Investigating Deep Learning Architectures towards Autonomous Inspection for Marine Classification

Andersen, Rasmus Eckholdt; Nalpantidis, Lazaros; Ravn, Ole; Boukas, Evangelos

Published in:

Link to article, DOI:
10.1109/SSRR50563.2020.9292621

Publication date:
2020

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Investigating Deep Learning Architectures towards Autonomous Inspection for Marine Classification*

Rasmus Andersen    Lazaros Nalpantidis    Ole Ravn    Evangelos Boukas

Abstract— Marine vessels undergo periodic inspections under which the condition of the vessel is documented. A part of these inspections is to detect defects such as corrosion that degrade the structural integrity of the vessel. The goal of this paper is to evaluate several deep learning architectures and create a hierarchical pipeline that best fits an autonomous inspection system, in the form of an unmanned aerial vehicle, capable of detecting defects in the ballast tanks of a marine vessel. Due to the limited resources available on such an autonomous system, we devised and tested a pipeline to use a smaller deep learning architecture to trigger a larger one when the presence of corrosion is detected. The produced segmentation can then be used to compute the condition of the vessel. In total ten architectures/combinations were tested ranging from traditional classification to object detection and instance segmentation. All the architectures were trained on a dataset containing images from ballast tanks with varying degree of corrosion. The results presented in this paper show that regular object localization architectures such as YOLO and Faster-RCNN suffer from overestimation of the affected corroded area. Binary whole image classification followed by instance segmentation proved to be the best performing pipeline.

I. INTRODUCTION

One of the most efficient and cost effective ways of transporting goods is by sea. In fact, according to the UN [1], over 80% merchandise trade by volume is carried by sea. However, despite the efforts of shielding the marine vessels from corroding, accidents due to weakened structures still happen [2]. Therefore, classification societies investigate the compliance of vessels’ structural integrity with international standards in a system called marine classification. Marine classification involves inspections which are performed annually, with every fifth year requiring an extensive inspection in dry-dock. Working towards autonomous inspection of vessels, we investigate the employment of Machine Learning (ML) for automatic identification of defects. This work is performed in the context of the Inspectrone project, where the inspection will be performed by an autonomous agent such as the one presented in [3], or similarly to an unmanned aerial vehicle (UAV) as shown in Fig. 1.

*This work has been funded by the Innovation Fund Denmark (IFD), through the Inspectrone (Autonomous and high-level commanded system for remote inspection of marine vessels to support classification and commercial operations) project, under contract number 8090-00080B.

Rasmus Andersen, Lazaros Nalpantidis, Ole Ravn and Evangelos Boukas are with section of Automation and Control, Department of Electrical Engineering, Technical University of Denmark, 348, Ørsteds Pl., 2800 Kongens Lyngby, Denmark recan@elektro.dtu.dk, lanalpa@elektro.dtu.dk, or@elektro.dtu.dk, evbou@elektro.dtu.dk

Fig. 1: A drone flying in a ballast tank equipped with a camera used to detect corrosion.

A. Problem Formulation

The inspection is divided into three surveys each contributing to the final classification of the vessel; an overall, a close-up, and a thickness survey. The output of all the surveys is a classification of the vessel, or parts of the vessel, into the three categories (also shown in Fig. 2):

- Good: < 3% corrosion.
- Fair: < 20% corrosion.
- Poor: > 20% corrosion.

The overall survey documents the condition of a vessel at a macro scale and is used to determine whether a vessel follows international safety standards and is allowed to sail both locally (i.e. follows the local countries specifications for marine vessels) and internationally (i.e. standards set by the International Maritime Organisation (IMO)).

In contrast to the overall survey, where the ship is considered at a macro scale, during a close-up survey the vessel is inspected in more detail to identify local bucklings, cracks, coating breakdown, and other defects, so that they can be repaired. Despite the efforts of the classification societies in detailed guidelines and extensive training, it is known that the major cause of inconsistencies in close-up surveys, is the subjectivity of human surveyors [4]. Different surveyors may classify the severity of faults found in a close-up survey differently. Even the same surveyor may classify the exact same vessel differently depending on tiredness and past experience.
The main contribution of this paper is the introduction of a hierarchical learning system and the evaluation of popular existing architectures in the area of corrosion and defect detection, specifically on marine vessels. Generally, the topic has attracted limited research in the usage of deep learning to classify the condition of marine vessels. With the results presented in this paper, there will be a better understanding of which architectures should be used in such an application area and what should be the topology of a deep learning-based defect detection system.

