

Article

A Deep Learning Image Corrosion Classification Method for Marine Vessels Using an Eigen Tree Hierarchy Module

Georgios Chliveros ^{1,*}, Iason Tzanetatos ² and Stylianos V. Kontomaris ¹

¹ Department of Engineering, Metropolitan College, Marousi Campus, 15125 Athens, Greece; skontomaris@mitropolitiko.edu.gr

² Core Innovation Center, Core Innovation and Technology OE, 17343 Athens, Greece

* Correspondence: gchliveros@mitropolitiko.edu.gr

Abstract: This paper involves the automation of a visual characterisation technique for corrosion in marine vessels, as it appears in the hull preventive coatings of marine vessels and their surfaces. We propose a module that maximizes the utilisation of features learned by a deep convolutional neural network to identify areas of corrosion and segment pixels in regions of inspection interest for corrosion detection. Our segmentation module is based on Eigen tree decomposition and information-based decision criteria in order to produce specific corroded spots—regions of interest. To assess performance and compare it with our method, we utilize several state-of-the-art deep learning architectures. The results indicate that our method achieves higher accuracy and precision while maintaining the significance score across the entire dataset. To the best of our knowledge, this is the first Eigen tree-based module in the literature in the context of trained neural network predictors for classifying corrosion in marine vessel images.

Keywords: marine corrosion; corrosion detection; preventive monitoring; image segmentation; deep learning



Citation: Chliveros, G.; Tzanetatos, I.; Kontomaris, S.V. A Deep Learning Image Corrosion Classification Method for Marine Vessels Using an Eigen Tree Hierarchy Module. *Coatings* **2024**, *14*, 768. <https://doi.org/10.3390/coatings14060768>

Academic Editors: Jordan Todorov Maximov and Galya Velikova Duncheva

Received: 30 May 2024

Revised: 13 June 2024

Accepted: 14 June 2024

Published: 18 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The problem of marine corrosion can be defined as the deterioration of materials, particularly metals, due to electrochemical reactions with the marine environment, which includes seawater, salt spray, and marine organisms. The high salinity and presence of various dissolved ions in seawater make for a highly aggressive and corrosive environment. Marine corrosion is a significant concern for vessels, offshore structures, and other marine equipment as it can lead to structural failure, safety hazards, and increased maintenance costs. In the context of vessels, the primary expenses associated with corrosion are primarily indirect in nature. These costs encompass increased mass, heightened workload during the design and construction, decreased performance, and the expenditures associated with repairs. Regardless, improved design practises should incorporate protective coatings that can in principle save up to one-fifth of the maintenance costs [1,2]. However, corrosion is a function of many stochastic variables, and thus, probabilistic models are only appropriate for reliability predictions (i.e., predicting structural strength deterioration) [3,4]. These probabilistic models make for a further issue, that of fatigue damage prediction (crack growth), whereas incorporating uncertainty by means of probability density functions [5,6].

Protective coatings are commonly used in marine vessels, limiting the need for regular inspections. However, under the harsh marine conditions that vessels operate, coatings may need reapplication every 5 years. Furthermore, the regulatory framework necessitates complete inspection of vessel hulls at least three times in five-year periods, with intermediate surveys being performed within 24 months. This relates to detecting subtle signs of coating failure, such as coating discolouration due to chemical changes and the presence of moisture bubbles in the coating, before severe corrosion occurs. Provided with the SOLAS

regulatory framework [7], human inspectors regularly monitor corrosion on marine vessels' surfaces in order to maintain safety, operational efficiency, environmental protection, and compliance with regulations. Early detection by inspectors enables proactive maintenance, since it identifies corrosion hotspots and fosters continuous improvement in corrosion management. Inasmuch, inspectors pinpoint areas of accelerated deterioration, allowing for further targeted inspections and suggest preventive measures. This helps optimize fleet maintenance efforts and ensures that critical components receive adequate attention and timely intervention.

Visual inspection by a human operator/inspector is employed as a direct way of pinpointing the aforementioned visible signs of corrosion areas (e.g., rust, pitting, discolouration). This involves the use of magnifying tools, flashlights, and mirrors for hard-to-reach areas. Visual inspection is often combined with cleaning the surface to remove dirt and rust for a better view. However, dirt and corrosion are visibly separable in marine environments, in terms of their annotation in RGB colour intensity (R, G, B triplets) [8]. Dirt typically ranges from brown (165, 42, 42) to dark brown (92, 64, 51), depending on its source, as opposed to corrosion which presents itself in a colouring of red (255, 0, 0) to blue-green (102, 205, 170), depending on the type of metal. The result is typically the number of marked spots that have been pointed out as required to be checked. Subsequently, the operator uses Ultrasonic Testing sensors, whereas high-frequency sound waves are used to detect thickness loss due to corrosion [9]. In such an inspection, a probe is placed on the surface with a coupling gel in order for the sound waves to better penetrate the material, and subsequently, the primary reflection and each echo are measured. This leads to the inference of thickness readings which can be recorded and further analysed. The overall method is laborious, subjective, and requires from several hours to a few days to be completed, depending on the vessel's size and the number of spots required to be checked [3]. The automation of the visual aspect of such a laborious task would lead to massive savings both in terms of person hours and time required for a vessel to no longer be operational.

The automation of corrosion visual surveying (RGB images) by human inspectors can be seen as an extension of semantic segmentation in images, i.e., the association of pixels to a specific class/label (corrosion) [10–12]. Corrosion in the form of isolated area units on a vessel surface is difficult to detect and/or directly predict in an RGB image space using standard techniques (e.g., [13–15]). This is mainly due to the diverse geometrical shape, which makes it difficult to postulate prior knowledge on the basis of a generalised morphology. Recent advances in deep learning models [16,17] have examined the applicability and potential of machine vision for the inspection of large structures [18–20] and segmentation of corrosion in marine vessels [2,11,21–23].

In particular, the deep learning models of feed-forward neural networks (FFNNs) [24], fully convolutional neural networks (FCNs) [25], convolutional neural networks (CNNs) [15], and Bayesian neural networks (BNN) [11], as types of deep (supervised) learning architectures have been demonstrated to exhibit good performance in segmenting images of corroded versus non-corroded surface areas. These techniques seem to achieve good performance with respect to detection/classification, with relative precision and accuracy. However, these methods depend on the size of the training datasets, the structure of a data-driven approach, and the environmental variability factors (e.g., sunlight conditions) in the produced scenes. As a result, these deep-learning-trained models produce many false positives. In particular, such deep learning models seem to exhibit a decreased accuracy in recognition when an increase in crack-like texture and/or locally dispersed illumination appear due to crack depth in the training images [2,12,23]. This surface texture produces high 'colour' similarity between groups of pixels that results in reduced precision and accuracy [12,21,22]. A possible solution is to fine-tune the neural network during its training phase in order to enhance specificity, although this leads to decreased method generalisation. Finally, there is at present a limited availability of corrosion datasets with significant image numbers inclusive of ground truth (labelled images) and there is difficulty to produce semi-synthetic sets due to the morphology of corrosion.

In this paper, we propose a deep learning characterisation technique for corrosion in marine vessels, as it appears in the standard images of marine vessels' hulls treated with preventive coatings and their surfaces. We introduce a new perspective on corrosion detection using deep learning and segmentation by means of the incorporation of the Eigen tree decomposition module, adding a novel dimension to existing methodologies for corrosion classification. By presenting detailed results and comparisons with other methods, we establish the credibility and effectiveness of the proposed method. As such, we devise an Eigen tree decomposition module that can act upon pre-trained neural network models, and correctly segment identified areas of corrosion. We provide evidence of validating the methods by means of comparison with other methods used in camera-based corrosion detection from the literature. Examples that illustrate the performance of all methods over the dataset are also provided. An analysis of standard performance and significance metrics that have been used in the literature are reported in tabular and visual (boxplot) format over the entire dataset. We conclude that our convolutional neural network in conjunction with the Eigen module (referred to as YOLO-Eigen) performs better in terms of accuracy (by more than 10%) and precision (by more than 20%) over all other methods, whilst maintaining significance scores (in the margin of 0.01 to 0.06 absolute difference) against other methods used for comparison.

