applied sciences

MDPI

Article

Estimation of Wind Turbine Angular Velocity Remotely Found on Video Mining and Convolutional Neural Network

Mahdi Bahaghighat [1,*], Qin Xin [2,*], Seyed Ahmad Motamedi [3], Morteza Mohammadi Zanjireh [4] and Antoine Vacavant [5]

[1] Department of Electrical Engineering, Engineering Faculty, Raja University, Qazvin 95834, Iran
[2] Faculty of Science and Technology, University of the Faroe Islands, Tórshavn FO 100, Faroe Islands
[3] Electrical Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran 15875-4413, Iran; motamedi@aut.ac.ir
[4] Computer Engineering Department, Imam Khomeini International University (IKIU), Qazvin 21333, Iran; Zanjireh@eng.ikiu.ac.ir
[5] Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France; antoine.vacavant@uca.fr
* Correspondence: m.bahaghighat@raja.ac.ir (M.B.); qinx@setur.fo (Q.X.)

Received: 2 March 2020; Accepted: 15 April 2020; Published: 19 May 2020

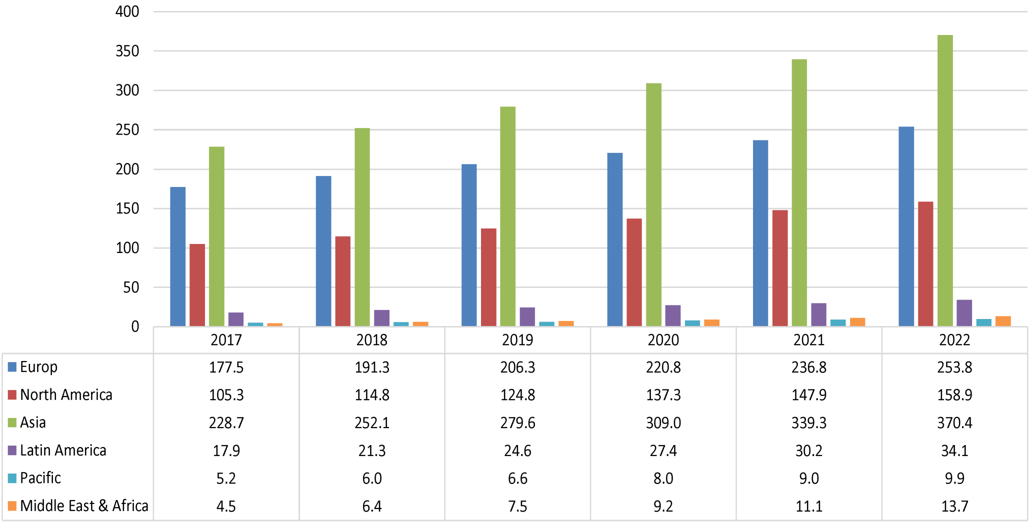
Abstract: Today, energy issues are more important than ever. Because of the importance of environmental concerns, clean and renewable energies such as wind power have been most welcomed globally, especially in developing countries. Worldwide development of these technologies leads to the use of intelligent systems for monitoring and maintenance purposes. Besides, deep learning as a new area of machine learning is sharply developing. Its strong performance in computer vision problems has conducted us to provide a high accuracy intelligent machine vision system based on deep learning to estimate the wind turbine angular velocity, remotely. This velocity along with other information such as pitch angle and yaw angle can be used to estimate the wind farm energy production. For this purpose, we have used *SSD (Single Shot Multi-Box Detector)* object detection algorithm and some specific classification methods based on *DenseNet*, *SqueezeNet*, *ResNet50*, and *InceptionV3* models. The results indicate that the proposed system can estimate rotational speed with about 99.05% accuracy.

Keywords: machine vision; deep learning; object detection; image classification; remote sensing; wind turbine; WTCM; angular velocity

1. Introduction

Energy is one of the most basic requirements in human life and has become even more important in our advanced world. In recent years, in terms of technology, economics, and environmental issues, renewable energy sources have been considered more than ever. Endless energy resources such as wind and solar energies are clean and can reduce environmental impacts [1,2]. Wind energy is one of the most important renewable energy sources and many countries are predicted to increase wind energy portion of their whole national energy supply to about twenty percent in the next decade [1–3]. Figure 1 shows wind farms development from 2017 to 2022 in the globe [2].

Compared to steam, hydro, and gas turbines used in traditional power plants, wind turbines (WTs) are usually operated in harsher environments and, therefore, have relatively higher failure rates. The faults in WTs can be classified into two categories: wear-out failures and temporary random faults. Wear-out failures are long-term and permanent events. Repairing or replacing a failed component needs additional costs and results in a loss of energy production. If a failed component is not identified and repaired or replaced in appropriate time, it may cause consequent failures of other components and even the entire WT system [4]. In [5], the authors reviewed non-destructive testing (NDT) for wind turbine inspections. It includes several techniques such as vision inspection [6], X-ray inspection, infrared thermography, and ultrasonic scanning that are used in different industries, to monitor the structures of materials, components or systems without causing damage to them.



	2017	2018	2019	2020	2021	2022
■ Europ	177.5	191.3	206.3	220.8	236.8	253.8
■ North America	105.3	114.8	124.8	137.3	147.9	158.9
■ Asia	228.7	252.1	279.6	309.0	339.3	370.4
■ Latin America	17.9	21.3	24.6	27.4	30.2	34.1
■ Pacific	5.2	6.0	6.6	8.0	9.0	9.9
■ Middle East & Africa	4.5	6.4	7.5	9.2	11.1	13.7

Figure 1. Cumulative wind energy market forecast by region 2018-2022 (GW) [2].

Real-time energy monitoring and its precise estimation can be very helpful for energy management and preventing unwanted failures in the power grid. In the emerging power networks, so-called smart grids (SGs) [6–8], advanced communication network infrastructures [8–14] along with two-way power flows provide appropriate conditions for precise services throughout the power generation chains. The energy exchange always requires accurate estimation of produced energy by distributed energy generators (DGs).

Condition monitoring (CM) is widely deployed in the wind farms to ensure WT normal operation found on detecting failures and planning for operations and maintenance services [15]. It is a practical tool massively employed for the early event detection of faults/failures in the wind turbine farms in order to maximize productivity and minimize downtime of the system. In order to increase the lifetime of the WT farm and reduce the maintenance cost, well accurate predictions of system faults and failures are needed [6,7]. In particular, in future grids, these predictions can be available based on advanced nondestructive testing (NDT) approaches such as intelligent Vision Inspection (VI). Integration of Sensor Networks, IoT, and industrial cameras can provide advanced structures to apply vision inspections (VI) for fault detection/diagnosis of wind turbine such as Mechanical Deformations, Surface Defects, Overheated Components in rotor blades, nacelles, slip rings, yaw drives, bearings, gearbox, generators, and transformers [6,7]. Based on our best knowledge, there are not noticeable researches that are concentrated on the integration of these methods based on video mining. In this work, we present a powerful method for measuring the rotational speed of wind turbines that can be used along with other information to estimate the amount of energy produced by the wind farm [16]. Providing infrastructure requirements for our proposed system such as camera networks and communication devices, it is also possible to benefit more vision inspection tasks and NDTs, like those mentioned above, in just one system.This article is structured as follows:

Related works are indicated in Section 2. In Section 3, our proposed model is presented in detail. Then in Section 4, simulation results are discussed in detail. Finally, Section 5 concludes the article and proposes future work.

2. Related Works

In [5], they focused on the NDT techniques applicable to Wind Turbine Condition Monitoring and Fault Diagnosis (CMFD) of wind turbine blades (WT-CMFD), but it did not discuss what blade failure modes or how they should be diagnosed by the NDT techniques [4]. In [17] a non-destructive testing was used to implement fault diagnosis for wind turbine blades. Ultrasonic scanning is one of the most favorite NDT techniques used in various industries [18,19]. It can be accomplished to find out whether observed damages are present in a WT blade structure [20,21]. Infrared thermography or IRT is the way to actually measure a surface heat distribution. It can be used to inspect the operational status of all blades based on measuring the temperature difference in the laminate and adhesive joints, which are the most critical points in their structures [22,23]. X-rays are a type of radiation called electromagnetic waves. It can penetrate into a wide range of materials [20,21,24–26]. It is possible to reveal materials' structural variations by X-ray images so these can be deployed to detect defects and faults in wind turbine component structures. For example in [20], the authors demonstrated an effective radiographic system to inspect WT blades [20]. The sonic waves tap test can be deployed to detect a disbond between the main spar and the skin laminate for a blade [27]. The NDT techniques are strong enough to explore incipient defects and fault propagation monitoring. However, most of the time, these implementations are expensive. Furthermore, NDT approaches are usually offline and intrusive techniques. It means that NDTs require stopping the current operations of the wind turbine for further measurements and analysis [19]. In Figure 2, some signal processing and intelligent methods are depicted for WT-CMFD.

