energies

MDPI

Article

Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis

ASM Shihavuddin [1,*], Xiao Chen [2,*], Vladimir Fedorov [3], Anders Nymark Christensen [1], Nicolai Andre Brogaard Riis [1], Kim Branner [2], Anders Bjorholm Dahl [1] and Rasmus Reinhold Paulsen [1]

1 Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), 2800 Lyngby, Denmark; anym@dtu.dk (A.N.C.); nabr@dtu.dk (N.A.B.R.); abda@dtu.dk (A.B.D.); rapa@dtu.dk (R.R.P.)
2 Department of Wind Energy, Technical University of Denmark (DTU), 4000 Roskilde, Denmark; kibr@dtu.dk
3 EasyInspect ApS, 2605 Brøndby, Denmark; vlf@easyinspect.io
* Correspondence: shihav@dtu.dk (A.S.); xiac@dtu.dk (X.C.)

Received: 24 January 2019; Accepted: 15 February 2019; Published: 20 February 2019

Abstract: Timely detection of surface damages on wind turbine blades is imperative for minimizing downtime and avoiding possible catastrophic structural failures. With recent advances in drone technology, a large number of high-resolution images of wind turbines are routinely acquired and subsequently analyzed by experts to identify imminent damages. Automated analysis of these inspection images with the help of machine learning algorithms can reduce the inspection cost. In this work, we develop a deep learning-based automated damage suggestion system for subsequent analysis of drone inspection images. Experimental results demonstrate that the proposed approach can achieve almost human-level precision in terms of suggested damage location and types on wind turbine blades. We further demonstrate that for relatively small training sets, advanced data augmentation during deep learning training can better generalize the trained model, providing a significant gain in precision.

Dataset: doi:10.17632/hd96prn3nc.1

Keywords: wind energy; rotor blade; wind turbine; drone inspection; damage detection; deep learning; Convolutional Neural Network (CNN)

1. Introduction

Reducing the Levelized Cost of Energy (LCoE) remains the overall driver for the development of the wind energy sector [1,2]. Typically, the Operation and Maintenance (O&M) costs account for 20–25% of the total LCoE for both onshore and offshore wind in comparison to 15% for coal, 10% for gas, and 5% for nuclear [3]. Over the years, great efforts have been made to reduce the O&M cost of wind energy using emerging technologies, such as automation [4], data analytics [5,6], smart sensors [7], and Artificial Intelligence (AI) [8]. The aim of these technologies is to achieve more efficient operation, inspection, and maintenance with minimal human interference. However, having a technology that can be deployed for on-site blade inspection for both onshore and offshore wind turbines under unpredictable weather conditions and that can acquire high-quality data efficiently is still a challenging task. One possible solution is to use the drone-based inspection of the wind turbine blades.

The drone-based approach enables low-cost and frequent inspections, high-resolution optical image acquisition, and minimal human intervention, thereby allowing predictive maintenance at

lower costs [9]. Wind turbine surface damages exhibit recognizable visual traits that can be imaged by drones with optical cameras. These damages include for example leading edge erosion, surface cracks, damaged lightning receptors, damaged vortex generators, and so forth. They are externally visible even in their early stages of development. Moreover, some of these damages, such as surface cracks, even indicate severe internal structural damages [10]. Nevertheless, internal damages such as delamination, debonding, or internal cracks are not detectable using the drone-based inspection with optical cameras. This study is limited to surface damages of the wind turbine blades.

Extracting damage information from a large number of high-resolution inspection images requires significant manual effort, which is one of the reasons for the overall inspection cost still remaining at a high level. In addition, manual image inspection is tedious and therefore error-prone. By automatically providing suggestions to experts on highly probable damage locations, we can significantly reduce the required man-hours and simultaneously enhance manual detection performance, as a result minimizing the labor cost involved with the analysis of inspection data. With regular cost-efficient and accurate drone inspection, the scheduled maintenance of wind turbines can be performed less frequently, potentially bringing down the overall maintenance cost, contributing to the reduction of LCoE.

Only very few research works have addressed the machine learning-based approaches for surface damage detection of wind turbine blades from drone images. One example, however, is Wang et al. [11], who used drone inspection images for crack detection. To automatically extract damage information, they used Haar-like features [12,13] and ensemble classifiers selected from a set of base models including logitBoost [14], decision trees [15], and support vector machines [16]. Their work was limited to detecting the crack and relied on classical machine learning methods.

Recently, deep learning technology has become efficient and popular, providing groundbreaking performances in detection systems for the last four years [17]. In this work, we addressed the problem of damage detection by deploying a deep learning object detection framework to aid human annotation. The main advantages of deep learning over other classical object detection methods are: it automatically finds the most discriminate features for the identification of objects, and it is achieved through an optimization process by minimizing the identification and localization errors.

Large size variations of different surface damages of wind turbine blades, in general, are a challenge for machine learning algorithms. In this study, we overcame this challenge with the help of advanced image augmentation methods. Image augmentation is the process of creating extra training images by altering images in the training sets. With the help of augmentation, different versions of the same image are created encapsulating different possible variations during drone acquisition [18].

The main contributions of this work are three-fold:

1. **Automated suggestion system for damage detection in drone inspection images**: implementation of a deep learning framework for automated suggestions of surface damages on wind turbine blades captured by drone inspection images. We show that deep learning technology is capable of giving reliable suggestions with almost human-level accuracy to aid manual annotation of damages.
2. **Higher precision in the suggestion model achieved through advanced data augmentation**: The advanced augmentation step called the "Multi-scale pyramid and patching scheme" enables the network to achieve better learning. This scheme significantly improves the precision of suggestions, especially for high-resolution images and for object classes that are very rare and difficult to learn.
3. **Publication of the wind turbine inspection dataset**: This work produced a publicly-available drone inspection image of the "Nordtank" turbine over the years of 2017 and 2018. The dataset is hosted within the Mendeley public dataset repository [19].

The surface damage suggestion system is trained using faster R-CNN [20], which is a state-of-the-art deep learning object detection framework. Faster R-CNN works efficiently and with

high accuracy compared to other frameworks, while identifying objects in terms of the bounding box from large images. The Convolutional Neural Network (CNN) is used as the backbone architecture in that framework for extracting feature descriptors with high discriminative power. The suggestion model is trained on drone inspection images of different wind turbines. We also employed advanced augmentation methods (as described in details in the Materials and Methods Section) to generalize the learned model. The more generalized model helps the system perform better on challenging test images during inference. Inference is the process of applying the trained model to an input image to receive the detected or, in our case, the suggested object in return.