2. Methodology

The overall schema of our method is illustrated in Figure 1. The pre-trained multi-layer neural network is indicated by the light grey background in Figure 1. This trained model network is composed of the convolution layer (C_n), the pooling layer (P_n), and a full convolution/connection layer (F_n). The convolution layer contains of n filters, each of which contains a matrix of weights. These filters are convolved with the input image, and subsequently transformed with a non-linear activation function. A number of feature maps are produced in this way, albeit with redundant information. To reduce redundancy, a pooling layer (P_n) summarizes feature maps into smaller local subsets. The convolution and pooling layers are led to the fully connected layer (F_n) for categorisation, that is, the production of the bounding box of a predicted segmentation area. The produced bounded box areas are further refined by means of the Eigen tree decomposition (dark grey boxes in Figure 1), enriched by decision criteria for segmenting specific areas of interest (pixel segmentation).

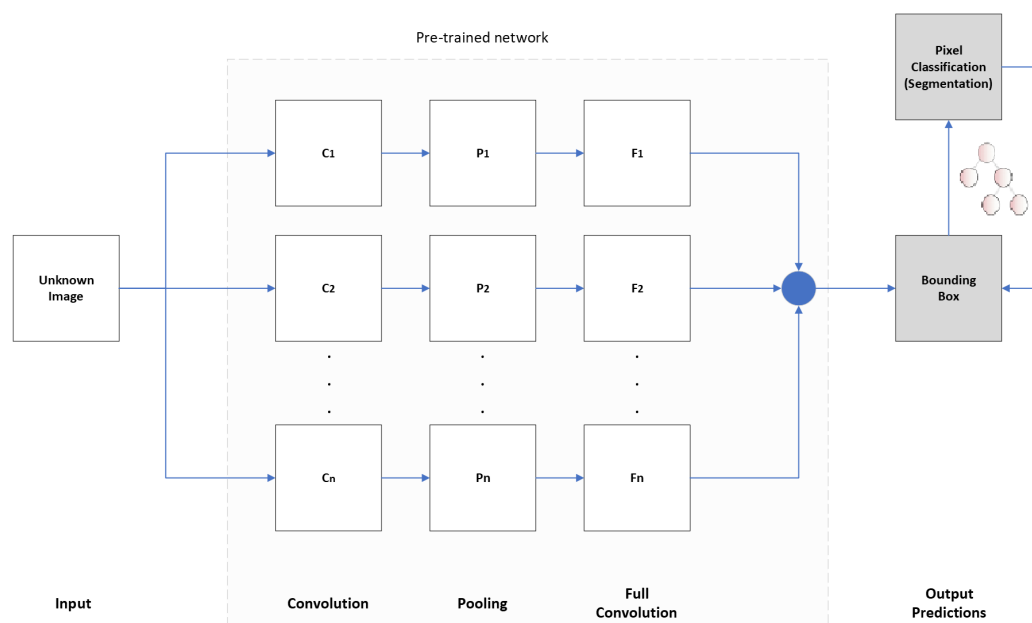


Figure 1. General methodology: pre-trained network receives a previously unseen image; the CNN devises bounding boxes, leading to refinement by a segmentation module.

We use a dataset devised from dry-dock conditions described in Section 2.1, both for training and testing input of a convolution neural network, namely YOLOv8, thus producing a new model for corrosion based on our dataset. This produced model is then used for output prediction. The implementation of the YOLOv8 trained model is described in Section 2.2. The output prediction (bounding boxes) are refined and the segmentation of pixels is produced by means of the Eigen decomposition module (Section 2.3).

2.1. Data

In order to explore performance characteristics of the selected methods, we use the [MaVeCoDD dataset](#) (Last Accessed: 12 June 2024), which has been utilised in similar other studies (e.g., [22,23,26,27]), and comprises of corrosion images for marine vessels in dry-dock and moored conditions. The dataset incorporates several artifacts under various lighting conditions as well contrast (e.g., changing lighting conditions, sky, sea) and background complexities (e.g., objects in front of the hull: maintenance ladders, rudders), with the camera viewpoint set at various angles and distances from corrosion areas of interest. The dataset contains the following: (a) high-resolution RGB images of 3799×2256 pixels (72 dpi, 24 bit depth); (b) low-resolution RGB images of 1920×1080 pixels (96 dpi, 24 bit depth); alongside (c) labeled (ground truth) images that have RGB triplet values for each pixel that corresponds to corrosion and zero values elsewhere. The labeled areas in the image are manually annotated by human inspectors, denoting regions of interest characterised by rust but may include areas that are suspected to produce surface rust in the near future.

However, for supervised training required by techniques such as neural nets, a larger dataset would be needed. As such, we split all images in the original dataset (either high or low resolution) into four equally sized images maintaining their original form, with the camera/image principle point being the reference for splitting into four new images, in order to preserve the spatial information and characteristics of the corrosion patterns. Subsequently, the new images are resized into bounded 512×512 pixels, resulting in a larger dataset of 980 images of pixel analysis 512×512 pixels of 72 dpi and a bit depth of 24 bit.

2.2. YOLOv8 Trained (Large) Model

As previously mentioned, we have used the [YOLOv8](#) (Last Accessed: 12 June 2024) architecture since it has been known to predict fewer boxes and has a faster non-maximum suppression process. The architecture offers five different scaled versions, known as nano (YOLOv8n), small (YOLOv8s), medium YOLOv8m, large (YOLOv8l), and extra-large (YOLOv8x). We have experimented in training all versions with our dataset under a 60:40 split (training to testing) ratio, for 600 epochs of batch size 1, with early stopping set at 10 epochs. Under these conditions, the YOLOv8l variant produced the best trained models and achieved increased corrosion recognition accuracy, under mosaic augmentation (not enabled in the last ten epochs to avoid bias).

A single-stage object detection model was employed, which performs object localisation (position of objects in image frame) and classification within the same network (production of bounding boxes). The trained backbone architecture includes some confidence of the corrosion feature maps, an aggregation operator (merge corrosion maps), and a head to produce the final predictions in the form of bounding boxes assigned to each box. In effect, the trained model divides each image into grids, and for each grid (equal size) predicts a number of bounding boxes alongside the confidence. The confidence reflects the accuracy of the bounding box, which contains an object regardless of class. The classification score for each box and for every class in training can be combined to produce the probability of each class being present in a predicted box.

An example output of the trained YOLOv8l model can be observed in Figure 2c, alongside the raw image (Figure 2a) and ground truth (Figure 2b). This example visually illustrates the performance of the trained model; that is, the image artefacts (e.g., sea level,

block tire) are correctly ignored, alongside the reported confidence for each bounding box. Furthermore, by means of comparison to the ground truth, the produced bounding boxes only include areas of corrosion. Specific performance metrics will be examined in latter sections, albeit we have used the large trained model (YOLOv8l).

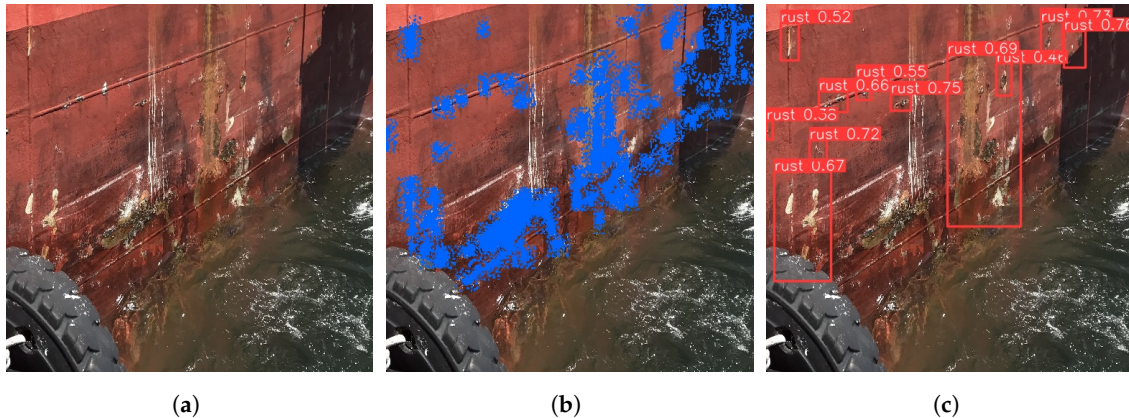


Figure 2. Example image model result—produced YOLOv8l bounding boxes. (a) Raw Image. (b) Ground Truth—Label. (c) YOLOv8l Bounding Boxes.

