Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific research

Kasdi Merbah University,Ouargla

Faculty of new information and communication technologies

Department of Computer Science and Information Technology

MASTER’S THESIS

Submitted in partial fulfillment of the requirements for the degree of

Master of Science

A new scheme of ensemble deep models for corrosion detection in oil and gas structures

Presented by:

Abderrahmane Behmene

Jury members

President	Doctor Belal Khaldi
Examiner	Professor Mohammed Lamine Kherfi
Supervisor	Doctor Oussama Aiadi

Academic Year: 2019-2020

To my father and mother,
To all my family,
To all my friends and loved ones,
I dedicate this humble work.

Acknowledgment

The completion of this undertaking could not have been possible without the participation and assistance of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged.

I would like to express my gratitude to my supervisor Doctor Oussama Aiadi, for his huge support, valuable time and engagement through the learning process of this master thesis. He has spared no effort to guide me during these difficult times of the pandemic. Furthermore, I am keen on thanking Professor Mohammed Lamine Kherfi for his precious lectures,

which introduced me to the marvelous field of Data

science and Machine learning.

Also, I would like to thank Doctor Belal Khaldi, who has willingly shared his precious time and advice.

Last but not least, I want to thank my company SH-FCP, for the support that gave to me during the research. I wish this modest study could bring an addition in the field of Oil and Gas.

Abstract

Automatic visual inspection has gained a lot of attention last decades, thanks to the advancement in computer vision techniques such as deep learning. In oil and gas industry, keeping facilities in good serviceability is vital for the safety, environmental, production and cost . Corrosion is one of the threats that obviously reduces the mechanical proprieties of structures. Thus, without effective management of corrosion of the structures, rust of metal leads to catastrophic accidents, causing loss of lives and damages in environment and serious economic destruction. In this thesis, we present a new method for corrosion detection based on convolution neural networks (CNN). In particular, the major contributions of this thesis are: 1) we investigate the performance of multiple deep nets for the task of corrosion detection and we propose an ensemble scheme to strengthen the individual performance of the different deep nets, 2) we present a comprehensive literature review of corrosion detection works, 3) Introduce a new customized self-made dataset which is made up of corroded/non-corroded petroleum pipeline images, 4) we conduct an experimental evaluation for several methods from the state of the art as well as different handcrafted features, and 5) we investigate the effect of varying color space on the system performance. Experimental results proved the effectiveness of the proposed method. The proposed ensemble CNNs has outperformed several relevant state of the art methods as well as multiple handcrafted features.

Keywords: Corrosion detection,Oil and Gas, Image processing,Deep learning,CNN, ensemble models.

Resumé

L’inspection visuelle automatique a gagné du terrain ces dernières décennies et ce grâce aux avancées faites dans le domaine des techniques de vision par ordinateur tel que l’apprentissage profond. Dans le secteur pétrolier et gazier, tenir les installations en bon état de fonctionnement est devenu crucial pour la sécurité, l’environnement, la production et la gestion des couts. L’effet de la corrosion représente l’une des menaces apparentes qui agit manifestement sur les propriétés mécaniques des structures. D’ici, en l’absence de gestion efficace de la corrosion de structures, la rouille des métaux peut provoquer de grave accident causant des pertes en vies humaines, des impacts négatifs sur l’environnement et des répercussions sérieuses sur l’économie. Dans cette thèse, Nous présentons une nouvelle méthode de détection de corrosion basée sur les réseaux de neurones convolutifs (CNN). En particulier, les contributions majeures de cette thèse sont: 1) étudier les performances de plusieurs réseaux profonds pour la tâche de détection de corrosion et nous proposons un schème d'ensemble pour renforcer les performances individuelles des différents réseaux profonds, 2) présenter un revue de la littérature des travaux de détection de corrosion, 3) Introduire une base de données personnalisés et improvisé, composée d'images de pipelines de pétrole corrodées / non corrodées, 4) Mener une évaluation expérimentale de plusieurs méthodes de l'état de l'art ainsi que différentes fonctionnalités artisanales, et 5) Investiguer l'effet de la variation de l'espace colorimétrique sur les performances du système. Les résultats expérimentaux ont prouvés l'efficacité de la méthode proposée. L'ensemble proposé CNNs a surpassé plusieurs méthodes pertinentes de pointe ainsi que de multiples fonctionnalités artisanales.

Mots clés : Détection de Corrosion , Industrie pétrolière, Apprentissage profond, traitement d’image , Réseau neuronal convolutif, ensemble de modèles .

ملخص

جلبت المراقبة البصرية عبر الآلة في العقود الأخيرة انتباه الكثير من الدارسيين والفضل يعود الى التقدم الكبير الذي عرفته الرؤية الآلية خاصة الدفع الكبير الذي أمده التعلم العميق للمجال في السنوات القليلة الماضية. في مجال البترول والغاز تعتبر المحافظة على المنشآت الصناعية حيويا بالنسبة اللأمن البيئي والسلامة فضلا عن الانتاج ومحاولة تخفيض تكاليفها . تصدؤ المعادن يعتبر إحدى هذ المخاطر التي تهدد بوضوح ميكانيكية وصلابة المعدات ولذلك من دون كفاءة تسيير عالية للمنشآت ومراقبتها دوريا , قد يؤدي هذا الى حوادث في الارواح والعتاد ومنه كوارث اقتصادية وبيئية.

في هذه الدراسة قدمنا طريقة جديدة لاكتشاف التصدؤ تعتمد على الشبكات العصبونية الالتفافية , على وجه الخصوص, المساهمات الكبرى لهذا العمل هي :1) دراسة أداء العديد من الشبكات العصبونية الالتفافية الموجهة لاكتشاف التصدؤ واقتراح نموذج موحد لتقوية أداء الشبكات الفردية العميقة ,2) تقديم مراجعة أدبيات للأعمال المنجزة على اكتشاف التصدوء 3) انشاء قاعدة بيانات جديدة ومنشئة من الصفر , تتكون من صور معدات البترول وأنابيبها مقسمة الى صدأة والى غير صدأة .4) القيام بتقييم عديد الطرق من الحالة الفنية التي لدينا , منها استعمال بعض تقنيات معالجة الصورة التي تعتمد الميزات اليدوية. 5)التحقق من تأثير معرفات الألوان على أداء النظام . نتائج التجارب أكدت فعالية النموذج المقترح بما أن توحيد مختلف النماذج قد تخطى في الأداء عديد الطرق في الحالة الفنية وبالتأكيد حتى تلك التي تعتمد المميزات اليدوية .

كلمات مفتاحية : الكشف عن التصدؤ , البترول والغاز, التعلم العميق ,معالجة الصورة,الشبكات العصبونية الالتفافية , توحيد النماذج .

Table of Contents

LIST OF FIGURES

LIST OF TABLES

GENERAL INTODUCTION

1 Introduction

Automated defect detection in computer vision has become increasingly area of research, and this has affected highly and directly on the application domain of visual inspection. Corrosion is one of the application that has gained a lot of attention in the last decade, many attempts have been made to replace conventional methods by other intelligent systems using CCTV , robots , drones or unmanned vehicles.

Corrosion is a natural occurring phenomenon commonly defined as the deterioration of a substance (usually a metal) or its physical properties because of a reaction with its environment. It mostly occurred in the highly humid environments ,in addition of other corrosion causes. Corrosion of oil structures is a significant issue, as inspecting oil and gas structures and facilities requires a great deal of time, a high cost and attention, and sometimes becoming a tedious task and labor-intensive.

Metal rust can bring failure to structures; it occurs when no longer able to withstand stresses imposed during exploitation. Furthermore, corrosion engenders damages to the environment and in many times causes casualties.

NACE -International, -The Worldwide Corrosion Authority- estimates the global cost of corrosion in 2013 to be US$2.5 trillion equivalent to approximately 3.4 percent of the international Gross Domestic Product (GDP)[1].

The only and simple way to prevent corrosion is to mitigate it by using a corrosion resistant metal such as aluminum or stainless steel, however, inspecting and monitoring oil and gas facilities is a difficult and time consuming job. Detecting corroded structures in early stage can bring down the rate of corrosion and mitigate the metal corrosion, and allows replacing extremely corroded parts if necessary.

Damage from corrosion often leads to failures in oil and Gas plants and equipment, which interrupt facilities such as refineries and CPFs (Central Processing Facilities) operations and cerate safety hazards. The existence as well as the degree of damage is dependent on the particular process operating conditions and contaminants present in the process stream.

Billions of dollars are spent every year on corrosion-related problems that could have been eliminated or reduced by applying corrosion fighting fundamentals.

WCO Shenyang Declaration, in 24 April 2019[2] “The annual cost of corrosion worldwide exceeds the cost of all natural disasters” , “corrosion is a “cancer” of industry is difficult to solve”, “results of Corrosion also pollute environment and affect our health”[3].

Nowadays , using a visual intelligent system at several sites and facilities for operations surveillance and security , is crucial for the safety of the structures . Thus, will identify shortcomings instantly -in real time- , predict failures and reduce the routine site technician’s visits specially in severe weather conditions . Eventually, the deployed system will reduce the cost of production on different aspect.

Monitoring Oil and Gas structures , has became one of the most important operations to make production more efficient. Moreover, using remote control system would reduce shuttles that has traditionally involved the high costs associated with labor, transportation, and administration. It is often felt that the extreme weather condition makes it harder to technician to visit facilities periodically , especially in the great Sahara desert where the heat reaches above the fifty degree Celsius in summer.

2 Problematic

Inspecting Oil structures with the conventional approaches such as magnetic flux leakage, ultrasonic testing, or external manual assessment, these methods have drawbacks and limitations, for instance equipment cost, restricted area of inspection and inability to detect small pitting considering the large surface of oil and gas facilities needed to be perambulated, and often are situated in different far locations. Therefore, intelligent vision is an alternative solution with effective results and low cost. Hence, it is crucial to use remote monitoring by cameras CCTV system, unmanned vehicles or robot systems. These methods can increase the diagnostic speed and quality and reduce the costs associated with inspections.

In Algeria , an explosion flattened a large part of the Algerian port of Skikda, in 20 January 2004 ,left 29 victims , and 70 injuries and almost complete demolition of the complex .one of causes was the extreme corrosion of some lines[4].

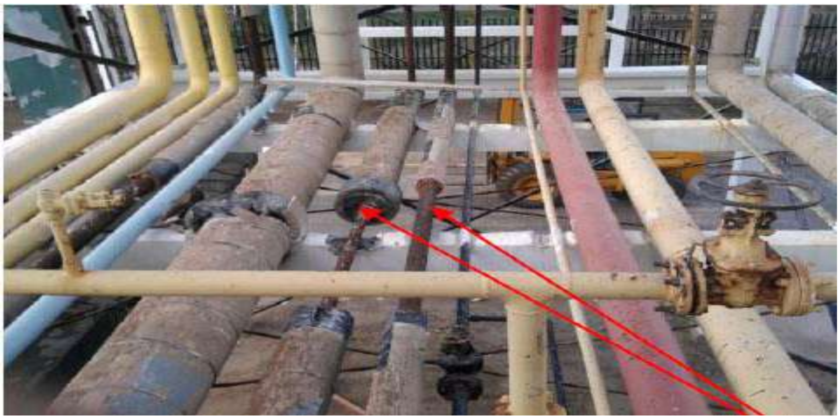


Fig1: Corrosion of lines in GL1K plant Skikda,Algeria 2004 ,before the blast occurred in January 20, 2004[5].

The only and simple way to prevent corrosion is to mitigate it by using a corrosion resistant metal such as aluminum or stainless steel. However, inspecting and monitoring oil and gas facilities is hard job. Visual inspection is one important operation that must be performed periodically by well-trained technicians, corrosion departments or similar entities. Inspections are often carried out manually, sometimes in unsafe conditions.