II. RELATED WORK

Visual inspection plays an intrinsic role in many quality assurance applications. This includes manufacturing as well as maintenance. Historically, image processing and image classification has relied on engineered features that are used to segment or classify images. Though detection of corrosion is fit for image analysis, it has received little interest from the research community with few contributions. A scholar reviewing the literature will find that a significant part of the existing contributions does not utilize the recent major advances within deep learning but, instead, relies on traditional computer vision in the form of thresholding, weak classifiers, and image gradients [16], [17].

The work in [18] presented a method which employs contrast and symmetry metrics as features for defect detection. Similarly, the works in [19]–[22] uses weak classifiers in combination with features based on the color intensities in the image. The authors in [23] applied watershed on the gradient of gray-scale images to detect corrosion in metal surfaces. Even though custom made features are easy to explain and interpret, they create a restricted and non-abstract feature-space and, therefore, are limited in description performance.

More recent advances such as those within artificial neural networks have also been explored to a limited extend. A 3-layer feed forward artificial neural network was presented in [24] consisting of purely dense layers which, however, suffered from a high false positive rate. Some custom deep learning approaches for corrosion detection have also been developed [20], [25], [26]. These approaches employ whole-image classification and since none of their architectures are tested within the subject of marine vessel, their applicability to the problem at hand is yet to be defined. In fact, their applicability in the considered domain is not straightforward as the objective of marine vessel inspection is to document the extend of corrosion in an area, rather than the severity of local corrosion, as the latter is performed by ultrasonic thickness measurements, due to the need for high accuracy.

The team of authors in [27] showed that using a convolutional neural network as the feature extractor in combination with classification layers can be used as an approximate segmentation by sliding the classification window over the input image and subsequently classify each region. This requires inference on hundreds of regions for each image, and is thus a slow process. Naturally, they moved to a Faster-RCNN model in [5] employing a ZF-net feature extractor for better real-time performance. Similarly, [28] used a Faster-RCNN model with a VGG16 feature extractor to detect corrosion. As mentioned in the introduction, the

B. Approach

With the recent advances in deep learning (DL) for image classification and segmentation, it is possible to adapt existing general purpose architectures for case-specific problems such as corrosion detection [5]. The goal of this paper is to create a DL pipeline consisting of existing architectures that best fit an autonomous inspection system. The existing architectures for this include domains such as whole image classification, object detection, and image segmentation. Additionally, the performance of each architecture will be compared within each classification domain.

Even though each of those domains by itself is not able to solve the problem both accurately and efficiently, a combination of those architectures may yield a satisfactory solution. The whole image classification could theoretically provide a solution to the overall classification problem and is relatively computational efficient, but is not able to pinpoint where any corrosion is located in the image but only whether there is corrosion present. The object detection architectures can provide targets for the thickness measurement process, but would not be a good standalone solution for the overall classification as they, by design, overestimates the affected area. The segmentation architectures allow for computing a more precise level of corrosion as well as easily determine exactly where the corrosion is located and how widespread it is, but have the cost of increased computational power [6] and, hence, longer overall mission duration. The considered architectures are: for whole image classification: VGG16, VGG19 [7], ResNet50 [8], Inception V3 [9], and Inception ResNet V2 [10]; for object detection YOLO [11], Yolact [12], and Faster-RCNN [13], [14]; and for image segmentation Mask-RCNN [14], [15], and Yolact.

The main contribution of this paper is the introduction of a hierarchical learning system and the evaluation of popular existing architectures in the area of corrosion and defect detection, specifically on marine vessels. Generally, the topic has attracted limited research in the usage of deep learning to classify the condition of marine vessels. With the results presented in this paper, there will be a better understanding of which architectures should be used in such an application area and what should be the topology of a deep learning-based defect detection system.

Fig. 2: Examples of corrosion and the corresponding masks to identify.
Fig. 3: Two approaches for the overall system design. Both approaches use a hierarchical model to first identify whether the image contains any corrosion, secondly, further investigate and localize the defects. The first approach uses the total area of defects for the condition assessment, while the second one performs the assessment directly in the whole image. Those two approaches are compared in this paper along with the different architectures for each of the learning modules. Highlighted are the two main products of the system: 1) the location of the defects and the condition assessment of every image.

