alejandro Carnero
ITIS Software
University of Malaga
Malaga, Spain

Cristian Martín
ITIS Software
University of Malaga
Malaga, Spain

Manuel Díaz
ITIS Software
University of Malaga
Malaga, Spain

***Abstract*—Currently, wind energy plays a fundamental role in the process of generating energy in a sustainable and environmentally friendly manner. However, their infrastructures require ongoing maintenance tasks that involve considerable risk. This is why a predictive maintenance system for the surface inspection of wind turbine blades based on machine learning techniques has been developed. Specifically, convolutional neural networks have been applied to detect and classify turbines and their blades, as well as the surface defects that may appear on them. The system comprises a mobile application that makes use of a telescope to take pictures with certain precision, a computing edge node responsible for processing the images that are captured, and a motorized mount that allows the telescope to move. The objective of this open-source project is to detect and classify different surface defects on the blades of wind turbines and carry out the maintenance of these infrastructures. The system is responsible for undertaking a complete sweep of the surface of the turbine blades in an autonomous way and finally presents the defects found to the user. The deep neural networks also help the system to decide which movements the motorized mount has to make together with the telescope to perform the inspection. Accuracies of around 97% for label predictions and 90% for bounding box coordinate predictions have been achieved for the convolutional deep learning models. Two possible approaches have been considered for the project: the first is to carry out all the necessary computation on a mobile phone to have a portable solution, and the second option considers a edge node to balance the load and thus not overload the mobile device. Tests show that the edge node approach gives better results overall. The proposed system for detecting surface damage on blades was experimentally validated on a wind farm.**

***Index Terms*—structural health monitoring, wind turbines, motorized telescope, machine learning**

I. Introduction

Today, the world of wind energy continues to grow exponentially, with numerous wind turbines in operation on a regular basis. These infrastructures are located in environments where conditions are complex and therefore these experience numerous failures over time. Due to this and the large size of the turbines, their maintenance carries a lot of risk [1] [2]. The repair of these systems supposes an increase in their lifetime but in turn, late failure detection increases their operating cost. For this reason, it is important to detect the problems early and thus reduce the possibility of a major failure that increases the operating cost of the turbines [3].

ICMLA 2022. DOI: https://doi.org/10.1109/ICMLA55696.2022.00022

The blades are a fundamental element of wind turbines. They are expensive, accounting for 20% of the total cost [4] [5] and slightly less than 15% of turbine failures approximately [6]. Some of the consequences of blade failures are considerable economic losses due to the repair or replacement, unplanned system shutdowns, and other types of accidents that can endanger human and environmental safety [7]. Consequently, it is important to develop a system that monitors the structural health of wind turbines safely and efficiently in order to avoid these problems.

There are different types of traditional wind turbine inspections. The use of various sensors has allowed the internal state of the turbines to be monitored, however, for their external state, there are still no methods that are safe and efficient enough, which are needed. For external inspection, the most common method is manual [8], which is carried out by an inspector with the help of ropes, usually requiring a lot of time and endangering his integrity [9] [10]. Other approaches also consider the use of drones or similar solutions, which may require prior experience in handling drones and certain licenses. To address this challenge, in this work we propose an autonomous solution based on a motorized telescope and machine learning techniques that are responsible for processing images taken from the surface of the turbine blades and detecting damage on them. Thanks to this, the inspection time and the risk of suffering an accident would be considerably reduced [11]. The evolution of computer vision in recent years, such as image classification and object detection [12], allows for dealing with problems in different areas with acceptable results, also providing safety in the work environment.

Due to the current and growing importance of the wind sector, this open-source project aims to provide an alternative solution for monitoring the external structural health of wind turbine blades based on the use of convolutional neural networks [13] [14], which are capable of analyzing and finding possible surface defects with high precision. Our system comprises the following components: i) a mobile phone to take pictures of the blades in a convenient way for the operators; ii) a telescope that allows high magnification to correctly visualize the surface of the blades; iii) a motorized mount that allows the movement of the telescope; iv) and an edge computing node to process and analyze all the images captured and finally present the defects found to the user. Finally, two deep convolutional neuronal network models which, according to their predictions, help the system to decide which way to move the mount to carry out the inspection and what defects appear on the surface of the blades.

The edge computing paradigm is based on the fact that the data generated by the devices is not processed by the devices themselves or sent directly to the cloud, but is first processed in a smaller decentralised data center, thus avoiding latency issues and device overloads. So, with the goal of reducing the load on the phone and getting faster predictions, we make use of this paradigm in our work.

The rest of the paper is organized as follows. Section II introduces the traditional types of wind turbine inspections and the related work. The methods and datasets used for blade classification and detection are presented in Section III. Then, in Section IV the system architecture and its components are discussed. Section V shows the results of the validation carried out to evaluate the system through two different approaches: the first one integrates the entire solution on the mobile device in a standalone way, and the second one considers the use of the edge node to balance the load between the two devices. Finally, Section VI presents our conclusions and future possible works.