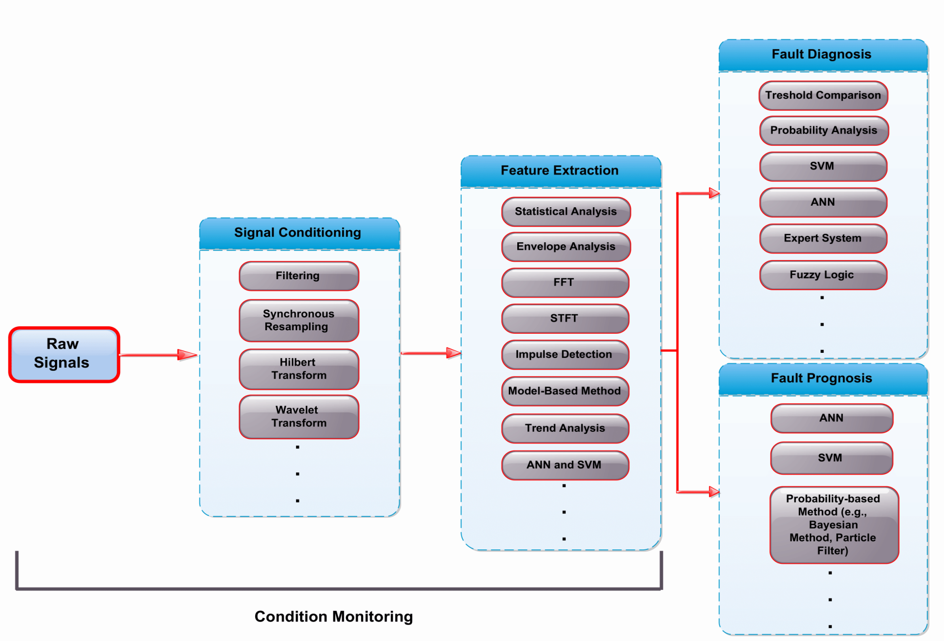


Figure 2. Signal processing and intelligent methods for Wind Turbine Condition Monitoring and Fault Diagnosis (WT-CMFD) and prognosis [19].

In [28], a research was introduced to smartly detect surface cracks of wind turbine blade based on unmanned aerial vehicles (UAVs) video stream. They used Haar-like features to detect crack regions. They trained an extended cascading classifier with some base models such as Support Vector Machine (SVM), LogitBoost and Decision Tree (DT) to solve this detecting problem. A comparison was performed to evaluate the success rate of their algorithm for both identification and localization of cracks [28]. In [29], they introduced a system for wind turbine inspection based on a multi-copter. The project was defined as object recognition and tracking problems. They used the Hough line transform to implement the feature extraction module for a wind turbine. The hub positioning was handled by an algorithm designed to identify the three-point star similarity and was tracked using the Kalman filter [29].

On the other side, the WT is not a perfect technology and it is flawed by its special biological impact. For example, it may pose a serious threat to some bird species and also bats [30]. Bird migrations in particular seasons may increase the risk of collision with WTs. It is an unwanted event that also can influence wind turbine operation and impose financial losses. Therefore, it is very important to perform real-time Vision Inspection (VI) of WTs in wide areas to avoid any collision thread [31].

The majority of wind turbines are embedded with gear and statistical reports show that damages due to gearbox account for at least the 20% of their unavailability time [15]. In [15], they tried to effectively address a novel and accurate method for wind turbine gearboxes fault detection based on both peak values and RMS of vibration signals. They presented well in their study that the overall accuracy of each CM technique depends on the physics of failure. Their studies suggested that a technology that benefits multiple techniques is essential for a holistic and reliable health assessment of wind turbine components [15].

In [32], an innovative investigation of jerk data extracted directly from vibration acceleration (VA) data of the WT gearbox was presented. The vibration magnitude can be measured by either displacement or velocity or acceleration. In [32], the vibration acceleration (VA) data were actually gathered. In order to do test the gearbox, several vibration accelerometers had mounted on various locations of the drive-train system without hub and blades to measure the VA. Found on the extracted jerk data, two important analyses can be addressed: failure identification of components and monitoring gearbox high-speed stage vibration. In their study, the gearbox high-speed stage failures were spotted, but some root causes could be inferred based on the pattern of data due to several specific sensors. In addition, it only discussed the monitoring model for the high-speed stage vibration and not for monitoring the vibration excitement of other parts of the gearbox [32].

In [33], a novel algorithm to make diagnosis of drive-train bearings damages was introduced. Their main idea was that vibrations can be measured at the wind turbine tower instead of at the gearbox without interrupting the WT current operation. Their main focus was to evaluate the vibrations at the tower rather than traditional approaches that do measure at the gearbox. The gearbox usually is located in the nacelle, which makes it hardly accessible. This provides a simpler and low-cost alternative solution. Besides, to have high-frequency and reliable rotor angular velocity measurements this kind of information along with the gearbox geometry are crucial to getting a precise damage location identification [33].

The vision inspection (VI), as an NDT approach, can be an alternative to measure vibration [34]. For example, in [35], a new active sensing methodology based on the visual inspection was introduced to determine dynamic properties of a vibrating object. They deployed high-frame-rate stereo video to observe the 3D vibration distribution of a monitored object under undisclosed excitations. Generally, it was proved that such algorithms could detect vibration considering the quickness and the simplicity without human interaction. The main point is that it neither affect nor interrupt the current operation of the WTs. This could encourage the integration of vibration monitoring system in the maintenance regimes of WT farms, to ensure minimal downtimes and higher reliability in the future farms.

3. Proposed Model

In this section, we present our proposed algorithm to estimate the turbine blade rotational velocity. For this purpose, a wind turbine tower in a wind turbine farm is depicted in Figure 3. In this figure, two special objects are distinguished: hub and blade. First and foremost, we should locate the wind turbine tower (WTT). From a morphological point of view, the hub object is the best and most distinctive part of a WTT for localization. Therefore, the main step is the good estimation of its position with the highest accuracy. After finding the hub in an input image, it is time to get a sub-image at the predefined angle and distance pairs from its center. This sub-image is then analyzed by some proposed binary classification algorithms to determine whether a blade object is presented or not. The described process should be performed on several consecutive video frames, then the resulting data sets should be analyzed for a given time window to estimate the angular velocity based on gathered information.

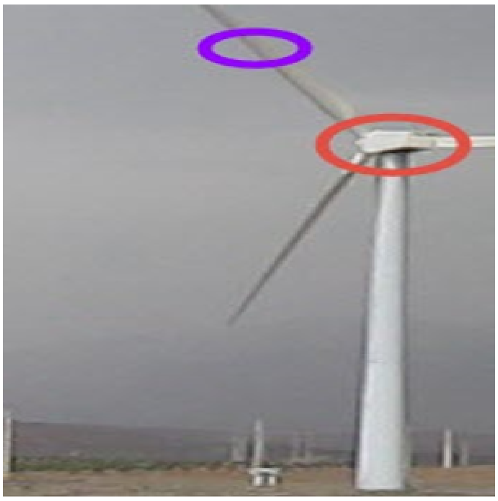


Figure 3. The hub and a part of one blade in a sample wind turbine tower image.

3.1. Hub Object Detection

There are various problems in computer vision and image processing [36–40]. Object detection, as one of the most applicable problems in computer vision, is the process of categorizing especial objects such as cars, pedestrian, bicycles, traffic signs, faces, etc. [36,37]. Similar to many computer vision tasks, related algorithms can be divided into two categories: feature-based object detection [41] and Deep learning (DL) object detection (OD) [42]. DL is considered a special type of machine learning (ML). Both ML and DL commence with two training and test data with a model to probe the weights that make the model best fit the data [36,43–45]. Nowadays, deep learning has experienced a sharp adoption rate in wide applications [7,36,44]. It can be used in emerging traffic intelligent services, self-driving cars, image retrieval, security, sports teams analysis to generate heat maps for players, crowd video analysis, implementation of intelligent quality control (QC) systems [36,37,45,46] for smart manufacturing, etc.

In [47], the authors presented an innovative algorithm called Single Shot Multi-Box Detector (SSD) to implement object detection using a single deep neural network combining regional proposals and feature extraction. It discretizes bounding boxes output space into a set of default boxes over different aspect ratios and scales per feature map location. This set is used and applied to the feature maps. Since they are computed by passing an image through an image classification network, the feature extraction for the bounding boxes can be extracted in just a single step. Scores are generated for each object category in every of the default bounding boxes. In order to better fit the ground truth boxes, adjustment offsets are calculated for each box. In addition, this network merges all predictions obtained from several feature maps with divers resolutions to handle large number of objects with various sizes. In fact, SSD strongly benefits from multi-scale feature maps, which are generated from a backbone network such as VGG-16 [48] to detect objects having various sizes. The SSD fully eliminates proposal generation and subsequent pixel or feature resampling stages. It also encapsulates all computations in just one network. As a result, it is easy to train SSD and easy to integrate into other systems. Experimental outcomes on the ILSVRC , PASCAL VOC, and COCO datasets verify that in comparison to methods that utilize some additional object proposal steps, the SSD not only has competitive accuracy but also has much lower runtime while providing a consistent framework for training and inference. Figure 4 illustrates SSD architecture [47].

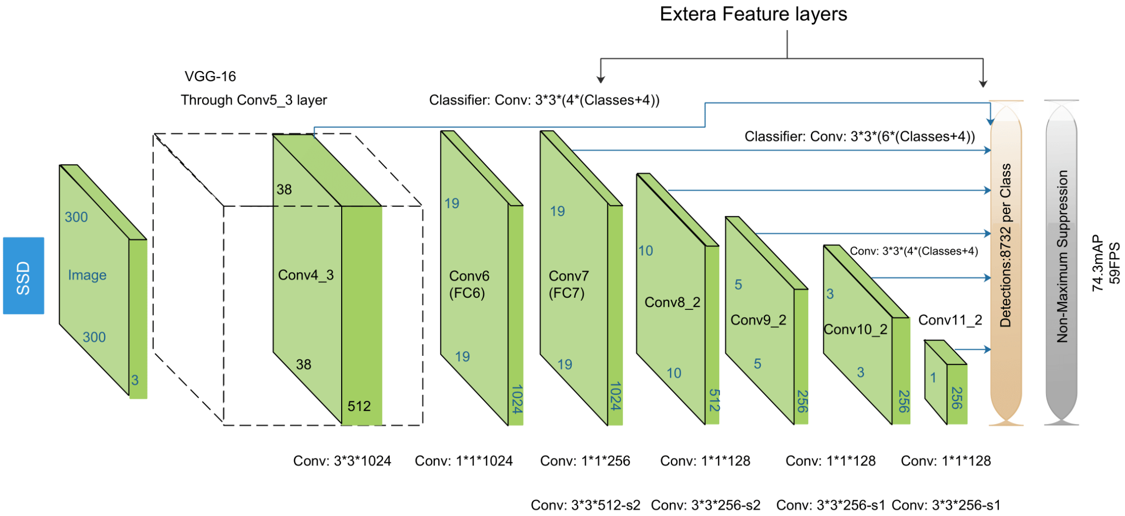


Figure 4. Single Shot Multi-Box Detector (SSD) architecture [47].