2.3. Proposed Eigen Module (YOLO-Eigen)

Our Eigen module acts upon the produced bounding boxes (as per example of Figure 2) of the YOLOv8l trained model (Section 2.2) in a binary tree structure. The Eigen tree decomposition selects a candidate node and, under certain conditions, produces a split in the leaf nodes. In our case, this manifests as a binary split of two leaf nodes, whereas two quantisation levels (Q_{2n} , Q_{2n+1}) are estimated and each member of a cluster is associated to that of the closest quantisation level [28]. It is assumed that the mean intensity value of each colour channel is the histogram point with the least variance in the Eigen space, leading to a specific quantisation level value Q_n . The quantisation level value of each colour channel, and for each node, is defined as $Q_n = M_n/N_n$, where M_n is the number of pixels belonging to some cluster, and N_n is the number of clusters.

The Eigen module is based upon a representation of decision trees, which performs binary splits so that an appropriate hyperplane is selected; that is, hierarchically separates data into clusters in a sequence, and until a predefined number of such clusters k has been reached. Inasmuch, current iteration clusters are not re-evaluated in their totality but only in the sequence of a current branch. For a binary split decision tree, this means that the average of square distances of all data points (image pixel quantisation levels Q_n) sequentially generate leaf nodes of new clusters C_{2n}, C_{2n+1} . As it was presented in [22], the solution for \mathbf{w} parameters for plane h can be found by optimising

$$h(\mathbf{w}) = \frac{\mathbf{w}_n^T \mathbf{R}_{2n+1} \mathbf{w}_n}{\mathbf{w}_n^T \mathbf{R}_{2n} \mathbf{w}_n} \quad (1)$$

so that

$$\begin{aligned} \mathbf{w}_{2n} &= \arg \max_{w \neq 0} h(\mathbf{w}) \\ \mathbf{w}_{2n+1} &= \arg \min_{w \neq 0} h(\mathbf{w}) \end{aligned} \quad (2)$$

Thence, the optimisation problem of Equation (2) can be solved for Eigen values λ_n in the generalised framework of [29], whereas $\mathbf{R}_{2n} \mathbf{w}_{2n} = \lambda_n \mathbf{R}_{2n+1} \mathbf{w}_{2n+1}$. The binary split is produced by inferring the optimal values for the hyperplane of a current cluster, i.e., finding the min and max eigenvalues (λ) of the generalised Eigenvalue problem [30]. As a result, the chroma (RGB) binary quantisation decision [28] is taken at each parent node based on $\max\{\lambda_n\}$ (right-hand node) and $\min\{\lambda_n\}$ (left-hand node), representing a

combined solution to the aforementioned hyperplanes for our Eigen decomposition binary tree module.

In effect, a node split is determined by its eigenvector being that of the largest Eigenvalue over all previous nodes, which in turn determines the pixel indices in cluster C_n that will be assigned into the new clusters C_{2n}, C_{2n+1} . The binary split for node image indices ℓ association over cluster $C_n \rightarrow \{C_{2n}, C_{2n+1}\}$ is performed using the following schema:

$$\begin{aligned} C_{2n} &= \{\ell \in C_n : e_n^T x_\ell \geq e_n^T Q_n\} \\ C_{2n+1} &= \{\ell \in C_n : e_n^T x_\ell < e_n^T Q_n\} \end{aligned} \quad (3)$$

In order to identify the leaf node that best captures the corroded regions found within the input frame, the YOLO-Eigen module also incorporates the standard information entropy measure H_n . As such, when the algorithmic decision is to be taken as to which node best represents corrosion, it iterates over the leaf nodes C_n, C_{n-k} , identifying the nodes that correspond to the maximum Eigenvalue λ_n and maximum entropy H_n . In the event that $\max\{\lambda_n\}, \max\{H_n\}$ point to different leaf nodes, the node of $\max\{H_n\}$ is eliminated from the candidate pool. Conversely, if both max values refer to the same node, we assume that the selected node converges to the maximum of all entropy values. Since the predicted leaf node contains only pixel indices, the prediction methodology preserves and reconstructs the predicted frame based on said cluster indices.

An example of the tree generation can be inspected in Figure 3, where the 'red' branches lead to what we mention later on as 'pruned' nodes. It should be evident that the top (root) image is split into corresponding nodes: N1, N2. Continuing along the path of node N2, which has a larger Eigenvalue than N1, it is further split into nodes N2.1 and N2.2. The new node splits will be performed at node N2.2 due to it having the largest Eigenvalue, whilst N2.1 could produce nodes of lower Eigen values than those of N2.2, and would be pruned. In this example, the tree continues forming in such a way, until a tree depth $k = 7$ of end-leaf nodes is reached. Note that for the right-hand-side (RHS) direction of the tree, the bottom-most nodes emanating from the N2.1 parent node have entropies of 7.19 and 7.26, with a parent node of 7.24. This means that the leaf nodes do not hold more information than their parent and are pruned. In the final step for RHS leaves, the remainder leaf nodes for the N2.2 parent remain (i.e., they are not pruned): they have an entropy value of 7.42, which is higher than that of N2.1, whose value is 7.24, is considered to hold more information, and is thus preserved. Since there are no further steps at RHS, the process moves to LHS accordingly.

In line with this specific example (Figure 3), the selected 'best' node is framed in green colour, with the ground truth being framed in 'blue' colour. Notice that the selected node is that of light-green colouring and has less coverage than the ideal annotated ground truth (blue node). The pruned tree leads to a decision criterion which identifies a node (green) that coincides with the ground truth (blue). This is a bottom-up approach, whereas from all nodes below, N1.2 and N2.1 are pruned (eliminated from the candidate pool), and the selected dominant (green) node has both the largest information entropy and the largest Eigenvalue. Notice that the parent node N2.2 has the higher Eigenvalue but lower entropy, hence why it is not selected as dominant by the decision criteria.