Inspecting oil and gas structures and facilities requires a lot of time, cost, and very high attention, sometimes becoming a tedious task. For instance, a large facility such as CPF, with some hectares of surface, and thousands of equipment and lines , recognizing the corrosion by conventional methods becomes a hard and hazard job for engineers and technicians specially in severe whether conditions .

Most of Oil and gas companies have multiple well sites and often situated hundred miles away from each other, therefore, it is crucial to use remote monitoring by cameras CCTV system, drones or unmanned vehicles to monitor the corrosion remotely.

Now, using a visual intelligent systems at different sites and facilities for operations Surveillance and security. Thus, will identify shortcomings instantly -in real time- , predict Failures and reduce the routine well site technician’s visits whether is weather Condition or time. Eventually, the deployed system will reduce the cost of production on different aspects.

3 Overview on the relevant works

Talking about the automated corrosion detection, the scientific literature of computer vision comprises a few works. However, after detailed review of the found papers, they can be mainly categorized by the approach used to address ferrous rust, we have learned that mainly two approaches have been used to tackle the problem, first by handcrafted extractions techniques combined with some classifiers algorithms and the second is deep learning.

The first approach involves image processing techniques, such as color spaces and histograms to analyses color and other works rely on feature extraction for instance texture, the latter is robust since the corrosion has no specific shape or color but the corroded regions have roughness in texture. Several works used GLCM and GRLM to extract texture roughness from the images. Eventually, to classify vectors, Machine learning is employed. Among many ML algorithms SVM is most common used due to its effectiveness in high dimensional spaces. It is effective in cases where the number of dimensions is greater than the number of samples.

For example, Mohammadreza Motamedi *et al* [6] their main aim was to use the Digital Image processing techniques including image smoothing, point processing, morphology, filtering and edge detection and other operations that can be used for pipeline networks which results in related defects that have been recognized in each image.

Venkatasainath Bondada et al [7], used threshold on mean saturation value of an image to measure the corrosion regions and eventually locate the centers of densely corroded area.

Gang Ji et al [8] tried to improve the corrosion detection by using the improved watershed segmentation method to eliminate over-segmentation, this leads to increase the accuracy detection of the coating material corroded defects.

On the other hand, computer vision has known great forward steps after the raising of the deep learning in 2009 , when Stanford's Fei-Fei Li created ImageNet[5]. This massive training dataset made it easier than ever for researchers to develop computer vision algorithms and directly lead to similar paradigms for natural language processing and other cornerstone of Deep learning that we take for granted now.

, M J Jiménez-Come et al [9] worked on different models based on artificial neural networks (ANNs), support vector machines (SVMs), classification tree (CT) and k-nearest neighbors (KNN) these algorithms were presented to develop an automatic way to predict pitting corrosion.

W.T. Nash1 et al [10] tried to approach the human level and getting accurate results by training of a deep learning model.

Luca Petricca et al[11] presented a comparison between classical computer vision techniques and deep learning for corrosion detection, they deduced that deep Learning model performs better in a real case scenario.

Yuan Yaoa et al [12] used CNN for hull structural plate corrosion damage detection and recognition, a new method for hull structural plate corrosion damage detection and recognition has been proposed.

Deegan J Atha et [13] presented an experimental evaluation of four deep networks, including VGG16 and ZFNet and two other customized models , to assess corrosion in metallic surfaces. Authors have also investigated the effect of varying color space and window size on the performance of deep networks.

Deep learning is powerful tool for image classification tasks ,However , it requires a large amount of data , in order to perform better than other techniques ,otherwise, the process will result the phenomenon over-fitting and under-fitting .

4 Motivation

Besides, the works on corrosion detection are not ample, more focus on the field should be given . therefore, yet still many aspects in the research domain need to be highlighted .

We can summarize our contribution in these points:

- Evaluate image-based processing approaches and compare them with deep learning, can bridge the gap in the literature.
- The most of studies focused on the pipelines, so that ,we created a new dataset that englobes images taken from oil and gas plants and equipment .
- Image processing techniques could be efficient in certain cases, but it shows obvious limit when it is dealt with real world problems, any variation in contrast or illumination would lead to false prediction.
- Exploring more deep learning techniques and ensemble models for the first time in corrosion detection can also give an addition to the existed methods proposed previously.
- Analyzing the color space and widows size effect on the performance , lastly ,a sliding window is used to recognize regions in new introduced images , these images varied in illumination , contrast and which can prove how powerful is the proposed model .

5 Contribution

Many works have studied the corrosion whether with image processing or with the deep learning no study has compared both approaches, below you find some highlights on the contribution:

- An extensive comparative study, the latter experimented many methods starting from image-based processing techniques such GLCM ,GRLM, LBP,HOG and combine them with different classifiers, ending by deep learning and its techniques ,mainly, we have used convolutional neural networks to classify the corroded images from the non corrosive ones
- Introduce a new dataset from scratch is not an easy task , taking high quality images and split them into small piece , and classify them to desired classes , and finally , remove the noisy samples and balance the classes to avoid under-fitting or overfitting . the dataset consists of 10000 images of 28x28 RGB images
- To build more balanced and robust model, we have proposed a new deep scheme the ensemble CNNs and analyze the two approaches . The first is a model average ensemble consists of multiple neural network models, and each model is trained with the same dataset images .When classifying an input image, the data is passed to each network, and it classifies the image independently. Final prediction is obtained by the combination of all the predictions from these models by averaging. The second is combining features of pre-trained models and connect them in fully connect layer.
- We investigated the effect of color spaces on the deep learning's performance along with model's hyperparameters. Also, we tried out different window sizes for our sliding window algorithm.

6 Thesis structure

The thesis is organized into four chapters in addition to the first one, which presents the general introduction. Below you find each chapter with its content:

- The first chapter is dedicated to introduce the background of image processing by exploring the different color spaces and histograms , also , we define filters which is considered the backbone of feature extraction, before we limp on the some texture and contours descriptors such as GLCM , Gabor and HOG ..etc.
- The second chapter is designed to survey the previous works in the literature, and discuss the methods, as long as it concerns the corrosion detection or recognition. Since we have two approaches , we have seen the state of art of each and come back with some critics and observation .

- We consecrated The third chapter to bring up the proposed methods and explain them in details, we also allocated a space to the image-based processing methods that are found in the literature. The ensemble learning will take further explanation as it yields the best results.
- The last chapter will be devoted to experimental setup and results , we describe our dataset and define the performance matrix used for result , we review the experiments we conducted in order to prove the effectiveness of our approach .

At the end of the thesis, we draw the conclusions on the study, highlight some perspectives and plans, and give concrete solutions that can tackle the corrosion in real world.

Chapter I. WORK BACKGROUND

I.1 Introduction

Vision plays a vital role in our life, by allowing us to interact with the environment in an effective way. Whereas, the Machine Vision aim is to endow computing gadgets, and more generally intelligent systems, with visual capabilities in order to deal with different situations. In last decades, with the hardware evolution ,as we are living a new golden age of computer architecture with the raising of new CPU generations such as GPU and TPU . These elements lead a continuous adaptation and optimization of the usual visual processing techniques, such as ones developed in Computer Vision and Image Processing.

Computer vision solutions and applications are being developed increasingly to supervise production processes in order to ensure their correct operation. Until now, most of them could only be properly overseen by humans. Industry, and in particular corrosion detection, is one of the areas that provides a relatively broad spectrum of possible applications for computer vision solutions. In the past, most methods focused on image processing and delivery of results in the most readable for analysis.

This chapter will focus on handcrafted features related to visual processing which is the backbone of deep learning convolution process. Recalling the basics of image and its processing from image features histograms, to filters, edge detecting and contours.

I.2 Computer vision

Computer vision refers to an artificial intelligence technique used to analyze images captured by equipment such as a camera. Concretely, computer vision is presented as an AI-based tool capable of recognizing an image, understanding it, and processing the resulting information. For many, computer vision is the AI equivalent of human eyes and our brains' ability to process and analyze perceived images. One of the main goals of computer vision is the reproduction of human vision by computers. Today, there are many fields of application, for instance the automotive sector, with the emergence of autonomous cars capable of recognizing road images.

I.3 Image

An image (originated from Latin word : imago) is an artifact that depicts the visual perception, such as a photograph . A digital image migh represent a cartoon, a face, a map or product . Technically , it is 2-Dimentional representation of the visible light spectrum, it has a multitudes of descriptive such as format (JPG,TIF,PNG …),type(vectorial , matricial), resolution(High, medium, low) and characteristics such color (RGB,BGR,Grayscale) , form and texture.

I.4 Color Image

Color image is interpreted as 3D layers of values with width, height, and depth. The depth is the number of layer-color used to form other colors. Most color images used RGB that's mean that they are represented by combinations of only 3 colors: red, green, and blue values. therefore , for RGB images, the depth is three layers.

Gonzalez and wood [14], defined the image as follow:

An image may be defined as a two-dimensional function, f (x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point.

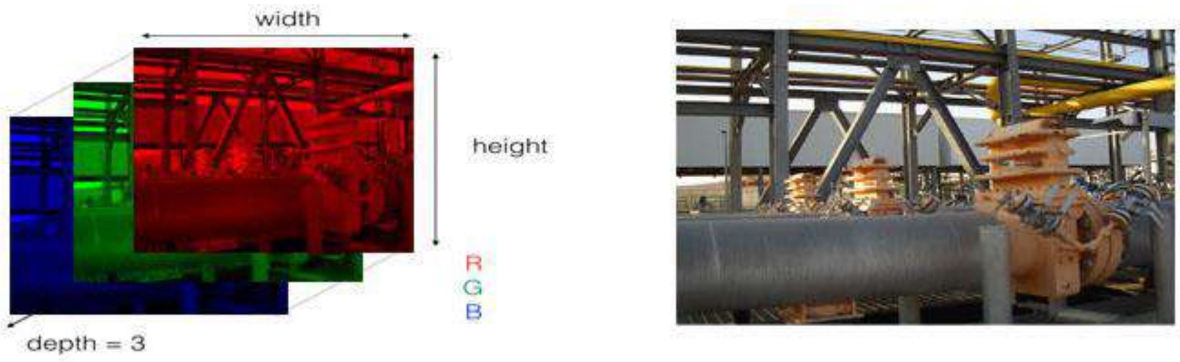


Fig2. RGB layers of equipment image I1

I.4.1 Images as Numerical Data

Every pixel in an image is just a numerical value and, we can also change these pixel values. We can multiply every single one by a scalar to change how bright the image is, we can shift each pixel value to the right, and many more operations. Treating images as grids of numbers is the basis for many image-processing techniques. Every value varies between 0 and 255 and some time 0 to 1 , and other ranges.

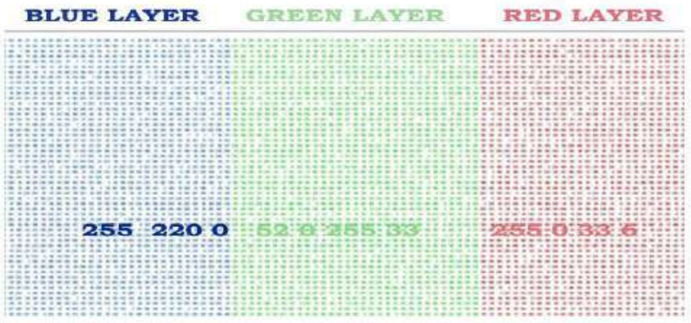


Fig3. RGB image numerical representation

Most color and shape transformations are done just by mathematically operating on an image and changing it pixel-by-pixel.