Figure 1 illustrates the flowchart of the proposed method. To begin with, damages on wind turbine blades that are imaged using drone inspections are annotated in terms of bounding boxes by field experts. Annotated images are also augmented with the proposed advanced augmentation schemes (such as the pyramid, patching, and regular augmentations, as described in details in the Materials and Methods Section) to increase the number of training samples. The faster R-CNN deep learning object detection framework is applied to train from these annotated and augmented annotated images. Within the faster R-CNN framework, the backbone CNN in this case is the deep one, called the Inception-ResNet-V2 architecture. CNN converts images into high-level spatial features called the feature map. The region proposal network tries to estimate where the objects could be located, and ROI pooling is used to extract relevant features from the feature map for that particular region and based on that classifier, making the decision of whether an object of that particular class is present or not. After the training, the deep learning framework produces a detection model that can be applied for new inspection image analysis.

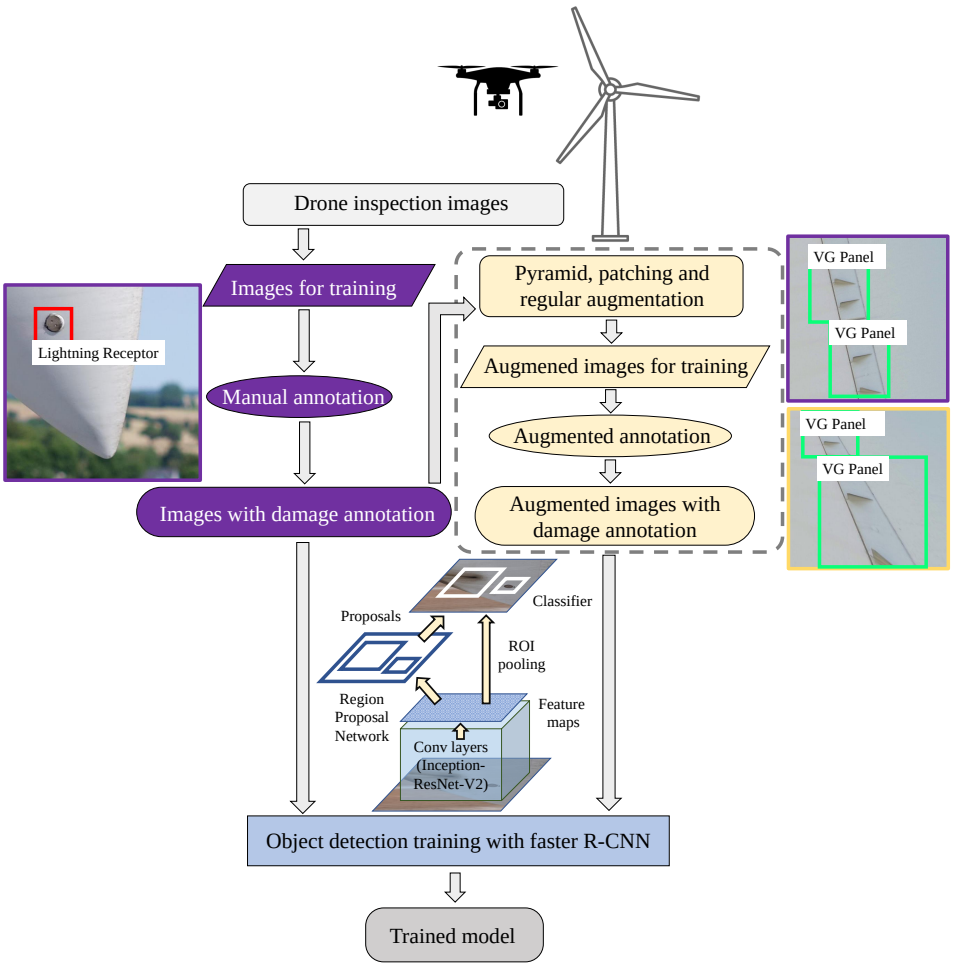

Figure 1. Flowchart of the proposed automated damage suggestion system. VG, Vortex Generator.

2. Materials and Methods

2.1. Manual Annotation

For this work, four classes were defined: Leading Edge erosion (LE erosion), Vortex Generator panel (VG panel), VG panel with missing teeth, and lightning receptor. Examples of each of these classes are illustrated in Figure 2. Figure 2a,g,h illustrates the examples of leading edge erosion annotated by experts using bounding boxes. Figure 2b,d,e illustrates the lightning receptors. Figure 2c shows the example of a VG panel with missing teeth. Figure 2c,f show the examples of well-functioning VG panels. These classes served as the passive indicators of the health condition of the wind turbine blades. The reason behind choosing these classes for experimentation was that all these types of damages produce specific visual traits recognizable by humans and would be valuable if a machine could learn to do the same.