II. Background

The idea of preventive maintenance is to repair or replace components before they break [15]. In the inspection of the blades of wind turbines, there are different traditional methods for early failure detection. On the one hand, there are manual inspections with rope access, which require a lot of time and entail so much risk for the inspector. On the other hand, inspections can be performed with drones, but it is necessary to have a license to fly, in addition to the restrictions that the battery implies along with the weather conditions. Finally, the inspections with telescopes require manual calibration and some training and experience on the part of the inspectors. Figure 1 shows the traditional types of blade inspections.

Fig. 1. Traditional types of wind turbine inspections, from left to right: manual inspection with ropes [16], drone inspection [17], and telescope inspection [18]

There are also preventive maintenance techniques based on system conditions measured through different sensors. This method refers to the state of the structure and allows continuous monitoring of various parameters that helps early detection of failures [19]. However, this requires acquiring, processing, analyzing, and interpreting a lot of data, in addition to the fact that this solution would fit only one turbine, entails the installation and maintenance of various sensors, and is not intended to detect surface faults.

There are numerous works that address extensively the structural health monitoring of wind turbine blades. Yang et al. [6] presented state-of-the-art blade structural health monitoring techniques; and a newly developed technique for monitoring the structural health of the wind turbine based on the concept of transmissibility of frequency response functions. Nonetheless, its successful implementation relies on the in-situ installation of several sensors, and it does not allow the inspection of different turbines from distances between 200 and 300 meters.

Recently, the rapid advance of deep machine learning and computer vision has led to the use of these technologies in the field of inspection of wind turbine blades, as well as in many other fields of application. These techniques provide inspectors with highly likely suggestions of damage locations within the images that help significantly to reduce inspection and data analysis time while improving the quality of manual inspection [20].

In [21], an image-based fault diagnosis method for wind turbine blades is proposed. The blade damage recognition is realized by two-stage learning. The first learning stage is a deep-feature extractor learning stage and the second is a pattern learning stage. This work is not an open-source project, they only consider trailing edge damage and delamination as possible superficial defects and they have not validated it in a real wind farm.

In [22], a deep-learning-based method for crack detection is proposed. A supervised deep convolutional neural network is trained to classify each image patch in the collected images. This work considers only cracks as possible surface defects, leaving out numerous other types of damages.

In [23], a multi-class classification framework for power infrastructure detection and classification using deep learning and drone-captured images is proposed. Unlike our project, which makes use of a telescope together with a motorized mount, they use a drone to take the images from a certain height, which may require a specialist.

To sum up, external visual inspection of wind turbine blades can be considered as image detection and classification tasks. Deep computational learning techniques are one of the best approaches for carrying out object recognition tasks, achieving very good results in a wide variety of artificial vision problems. In addition, new approaches are needed for the semi-autonomous monitoring of blade turbines without the need for qualified personnel.

III. Methods and Datasets for Blade Classification and Detection

To achieve the objective of detecting faults for the preventive maintenance of the turbine blade surface, two convolutional network models have been implemented. These networks share the same structure but are responsible for classifying and detecting different objects from the parts of a wind turbine. The first of them is responsible for detecting the turbines themselves and their blades so that it is known where the blades are located within the images and therefore the direction in which the motorized mount must be moved in order to carry out a complete scan of its surface. The second model is responsible for detecting and classifying the possible defects on the surface of the turbine blades. Figure 2 shows a blade sweep example taken by our system in a real wind farm.

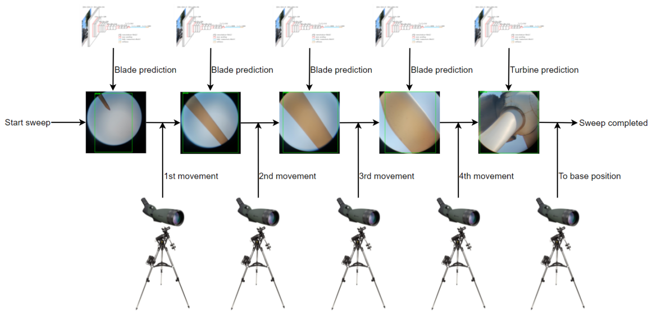

Fig. 2. Blade sweep process carried out with our solution

The architecture used for the models is the well-known VGG16[1], which comes pre-trained with the ImageNet[2] dataset. In addition, the model has been modified for the adaptation in this field through a transfer learning process. In this case, the model has been used without its final layers, all its weights have been frozen, and new blocks of trained layers have been added at the end. An input size of 224×224 has been defined to achieve greater speed when training the models. The chosen training configuration was 20 epochs, a learning rate of 0.001, and a batch size of 32. The Keras[3] framework has been used along with the machine learning framework TensorFlow[4]. Figure 3 shows some examples of the second model's prediction on some test images to highlight its ability to locate and classify surface defects.

