



Detection of corrosion on steel structures using automated image processing

M. Khayatazad^{a,b,*}, L. De Pue^a, W. De Waele^a



^a SOETE LABORATORY, Department of Electromechanical, Systems and Metal Engineering, Faculty of Engineering and Architecture, Ghent University, 9052, Zwijnaarde, Belgium

^b SIM Vzw, Technologiepark 48, 9052 Zwijnaarde, Belgium

ARTICLE INFO

Keywords:

Steel structures
Corrosion detection
Roughness analysis
Color analysis
HSV color Space

ABSTRACT

The traditional method used for corrosion damage assessment is visual inspection which is time-consuming for vast areas, impossible for inaccessible areas and subjective for non-experts. A promising way to overcome the aforementioned drawbacks is to develop an artificial intelligence-based algorithm that can recognize corrosion damage in a series of photographic images. This paper reports on the implementation and use of an algorithm that quantifies and combines two visual aspects – roughness and color – in order to locate the corroded area in a given image. For the roughness analysis, the uniformity metric calculated from the gray-level co-occurrence matrix is considered. For the color analysis, the histogram of corrosion-representative colors extracted from a data-set in HSV color space is used. The algorithm has been applied to a large dataset of photographs of corroded and non-corroded components and structures. Our findings show that the developed algorithm can efficiently locate corroded areas.

1. Introduction

Corrosion is a frequently occurring failure mechanism for steel structural members and components. To give an idea, for engineering components this mechanism takes the lead in terms of the frequency of failure with 42%, according to (Petrovic, 2016). A study of NACE (Koch et al., 2016) estimates the global annual cost of corrosion in all its forms at US\$ 2,5 trillion, or about 3,4% of the global gross domestic product (year 2013). These numbers solely represent the direct costs such as forced shut-downs or accidents; neither individual safety nor environmental consequences are included. According to the NACE study, an appropriate corrosion strategy could decrease this cost by 18–35%. Early detection of structural degradation prior to failure does not only have financial benefits but can also prevent catastrophic collapses of structures and avoid harmful situations for both humans and the environment.

The first step towards the maintenance of structures is Visual Inspection (VI). This approach only treats surface defects and delivers a rough description of the condition of the structure and its deterioration. Nowadays this is mainly done by humans to collect qualitative data. Despite that these inspectors are certificated, the performance of this time-consuming method is subjective and largely dependent on the experience and qualifications of the individual, (Agdas et al., 2016). On

top of that, a lot of locations are difficult or completely inaccessible because of safety reasons e.g. toxic gasses or a hindering construction. In this paper, image processing towards the detection of corrosion on steel structures is investigated. The proposed objective-based technique aims to support the inspector during the VI, to quickly screen the structures through images taken by a drone reaching the inaccessible locations without bringing the safety of the inspector in danger.

The occurrence of corrosion comes with two main visual characteristics. In the first place, it creates a rough surface texture and secondly, the colors of the by-products are situated within a well-defined color spectrum. Therefore the use of texture analysis, color analysis or a combination of both is often used to develop algorithms for corrosion detection. These two features can be applied on a stand-alone basis or implemented in a pattern recognition technique.

Texture is the first feature that has been used for corrosion detection (Chen and Chang, 2003; AbdelRazig, 1999; Lee et al., 2005; Enikeev et al., 2017; Alkanhal, 2014). One of the requirements of texture analysis is the conversion of color images to grayscale ones. Chen et al. (2002) were one of the first to develop an image recognition technique for a bridge coating assessment. They extracted statistical features of the Gray-Level Co-occurrence Matrix (GLCM) of digital images and apply a clustering technique called Multi-resolution Pattern Classification

* Corresponding author. Ghent University, Faculty of Engineering and Architecture, Department of EMSME, Laboratory Soete, 9052 Zwijnaarde, Technologiepark 46, Belgium.

E-mail address: mojtaba.khayatazad@ugent.be (M. Khayatazad).

(MPC). Pidaparti et al. used an analysis based on wavelet transforms and fractals to classify the pits/cracks in the grayscale images of nickel aluminum bronze metal samples, (Pidaparti et al., 2010).

During the conversion of color images to grayscale, color as a piece of important information is lost, so several researchers have benefited from the color images. For instance, Lee et al. (2006) focus on the color information to distinguish the rust defects from the background. This study investigates images with tiny rust defects and a contrasting background. The spectrum of the rusted color is statistically determined in the Red-Green-Blue (RGB) color space. Chen et al. investigated 14 color spaces to find out the best one for the detection of corrosion in non-uniformly illuminated digital images, (Chen et al., 2009). Selecting a^*b^* as the best color configuration, they have shown the effectiveness of their method, adaptive ellipse approach. Khan et al. used color information to estimate subsea pipeline corrosion, (Khan et al., 2018). They developed an image restoration and enhancement algorithm which minimizes blurring effects and enhances color and contrast of the degraded underwater images.

Among the others, some researchers have used both texture and color features for corrosion detection. Petricca et al. used a python-based deep learning approach for automatic metal corrosion (rust) detection, (Petricca et al., 2016). They trained their model with more than 3500 images. Gibbons et al. applied the $L^*a^*b^*$ color space and Gabor texture features to train a Gaussian mixture model for corrosion detection in remanufacturing, (Gibbons et al., 2018). Both Medeiros et al. (2010) and Bonnin-Pascual and Ortiz (2014) utilize two weak classifiers to automatically detect corrosion in storage tanks, vessels and on pipelines. The first classifier, roughness, is measured by means of the energy property of the GLCM. The color is used as the second classifier and is examined in the Hue-Saturation-Intensity (HSI) and Hue-Saturation-Value (HSV) color space. To provide the system of a reference color spectrum, Medeiros et al. use a discriminant analysis while Bonnin-Pascual and Ortiz apply different filtering strategies on a trained Hue-Saturation (HS) histogram.

Due to the advantages of the simultaneous usage of texture and color features, in this paper, the proposed algorithm by Bonnin-Pascual and Ortiz (2014) is implemented and optimized for corrosion detection. In the following, first, an introduction to this algorithm is given (section 2)

and then the used dataset is introduced in section 3. The results of the corrosion detection framework are discussed in section 4; the detailed results are listed in the appendix. In section 5 the conclusions are presented.

