applied sciences

MDPI

Article

Automatic Detection of Corrosion in Large-Scale Industrial Buildings Based on Artificial Intelligence and Unmanned Aerial Vehicles

Rafael Lemos [1], Rafael Cabral [2], Diogo Ribeiro [3,*], Ricardo Santos [3], Vinicius Alves [1] and André Dias [4]

[1] Department of Civil Engineering, School of Mines, Federal University of Ouro Preto, Ouro Preto 35400-000, Brazil
[2] CONSTRUCT-LESE, Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal
[3] CONSTRUCT-LESE, School of Engineering, Polytechnic of Porto, 4249-015 Porto, Portugal
[4] INESC TEC, School of Engineering, Polytechnic of Porto, 4249-015 Porto, Portugal
* Correspondence: drr@isep.ipp.pt

Abstract: In recent years, Artificial Intelligence (AI) provided essential tools to enhance the productivity of activities related to civil engineering, particularly in design, construction, and maintenance. In this framework, the present work proposes a novel AI computer vision methodology for automatically identifying the corrosion phenomenon on roofing systems of large-scale industrial buildings. The proposed method can be incorporated into computational packages for easier integration by the industry to enhance the inspection activities' performance. For this purpose, a dedicated image database with more than 8k high-resolution aerial images was developed for supervised training. An Unmanned Aerial Vehicle (UAV) was used to acquire remote georeferenced images safely and efficiently. The corrosion anomalies were manually annotated using a segmentation strategy summing up 18,381 instances. These anomalies were identified through instance segmentation using the Mask based Region-Convolution Neural Network (Mask R-CNN) framework adjusted to the created dataset. Some adjustments were performed to enhance the performance of the classification model, particularly defining an adequate input image size, data augmentation strategy, Intersection over a Union (IoU) threshold during training, and type of backbone network. The inferences show promising results, with correct detections even under complex backgrounds, poor illumination conditions, and instances of significantly reduced dimensions. Furthermore, in scenarios without a roofing system, the model proved reliable, not producing any false positive occurrences. The best model achieved metrics' values equal to 65.1% for the bounding box detection Average Precision (AP) and 59.2% for the mask AP, considering an IoU of 50%. Regarding classification metrics, the precision and recall were equal to 85.8% and 84.0%, respectively. The developed methodology proved to be extremely valuable for guiding infrastructure managers in taking physically informed decisions based on the real assets condition.

Keywords: corrosion; industrial buildings; deep-learning; Mask R-CNN; instance segmentation; Unmanned Aerial Vehicles (UAVs)

Citation: Lemos, R.; Cabral, R.; Ribeiro, D.; Santos, R.; Alves, V.; Dias, A. Automatic Detection of Corrosion in Large-Scale Industrial Buildings Based on Artificial Intelligence and Unmanned Aerial Vehicles. *Appl. Sci.* **2023**, *13*, 1386. https://doi.org/10.3390/app13031386

Academic Editors: Maria Sozanska, Zbigniew Perkowski and Mariusz Jaśniok

Received: 18 December 2022
Revised: 15 January 2023
Accepted: 17 January 2023
Published: 20 January 2023

1. Introduction

The best strategies for the visual inspection of large-scale industrial buildings are still a challenge to be addressed by civil infrastructure engineers. Typically, it is a time-consuming activity, with high human risks and financial costs, that increases in complexity with the surveyed area. In recent years, Unmanned Aerial Vehicles (UAVs) have been incorporated into this task, enabling a remote and enhanced procedure.

Metallic sandwich panels are a versatile solution with properties that ensure a simple on-site installation and durability. In these elements, corrosion represents an early damaged state that can be directly or indirectly responsible for critical failure mechanisms (e.g.,

delamination, debonding, and perforation, etc.), as well as serviceability constraints of the interior spaces of the assets related to loss of impermeability and water leakage.

In recent years, essential developments in software and hardware have led to undeniable advances in Artificial Intelligence (AI) techniques, allowing several novel applications of the image pattern recognition [1], which decisively contribute to the solution of problems such the one of the automatic detection of corrosion on metallic sandwich panels. Furthermore, the combination of these advances with a technology that allows the remote survey of large areas and buildings in real-time [2], i.e., UAVs, could help to diminish the occurrence of fall from height accidents, which was the most critical risk factor associated with construction activities in Great Britain in 2022 [3] and also represents an important concern in the rest of the world [4].

Computer vision has made progress with incorporating deep learning techniques for pattern recognition, in this case, anomaly identification. The landmark of this success was the remarkable performance of the Convolutional Neural Network (CNN) architecture developed in 2012 by Krizhevsky et al. [5], widely known as AlexNet, which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [6].

Nowadays, the most advanced deep learning techniques for image analysis allow performing one or several of the following image recognition tasks, depending on the framework used and the level of information required [7–9]: (i) classification, for identification of the existence (or not) of anomalies, based on the classical CNN algorithm; (ii) detection, which additionally localizes the anomaly and specifies the type of anomaly, based on the so-called Region-CNN (R-CNN) algorithm, and (iii) segmentation, which additionally specifies which pixels belong to each of the identified anomalies, based on the Mask R-CNN algorithm. The Mask R-CNN allows instance segmentation, which consists of the consecutive application of the classification, detection, and segmentation.

In the construction sector, the application of these algorithms can be performed in three distinct areas [10]: (i) health and safety, (ii) management and tracking, and (iii) damage assessment. In the first area, the development of technology to automatically detect the absence of the use of personal protective equipment is a concern. Shen et al. [11] developed a methodology for detecting the use of safety helmets on construction sites. Based on transfer learning and the DenseNet network, these authors created a bounding-box regressor capable of surpassing common challenges of complex backgrounds, like scale variance and perspective distortion. The authors were the first to apply a deep learning technique to this problem successfully. The study points out that the proposed solution is competitive with other existing detection methodologies, like the YOLO's families.

Applications involving asset management and construction progress are more common and broad. For example, Li et al. [12] created a methodology for rebar counting in on-site construction, based on an improved version of the YOLOV3 network [13]. They obtained an average precision for detection equal to 99.7% for an IoU of 50%. However, the proposed methodology proved limited since the counting was only based on the transversal section of the rebars. The high average precision achieved also draws attention, indicating that the model's generalization might be compromised when applied to distinct scenarios in complex backgrounds. Nevertheless, this innovative idea inspired other studies, such as the one developed by Kardovskyi and Moon [14], that proposed a complete methodology to perform steel rebar assessment, resorting to high-performance hardware. In this study, the Mask R-CNN algorithm, with the support of a stereo vision system, was upgraded to measure not only the number of rebars but also the spacing, length, and diameter of the rebars. However, the dataset, containing only 240 images, was the main drawback of the work. Similarly, Xiao and Kang [15] developed a large-scale dataset for machinery operating on the construction site, including a reliable labeling method that enhances detection and classification. Despite this performance, further improvements can benefit the work since it only includes segmentation annotations for particular cases. Furthermore, all the images were taken from the ground level, which is less efficient and more timing consuming when compared to aerial acquisition.

The indoor tracking of the construction process was addressed by Wei et al. [16] with the Mask-RCNN algorithm and a stereo camera to capture 738 images and monitor the execution progress of a base floor including coatings, with the results being transferred to a BIM digital model. This study had the challenge of extrapolating the learning to other building construction stages. In the area of waste management and disposal, which is a current topic of concern, Lu et al. [17] applied semantic segmentation to recognize the composition of construction waste (e.g., rock, stone, packaging, fabric, and wood, etc.), based on a DeepLabv3+ network [18], and achieved a Mean Intersection Over Union (mIoU) of 56%. Chen et al. [19] proposed the application of the Mask R-CNN to estimate the overall built area in rural regions, using open-source satellite images and a transfer learning strategy for the training stage, as well as UAV-acquired images for the test/inference stage. However, this study did not take full advantage of the UAV images, missing the opportunity to use detailed high-resolution images in the training stage.

Damage assessment is currently a significant concern for infrastructure managers and is where most studies involving advanced image processing are performed. Karaaslan et al. [20] proposed a semi-supervised methodology to detect spalling in real-time, providing a 30% improvement in precision compared to a human inspector. Moreover, Santos et al. [21] classified exposed steel rebar images from an industrial building using a CNN, innovatively using the support of a UAV to obtain orthomosaic maps with the identified anomalies.

Instance segmentation was also performed for the damage assessment, but this technique is still underused when compared to other AI algorithms [10]. Zhan et al. [22] used the Mask R-CNN framework and aerial images to precisely identify damaged buildings after the Kumamoto earthquake in 2016, reaching 88% of accuracy but lacking the report of the segmentation metrics. In addition, Hou et al. [23] applied the Mask R-CNN using ground penetrating radar images to automatically detect and segment abnormal instances that might indicate corrosion on concrete bridges, reaching an average accuracy for detection and segmentation of 58.6% and 47.6%, respectively.

Corrosion defects were also identified with machine learning in several applications involving bridges and buildings [24–26]. It is worth noticing other potential applications within the automatic detection of defects in welded joints [27]. However, none of these authors explored the potential of the Mask R-CNN to detect corrosion in metallic structures, with the exception of Forkan et al. [28], who developed a platform based on the Mask R-CNN, called CorrDectector, to segment corrosion in telecommunication towers.

The present work shifts the contributions of AI in the field of civil infrastructure remote inspection, addressing both the lack of applications of instance segmentation algorithms to the area, as well as developing a methodology capable of identifying corrosion on sandwich panels belonging to large-scale industrial buildings. It also creates a novel dataset containing more than 8k high-resolution images acquired with a modern UAV. The labeled dataset contains about 18k segmented instances to overcome the presence of complex backgrounds, typically derived from the use of aerial images and due to the particularities of the location where these industrial buildings are usually situated.

The innovative nondestructive methodology combines data analytics capabilities derived from deep learning through the yet underexplored Mask R-CNN framework, with some proposed adjustments, and the UAV versatility to help management and maintenance planning assess the condition of their buildings. As far as the authors know, this is the first fully dedicated methodology to identify corrosion on metallic sandwich panels efficiently.

2. Methodology for Automatic Detection of Corrosion

2.1. Overview

The methodology for the automatic identification of corrosion in industrial buildings comprises two stages (Figure 1): (i) the image acquisition based on a computer vision system integrated into a UAV platform, and (ii) the image processing by the application of a dedicated AI algorithm. Typically, the pre-programmed flights take a height of 12 m to

15 m alongside the buildings, capturing images from the roofing and façades. The UAV positioning is adjusted using a correction system of the Real Time Kinematic (RTK) type. Finally, the images are processed by a trained Mask R-CNN algorithm that provides each corrosion anomaly's mask, label, and exact location.