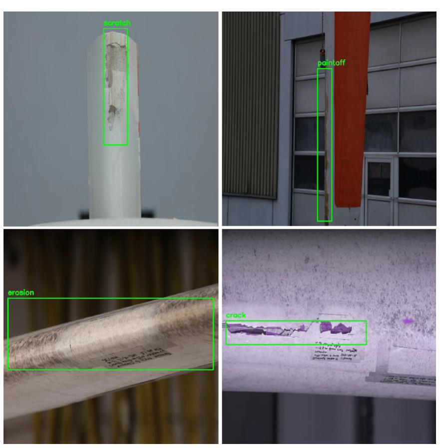

Fig. 3. Surface defect detection in images from [24] and [25] datasets

The dataset used to train the first model is the DTU-Drone inspection images of wind turbine [26], which contains images of wind turbines and their blades from many varied angles. On the other hand, the second model has been trained with two different data sources, which are the Wind Turbine Blade Surfaces Dataset [24] and the Wind Turbine Blade SfM Image Capturing Setups Dataset [25], which contain images of different possible surface defects of the wind turbine blades. The surface defects chosen for this project are cracks, erosions, discolorations, mechanical failures, and scratches.

Before moving on to training the models, it is important to correctly prepare the data. All images from both datasets have been labeled using LabelImg[5]. This process includes specifying which object and where it is located within each of the images. In addition, Albumentations[6], an artificial vision tool to generate new images from existing ones, has been used. The first dataset initially contained 535 images, corresponding to the turbines and their blades, while the second dataset contained 500 images, corresponding to the different surface defects. After applying the data augmentation techniques, a total of 1070 images have been obtained from the first dataset and 1000 from the second. For the sake of the reusability of this data, we have uploaded the two datasets to our GitHub project repository[7] along with the deep learning models used.

IV. Motorized telescope system architecture

Our solution consists of an open-source project[8] made up of a set of components whose main goal is to provide inspectors with a preventive maintenance system for wind turbine blades surface based on deep learning and computer vision techniques. The idea is to perform a full sweep of the blade surface in search of defects. This can be achieved thanks to the predictions of the neural network models that, based on the photographs taken with the mobile phone and the telescope, can infer what defect appears in the image and where it is located. To provide all the required functionality, the system has the following components:

- **Deep convolutional neural network models**: There are two neural network models: the first one is responsible for locating the wind turbines and their blades, and the second one is in charge of identifying the different possible surface damages.
- **Telescope**: A monocular terrestrial telescope is used to achieve a certain magnification when taking photographs.
- **Motorized mount**: A motorized tripod that allows the telescope to move. The mount is controlled by the system.
- **Mobile application**: Android application that is responsible for taking the photographs and sending them to the edge node for further processing.
- **Edge node**: Computation node in charge of controlling the motorized mount based on the predictions of the deep-learning models and presenting the defects found to the user.

[1]https://neurohive.io/en/popular-networks/vgg16/
[2]https://image-net.org/
[3]https://keras.io/
[4]https://www.tensorflow.org/?hl=es-419
[5]https://github.com/tzutalin/labelImg
[6]https://albumentations.ai/
[7]https://github.com/ertis-research/shm-wind-turbines/tree/main/datasets
[8]https://github.com/ertis-research/shm-wind-turbines

The operation of the system is detailed as follows. At first, the operator positions the telescope on a corner of a blade to be inspected and takes a picture from the developed mobile application of its surface, which is sent to the edge node and processed by the first neural network model. If the model predicts that a blade appears in the image, the edge node sends a movement command to the mount to continue analyzing its surface. Afterward, the image is processed by the second neural network model to detect surface defects. This cycle repeats until the blade is no longer detected or until a maximum number of mount moves is reached, which in this case has been set to seven moves. This is due to the fact that with this number of movements the mount is positioned completely horizontally, which may not be practical for the inspector. Once either of the two termination conditions is reached, the edge node sends a move command to the mount to return to its base position and the defects found are presented to the operator. Figure 4 shows the system workflow.

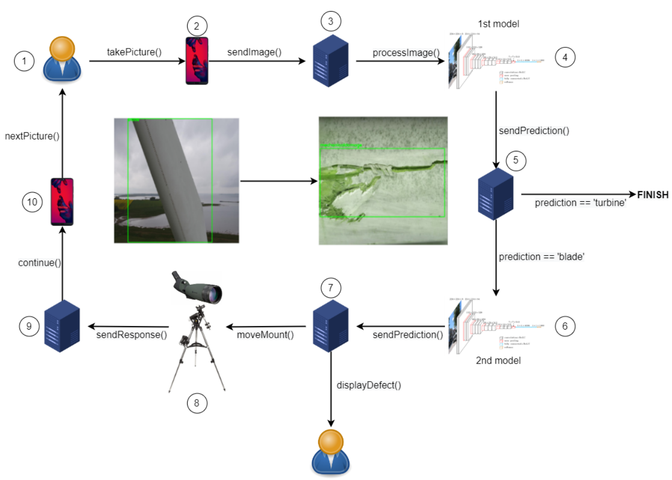

Fig. 4. System workflow

A. Telescope

The telescope chosen for this work was the BRESSER Pirsch 25-75×100[9]. It is a monocular terrestrial telescope with zoom and 100-millimeter optics that makes high magnifications possible. It has a 45º angle view, a full multi-layer coating, and a high light-gathering capacity. It is also great for digiscoping. With a digital camera adapter, it can be used with almost all digital cameras or compact camcorders.