2. Corrosion detection algorithm

From the visual inspection point of view, a corroded area has a surface rougher than a non-corroded one and its color looks like a hue between red and brown. So the algorithm developed by Bonnin-Pascual and Ortiz (2014), quantifies these two visual aspects to locate the corroded area in a given image, Fig. 1.

Since an image with small dimensions, but still showing important features, demands less time for the processing, resizing large and high quality images to smaller ones might be an option for an end-user. Therefore this algorithm starts with resizing images; the final horizontal size of the image is 256 pixels and the final vertical dimension is selected in such a way that the resized image has the same aspect ratio as the original one. It should be mentioned that the images within the database developed for this investigation have been acquired from a distance of 1 to 1,5 m with a Canon EOS 1100 and a Canon M10 digital camera with full resolution of 12,2 and 18 megapixels respectively. With these resolutions and distances, our study revealed that 256 pixels in one direction is sufficient for feature extraction.

Next, the main part of the algorithm first consists of a roughness analysis. The identified rough area is transferred as a candidate corroded region to the second step, i.e. the color step, for further investigation. In the color step, the color of the candidate area is compared with the predefined colors of corrosion. Finally, the outcome of this algorithm is a map showing the locations of detected corrosion.

In the following of this section, the roughness step, the color step and performance metrics are described in more detail. The performance metrics are defined to assess the performance of the algorithm.

2.1. Roughness step

A non-corroded surface has a quite uniform color distribution, Fig. 2-

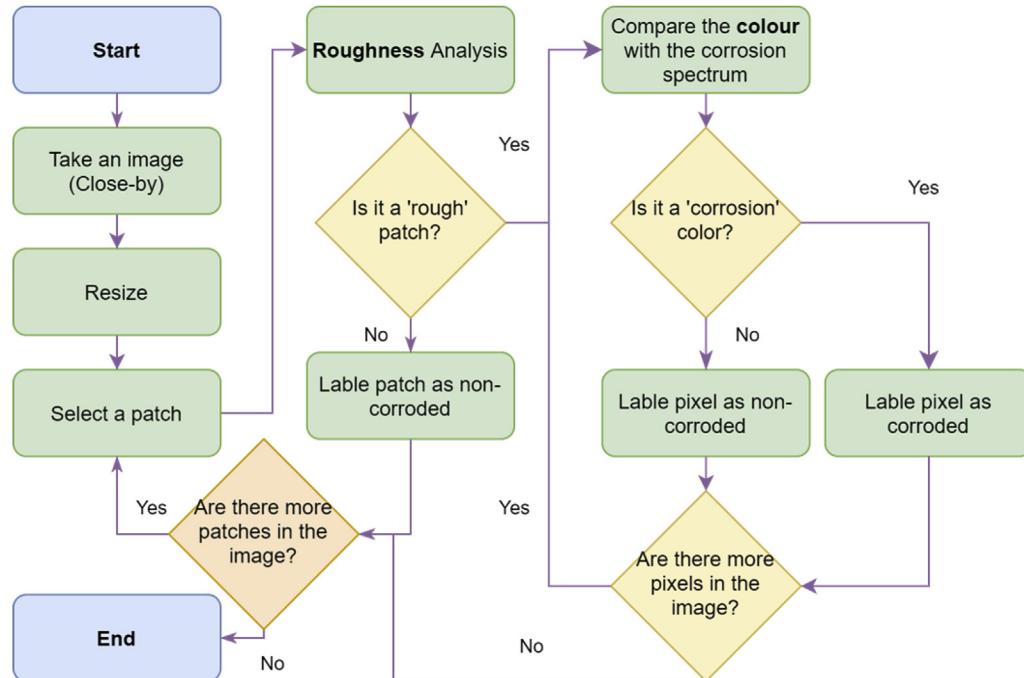


Fig. 1. The corrosion detection algorithm uses the visual aspects 'roughness' and 'color' of digital images to find corroded areas, (Bonnin-Pascual and Ortiz, 2014). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)



Fig. 2. Corroded surfaces (a) generally have a more non-uniform distribution of colors than non-corroded ones (b). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

b, but a corroded surface has a non-uniform distribution of corrosion colors, Fig. 2-a.

One way to quantify the color distribution of a portion of an image, hereafter called patch, is to measure its *uniformity* (Baraldi and Parmigiani, 1995). It should be mentioned that *uniformity* yields a value between 0 and 1. A value equal to 1 means that the investigated patch has a uniform color distribution which is interpreted as a non-corroded patch, and a value equal to 0 means that the patch has a non-uniform distribution of colors which might indicate the presence of corrosion. Equation (1) presents *uniformity*:

$$\text{uniformity} = \sum_{i,j} p(i,j)^2 \quad (1)$$

Where p is the *Gray Level Co-occurrence Matrix* (GLCM) which is explained hereafter. Actually, most of the texture analysis techniques make use of the grayscalepixels to investigate the spatial distribution of images (Feliciano et al., 2015). For this purpose, color images should be converted to a grayscale image; white and black in a color image are converted to white and black correspondingly and the other colors are converted to different shades of gray. In reality, there are numerous shades of gray but from a practical point of view, a gray spectrum can be divided into a limited number of levels, i.e. *gray level*, Fig. 3. After converting a color image to a grayscale one with predetermined gray levels, the GLCM is constructed. As an example, Fig. 4-a shows a grayscale patch and Fig. 4-b represents its corresponding gray levels. In this example, 8 gray levels are used, so for black and white pixels values equal to 0 and 7 respectively are considered. Because the used number of gray levels is 8, the produced GLCM is an 8×8 matrix, Fig. 4-c. Element $p(i,j)$ of this matrix quantifies how many times the gray level of i is in the neighborhood of the gray level j , Fig. 4-d. For defining the neighborhood, two parameters should be considered, i.e. direction and distance, Fig. 4-e. In this example, the direction is horizontal and the distance is 1 pixel.