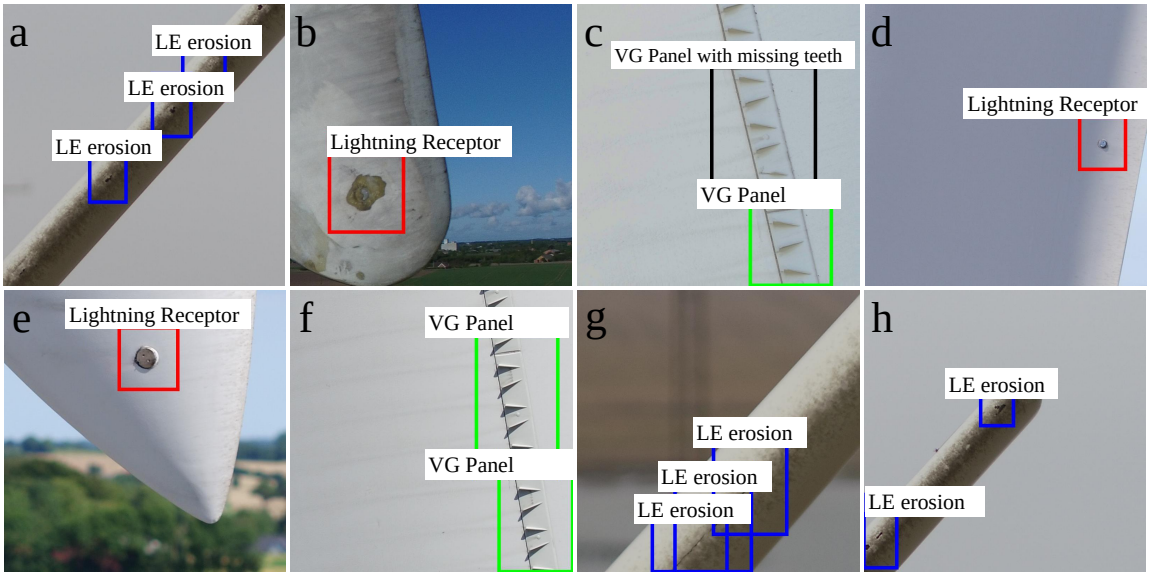

Figure 2. Examples of manually-annotated damages related to wind turbine blades. LE, Leading Edge.

A VG panel and a lightning receptor are not specific damage types, but rather external components on the wind turbine blade that often have to be visually checked during inspections. For inspection purpose, it is of value to detect the location of the lighting receptors and then identify if they are damaged/burned or not. In such cases, the machine learning task could be designed to identify damaged and non-damaged lightning receptors automatically. In this work, we simplified the task into first detecting the lightning receptor and, afterward, if needed, classifying them into damaged or non-damaged ones.

Table 1 summarizes the number of annotations for each class, which are annotated by experts and considered as ground truths. These annotated images were taken from the dataset titled "EasyInspect dataset" owned by EasyInspect ApS company, comprising drone inspection of different types of wind turbine blades located in Denmark. From the pool of available images, 60% were used for training and 40% for testing. As there was a limited number of examples of some damage types such as damaged lightning receptors, 40% of the samples were kept for reliable testing. To evaluate the performance of the developed system it is important to have a substantial number of unseen samples to examine. Annotations in the training set comprised of the annotations from full resolution images that were randomly selected. Annotations in the testing set comprised of the annotations from full-resolution images and also from cropped images containing objects of interest. This was done to make the testing part challenging by varying the scale of an object compared to the size of the image.

Table 1. List of classes in the training and testing sets related to the EasyInspect dataset.

List of Classes		
Classes	**Annotations-Training**	**Annotations-Testing**
Leading Edge (LE) erosion	135	67
Vortex Generator panel (VG)	126	65
Vortex Generator with Missing Teeth (VGMT)	27	14
Lightning receptor	17	7

2.2. Image Augmentations

Some types of damages on wind turbine blades are rare, and it is hard to collect representative samples during inspections. For deep learning, it is necessary to have thousands of training samples that are representative of the depicted patterns in order to obtain a good detection model [21]. The general approach is to use example images from the training set and then augment them in ways that represent probable variations in appearances, maintaining the core properties of object classes.

2.2.1. Regular Augmentations

Regular augmentations are defined as conventional ways of augmenting images for increasing the number of training samples. There are many different types of commonly-used augmentation methods that could be selected based on the knowledge about object classes and the possible occurrences of variances during acquisition. Taking drone inspection and wind turbine properties into consideration, four types of regular augmentation were chosen to be used in this work, which are listed below. These four types of regular augmentations were selected to represent real-life possibilities that could occur during drone inspection of wind turbine blades and have been proven to provide positive impacts on deep learning-based image classification tasks [22–24].

- Perspective transformation for the camera angle variation simulation.
- Left-to-right flip or top-to-bottom flip to simulate blade orientation: e.g., pointing up or down.
- Contrast normalization for variations in lighting conditions.
- Gaussian blur simulating out of focus images.

Figure 3 illustrates the examples of regular augmentations of the wind turbine inspection images containing damage examples. Figure 3e is the perspective transform of Figure 3a, where both images illustrate the same VG panels; Figure 3f is left-to-right flipping of the image patch of Figure 3b containing a lightning receptor. Figure 3g is the contrast normalized version of Figure 3c. Figure 3h is the augmented image of the lightning receptor in Figure 3d simulating de-focusing of the camera using approximate Gaussian blur.

2.2.2. Pyramid and Patching Augmentation

Drones deployed to acquire the dataset typically were equipped with very high-resolution cameras. High-resolution cameras allowed the drones to capture essential details, being at a further and safer distance from the turbines. These high-resolution images allowed the flexibility of training from rare types of damages and a wide variety of backgrounds at a different resolution using the same image.

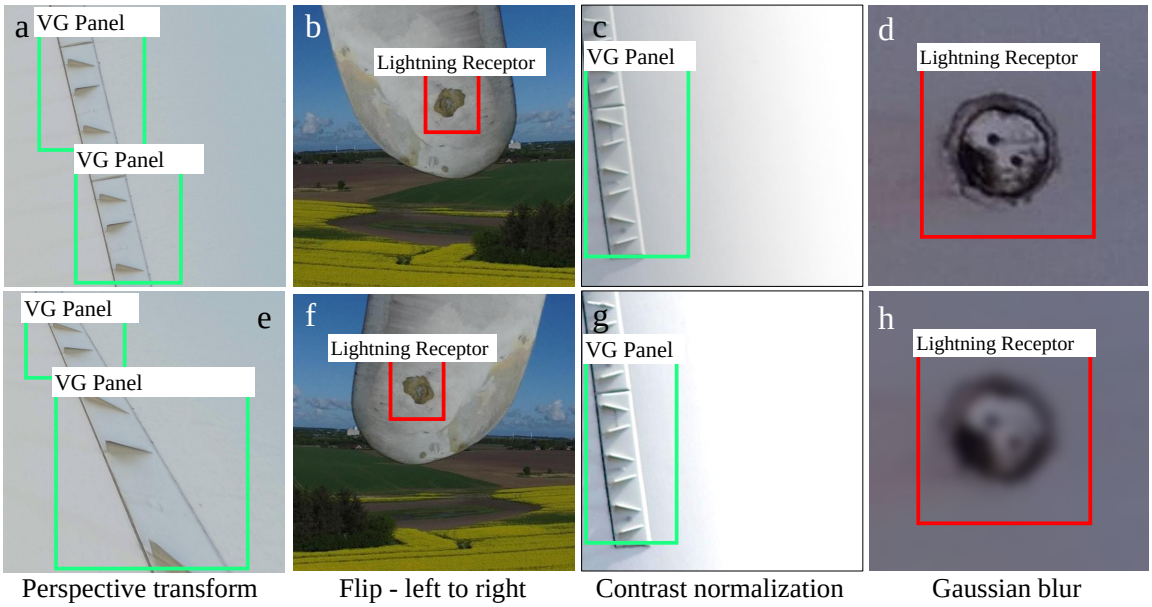

Figure 3. Illustrations of the regular augmentation methods applied to drone inspection images.