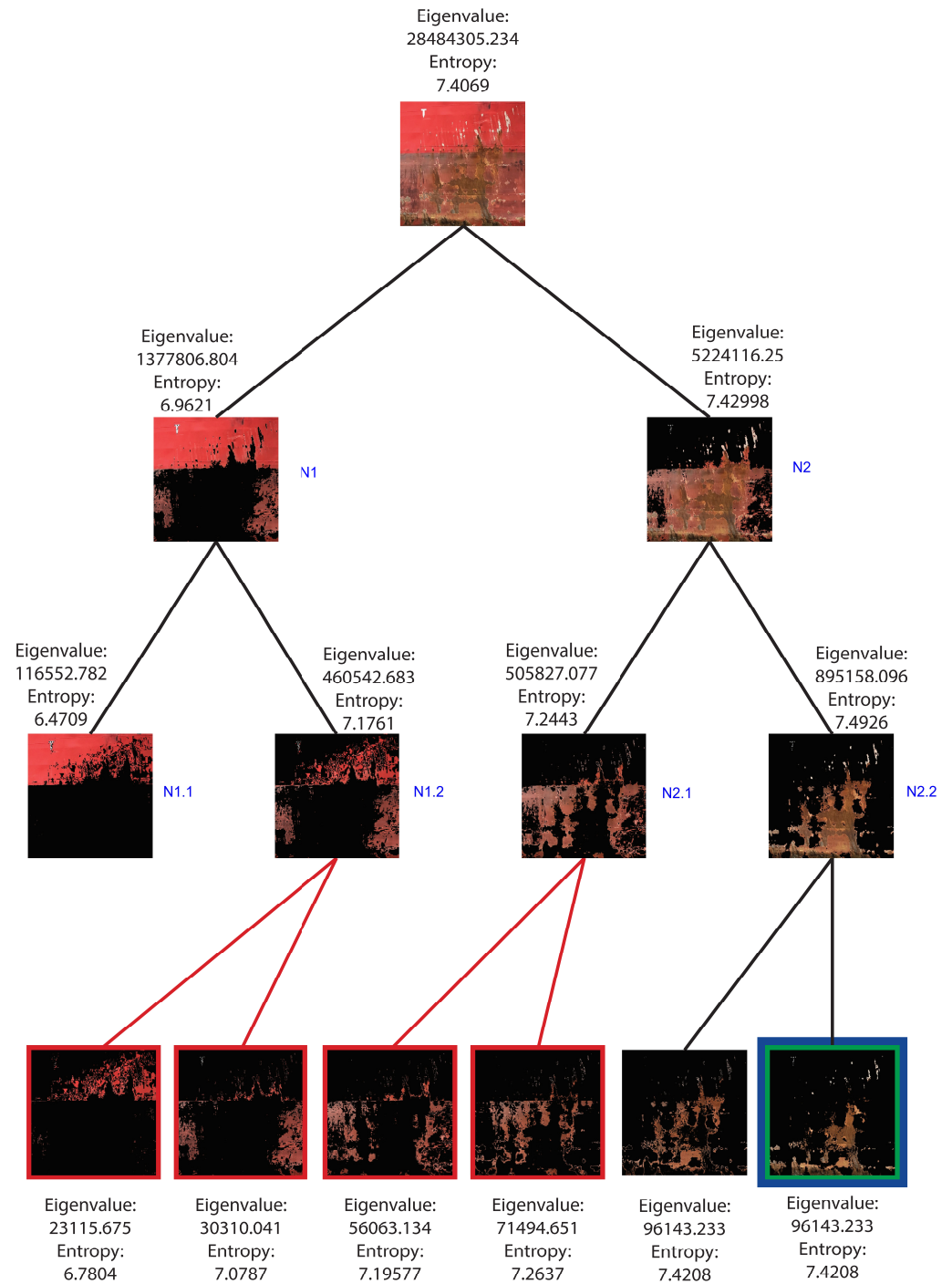


Figure 3. Example Eigen tree decomposition and pruning; tree depth $k = 7$, with all denoted eigenvalues and entropy per node. Blue frame denotes ground truth; green frame denotes the dominant node decision; red lines denote pruned/eliminated branches.

We conclude by expanding upon the example in Figure 2. The use of our Eigen module refines the trained YOLOv8's produced bounding boxes, and the subsequent process by means of the decision criteria to the segmentation result is shown in Figure 4. In this particular example, it should be obvious that the refinement of YOLO bounding boxes at Eigen tree $k = 5$ produces better coverage than for $k = 7$ at the expense of producing more false negatives (FNs). For $k = 7$, there is a smaller but more specific segmentation

coverage (reduced FN) at the expense of a simultaneous reduction in true positives (TPs). This example serves as the starting point for our further investigations in Section 3.

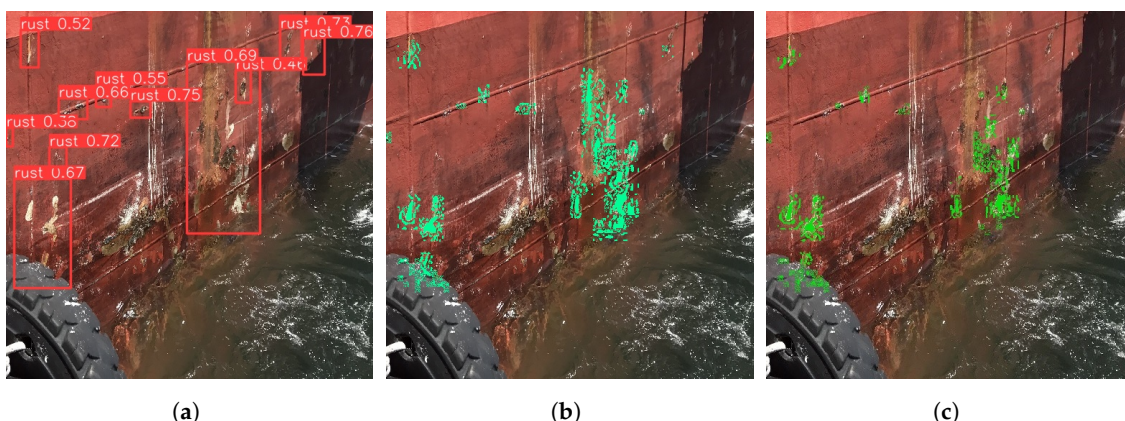


Figure 4. Example of the Eigen module (tree decomposition/prediction) on YOLO-produced bounding boxes from Figure 2. (a) YOLOv8l bounding boxes. (b) YOLO-Eigen at $k = 5$. (c) YOLO-Eigen at $k = 7$.

3. Research Findings

We compare our method with standard deep learning image segmentation methods as provided in freely available implementation, and we train new models from our dataset. All methods used are trained upon the dataset as described in Section 2.1. In the case of a Bayesian neural network (BNN), we use [SpotRust](#) [11], which relies on a base network derived from [HRNetV2](#) [31]. For a convolutional neural network (CNN), we use the optimised implementation [UNet](#) [32] through Sensitive Residual Network blocks (SEResNet). As previously mentioned, our hybrid method uses a CNN implementation of [YOLOv8](#) [33] for bounding the areas of corrosion interest, and subsequently segments these areas using the Eigen decision tree hierarchies.

The UNet implementation was trained and tested under a 60:40 split of the dataset using the SEResNet, which incorporates Squeeze-and-Excitation (SE) blocks that enhance the network's ability to capture and utilize information. In principle, this is expected to lead to improved performance in image segmentation, distinguishing between true positive/negative and false positive/negative pixel areas depending on the number of SE blocks used. In our case, we have reported results from the use of 18 and 34 SE blocks. Furthermore, we utilised a sigmoid activation function and the Adam optimizer, with binary cross-entropy loss function enabled for training the model. The learning rate was set at 10×10^{-3} for 50 epochs.

The BNN implementation (SpotRust) was trained and tested under a 60:40 training to testing split, using the variational inference (VI) as well as Monte Carlo dropout (DO) methods. The drop out rate was set at 0.4, a learning rate at 3×10^{-3} , and max epochs at 550. However, it should be noted that despite model fine-tuning to our dataset, the produced Gaussian noise in the dataset by the models increased aleatoric uncertainty. Both variational (VO) and Monte Carlo (MC) models produced outputs of reduced performance.

The YOLO-SAM implementation consists of using our trained YOLOv8l backbone (Section 2.2) and enforce segmentation by means of the Segment Anything (SAM) methodology [34]. The SAM model was configured to run inference via CPU utilisation and for each extracted bounding box that the trained YOLO model will produce on a single image. We then prompted the input of the SAM model by identifying the corresponding coordinates of the bounding box within the label image. However, SAM is a computationally expensive model and there were some images where the model was unable to produce predictions due to RAM constraints. Thus, in the event of RAM depletion, predictions of classified images were set as empty images, i.e., no regions of interest (ROIs) found. This can most likely be attributed to the complexity of the scenery found within the input ROIs and

their corresponding labels. To minimize the number of affected images, morphological closing on the prompt labels has been applied, with various kernel sizes. This operation was performed until either the model produced an output, or the kernel size reached the same size as the ROIs in question.

3.1. Performance Metrics

To assess the performance of corrosion detection, the pixel coordinates of corrosion in an image under investigation are found by comparison to the ground-truth-annotated images. This is performed by applying the labelled image mask on the input raw image, thence generating the dominant cluster result. The pixels defined as ‘True Positive’ pixels (TP) of a cluster are those that match the labeled mask, and ‘False Positive’ pixels (FP) are those that do not fall within the labeled mask. Inasmuch, TP explores the number of pixels in the image that are correctly identified as being corrosion; TN presents the number of pixels in the image that are correctly identified as not being corrosion; FP indicates the number of pixels in the image that are incorrectly identified as corrosion; and FN indicates the number of pixels in the image that should be identified as corrosion but are not. Under the aforementioned pixels’ definitions, we employ specific metrics that measure sensitivity, specificity, precision, accuracy, and significance (i.e., the *f*-score).