To summarize, *uniformity* is calculated for every patch in the roughness step and then compared to a threshold. If the calculated *uniformity* is less than this threshold, the investigated patch is considered as a corroded patch.

2.2. Color step

By-products of steel corrosion at atmospheric conditions are shades of red, yellow and red-brown. So by quantifying corrosion colors and

comparing them with a reference color, one can make a classifier for corrosion detection. The first step for quantification is to select the appropriate color space. Based on our investigation, the HSV color space seems to be the most appropriate color space for describing colors related to corrosion. For example, Fig. 5-a shows a fully corroded steel plate. Depicting all observed colors in the RGB color space results in Fig. 5-b. Describing Fig. 5-a in the HSV color space will lead to Fig. 5-c. As can be seen, the plotted color spectrum in the RGB introduces some complexity in the sense that applying thresholds on the red, green and blue would include a lot more colors than the wanted rust color spectrum. On the other hand, the position of the spectrum in the HSV color space does allow to more efficiently apply thresholds on the hue, saturation and value. Hence, in this paper, HSV color space is selected for the color step.

As the digital images in our database are defined in the RGB color space, conversion from RGB to HSV is essential which can be easily done by the Matlab function *rgb2hsv*. Among hue (H), saturation (S), and value (V), V can just be used to prevent the well-known instabilities in the computation of hue and saturation when a color is close to white or black. In that case the pixel is classified as non-corroded (Bonnin-Pascual and Ortiz, 2014). Regarding H and S, one can apply two methods. The first method is to apply rough limitations on the H and S values, i.e. the threshold method. In this case the H and S values related to corrosion colors would be a rectangle in an H-S plane. The second method, i.e. histogram method, is more conservative and makes use of a normalized histogram of H and S values of corrosion colors and then applies a filter such as the two-dimensional Gaussian filter, Equation (2).

$$G(H, S, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \left(\frac{(H-\mu_H)^2}{\sigma^2} + \frac{(S-\mu_S)^2}{\sigma^2} \right)} \quad (2)$$

Since this histogram is a normalized one, applying a threshold means to filter out the low-probability combinations of H and S to represent a corrosion color. This method leads to a simple closed curve boundary in the H-S plane. It should be mentioned that σ is a parameter to determine the probability of each H-S combination. In other words, by increasing σ the diameter of the circular boundary increases because it increases the probability of the H-S combination.

2.3. Performance metric

To assess the performance of the corrosion detection algorithm for an arbitrary image, e.g. Fig. 6-a, the locations of corrosion on the under investigation image which are obtained manually, Fig. 6-b, and determined by the algorithm, Fig. 6-c, are required. This comparison can lead to 4 different situations:

- *True positive*, TP, which means a corroded area is classified as corroded,
- *True negative*, TN, which means a non-corroded area is classified as non-corroded,
- *False positive*, FP, which means a non-corroded area is classified as corroded,
- *False negative*, FN, which means a corroded area is classified as non-corroded.

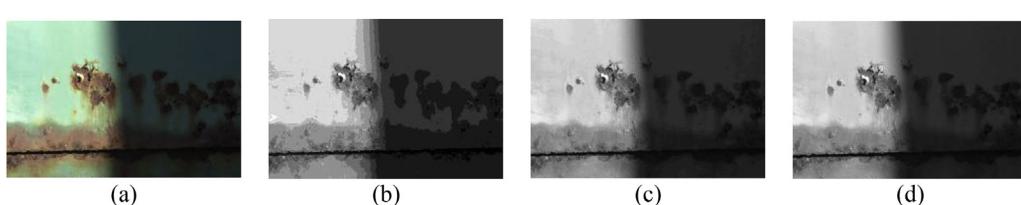


Fig. 3. The more gray levels are involved, the more information about the surface texture can be extracted, (a) the original color image, (b) 8 gray levels, (c) 32 gray levels and (d) 256 gray levels. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

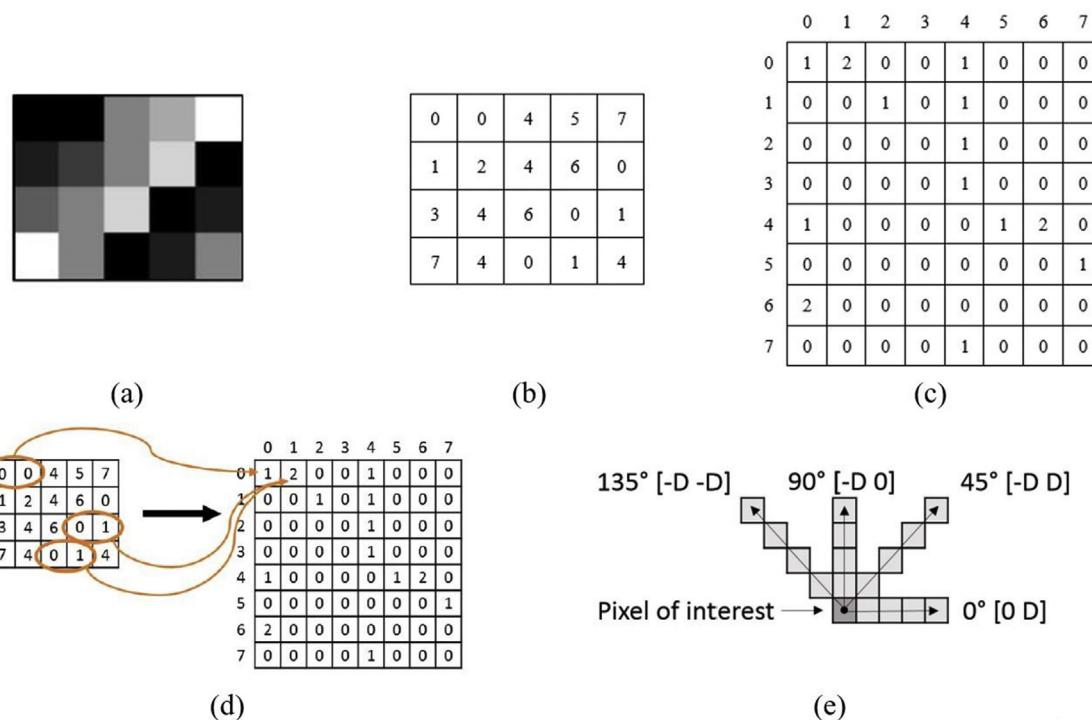


Fig. 4. A grayscale patch (a), its corresponding graylevels (b) and gray level co-occurrence matrix (c). (d) illustrate how the GLCM is constructed. (e) defines distance and direction for searching around a pixel.