The deep learning object detection frameworks such as the faster R-CNN framework have predefined input image dimensions that typically allow for a maximum input image dimension (either height or width) of 1000 pixels [20,25]. This is maintained through re-sizing higher resolution images while keeping the ratio between width and height in situ. For high-resolution images, this limitation creates new challenges due to damages being minimal in pixel sizes compared to full image size.

In Figure 4, on the right is the pyramid scheme, and on the left is the patching scheme. The bottom level of the pyramid scheme is defined as the image size where either the height or width is 1000 pixels. In the pyramid, from top to bottom, images are scaled from 1.00× to 0.33×, simulating from the highest to the lowest resolutions. Sliding windows with 10% overlap were scanned over the images at each resolution to extract patches containing at least one object. Resolution conversions were performed through the linear interpolation method [26]. For example, in Figure 4, top right, the acquired full resolution image is 4000 × 3000 pixels, where the lightning receptor only occupies around 100 × 100 pixels. When fed to CNN during training in full resolution, it would be resized to the pre-defined network input size, where the lightning receptor would be occupying a tiny portion of 33 × 33 pixels. Hence, it is rather complicated to acquire enough recognizable visual traits of the lightning receptor.

Using the multi-scale pyramid and patching scheme on the acquired high-resolution training images, scale-varied views of the same object were generated and fed to the neural network. In this scheme, the main full-resolution image was scaled to multiple resolution images (1.00×, 0.67×, 0.33×), and on each of these images, patches containing objects were selected with the help of a sliding window with 10% overlap. The selected patches were always 1000 × 1000 pixels.

The flowchart of this multi-scale pyramid and patching scheme is shown in Figure 4. This scheme helps to represent object capture at different camera distances, allowing the detection model to be efficiently trained on both low- and high-resolution images.

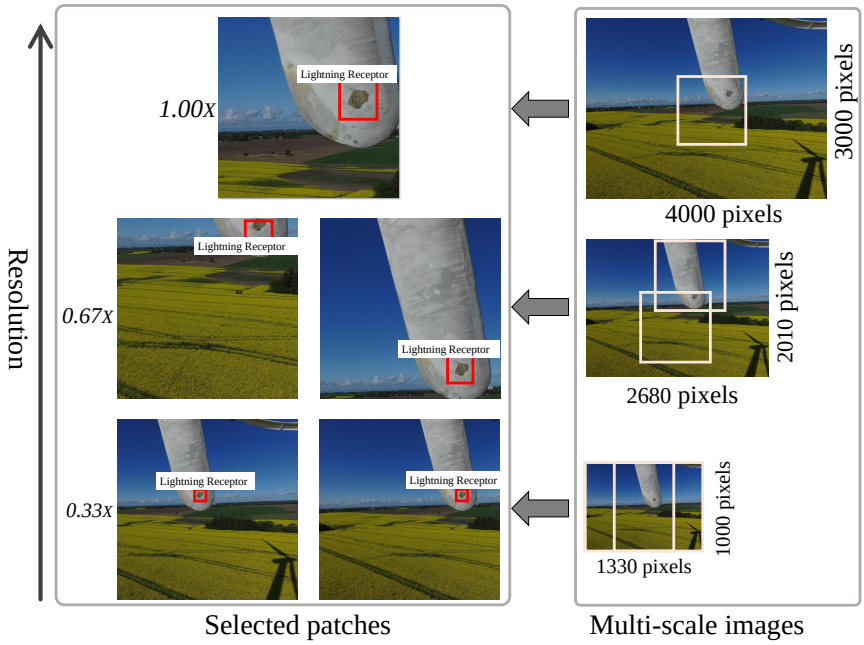

Figure 4. Proposed multi-scale pyramid and patching scheme for image augmentation.

2.3. Damage Detection Framework

With recent advances in deep learning for object detection, new architectures are frequently proposed, establishing groundbreaking performances. Currently, different stable meta architectures are publicly-available and have already been successfully deployed for many challenging applications. Among deep learning object detection frameworks, one of the best-performing methods is the faster R-CNN [20,27]. We also experimented with other object detection frameworks such as R-CNN [25], fast R-CNN [28], SSD[29], and R-FCN [30]. It was found that indeed, the faster R-CNN outperformed others in terms of accuracy when real-time processing was not needed, and deep CNN architectures like ResNet [31] were used for feature extraction. In our work, the surface damage detection and classification using drone inspection images were performed using faster R-CNN [20].

Faster R-CNN [20] uses a Region Proposal Network (RPN) [32] trained on feature descriptors extracted by CNN to predict bounding boxes for objects of interest. The CNN architecture automatically learns features such as texture, spatial arrangement, class size, shape, and so forth, from training examples. These automatically-learned features are more appropriate than hand-crafted features.

Convolutional layers in CNN summarize information based on the previous layer's content. The first layer usually learns edges; the second finds patterns in edges encoding shapes of higher complexity, and so forth. The last layer contains a feature map of much smaller spatial dimensions than the original image. The last layer feature map summarizes information about the original image. We experimented with both lighter CNN architectures such as InceptionV2 [33] and ResNet50 [31] and heavier CNN architectures such as ResNet101 [31] and Inception-ResNet-V2 [34].