Now, SSD is one of the top object detection algorithms in both aspects, accuracy and speed [49]. In [49], how to use feature maps effectively to improve the performance of the conventional SSD was proposed and analyzed.

In this work, we deploy the special version of SSD which is called SSD7. It is a smaller 7-layer version that can be trained from scratch relatively quickly, even with an ordinary GPU, while it is capable enough to solve less complex object detection tasks. The idea is just to build a low-complexity network that is fast for our purposes. Table 1 shows some primary configuration of our trained model.

The first step of our algorithm is the hub detection. We propose to use SSD7 object detection to localize a hub object in the input image. Figure 5 depicts some selected hubs and blades from our dataset.

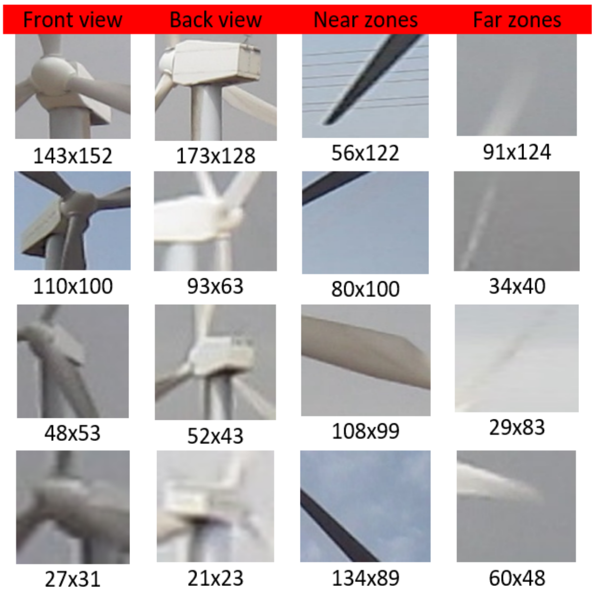


Figure 5. Some selected samples of hubs and blades images at original sizes from our dataset: hubs (front view and back view), blades (near zones and far zones).

Table 1. Primary configurations for SSD7.

Parameter	Values	Description
H	300	Height of the input images
W	480	Width of the input images
iC	3	Number of color channels of the input images
n_classes	1	Number of positive classes
Scales	[0.08, 0.16, 0.32, 0.64, 0.96]	An explicit list of anchor box scaling factors.
AR(aspect ratios)	[0.5, 1.0, 2.0]	The list of aspect ratios for the anchor boxes

Selecting the Best Object

Usually, the object detection algorithm proposes several candidates, which should be filtered to select the best one. We define D as a set denoting all information including scores (S_i), positions(x_i, y_i) and bounding box dimensions (h_i, w_i) for all detected candidates. Therefore, we can write D as below:

$$D = \{D_i | i = 1, 2, .., N; D_i = (S_i, x_i, y_i, h_i, w_i)\} \tag{1}$$

where N is the total number of objects detected for the current image. In order to suppress weak objects, first we should sort D regarding to object scores to get D_s:

$$D_s = sort(D|S_i, i = 1, 2, ..N) = \{D_j | j = j_1^*, j_2^*, ..., j_N^*\} \tag{2}$$

To obtain it, we define Equation (3):

$$I_k = j_k^* = Argmax(S_i | i = 1, 2, .., N, i \notin J_{k-1}^*) \tag{3}$$

where j_k^* is the argmax for k'th iteration, and we can write Eqation (4):

$$J_0^* = \phi, J_1^* = \{J_0^*, j_1^*\}, J_2^* = \{J_1^*, j_2^*\}, ..., J_N^* = \{1, 2, .., N\} \tag{4}$$

Sometimes there are some detections even with high scores while they are false positives (FPs). In order to reject these types, we apply dimensional filtering for objects in sorted detection set D_s. According to Equation (5), the *best object* **(Bo)** can be found by selecting the lowest amount of k, which satisfies both conditions $h_{j_k^*} > k_h \times \bar{H}$ and $w_{j_k^*} > k_w \times \bar{W}$:

$$\begin{gathered} Bo = j_k^* | h_{j_k^*} > k_h \times \bar{H} \qquad \&\& \qquad w_{j_k^*} > k_w \times \bar{w} \\ for \quad min(k), k \in \{1, 2, .., N\} \end{gathered} \tag{5}$$

$$\bar{H} = 1/N \times \sum_{i=1}^{N} h_i \tag{6}$$

$$\bar{W} = 1/N \times \sum_{i=1}^{N} w_i \tag{7}$$

3.2. Blade and Non-Blade Binary Classification

After locating the position of the hub in the input image, the positioning information $\{\hat{W}_h, \hat{H}_h, \hat{x}_h, \hat{y}_h\}$ is estimated with the confidence scores. We select a random point with distance r and angle θ relative to the hub center $(\hat{x}_h, \hat{y}_h)$. Distance is calculated based on Equations (8)–(10):

$$r = \sqrt{x_r^2 + y_r^2} \tag{8}$$

$$x_r = \hat{x}_h + k_x \times \hat{W}_h, k_x > 0 \tag{9}$$

$$y_r = \hat{y}_+ k_y \times \hat{H}_h, k_y > 0 \tag{10}$$

According to Equation (11), θ is supposed to have a random uniform distribution, and it will be selected randomly between $\theta 1$ and $\theta 2$:

$$f_\theta(\theta) = \begin{cases} 1/(\theta 2 - \theta 1) & \theta 1 \leq \theta \leq \theta 2 \\ 0 & otherwise \end{cases} \tag{11}$$

Based on these equations the pair (r, θ) is estimated then a sub-image named I_h is masked at this point. Now, for the resulting image I_h, with dimensions (W_i, H_i), the blade/non-blade classification algorithm can be run. From the morphological point of view, a blade and consequently a random part of it are directional objects. In traditional approaches, there are a lot of methods such as Radon transform [37,45,50] and Gabor wavelet transform (GWT) [6,51–53] that can be used for directional pattern analysis. In our previous work [6], we applied GWT to solve the issue. However, these approaches have deficiencies and suffer from scale and illumination variations. Therefore, in this work, we develop our classification model based on deep neural networks [54]. To tackle this issue, we define a binary classification problem. We propose to deploy transfer learning and fulfill the task by fine-tuning some strong classification models such as *DenseNet*, *SqueezeNet*, *ResNet50*, and *InceptionV3* for our developed dataset including 10,000 images with high precision. This dataset includes 5000 blade images with diverse angles (0 to 360 degrees), different thicknesses and lengths, as well as 5000 non-blade images. Some blade samples are depicted in Figure 5. In Figure 6, histograms of both height and width of the blade images in our dataset are shown.

As said before, some of the more powerful methods are used and compared in our work.

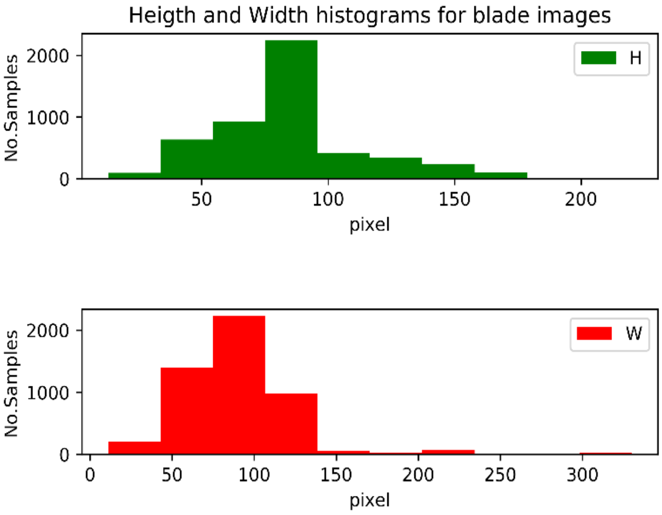


Figure 6. Distribution of height and width for 5000 blade sub-images in our dataset.

3.2.1. ResNet

ResNet or Residual Network is a well-known classification algorithm based on residual learning. In general, in a deep CNN, a lot of network layers are stacked together. The training network tries to learn features as much as possible at the end of its layers in different levels (Low/Middle/High). However, in this method of learning, learning some features is replaced by learning some residual. Residual can be defined as subtraction of feature learned from the input of that layer. ResNet fulfill the task by establishing shortcut connections from one layer to some other layers. It has been proved that residual learning is easier than training simple deep convolutional neural networks [36] and also the problem of degrading accuracy is resolved. During these recent years, in order to solve more complex tasks and also to improve the accuracy of classification algorithms as much as possible, there is a rigid trend to dive into deeper and deeper networks. However, as it goes deeper, training of the neural network becomes difficult, and as a result, the accuracy starts saturating and then degrades. Residual Learning tries to solve both these problems [55,56]. In this paper, we deploy ResNet50 which is a 50 layers Residual Network to solve our classification problem.

3.2.2. DenseNet

Densely Connected Convolutional Network (DenseNet) was introduced in [57]. Outstanding results on image classification tasks are obtained by DenseNet. The idea of DenseNet is based on the observation that if each layer is directly connected to every other layer in a feed-forward fashion then the network will be more accurate and easier to train [58]. Due to all improvements applied on DenseNets architecture, it has one of the lowest error rates on CIFAR/SVHN datasets [59]. Although DenseNets have several compelling advantages, they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters; they struggle to make use of more and more memory. In comparison to Wide ResNet, DenseNet is less memory/speed-efficient. Figure 7 shows the basic structure for classification based on DenseNets.