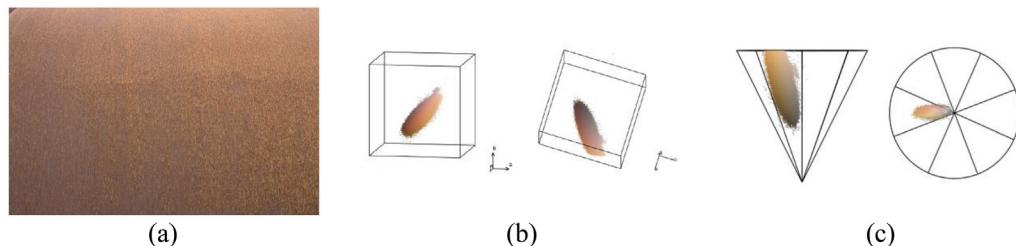


Fig. 5. The colors of a corroded steel plate (a) depicted in both RGB color space (b) and HSV color space (c). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

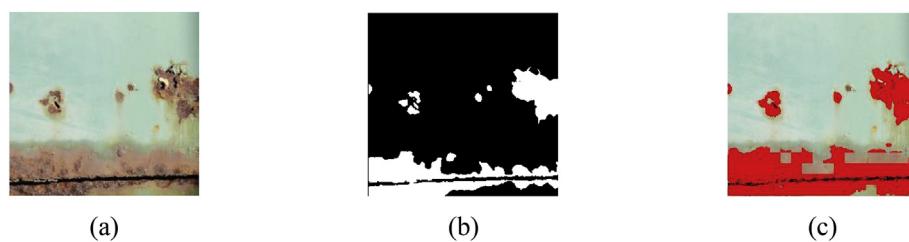


Fig. 6. The original image (a) and its corroded areas which are detected manually (b) and by the algorithm (c).

Based on these terms that are depicted in Fig. 7-a for a schematic patch, two performance metrics, precision and recall, can be defined as follows, Equations (3) and (4):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

In a perfect corrosion detection, the values for both precision and recall are equal to 1. Although having an algorithm which leads to a

precision and recall near 1 is ideal, this situation is unlikely in real applications. So one has to select one of these two values as the final performance metric. Since high precision does not guarantee a complete corrosion detection, Fig. 7-e, recall is considered as the performance metric in this paper, Fig. 7-d.

3. Dataset

The developed dataset is divided into a preliminary training set and a testing set. The training set was used solely to define an optimized HS-histogram. This dataset contains 8 full-resolution images of entirely

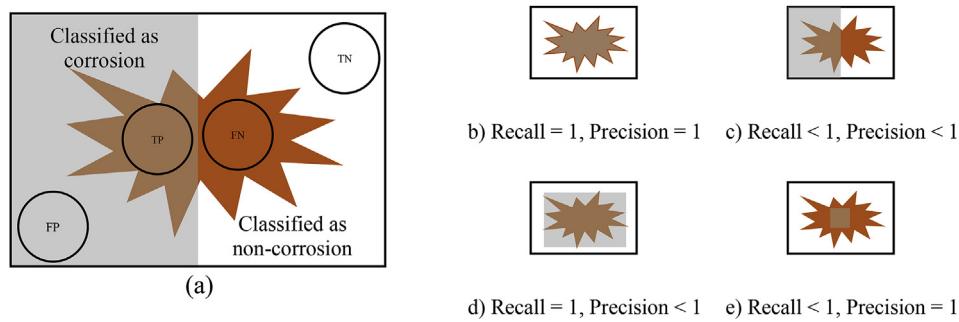


Fig. 7. Interpretation of precision and recall performance metrics.

corroded surfaces. The test dataset is meant to quantify the performance of the algorithm and to evaluate the influence of the parameters. This dataset is divided into three categories. The first category, A, contains images with uniform illumination and without misleading objects or colors. The second category, B, are images with non-uniform or bad illumination. And in the last category, C, images with misleading objects are investigated. In this case, rust stains and colors that are similar to rusty colors are included.

There are no concrete requirements or standardized rules about the content or dimension of a good dataset. The only advice that is given, is to introduce as much variety as possible to avoid creating a biased dataset (Piattini et al., 1999).

4. Results

To determine the influence of the parameters and eventually decide on the settings of the final algorithm, the parameters are evaluated one by one. When changing one parameter, the others adopt their reference values. The reference of the roughness step is based on the study of Bonnin-Pascual and Ortiz (2014). This involves a symmetric composition

of the GLCM with 32 gray-levels, only one direction (horizontal) and a distance of 5 pixels. On the uniformity a threshold of 0.1 is applied. During the training of the color step, 5 different values of the σ value of the Gaussian kernel (5, 10, 12, 15 and 20) and 5 different values of the probability threshold (2%, 5%, 10%, 15% and 20%) are combined to find their optimal combination for the histogram method. Wider ranges of H and S than those reported in (Bonnin-Pascual and Ortiz, 2014) are also examined for the threshold method. In the following sections, the results of the tests are illustrated and discussed.