The Inception [33] architecture by Google contains an inception module as one of the building blocks, and the computational cost is about 2.5-times higher than that of GoogLeNet [35]. The ResNet architecture [31] is known for its residual blocks, which help to reduce the impact of vanishing gradients from the top layers to the bottom layers during training. Inception-ResNet-V2 [34] is one of the extensions of the inception architectures that incorporates both the inception blocks and residual ones. This particular CNN network was used as the backbone architecture in our final model within a faster R-CNN framework for extracting highly discriminating feature descriptors. This network is computationally heavy and was found to provide state-of-the-art performances for object detection tasks [27].

2.4. Performance Measure: Mean Average Precision

All the reported performances in terms of Mean Average Precision (MAP) were measured during inference on the test images, where the inference is the process of applying the trained model to an input image to receive the detected and classified object in return. In this work, we called it suggestions if the trained model from deep learning was being used.

MAP is commonly used in computer vision to evaluate object detection performance during inference. An object proposal is considered accurate only if it overlaps with the ground truth with more than a certain threshold. Intersection over Union (IoU) is used to measure the overlap of a prediction and the ground truth where ground truth refers to the original damages identified and annotated by experts in the field.

$$\text{IoU} = \frac{P \cap GT}{P \cup GT} \tag{1}$$

The IoU value corresponds to the ratio of the common area over the sum of the proposed detection and ground truth areas (as shown in Equation (1), where P and GT are the predicted and ground truth bounding boxes, respectively). If the value is more than 0.5, the prediction or, in this case, the suggestion is considered as a true positive. This 0.5 value is relatively conservative, as it makes sure that the ground truth and the detected object have a very similar bounding box location, size, and shape. For addressing human perception diversities, we used a 0.3 IoU threshold for considering a detection as a true positive.

$$\text{P}_C = \frac{N(\text{True Positives})_C}{N(\text{Total Objects})_C} \tag{2}$$

$$\text{AP}_C = \frac{\sum \text{P}_C}{N(\text{Total Images})_C} \tag{3}$$

$$\text{MAP} = \frac{\sum \text{AP}_C}{N(\text{Classes})} \tag{4}$$

Per class precision for each image, P_C, was calculated using Equation (2). For each class, average precision, AP_C was measured over all the images in the dataset using Equation (3). Finally, MAP was measured as the mean of average precision for each class over all the classes in the dataset (see Equation (4)). Throughout this work, the MAP is reported in percentage.

2.5. Software and Codes

We used the Tensorflow [36] deep learning API for experimenting with the faster R-CNN object detection method with different CNN architectures. These architectures were compared with each other under the proposed augmentation schemes. For implementing the regular augmentations, the imgaug (https://github.com/aleju/imgaug) package was used, and for the pyramid and patching augmentation, an in-house python library was developed. Inception-ResNet-V2 and other CNN weights were initialized from the pre-trained weight on the Common Objects in COntext (COCO) dataset [37]. The COCO dataset consists of 80 categories of regular objects and is commonly used for bench-marking deep learning object detection performance.

2.6. Hardware

All the experiments reported in this work were performed on a GPU cluster machine with 11-GB GeForce GTX 1080 Graphics Cards within a Linux operating system. The initial time required for training 275 epochs (where one epoch is defined as when all the images in the training set had been used at least once for optimization of the detection model) using Inception-V2, ResNet-50, ResNet-101, and Inception-ResNet-V2 networks was on average 6.3 h, 10.5 h, 16.1 h, and 27.9 h, respectively.

2.7. Dataset

2.7.1. EasyInspect Dataset

The EasyInspect dataset is a non-public inspection dataset provided by EasyInspect ApS, which contains images (4000 × 3000 pixels in size) of different types of damages on wind turbines from different manufacturers. The four classes are LE erosion, VG panel, VG panel with missing teeth, and lightning receptor.

2.7.2. DTU Drone Inspection Dataset

In this work, we produced a new public dataset entitled DTU—Drone inspection images of the wind turbine. It is the only public wind turbine drone inspection image dataset containing a total of 701 high-resolution images. This dataset contains temporal inspection images of 2017 and 2018 covering the "Nordtank" wind turbine located at DTU Wind Energy's test site at Roskilde, Denmark. The dataset comes with the examples of damages or mounted objects such as VG panel, VG panel with missing teeth, LE erosion, cracks, lightning receptor, damaged lightning receptor, missing surface material, and others. It is hosted at [19].

3. Results

3.1. Augmentation of Training Images Provides a Significant Gain in Performance

Comparing different augmentation types showed that a combination of the regular, pyramid, and patching augmentations produced a more accurate suggestion model, especially for the deeper CNN architectures as the backbone for the faster R-CNN framework. Using CNN architecture ResNet-101 for example (as shown in Table 2 in column "all"), without any augmentation, the MAP (detailed in the Materials and Methods Section) of damage suggestion was very low with a value of 25.9%. With the help of the patching augmentation, the precision improved significantly (as for this case, the MAP increased to 35.6%). In Table 2, all the experimental results are reported in terms of MAP. VG and VGMT represent the VG panel and the VG with Missing Teeth, respectively. "All" is the overall MAP comprising all four classes.

Table 2. Experimental results for different CNN architectures and data augmentation methods.

CNN	Augmentation	MAP (%)				
		LE Erosion	VG	VGMT	Lightning Receptor	All
Inception-V2	Without	20.34	13.15	4.21	41.39	19.77
	Patching	37.98	53.86	32.74	22.12	36.67
	Patching + regular	36.90	52.58	36.08	23.54	37.28
	Pyramid + patching	88.11	86.61	58.28	70.18	75.80
	Pyramid + patching + regular	89.94	84.15	71.85	40.76	71.67
ResNet-50	Without	37.51	17.54	7.02	40.88	25.74
	Patching	34.48	54.92	32.17	34.96	39.13
	Patching + regular	35.06	47.87	24.88	34.96	35.69
	Pyramid + patching	90.19	85.15	61.43	73.85	77.66
	Pyramid + patching + regular	90.47	88.84	35.27	73.15	71.93
ResNet-101	Without	29.61	20.29	4.21	49.33	25.86
	Patching	34.79	47.35	25.35	34.96	35.61
	Patching + regular	36.06	51.41	32.16	33.46	38.27
	Pyramid + patching	88.86	87.46	41.53	64.19	70.51
	Pyramid + patching + regular	86.77	86.15	41.53	77.00	72.86
Inception-ResNet-V2	Without	31.54	18.35	5.96	54.14	27.50
	Patching	36.30	44.43	33.37	42.12	39.06
	Patching + regular	32.73	47.59	19.58	34.96	33.72
	Pyramid + patching	90.22	91.66	69.85	69.56	80.32
	Pyramid + patching + regular	90.62	89.08	73.11	71.58	81.10

Together with the patching and the regular augmentations, the MAP increased slightly to 38.3%. However, the pyramid scheme dramatically improved the performance of the trained model up to 70.5%. The best performing configuration was the last one with the combination of the pyramid, patching, and regular augmentation schemes, generating an MAP of 72.9%.