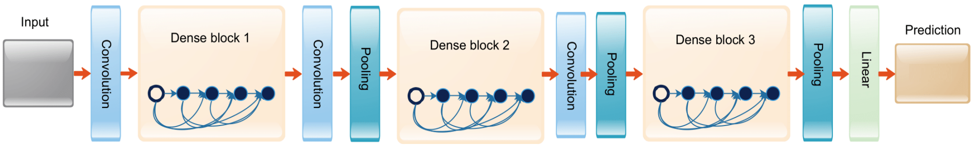


Figure 7. Classification based on Densely Connected Convolutional Network (DenseNet) [57].

3.2.3. InceptionV3

One example of very deep convolutional networks is InceptionV3. It is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset [56,60]. The model is the culmination of many ideas developed by multiple researchers over the years. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batch-norm is used extensively throughout the model and applied to activation inputs. The loss is also computed via Softmax [60].

3.2.4. SqueezeNet

The last classification approach that is used in our work is called SqueezeNet. It benefits from a light architecture as well as a quantitative analysis. Comparing to AlexNet, SqueezeNet can be three times faster and 500 times smaller for the same accuracy. It deploys point-wise filters and 1×1

filters as a bottleneck layer in its structure [55,61]. Although it is very popular and its application is developing [62,63], its performance for solving our problem is disappointing and not comparable to others.

3.3. Blade Angular Velocity Estimation Method

In the last section, our blade detection approach was presented. The object detection step leads to blade score S_B that can be used for decision making between presence or absence of a blade in the k'th frame. The final decision can be simply made by comparing S_B to S_thr. If it is greater than or equal to S_thr, it will mean that a blade has been detected in the input image. According to Equation (12), non-zero $S_B(k)$ indicates absolutely that a blade is detected for frame index k.

$$S_B(k) = \begin{cases} S_thr & S_thr \leq S_B \\ 0 & otherwise \end{cases} \tag{12}$$

The procedure must be recalled for all consecutive image frames in the observation time, T_w. As a result, all needed information for the estimation angular velocity is gathered according to Equation (13):

$$S_B^T w = \{S_B(i) | i = 1, 2, .., N\}, N = T_w f_s \tag{13}$$

where in the last equation f_S would be the frame rate of the input video in frames per second (fps). Finally, the velocity of blade, V_B, can be calculated by our proposed formula:

$$V_B = 2\pi (W^i f_s)^- 160(RPM) \tag{14}$$

where W^i is the dominant frequency of $S_B^T w$ sequence.

4. Experimental Results

In this study, simulation results have been obtained by means of Python programming language and TensorFlow. We deployed the NVIDIA GTX 1050 GPU with a Core i7 CPU and 16 GB DDR4 RAM memory. As indicated earlier, we conducted our experiments using images which were captured by us in a real wind farm.

4.1. Dataset

In our research, two types of dataset have been developed. The first type includes still images (snapshot images), and the second one includes the video dataset. We developed two snapshot image datasets named DB1 and DB2 along with the video dataset named DB3. DB1 includes 6000 images that have been captured by us from different angles, distances, and backgrounds in a real wind turbine farm. In this wind turbine farm, more than 300 WTTs exist, and on average, 20 images/WTT are available in DB1. This dataset is used for the object detection problem (Hub detection). In addition, DB2 includes about 10,000 snapshot images uniformly distributed in two different classes (Blade/Non-Blade) for the second problem (Binary classification). DB2 includes 5000 blade images with diverse angles (0 to 360 degrees), different thicknesses and lengths, as well as 5000 non-blade images (please see Figure 5). For velocity estimation, we have deployed our third dataset DB3 including 4 video files. These four videos were captured with 30 frames per second (fps) with different distances and angles from a shared object (a wind tower target), synchronously. Each of these includes 3600 images with 120 s time duration. It is worth mentioning that we measured the ground truth velocity (reference rotational speed) in a time window about 120 s (time duration of each video file) using Long Range Finder Sensor UART(TTL) with USB RB605B 703A U81. The resolution of our B605B laser distance meter sensor was 1mm, with measuring range from 0.03 to 100 meters for constant conditions.

4.2. Hub Object Detection

In this section, we intend to examine the result of applying the *SSD7* algorithm to find the hub object in an input image. For the learning process, the bounding box ground truth information for all images should be created and applied to the algorithm. To handle that, we use the *labelImg* software [64] to label our image dataset. For this purpose, first we created a data set including 10,000 images; then, for convenience, we randomly sub-sampled it to have just 1000 images. Then, we augmented all these images by labelImg to feed bounding box information to SSD7. Table 2 shows the results of the implementation of the SSD7 algorithm on this reduced dataset. In this experiment, in order to reduce the time overhead, the dimensions of the input images from the data are reduced to the values $(300, 480)$, then the algorithm is run for *confidence threshold* = 0.3 and *Intersection Over Union threshold (**IoU**)* = 0.5. In the literature, there are some measurement metrics such as *True Positive* **(TP)**, *True Negative* **(TN)**, *False negative* **(FN)**, and *False Positive* **(FP)** that are widely used for object detection problems [65,66]. TP indicates the total number of samples that are classified correctly. FN means the number of samples belonging to the class set but incorrectly classified. TN represents the number of correctly classified samples not belonging to the class set, and FP refers to the number of unclassified samples that have been misclassified. Based on defined parameters, accuracy can be calculated using Equation(15) [66]:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{15}$$

In Table 2, the results for *Accuracy* **(ACC)**, and *confidence mean* (**Conf** or **mAP** *(mean Average Precision)*) are obtained in a state where the *Non-Max Suppression* **(NMS)** [67] is also implemented. Now, to determine the best configuration, we change the *epoch* value from 20 to 600. For $epoch = 20$, we have some false negatives and a lot of false positives, while it increases to 60, the FPs are significantly reduced, but the FN is very high. While the *epoch* is equal to 200, we have no FN with a few FPs. After that, with the *epoch* incerement from 200 to 600, the results are not improved (the validation loss and training loss for $epoch = 600$ are shown in Figure 8). As we can see in Table 2, the best accuracy with the highest confidence mean is obtained for the $epoch = 200$ where $ACC = 99.5\%$ and $Conf = 0.6699$ are satisfied. Hereafter, we consider the *epoch* value to be 200.

Table 2. Results of proposed hub object detection based on SSD7. Input image: $[(h, w) = (300, 480)]$.

Epochs	Val loss	ACC (%)	Conf (mAP)
20	3.5099	49.5	0.5119
60	2.5054	82.5	0.4949
200	1.9740	99.5	0.6699
400	2.1127	97.1	0.5368
600	2.2741	95.5	0.5170

Now, the impacts of two important parameters including $confidence\ thresh$ and IoU on accuracy should be evaluated. For this purpose, these two parameters are changed according to Table 3, and at each stage, the run time or *Prediction Time per Image* **(PTPI)** is calculated with ACC. Obtained results show that for a high confidence level, the FN value will be very high ($FN = 156$ and 160 for *confidence thresh* = 0.7 with different IoUs). For the $confidence\ thresh$ = 0.7, the ACC is obtained at 22% and 20% when $IoU = 0.50$ and 0.35, respectively. In comparison, for $confidence\ thresh$ = 0.3, the accuracy is increased sharply to 99.5%. Further reduction of the $confidence\ thresh$ from 0.3 to 0.22 can lead to an increment on FP and consequently a decrease of ACC. In addition, the average run time for hub detection is estimated at about 0.01594 s.

Table 3. Evaluation of confidence threshold and Intersection Over Union threshold (IoU) impacts on Accuracy (ACC), Prediction Time per Image (PTPI), and Conf. Input image: $[(h, w) = (300, 480)]$.

Confidence Threshold	IoU	ACC (%)	PTPI(sec)	Conf(mAP)
0.22	0.50	99.0	0.01450	0.5561
0.22	0.35	99.0	0.01316	0.5664
0.30	0.50	99.5	0.01500	0.6699
0.38	0.50	89.5	0.01599	0.5734
0.51	0.35	64.5	0.01453	0.6466
0.51	0.50	67.5	0.01479	0.6668
0.70	0.50	22.0	0.02250	0.7435
0.70	0.35	20.0	0.01710	0.7546

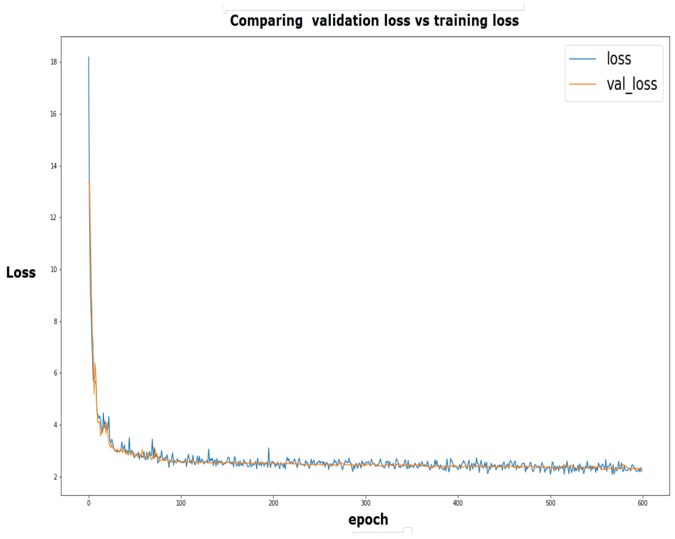


Figure 8. Comparing validation loss and training loss for SSD7 object detection ($epoch$ = 600).