4.1. Roughness step

Gray level: Gray level affects the detection because the variation and spatial distribution of the gray levels can be simplified to such an extent that it is no longer possible to properly recognize the rough surface. A low number of gray level leads to the misclassification of patches as smooth and thus already labels these as non-corroded. The influence on the performance of the number of gray levels for the entire dataset is given in Fig. 8-a. It shows that increasing the number of gray level has a major effect on the recall, at least for 8–64 gray levels. The monotonic

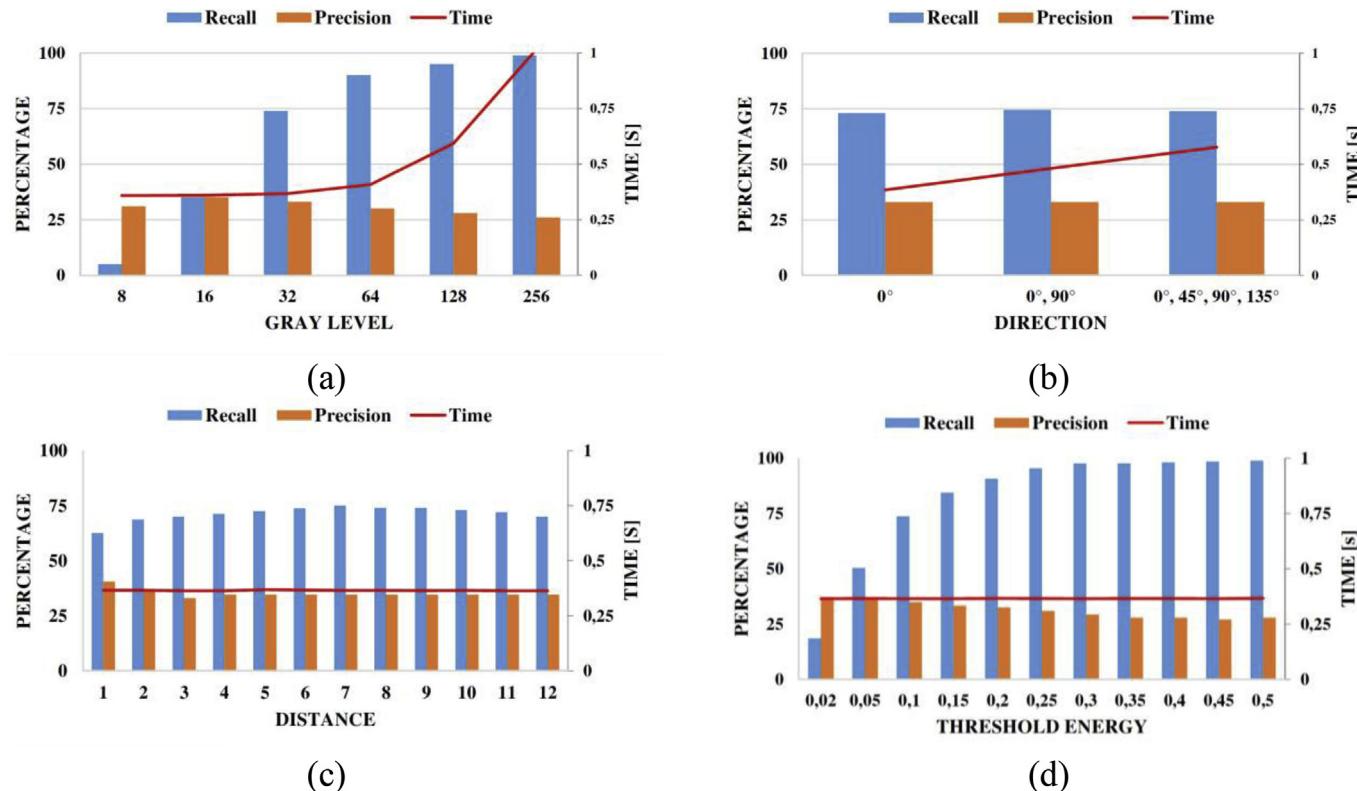


Fig. 8. The effect of (a) gray level, (b) direction, (c) distance, and (d) threshold uniformity on recall, precision and computation time for the entire dataset.

increasing recall already reaches around 95% for 64 gray levels. Further increasing the number of gray levels will improve that result, but less strong than before. The precision does not follow a monotonic pattern, but overall it can be stated that a higher number of gray levels reduces the precision. Beside the performance, the computing time is also affected. Involving more gray levels leads to larger dimensions of the GLCM and demands more time to calculate. The computation time values reported in Fig. 8, have been obtained using a PC with an Intel(R) Xeon(R) Gold 6152 CPU running at 2.10 GHz. The stated values represent the average time of performing the calculations for one image in the dataset.

Direction: The direction of composing the GLCM can either be a single direction or a combination of multiple directions. Fig. 8-b represents the results of the entire dataset when applying one, two or four directions. Even though the results for individual images slightly vary, considering multiple directions does not seem to deliver a surplus value for neither the precision nor the recall. Contrary, it increases the computing time with 50% for four directions compared to one direction.

Distance: The influence of the distance for composing the GLCM on the entire dataset is given in Fig. 8-c. When only taking a distance of 1 pixel into account, the precision reaches its maximum and the recall its minimum. It is hypothesized that the narrow frame of only one pixel is too small to conclude things about the spatial distribution. When increasing the distance, the mean value of precision for the datasets stays more or less constant. The recall obtains a maximum at a distance of 7 pixels. As can be seen in Fig. 8-c, the computing time is not affected by the distance.

Threshold uniformity: A low threshold is a translation of a strict judgment towards labeling a texture as ‘rough’. This means that together with an increasing threshold, fewer pixels are missed out but also more pixels are unnecessarily concluded as possibly corroded. Or in other words, the recall rises and the precision decreases, see Fig. 8-d. While the recall strongly increases between threshold values 0,02 and 0,15, the precision decreases less significantly. Finally, a threshold of 0,15 is chosen because the mean recall for all the datasets lies around or above 80%. This threshold also presents a precision of 33% which is more than the average precision of different thresholds (31%). Similar to the distance, threshold uniformity does not affect the computing time.