Sensitivity is defined as the measure of the number of pixels that were correctly predicted, with respect to those incorrectly predicted as negative pixels. The expression for the described metric is:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (4)$$

Specificity is defined as the measure of the number of pixels that were incorrectly predicted, with respect to the correctly predicted as negative pixels. The expression is

$$\text{Specificity} = \frac{FP}{FP + TN} \quad (5)$$

Precision is defined as the measure of the number of pixels that were predicted correctly, out of the total of positively predicted pixels. The expression is

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

Accuracy is defined as a measure of the number of pixels predicted correctly (either positive or negative) out of all possible predictions. The expression is

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

The *f*-score is a metric that relates the calculated sensitivity to precision instances. It is a special case of the harmonic mean of the measures, as expressed in Equation (4), by using that of Equation (6). It measures the relative significance of the results in terms of a method’s precision or accuracy given its sensitivity. The expression is

$$\text{f-score} = \frac{2 \times \text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}} \quad (8)$$

The mean average precision (*mAP*) is a popular metric in measuring the accuracy of object detectors. In general, the average precision (AP) is defined by (approximately) computing the area defined on precision versus sensitivity. As such, if we assume pairs of corresponding values, precision (p_i) and sensitivity (s_i), where there is some underlying function such that $p_i = g(s_i)$, which can be discretised over i , then the *mAP* would be that of all APs averaged for all classes of objects.

These performance metrics can be thought of as constituting a standard (in the machine learning literature) confusion matrix of ‘observability’ versus ‘predictability’. That is

to say, the metric relating to how sensitive the predicted pixel segmentation is to false negatives (FNs), upon observing false positives (FPs), on a scale of 0 to 1. This leads to the corresponding precision-based f-score, and hence the precision significance of the segmentation result. Similar argumentation applies for the significance of accuracy based on the mean average precision estimate mAP .

3.2. Segmentation Performance

The segmentation performance of all methods can be investigated by means of the average behaviour of pixels, either correctly ('true') or incorrectly ('false') classified as corrosion, with example instances provided in Figure 5. The overall behaviour of correctly classified pixels (TP and TN) and incorrectly classified pixels (FP and FN) are presented by means of boxplots in Figures 6 and 7, respectively. In general, an ideal scenario would be that the output prediction matches the labelled (ground truth) image, as illustrated by example outputs. Consequently, the boxplot presents itself with a spread of one standard deviation length from the average value of the method employed, and the average value is close to the total average of the entire dataset for true positive or negative predictions, and close to zero for false positive or negative predictions.

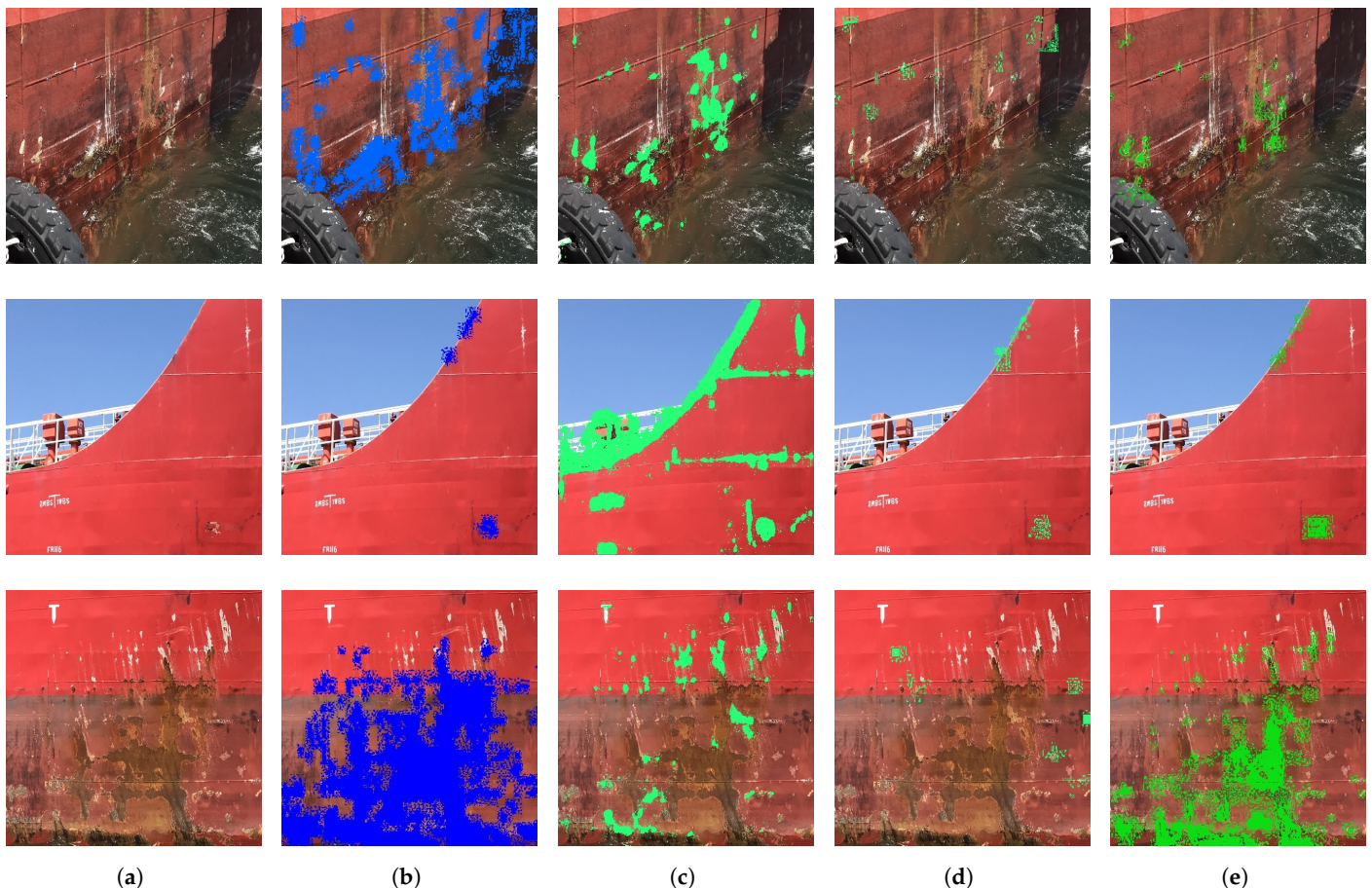


Figure 5. Example image validation of methods reported in Table 1. (a) Raw Image. (b) Ground Truth. (c) UNet SE34. (d) YOLO-SAM. (e) YOLO-Eigen.

In Figure 5, three characteristic cases in the dataset are depicted, that is to say, the raw image alongside the ground truth denoted as corrosion (blue pixels), as well as the results from different methods. In these specific cases, it can be evident from Figure 5c that the UNet model produces an increased number of false positives (FPs), especially in the case of background segments (e.g., sea water, skyline, variable light and shadows). The YOLO-SAM approach in Figure 5d appears to lack sufficient coverage of corrosion patterns, thus making it more likely to not produce false positives (FPs) or negatives (FNs).

Conversely, our YOLO-Eigen approach in Figure 5e seems to provide adequate coverage and exhibits high accuracy, particularly when visually compared to the ground truth, even for specific artifacts present in the background. However, the visual inspection of the examples in Figure 5 does not provide the full case and the population performance over the whole dataset. As a result, we have produced boxplots of the corrosion identification results in terms of true pixel associations, with respect to the ground truth, in Figure 6, and for false pixel associations in Figure 7.