B. Motorized mount

The motorized mount chosen for this project has been the Explore Scientific IEXOS-100 PMC-Eight Goto Wifi[10]. It is a lightweight astronomical mount with Goto and Wifi, incorporating the PMC-Eight[11] precision tracking system and stepper motors that allow effortlessly finding and tracking thousands of objects with high precision. It is a German equatorial mount that is characterized by its excellent precision in keeping objects centered in the instrument's field of view for long periods of time. For this project, the connection between the mount and the edge node is made by Wifi, using the well-known Telnet communication protocol[12].

C. Mobile application

To use the telescope, the phone's camera has to be placed in the eyepiece of the telescope with the help of a bracket so that photos can be taken through the telescope. The Android application is responsible for taking photographs and for sending them to the edge node so they can be processed through the neural network models. The application begins by taking photographs of the blade surface and performing a complete sweep over it, either until the blade is no longer detected or the maximum range of movement of the mount is reached.

D. Edge node

The edge computing node is an intermediate processing interface responsible for receiving the photographs taken by the Android application and processing them through the two neural network models. After each processing, the edge will send the corresponding movement commands to the motorized mount based on the movement predictions (model 1); and present the surface defects found on the blade to the user (model 2). Communication between the mobile application and the edge node has been established via a USB connection.

V. Evaluation

This section presents the evaluation of the developed system, which has been validated in a real work environment. Figure 5 shows our solution ready to carry out an inspection in a wind farm. Mainly, the objective of this evaluation is to evaluate the performance of the two deep learning models developed, both for damage classification and to move the telescope, and to compare the two possible approaches depending on the implementation of the system. The first approach consists of carrying out all the computational processes on the mobile device, i.e., processing the images through the neural network models, moving the motorized mount, and presenting the defects found to the user; while the second approach makes use of an intermediate edge node that takes care of all these processing tasks, thus freeing up the hardware resources of the mobile phone. It should also be noted that the two neural network models have remained the same in both approaches, so the accuracy of the predictions is the same.

The mobile phone parameters monitored are CPU, RAM, and battery usage. These first three tests measure the performance parameters of the mobile device itself. In addition, the response times of the system for different tasks, such as image processing, model predictions, communication with the mount, and the presentation to the user of the defects found, were also

[9]https://www.bresser.de/en/Sport-Optics/Spotting-Scopes/BRESSER-Pirsch-25-75x100-45-Spotting-Scope.html

[10]https://www.bresser.de/en/Astronomy/Accessories/EXPLORE-SCIENTIFIC-iEXOS-100-PMC-Eight-Wifi-Goto-Mount.html

[11]http://02d3287.netsolhost.com/pmc-eight/

[12]https://en.wikipedia.org/wiki/Telnet

Fig. 5. Developed system evaluation in a wind farm

monitored. This last latency test compares times measured on both approaches.

The mobile phone used to evaluate the system was a Huawei P20 Pro with 6GB of RAM and a HiSilicon Kirin 970 processor with 2 cores: a Cortex A73 CPU with 4 cores at 2.36 GHz speed and another Cortex A53 with 4 cores at 1.8 GHz speed. The main specifications of the edge node on which the tests were carried out are Intel Core i9-10850L 1200 processor, 16GB DDR4 3200MHZ RAM (4x16GB), and Gigabyte RTX 3090 24GB graphics card.

A. *Convolutional deep learning models performance*

This evaluation aims to assess the performance of both models, both the results of the first one, in charge of detecting wind turbines and their blades, and the results of the second one, in charge of locating and classifying the different surface damage on them. In particular, we will present the progression of the model's accuracy for the prediction of the defect that appears in the image, and another one for the points that delimit the box where each defect is located. In general, regarding the results of the validation of the two models, accuracies of around 97% for label predictions and 90% for bounding box coordinate predictions have been achieved on the validation dataset.

*1) **First model**:* The progression of the label accuracy of the first neural network model is shown in Figure 6. As it can be appreciated, label accuracy progression reaches values over 0.95 as epochs progress, which is a good result to predict the next movement.

In the case of the progression of the bounding box accuracy, values very close to 0.9 have been obtained, which is also a fairly acceptable result. Figure 7 shows the progression of the bounding box accuracy of the first model.