4.2. Color step

For defining the corrosion color spectrum, images of 8 fully corroded steel plates have been chosen, Fig. 9. To prevent the well-known instabilities for both threshold and histogram methods, the following limitations are applied:

- Exclude colors near black: Value < 50
- Exclude colors near white: Value > 230 and Saturation > 35



Fig. 9. 8 images of fully corroded steel plates for defining the corrosion color spectrum. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

In the threshold method, the corroded colors are those with $0 < \text{Hue} < 40$ or $230 < \text{Hue} < 256$ and $\text{Saturation} < 230$; these values were taken from (Bonnin-Pascual and Ortiz, 2014). In the histogram method, the hue-values in the matrix range from 1 to 256 but they actually represent a circle. The hue values of interest are located both at the beginning and at the end of the matrix columns, Fig. 10-a. When the filter (equation (2)) would immediately be applied, the hue values 1 and 256 are seen as items at a far distance, but in reality they are neighbors. This way, information on the sides gets lost after applying the filter. Therefore, the matrix columns are shifted so the area of interest is located in the middle of the matrix, Fig. 10-b. After applying the filter, Fig. 10-c, the probability threshold is applied, Fig. 10-d. Fig. 10-e shows the simple closed curve boundary in the H-S plane. Finally the matrix columns are shifted back to their original position as shown in Fig. 10-f.

In the threshold method, the algorithm yields a recall of 85% and precision of 55% over the entire dataset. The performance of the algorithm with a predefined HS-histogram is influenced by the parameters σ of the Gaussian filter and the applied probability threshold on the color spectrum. With increasing σ of the Gaussian filter, more colors are included in the corrosion color spectrum. This results in a monotonic reduction of the precision and a rise of the recall. An increasing threshold on the color spectrum results in an inverse effect. This means that for both the σ value and the probability threshold, a trade-off has to be made between precision and recall. During the analysis, the σ varied between 5 and 20 and the probability threshold between 2% and 20%

The histogram method with a probability threshold of 5% and σ equal to 12 revealed the best recall of 92% for the entire dataset. Because the main focus of this study lies on a good recall, this method has been selected for the final corrosion detection algorithm.

4.3. Combination of roughness and color steps

A summary of the final choices of the algorithm parameters is listed below. In the roughness step:

- 32 gray levels
- 1 direction (0°)
- Distance of 7 pixels
- Threshold on uniformity of 0,15

And for the color step:

- Gaussian filter with σ equal to 12
- Probability threshold on color spectrum of 5%

The benefit of combining both classifiers, roughness, and color, are illustrated in the first and second examples shown in Fig. 11. The first image contains a detail of a steel bridge with green paint. The roughness



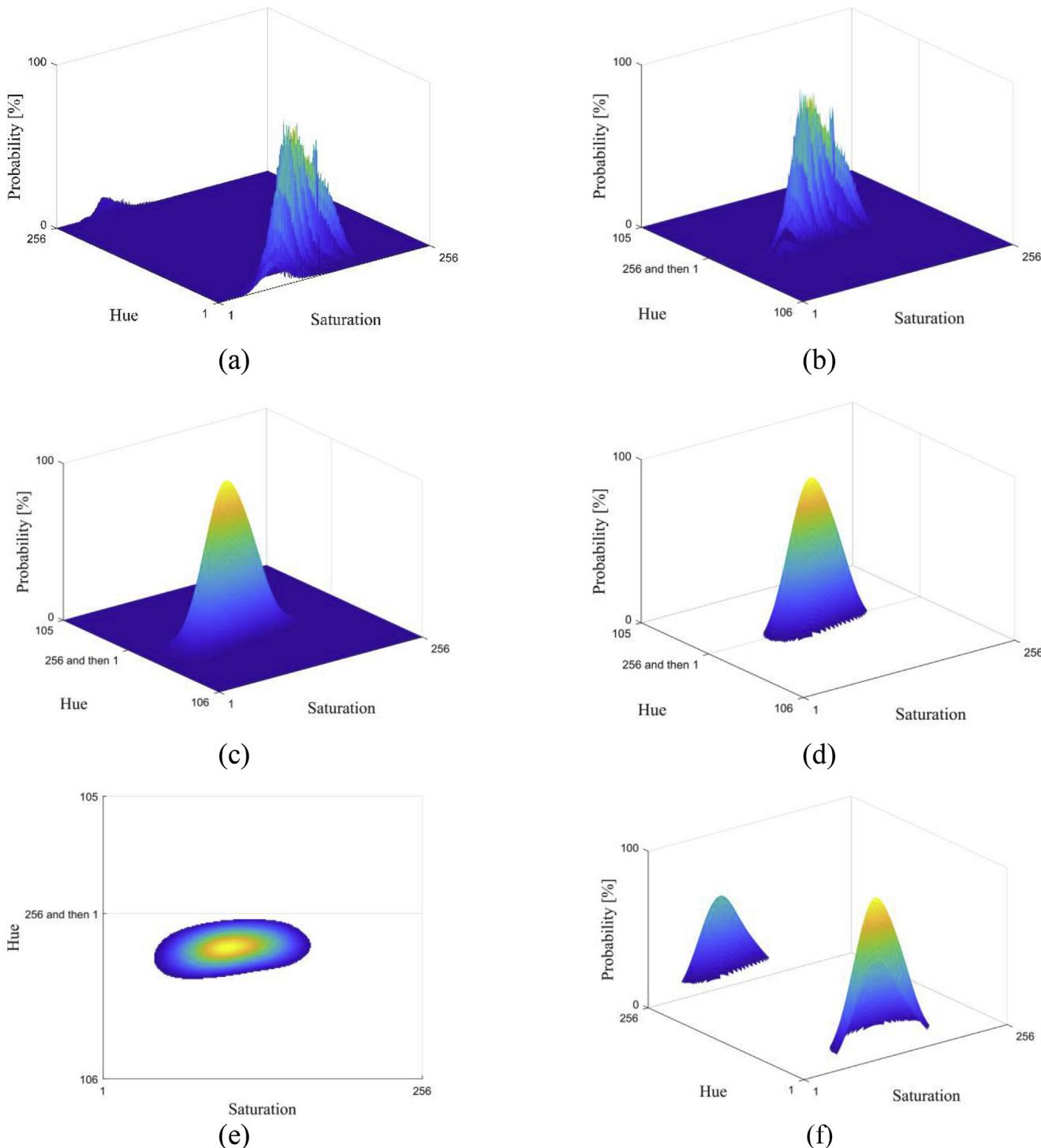


Fig. 10. The real histogram of the corroded colors (a), shifted to the middle of the H-S plane (b). After applying the Gaussian filter (c) and the probability threshold (d) a closed curve area is obtained in the H-S plane as a representative for corroded colors (e). For real applications, this histogram is again shifted to the original position (f). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

analysis appropriately determines almost the entire image as ‘rough’. Because not every rough surface implies corrosion, the color classification filters out the corroded area. Note that the bricks in the background (bottom part of image) are also labeled as corrosion because of their rough texture and a similar color.