The application of regional proposal for determining the overall classification is practically unfit, since it overestimates the affected area, e.g.: by not being able to distinguish between a fully corroded convex area and a slightly corroded concave one.

The paper at hand is unique in the area of marine inspection, as it’s the first one to present a hierarchical structure by evaluating the available options both in terms of detection performance and computational efficiency.

III. SYSTEM ARCHITECTURE

System developers of modern deep learning architectures are faced with a major trade-off between performance and required computing power. This trade-off becomes especially apparent when a system operates with limited energy capacity, as is the case within the application considered in this paper. Since corrosion can occur on all metal structures on a maritime vessel, any autonomous inspection system has to be able to physically fit in confined spaces, such as ballast tanks. Additionally, such a system must be operational for a long enough period of time to allow for any meaningful progress of the inspection to be carried out. By nature the effective power/time in a UAV is limited compared to locomotion and navigation time and, therefore, we have to design computationally efficient software systems.

In Fig. 3 we present the two proposed competing approaches (A, B) for limiting the required computational load. Both of the proposed hierarchical approaches employ a smaller neural network architecture to provide an indication of whether the currently observed image is worth the extra energy and time on further analysing, using a larger network to localize the defects. While both architectures can localize defects, the main difference between the two approaches lies in the way that the assessment—in terms of condition classification (good, fair or poor)—is performed. Approach A evaluates the image as whole, while approach B performs the condition assessment by measuring the total area of defects.

The preliminary assessment of whether an image contains defects can be setup as a whole image classification problem. In approach A a 4-category —i.e.: No Defect, Good, Fair, Poor— classifier is used, while we employ a binary classifier in approach B. For the in-image location of defects —so as to be able to relocate them for reparation— we compare among the most proven and widely applied works in object detection and image segmentation (or instance detection).

IV. INVESTIGATED DL ARCHITECTURES FOR MARINE CLASSIFICATION

The architectures investigated in this paper constitute the results of an evolutionary development and can be viewed as iterations towards increasing the overall performance of image classification and object detection.

A. Whole image classification

The efforts to achieve higher performance in image classification have been driven by the usage of evermore deeper networks. The main problem with deeper networks is vanishing gradients. As more layers are added, the number of parameters grow and, therefore, the contribution of a single parameter to the output vanishes. The architectures, investigated in this paper, deal with the vanishing gradient
problem in different ways. An abstract view of the principles for each architectures is shown in Fig. 4. The VGG16 and VGG19 networks utilize a ReLu activation function as opposed to the, then standard, sigmoid function. Inception decreases the number of learnable parameters by factorizing the layers to use smaller convolutional kernels (i.e. having two 3x3 kernels instead of one 5x5), and factorizing the spatial shape of the network into inception modules consisting of layers of different size and shape. Additionally, it introduced an auxiliary classifier as a regularizer. The ResNet50 architecture is —similarly to the inception network— divided into blocks but in a simpler fashion by using identity blocks. The shortcuts introduced in each identity block, allow the gradient of the lower layers to reach the higher layers and, thus, the gradient doesn’t vanish. Because the layers don’t downscale the features, the depth of the network can be greatly increased by stacking more identity blocks.

The final architecture for this paper is the Inception ResNet that adds the shortcuts from the ResNet architecture to the Inception architecture. Even though there is a clear evolution between those architectures, their performance to our application cannot be guaranteed, and, therefore, need to be evaluated.

B. Object detection

The main component of the whole-image classifiers, is the feature extraction. Consequently, these features can be used to model or locate objects in an image. Therefore, all object detection architectures consist of a feature extraction part followed by region proposal. Thus, the main difference among the object detection methods is the region proposal. For the Faster-RCNN architecture, anchor regions are classified as background or foreground using a regional proposal network, while the YOLO architecture divides the image into a static number of regions that each produce a static number of bounding boxes. One of the practical differences between these architectures is the fact that YOLO sacrifices the ability to detect smaller objects to achieve higher computational performance than Faster-RCNN [29]. In this paper we re-evaluate this trade-off in the context of our application.

C. Instance segmentation

By adding a new branch to the Faster-RCNN architecture that outputs a mask, Faster-RCNN can be upgraded to produce object segmentations in a new architecture named Mask-RCNN. For each region detected by Faster-RCNN, a mask containing the object will also be produced. While both Mask-RCNN and Yolact use ResNet101 as a backbone, Yolact —instead of using a single mask— splits the segmentation into two branches that both produce object-independent segmentations which can be linearly combined to a final segmentation output.