*2) **Second model**:* The progression of the label accuracy of the second neural network model is shown in Figure 8. In

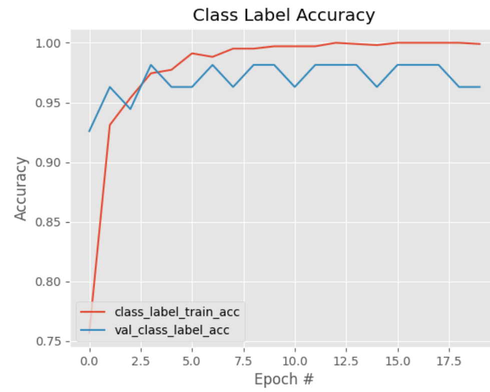

Fig. 6. First model class label accuracy

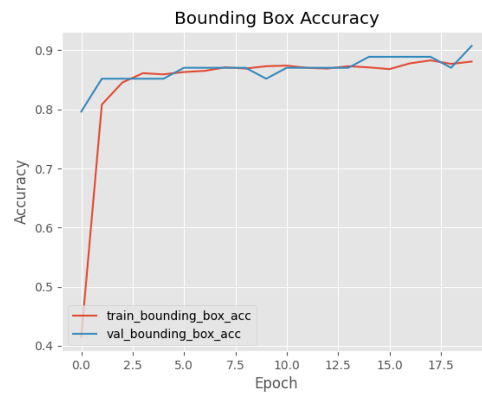

Fig. 7. First model bounding box accuracy

this case, good results are also obtained for the label accuracy, which are very close to 1.

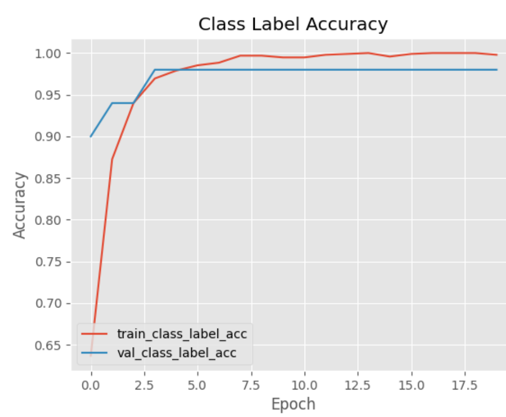

Fig. 8. Second model class label accuracy

The results for the bounding box accuracy progression for the second model are also adequate. They reach values over 0.9 as epochs progress, as seen in Figure 9.

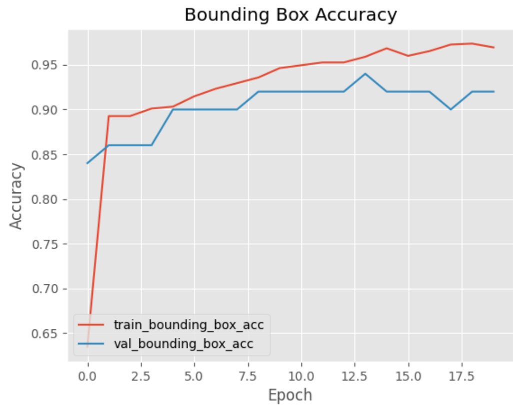

Fig. 9. Second model bounding box accuracy

B. CPU usage

The percentage of CPU usage of the Android application using the mobile phone as the only device is shown in Figure 10, and the use of the edge computing paradigm is shown in Figure 11. As can be seen, both graphs contain a series of peaks corresponding to each of the photos taken through the mobile phone and their processing through the neural network models. We can see that the percentage levels obtained if the mobile phone is used as a single processing device reach values very close to 50%, while with the edge node the application reaches a maximum of 20%, which significantly improves the overloading of the phone.

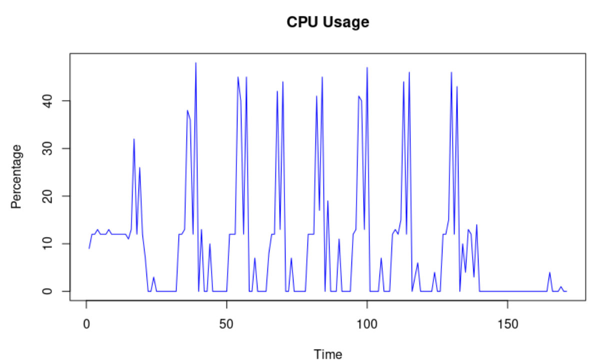

Fig. 10. Mobile approach - CPU usage

C. RAM usage

The amount of RAM in gigabytes used by the application with both approaches is shown in Figures 12 and 13, respectively. As can be seen, in both cases there is an initial increase in usage due to the memory load of the application itself. As time progresses, we see a series of peaks corresponding to the photos taken with the mobile phone and the predictions of the models for the same reason as before. However, in the case of using edge computing, a memory reduction of almost 90% is achieved, as the highest peak usage with the phone approach

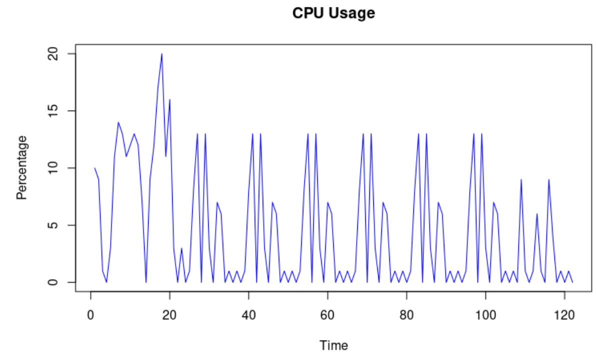

Fig. 11. Edge approach - CPU usage

reaches 1GB while with the edge approach it slightly exceeds 0.12GB.