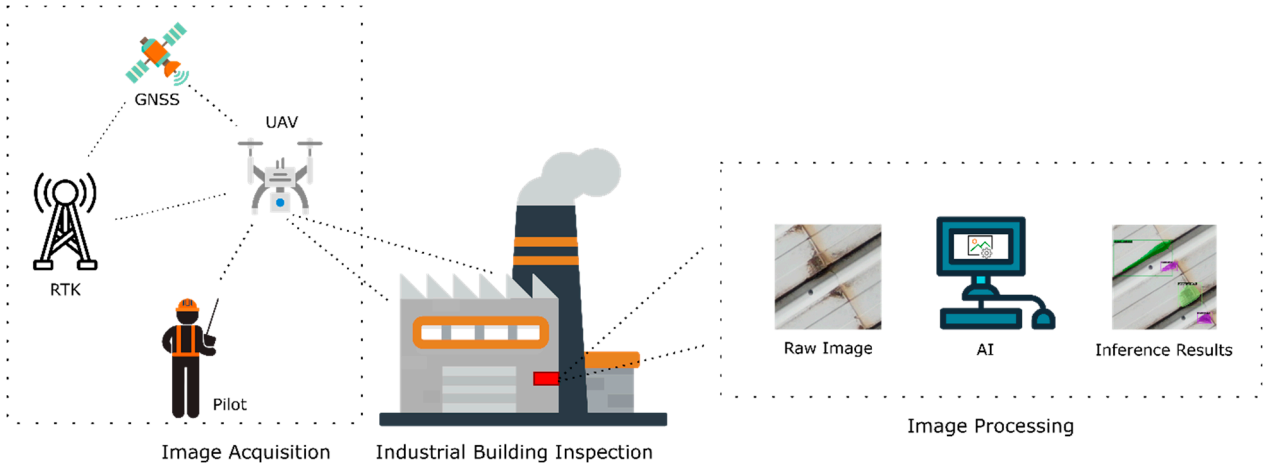

Figure 1. Overview of the methodology for automatic detection of corrosion.

2.2. Equipment

The image acquisition is performed by a drone DJI Mavic 2 Enterprise Advanced (M2EA) (Figure 2). This drone has a maximum recommended take-off weight of 1100 g, reaches a maximum speed of 20 m/s, and a flight autonomy of 30 min. The UAV is equipped with a dual gimbal camera that captures images in visible and infrared (IR) spectra. The first camera comprises a CMOS sensor with a resolution of 48 MPx. It also contains an RTK module that achieves centimeter-level positioning accuracy and supports internet protocol (NTRIP). The processing of images was performed by a PC station running the Windows 11 operative system, equipped with an NVIDIA GTX3090 graphic card with 24 GB of memory, processor i7 11700, 32 GB of RAM, and 1 TB of SSD storage.

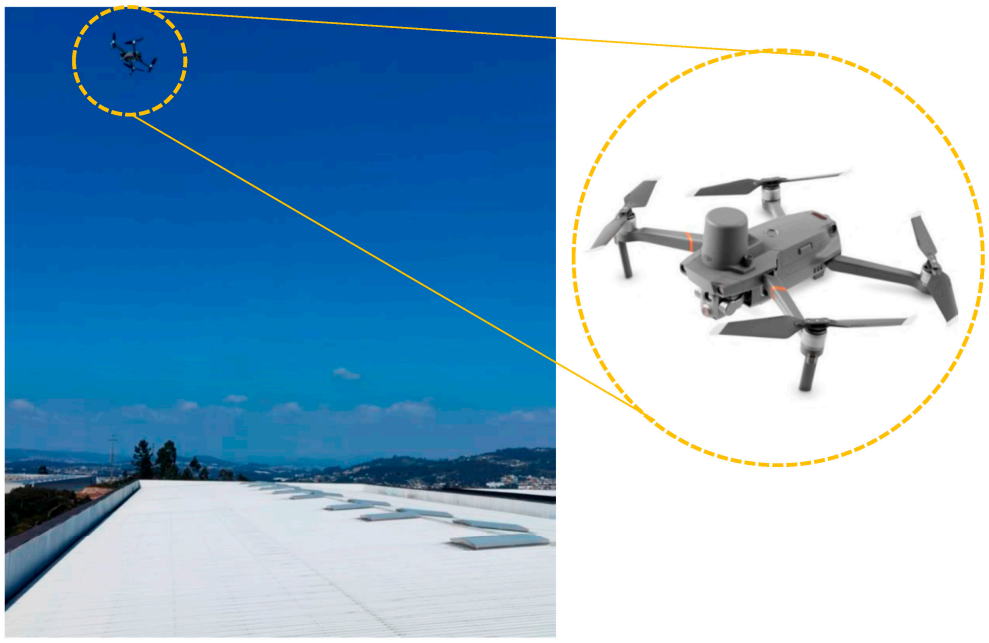

Figure 2. Drone Mavic 2 Enterprise Advanced in operation.

3. Mask R-CNN Framework

Figure 3 presents the main steps for implementing the Mask R-CNN algorithm. The proposed framework involves (i) the creation of the dataset, including the collection of images and the insertion of masks (labeling); (ii) the training process, involving the application of a data augmentation technique, the transfer learning from a predefined dataset, and the hyperparameter tunning; and (iii) the test of the final model based on appropriate metrics. The singularity of the solution consists in performing instance segmentation based on high-resolution aerial images of metallic roofing systems of large-scale industrial buildings within a competitive processing time. The programming language used was Python 3.7. In the following sections, the steps of the proposed framework are discussed in detail.

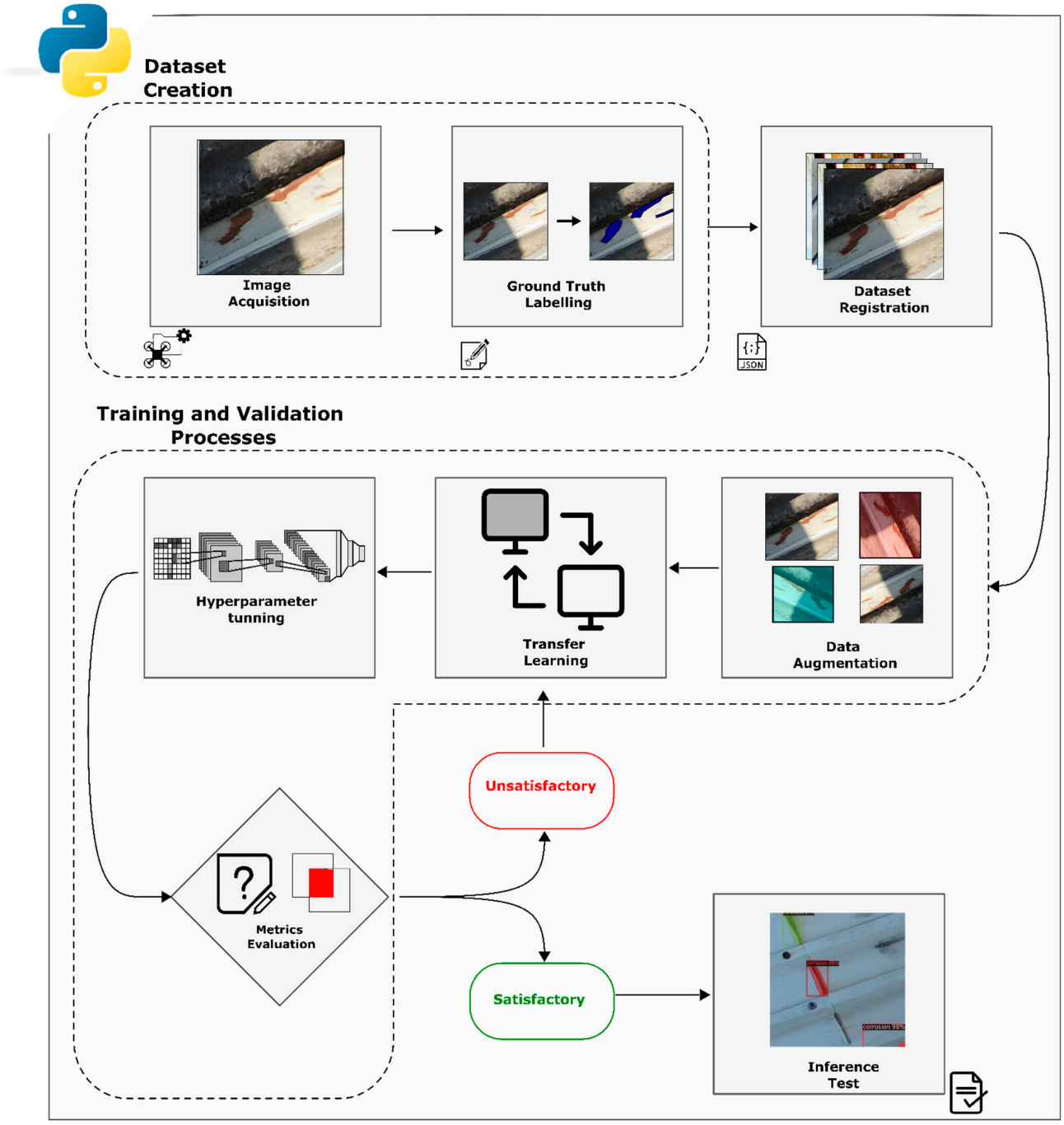

Figure 3. Framework of the Mask R-CNN algorithm.

3.1. Image Dataset

The image dataset is a vital aspect of the success of AI-based computer vision algorithms. A sufficient and consistent number of images should exist to comprehend as many situations as possible in the real environment. In a supervised learning approach, the definition of ground truths, i.e., the precise annotation of the anomalies based on direct observation, is also crucial since they constitute the targeting for the learning of the Mask R-CNN algorithm, being also considered the most time-consuming task.

3.1.1. Image Acquisition

The database contains 8400 images, with a resolution of 2000 × 2000 pixels, totalizing 18,381 corrosion instances. The images were collected under good meteorological conditions, particularly at high sunlight exposure, from several buildings located in industrial zones on the north of Portugal. Figure 4 presents some industrial facilities used for the image database collection, where the roofing systems are marked in purple. The aerial missions totalize more than 18,000 m^2 of roofing systems under complex backgrounds, including vegetation, people, cars, and roads, etc.

Figure 4. Some of the industrial buildings used for the image database composition.

3.1.2. Ground Truth Labelling

Ground truth annotations were performed with the Visual Image Annotator (VIA) software, an open-source solution from the Oxford Visual Geometry Group [29]. It is a very user-friendly software that runs directly in the internet browser. The segmentation annotations were manually defined, circumscribing the corrosion instances, and posteriorly exported in a JSON (JavaScript Object Notation) format.

3.2. Algorithm

The Mask R-CNN framework was developed from a family of CNN-based solutions that started with the R-CNN. Currently, the Mask R-CNN model has the best benchmark scores regarding instance segmentation [9]. This model combines multiple algorithms from

computer vision and artificial intelligence fields. Therefore its architecture, is formed by several regions associated with different operators, as presented in Figure 5.