The second example indicates that, both in the roughness classification and in the color classification, too many areas are labeled as corroded. The roughness analysis includes rougher parts of the white

surface and a part of the orange bar while the color analysis includes the rust stains and parts of the orange bar. However, by combining the two classifiers, the excessive indicated areas are filtered out and as a final result the corroded area is relatively accurate determined.

In the third example including a non-uniform illumination, the combination of roughness and color step does not perform better than the individual color analysis. The roughness analysis indicated properly the possible corroded areas at the properly illuminated side (left) but also

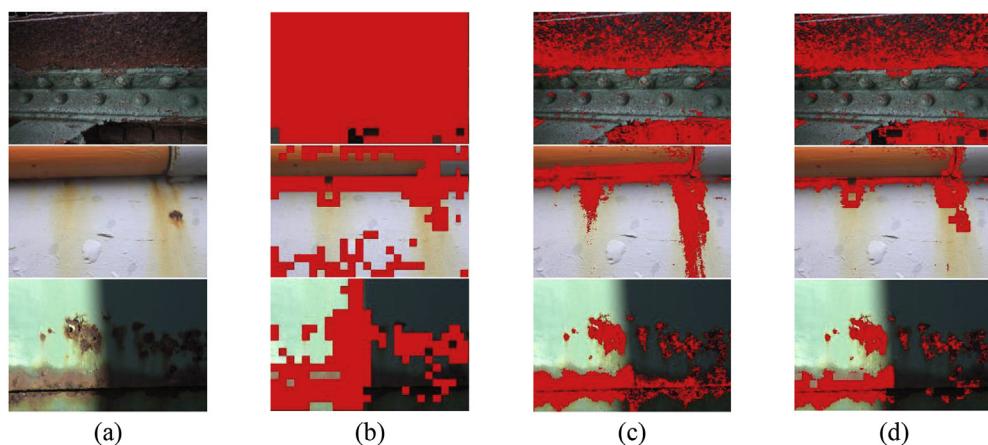


Fig. 11. Examples of the results of the two classifiers separately and together. Red indicates the pixels labeled as corroded. (a) original images, (b) result after roughness analysis, (c) result after color analysis and (d) result after roughness and color analysis. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

eliminated regions of interest at the under-exposed side (right). Even though the color analysis indicated quite accurately the corroded areas, the eliminated regions of interest are not labeled as corroded. This indicates that for specific cases, especially non-uniform illumination, this algorithm needs to be further improved.

The results of the entire dataset are given in the appendix.

5. Conclusion

The results confirm that the implemented algorithm is a promising tool for automated detection of corrosion by digital image analysis. Performance metrics precision and recall have been defined to assess the performance of the algorithm. The current algorithm does not deliver a performance of 100% on neither the recall nor the precision. This is a consequence of the rather simple classification method and the trade-off that has to be made at certain points. On top of that, it should be noted that a reference mask of corroded area on each image has been manually generated. This implies that a complete resemblance at the pixel level is almost impossible to achieve. When looking at the bigger picture, i.e. corrosion inspection of large industrial assets, it is not the intention to classify every pixel correctly but rather to rapidly screen structures and indicate corroded areas.

Preliminary findings show that the developed algorithm can

efficiently find the corroded areas in digital images of a wide variety of corroded steel components. Among these 31 investigated images, the corroded area in those images with uniform illumination and without misleading objects are categorized with an average recall of 85%. The average recall values for images under non-uniform illumination and images containing miss-leading objects are 72% and 69% respectively. Non-uniformly illuminated images and images containing misleading objects are challenging for the current image processing algorithm. The authors are working on these issues to further improve the accuracy and robustness of this algorithm.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors acknowledge the financial support of Vlaio through the SafeLife project (project number 179P04718W) and also the support by SIM (Strategic Initiative Materials in Flanders) and the IBN Offshore Energy.

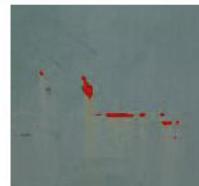
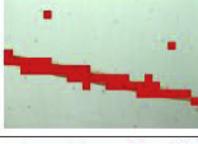
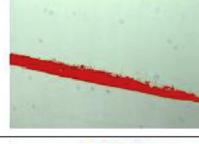
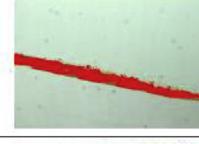
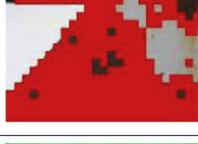
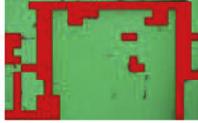
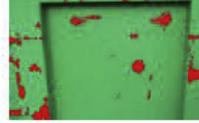
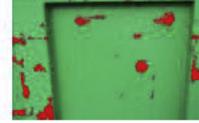
Appendix

Table 1 shows the results of the individual color and roughness steps, and of their combination in the corrosion detection algorithm. Red indicates the pixels labeled as corroded.

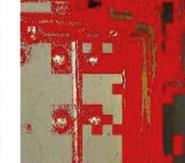
As shown in **Table 1**, recall and precision vary widely for the different photographs. This is because these images originated from a very diverse dataset (different applications, different types of corrosion, different noise contributions). In applications where the images of a structure are quite similar with respect to color and background, stable performance is achievable. Such stable behaviour is for example reported in (Kim et al., 2006) that applies an artificial intelligence based algorithm to samples of a dataset having similar features.

Table 1

The results of roughness step, color step and corrosion detection algorithm for an extended dataset.