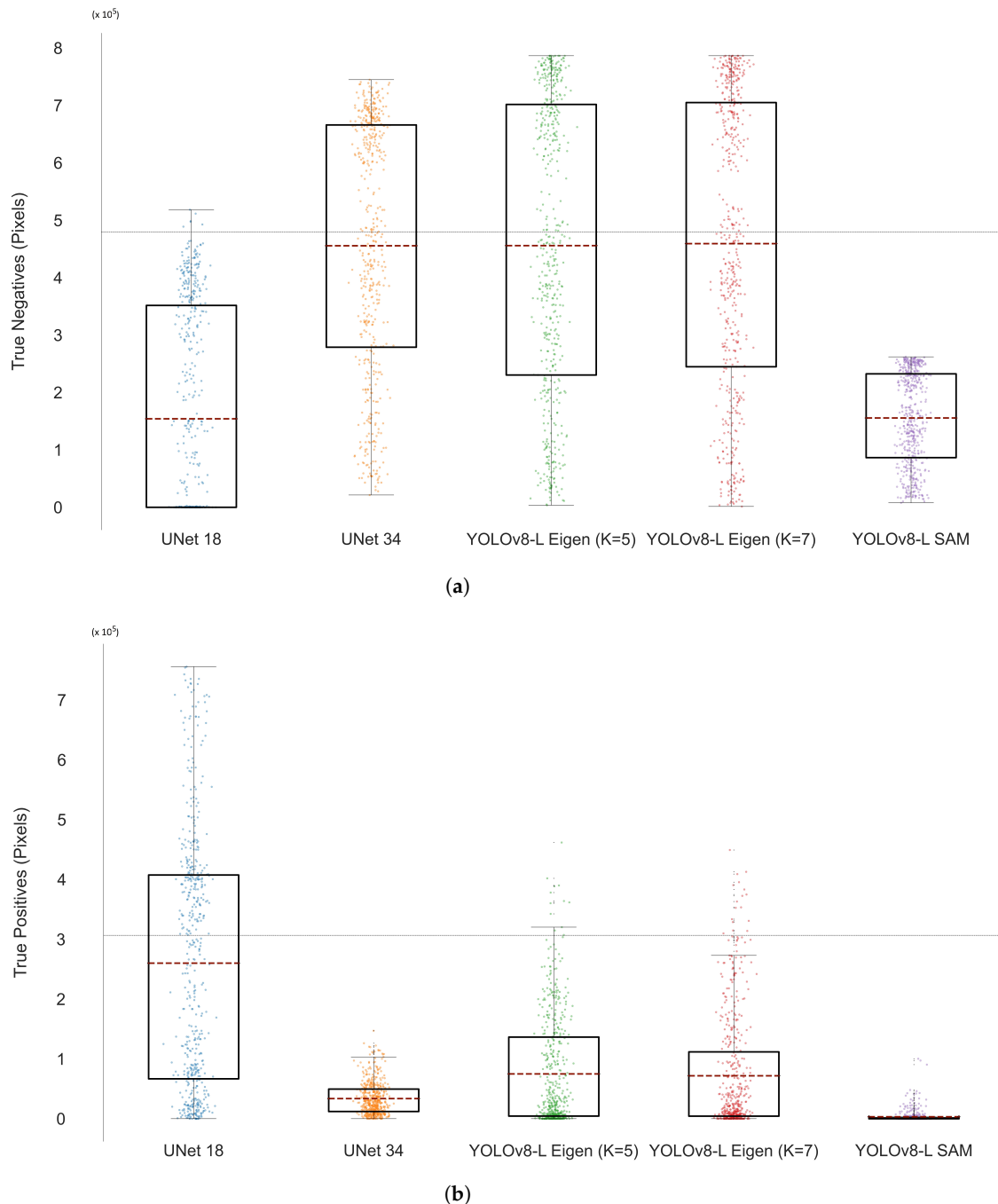


Figure 6. True pixel associations as boxplot representations for the methods used; from left to right: UNet SE18, SE34, YOLO-Eigen $k = 5$, $k = 7$, and YOLO-SAM. Note the dashed line baseline running across the plot reporting the average (in number of pixels) over the ground-truth-labelled images. (a) True negative (TN) boxplots. (b) True positive (TP) boxplots.

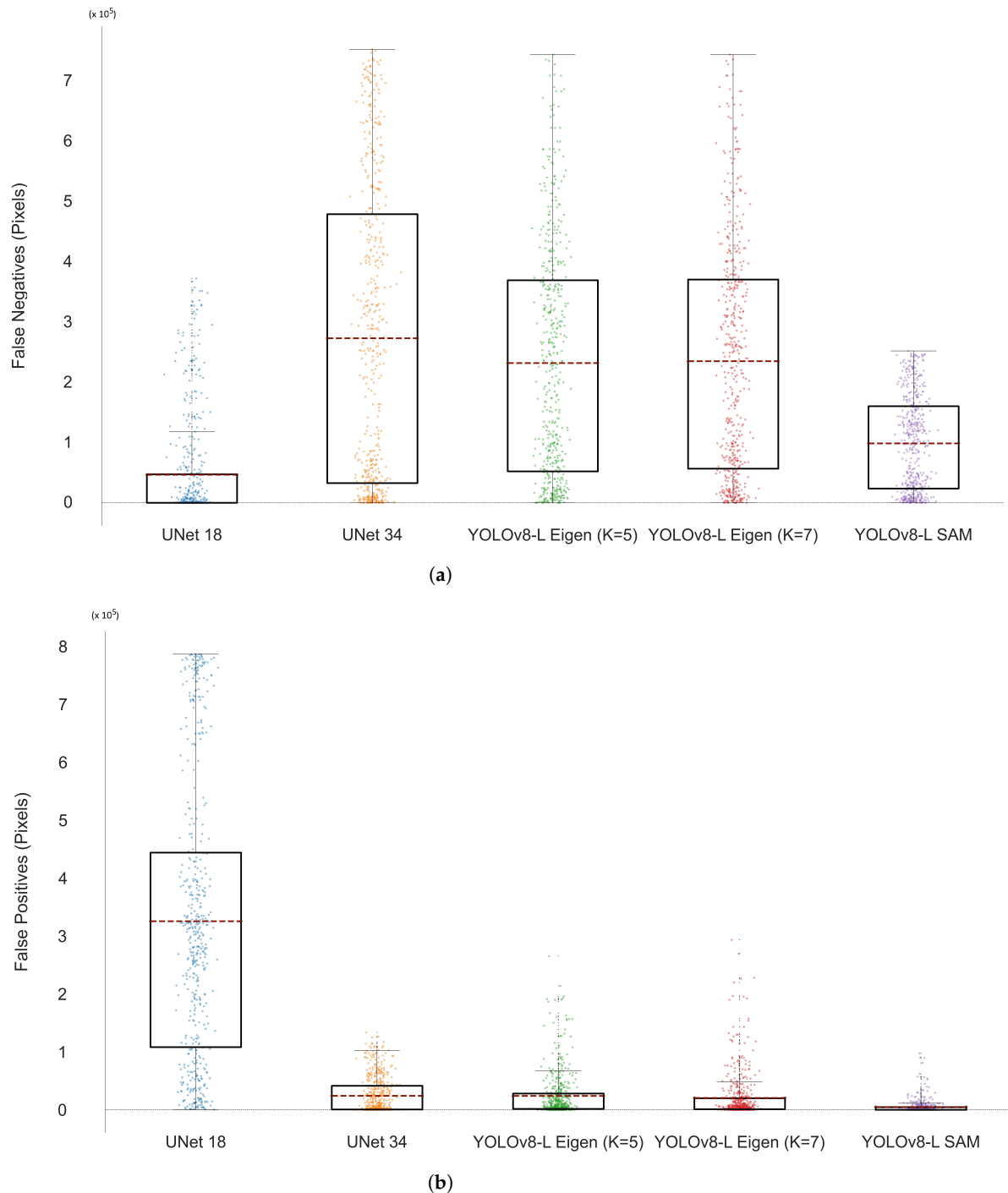


Figure 7. False pixel associations as boxplot representations for the methods used; from left to right: UNet SE18, SE34, YOLO-Eigen $k = 5$, $k = 7$, and YOLO-SAM. (a) False negative (FN) boxplots. (b) False positive (FP) boxplots.