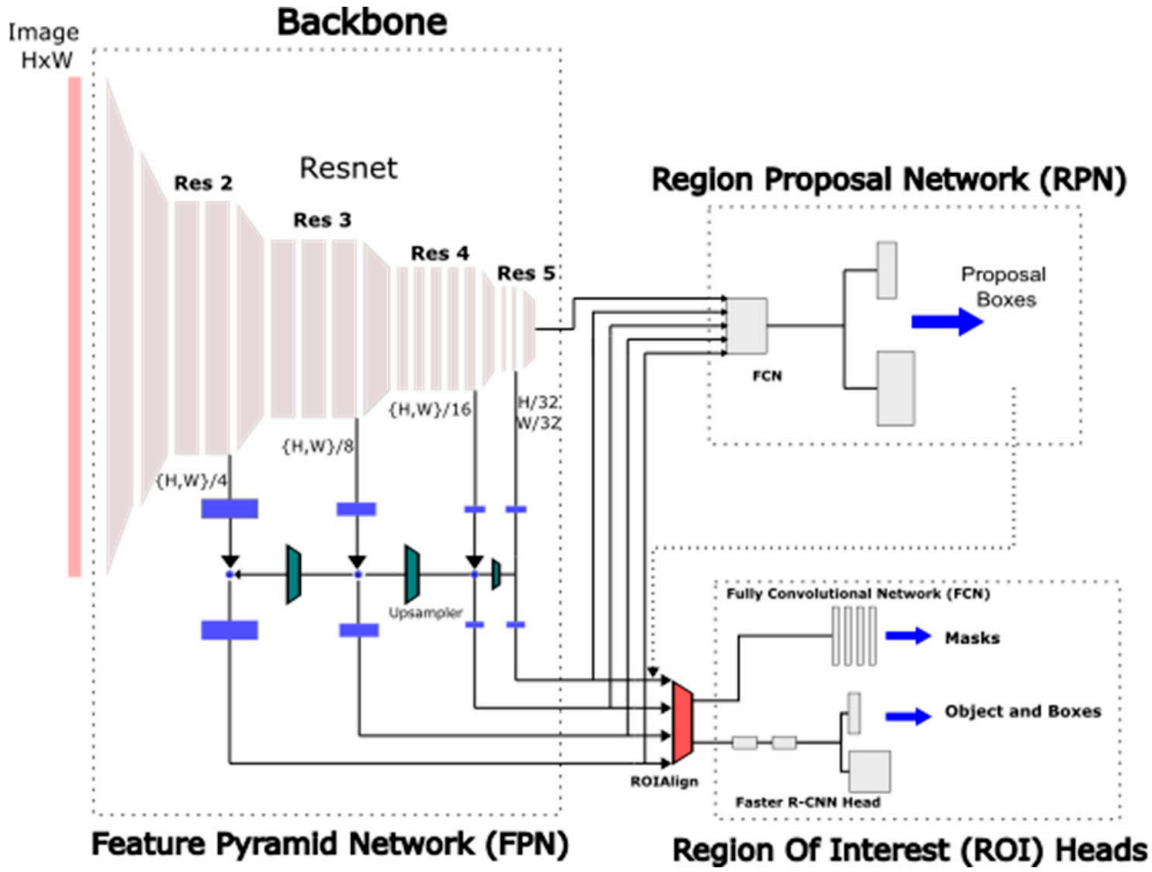

Figure 5. General Mask R-CNN architecture [30].

This work adopted the Mask R-CNN implementation denominated Detectron2 [31], a Python library developed by the Facebook AI research team. It was adapted with minimal modification, simply through training, finding the best parameters of image size, data augmentation, backbone network, and Region of Interest Intersection Over Union, as will be discussed in Section 4. In the following subsections, the primary operations performed by this framework are detailed, particularly the backbone network, the Region Proposal Network (RPN), the Feature Pyramid Network (FPN), the RoI Align, and RoI Heads.

3.2.1. Backbone Network

The backbone network is the CNN responsible for taking the image as input and performing the feature extraction. The models of the Resnet series are the most widely adopted within CNNs architecture. As depicted in Figure 6, the model is composed of residual connections between different convolutional layers. This approach allows one input image (x) to have multiple output features from different convolutional operations stages, denominated Res2, Res3, Res4, and Res5. It is assumed that a series of stacked nonlinear layers can map residual functions such as $H(x) - x$. However, in the Resnet series, shortcut connections are defined, setting $F(x) := H(x) - x$ and yielding to $H(x) := F(x) + x$ [32], where $F(x)$ represents the target task to be learned and $H(x)$ the mapped function. This solution tackles two common problems in regular CNNs: (i) degradation of accuracy and (ii) elimination of vanishing gradient since the outputs of the convolutional blocks are always non-zero values.

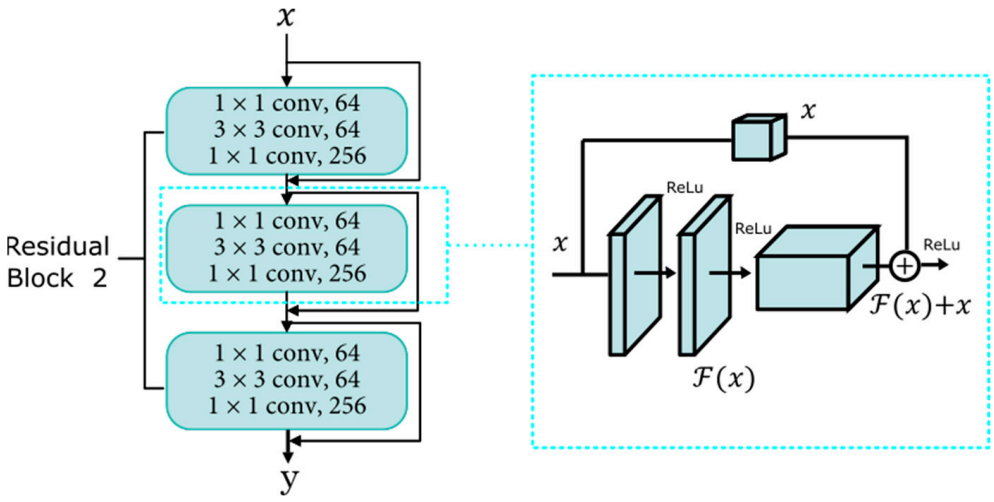

Figure 6. Detail of the residual block 2 (Res2) and its shortcut connection.

3.2.2. Region Proposal Network

The Region Proposal Network (RPN) consists of a sliding window algorithm based on a small convolutional network. This operator inputs the features map from the backbone and outputs object detection probability, as well as a series of rectangular region proposals, the so-called object bounding boxes. Considering the variety of object sizes in a dataset, each sliding window takes multiple scales and aspect ratios centered at an anchor (Figure 7).

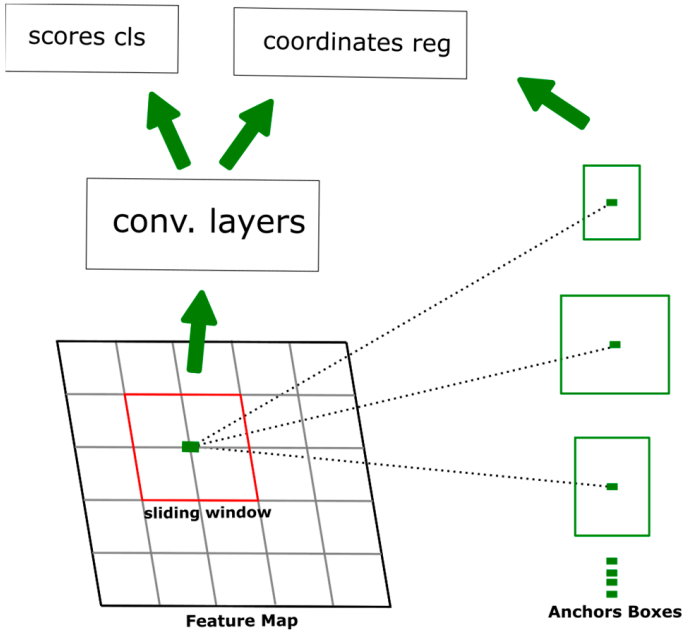

Figure 7. Region Proposal Network (adapted from [8]).

Typically, a feature map of dimension $W \times H$ will produce $W \times H \times k$ anchors, with W and H as the width and height of the feature map, respectively, and k as the number of aspect ratios times the number of scales. After being mapped to a lower dimensional feature, the sliding windows will feed two fully connected layers, one for regression and another for classification, being trained according to [8]:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where p_i and p_i^* are the probabilities of a predicted anchor i and corresponding ground truth, respectively. N_{cls} and N_{reg} are regularization terms represented by the batch size and anchor locations, L_{cls} is the binary log-loss, and λ is a balancing parameter whose default value is 10. The rectangular coordinates (t_i, t_i^*) are parametrized accordingly to [8] and L_{reg} is the smooth L_1 function, defined in [33].

3.2.3. Feature Pyramid Network

The Feature Pyramid Network (FPN) comprehends the lateral connections that are linked to the backbone network stages (res2, res3, res4, and res5) in a top-down and bottom-up approach (see Figure 5). Its primary purpose is to sample feature maps from the different stages, detecting objects in as many scales as possible and without loss of efficiency [34]. Then, it feeds the RPN, which will decide the best scale to extract the feature maps based on the following equation:

$$k = \left[k_0 + log_2\left(\sqrt{wh}/224\right)\right] \tag{2}$$

where k_0 is a constant equal to 4, and w and h are the feature maps dimensions.

Parameter k is a value between 2 and 5, corresponding to feature maps of multiple scales in the backbone network (Res2 to Res5 in Figure 5).

3.2.4. RoI Align

The Region of Interest (RoI) Align extracts the features from the maps indicated by the RPN, performing a bilinear interpolation to pool the information, preserving the pixel spatial correlation, accordingly to [9].

3.2.5. RoI Heads

The RoI Heads region of the mask R-CNN is responsible for performing the final prediction of the object specifying its class, rectangular bounding boxes, and masks (Figure 5). The first two tasks are performed by the Fast-R-CNN head [33] with a softmax function for multiclass classification and, again, the smooth L1 function for bonding box regression (Equation (2)). The mask is predicted in its own branch, formed by a Fully Convolutional Network (FCN) with seven convolutional layers. A per-pixel average binary cross entropy loss is used for training:

$$L_{mask} = \frac{-1}{N}\{p^* \cdot logp + (1 - p^*) \cdot log(1 - p)\} \tag{3}$$

where N is the total number of samples. This loss function does not compete with the classification one, since it is only binary, which was pointed out as the most important factor for the success of the framework [9]. For each sampled RoI, a multi-task loss is defined as:

$$L_{total} = L_{class} + L_{box} + L_{mask} \tag{4}$$

with L_{class} and L_{box} defined as in Equation (2).

3.3. Training and Validation

The images were divided into three datasets: train, validation, and test, in a proportion of 70:15:15, respectively. In the training stage, the influence of the primary hyperparameters of the Mask R-CNN algorithm is tuned to optimize the model's performance. Moreover, data augmentation techniques and transfer learning strategies are used to enhance the efficiency and robustness of the proposed model. Finally, the validation and test stages perform inferences to evaluate the performance of the algorithm using dedicated metrics. Figure 8 summarizes the sequential steps of the procedure.