In Figure 9, the effect of the changes of two parameters on ACC is well illustrated. The results indicate that the higher the $confidence\ thresh$, the lower the accuracy while the effect of IoU variation on accuracy is not very noticeable. Based on achieved results, we set the $confidence\ thresh$ and IoU values to be 0.3 and 0.50, respectively, in our work.

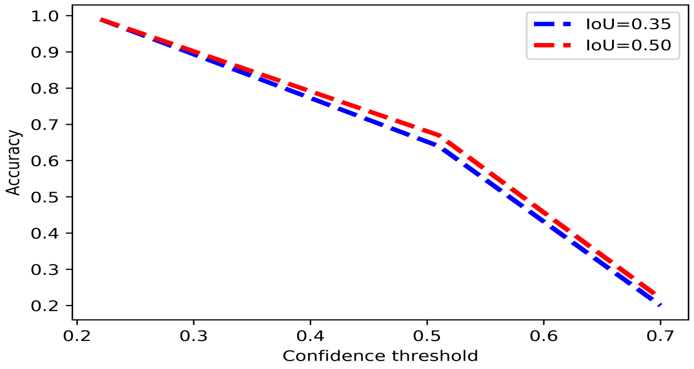


Figure 9. Evaluation of confidence threshold and IoU impacts on the accuracy.

In Figure 10, the results of hub object detection based on the SSD7 algorithm with $confidence\ thresh$ = 0.21 are shown for two sample images from our dataset. In these experiments, for testing, the NMS algorithm is not applied, so several hub objects are identified and located for just an image.

In this figure, all objects are shown with their confidence factors, and there are neither FN nor FP. Results indicate that almost all estimated bounding boxes are well matched to hub positions.

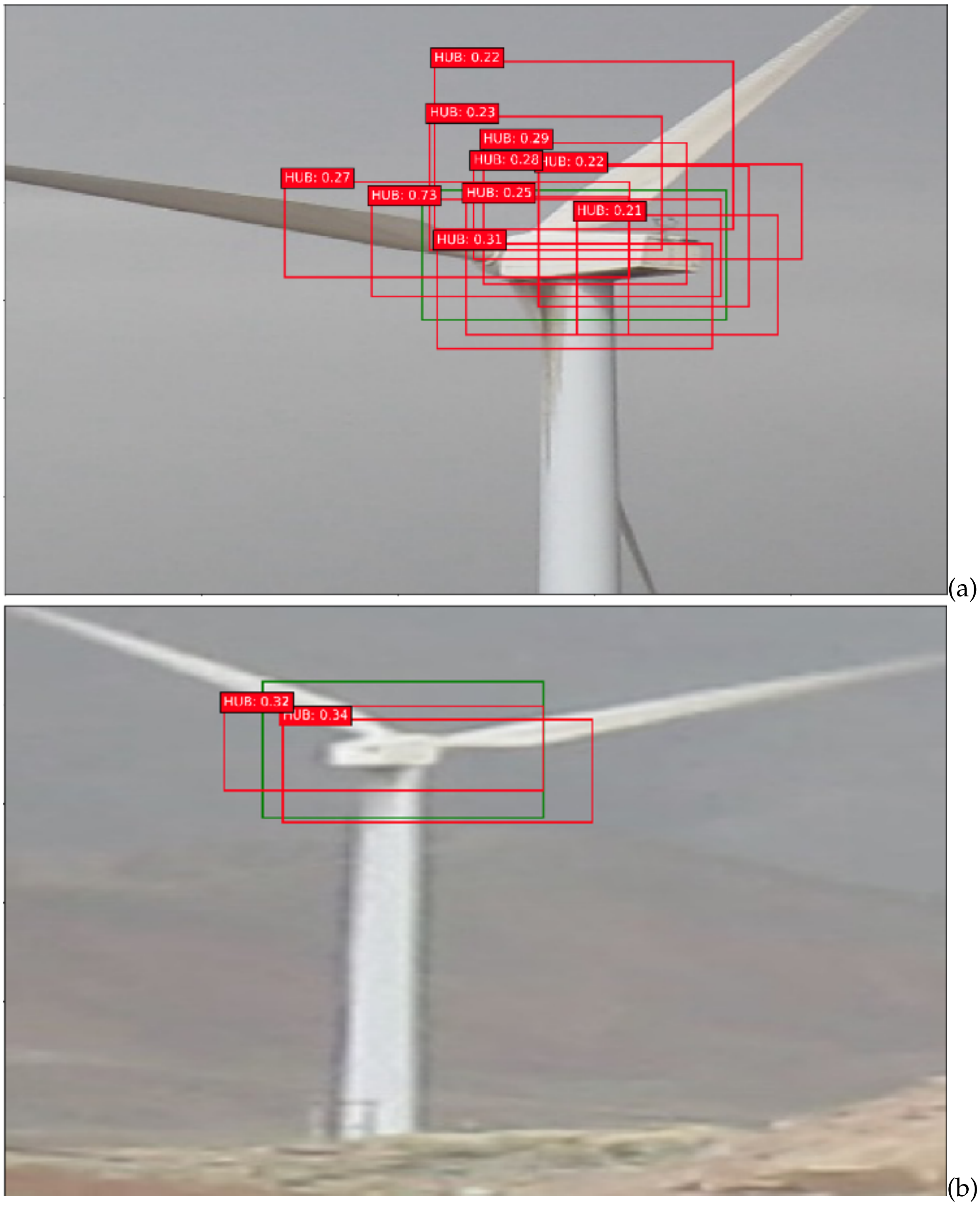


Figure 10. Detection of the hub with different accuracies without NMS (Green box: Ground truth box). (**a**) A near-field sample (**b**) A far-field sample.

In Figure 11, another example is shown where some false positives can be seen for new sample images. False positives can be either due to the low accuracy of the algorithm or the lower threshold for confidence. Therefore, it can be expected that with increasing $confidence\ thresh$ or improving accuracy, this problem will be solved. In addition, it is worthy to mention that in our simulation, in most cases, false positives exist along with the true detection. Actually, this means that the main object can easily be discriminated by filtering false estimations. If we add NMS to the algorithm, we can filter all estimated bounding boxes with low confidence scores. For example, in Figure 11a the bearing box bonding with $confidence = 0.52$ and in Figure 11b the bounding box with a $confidence\ =\ 0.44$ can accurately extract the actual position of the hub object.

Finally, in Figure 12, new scenarios with false negatives are displayed for two low-quality images. In Figure 12a,c, because of the low image quality and also the high amount of $confidence\ thresh$, our proposed object detection algorithm fails to detect the hub. To solve the issue, the threshold value is reduced and adjusted to 0.3. New results in Figure 12b,d clearly indicate that true positions can be estimated even for low-quality images. It is worth mentioning that the results of recent figures are illustrated after the NMS.

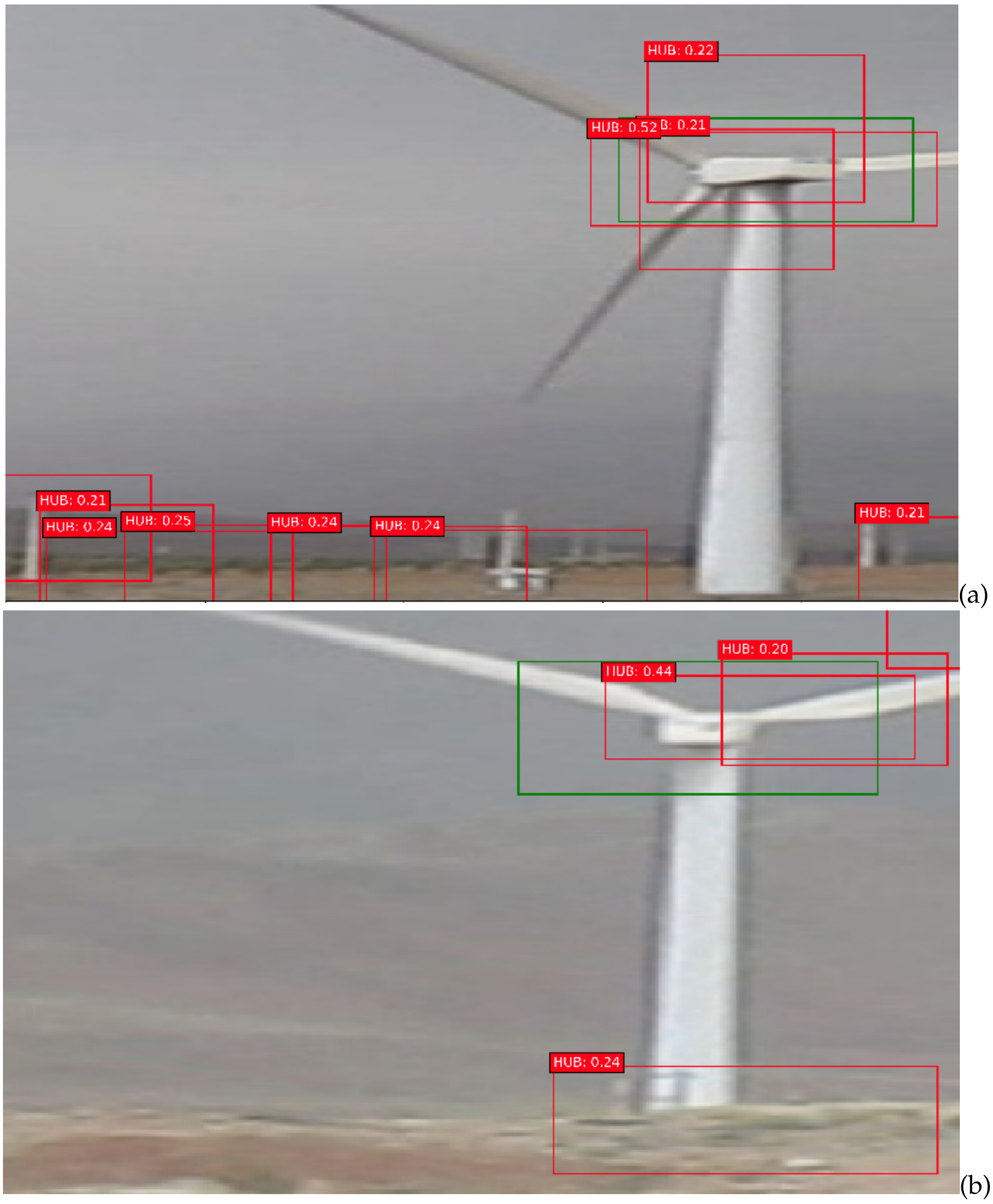


Figure 11. False positive detection without NMS ($epoch$ = 10, $confidence\ thresh$ = 0.20, IoU = 0.50) (**a**) The first sample (**b**) The second sample.