I.4.2 Digital Image Processing

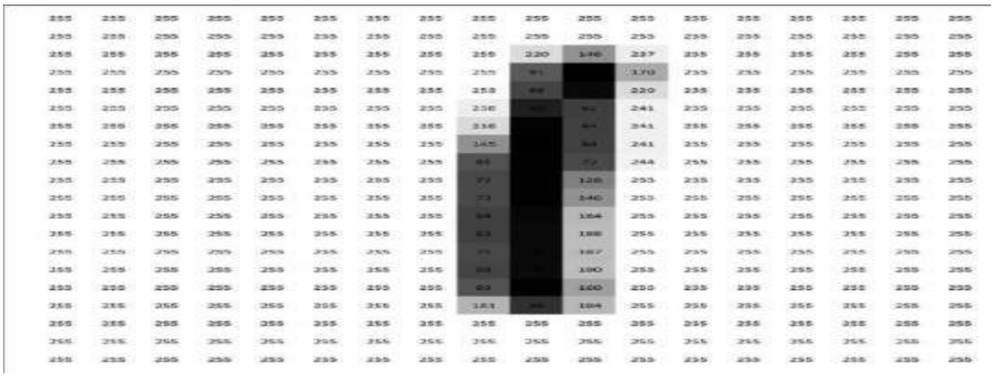
Digital image processing refers to processing digital images by means of a digital computer ,we can define image processing as a discipline in which we deal with images as inputs and we get different outputs whether images ,data,predictions, or annotations .We believe this to be a limiting and somewhat artificial boundary .

I.5 Color Space

I.5.1 Gray scale Image

Gray-scale image is the one in which the dimension of single pixel is one, representing only the brightness esteem , therefore, it carries only bright-darkness information. Gray scale image, a kind of black-and-white or gray monochrome.

Fig4. A gray scale imageI1



255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	220	146	237	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	91	[illegible]	170	255	255	255	255	255	255
255	255	255	255	255	255	255	255	253	68	[illegible]	220	255	255	255	255	255	255
255	255	255	255	255	255	255	255	236	[illegible]	92	241	255	255	255	255	255	255
255	255	255	255	255	255	255	255	116	[illegible]	87	241	255	255	255	255	255	255
255	255	255	255	255	255	255	255	145	[illegible]	94	241	255	255	255	255	255	255
255	255	255	255	255	255	255	255	61	[illegible]	72	244	255	255	255	255	255	255
255	255	255	255	255	255	255	255	77	[illegible]	128	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	73	[illegible]	146	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	64	[illegible]	184	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	63	[illegible]	188	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	75	[illegible]	187	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	80	[illegible]	190	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	83	[illegible]	160	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	181	36	184	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Fig5. Gray scale pixels representation

I.5.2 HSV Color space

HSV(Hue, Saturation& Value/Brightness) is a color space that attempts to represent colors the way humans perceive it . It stores color information in a cylindrical representations of RGB color pixels.

- ○ Hue – Color Value (0-360)
- ○ Saturation –Vibrancy of color (0-255) or (0-1)
- ○ Value – Brightness or intensity(0-255) or (0-1)

It is useful in computer vision for color segmentation.

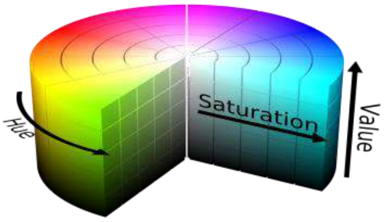


Fig6. HSV diagram

In RGB, filtering colors isn’t easy, however, HSV makes it much easier to set color ranges to filter specific colors as we perceive them .

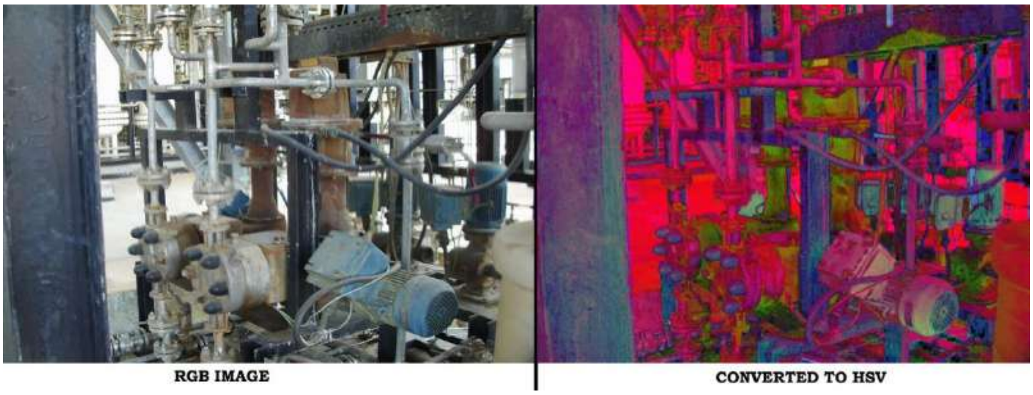


Fig7. RGB vs HSV converted image I2

We can clearly deduce that most of corroded spots are highlighted in green but there are other regions that are also highlighted in green .

Let’s now calculate the HSV histogram of the image, the dimensions are (x:0-255,y: the number of pixels), The first diagram is calculated without separating colors, however, the second we split colors by Red, Green, and Blue.

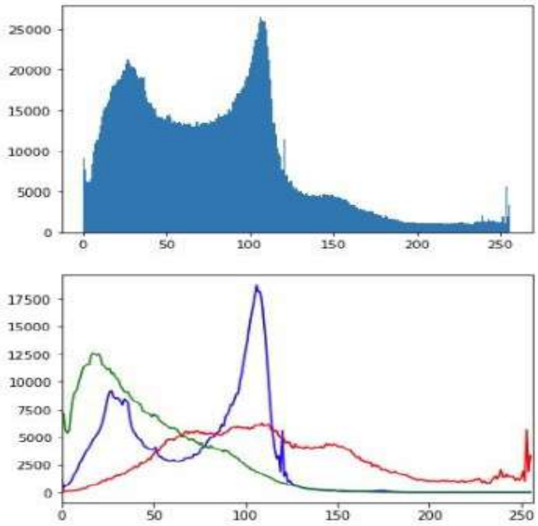


Fig8. HSV Histogram with and without splitting colors for image I2

We see from the diagram that our image has a lot of dark area with three colors, in contrast, we have a few light area, in addition we have a spike of blue color in medium axe.

I.6 Transforms

II.6.1 Frequency in images

We have an intuition of what frequency means when it comes to sound. High-frequency is a high pitched noise, like a bird tweets , and low frequency sounds are low pitch, like a deep voice or a bass drum. For sound, frequency in images actually stands on the same principle, when it comes to image , refers to the rate of change , low frequency means there is no change in the area , and high means that the rate of changes is high.

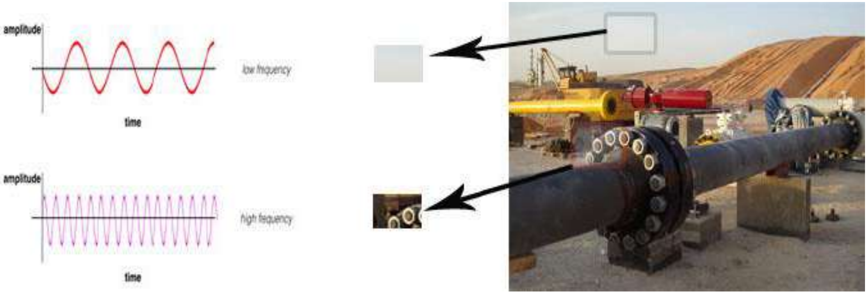


Fig9. Frequency of an image of oil pipeline

I.6.2 Fourier Transform

The Fourier Transform (FT) is an important image processing instrument, which is used to analyze an image and decompose it into its frequency components. The output of an FT represents the image in the frequency spectrum , while the input image is the spatial domain (x, y) equivalent. In the frequency image, each point represents a particular frequency contained in the spatial domain image. The frequency in image represents a specific container in spatial area, for pictures with a great deal of high-recurrence segments (edges, corners, and stripes), there will be various focuses in the frequency area at high frequented esteems.

Fourier Transform is used to analyze the frequency characteristics of several filters. For images, 2D Discrete Fourier Transform (DFT) is used to find the frequency domain. Another fast algorithm called Fast Fourier Transform (FFT) it can be used to calculate DFT.

For a sinusoidal signal,

$$x(t) = Asin(2\pi ʃ t) \qquad (1)$$

We can say ʃ is the frequency of signal, and if its ʃ domain is taken, we can see a spike at ʃ . If signal is examined to form a discrete signal, we get the same frequency domain, but is periodic in the range [-π,π] or [0,2π] (or [0,N] for N-point DFT). You can consider an image as a signal which is sampled in two directions. So measuring Fourier transforms in both X and Y directions yield the frequency representation of image.

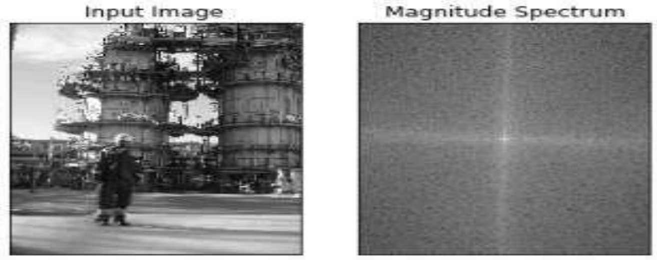


Fig10. Magnitude spectrum applied to an image of a worker in CPF .

Fig10. is an image of an employee stands beside a CPF columns along with the corresponding frequency domain image (right). The concentrated points and four rays the center of the frequency domain image mean that this image has a lot of high frequency (rough background) components.

This decomposition is particularly focuses in the context of band-pass filters, which can isolate a certain range of frequencies and mask an image according to a low or high frequency threshold.

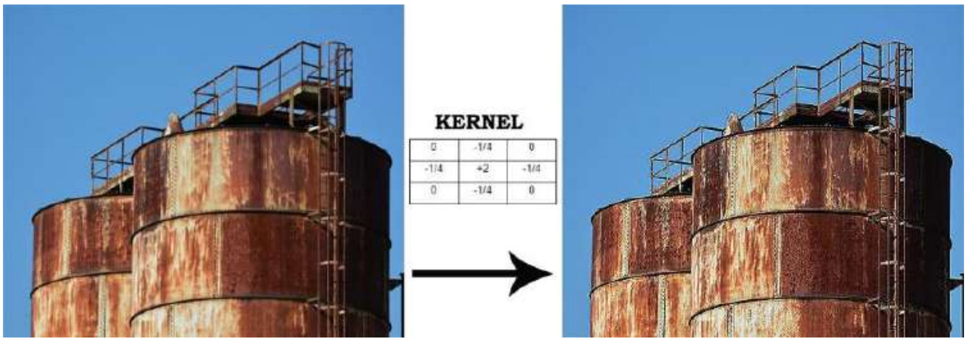
I.7 Filters

I.7.1 High-pass filters

High-pass filter is generally used to sharpen images. These filters emphasize to discover details in through the image – exactly the counter of the low-pass filter. However, High-pass and low-pass filters work precisely the same way , they just utilize a different convolution kernel. See the example below, notice the minus signs for the pixels in the edges. If there is no change in intensity, nothing happens. But if one pixel is brighter or darker than its adjacent neighbors, it gets boosted.

$$\begin{vmatrix} 0 & -1/4 & 0 \\ -1/4 & 2 & -1/4 \\ 0 & -1/4 & 0 \end{vmatrix} \qquad (2)$$

Nevertheless, while low-pass filter mitigates noise, high pass filter amplifies the noise, you may get away if the image has a lot of noise.