The boxplots of Figure 6 represent the behaviour of true pixel (TP and TN) association instances, with the ground truth average baseline denoted as a dashed line across all plots; this is similar to the false pixel associations in Figure 7, which represent FP and FN instances. These boxplots illustrate the spread of the detected pixel group (size of box) with respective average values (line inside each box) and for each method, i.e., from left to right: UNet SE18 and SE34, YOLO-Eigen for $k = 5$ and $k = 7$, and YOLO-SAM.

For the true negative (TN) group of pixels of Figure 6a, the best performance seems to be that of YOLO-Eigen, with the rest of the methods being significantly away from

the ground truth baseline, which represents the TN total average. However, the inverse statement applies for the true positive (TP) pixels in Figure 6b. This provides us with the indication that the UNet approach is more accurate in its prediction of true positive, and less accurate in the prediction of true negative pixels. The YOLO-Eigen approach seems to be more accurate with true negative as opposed to true positive pixels.

With regard to the boxplot investigation on the predictive power of the methods with respect to the pixel group association, it can also be observed in Figure 7a,b that the UNet approach produces a significant number of erroneously predicted false positive pixels and significantly less false negative pixels. The exact opposite is true for our YOLO-Eigen approach (as per Figure 6) as it produces significantly fewer erroneous predictions of false positives, albeit at the expense of increased false negatives.

It is thus evident in this performance evaluation that in terms of predictive accuracy and respective significance of predictions, the UNet and our YOLO-Eigen methods produce similar results, as opposed to the BNN and YOLO-SAM approaches. However, an important conclusion is that the UNet approach correctly predicts more positive than negative pixels, albeit at the expense of incorrect predictions of the same type. Therefore, the important conclusion is that the UNet approach correctly predicts more positive than negative pixels, at the expense of incorrect predictions of the same type; for our YOLO-Eigen approach, however, the exact opposite seems to be the case.

3.3. Results Analysis

Following our main conclusion from Section 3.2, it follows that the UNet approach is biased to providing more positive segmentation pixel predictions, as opposed to our YOLO-Eigen approach that generates more negative segmentation pixel predictions. The BNN and YOLO-SAM methods seem to not maintain a sufficient coverage of corrosion segmentation when compared to the ground truth. As a consequence, we proceed in utilising the metrics of Section 3.1 to provide a specific analysis of the methods' segmentation results over the entire dataset. These results are reported in Table 1.

With reference to Table 1, it can be deduced that the BNN approach is more sensitive and specific in the predictions it produces, but since it does not have sufficient coverage, it achieves reduced performance in terms of accuracy and precision. The YOLO-SAM approach seems to over-fit in terms of accuracy and precision, since said predictions seem to be neither specific enough nor sensitive enough. As a result, the significance score (f-score) is sufficiently smaller than the YOLO-Eigen and UNet methods.

In addition to the result analysis, it should be evident from Table 1 that the best result accuracy as well as precision is that of our YOLO-Eigen method, with significance (f-score) comparable to that of the UNet method. It should be noted that the UNet method produces comparably good results in all the metrics that we have examined. As a result, the UNet method seems to be producing higher significance scores than any other method. This verifies the discussion of Section 3.2, whereas YOLO-Eigen and UNet were significantly better as opposed to the other methods in terms of true corrosion pixel predictions even though YOLO-Eigen produced more TP-versus-TN pixel predictions, and vice versa for UNet.

However, observing Table 1 and closely comparing the significance scores between UNet and YOLO-Eigen, we can conclude that our YOLO-Eigen for $k = 5$ produces the same score as that of UNet SE-18 (both at 0.41), and has a maximum difference of less than 0.09. Additionally, and for the mAP metric, it can be argued that the best methods are the YOLO-Eigen at $k = 7$ and UNet SE34 (both with a score of 0.53), with the next best being that of YOLO-Eigen at $k = 5$ (with a score of 0.52). As a result, it depends on the metric one uses to infer which method produces more significant results: the UNet approach seems to produce better significance f-scores upon its accuracy and precision, whilst our YOLO-Eigen approach produces better significance scores upon the mean average precision scores. In all cases, our YOLO-Eigen approach significantly outperforms all methods related to metrics for accuracy and precision.

Table 1. Method comparison: All reported values are mean values over the testing dataset portion. Best metric score is reported in dark blue, with next best score reported in light blue.

| | BNN (SpotRust) | | UNet (SEResNet) | | YOLO-Eigen | | YOLO-SAM |
|----------------------------|----------------|--------------|-----------------|-------------|--------------|--------------|----------|
| | Variational | Drop Out | SE-18 | SE-34 | $k = 5$ | $k = 7$ | |
| Accuracy (%) | 14.70 | 10.58 | 45.68 | 51.57 | 68.74 | 67.42 | 61.82 |
| Sensitivity (%) | 83.28 | 86.06 | 50.76 | 56.04 | 28.09 | 25.39 | 16.35 |
| Specificity (%) | 85.31 | 89.43 | 44.29 | 51.29 | 25.27 | 25.71 | 17.83 |
| Precision (%) | 11.25 | 11.21 | 34.02 | 41.12 | 77.28 | 73.97 | 64.89 |
| <i>mAP</i> (precision) | 0.42 | 0.26 | 0.44 | 0.53 | 0.52 | 0.53 | 0.41 |
| <i>f-score</i> (precision) | 0.19 | 0.19 | 0.41 | 0.47 | 0.41 | 0.39 | 0.25 |

4. Conclusions

In this paper, we have proposed an Eigen tree decomposition module for pre-trained YOLOv8 neural network models, referred to as YOLO-Eigen. The YOLOv8 model was trained via a custom marine corrosion dataset used. We have compared our YOLO-Eigen against other state-of-the-art methods, such as freely available source code for the UNet convolutional neural network, the Bayesian neural networks, and SAM as an add-on module to our pre-trained YOLOv8 model. We have hypothesised that due to the multiple pixel quantisation levels of the Eigen Tree decomposition, the YOLO model would produce better segmentation results than other techniques that are similar in nature for corrosion in marine vessels. The methodology presented in this study has the potential to significantly impact the field of marine maintenance and inspection by providing a more efficient and reliable solution for timely identification and resolution of corrosion issues.

We have verified that for the YOLOv8 pre-trained models, our Eigen module segments corrosion on vessel surfaces accurately (metric score higher by at least 10%) and is more precise (metric score higher by at least 30%), with significance comparable to that of the next best method (UNet), across testing dataset inputs (significance within the margin of 0.01 to 0.06 absolute difference). Inasmuch, the next best method (UNet) produces similar significance score results (*f-score*, *mAP*) but achieves lower accuracy and significantly lower precision performance. However, our YOLO-Eigen method seems to be biased to predictions of higher false positives, as opposed to false negatives that the UNet method produces. It remains a question of risk assessment for the inspection surveyor as to whether having false positive predictions (YOLO-Eigen) is more preferable than having more false negative (UNet) predictions.

As future work, we aim to examine further enhancing our technique with three-dimensional point-clouds in the hopes that these will correlate to further identifying different types of corrosion. That is, assuming a 3D point-cloud with chroma (RGB) values, it may be possible to identify cracks on the surface and subsequently correlate to corrosion spread. It may also be possible to predict the extent of hull damage and/or predictive maintenance to avoid failure. In addition, we propose future work whereby a full integration of mobile robot arms is involved, along with the use of the computer vision and algorithm described in this paper, so as to proceed to targeted micro-indentation mapping, and consequently to be used as a novel procedure in different types of corrosion detection (e.g., microbial) and in predicting potential fatigue damage.

Author Contributions: Conceptualisation, G.C. and S.V.K.; methodology, G.C.; investigation, G.C. and S.V.K.; data curation, I.T.; validation, I.T. and G.C.; software, I.T. and G.C.; formal analysis, G.C. and I.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable; data acquired are standard camera images that do not include any human or vessel identification features or markings.