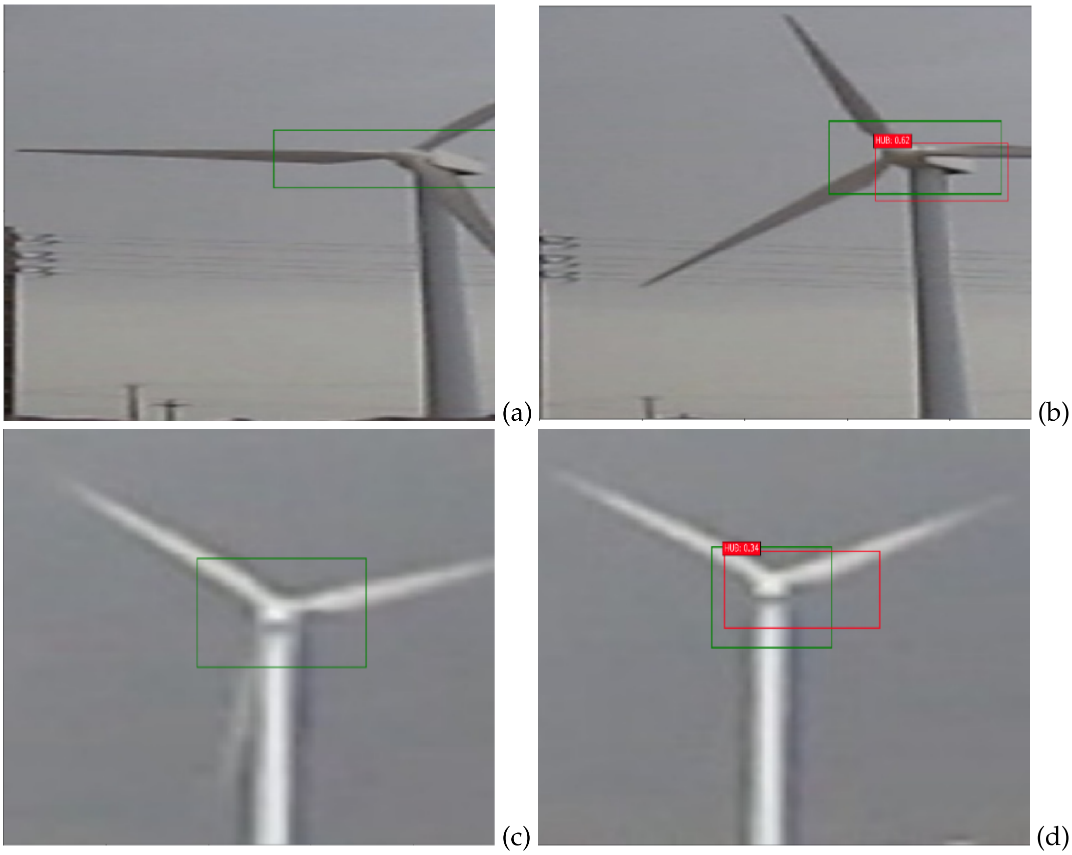


Figure 12. Solving false negative detection problem. (**a**,**c**) False negatives with $epoch$=10, $confidence$ $thresh$ = 0.70, IoU = 0.50. (**b**,**d**) True positives with $epoch$ = 10, $confidence\ thresh$ = 0.3, IoU = 0.50 (with NMS).

4.3. Blade/Non-Blade Classification

After elaborating the proposed hub object detection, we are going to evaluate the proposed algorithm for binary categorization to identify a blade. For this purpose, as mentioned in Section 2, we use powerful pre-trained models such as *DenseNet*, *SqueezeNet*, *ResNet50*, and *InceptionV3*. In this work, we removed the last layer of mentioned models and added a fully connected layer with two outputs to solve our binary classification problem. Our first experiment is conducted based on the ReNet50 model with 10 epochs fed by 1000 images with dimension of 32 × 32 pixels. The achieved results for both accuracy and loss curves for train and test phases are shown in Figure 13. The results indicate that ResNet50 accuracy is more than 90% after the 10th iteration.

Then, for a more complete review of the effect of various parameters on our classification accuracy, ten experiments have been performed. In these experiments, the variable parameters are: *database size* (1000 and 10, 000), *image size* (32 × 32, 64 × 64, and 480 × 480), *epochs* (10, 20, 50, and 100) for four selected models. The implementation results are presented in Table 4. It indicates that after the tenth epoch, the DenseNet fine-tuned model, which is trained on our dataset with 10,000 images can achieve the highest validation accuracy of about 0.9912 among all scenarios.

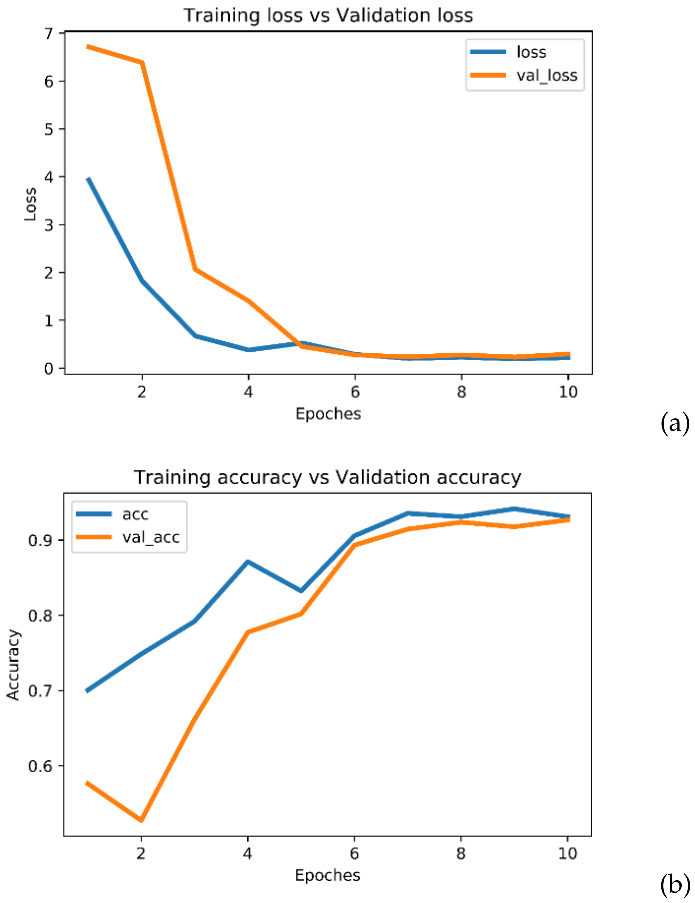


Figure 13. Accuracy and loss graphs for ResNet50 for Blade/Non-Blade classification. (*image size* = 32 × 32, *database size* = 1000, *epoch* = 10). (**a**) The loss (**b**) The accuracy.

Table 4. Comparison of four pertained classification approaches for Blade and Non-Blade binary classification problem.

Model Name	Database Size	Image Size	Epochs	val_loss	val_acc
DenseNet	10,000	(32, 32)	10	0.1085	0.9912
DenseNet	1000	(32, 32)	10	0.7901	0.9421
SqueezeNet	1000	(32, 32)	10	0.6931	0.5000
SqueezeNet	1000	(32,32)	20	0.6931	0.5030
SqueezeNet	1000	(32, 32)	100	0.6931	0.5000
SqueezeNet	1000	(64, 64)	10	0.6931	0.4970
SqueezeNet	1000	(480, 480)	10	0.6931	0.5020
ResNet50	1000	(32, 32)	10	0.2938	0.9268
InceptionV3	1000	(32, 32)	10	0.1757	0.9573
InceptionV3	1000	(32, 32)	100	0.2599	0.9360
InceptionV3	1000	(32, 32)	50	0.1987	0.9543

4.4. Angular Velocity Estimation

We used our sample videos recorded in DB3, to evaluate the average accuracy of the angular velocity estimation for the proposed system. Our algorithm is loaded by all these 3600 images/video but for better visualization, obtained results for just 200 consecutive images are plotted in Figure 14. The achieved results for different models with different precision (0.99, 0.93, 0.78, and 0.60) are compared in this figure. In addition, in Figure 14b, the DenseNet implementation result with *accuracy* = 99% is given. As we can see in this figure, our proposed method is able to extract a mono tone and soft signal that precisely indicate the frequency of the wind turbine rotation. Table 5 makes a comparison between estimated wind turbine velocity ($\hat{V}_B$) and true one (V_B) (in RPM) for our proposed model and [6]. It clearly indicates our new algorithm can achieve about 99.05% accuracy.