0	-1/4	0
-1/4	+2	-1/4
0	-1/4	0

Fig11. Sharpening oil tank image by high pass filter

I.7.2 Low-pass filters

I.7.2.1 T tan^{-1}

It is also called blurring, the main effect of low pass filter is reducing the intensity in image. The simplest low-pass filter just calculates the average of a pixel and all of its eight adjacent neighbors. The result yielded replaces the original pixels values. The process will be repeated over the image.

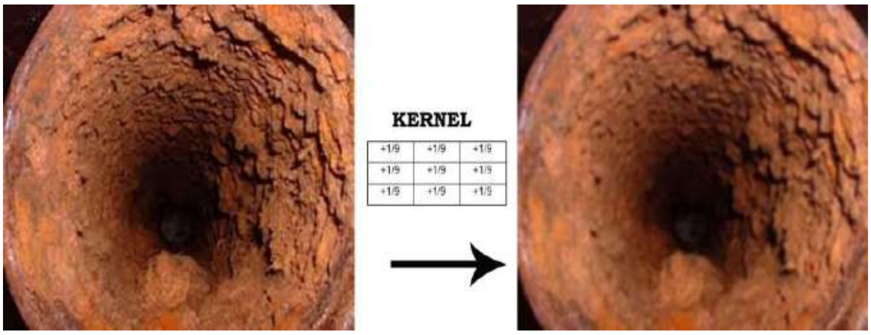


Fig12. Blurring inside a pipeline by a low pass filter

Low-pass filter returns images foggy. But why would we want a blurry image? Plethora images can be noisy – No matter is good your camera-, we always get some noisy points such as light, dots etc.

Noise always changes randomly from pixel to pixel since each pixel automatically generates noise. So the low-pass filter smooths down the noise more than it does the image. By deleting or mitigating it , and many changes can't be seen without low-pass filter . Therefore we use this filter sometimes to display faint details.

I.7.2.2 Gaussian Blur

Probably, the most frequently used filter in computer vision, named after the German mathematician and scientist Carl Friedrich Gauss, it is essentially a weighted average which gives bigger weight to the center pixel, but taking into account surrounding pixels.

In Mathematic applying Gaussian Blur on image, is sort of convolving it. Sometimes they call it two-dimensional Weierstrass transform. Contrary, convolving by a circle (i.e., a circular box blur) would more accurately reproduce the 'bokeh' effect. Since the Fourier transform of a Gaussian is another Gaussian. Bluring with Gaussian reduce the image high-frequency components, so it is thus a low pass filter.

The Gaussian function:

$$\mathrm{G}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \quad (3)$$

Below , a sample of applying Gaussian blur on image of a corroded metal pole.

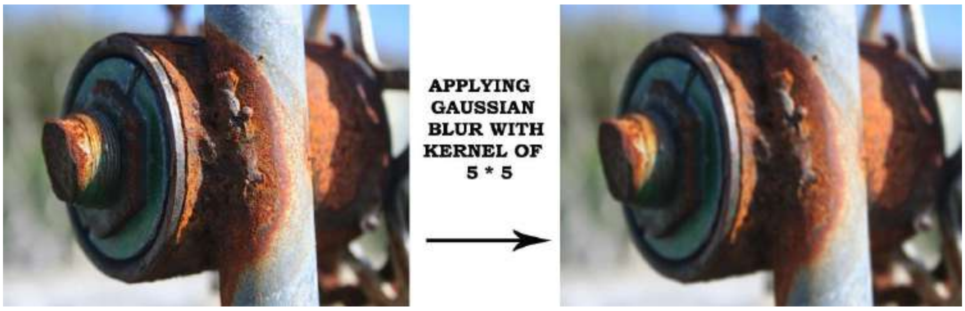


Fig13. Gaussian blur applied to a corroded metal.

I.7.2.3 Sobel filters

The Sobel filter is very commonly used in edge detection and in finding patterns in intensity in an image. Applying a Sobel filter to an image is a way of taking (an approximation) of the derivative of the image in the xx or yy direction. The operators for Sobel_xSobelx and Sobel_y Sobely, respectively, look like this:

$$Sx = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad Sy = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \qquad (4)$$

Next, let's see an example of these two filters applied to an image of the brain.

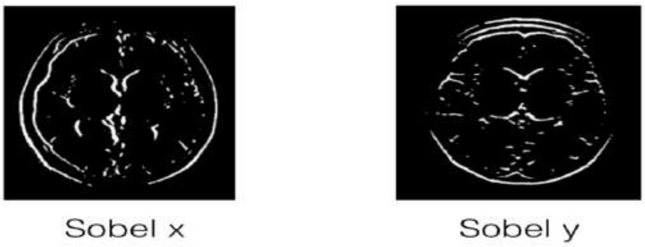


Fig14. Sobel x and y filters (left and right) applied to an image of a brain [XXX]

xx vs. *yy*

In the above images, you can see that the gradients taken in both the x and the y directions detect the edges of the brain and pick up other edges. Taking the gradient in the x direction emphasizes edges closer to vertical. Alternatively, taking the gradient in the y direction emphasizes edges closer to horizontal.

Sobel also detects which edges are strongest. This is encapsulated by the magnitude of the gradient, the greater the magnitude, the stronger the edge is. The magnitude, or absolute value, of the gradient is just the square root of the squares of the individual x and y gradients. For a gradient in both the xx and yy directions, the magnitude is the square root of the sum of the squares.

$$|Sobelxy| = \sqrt{sobelx^2 + sobely^2} \qquad (5)$$

II.7.2.3.1 Direction

In many cases, it will be useful to look for edges in a particular orientation. For example, we may want to find lines that only angle upwards or point left. By calculating the direction of the image gradient in the x and y directions separately, we can determine the direction of that gradient!

The direction of the gradient is simply the inverse tangent (arctangent) of the yy gradient divided by the xx gradient:

$$tan^{-1}(\frac{sobelx}{sobely}) \qquad (6)$$

Sobel: Emphatizes horizontal and vertical edges.

Laplacian :Gets all orientations.

Canny: Optimal due to lower error rate, well defined edges and accurate detection.

But , we will study only sobel and cunny as they are the most frequented used .

Canny Algorithm

Is a popular edge detection algorithm, developed by John F. Canny in 1986.

The strength of this algorithm comes from the multitude stages applied before getting the finale image.

Noise Reduction

The first stage is to delete the noisy area in the image with a 5x5 Gaussian blur filter. We have already seen in this chapter.

Finding Intensity Gradient of the Image

The second stage is to Smooth the image then filter it with a Sobel kernel in both horizontal and vertical direction to yield first derivative in horizontal direction (G_x) and vertical direction (G_y). From these two images, we can find edge gradient and direction for each pixel ***.see the equation number (X)---- sobel gradient equation and equation number X sobel direction***

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

Non-maximum Suppression

After getting gradient magnitude and direction, a thorough scan will be done to delete any unwanted shapes that may not form edges . For this, at every pixel, pixel is checked if it is a local maximum in its adjacent neighborhood in the direction of gradient.

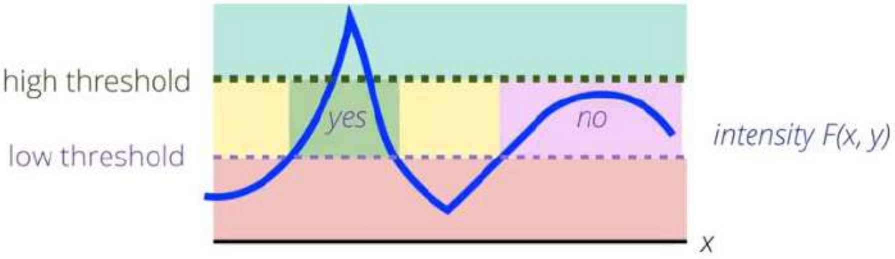


Fig15. High and low threshold to eliminate weak edges and noise

Here , is an example of applying canny edge:

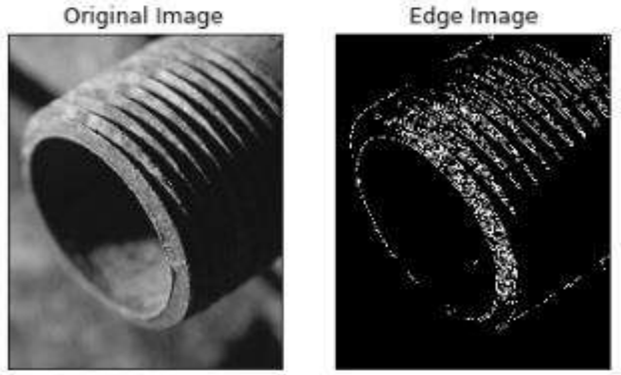


Fig16. Canny edge detection algorithm applied on a thread of pipe.

I.8 Feature extraction

In an application of computer vision, whether it is classification or regression, Machine learns to observe data that we believe contain information are taken as inputs and fed to the system for decision-making. Ideally, we should not need feature extraction as a separate process, the classifier (or regressor) should be able to use whichever features are the best to learn quickly and efficiently , In this section we will describe some features that we think are necessary in our study, some feature extractor algorithms will be used in our experiment. So, it is crucial to take a look on their basics and principals.

I.8.1 Gabor filter

In image processing This filter is very efficient to texture analysis, basing on image frequency content and specifies the direction in a localized region , frequency and orientation representations of Gabor filters are claimed by many vision scientists to be similar to the human visual system.

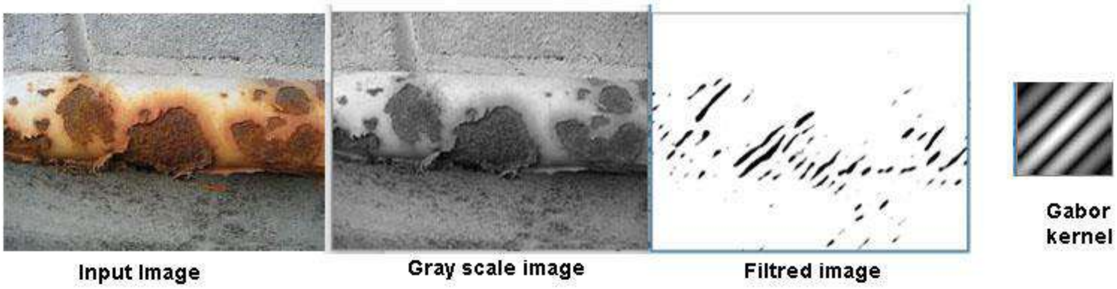
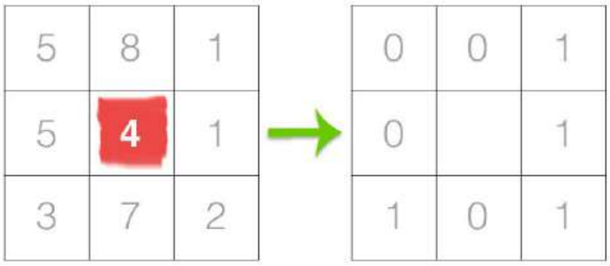


Fig17. Gabor filter applied on a corroded pipe

I.8.2 Local Binary Pattern (LBP)

LBP is a simple and very efficient texture algorithm which labels the pixels of a given image by thresholding the neighbors of each pixel and considers the result as a binary number.

The first step in constructing the LBP texture descriptor is to convert the image to grayscale. For each pixel in the grayscale image, we select a neighborhood of size *r* surrounding the center pixel. Then we calculate the value for this center pixel and store it in the output 2D array with the same width and height as the input image.



5	8	1
5	4	1
3	7	2

0	0	1
0		1
1	0	1

Fig18. The first step in constructing a LBP is to take the 8 pixel neighborhood surrounding a center pixel and threshold it to construct a set of 8 binary digits.