Informed Consent Statement: Informed consent was obtained from human operators involved in the study at the data collection stage; the dataset images include no identification features or markings.

Data Availability Statement: The raw data used in this study are from the freely available [MaVeCoDD marine vessel corrosion dataset](#) (Last Accessed 12 June 2024).

Conflicts of Interest: Author Iason Tzanetatos was employed by the company Core Innovation and Technology OE. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Thompson, N.G.; Yunovich, M.; Dunmire, D. Cost of corrosion and corrosion maintenance strategies. *Corros. Rev.* **2007**, *25*, 247–262. [[CrossRef](#)]
2. Imran, M.M.H.; Jamaludin, S.; Mohamad Ayob, A.F. A critical review of machine learning algorithms in maritime, offshore, and oil & gas corrosion research: A comprehensive analysis of ANN and RF models. *Ocean. Eng.* **2024**, *295*, 116796. [[CrossRef](#)]
3. Melchers, R.E. Corrosion uncertainty modelling for steel structures. *J. Constr. Steel Res.* **1999**, *52*, 3–19. [[CrossRef](#)]
4. Jones, R.; Singh Raman, R.; McMillan, A. Crack growth: Does microstructure play a role? *Eng. Fract. Mech.* **2018**, *187*, 190–210. [[CrossRef](#)]
5. Rodopoulos, C.; Chliveros, G. Fatigue damage in polycrystals—Part 1: The numbers two and three. *Theor. Appl. Fract. Mech.* **2008**, *49*, 61–76. [[CrossRef](#)]
6. Rodopoulos, C.; Chliveros, G. Fatigue damage in polycrystals—Part 2: Intrinsic scatter of fatigue life. *Theor. Appl. Fract. Mech.* **2008**, *49*, 77–97. [[CrossRef](#)]
7. International Maritime Organization. International Convention for the Safety of Life at Sea. Available online: <https://www.refworld.org/docid/46920bf32.html> (accessed on 12 May 2023).
8. Momber, A.W.; Langenkämper, D.; Möller, T.; Nattkemper, T.W. The exploration and annotation of large amounts of visual inspection data for protective coating systems on stationary marine steel structures. *Ocean. Eng.* **2023**, *278*, 114337. [[CrossRef](#)]
9. Zhang, J.; Cho, Y.; Kim, J.; Malikov, A.K.u.; Kim, Y.H.; Yi, J.H.; Li, W. Non-destructive evaluation of coating thickness using water immersion ultrasonic testing. *Coatings* **2021**, *11*, 1421. [[CrossRef](#)]
10. Bonnin-Pascual, F.; Ortiz, A. On the use of robots and vision technologies for the inspection of vessels: A survey on recent advances. *Ocean. Eng.* **2019**, *190*, 106420. [[CrossRef](#)]
11. Nash, W.; Zheng, L.; Birbilis, N. Deep learning corrosion detection with confidence. *NPJ Mater. Degrad.* **2022**, *6*, 26. [[CrossRef](#)]
12. Ali, A.A.I.M.; Jamaludin, S.; Imran, M.M.H.; Ayob, A.F.M.; Ahmad, S.Z.A.S.; Akhbar, M.F.A.; Suhrab, M.I.R.; Ramli, M.R. Computer Vision and Image Processing Approaches for Corrosion Detection. *J. Mar. Sci. Eng.* **2023**, *11*, 1954. [[CrossRef](#)]
13. Alboul, L.; Chliveros, G. A system for reconstruction from point clouds in 3D: Simplification and mesh representation. In Proceedings of the 11th International Conference on Control Automation Robotics & Vision, Singapore, 7–10 December 2010; pp. 2301–2306. [[CrossRef](#)]
14. Chliveros, G.; Pateraki, M.; Trahanias, P. Robust Multi-hypothesis 3D Object Pose Tracking. In *Proceedings of the Lecture Notes in Computer Science, Vol. 7963: Computer Vision Systems*; Chen, M., Leibe, B., Neumann, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 234–243. [[CrossRef](#)]
15. Atha, D.J.; Jahanshahi, M.R. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct. Health Monit.* **2018**, *17*, 1110–1128. [[CrossRef](#)]
16. Coelho, L.B.; Zhang, D.; Van Ingelgem, Y.; Steckelmacher, D.; Nowé, A.; Terry, H. Reviewing machine learning of corrosion prediction in a data-oriented perspective. *NPJ Mater. Degrad.* **2022**, *6*, 8. [[CrossRef](#)]
17. Hussein Khalaf, A.; Xiao, Y.; Xu, N.; Wu, B.; Li, H.; Lin, B.; Nie, Z.; Tang, J. Emerging AI technologies for corrosion monitoring in oil and gas industry: A comprehensive review. *Eng. Fail. Anal.* **2024**, *155*, 107735. [[CrossRef](#)]
18. 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. *Expert Syst. Appl.* **2022**, *193*, 116461. [[CrossRef](#)]
19. Guzmán-Torres, J.A.; Domínguez-Mota, F.J.; Martínez-Molina, W.; Naser, M.; Tinoco-Guerrero, G.; Tinoco-Ruíz, J.G. Damage Detection on Steel-Reinforced Concrete Produced by Corrosion via YOLOv3; A detailed guide. *Front. Built Environ.* **2023**, *9*, 41. [[CrossRef](#)]
20. Brandoli, B.; de Geus, A.R.; Souza, J.R.; Spadon, G.; Soares, A.; Rodrigues, J.F.; Komorowski, J.; Matwin, S. Aircraft Fuselage Corrosion Detection Using Artificial Intelligence. *Sensors* **2021**, *21*, 4026. [[CrossRef](#)] [[PubMed](#)]
21. Das, A.; Ichi, E.; Dorafshan, S. Image-Based Corrosion Detection in Ancillary Structures. *Infrastructures* **2023**, *8*, 66. [[CrossRef](#)]
22. Chliveros, G.; Kontomaris, S.V.; Letsios, A. Automatic Identification of Corrosion in Marine Vessels Using Decision-Tree Imaging Hierarchies. *Eng* **2023**, *4*, 2090–2099. [[CrossRef](#)]
23. Lin, B.; Dong, X. A multi-task segmentation and classification network for remote ship hull inspection. *Ocean. Eng.* **2024**, *301*, 117608. [[CrossRef](#)]
24. Ortiz, A.; Bonnin-Pascual, F.; Garcia-Fidalgo, E.; Company-Corcoles, J. Vision-Based Corrosion Detection Assisted by a Micro-Aerial Vehicle in a Vessel Inspection Application. *Sensors* **2016**, *16*, 2118. [[CrossRef](#)]

25. Nash, W.; Drummond, T.; Birbilis, N. Deep Learning AI for corrosion detection. In Proceedings of the NACE International CORROSION Conference Proceedings, NACE-2019-13267, Nashville, TN, USA, 24–28 March 2019.
26. Yigit, K.; Adanur, M. Examination of the Potential Effect of Corrosion Current Density of Ship Hulls on the Sacrificial Anode Cathodic Protection. *Bitlis Eren Üniversitesi Fen Bilim. Derg.* **2023**, *12*, 292–298. [[CrossRef](#)]
27. Lin, B.; Dong, X. Ship hull inspection: Survey. *Ocean Eng.* **2024**, *289*, 116281. [[CrossRef](#)]
28. Orchard, M.T.; Bouman, C.A. Color Quantization of Images. *IEEE Trans. Signal Process.* **1991**, *39*, 2677–2690. [[CrossRef](#)]
29. Manwani, N.; Sastry, P.S. Geometric Decision Tree. *IEEE Trans. Syst. Man Cybern.* **2012**, *42*, 181–192. [[CrossRef](#)]
30. Sander, T.; Sander, J. Tree decomposition by eigenvectors. *Linear Algebra Its Appl.* **2009**, *430*, 133–144. [[CrossRef](#)]
31. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. [[CrossRef](#)]
32. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
33. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 27–30 June 2016; pp. 779–788. [[CrossRef](#)]
34. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 4015–4026.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.