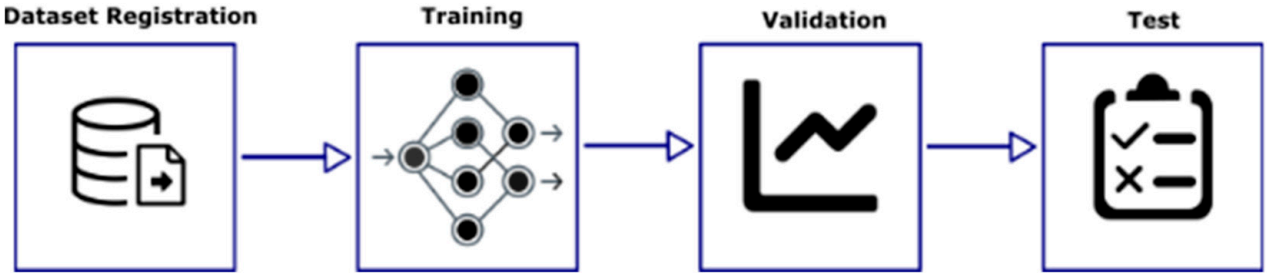

Figure 8. Training, validation, and test.

3.3.1. Data Augmentation

Morphological data augmentation operations in AI have been considered good practice since the early applications of CNNs to enhance model generalization [35]. It usually refers to transformations such as flipping, rotating, lighting, and distortion. The adopted data-augmentation technique, denominated as on-the-fly, guarantees that the generated data is not stored in memory but randomly created at each training attempt and then discarded. Attending to the nature of each application, the datasets will have different optimum data augmentation strategies [36].

The random data augmentation techniques used in this study are illustrated in Figure 9 and for the case of brightness, saturation, and contrast operations is performed by:

$$M_{out} = M_{ori}(1 - \alpha) + M_{transf} \times \alpha \quad (5)$$

where M_{out} and M_{ori} represent the processed and original images, respectively, M_{transf} is the transformation matrix according to the type of operation (see Table 1) and α is the intensity factor.

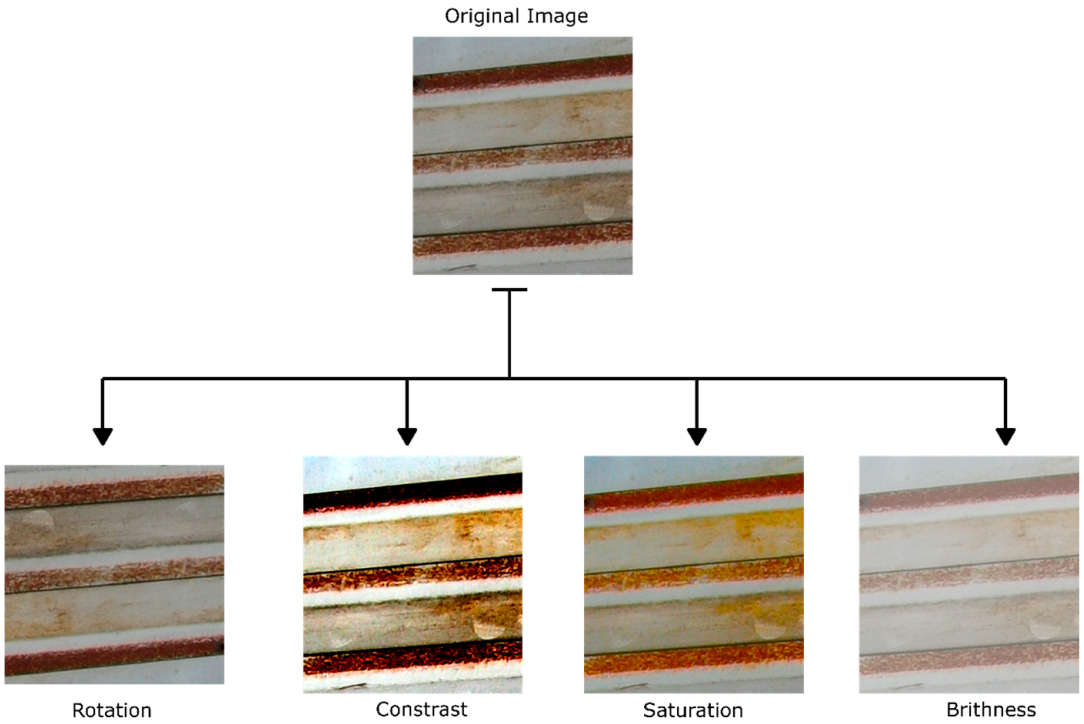

Figure 9. Data augmentation operations.

Table 1. Data augmentation operations of brightness, saturation, and contrast.

Operation	α	M_{transf}
Brightness	1.1 & 1.3	Black version of the original image (0 for all pixels in the RGB channels).
Saturation	1.2 & 1.4	Gray scale version of the original image transforming each pixel according to: $0.299R + 0.587G + 0.144B$
Contrast	1.2 & 1.4	Mean of pixels of the gray scale version of the original image, transformed with the same equation of saturation

For the specific case of rotation operation, only a simple image rotation of 180° was considered. All procedures were performed using the Python library denominated Pillow [37], and the probability of the operation applied to each dataset image varies between 0.10 and 0.50.

3.3.2. Transfer Learning

Transfer learning is a common strategy used in machine learning to enhance the model's convergence to the optimal solution [38]. It is usually performed by taking the hyperparameters from an already trained model in a large dataset and applying them to a new model considering the same architectures. In this work, transfer learning was adopted from the Microsoft Common Objects in Context (MS COCO) dataset [39], one of the largest already produced, containing 330k images with more than 91 categories. Figure 10 illustrates the transfer learning strategy adopted in this study.

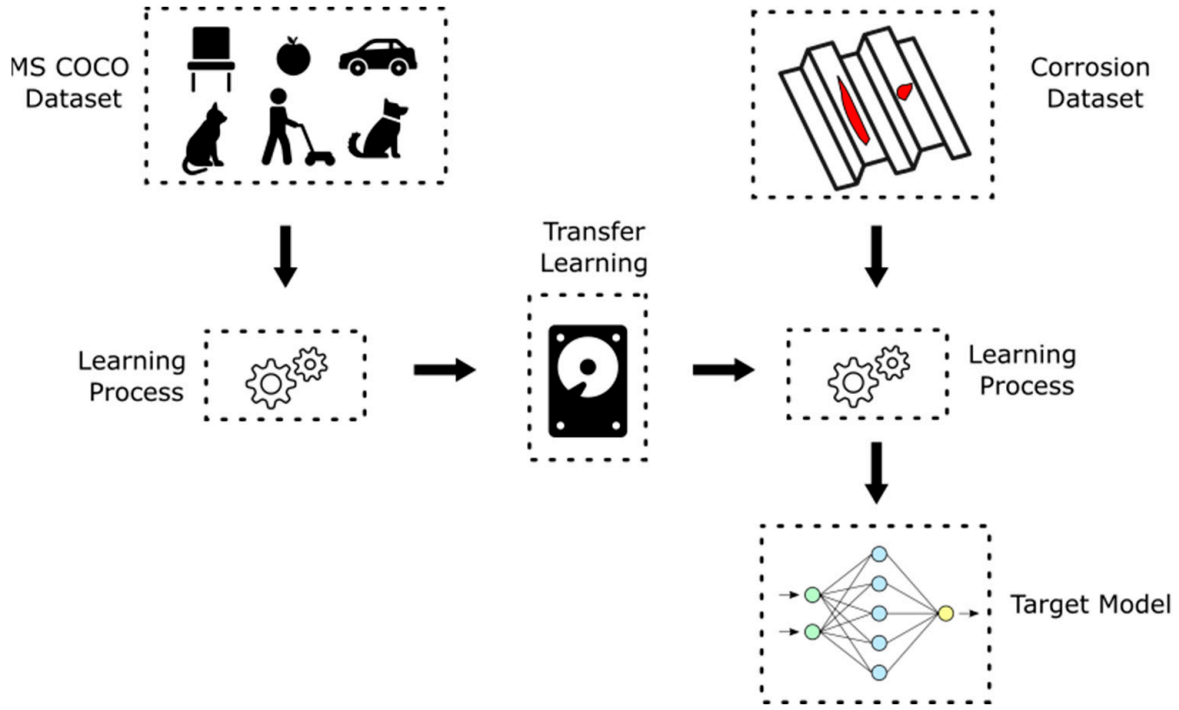

Figure 10. Adopted transfer learning strategy.

3.3.3. Metrics

The quality of the model was basically evaluated in terms of precision. For the detection and segmentation, the precision metrics are the Recall (R), Precision (P), and $F1$ score, which are defined as:

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$F1 = 2 \times \frac{R \times P}{R + P} \tag{8}$$

where TP are the true positives, FN the false negatives, and FP the false positive samples. The Precision $\times$ Recall curve (PR curve) captures the inverse proportional relationship between these metrics, and the Area Under the Curve (AUC) defines the so-called average precision (AP):

$$AP_{\text{IoU}} = \sum_{i=0}^{N} [w_i P(r_i)] = AUC \tag{9}$$

It is computed numerically with a given weight (w_i) and a precision that decreases monotonically with the recall rate (r_i) at a fixed Intersection Over Union (IoU) [39].

The IoU attest to the quality of the location of objects and their masks in the images. Specifically, the IoU is the percentage of overlapping between a ground truth and its prediction:

$$\text{IoU} = \frac{Area\ (B_p \cap B_{p*})}{Area\ (B_p \cup\ B_{p*})} \tag{10}$$

where B_p and B_{p*} stand for the areas of the predicted instance and ground truth, respectively.

Furthermore, the AP_{IoU} is a relevant metric to understand the percentage of instances being detected at a given superposition with the annotated data.

3.3.4. Dataset Registration, Training, and Validation

Dataset registration involves extracting the information from the ground truth to an input format compatible with Detectron2 to initialize the training stage. The training dataset contains 5880 images and 12,236 instances of corrosion. Then, the hyperparameters are set in a key-value configuration system with YAML (Ain't Markup Language^{TM}), a human-friendly data serialization language.

To enhance the performance of the ML an adequate selection of the values of the hyperparameters must be made. The hyperparameter values were manually defined based on recommendations from the literature and previous experience resorting to a trial-and-error approach. The definition of each hyperparameter can be consulted in references [9,10]. Parameter optimization was made with Adam [39], using learning rate warm-up and decay. Batch normalization was also adopted. Table 2 shows the most relevant hyperparameters modified within this work, with the remaining settings defined according to [9]. Data augmentation is performed after the serialization of the hyperparameters according to Section 3.3.1.

Table 2. Hyperparameters and adopted values for training of Mask R-CNN.

Hyperparameter	Adopted Value
Learning rate	0.0001
Momentum	0.9
Region of Interest Head batch size	512
Images per batch	2
Number of iterations	40,000
Validation period	500
Region of Interest Head IoU	0.53, 0.55 and 0.60
Backbone network	Resnet-50 and Resnet-101

The validation dataset contains 1260 images with 3186 segmented instances. The model validation occurred at every 500 iterations based on the metrics of average precision with IoU of 50% and 75%, for both detection and segmentation, using the COCO API [39]. Additionally, the total loss was estimated to verify the occurrence of overfitting.

3.4. Test

The test dataset contains 1260 images with 2959 instances. The metrics used for its evaluation were the same applied for the validation test. Additionally, the precision and recall, for the classification of each instance was computed, to give a measure of the quality of the application at this level.