Then , we need to calculate the LBP value for the center pixel. We start from any neighboring pixel and work our way clockwise direction or counter-clockwise, but our ordering must be kept *consistent* for all pixels in our image and all images in our dataset.

Given a *3x3* neighborhood, we thus have 8 neighbors that we must perform a binary test on. The results of this binary test are stored in an 8-bit array, which we then convert to decimal, like this:

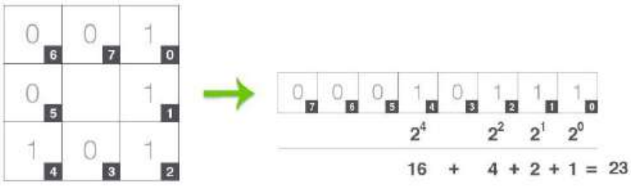


Fig19. Taking the 8-bit binary neighborhood of the center pixel and converting it into a decimal representation.

This process of thresholding, accumulating binary strings, and storing the output decimal value in the LBP array is then repeated for each pixel in the input image.

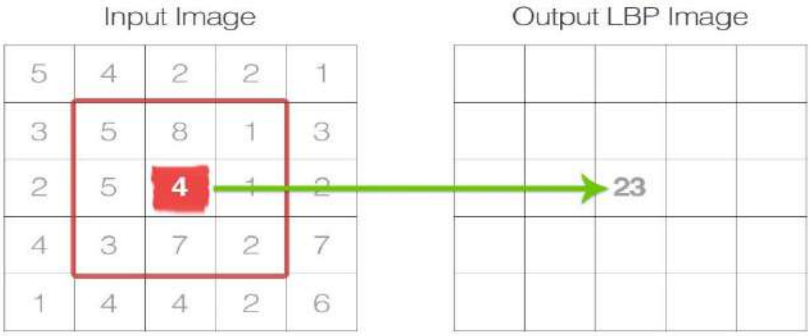


Fig20. The calculated LBP value is then stored in an output array with the same width and height as the original image.

I.8.3 Histogram of oriented gradient (HOG):

HOGs are a feature descriptor that has been widely and successfully used for object detection. Its aim is to count the occurrence of gradient orientations in a portion of an image.

HOGs are often used along with SVM (Support vector machine) classifiers. Each HOG descriptor that is computed is fed to SVM classifier to determine if the object was found or not.

The implementation of the HOG descriptor algorithm is as follows:

1. Split the image into tiny connected areas called cells, and for each cell calculate a histogram of gradient directions or edge orientations for pixels that are the cell.

Specifically, this method requires filtering the color or intensity data of the image with the following filter kernels: [-1,0,1] and $[-1,0,1]^T$

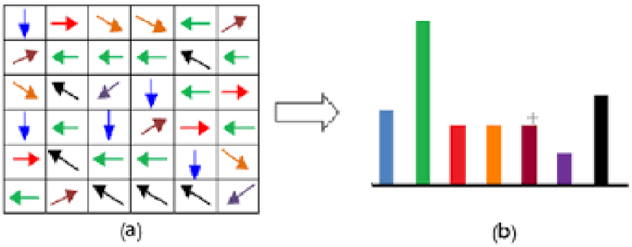


Fig21. Histogram of gradient directions or edge orientations

2. Discretize each cell into angular bins according to the gradient orientation.
3. Each cell's pixel has its weighted gradient to its corresponding angular bin.
4. Groups of neighbors' cells are deemed as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.

   Dalal and Triggs [15] examined four different methods for block normalization

5. Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor

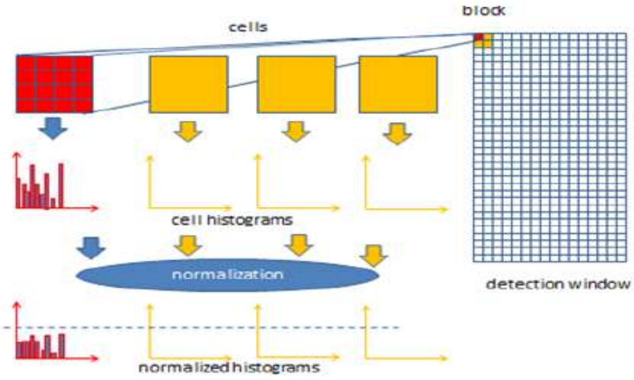


Fig22. HOG Histogram applications steps

Bellow an example of HOG histogram applied on an image of our dataset.

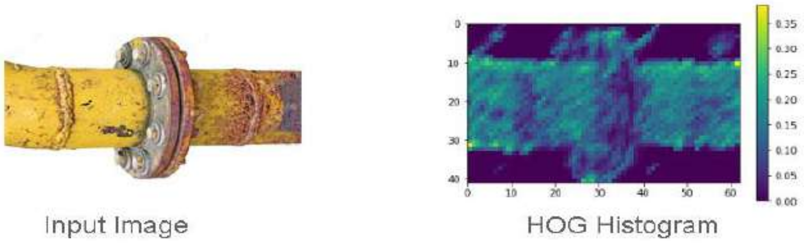


Fig23. HOG Histogram applied on a pipe

I.8.4 Texture

I.8.4.1 GCLM

Is essentially a descriptor of texture[18], in gray scale space and a simple principle based on examining the relationship spatial between pixels, the GLCM characterizes the texture of an image to calculate how many times each pair of pixels appears in the image. At the beginning you have to scale the image to gray in order to find the 256 intensity values, from these values we create a matrix 256x256 which describes the number of appearance of each pair of pixels at a coordinate system of a vector orientation. We can always modify the intensity scale in the result matrix in order to save time and minimize the memory required.

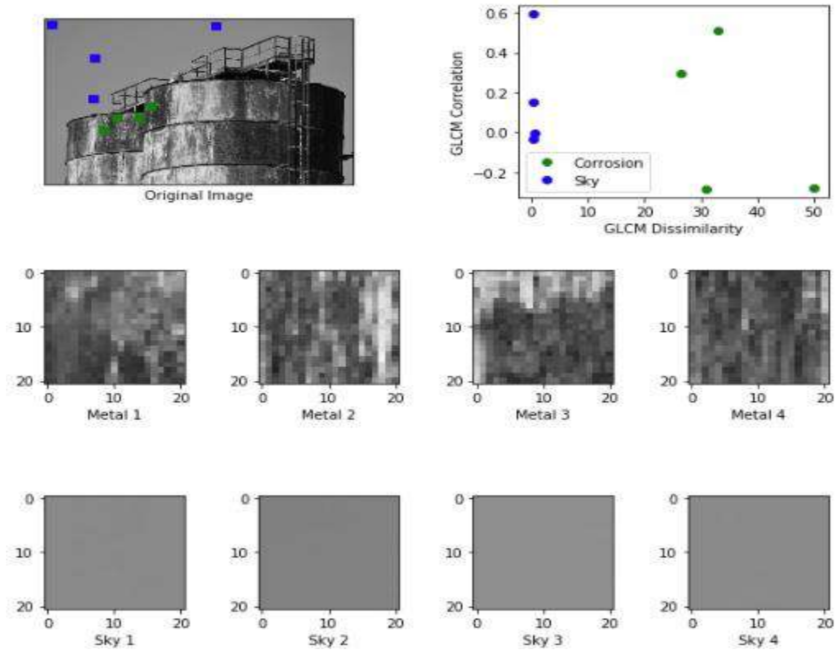


Fig24. GCLM applied to a corroded tank with associated correlation and dissimilarity diagram .

We can see in the figure above fig24. that we have been pointed 8 spots , 4 of corroded metal and 4 for the sky , before we drew the diagram of GLCM correlation and dissimilarity.

It’s pretty clear that sky points in blue have great similarity, so they are aligned approximately in 0 line and their correlations are too high.

I.9 Summary of the chapter

After briefing a multitude of image processing techniques and their utility and application in our work. Contour, edge, textures, color map, all these feature can be extracted by applying the previous filters , descriptors, transforms. Besides, these techniques are computer vision basics, it is vital to know the image processing fundamentals in its early years, since we will employ them whether explicitly in the image processing or implicitly deep learning experiments.

For the task of corrosion detection, we will see to what extent handcrafted features are efficient to distinguish corroded image from the non-corrosive ones.

Chapter II. LITERATUERE REVIEW

Although, works on corrosion detection are not ample in the literature of computer vision, several studies tried to tackle this phenomenon by using different approaches , Corrosion detection based on image analysis can broadly be categorized into two categories namely deep-based methods and image processing-based methods .The most of former attempts use image processing , that’s to say , extract handcrafted features and classify them with conventional classifiers, whereas, some others prefer deep learning as an extremely powerful tool to recognize the defected areas.

In the literature of computer vision, we find both approaches that have been used in order to find a solution for corrosion detection in different structures, such water pipes , bridges , civil work metals ..etc.

II.1 Image-based processing methods

The first approach involves image-processing techniques, Bondada *et al.* [7], used threshold on mean saturation value of an image to measure the corrosion regions and eventually locate the centers of densely corroded area.

Po-Han Chen *et al*. [5] relied on a new support-vector-machine-based rust assessment approach and developed a method for steel bridge rust recognition. SVMRA extracted features from Fourier transform and combine them with support vector machine to provide an effective method for non-uniformly illuminated corroded image recognition.

Margarita R. Gamarra Acosta *et al*.[16].presented a novel technique to test the system under a large number of possible corrosion configurations (quantity and morphology) ,for the first time oxide texture simulation with Perlin Noise, which allowed validating the performance of the rust detector model under different levels of corrosion, including extreme rust conditions.

Probabilistic descriptors are determined by means of discriminant analysis using Fisher indexes. A Bayesian classifier is used to identify rusted regions. Additionally, performance tests under different noise conditions and texture variations, generated with Perlin Noise, are presented.

Choi *et al*. [17], used color, texture and shape features to calculate the corrosion surface instead of electrochemical measures that has been judged overcome by this new technique. Mohammadreza Motamedi *et al*. [6], their main work’s aim is to use the Digital Image processing techniques including image smoothing, point processing, morphology, filtering and edge detection operations that can be used for pipeline networks which results in related defects that have been recognized in each image.

Gand Ji *et al*. [8], tried to improve the corrosion detection by using the improved watershed segmentation method to eliminate over-segmentation, this leads to increase the accuracy detection of the coating material corroded defects.

Jiménez-Come *et al*.[9], worked on different models based on artificial neural networks (ANNs), support vector machines (SVMs), classification tree (CT) and k-nearest neighbour (KNN) these algorithms were presented to develop an automatic way to predict pitting corrosion. Another attempts is Nhat-Duc Hoang *et al* [16]. They put forward an automatic method based on image processing and machine learning for water pipe corrosion recognition

Their method based on texture analysis in the first place besides RGB color channels properties. Statistical Properties of Color Channels. Such as , the statistical properties of three color channels (red, green, and blue) of an image samples can be employed to represent

Color feature .Thus, an image is described in a RGB color space. It is noted that besides RGB, there are other color spaces such as HSV , HIS, YCbCR…ect which can also be useful in the task of corrosion detection.

To extract texture features, some works relied on Gray-Level Co-Occurrence Matrix (GLCM). To employ this technique, a color image must first be converted to a gray scale one. the GLCM discriminates different image textures based on the repeated occurrence of some gray-level patterns existing in the texture. Haralick *et al*.[22] Suggested four angles 0, 45°, 90°, and 135° with r=1 . Accordingly, angular second moment (AM),contrast (CO), correlation (CR), and entropy (ET) matrix’s outputs are considered.

Another texture descriptor which has been used along with GLCM . GLRL is a texture description method proposed by Galloway [25] this is method is highly effective in discriminating textures featuring different aspects and has been successfully applied in various fields of study .This is because GLRL is developed based on the fact that relatively long gray-level runs are observed more frequently in a coarse texture and a fine texture typically has more short runs .