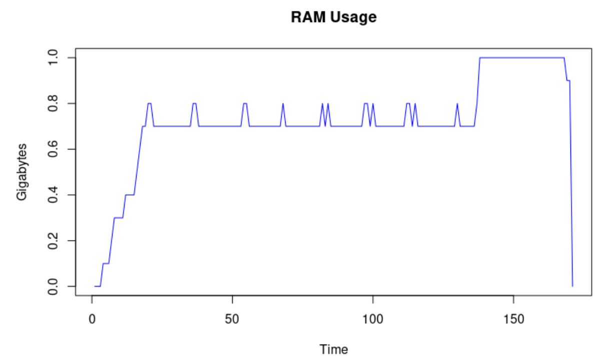

Fig. 12. Mobile approach - RAM usage

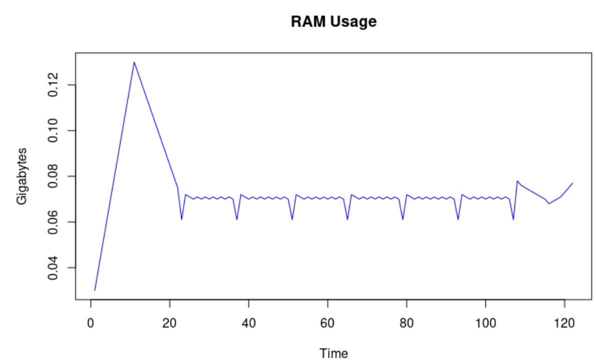

Fig. 13. Edge approach - RAM usage

D. Energy usage

The battery usage level of the mobile application is shown in Figures 14 and 15 for the two approaches studied. This parameter is one of the most important as it is affected by the previous ones and marks the lifetime of the phone and the inspection. In this case, four possible degrees of use have been established: none, very little use, corresponding to value 0; light, corresponding to value 1; medium, corresponding to value 2; and heavy, corresponding to value 3. In the first case, we see third-degree peaks in battery usage, which correspond to the predictions of the neural network models introduced in the mobile application. In the edge computing evaluation, there is a peak of degree 2 at the beginning due to the start of the application itself, and from then on it only varies between degrees 0 and 1, which corresponds to the Android activity responsible for taking the photographs. In the approach that makes use of the edge computing paradigm, there is also a significant battery saving, about two-thirds of its total percentage.