4. Model Evaluation

In this section, the evaluation of the model performance is presented for the training and validation stages. For this purpose, a sensitivity analysis was performed with the following parameters: (i) size of the images used in the training stage (Section 4.1); (ii) strategy adopted for Data Augmentation (Section 4.2); (iii) influence of the ROI Heads IoU hyperparameter (Section 4.3); and (iv) influence of the Backbone network (Section 4.4). In this sensitivity analysis, detection and segmentation metrics were used, denoted by the acronyms detec and segm, respectively. This parametric study was performed with the aim of defining the best strategy to reduce the computational cost without compromising the efficiency and robustness of the methodology.

4.1. Image Size

The processing of the original images, with the size of 8000 × 6000 pixels, is not feasible due to hardware limitations. Thus, two different approaches were evaluated: (i) resize the images for 1333 × 800 pixels, which are the Detectron2 default dimensions; and (ii) cropping the images with dimensions of 2000 × 2000 pixels.

The first approach has the advantage of keeping the original contours of the anomalies, but it loses information due to the small size that usually has the corrosion instances. The second approach maintains the image's resolution but takes more training time and the contours of some cases might change because of the cropping edges.

Table 3 shows the maximum percentage values for the precision metrics for models based on resized and cropped images for the validation dataset and, also, the iteration of occurrence, which represents other states of the model presented. It is possible to notice satisfactory metric values for the cropped images in terms of precision with IoU 50% for the detection and segmentation. These values are about 10% higher than those obtained from the resized images. The iteration number where these maximum values occur indicates the importance of running the algorithm with 40k training cycles.

Table 3. Metrics values for models based on resized and cropped images for the validation dataset.

	Resized Images		Cropped Images	
	Maximum Value (%)	**Iteration**	**Maximum Value (%)**	**Iteration**
AP_{50}^{Detect}	43.5	26,999	55.8	18,499
AP_{50}^{Segm}	37.4	25,999	50.5	17,499
AP_{L}^{Detect}	47.9	26,999	43.6	20,499
AP_{L}^{Segm}	32.5	25,999	31.0	13,499

Figure 11 shows the examples of three inferences (EX1 to EX3) performed by models trained with resized and cropped images in the validation dataset. Example 1 demonstrates close results between the two models but with one more instance detected by the model using cropped images. In Example 2, the model with resized images detected an instance close to one edge, showing that the shape of the corrosion is a determinant factor since, in this specific case, the cropping/separation of the image changed a larger dimension instance. Example 3 illustrates the substantial improvement that the cropped images can provide in small dimension instances, with 9 out of 12 (75%) annotates detected, while in the resized model, no annotation was detected. This is because the small instances become

imperceptible to the algorithm, resembling only noise since the resized model reduces the image by approximately eight. These results motivated the selection of the strategy where images are cropped before entering the training process.

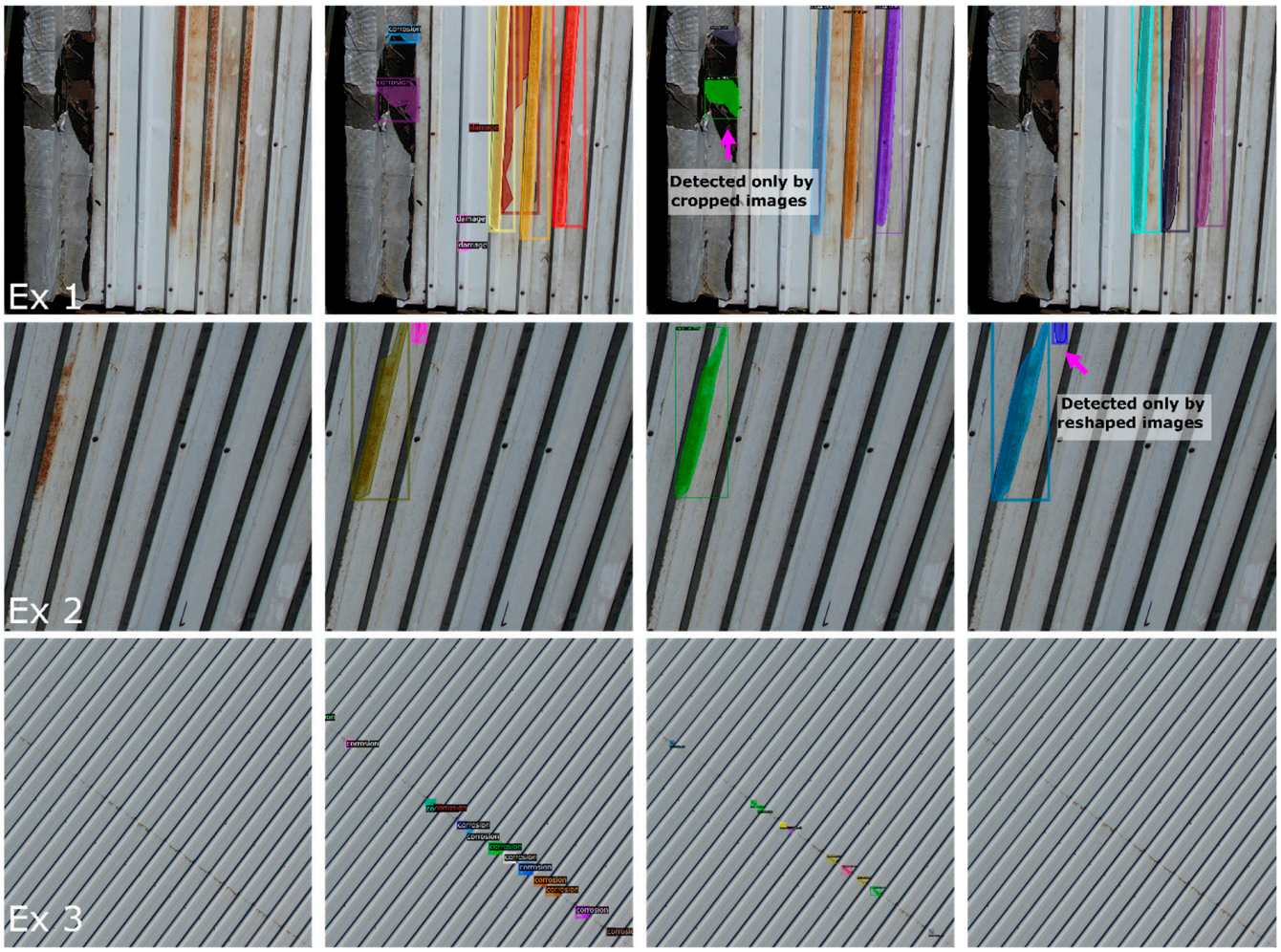

Figure 11. Evaluation of the influence of image size on the efficiency of image classification based on different examples (EX1 to EX3): (**a**) original images, (**b**) ground truth, (**c**) inference with cropped images, and (**d**) inference with resized images.

4.2. Data Augmentation

Data augmentation operation increases the data variability and the generalization of the model; however, the intensity of its application can also affect the precision of the model, especially for small-scale datasets [36]. Table 4 illustrates the three analyzed data augmentation (DA) strategies, particularly, no DA, moderate DA, and intense DA, as well as their corresponding intensities (α) and application probabilities (p).

Table 5 shows the results for the precision metrics for each of the analyzed models. It is a visible improvement from the model without DA to the model with moderate DA. The model with intense DA demonstrates the degradation of the precision metrics when more intense transformations are applied.

Figure 12 shows the training and validation curves of the three models based on the total loss function values. These graphs allow the verification of the overfitting phenomena precisely on the iteration when the training and validation curves are convergent. This occurs due to the increasing adjustment of the model to the characteristics of the training dataset and therefore, the classification becomes more and more efficient. However, under

these circumstances, the model tends to fail more in the prediction of new situations since they lose their ability to generalize. This phenomenon is evident from iteration number 15k and seems more visibly in the model without DA (Figure 12a), while in models with moderate and intense DA it is almost imperceptible. This result was somehow expected, due to lack of variability in the training data when no DA is applied.

Table 4. Implemented data augmentation strategies.

Transformation		Saturation	Contrast	Brightness	Rotation
No DA	α	No application			
	p				
Moderate DA	α	1.2	1.2	1.1	180°
	p	0.1	0.1	0.1	0.1
Intense DA	α	1.4	1.4	1.3	180°
	p	0.5	0.5	0.5	0.5

Table 5. Metrics values for models without DA, moderate DA, and intense DA.

	No DA		Moderate DA		Intense DA	
	Maximum Value (%)	Iteration	Maximum Value (%)	Iteration	Maximum Value (%)	Iteration
AP_{50} Detect	55.8	18,499	57.1	35,999	53.8	36,499
AP_{50} Segm	50.5	17,499	51.9	35,999	48.9	22,999
AP_L Detect	43.6	20,499	46.3	38,999	46.1	27,999
AP_L Segm	31.0	13,499	33.4	29,499	32.8	25,499

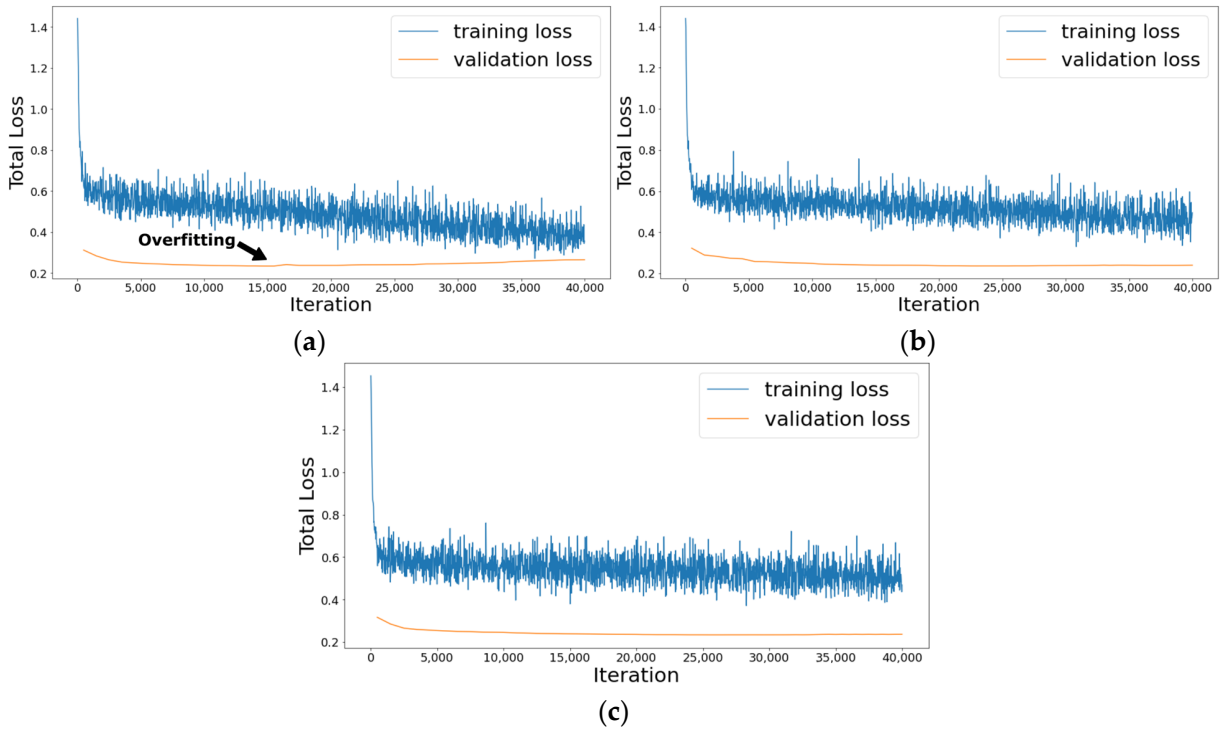

Figure 12. Training and validation curves for the loss function: (**a**) without DA, (**b**) moderate DA, and (**c**) intense DA.