As shown in Figure 5a–d, the proposed combination of all the proposed augmentation methods significantly and consistently improved the performance of the model and lifts it to above 70% for all four CNN architecture. Figure 5a–d represents sequentially lighter to deeper CNN backbone architectures used for deep learning feature extraction. The CNN networks explored in this work were: Inception-V2, ResNet-50, ResNet-101, and Inception-ResNet-V2. In each individual figure (a–d): the y-axis represents the MAP of the suggestion on the test set, which are reported in percentage.

For any specific type of augmentation, MAPs, in general, were higher for the deeper networks (which were ResNet-101 and Inception-ResNet-V2) than for the lighter ones. For these two deeper networks, note that regular augmentation on top of the multi-scale pyramid and patching scheme added on average 2% gain in MAP. For lighter networks (Inception-V2 and ResNet-50), due to the limited search space, the network tended to learn and map better without the addition of regular augmentation. The results also demonstrated that for the small dataset (where some types of damages were extremely rare), it was beneficial to generate augmented images following class variation probabilities in terms of scale, light conditions, focuses, and acquisition angles.

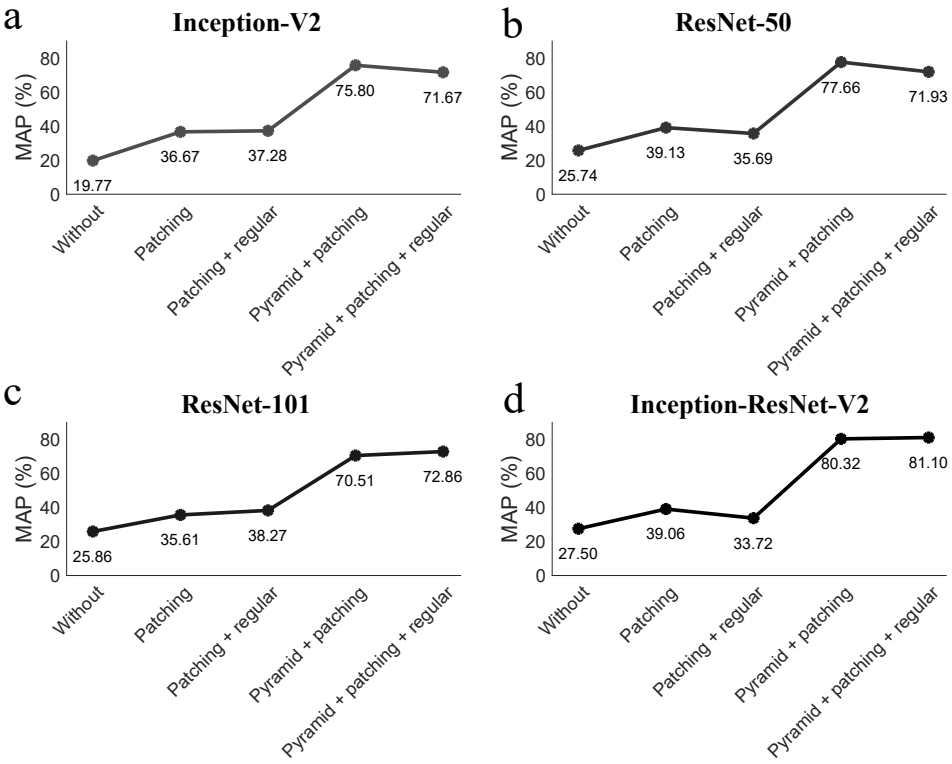

Figure 5. Comparison of the precision of trained models using various network architectures and augmentation types.

3.2. The CNN Architecture Performs Better as It Goes Deeper

When comparing the four selected CNN backbone architectures for the faster R-CNN framework, the Inception-ResNet-V2, which was the deepest, performed the best among all. If we fixed the augmentation to the combination of pyramid, patching, and regular, the MAP of the Inception-V2, ResNet-50, ResNet-101, and Inception-ResNet-V2 would be 71.67%, 71.93%, 72.86%, and 81.10%, respectively (as shown in Figure 5 and in Table 2). The number of layers in each of these networks representing the depth of the network could be arranged in the same order, as well (where Inception-V2

was the lightest and Inception-ResNet-V2 the deepest). This demonstrates that the performance regarding MAP increased as the network went deeper. The gain in performance of deeper networks comes with the cost of a longer training time and higher requirements on the hardware.

3.3. Faster Damage Detection with the Suggestive System in Inspection Analysis

The time required for automatically suggesting damage locations for new images using the trained model depends on the size of the input image and the depth of the CNN architecture used. In the presented case, the average time required for inferring a high-resolution image using Inception-V2, ResNet-50, ResNet-101, and Inception-ResNet-V2 networks (after leading the model) was respectively 1.36 s, 1.69 s, 1.87 s, and 2.11 s; whereas, for human-based analysis without suggestions, it can take around 20 s–3 min per image depending on the difficulty level for identification. With the deep learning-aided suggestion system for humans, the analysis time went significantly down (almost to two-thirds) compared to human speed without suggestions and also produced better accuracy (see Table 3).

Damages on unseen inspection images were annotated by experts having deep learning-aided suggestions as bounding boxes and without. In the case of suggestive bounding boxes, experts only needed to correct, whereas in the case of "without suggestion", they needed to draw the bounding box from scratch. While annotating 100 randomly-selected images, with suggestion, it took on average 131 s per image, whereas without suggestion, it was around 200 s per image. Human results (in terms of precision) with and without suggestions are called "Human" and "Suggestions aiding human". The precision of the deep learning trained model's suggestion is called "Suggestions". To access the precision of "Suggestions", the best-performing model Inception-ResNet-V2 within the faster R-CNN framework equipped with pyramid, patching, and regular augmentation was used.