SVM support vector machine, described in [19], has been employed in the experiment for the classification task . SVM is a robust pattern recognition method established on the theory of statistical learning. Given the task at hand is to classify a set of input features. SVM model constructs a decision surface that separates the input space into two distinctive regions characterizing the two different two categories. The SVM algorithm aims at identifying a decision boundary so that the gap between classes is as large as possible.

To obtain the best result of SVM , tuning hyperparameters is needed .For that reason , the work of Nhat-Duc Hoang *et al* [16] utilize the algorithm. Differential Flower Pollination (DFP), is a population-based metaheuristic that combines the advantages of the standard algorithms of differential evolution SVM model depend on a proper selection of its hyperparameters including the penalty coefficient (c) and the kernel function parameter (σ). The first hyperparameter affects the penalty imposed on data samples deviating from the established decision surface, the later hyperparameter specifies the smoothness of the decision surface.

Finally, the result given by this approach as listed in the paper reaches high accuracy in training but give lower results in test phase.

Another attempt Bonnin-Pascual *et al.* [11], introduced two algorithms (Weak-classifier Colour-based and AdaBoost) to evaluate corrosion detector performance and concluded that False Negative percentage is not null however False Positive percentage is null due to the presence of different structures in the image that are misclassified as defects. Comparing two algorithms it seems that Weak classifier color-based outperforms the Adaboost.

Even though image processing gives good result, however, it is still venerable to image variations e.g., noise, change to scale and illumination. For instance, Analyzing color for corrosion has no steady performance due to many factors likewise illumination and contrast, and the corrosion has divergent colors that ranges from bright orange ,to dark maroon and some times in found in purple .

Texture analysis has also its drawback, take GLCM as an example, this descriptor is frequently used in biomedical field, it can distinguish abnormal lesions in chest or breast' radiographs[20]. However, for some document images with a large amount of noise, the GLCM features are not appropriate [20].

II.2 Deep Learning-based methods

Fast progressions in deep learning and improvements in device capabilities including computing, memory capacity, and image resolution; have improved performance and cost-effectiveness of vision-based applications and systems. Compared to conventional Computer Vision including image-processing techniques, deep neural networks (DNNs) enable handcrafted features to achieve greater accuracy in tasks such as image classification, semantic segmentation.

DNNs also provide superior flexibility, since we can retrain models by using transfer learning for other purposes, conversely to computer vision algorithms, which are more domain-specific.

The cornerstone of methods belonging to the first category is to describe and recognize images using Convolution Neural Networks (CNN) due to their significant impressive capabilities. For instance, Yao *et al*. [12], proposed a model to recognize corrosion in hull structural plates. They have constructed a classification model by training a CNN on a set of corroded and non-corroded samples. A sliding window is then utilized to locate the damage position within the image.

Atha. et al.[13] presented an experimental evaluation of four deep networks, for assessing corrosion in metallic surfaces.

They proposed to use a CNN to classify image regions as corroded or not. To this end, two existing networks that have shown significant success on ImageNet[23], namely VGG16 and ZF Net ,are used. In addition, two additional networks are proposed ,and itentiolly tried to reduce parameters to be faster than ZF Net and VGG16 in training and testing phases.

In order to identify the optimal color space for corrosion detection . using a CNN, the color spaces RGB, YCbCr, CbCr, and grayscale are used to train and test ZF Net. It was demonstrated in Mohammad R. Jahanshahi *et al*.[24] work , that identifying corrosion is not always the most accurate in the RGB color space. After determining the optimal color space, different architectures are evaluated for that color space. In addition, to classify the regions of an image as corroded or not, multiple sliding window sizes (128 x 128, 64 x 64, and 32 x 32) are used. A sliding window approach is utilized to determine corroded regions within an image. Analyzing these different window sizes will help identify the optimum window size for a sliding window approach .

They created a dataset from 926 images were gathered using different digital cameras eventually , samples of 33,039 corroded and 34,148 non corroded images of 128 x 128 pixels, obtained from the original 926 images, is relatively small compared to the size of ImageNet. In this case, overfitting could be an issue.

In addition, two networks have been proposed. Corrosion7 is seven-layer network with five convolution layers followed by two fully connected layers. This network contains half
as many feature maps for each convolution layer compared to ZF Net. Decreasing the number of feature maps will decrease the computation time of the network.

In addition, the first fully connected layer has 1024 neurons and the second is the classification layer and has two neurons. This is compared to two layers of 4096 neurons followed by the classification layer of two neurons in ZF Net. It is proposed that accuracy will not suffer when we reduce the fully connected layer neurons , because less features are needed to perform a binary classification of corrosion compared to classifying 1000 different objects for which ZF Net was designed. Corrosion5, the second proposed CNN, is a five-layer network with three convolution layers followed by two fully connected layers. The two fully connected layers are the same as Corrosion7. Corrosion5 is used to determine whether or not reducing the number of convolution layers or feature maps result in a better performance.

To tune hyperparameters they set learning rate to 0.001, the number of iterations used is 8800 and they increased the number of iteration regarding the windows size input. They concluded that the larger image size we use , the more feature extracted we get , so it is recommended to use image with higher resolution.

Nash *et al*. [10], tried to approach the human level and getting accurate results by training of a deep learning model, they created a website to get receive images from website visitors . Luca Petricca *et al*.[11], presented a comparison between classical computer vision techniques and deep learning for corrosion detection, they deduced that deep Learning model performs better in a real case scenario.

Another recent and interesting work is that of , Blossom Treesa Bastian *et al*[18], which have proposed a computer vision based approach to detect corrosion in water, oil and gas pipelines and to classify the images of pipelines based on their corrosion level.they used different pre-trained models and windows sizes to evaluate the best performance.

they created a new dataset for the task ,to classify corrosion based on their level , high, medium, low and no corrosion detected. They proposed Custom-CNN employed the same number of convolutional layers as that of ZFNet. Different windows size and Thus the Custom-CNN trained on with an input size of 256 × 256 provides the best performance.

Another recent attempt of deep learning has been made by Chijioke Ejimudaet *et al*.[26] they applied instance segmentation using ResNet to detect stress corrosion cracking, pitting corrosion and uniform corrosion defects that exist in industrial facilities(including oil and gas) successfully. Although their model accurately classified and segmented the corrosion defects for the most part, however, to address the high total network loss value, they suggested:

1) Additional quality data of the corrosion defect classes,2)Using advanced architectures such as SegNet or U-Net, 3)Varying the current model architecture configuration options

To conclude, CNNs need a big amount of data and sometimes we can’t collect enough, in this case, models are prone to be over-fitted when the training data is not able to generalize for the task at hand. So for that we seek the data augmentation and transfer learning techniques. Data augmentation is a common pre-processing task which is used when there is limited training data .It can involve performing random rotations, shifts, shears, noising.. etc.

On the top of that, DL-based models takes a lot of time to train and to test also, even if we use machines endowed by high computational resources , still slower than any handcrafted feature extraction .

Summary

A lot of the CV techniques invented in the last decades, have become irrelevant and unused recently because of DL emergence. However, knowledge is never being obsolete and there is always something worth to learn from each generation of innovation. That knowledge can give you more intuitions and tools to use especially when you wish to deal with 3D CV problems for example. Knowing only DL for CV will dramatically limit the kind of solutions in a computer vision engineer’s arsenal.

In this literature review chapter , we have laid down many works for some traditional CV techniques that are still very much useful in corrosion detection problem , as well as in the age of DL there are new technique in the literature .

From the first impression, we can acquaint that DL is advantageous in many aspects, against Image-based processing, but, both approaches have weaknesses and strengths and it is too early to draw conclusions.

Chapter III. PROPOSED METHODS

At first, we start by giving an overview on the proposed approaches and method that have been conducted in our experiment, including approaches whose review has been studied in the previous chapter .As usual , we commence by examining handcrafted features , then we explore deep learning techniques .

The first part, we explain how our is done by applying certain methods in image-based processing, which leads to extract features, the latter used eventually to classify image based on machine learning algorithm .

The second part is consecrated to deep learning ,on which we apply the techniques that are belong to such transfer learning , data augmentation and fine tuning .

In the final step , we introduce our proposed approach for ensemble models, we give only basics and general idea.

III.1 Handcrafted features

III.1.1 Color Moment

The method consists of calculating the Red, Green and Blue means respectively; alongside with their dispersion with variance, it is a versatile method that can be used for different purposes.

Mean

The mean of a color moment can be interpreted as the average color in the image, and it can be calculated by using the following formula:

$$\mathrm{M_i}= \sum_{j=1}^{N} \frac{1}{N} P_{ij} \qquad (10)$$

Variance

The variance represents the dispersion of a color in the image, and that can be calculated by using the following formula:

$$\mathrm{V_i}= \sum_{j=1}^{N} \frac{1}{N} (P_{ij})^{2} \qquad (11)$$

Before applying machine learning algorithms, we can intuitively see that R G B means and Variances of corroded and non-corroded images are overlapped in the figure bellow :

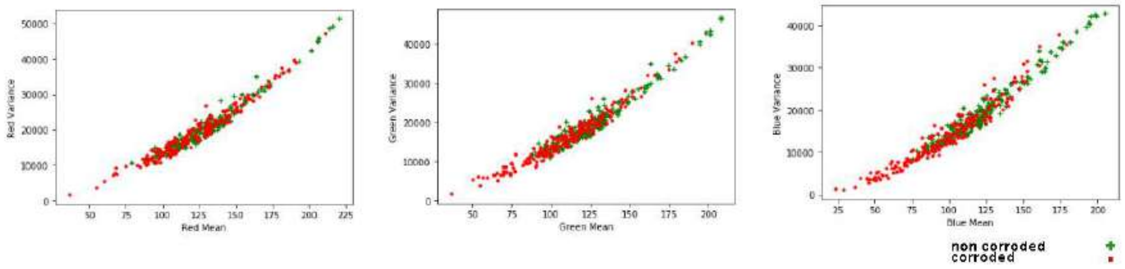


Fig25. R,G and B channel's means and variances of dataset images

III.1.2 Red average

Another classical technique method has been used by Luca Petricca et al [11] before going to deep learning , since a corroded area (rust) has no clear shape, they focused on the colors and in particular the red component. By extracting the red color from the HSV image (in OpenCV, Hue range is [0,179], Saturation range is [0,255] and Value range is [0,255]).The red components is spread in a non-contiguous interval (range of red color in HSV is around 160-180 and 0-20 for the H component) .

We applied this technique to calculate the percentage of the red in images combined with red average(Mean), and we got these results :

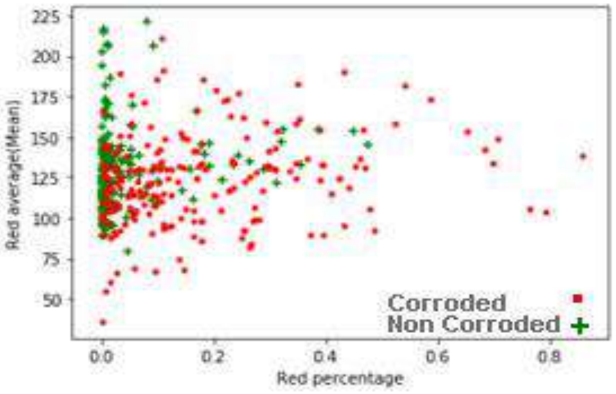


Fig26. Two-dimensional vectors Data distribution of Red percentage and Red mean

SVM (Support Vector Machines)[27]: is supervised learning method, and is known for its effectiveness in classification, regression and outliers detections. SVM works by finding a linear hyperplane, which separates the training datasetinto two classes. As there are many such linear hyperplanes, the SVM algorithm tries to find the optimal separating hyperplane (as shown in Fig. Fig 6) which is intuitively achieved when the distance (also known as the margin) to the nearest training data samples is as large as possible. It is because, in general, the larger the margin the lower the generalization error of the model.