V. EXPERIMENTAL WORK

The dataset used in this paper consists of 820 images of corrosion and 494 of non-corrosion at various stages in marine vessels’ lifetime, though the majority of images originate are from vessels with either good or fair condition - i.e. early in the lifetime of the vessel. 70% of the images were used for training while the remaining 30% was used for validation. The dataset contains different types of annotation. There is an annotation of the condition of the whole image in the categories good, fair, poor. Additionally, we have included an annotation for localization of the defects (bounding box and segmentation). Finally for each defect, there are further categories as: Buckling, Welded Seam Corrosion, Edge Corrosion, Isolated Spot Corrosion. The motivation for including separate categories for each type of corrosion is to be able to prioritize repairs.

With the relatively small dataset, all networks have been pre-trained on the imagenet dataset. Similarly, the architectures and training procedures have not been modified other than outputting the four given types of defect classes rather than the classes from the imagenet.
A. Whole image classifier

1) 4-category classifier: The idea behind the system architecture (Fig. 3) is to avoid the execution of localization architectures when no corrosion is present in the image. Naturally, classifying the categories: good, fair, poor; becomes problematic due to the fact, that no defects — i.e. a ship in perfect condition — will fall in the category good. Thus the classification has to include 4 categories; no defect, good, fair, poor. Of the five classifiers, only three of them were able to converge to a solution of which the results are shown in Fig. 5. From the figure, it is apparent that all classifiers had trouble identifying the fair category from which almost none were classified correctly. The average precision between the three architectures are also close with Inception and Inception ResNet providing only a 4% performance increase.

2) Binary classifier: In accordance to Fig. 3, it is possible to merge the good, fair, poor categories into one, such that the problem becomes a binary classification, i.e. defect vs no defect. Fig. 6 shows the results of the 5 tested classifiers on the binary corrosion dataset. It can be seen that all classifiers are able to obtain a high performance with all classifiers achieving above 95% average precision. Since missing a defect can potentially have catastrophic consequences, both ecologically as well as economically, it is generally preferred to sacrifice precision to ensure a lower false negative rate. Of the five classifiers, ResNet achieved the highest recall score of 0.99 slightly more than the second best score of 0.98 from the Inception architecture.

Based on the investigation of the two different approaches for condition assessment we find that the binary whole image classification outperforms the multi-class classifier. Among the binary classifiers, the ResNet was able to produce the highest precision while maintaining an advantageous recall.

B. Object detection

As already mentioned, using bounding boxes for object localization causes inaccuracies in the sense as corrosion may be present along a thin diagonal edge (e.g. a welded seam) which may produce a large bounding box for a potential insignificant amount of corrosion, as shown in Fig. 7. For these experiments, two object detection architectures were used, as well as the intermediate bounding box object detection produced by the Yolact architecture. Based on the results shown in table I, the best performing architecture is the Faster-RCNN with only a 2% mean average precision increase (i.e. the best average precision over all four corrosion categories). However, both the precision and recall is low indicating that for the given scenario, the architecture produces a high number of false negatives and false positives.
TABLE I: COCO evaluation results from corrosion detection for the 5 different architectures.

<table>
<thead>
<tr>
<th>IoU</th>
<th>area</th>
<th>maxDets</th>
<th>Object detection</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>F-RCNN</td>
<td>Yolact</td>
</tr>
<tr>
<td>Average Precision 0.50/0.95</td>
<td>all</td>
<td>100</td>
<td>0.172</td>
<td>0.048</td>
</tr>
<tr>
<td>Average Precision 0.50</td>
<td>all</td>
<td>100</td>
<td>0.376</td>
<td>0.193</td>
</tr>
<tr>
<td>Average Precision 0.75</td>
<td>all</td>
<td>100</td>
<td>0.152</td>
<td>0.025</td>
</tr>
<tr>
<td>Average Precision 0.50/0.95 small</td>
<td>100</td>
<td>0.01</td>
<td>0</td>
<td>0.003</td>
</tr>
<tr>
<td>Average Precision 0.50/0.95 medium</td>
<td>100</td>
<td>0.103</td>
<td>0.049</td>
<td>0.089</td>
</tr>
<tr>
<td>Average Precision 0.50/0.95 large</td>
<td>100</td>
<td>0.234</td>
<td>0.066</td>
<td>0.231</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95</td>
<td>all</td>
<td>100</td>
<td>0.142</td>
<td>0.065</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95</td>
<td>all</td>
<td>10</td>
<td>0.213</td>
<td>0.147</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95</td>
<td>all</td>
<td>100</td>
<td>0.213</td>
<td>0.162</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95 small</td>
<td>100</td>
<td>0.016</td>
<td>0.02</td>
<td>0.007</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95 medium</td>
<td>100</td>
<td>0.126</td>
<td>0.102</td>
<td>0.099</td>
</tr>
<tr>
<td>Average Recall 0.50/0.95 large</td>
<td>100</td>
<td>0.294</td>
<td>0.226</td>
<td>0.269</td>
</tr>
</tbody>
</table>