Table 3. Summary of the experimental results.

	Mean Average Precision (MAP) in Percentage		
Classes	**Suggestions**	**Human**	**Suggestions Aiding Human**
LE erosion	90.62	68.00	86.60
VG panel (VG)	89.08	85.91	81.57
VG with Missing Teeth (VGMT)	73.11	80.17	91.37
Lightning receptor	71.58	98.74	81.71
Overall precision	81.10	83.20	85.31
Average processing time per image	2.11 s	200 s	131 s

4. Discussion

With the deep learning-based automated damage suggestion system, the best performing model in the proposed method produced 81.10% precision, which is within the 2.1% range of the average human precision of 83.20%. In the case of deep learning-aided suggestion for humans, the MAP improved significantly to 85.31%, and the required processing time became two-thirds that on average for each image. This result suggests that humans can benefit from suggestions by knowing where to look for damages in images, especially for difficult cases like VG panels with missing teeth.

The experimental results show that for a smaller image dataset of wind turbine inspection, the performance was more sensitive to the quality of image augmentation than the selection of CNN architecture. One of the main reasons is that most damage types can have a considerably large variance in appearance, which makes the deployed network dependent on a larger number of examples from which to learn.

The combination of ResNet and inception modules in Inception-ResNet-V2 learned difficult damage types such as missing teeth in a vortex generator with more reliability than by the other CNN architectures. Figure 6 illustrates some of the suggestion results on inspection images for

testing. Figure 6a,d,f illustrates suggested lighting receptors; Figure 6b,h shows LE erosion suggestion; Figure 6c,e illustrates the suggestion of VG panels with intact teeth and those with missing teeth, respectively. The latter exemplifies one of the very challenging tasks for automated damage suggestion method. Figure 6g shows the example of when no damage is detected. The suggestion model developed in this study performed well for challenging images, providing almost human-level precision. When there was no valid class present in the image, the trained model found only the background class and presented "No detection" as the label for that image (an example is shown in Figure 3g).

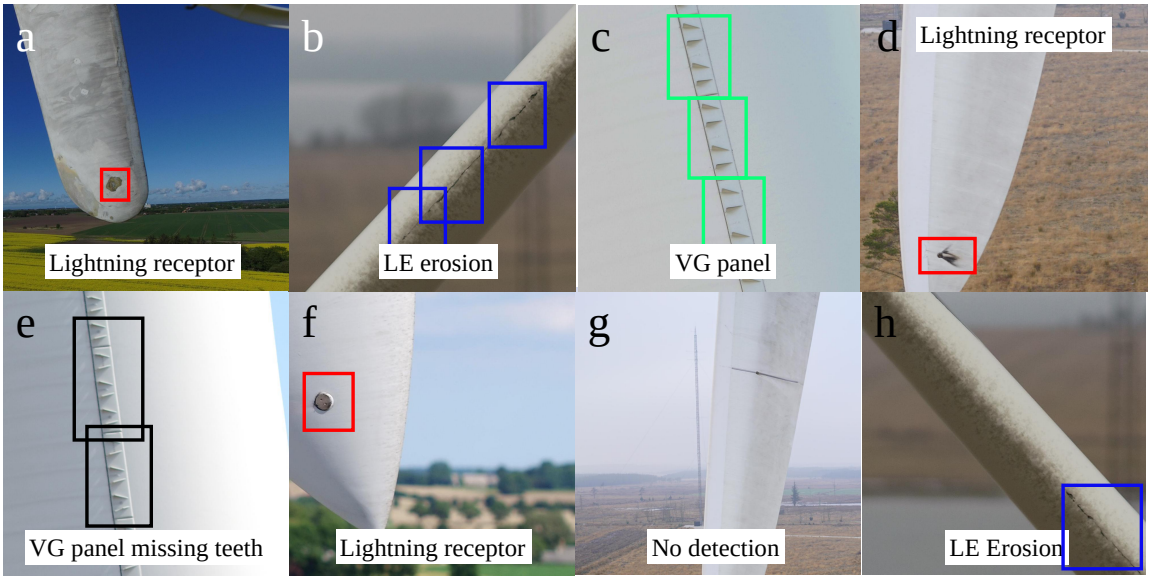

Figure 6. Suggestion results on the test images for the trained model using Inception-ResNet-V2 together with the proposed augmentation schemes.

This automated damage suggestion system has a potential cost advantage over the current manual one. Currently, drone inspections typically can cover up to 10 or 12 wind turbines per day. Damage detection, however, is much less efficient, as it involves considerable data interpretation for damage identification, annotation, classification, etc., which has to be conducted by skilled personnel. This process would incur significant labor cost considering the huge amount of images taken by the drones from the field. Using the deep learning framework for the suggestion to aid manual damage detection, the entire inspection and analysis process can be partially automated to minimize human intervention.

With suggested damages and subsequent corrections by human experts, over time, the number of annotated training examples would be increased and fed to the developed system for updating the trained suggestion model. This continuous way of learning through the help of human experts can increase the accuracy of the deep learning model (expected to provide 2–5% gain in MAP) and also reduce the required time for human corrections.

Relevant information about the damages, i.e., their size and location on the blade, can be used for further analysis in estimating wind turbine structural and aerodynamic conditions. The highest standalone MAP achieved with the proposed method was 81.10%, which is almost within human-level precision given the complexity of the problem and the conservative nature of the performance indicator. The developed automated detection system at its current state can safely work as a suggestion system for experts to finalize the damage locations from inspection data.

The required computational power and deep learning training can incur higher initial cost mainly due to the fact that acquiring training images comprising damage examples is expensive. However in the long run, a well-trained model on damage types of interest can produce a less expensive and more reliable solution to the drone inspection analysis task. With this proposed method, the majority

of surface damages on the wind turbine blade can be semi-automatically recognized and reported for future actions in terms of maintenance.

5. Conclusions

This work presented a deep learning-based automated method to aid manual detection of wind turbine blade surface damages. The method can reduce human intervention by providing an accurate suggestion of damages on drone inspection images. Using the Inception-ResNet-v2 architecture inside faster R-CNN, 81.10% MAP was achieved on four different types of damages, i.e., LE erosion, vortex generator panel, vortex generator panel with missing teeth, and lightning receptor. The authors adopted a multi-scale pyramid and patching scheme that significantly improved the precision by 35% on average across the tested CNN architectures. The experimental results demonstrated that deep learning with augmentation can overcome the challenge of the scarce availability of damage samples for learning. In this work, a new image dataset of wind turbine drone inspection was published for the research community [19].