In addition, it is observed in all graphs of Figure 12 that the validation loss error is smaller than the one correspondent to the training. This is because the validation dataset was run without DA and it was adopted a warm-up range of 500 iterations for starting the validation. This period reduces the relevance of the early training avoiding extra iterations to obtain the convergence. It was not observed the leakage phenomenon due to training images wrongly allocated to the validation stage.

Finally, as conclusion, the results indicate that the most suitable strategy is the moderate DA, because is able to create variability in the data and leading to an improvement of the model reflected in an enhanced generalization.

4.3. Region of Interest Intersection over Union

The ROI IoU hyperparameter defines the minimum percentage of overlap between the predicted instance and the ground truth to an inference to be classified as positive. This hyperparameter is applied in the ROI Heads region of Mask R-CNN (Section 3.2.5) and its adjustment can lead to a more optimized training process, depending on the similarity that the classified instances have among themselves.

Table 6 presents the results of the three evaluated models, with RoI IoU values equal to 0.55, 0.57, and 0.60, chosen according to the default value of 0.50 [9]. The results show that the RoI 0.55 and RoI 0.57 models present very similar metrics with a slight superiority of the model with RoI 0.57. The RoI 0.60 model presents a very evident degradation of all metrics, which may lead to a strong bias in the results of some dataset samples.

Table 6. Metrics values for models RoI IoU 0.55, 0.57, and 0.60.

	RoI IoU 0.55		**RoI IoU 0.57**		**RoI IoU 0.60**	
	Maximum Value (%)	**Iteration**	**Maximum Value (%)**	**Iteration**	**Maximum Value (%)**	**Iteration**
AP_{50} ^{Detect}	56.4	24,499	57.1	35,999	25.5	16,999
AP_{50} ^{Segm}	51.8	26,499	51.9	35,999	21.1	9999
AP_{L} ^{Detect}	46.2	24,999	46.3	38,999	26.0	18,499
AP_{L} ^{Segm}	33.3	24,999	33.4	29,499	15.7	16,999

Figure 13 shows the loss function curves for models with ROI Heads IoU equal to 0.55, 0.57, and 0.60. Figure 13c shows that high values for the IoU ROI cause the model overfitting with an increase in the validation loss function, approximately after 20 k iterations, and consequent loss of generalization of the algorithm.

These results points toward a RoI IoU Head parameter of 0.57 as the one with the best performance, showing that a medium overlapping between the ground truth and the predictions are adequate to optimize the performance of the algorithm.

4.4. Backbone Network

The backbone network is responsible for extracting features that will be processed and classified along the convolution layers of the algorithm. A higher number of convolution layers represents a greater classification capacity; however, it may lead to premature overfitting and consequently less favorable metrics. Deeper networks also tend to overload computational resources in such a way that make their use unfeasible.

Table 7 shows the values of the main metrics obtained for the Resnet-50 as well as Resnet-101, both including FPN (see Section 3.2) and considering the best hyperparameters identified in the previous sections. The results show the superiority of the Resnet-101 network in all the precision metrics, especially for AP_{50} ^{Detect} and AP_{50} ^{Segm} that increased more than 10%. The Resnet-50 network presents better results for larger instances. In turn, Resnet-101 detects a greater amount of small and medium instances and in large instances is less efficient.

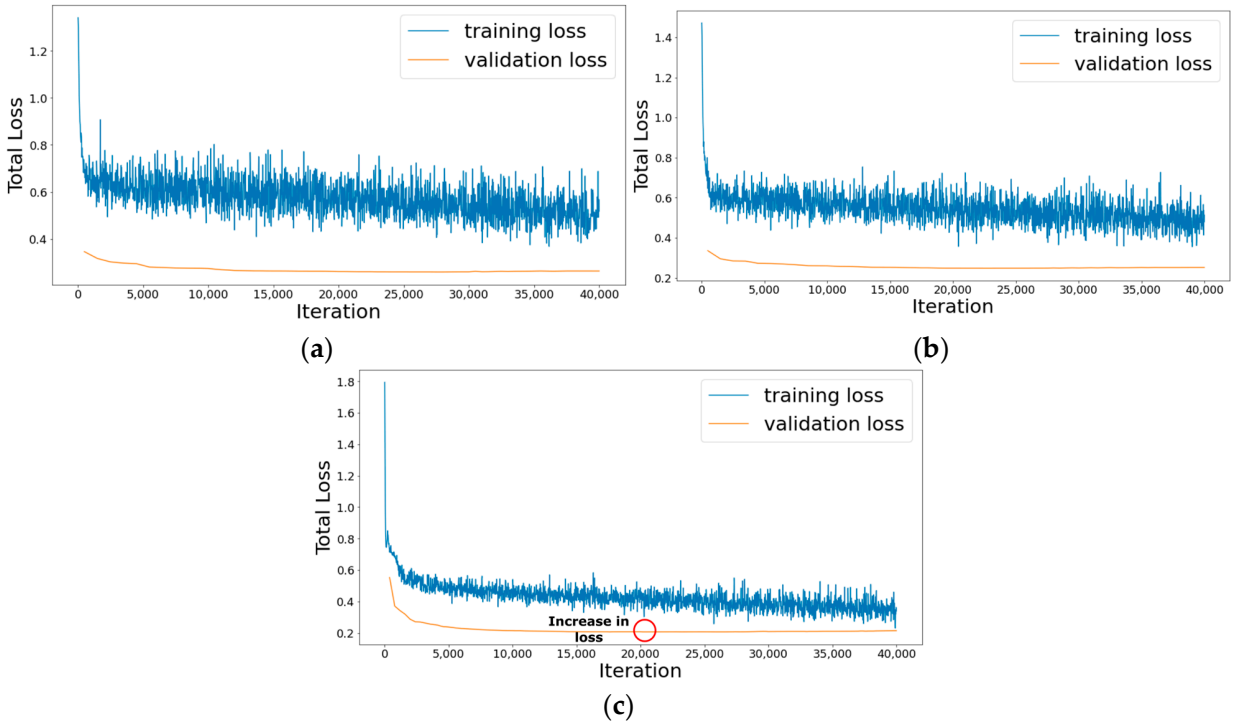

Figure 13. Training and validation curves for the loss function considering RoI IoU Head value equal to: (**a**) 0.55, (**b**) 0.57, and (**c**) 0.60.

Table 7. Metrics values for backbone networks Resnet-50 and Resnet-101.

	Resnet-50		Resnet-101	
	Maximum Value (%)	**Iteration**	**Maximum Value (%)**	**Iteration**
AP_{50}^{Detect}	57.1	35,999	69.2	29,999
AP_{50}^{Segm}	51.9	35,999	64.9	20,999
AP_{L}^{Detect}	46.3	38,999	39.6	31,999
AP_{L}^{Segm}	33.4	29,499	33.4	24,999

Figure 14 shows the training and validation curves for the loss function for the Resnet-50 and Resnet-101 networks. It is possible to notice an enhanced tendency of overfitting for Resnet-101 in comparison to Resnet-50, as stated by the greater end values of the validation loss function (0.31 against 0.24). This behavior was expected due to the increased number of parameters handled by the Resnet-101 network.

Figure 15 shows three examples of classification (EX1 to EX3) performed by both networks based on images from the inference dataset. It is possible to notice the improvement provided by Resnet-101 network, particularly in the classification of minor corrosions, as stated by Example 1, although not all instances were identified as revealed by the comparison with the ground truth. Example 2 shows that Resnet-50 and Resnet-101 detected several corrosions not identified in the ground truth, demonstrating some extrapolation capacity of this type of algorithms. In this situation, Resnet-101 performed a better classification by not interpreting a dark spot as an anomaly. This capacity comes from the higher number of convolutional channels of this network, which allows the information to be acquired in a more granular level as the depth is increased [34]. Example 3 shows a case in which the poor lighting imposed limitations in the identification of small corrosions on the image labelling, but when viewing the image in detail, particularly in the extremities of the panels, it is possible to confirm that the pathology exists, and therefore, the algorithm again

performed an extrapolation according to learning. Classifications of this type penalize precision metrics; however, they are very common in supervised algorithms.

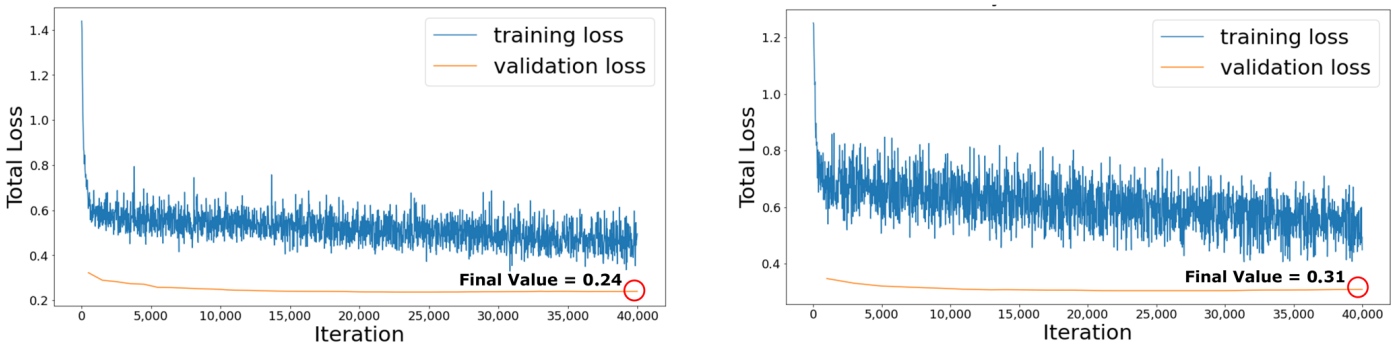

Figure 14. Training losses for (**a**) Resnet-50 (**b**) Resnet-101.