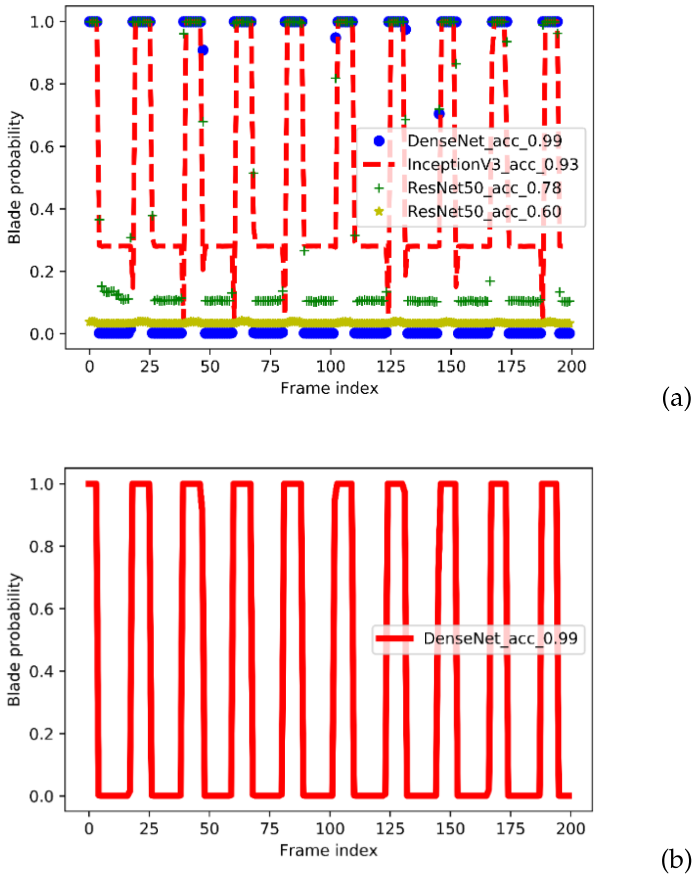


Figure 14. Final results of the wind turbine velocity estimation. (**a**) Comparison of different models with different accuracies. (**b**) Accurate estimation of the velocity based on DenseNet with *accuracy* = 99%.

Table 5. Comparison of the average accuracy of the velocity estimation algorithm found on DB3.

Method	V_B	$\hat{V}_B$	Average Accuracy
Pearson's correlation coefficient + Gabor Wavelet Transform [6]	68.25	50	73.26%
Proposed (Densnet + SSD7)	68.25	68.22	99.05%

5. Conclusions and Future Work

Nowadays, the generation of electric power from natural resources such as wind and solar is of great salience due to their clean and renewable characteristics. Smart grid technology has been targeted to respond efficiently to the increasing electric power demand on these resources. In this study, we introduced our novel system to estimate the angular velocity of the WT based on computer vision, deep learning, and video mining. The Single Shot MultiBox Detector (SSD) as our proposed object detection algorithm should be run to locate the hub position. We deployed the special version of SSD, which is called SSD7. It is a smaller 7-layer version that had been trained from scratch in our implementations. The simulation results showed that it was capable enough to solve the object detection tasks. As a result, the accuracy of about 99.5% achieved during the run time/Prediction Time per Image (PTPI) was just 0.01500 s. Then, at the appropriate angle and distance from the hub center, our proposed binary classification algorithms such as DenseNet, SqueezeNet, ResNet50, and InceptionV3 should be deployed to distinguish between a blade and non-blade object. The results obtained indicate that DenseNet (Densely Connected Convolutional Networks) fine-tuned model, which was trained on our dataset with 10,000 images, can achieve the highest validation accuracy, about 99.12%, among all other classifiers. After executing the classification algorithm, the blade probability is recorded for the current frame. Then, this process is repeated for all frames in a specified time window, and it can be used to estimate the angular speed. The comparison between our proposed model and [6] indicates that our new algorithm can achieve about 99.05% accuracy. Besides, The run time (Prediction Time per Image (PTPI)) for SSD7 algorithm varied from 13 ms to 22 ms depending on different conditions. The prediction time for the highest accuracy (confidence threshold = 0.30, IoU = 0.50, ACC = 99.5%) is about 15 ms. In our study, the prediction time of the DenseNet was also about 16 ms. Considering the video frame rate to be 30 fps, it means that inter-frame time spacing is about $1/30 = 0.0334$ s (33.4 ms). As a result, it needs 15 ms+16 ms = 31 ms for the processing time, and it is well below the 33.4 ms as the inter-frame time spacing. This shows that our approach is definitely practical for real world applications.

Demand response management systems need to have accurate predictions of energy power continuously. The proposed angular velocity approach along with other information such as pitch angle and yaw angle can be used further to estimate the wind farm energy production. Besides, historical figures of wind power production are not accessible when a new farm is going to generate power. Wind power is a function of weather variables, and it is likely that weather patterns of the new station are similar to some existing operational wind farms. It will thus be interesting to investigate how the forecast/prediction models of the existing wind farms can be adapted to generate a prediction model for new stations. This intelligent allocation based on our predicted information may lead to saving money in new projects, which would be the interesting research to do in future work.

The yaw system of wind turbines is the component responsible for the orientation of the wind. Yaw mechanism is used to turn the wind turbine rotor against the wind. We are going to add the ability to our algorithm to detect the wind turbine's yaw angle. By comparing the estimated velocity and yaw angle for several towers within a farm simultaneously, precise adjustment of the wind turbine can be applied in order to maximize the wind farm electric power production. Furthermore, computer vision based yaw angle measurement would make redundancy when system faces nacelle faults.

Author Contributions: Conceptualization, M.B., and Q.X.; methodology and software, M.B. and Q.X.; validation, M.B., M.M.Z., S.A.M., A.V., and Q.X.; formal analysis, M.B., and Q.X.; investigation, M.B., A.V., and Q.X.; writing—original draft preparation, M.B.; writing—review and editing, M.B., S.A.M., M.M.Z., A.V., and Q.X.; visualization, M.B., M.M.Z., and Q.X.; supervision, M.B.; project administration, M.B.; funding acquisition, Q.X. All authors have read and agreed to the published version of the manuscript .

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moccia, J.; Arapogianni, A.; Wilkes, J.; Kjaer, C.; Gruet, R.; Azau, S.; Scola, J. Pure power. wind energy targets for 2020 and 2030. *INIS* **2011**, *43*.
2. Lauha, F.; Steve, S.; Sgruti, S.; Limig, Q. *Global Wind Report Annual Market Update 2013*; Global Wind Energy Council: Brussels, Belgium, 2012.
3. Li, D.; Ho, S.C.M.; Song, G.; Ren, L.; Li, H. A review of damage detection methods for wind turbine blades. *Smart Mater. Struct.* **2015**, *24*, 033001. [CrossRef]
4. Qiao, W.; Lu, D. A survey on wind turbine condition monitoring and fault diagnosis—Part I: Components and subsystems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 6536–6545. [CrossRef]
5. Drewry, M.A.; Georgiou, G. A review of NDT techniques for wind turbines. *Insight-Non-Destr. Test. Cond. Monit.* **2007**, *49*, 137–141. [CrossRef]
6. Bahaghighat, M.; Motamedi, S.A. Vision inspection and monitoring of wind turbine farms in emerging smart grids. *Facta Univ. Ser. Electron. Energ.* **2018**, *31*, 287–301. [CrossRef]
7. Abedini, F.; Bahaghighat, M.; S'hoyan, M. Wind turbine tower detection using feature descriptors and deep learning. *Facta Univ. Ser. Electron. Energ.* **2019**, *33*, 133–153. [CrossRef]