Mathematically, SVM is a maximum margin linear model. Given a training dataset of ŋ Samples of the form {x_1,y_1}; ……..; {x_n, y_n} , where xi is an m-dimensional feature vector and y_i = {1,0} is the class to which the sample x_i belongs to. The goal of SVM is to find the Largest -margin hyperplane which divides the group of samples for which y_i = 1 from the group of samples for which y_i =0 .), this hyperplane can be written as the set of sample points satisfying the following equation:

$$W^T X_i + b = 0 \qquad (12)$$

Feature and classifications

Recall that SVM tries to maximize the distance between these two new hyperplanes demarcating two classes, which is equivalent to minimizing. Thus, SVM is learned by solving the following primal optimization problem:

$$\min(w, b) \ \frac{w^T w}{2} \qquad (13)$$

Nonlinear Decision Boundary

SVM can be extended to nonlinear classification by projecting the original input space (R^d) into a high-dimensional space (R^D), where a separating hyperplane can hopefully be found. Thus, the formulation of the quadratic programming problem is as bellow in Fig XXX.

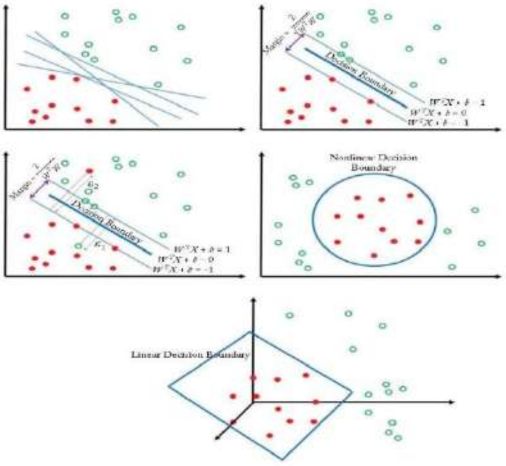


Fig27. SVM features and classifications

We have used SVM to classify our model ,see the table2 below , it shows the scores we got by differentiating hyperparameters, we can easily deduce that kernel ‘linear’ performs better than other kernels .

III.1.3 Texture analysis

In the realm of image classification, extracting features from the image is the process that we deal with image properties such color, texture …ect. Intuitionally, texture and color are the most useful features we can extract to discriminate corroded image from the non-corroded , by using ML classifiers algorithms , in our case , we use SVM machine learning method integrated with the DFP metaheuristic to be utilized to construct a decision boundary for classification pipe surface images.

Color properties statistics

It is to say that they used statistical properties of color channels with mere RGB color space, to calculate the mean(μ_c), standard deviation (σ_c), skewness(δ_c), kurtosis(η_c), entropy(ρ_c), range(Δ_c) , these values are calculated as follows :

$$\mu_c = \sum_{i=0}^{NL-1} (\mathrm{I\ i,c}) \times Pc(I) \qquad (14)$$

$$\sigma_c = \sqrt{\sum_{i=0}^{NL-1} (I_{i,c} - \mu c)^{2} \times P_c} \qquad (15)$$

$$\delta_c = \frac{\sum_{i=0}^{NL-1} (I_{i,c} - \mu c)^{3} \times P_c(I)}{\sigma_c^3} \qquad (16)$$

$$\eta_c = \frac{\sum_{i=0}^{NL-1} (I_{i,c} - \mu c)^{4} \times P_c(I)}{\sigma_c^4} \qquad (17)$$

$$\rho_c = \sum_{i=0}^{NL-1} P_c(I)^{2} \times log_2(P_c(I)) \qquad (18)$$

$$\Delta_c = \mathrm{Max}\,(I_c) - \mathrm{Min}\,(I_c) \qquad (19)$$

Gray-level co-occurrence Matrix (GLCM)

The GLCM [20] is a commonly used as texture extractor. Is it employed in gray-scale image, so we need to convert our image to grayscale befor applying the technique .Based on Heralick method , four angles have been established (0,45,90,135) .for the Angular second moment (AM), Contrast(CO), Correlation(CR),and entropy(ET).

In addition to the moments , Haralick has extracted in total 13 textural features, but we found them irrelevant in our case .

$$AM = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_{\delta}^{N} (i,j)^{\,2} \qquad (20)$$

$$CO = \sum_{K=0}^{N_g -1} K^2 \sum_{\substack{j=1 \\ |i-j|=k}}^{N_g} \sum_{j=1}^{N_g} P_{\delta}^{N} (i,j) \qquad (21)$$

$$CR = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \times j \; P_{\delta}^{N}(i,j) - \mu_x \, \mu_Y}{\sigma_X \sigma_Y} \qquad (22)$$

$$ET = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_{\delta}^{N} (i,j) \, \log \left(P_{\delta}^{N} (i,j)\right) \qquad (23)$$

Gray-Level Run Lengths (GLRL)

Is also a texture descriptor proposed by Galloway[18] is highly effective specially in Radiography. GLRL is highly effective discriminator textures for featuring different fineness, therefore , it has been successfully applied in various domains of study . It is because of its construction based on the fact that relatively long gray-level runs can be observed more frequently in a coarse texture and a fine texture typically has more short runs .we define a run-length matrix p(i · j) in a certain direction as the number of times that a run length j of gray level i is observed. .This matrix has different descriptor as shown on the figure bellow :

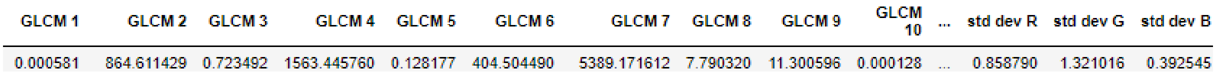
Descriptor	**Notation**
Short run emphasis	SRE
Long run emphasis	LRE
Gray-level nonuniformity	GLN
Run length nonuniformity	RLN
Run percentage	RP
Low gray-level run emphasis	LGRE
High gray-level run emphasis	HGRE
Short run low gray-level emphasis	SRLGE
Short run high gray-level emphasis	SRHGE
Long run low gray-level emphasis	LRLGE
Long run high gray-level emphasis	SRHGE

Table1. Texture descriptors using GLRL.

NB: N are the number of gray levels and the maximum run length, respectively. Let Nr and Np be the total number of runs and the number of pixels in the image, respectively.

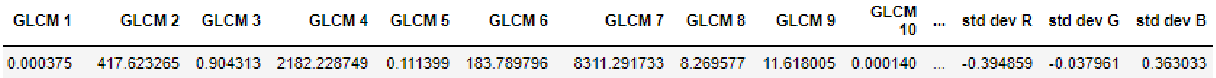
After extracting 78 features, SVM machine learning integrated with DFP metaheuristic is utilized to construct a decision boundary for classification. Differential Flower Pollination (DFP) is a genetic algorithm that optimizes parameters , since SVM model training and prediction depend on the combination of its hyperparameters including the penalty coefficient (c) and the kernel function parameter (σ). In this experiment we will employ the DFP metaheuristic to optimize the model training phase of SVM.

After extracting the features from GLCM and GLRLM and image proprieties, it yields a vector of 78 casa and that , before normalization of data :



GLCM 1	GLCM 2	GLCM 3	GLCM 4	GLCM 5	GLCM 6	GLCM 7	GLCM 8	GLCM 9	GLCM 10	...	std dev R	std dev G	std dev B
0.000581	864.611429	0.723492	1563.445760	0.128177	404.504490	5389.171612	7.790320	11.300596	0.000128	...	0.858790	1.321016	0.392545

Table2. A vector of a non-corrosive image



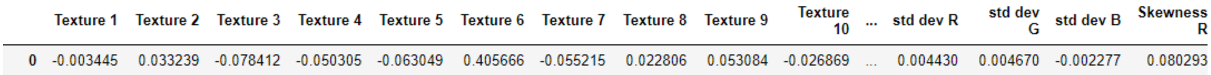
GLCM 1	GLCM 2	GLCM 3	GLCM 4	GLCM 5	GLCM 6	GLCM 7	GLCM 8	GLCM 9	GLCM 10	...	std dev R	std dev G	std dev B
0.000375	417.623265	0.904313	2182.228749	0.111399	183.789796	8311.291733	8.269577	11.618005	0.000140	...	-0.394859	-0.037961	0.363033

Table3. A vector of a corroded image

Normalization

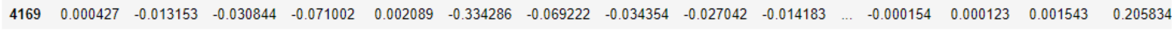
$$z = \frac{\text{x} - \text{mean(x)}}{\text{stdv(x)}} \qquad (24)$$

After applying the normalization , it yields likewise bellow vector :



	Texture 1	Texture 2	Texture 3	Texture 4	Texture 5	Texture 6	Texture 7	Texture 8	Texture 9	Texture 10	...	std dev R	std dev G	std dev B	Skewness R
0	-0.003445	0.033239	-0.078412	-0.050305	-0.063049	0.405666	-0.055215	0.022806	0.053084	-0.026869	...	0.004430	0.004670	-0.002277	0.080293

Table4. A vector of a non-corroded image after normalization



4169	0.000427	-0.013153	-0.030844	-0.071002	0.002089	-0.334286	-0.069222	-0.034354	-0.027042	-0.014183	...	-0.000154	0.000123	0.001543	0.205834

Table 5: A vector of corrosive image after normalization

By applying SVM to fit our model we used Grid-SearchCV to tune the SVM hyperparameters , Grid-Search is an algorithm which is included in DEAP python library , the latter uses genetic algorithms to get the best hyperparameters of SVM model.

After Fitting 5 folds for each of 96 candidates, totalling 480 fits , and the grid is :

param_grid = {'C': [0.1,1,5, 10, 100,500], 'gamma': [1,0.1,0.01,0.001],'kernel': ['poly','rbf','linear', 'sigmoid']}

III.2 Deep learning methods

CNNs are one of the best learning algorithms for image classification [28], it is widely used as one of the ML techniques, especially in vision-related applications. CNNs were inspired by the human neural system proposed by Fukushima [29] and LeCun et al. [30]. Since CNNs use a large features generation with classification and discrimination abilities. Generally, a typical CNN architecture encompasses convolution layers , Each convolution layer is followed by a rectified linear unit (ReLU)[31] , another layer of pooling to down-sample outputs , then, one or more fully connected layers at the end , the fully connected layer combine the extracted features from the previous layers , forming a new neural network , this time will be features classifier , finally , an output neuron for each classification category .

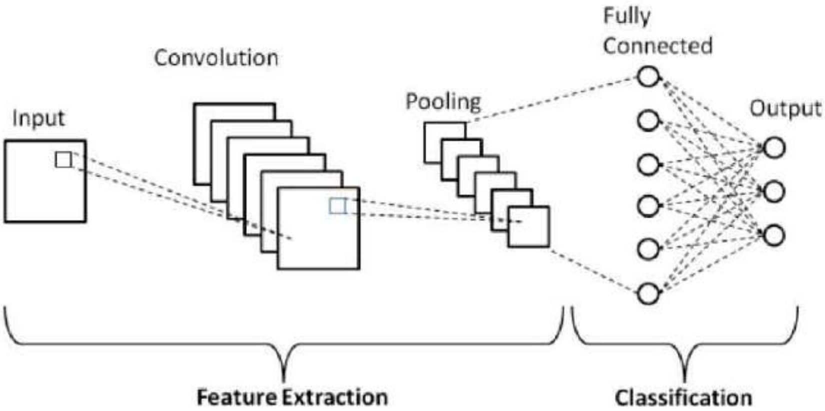


Fig28. CNN architecture overview

III.2.1 Transfer learning

Transfer learning is a common method in computer vision, it can build accurate models in a timesaving way,with transfer learning, instead of designing the learning process from scratch, you just adjust the models that have been learned already but for a different problem and get them fit to your dataset. This way you leverage previous learnings and avoid build new model from the beginning.