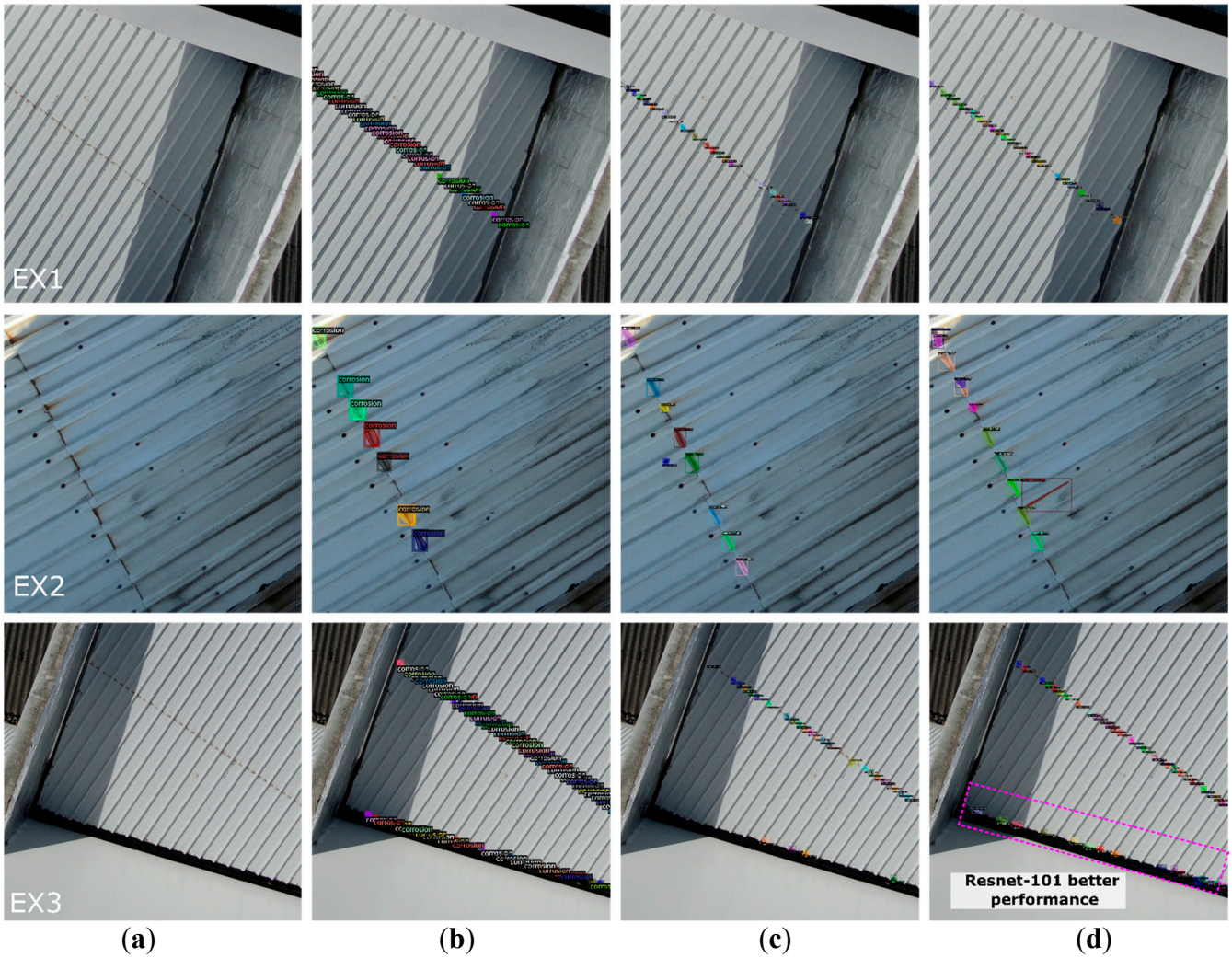

Figure 15. Evaluation of the influence of the backbone network on the efficiency of image classification based on different examples (EX1 to EX3): (**a**) original images, (**b**) ground truth, (**c**) Resnet-50, and (**d**) Resnet-101.

5. Model Application

This section shows the results of the test stage, both in terms of metrics and inference, aiming to evaluate the performance of the model in new real-world situations. The test stage was carried out considering a new set of images and considering the learnings derived from the model evaluation (Section 4), particularly the use of: (i) cropped images with dimensions of 2000 × 2000 pixels; (ii) moderate data augmentation strategy; (iii) ROI IoU hyperparameter equal to 0.57, and (iv) Resnet-101 network.

5.1. Metrics Evaluation

Table 8 presents the results of the main detection and segmentation metrics for the test dataset. It is possible to notice a similarity with the values obtained for the validation dataset which proves the robustness of the model. Additionally, Table 9 presents the results for each classified instance of the test dataset, most of them are true positives, showing the assertiveness of the model. From these numbers were obtained a recall and precision equal to 85.8% and 84.0%, respectively. The inference total time per image, for CPU and GPU, where 7.64 s and 0.49 s, respectively, which are competitive values envisaging real time applications.

Table 8. Precision metrics values for the test dataset.

Precision Metric	(%)
AP_{50} ^{Detect}	65.1
AP_{50} ^{Segm}	59.2
AP_{L} ^{Detect}	35.6
AP_{L} ^{Segm}	28.2

Table 9. Results of each classified instance of the dataset.

Parameter	Number of Samples
True positives	2514
False negatives	445
False positives	515

5.2. Results

Figure 16 shows three inference results (EX1 to EX3) where the classified instances are similar to the ground truth. Example 1 demonstrates the detection of small instances under normal lighting conditions, showing a successful inference given the similarity between the prediction and the ground truth. Example 2 shows a situation where a small dark spot was mistaken for corrosion, but all other instances, also of reduced size, were correctly detected. The several identified corrosion points are located at the ends/contours of the plates (in the intersection between panels), which illustrates that the difficult conditions in which the algorithm demonstrated assertiveness. It should also be noted that the roofing shape used in the prediction image is slightly different from the geometries used in training. Finally, Example 3 shows situations in which the model detected small instances at the edges of the roofing system and in poor lighting conditions (shadow zones). The algorithm successfully identified all the marked instances and even detected unlabelled corrosion by extrapolation.

Figure 17 shows three examples of inferences (EX1 to EX3) in the presence of complex backgrounds. In Example 1, the instances were successfully predicted, even with two similar roofing systems and some shading. The fiber cement roofing system has a texture that seems like a corrosion anomaly. However, no false positives were produced. Example 2 successfully identifies small and large instances under complex backgrounds. This identification is challenging because it is an interface region with a fiber cement roofing

system, with some visible dirt over the metallic plates and under different lighting conditions. Finally, example 3 illustrates the detection capability of the algorithm in a complex background with several debris and under quite distinct lighting conditions.

Figure 18 shows two examples (EX1 and EX2) where corrosion instances occur in areas outside the roofing system. In both situations, the model did not produce any false positives in not labeled elements with corrosion since they were not part of the roofing system. This is the case of the protection grid of a rainwater culvert located in an adjacent street, in the case of Example 1, and on roofing elements that are not metallic plates but are located close to them (e.g., metallic inclined hangers and other types of roofing system), in the case of Example 2.

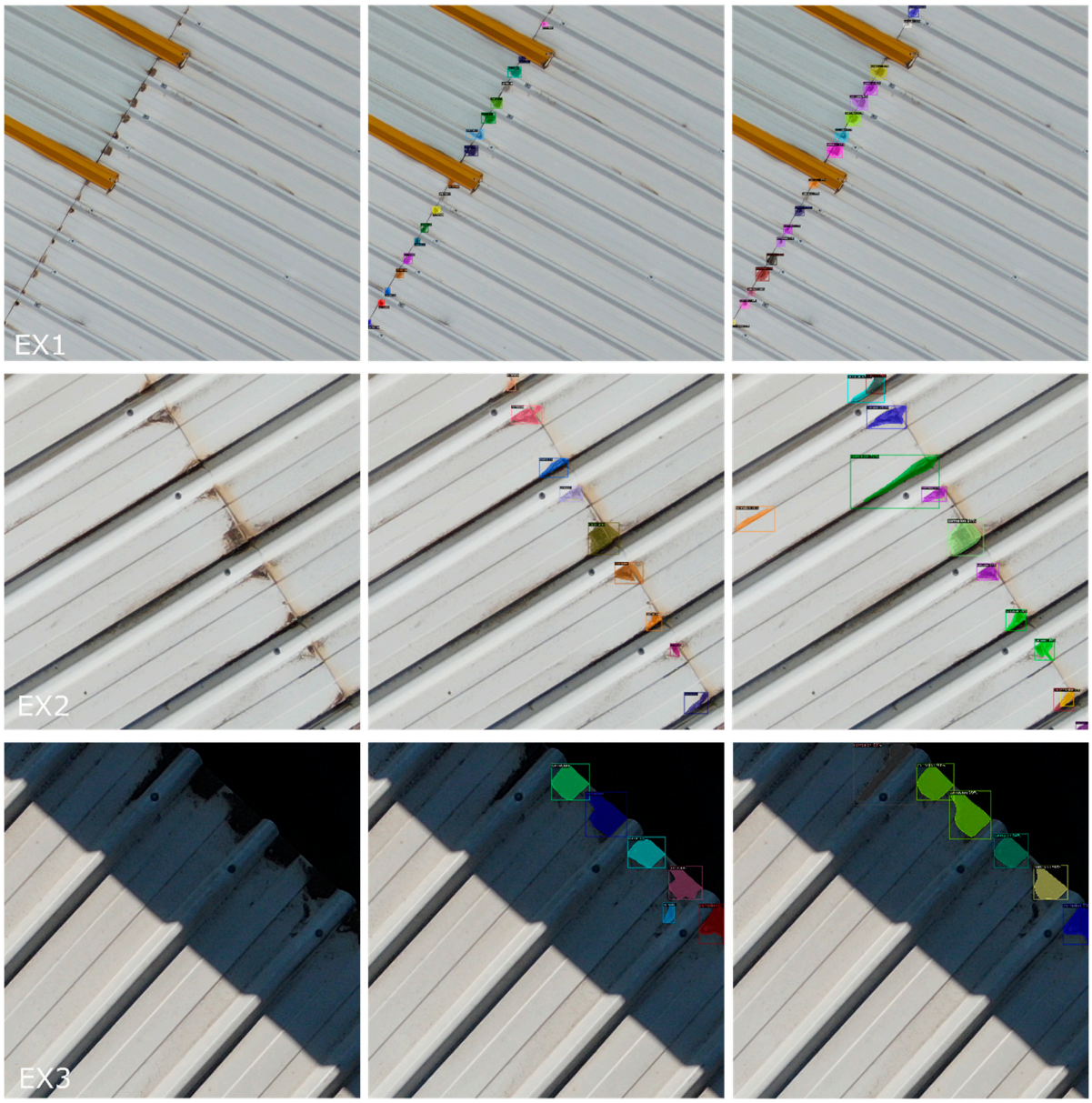

Figure 16. Examples of classification with small instances/poor illumination conditions (EX1 to EX3): (**a**) original image, (**b**) ground truth, and (**c**) prediction.

Figure 17. Examples of classification with complex backgrounds (EX1 to EX3): (**a**) original image, (**b**) ground truth, and (**c**) prediction.