An explanation for this may be the ambiguity of the problem. This ambiguity is then further amplified by corrosion residue that is spread by water flow. In such cases, we may detect seemingly corroded areas, but in reality, the residue could be cleaned off revealing no damage to the underlying paint or steel structure. In Fig. 7 a thin light-brown line can be observed below the actual corrosion segmentation. This is an indication of a past long-time stationary water level aligning with the exact spot of this residue. Also shown in table I is the performance for small, medium, and large objects which shows that all the networks performs the best on larger objects. It is believed this stems from the previous mentioned residue confusing the network and producing false positives and false negatives. The relevance of using a pure object detection architecture is also diminished since the category classification showed poor performance, thus the combination of the two not fulfilling the need for good, fair, and poor categorization.

C. Segmentation

Based on the results of the object detection approaches, unsurprisingly, the Mask-RCNN (an extension of Faster-RCNN) is also the best performing architecture for instance segmentation, though this may be due to difficulties in getting the Yolact architecture to converge on the dataset, thus explaining the 0.115 gap. The mean average precision and recall are low which is attributed to the aforementioned ambiguity between surface residue and actual corrosion. As mentioned in the dataset description, the masks are divided into the four categories: Welded Seam Corrosion, Isolated Spot Corrosion, Edge Corrosion, and Buckling. These categories are not mutually exclusive. The most common overlap is observed between edge corrosion and welded seam corrosion, since seams often end at an edge. There is no official standard for determining whether a defect is classified as edge corrosion or welded seam corrosion and it is, thus, up to the individual surveyor to decide. We resolved these ambiguities in our dataset by using expert surveyors for evaluation.

D. Verdict

Following the extensive evaluation of the different approaches, the results of this investigation are summarized here. Among approaches A and B (Fig. 3), we found that approach B is the best performing one. Concerning the condition classification, we are convinced that the binary classification outperforms the multi-class one. Specifically, the best performing architecture for determining whether an image contains a defect or not is the ResNet50. Concerning defect localization, the segmentation approaches produce significantly better results than the object detection ones. Specifically, the best performing network for the defect segmentation was the Mask-RCNN by a significant margin. Finally, we have decided that the best pipeline for DL-based marine vessel inspection (Fig. 8) is as follows: An input image is evaluated by the ResNet50. If the image is found to contain defects, it is evaluated by the Mask-RCNN. The pixel count of defects determine the condition assessment of the area under inspection and the location of the defects are saved to be used for further investigation by means of ultrasonic thickness measurements.

VI. Conclusion

In this paper, two approaches for balancing the accuracy and power consumption required for performing marine vessel inspection were presented. The original problem consists of identifying areas with good, fair, and poor corrosion to limit the area for thickness measurements to be carried out. A total of ten architectures/combinations were tested on
Fig. 8: The hierarchical model presented in this paper, that best fit a UAV for doing corrosion detection on marine vessels.

marine inspection with varying success. The most promising approach for determining the presence of corrosion relies on a binary whole image classifier (specifically the ResNet50) which can trigger a segmentation architecture. Concerning defect localization, the segmentation architectures (specifically Mask-RCNN) outperform the object detection ones and, therefore, are preferred. On the other hand, we observed low accuracy on the defect localization due to the ambiguity between corrosion and rust residue.

VII. ACKNOWLEDGEMENTS

The authors of this paper would like to thank Lloyd’s Register for sharing their expertise in corrosion classification and maritime inspection standards. Additionally, we are thankful to both Dampskibsselskabet NORDER A/S and DFDS A/S for coordinating visits to their maritime vessels for data collection and first hand experience. Lastly, we wish to acknowledge Force Technology for providing inspection know-how and for supplying hardware for data acquisition.

REFERENCES


[26] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and