Image Info	Original Image	Roughness step	Color step	Combination
No: A_001 Precision = 66 % Recall = 95 %				
No: A_002 Precision = 59 % Recall = 99 %				
No: A_003 Precision = 69 % Recall = 37 %				
No: A_004 Precision = 95 % Recall = 99 %				
No: A_005 Precision = 71 % Recall = 89 %				
No: A_006 Precision = 63 % Recall = 85 %				
No: A_007 Precision = 89 % Recall = 70 %				
No: A_008 Precision = 79 % Recall = 90 %				
No: A_009 Precision = 34 % Recall = 91 %				
No: A_010 Precision = 99 % Recall = 71 %				

No: A_011 Precision = 61 % Recall = 94 %				
No: A_012 Precision = 89 % Recall = 92 %				
No: A_013 Precision = 98 % Recall = 85 %				
No: A_014 Precision = 71 % Recall = 95 %				
No: B_015 Precision = 57 % Recall = 56 %				
No: B_016 Precision = 91 % Recall = 77 %				
No: B_017 Precision = 57 % Recall = 97 %				
No: B_018 Precision = 65 % Recall = 72 %				
No: B_019 Precision = 64 % Recall = 86 %				
No: B_020 Precision = 58 % Recall = 58 %				
No: B_021 Precision = 64 % Recall = 77 %				

No: B_022 Precision = 30 % Recall = 50 %				
No: C_023 Precision = 46 % Recall = 43 %				
No: C_024 Precision = 0 % Recall = 0 %				
No: C_025 Precision = 7 % Recall = 77 %				
No: C_026 Precision = 3 % Recall = 95 %				
No: C_027 Precision = 72 % Recall = 91 %				
No: C_028 Precision = 25 % Recall = 92 %				
No: C_029 Precision = 96 % Recall = 72 %				
No: C_030 Precision = 62 % Recall = 81 %				
No: C_031 Precision = 5 % Recall = 72 %				

References

- AbdelRazig, Y.A., 1999. Construction Quality Assessment: a Hybrid Decision Support Model Using Image Processing and Neural Learning for Defect Recognition. Purdue University.
- Agdas, D., Asce, M., Rice, J.A., Asce, M., Martinez, J.R., Lasa, I.R., 2016. Comparison of Visual Inspection and Structural-Health Monitoring as Bridge Condition Assessment Methods. vol. 30, pp. 1–10 no. 3.
- Alkanhal, T.A., 2014. ‘Image Processing Techniques Applied for Pitting Corrosion Analysis’, pp. 385–391 no. June 2014.
- Baraldi, A., Parmiggiani, F., 1995. An investigation of the textural characteristics associated with Gray Level Co-occurrence Matrix statistical parameters. *Geosci. Remote Sensing, IEEE Trans.* 33, 293–304.
- Bonnin-Pascual, F., Ortiz, A., 2014. ‘Corrosion Detection for Automated Visual Inspection,’ in *Developments In Corrosion Protection*. INTECH, pp. 619–632.
- Chen, P.H., Chang, L.M., 2003. Artificial intelligence application to bridge painting assessment. *Autom. ConStruct.* 12 (4), 431–445.
- Chen, P.-H., Chang, Y.-C., Chang, L.-M., Doerschuk, P.C., 2002. Application of multiresolution pattern classification to steel bridge coating assessment. *J. Comput. Civ. Eng.* 16 (4), 244–251.
- Chen, P.-H., Yang, Y.-C., Chang, L.-M., Aug. 2009. Automated bridge coating defect recognition using adaptive ellipse approach. *Autom. ConStruct.* 18 (5), 632–643.
- Enikeev, M., Gubaydullin, I., Maleeva, M., 2017. Analysis of corrosion process development on metals by means of computer vision. *Eng. J.* 21 (4), 183–192.
- Feliciano, F.F., Leta, F.R., Mainier, F.B., 2015. Texture digital analysis for corrosion monitoring. *Corrosion Sci.* 93, 138–147. January.
- Gibbons, T., Pierce, G., Worden, K., Antoniadou, I., 2018. A Gaussian mixture model for automated corrosion detection in remanufacturing. *Adv. Transdiscipl. Eng.* 8, 63–68.
- Khan, A., Ali, S.S.A., Anwer, A., Adil, S.H., Meriaudeau, F., 2018. Subsea pipeline corrosion estimation by restoring and enhancing degraded underwater images. *IEEE Access* 6, 40585–40601.
- Kim, H.-J., Shrestha, J., Kim, H.-N., Jo, G.-S., 2006. User action based adaptive learning with weighted bayesian classification for filtering spam mail. *Lect. Notes Comput. Sci.* 790–798. *Adv. Artif. Intell. AI 2006*.
- Koch, G., Varney, J., Thopson, N., Moghissi, O., Gould, M., Payer, J., 2016. International Measures of Prevention , Application , and Economics of Corrosion Technologies Study. *NACE Int.*, pp. 1–216
- Lee, S., Chang, L.-M., Chen, P.-H., 2005. Performance Comparison of Bridge Coating Defect Recognition Methods, vol. 61. *Corrosion*.
- Lee, S., Chang, L.M., Skibniewski, M., 2006. Automated recognition of surface defects using digital color image processing. *Autom. ConStruct.* 15 (4), 540–549.
- Medeiros, F.N.S., Ramalho, G.L.B., Bento, M.P., Medeiros, L.C.L., 2010. On the evaluation of texture and color features for nondestructive corrosion detection. *EURASIP J. Appl. Signal Process.* 2010.
- Petricca, L., Moss, T., Figueroa, G., Broen, S., 2016. Corrosion detection using A.I : a comparison of standard computer vision techniques and deep learning model. *Comput. Sci. Inf. Technol. (CS IT)* 91–99.
- Petrovic, Z., 2016. Catastrophes caused by corrosion. *Vojnoteh. Glas.* 64 (4), 1048–1064.
- Piattini, M., et al., 1999. Data Quality and Database Design 1.
- Pidaparti, R.M., Aghazadeh, B.S., Whitfield, A., Rao, A.S., Mercier, G.P., 2010. Classification of corrosion defects in NiAl bronze through image analysis. *Corrosion Sci.* 52 (11), 3661–3666.