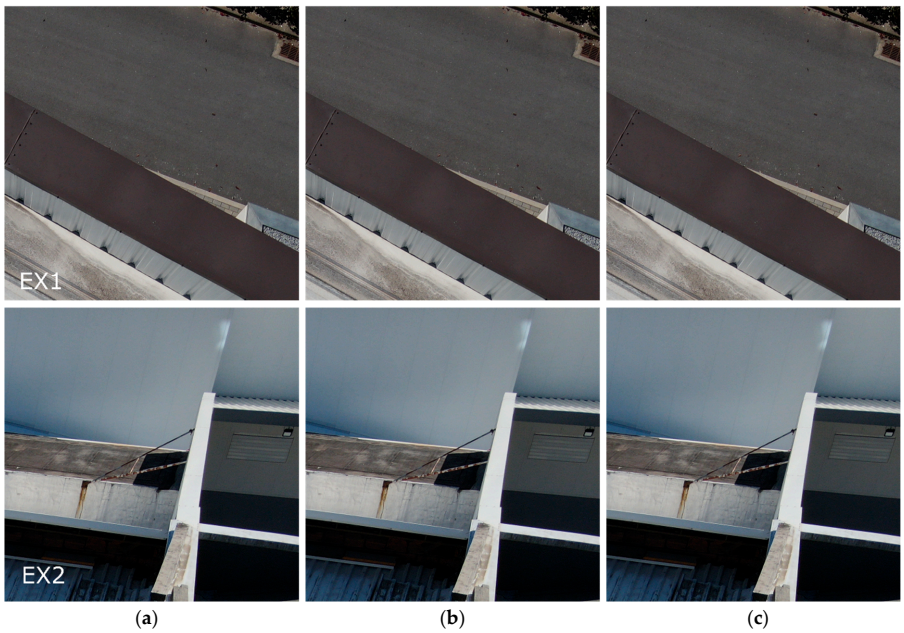

Figure 18. Examples of no classification in the presence of corrosion outside the area of the roofing system (EX1 and EX2): (**a**) original image, (**b**) ground truth, and (**c**) prediction.

6. Conclusions

This article proposes a methodology to automatically detect corrosion in the roofing systems of large-scale industrial buildings. First, the procedure relies on setting up an image database composed of more than 8k high-resolution images with the support of a UAV vision system. Second, the procedure entails the application of advanced image processing techniques based on the Mask R-CNN deep learning framework. The UAV used was the DJI MAVIC Enterprise Advanced, equipped with an RTK system that can provide the estimated position in real-time and, therefore, the ability to register high-accuracy georeferenced images. Finally, the images dataset, containing about 18k instances of corrosion, was annotated with the VIA software and processed in a JSON file compatible with the AI framework.

The training of the Mask R-CNN model involved tuning some hyperparameters from the advanced library made available by the Facebook AI research team, known as Detectron2. The adjusted hyperparameters were the size of the input images, the data augmentation strategy, the value of the RoI IoU Head hyperparameter, and the backbone network. The results are consistent for the training, validation, and test datasets. In terms of metrics, it is highlighted the average precision for detection and segmentation, considering an IoU of 50%, achieved values of 65.1% and 59.2%, respectively. Furthermore, the precision and recall computed reached 85.8% and 84.0% for all instances identified in the labeling process. Visually, the inferences show that the model can be trusted, identifying the anomalies even in the most complex backgrounds and lighting conditions. Indeed, the results of this research suggest a reliable and effective method for detecting corrosion on sandwich metallic panels, allowing for a long-distance, non-contact, low-cost, and automated inspection, culminating in cost savings within the facility management strategies of large-scale industrial buildings.

As future improvements, the authors are developing an application to integrate a new type of anomaly in the instance segmentation model, such as mechanical damages and water puddle accumulation. Furthermore, a semi-supervised technique to be applied in the already-made database is also being studied, which will support the automatic annotation of the corrosion instances in similar contexts. Finally, the integration of the georeferenced anomalies derived from the AI model within 3D photogrammetric reconstructions of the roofing systems are also planned, as well as the real-time inference of the images with embedded UAV hardware (e.g., NVIDIA JETSON Orin).

Author Contributions: Conceptualization, R.L., R.C., D.R., R.S., V.A. and A.D.; methodology, R.L., R.C., D.R. and R.S.; software, R.L., R.C. and R.S.; validation, R.L., R.C. and R.S.; investigation, R.L., R.C., D.R. and R.S.; resources, R.L., R.C., D.R. and R.S.; writing—original draft preparation, R.L.; writing—review and editing, R.C., D.R., V.A. and A.D.; visualization, R.L. and R.C.; supervision, D.R., R.S., V.A. and A.D.; project administration, D.R.; funding acquisition, D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by: Base Funding—UIDB/04708/2020 and Programmatic Funding—UIDP/04708/2020 of the CONSTRUCT—Instituto de I&D em Estruturas e Construções funded by national funds through the FCT/MCTES (PIDDAC), as well as the R&D project INSPECDRONE—Identification of anomalies in the external envelope of industrial buildings based on AI techniques and supported by UAVs, financed by the company Multiprojectus/Garcia Garcia. Additionally, the author Rafael Lemos acknowledges the Universidade Federal de Ouro Preto (UFOP) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the second author, Rafael Cabral, the doctoral grant UI/BD/150970/2021—Portuguese Science Foundation, FCT/MCTES.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is unavailable due to privacy restrictions.

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
2. Yağ, İ.; Altan, A. Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments. *Biology* **2022**, *11*, 1732. [CrossRef]
3. *Work-Related Fatal Injuries in Great Britain*; HSE: London, UK, 2022; p. 31. Available online: https://www.hse.gov.uk/statistics/pdf/fatalinjuries.pdf (accessed on 11 January 2023).
4. Rey-Merchán, M.D.C.; Gómez-de-Gabriel, J.M.; López-Arquillos, A.; Choi, S.D. Analysis of Falls from Height Variables in Occupational Accidents. *Int. J. Environ. Res. Public. Health* **2021**, *18*, 13417. [CrossRef] [PubMed]
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *60*, 1097–1105. [CrossRef]
6. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR* **2009**, *8*, 248–255. [CrossRef]
7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef]
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. Available online: https://arxiv.org/abs/1506.01497 (accessed on 17 May 2021). [CrossRef]
9. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
10. Pan, Y.; Zhang, L. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Autom. Constr.* **2021**, *122*, 103517. [CrossRef]
11. Shen, J.; Xiong, X.; Li, Y.; He, W.; Li, P.; Zheng, X. Detecting safety helmet wearing on construction sites with bounding-box regression and deep transfer learning. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 180–196. [CrossRef]
12. Li, Y.; Lu, Y.; Chen, J. A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector. *Autom. Constr.* **2021**, *124*, 103602. [CrossRef]
13. Arashpour, M.; Ngo, T.; Li, H. Scene understanding in construction and buildings using image processing methods: A comprehensive review and a case study. *J. Build. Eng.* **2021**, *33*, 101672. [CrossRef]
14. Kardovskyi, Y.; Moon, S. Artificial intelligence quality inspection of steel bars installation by integrating mask R-CNN and stereo vision. *Autom. Constr.* **2021**, *130*, 103850. [CrossRef]
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. Available online: http://arxiv.org/abs/1804.02767 (accessed on 31 August 2022).
16. Wei, W.; Lu, Y.; Zhong, T.; Li, P.; Liu, B. Integrated vision-based automated progress monitoring of indoor construction using mask region-based convolutional neural networks and BIM. *Autom. Constr.* **2022**, *140*, 104327. [CrossRef]
17. Lu, W.; Chen, J.; Xue, F. Using computer vision to recognize composition of construction waste mixtures: A semantic segmentation approach. *Resour. Conserv. Recycl.* **2022**, *178*, 106022. [CrossRef]
18. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. Available online: http://arxiv.org/abs/1706.05587 (accessed on 31 August 2022).
19. Chen, J.; Wang, G.; Luo, L.; Gong, W.; Cheng, Z. Building Area Estimation in Drone Aerial Images Based on Mask R-CNN. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 891–894. [CrossRef]
20. Karaaslan, E.; Bagci, U.; Catbas, F.N. Attention-guided analysis of infrastructure damage with semi-supervised deep learning. *Autom. Constr.* **2021**, *125*, 103634. [CrossRef]
21. Santos, R.; Ribeiro, D.; Lopes, P.; Cabral, R.; Calçada, R. Detection of exposed steel rebars based on deep-learning techniques and unmanned aerial vehicles. *Autom. Constr.* **2022**, *139*, 104324. [CrossRef]
22. Zhan, Y.; Liu, W.; Maruyama, Y. Damaged Building Extraction Using Modified Mask R-CNN Model Using Post-Event Aerial Images of the 2016 Kumamoto Earthquake. *Remote Sens.* **2022**, *14*, 1002. [CrossRef]
23. Hou, F.; Lei, W.; Li, S.; Xi, J.; Xu, M.; Luo, J. Improved Mask R-CNN with distance guided intersection over union for GPR signature detection and segmentation. *Autom. Constr.* **2021**, *121*, 103414. [CrossRef]
24. Jin Lim, H.; Hwang, S.; Kim, H.; Sohn, H. Steel bridge corrosion inspection with combined vision and thermographic images. *Struct. Health Monit.* **2021**, *20*, 3424–3435. [CrossRef]
25. Munawar, H.S.; Ullah, F.; Shahzad, D.; Heravi, A.; Qayyum, S.; Akram, J. Civil Infrastructure Damage and Corrosion Detection: An Application of Machine Learning. *Buildings* **2022**, *12*, 156. [CrossRef]
26. Han, Q.; Zhao, N.; Xu, J. Recognition and location of steel structure surface corrosion based on unmanned aerial vehicle images. *J. Civ. Struct. Health Monit.* **2021**, *11*, 1375–1392. [CrossRef]
27. Sezer, A.; Altan, A. Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. *Solder. Surf. Mt. Technol.* **2021**, *33*, 291–298. [CrossRef]
28. Forkan, A.R.M.; Kang, Y.-B.; Jayaraman, P.P.; Liao, K.; Kaul, R.; Morgan, G.; Ranjan, R.; Sinha, S. CorrDetector: A Framework for Structural Corrosion Detection from Drone Images using Ensemble Deep Learning. *arXiv* **2021**, arXiv:2102.04686. Available online: http://arxiv.org/abs/2102.04686 (accessed on 25 May 2021). [CrossRef]

29. Albanie, S.; Varol, G.; Momeni, L.; Afouras, T.; Brown, A.; Zhang, C.; Coto, E.; Camgoz, N.C.; Saunders, B.; Dutta, A.; et al. Signer Diarisation in the Wild. 2021. Available online: https://www.robots.ox.ac.uk/~{}vgg/research/signer_diarisation/ (accessed on 11 January 2023).
30. Hiroto, H. Digging into Detectron 2—Part 1. Available online: https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd (accessed on 4 November 2022).
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385. Available online: http://arxiv.org/abs/1512.03385 (accessed on 13 July 2022).
32. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083. Available online: http://arxiv.org/abs/1504.08083 (accessed on 15 May 2021).
33. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
34. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
35. Poole, B.; Sohl-Dickstein, J.; Ganguli, S. Analyzing noise in autoencoders and deep networks. *arXiv* **2014**, arXiv:1406.1831. Available online: http://arxiv.org/abs/1406.1831 (accessed on 14 September 2022).
36. Clark, A. Pillow (PIL Fork) Documentation. 2015. Available online: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf (accessed on 11 January 2023).
37. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
38. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2015**, arXiv:1405.0312. Available online: http://arxiv.org/abs/1405.0312 (accessed on 20 June 2022).
39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980. Available online: http://arxiv.org/abs/1412.6980 (accessed on 8 January 2023).