Author Contributions: R.R.P., K.B., and A.B.D. designed and supervised the research. A.S. and V.F. designed the experiments, implemented the system, and performed the research. X.C., A.N.C., and N.A.B.R. analyzed the data. A.S. wrote the manuscript with inputs from all authors.

Funding: This research was funded by the Innovation fund of Denmark through the DARWIN Project (Drone Application for pioneering Reporting in Wind turbine blade Inspections)

Acknowledgments: We would like to thank the Danish Innovation fund for the financial support through the DARWIN Project (Drone Application for pioneering Reporting in Wind turbine blade Inspections) and EasyInspect ApS for their kind contribution to this work by providing the drone inspection dataset.

Conflicts of Interest: The authors declare that they have no competing financial interests.

References

1. Aabo, C.; Scharling Holm, J.; Jensen, P.; Andersson, M.; Lulu Neilsen, E. *MEGAVIND Wind Power Annual Research and Innovation Agenda*; MEGAVIND: Copenhagen, Denmark, 2018.
2. Ziegler, L.; Gonzalez, E.; Rubert, T.; Smolka, U.; Melero, J.J. Lifetime extension of onshore wind turbines: A review covering Germany, Spain, Denmark, and the UK. *Renew. Sustain. Energy Rev.* **2018**, *82*, 1261–1271. [CrossRef]
3. Verbruggen, S. *Onshore Wind Power Operations and Maintenance to 2018*; New Energy Update: London, UK, 2016.
4. Stock-Williams, C.; Swamy, S.K. Automated daily maintenance planning for offshore wind farms. *Renew. Energy* **2018**, doi:10.1016/j.renene.2018.08.112. [CrossRef]
5. Canizo, M.; Onieva, E.; Conde, A.; Charramendieta, S.; Trujillo, S. Real-time predictive maintenance for wind turbines using Big Data frameworks. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 70–77.
6. Zhou, K.; Fu, C.; Yang, S. Big data driven smart energy management: From big data to big insights. *Renew. Sustain. Energy Rev.* **2016**, *56*, 215–225, doi:10.1016/j.rser.2015.11.050. [CrossRef]
7. Wymore, M.L.; Dam, J.E.V.; Ceylan, H.; Qiao, D. A survey of health monitoring systems for wind turbines. *Renew. Sustain. Energy Rev.* **2015**, *52*, 976–990, doi:10.1016/j.rser.2015.07.110. [CrossRef]
8. Jha, S.K.; Bilalovic, J.; Jha, A.; Patel, N.; Zhang, H. Renewable energy: Present research and future scope of Artificial Intelligence. *Renew. Sustain. Energy Rev.* **2017**, *77*, 297–317, doi:10.1016/j.rser.2017.04.018. [CrossRef]
9. Morgenthal, G.; Hallermann, N. Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures. *Adv. Struct. Eng.* **2014**, *17*, 289–302. [CrossRef]
10. Chen, X. Fracture of wind turbine blades in operation—Part I: A comprehensive forensic investigation. *Wind Energy* **2018**, 6, doi:10.1002/we.2212. [CrossRef]
11. 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]

12. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; Volume 1.
13. Shihavuddin, A.; Arefin, M.M.N.; Ambia, M.N.; Haque, S.A.; Ahammad, T. Development of real time Face detection system using Haar like features and Adaboost algorithm. *Int. J. Comput. Sci. Netw. (IJCSNS)* **2010**, *10*, 171–178.
14. Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stat.* **2000**, *28*, 337–407. [CrossRef]
15. Liaw, A.; Wiener, M. Classification and regression by random Forest. *R News* **2002**, *2*, 18–22.
16. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
17. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
18. Hauberg, S.; Freifeld, O.; Larsen, A.B.L.; Fisher, J.; Hansen, L. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*; Cornell University (Computer Vision and Pattern Recognition): New York, NY, USA, 2016; pp. 342–350.
19. Shihavuddin, A.; Chen, X. *DTU—Drone Inspection Images of Wind Turbine*; This Dataset Set Has Temporal Inspection Images for the Years of 2017 and 2018 of the 'Nordtank' Wind Turbine at DTU Wind Energy's Test Site in in Roskilde, Denmark; Mendeley: Copenhagen, Denmark, 2018.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Cornell University (Computer Vision and Pattern Recognition): New York, NY, USA, 2015; pp. 91–99.
21. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
22. Hussain, Z.; Gimenez, F.; Yi, D.; Rubin, D. Differential Data Augmentation Techniques for Medical Imaging Classification Tasks. In *AMIA Annual Symposium Proceedings*; American Medical Informatics Association: Bethesda, MD, USA, 2017; Volume 2017, p. 979.
23. Oliveira, A.; Pereira, S.; Silva, C.A. Augmenting data when training a CNN for retinal vessel segmentation: How to warp? In Proceedings of the 2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG), Coimbra, Portugal, 16–18 February 2017; pp. 1–4.
24. Hu, T.; Qi, H. See Better Before Looking Closer: Weakly Supervised Data Augmentation Network for Fine-Grained Visual Classification. *arXiv* **2019**, arXiv:1901.09891.
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
26. Blu, T.; Thévenaz, P.; Unser, M. Linear interpolation revitalized. *IEEE Trans. Image Process.* **2004**, *13*, 710–719. [CrossRef] [PubMed]
27. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
28. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
30. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*; NIPS: Barcelona, Spain, 2016; pp. 379–387.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
32. Kong, T.; Yao, A.; Chen, Y.; Sun, F. Hypernet: Towards accurate region proposal generation and joint object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 845–853.
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*; ICML: Lille, France, 6–11 July 2015; pp. 448–456.

34. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. *Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning*; AAAI: Palo Alto, CA, USA, 2017; Volume 4, p. 12.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 1–9.
36. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software. Available online: tensorflow.org (accessed on 12 November 2017).
37. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*; Springer: Zurich, Switzerland, 2014; pp. 740–755.