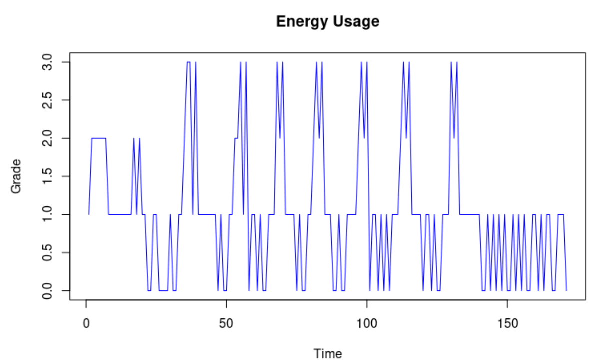

Fig. 14. Mobile approach - Energy usage

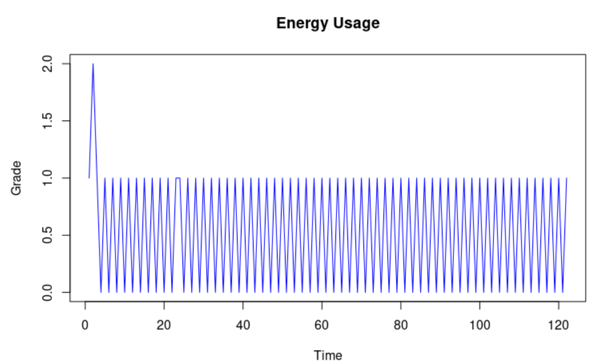

Fig. 15. Edge approach - Energy usage

E. Latency tests

Four tests related to latency were carried out. The first one was intended to measure the time the mobile phone and the edge node take to process an image, which includes pre-processing to apply filters and remove black fringes caused by the telescope eyepiece. The second test consists of measuring the time it takes for the second neural network model to make a prediction. The third test measures the time from the moment a command is sent to the motorized mount until a response is obtained. Finally, the fourth one measures the time the system takes to present the defects found to the user. For each test, ten different times have been measured, and the mean has been calculated and shown in Figure 16 for the mobile phone, and in Figure 17 for the edge node. As can be seen, the most time-consuming task is to process the input image, which exceeds 2500 milliseconds in the case of the mobile phone, and just under 200 milliseconds for the edge node, which improves latency times by around 90%. On the contrary, the other tasks, in general, are carried out in relatively short times, being the communication with the mount the quickest of all in both cases. In these tasks, the difference between using or not using the edge computing paradigm is also noticeable. An important aspect to note is that the tests with both devices were conducted in a closed environment, so the distance to the edge node was practically negligible, which would not be the case in a real wind farm inspection. This means that in a real inspection the edge times would be negatively affected, but if high-speed networks are available, better results in terms of latency would still be obtained than with the mobile phone-only approach.

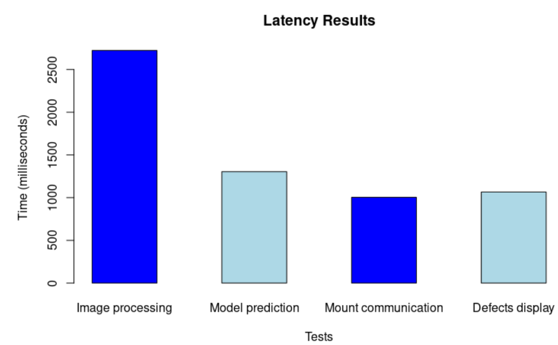

Fig. 16. Mobile approach - Latency results

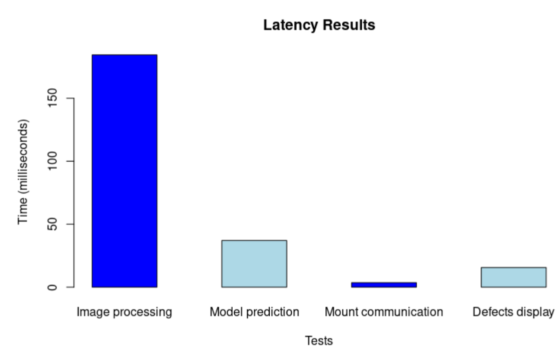

Fig. 17. Edge approach - Latency results

Therefore, according to the results, the mobile-only approach offers an integrated and autonomous solution, but by using the edge node a reduction in system overload and response times is achieved, which leads to increased uptime of the solution.

VI. Conclusions and future work

Traditional methods of inspecting wind turbine blades are time-consuming and involve considerable risk on the part of the inspectors. In contrast, our solution considers deep-learning-based techniques, which allow for visually inspecting blades without risk, reducing inspection time, and improving productivity and economic benefits. Using these deep learning models, a motorized telescope is moved automatically, which at the same time takes photos that will be analysed for damage detection. The defects considered in this work include cracks, erosions, mechanical failures, discolorations, and scratches.

In general, the performance of the machine learning models is adequate, with average accuracy values around 90% on the validation dataset. In addition, the system was validated in a real wind farm with satisfactory results. Transfer learning and data augmentation techniques have been applied to improve the generalizability of the deep-learning models, and to overcome the limited availability of superficial damage samples for learning, respectively.

As discussed in the evaluation, the use of the edge computing paradigm brings some advantages compared to the approach of leaving all processing on the mobile phone. Among its advantages, highlights the fact that it relieves the mobile phone's resources by spreading the processing load across both devices, thus achieving better results in terms of system performance, even if with small latency issues in the communication. As long as low latency communications are available for interconnection, there is a potential to integrate more devices using this paradigm and thus further carrying out a greater number of inspections at the same time.

Possible future lines to be followed in this project could be to study the possibility of extending the system to support incremental and federated learning in order to achieve continuous knowledge acquisition by models in the field, and to integrate this system with our framework Kafka-ML [27].

Acknowledgment

This work is funded by the Spanish projects UMA-CEIATECH-19 ("Wind Turbine Preventive Maintenance based on Deep Learning Techniques in the Fog"), RT2018-099777-B-100 ("rFOG: Improving Latency and Reliability of Offloaded Computation to the FOG for Critical Services"), PY20_00788 ("IntegraDos: Providing Real-Time Services for the Internet of Things through Cloud Sensor Integration"), and UMA18FEDERJA-215 ("Advanced Monitoring System Based on Deep Learning Services in Fog").

References

[1] X. Yang, Y. Zhang, W. Lv, 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 163 (2021) 386–397. doi:https://doi.org/10.1016/j.renene.2020.08.125.

[2] K.-Y. Oh, J.-Y. Park, J.-S. Lee, B. I. Epureanu, J.-K. Lee, A novel method and its field tests for monitoring and diagnosing blade health for wind turbines, IEEE Transactions on Instrumentation and Measurement 64 (6) (2015) 1726–1733. doi:10.1109/TIM.2014.2381791.