8. Bahaghighat, M.; Motamedi, S.A.; Xin, Q. Image Transmission over Cognitive Radio Networks for Smart Grid Applications. *Appl. Sci.* **2019**, *9*, 5498. [CrossRef]
9. Esmaeili Kelishomi, A.; Garmabaki, A.; Bahaghighat, M.; Dong, J. Mobile user indoor-outdoor detection through physical daily activities. *Sensors* **2019**, *19*, 511. [CrossRef]
10. Bahaghighat, M.; Motamedi, S.A. PSNR enhancement in image streaming over cognitive radio sensor networks. *Etri J.* **2017**, *39*, 683–694. [CrossRef]
11. Bahaghighat, M.; Motamedi, S.A. IT-MAC: Enhanced MAC Layer for Image Transmission Over Cognitive Radio Sensor Networks. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 234.
12. Naghdehforushha, S.A.; Bahaghighat, M.; Salehifar, M.R.; Kazemi, H. Design of planar plate monopole antenna with vertical rectangular cross-sectional plates for ultra-wideband communications. *Facta Univ.-Ser. Electron. Energ.* **2018**, *31*, 641–650. [CrossRef]
13. Bahaghighat, M.; Naghdehforushha, A.; Salehifar, M.R.; Mirfattahi, M. Designing straight coaxial connectors for feeder and jumpers in cellular mobile base stations. *Acta Tech. Napoc. Electron.-Telecomun.* **2018**, *59*, 1–5.
14. Hasani, S.; Bahaghighat, M.; Mirfatahia, M. The mediating effect of the brand on the relationship between social network marketing and consumer behavior. *Acta Tech. Napoc.* **2019**, *60*, 1–6.
15. Igba, J.; Alemzadeh, K.; Durugbo, C.; Eiriksson, E.T. Analysing RMS and peak values of vibration signals for condition monitoring of wind turbine gearboxes. *Renew. Energy* **2016**, *91*, 90–106. [CrossRef]
16. Li, S.; Wunsch, D.C.; O'Hair, E.A.; Giesselmann, M.G. Using neural networks to estimate wind turbine power generation. *IEEE Trans. Energy Convers.* **2001**, *16*, 276–282.
17. Watson, S.; Xiang, J. Real-time condition monitoring of offshore wind turbines. In Proceedings of the European Wind Energy Conference & Exhibition, Athens, Greece, 2 March 2006; pp. 647–654.
18. Rumsey, M.A.; Paquette, J.A. Structural health monitoring of wind turbine blades. In *Smart Sensor Phenomena, Technology, Networks, and Systems 2008*; International Society for Optics and Photonics: San Diego, CA, USA, 2008; Volume 6933, p. 69330E.
19. Qiao, W.; Lu, D. A survey on wind turbine condition monitoring and fault diagnosis—Part II: Signals and signal processing methods. *IEEE Trans. Ind. Electron.* **2015**, *62*, 6546–6557. [CrossRef]
20. Verbruggen, T. *Wind Turbine Operation & Maintenance Based on Condition Monitoring WT-Ω*; Final Report, April; ECN, Energy Research Center of the Netherlands: Sint Maartensvlotbrug, The Netherlands, 2003.
21. Ribrant, J.; Bertling, L. Survey of failures in wind power systems with focus on Swedish wind power plants during 1997–2005. In Proceedings of the Power Engineering Society General Meeting, Tampa, FL, USA, 24–28 June 2007; pp. 1–8.
22. Grubic, S.; Aller, J.M.; Lu, B.; Habetler, T.G. A survey on testing and monitoring methods for stator insulation systems of low-voltage induction machines focusing on turn insulation problems. *IEEE Trans. Ind. Electron.* **2008**, *55*, 4127–4136. [CrossRef]
23. Grubic, S.; Aller, J.; Lu, B.; Habetler, T. A survey of testing and monitoring methods for stator insulation systems in induction machines. In Proceedings of the Conference on Condition Monitoring and Diagnosis, Beijing, China, 21 April 2008; pp. 196–203.
24. Lading, L.; McGugan, M.; Sendrup, P.; Rheinländer, J.; Rusborg, J. *Fundamentals for Remote Structural Health Monitoring of Wind Turbine Blades—A Preproject, Annex B: Sensors and Non-Destructive Testing Methods for Damage Detection in Wind Turbine Blades*; Risø National Laboratory: Roskilde, Denmark, 2002.
25. Kristensen, O.J.; McGugan, M.; Sendrup, P.; Rheinländer, J.; Rusborg, J.; Hansen, A.M.; Debel, C.P.; Sørensen, B.F. *Fundamentals for Remote Structural Health Monitoring of Wind Turbine Blades—A Preproject Annex E—Full-Scale Test of Wind Turbine Blade, Using Sensors and NDT*; Risø National Laboratory: Roskilde, Denmark, 2002.
26. McGugan, M.; Larsen, G.C.; Sørensen, B.F.; Borum, K.K.; Engelhardt, J. *Fundamentals for Remote Condition Monitoring of Offshore Wind Turbines*; Danmarks Tekniske Universitet: Riso National laboratorie, Denmark, 2008.
27. Gandhi, A.; Corrigan, T.; Parsa, L. Recent advances in modeling and online detection of stator interturn faults in electrical motors. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1564–1575. [CrossRef]
28. Wang, L.; Zhang, Z. Automatic detection of wind turbine blade surface cracks based on uav-taken images. *IEEE Trans. Ind. Electron.* **2017**, *64*, 7293–7303. [CrossRef]
29. Stokkeland, M. A Computer Vision Approach for Autonomous Wind Turbine Inspection Using a Multicopter. Master's Thesis, Institutt for Teknisk Kybernetikk, Trondheim, Norway, 2014.

30. Takeki, A.; Trinh, T.T.; Yoshihashi, R.; Kawakami, R.; Iida, M.; Naemura, T. Combining deep features for object detection at various scales: finding small birds in landscape images. *IPSJ Trans. Comput. Vis. Appl.* **2016**, *8*, 5. [CrossRef]
31. Wei, L.; Mirzaei, G.; Majid, M.W.; Jamali, M.M.; Ross, J.; Gorsevski, P.V.; Bingman, V.P. Birds/bats movement tracking with IR camera for wind farm applications. In Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 27 March 2014; pp. 341–344.
32. Zhang, Z.; Verma, A.; Kusiak, A. Fault analysis and condition monitoring of the wind turbine gearbox. *IEEE Trans. Energy Convers.* **2012**, *27*, 526–535. [CrossRef]
33. Castellani, F.; Garibaldi, L.; Daga, A.P.; Astolfi, D.; Natili, F. Diagnosis of Faulty Wind Turbine Bearings Using Tower Vibration Measurements. *Energies* **2020**, *13*, 1474. [CrossRef]
34. Seo, K.H.; Park, Y.; Yun, S.; Park, S.; Park, J.W. Vibration occurrence estimation and avoidance for vision inspection system. In *Robot Intelligence Technology and Applications 2*; Springer: New York, NY, USA, 2014; pp. 633–641.
35. Yang, H.; Gu, Q.; Aoyama, T.; Takaki, T.; Ishii, I. Dynamics-based stereo visual inspection using multidimensional modal analysis. *IEEE Sens. J.* **2013**, *13*, 4831–4843. [CrossRef]
36. Bahaghighat, M.; Abedini, F.; S'hoyan, M.; Molnar, A.J. Vision Inspection of Bottle Caps in Drink Factories Using Convolutional Neural Networks. In Proceedings of the 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 5–7 September 2019; pp. 381–385.
37. Bahaghighat, M.; Mirfattahi, M.; Akbari, L.; Babaie, M. Designing quality control system based on vision inspection in pharmaceutical product lines. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; pp. 1–4.
38. Akbari, R.; Bahaghighat, M.K.; Mohammadi, J. Legendre moments for face identification based on single image per person. In Proceedings of the 2010 2nd International Conference on Signal Processing Systems (ICSPS 2010), Dalian, China, 5–7 July 2010; Volume 1.
39. Mohammadi, J.; Akbari, R. Vehicle speed estimation based on the image motion blur using radon transform. In Proceedings of the 2010 2nd International Conference on Signal Processing Systems (ICSPS 2010), Dalian, China, 5–7 July 2010; Volume 1.
40. Bahaghighat, M.K.; Mohammadi, J. Novel approach for baseline detection and Text line segmentation. *Int. J. Comput. Appl.* **2012**, *51*, 9–16.
41. Karami, E.; Prasad, S.; Shehata, M. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv* **2017**, arXiv:1710.02726.
42. Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, NW, Washington, DC, USA, 24–27 June 2014; pp. 2147–2154.
43. Bahaghighat, M.K.; Sahba, F.; Tehrani, E. Textdependent Speaker Recognition by Combination of LBG VQ and DTW for Persian language. *Int. J. Comput. Appl.* **2012**, *51*, 23.
44. Ghorbani, M.; Bahaghighat, M.; Xin, Q.; Özen, F. ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. *J. Cloud Comput.* **2020**, *9*, 1–12. [CrossRef]
45. Bahaghighat, M.; Akbari, L.; Xin, Q. A machine learning-based approach for counting blister cards within drug packages. *IEEE Access* **2019**, *7*, 83785–83796. [CrossRef]
46. Sajadi, M.S.S.; Babaie, M.; Bahaghighat, M. Design and implementation of fuzzy supervisor controller on optimized DC machine driver. In Proceedings of the 2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN), Qazvin, Iran, 10 April 2018; pp. 26–31.
47. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
48. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
49. Jeong, J.; Park, H.; Kwak, N. Enhancement of SSD by concatenating feature maps for object detection. *arXiv* **2017**, arXiv:1705.09587.

50. Babaie, M.; Shiri, M.E.; Bahaghighat, M. A new descriptor for UAV images mapping by applying discrete local radon. In Proceedings of the 2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN), Qazvin, Iran, 10 April 2018; pp. 52–56.
51. Karimimehr, N.; Shirazi, A.A.B. Fingerprint image enhancement using gabor wavelet transform. In Proceedings of the 2010 18th Iranian Conference on Electrical Engineering (ICEE), Isfahan, Iran, 11–13 May 2010; pp. 316–320.
52. Bahaghighat, M.K.; Akbari, R. Fingerprint image enhancement using GWT and DMF. In Proceedings of the 2010 2nd International Conference on Signal Processing Systems (ICSPS), Dalian, China, 5–7 July 2010; Volume 1.
53. Huang, D.Y.; Hu, W.C.; Chang, S.H. Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Exp. Syst. Appl.* **2011**, *38*, 6031–6042. [CrossRef]
54. Affonso, C.; Rossi, A.L.D.; Vieira, F.H.A.; de Leon Ferreira, A.C.P. Deep learning for biological image classification. *Exp. Syst. Appl.* **2017**, *85*, 114–122. [CrossRef]
55. Gysel, P.; Motamedi, M.; Ghiasi, S. Hardware-oriented approximation of convolutional neural networks. *arXiv* **2016**, arXiv:1604.03168.
56. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; Volume 4, p. 12.
57. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the CVPR, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 3.
58. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1175–1183.
59. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features From Tiny Images*; Technical report; Citeseer: State College, PA, USA, 2009.
60. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
61. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. *arXiv* **2016**, arXiv:1602.07360.
62. Wu, W.; Wu, A.; Zheng, W.S. Light Person Re-Identification by Multi-Cue Tiny Net. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1643–1647.
63. Agoes, A.S.; Hu, Z.; Matsunaga, N. Fine tuning based squeezenet for vehicle classification. In Proceedings of the International Conference on Advances in Image Processing, Bangkok, Thailand, 25–27 August 2017; pp. 14–18.
64. Tzutalin. Git code: LabelImg. 2015. Available online: https://github.com/tzutalin/labelImg/blob/master/README.rst (accessed on 15 April 2020).
65. Soudani, A.; Barhoumi, W. An Image-based Segmentation Recommender using Crowdsourcing and Transfer Learning for Skin Lesion Extraction. *Exp. Syst. Appl.* **2018**, *105*, 233–261. [CrossRef]
66. Nweke, H.F.; Teh, Y.W.; Al-Garadi, M.A.; Alo, U.R. Deep Learning Algorithms for Human Activity Recognition using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges. *Exp. Syst. Appl.* **2018**, *105*, 233–261. [CrossRef]
67. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]