[3] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, Y. Si, A data-driven design for fault detection of wind turbines using random forests and xgboost, IEEE Access 6 (2018) 21020–21031. doi:10.1109/ACCESS.2018.2818678.

[4] J. Marín, A. Barroso, F. París, J. Cañas, Study of fatigue damage in wind turbine blades, Engineering Failure Analysis 16 (2) (2009) 656–668, papers presented at the 24th meeting of the Spanish Fracture Group (Burgos, Spain, March 2007. doi:https://doi.org/10.1016/j.engfailanal.2008.02.005.

[5] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, G. Ekemb, Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges, Energies 7 (4) (2014) 2595–2630. doi:10.3390/en7042595.

[6] W. Yang, Z. Peng, K. Wei, W. Tian, Structural health monitoring of composite wind turbine blades: challenges, issues and potential solutions, IET Renewable Power Generation 11 (4) (2017) 411–416. arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-rpg.2016.0087, doi:https://doi.org/10.1049/iet-rpg.2016.0087.

[7] L. Wang, Z. Zhang, X. Luo, A two-stage data-driven approach for image-based wind turbine blade crack inspections, IEEE/ASME Transactions on Mechatronics 24 (3) (2019) 1271–1281. doi:10.1109/TMECH.2019.2908233.

[8] A. Reddy, V. Indragandhi, L. Ravi, V. Subramaniyaswamy, Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics, Measurement 147 (2019) 106823. doi:https://doi.org/10.1016/j.measurement.2019.07.051.

[9] M. Moradi, S. Sivoththaman, Mems multisensor intelligent damage detection for wind turbines, IEEE Sensors Journal 15 (3) (2015) 1437–1444. doi:10.1109/JSEN.2014.2362411.

[10] P. Arnold, J. Moll, V. Krozer, Design of a sparse antenna array for radar-based structural health monitoring of wind turbine blades, IET Radar, Sonar & Navigation 11 (8) (2017) 1259–1265. arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-rsn.2016.0355, doi:https://doi.org/10.1049/iet-rsn.2016.0355.

[11] H. Zhang, J. Jackman, A feasibility study of wind turbine blade surface crack detection using an optical inspection method, in: 2013 International Conference on Renewable Energy Research and Applications (ICRERA), 2013, pp. 847–852. doi:10.1109/ICRERA.2013.6749869.

[12] J. S. Pizarro Muñoz, Wind turbine blade damage identification using deep learning algorithms, Universitat Politècnica de València (2020).

[13] H.-c. Shin, H. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. Summers, Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning, IEEE Transactions on Medical Imaging 35 (02 2016). doi:10.1109/TMI.2016.2528162.

[14] C. Nebauer, Evaluation of convolutional neural networks for visual recognition, IEEE Transactions on Neural Networks 9 (4) (1998) 685–696. doi:10.1109/72.701181.

[15] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez, M. Papaelias, Condition monitoring of wind turbines: Techniques and methods, Renewable Energy 46 (2012) 169–178. doi:https://doi.org/10.1016/j.renene.2012.03.003.

[16] FaroWind, Blade inspection, https://farowind.com/ (2020).

[17] SkySpecs, Skyspecs autonomous drone inspections, https://skyspecs.com/ (2020).

[18] Cornis, Panoblade, https://home.cornis.fr/ (2020).

[19] F. P. García Márquez, D. Pedregal, C. Roberts, An algorithm approach to maintenance management, 2006.

[20] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, R. Reinhold Paulsen, Wind turbine surface damage detection by deep learning aided drone inspection analysis, Energies 12 (4) (2019). doi:10.3390/en12040676.

[21] Y. Yu, H. Cao, S. Liu, S. Yang, R. Bai, Image-based damage recognition of wind turbine blades, in: 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM), 2017, pp. 161–166. doi:10.1109/ICARM.2017.8273153.

[22] L. Zhang, F. Yang, Y. Daniel Zhang, Y. J. Zhu, Road crack detection using deep convolutional neural network, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3708–3712. doi:10.1109/ICIP.2016.7533052.

[23] A. Varghese, J. Gubbi, H. Sharma, P. Balamuralidhar, Power infrastructure monitoring and damage detection using drone captured images, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1681–1687. doi:10.1109/IJCNN.2017.7966053.

[24] J. G. C. M. Ivan Nikolov, Mikkel Nielsen, Wind turbine blade surfaces dataset, Mendeley Data (2020).
[25] C. M. Ivan Nikolov, Wind turbine blade sfm image capturing setups, Mendeley Data (2020).
[26] X. C. ASM SHIHAVUDDIN, Dtu - drone inspection images of wind turbine, Mendeley Data (2018).
[27] C. Martín, P. Langendoerfer, P. S. Zarrin, M. Díaz, B. Rubio, Kafka-ml: connecting the data stream with ml/ai frameworks, Future Generation Computer Systems 126 (2022) 15–33.