A pre-trained model is a model that is trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. Accordingly, due to the computational cost of training such models, it is popular practice to import and use models from published literature (1-

Alexnet[32] ,2-Squeezenet[33],3- Resnet[34],4- DenseNet[35];5-VGG[36]) A comprehensive review of pre-trained models' performance on computer vision problems using data from the ImageNet (Deng et al. 2009)[23] challenge is presented by Canziani et al. [37].

III.2.2 Fine-tuning

In fine-tuning, we knew that retrained model are trained on ImageNet dataset[21], so that we updated all of the model's parameters for our new task, in essence, retraining the whole model by keeping feature extraction frozen by unleashing weights of the whole networks ,this could bring better performance since the learnt patterns could have more weights than being only extracted in last layers of the network.

III.2.3 Data augmentation

It is well known that large amounts of data are imperative for DL model training and achieving high performance. In our relatively small dataset, we need data augmentation to achieve better score, and that is done by applying random geometric transforms without changing class labels. The table 10. Shows each augmentation with detailed parameters.

N°	Augmentation	Parameter
1	Rotation	45°
2	Horizontal flip	25%
3	Vertical flip	25%
4	Width shift	--
5	Height shift	--

Table6. Data augmentation parameters

III.3 PROPOSED APPROACH

III.3.1 Proposed model

We proposed a model whose the architecture is illustrated in Fig33. CROGF7 , refers to the English initial letters of "Corrosion Recognition for Oil and Gas Facilities" , and 7 refers to the number of layers see the detailed parameters , they are shown in Table 9. In the Input layer we resized our images to be 150x150 , the three layers input represent the three channels RGB. We applied 5x5 kernel sizes with striding of 2x2 for CONV1 and CONV2, for the Maxpooling we used window sizes of 3x3 for the all POOL layers. We use 128 filters for the first CONV layer, 256,512 filters for the second and third CONV layers and 256 f filters for the fourth CONV layer. For the first fully connected layer we used 1024 neurons for 512 neurons for the second FC layer. We applied dropout [38] to the classification part. We defined a dropout probability of 0.2 for both first the second and FC layers.

To avoid consumption of processing resources, the proposed model has a total of 7,910,402 parameters and this number comparing to other model such as Alexnet, which has 57,012,034 parameters but also there are many advancement in term of decreasing the size of parameters to have fast processing. Squeezenet is best example

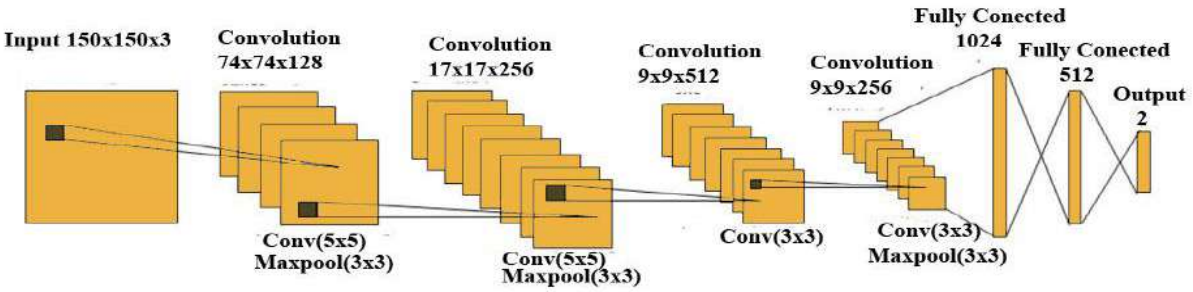


Fig29. The proposed CNN model architecture CROGF7.

```
===========================================================================
Layer (type:depth-idx)    Output Shape    Kernel  Stride  Drp   Param#
===========================================================================
├─Sequential: 1          --                --      --      --     --
|    └─Conv2d: 2-1       [ 128, 74, 74]    5x5     2x2     --     9,728
|    └─ReLU: 2-2         [ 128, 74, 74]    --      --      --     --
|    └─MaxPool2d: 2-3    [ 128, 37, 37]    3x3     2x2     --     --
|    └─Conv2d: 2-4       [ 256, 17, 17]    5x5     2x2     --     819,456
|    └─ReLU: 2-5         [ 256, 17, 17]    --      --      --     --
|    └─MaxPool2d: 2-6    [ 256, 9, 9]      3x3     2x2     --     --
|    └─Conv2d: 2-7       [ 512, 9, 9]      3x3     1x1     --     1,180,160
|    └─ReLU: 2-8         [ 512, 9, 9]      --      --      --     --
|    └─Conv2d: 2-9       [ 256, 9, 9]      3x3     1x1     --     1,179,904
|    └─ReLU: 2-10        [ 256, 9, 9]       --     --      --     --
|    └─MaxPool2d: 2-11   [ 256, 4, 4]      3x3     2x2     --     --
├─Sequential: 2          --                --      --      --     --
|    └─Linear: 2-12      [ 1024]           --      --      --     4,195,328
|    └─ReLU: 2-13        [ 1024]           --      --      --     --
|    └─Dropout: 2-14     [ 1024]           --      --      0,2    --
|    └─Linear: 2-15      [ 512]            --      --      --     524,800
|    └─ReLU: 2-16        [ 512]            --      --      --     --
      |    └─Dropout: 2-17     [ 512]            --      --     0,2    --
```

```
    |     └─Linear: 2-18       [ 2]             --       --     --      1,026
```

Table7. Architecture of proposed model CROGF7.

III.3.2 Network architecture

To analyze the impact of the architecture, the three proposed architectures are studied ,Squeezenet is deeper with 25 layers and but the less computational expensive , however , the other networks are shallow but require more computational resources and that due to the parameters that need to be trained , see table 8.

Alexnet and the proposed model are categorized as spatial exploitation networks[39], however , Squeeznet is considered as a Feature-map exploitation network .

Network	Depth	Category	Number of params	Trainable params
Proposed Model (CROGF)	7	Spatial exploitation	7,910,402	7,910,402
Proposed Squeeznet	25	Feature-map exploitation	723,522	1026
Proposed Alexnet	8	Spatial exploitation	57,012,034	8,194

Table8. Parameters of Proposed Networks

We have used pre-trained models for learning transfer, We have chosen the most successful models in the last ten years, see ILSVRC competition [39],two models have been assigned the task ,Squeeznet1_1 and Alexnet , we considered time and performance as the most important factors to select models .

The three networks were chosen because they have exhibited good performance among others on our dataset and provide less error rate in our task classification. Besides, they provide smaller size in depth and number of parameters. Shallow Convolutional Neural Networks (CNNs) require less communication across servers during distributed training and require less bandwidth in cloud to import new models into devices.

Below you find the detail architecture of Squeeznetnet detailed in **Table9.**, and Alexnet , detailed in **Table10.**

```
---------------------------------------------------------------------------
Layer (type:depth-idx)     Output Shape   Kernel    Stride    Drp     Param #
```

```
===========================================================================
├─Sequential: 1-1            --
|    └─Conv2d: 2-1          [64,36, 36]     11x11    4x4        (23,296)
|    └─ReLU: 2-2            [64,36, 36]                          --
|    └─MaxPool2d: 2-3       [64, 17,17]     3x3      2x2         --
|    └─Conv2d: 2-4          [192,17,17]     5x5      1x1        (307,392)
|    └─ReLU: 2-5            [192,17,17]                          --
|    └─MaxPool2d: 2-6       [192, 8, 8]     3x3      2x2         --
|    └─Conv2d: 2-7          [384, 8, 8]     3x3      1x1        (663,936)
|    └─ReLU: 2-8            [384, 8, 8]                          --
|    └─Conv2d: 2-9          [256, 8, 8]     3x3      1x1        (884,992)
|    └─ReLU: 2-10           [256, 8, 8]                          --
|    └─Conv2d: 2-11         [256, 8, 8]     3x3      1x1        (590,080)
|    └─ReLU: 2-12           [256, 8, 8]                          --
|    └─MaxPool2d: 2-13      [256, 3, 3]     3x3      2x2         --
├─AdaptiveAvgPool2d: 1-2    [256, 6, 6]                          --
├─Sequential: 1-3             --                                 --
|    └─Dropout: 2-14        [9216]          --       --    0.5   --
|    └─Linear: 2-15         [4096]                              (37,752,832)
|    └─ReLU: 2-16           [4096]                               --
|    └─Dropout: 2-17        [4096]                         0.5   --
|    └─Linear: 2-18         [4096]                              (16,781,312)
|    └─ReLU: 2-19           [4096]                                --
|    └─Linear: 2-20         [2]                                 8,194
===========================================================================
```

Table9. Proposed Alexnet architecture.

```
--------------------------------------------------------------------------
Layer (type:depth-idx)        Output Shape    Kernel  Stride     Param #
==========================================================================
├─Sequential: 1-1               --
|    └─Conv2d: 2-1            [64, 74, 74]    3x3    2x2        (1,792)
|    └─ReLU: 2-2              [64, 74, 74]                       --
|    └─MaxPool2d: 2-3         [64, 37, 37]    3x3    2x2         --
|    └─Fire: 2-4              [128,37, 37]                       --
|    |    └─Conv2d: 3-1       [16, 37, 37]    1x1     1x1       (1,040)
|    |    └─ReLU: 3-2         [16, 37, 37]                       --
|    |    └─Conv2d: 3-3       [64, 37, 37]    1x1     1x1       (1,088)
|    |    └─ReLU: 3-4         [64, 37, 37]                       --
|    |    └─Conv2d: 3-5       [64, 37, 37]    3x3     1x1       (9,280)
|    |    └─ReLU: 3-6         [64, 37, 37]                        --
|    └─Fire: 2-5              [128, 37, 37]                        --
|    |    └─Conv2d: 3-7       [16, 37, 37]    1x1      1x1     (2,064)
|    |    └─ReLU: 3-8         [16, 37, 37]                        --
|    |    └─Conv2d: 3-9       [64, 37, 37]    1x1      1x1     (1,088)
|    |    └─ReLU: 3-10        [64, 37, 37]                             --
|    |    └─Conv2d: 3-11      [64, 37, 37]    3x3      1x1      (9,280)
|    |    └─ReLU: 3-12        [64, 37, 37]                              --
|    └─MaxPool2d: 2-6         [128, 18, 18]   3x3      2x2              --
|    └─Fire: 2-7              [256, 18, 18]                             --
|    |    └─Conv2d: 3-13      [32, 18, 18]    1x1      1x1      (4,128)
|    |    └─ReLU: 3-14        [32, 18, 18]                              --
|    |    └─Conv2d: 3-15      [128, 18, 18]   1x1      1x1      (4,224)
|    |    └─ReLU: 3-16        [128, 18, 18]                             --
|    |    └─Conv2d: 3-17      [128, 18, 18]   3x3      1x1      (36,992)
|    |    └─ReLU: 3-18        [128, 18, 18]                             --
|    └─Fire: 2-8              [256, 18, 18]                             --
|    |    └─Conv2d: 3-19      [32, 18, 18]    1x1      1x1      (8,224)
|    |    └─ReLU: 3-20        [32, 18, 18]                              --
|    |    └─Conv2d: 3-21      [128, 18, 18]   1x1      1x1      (4,224)
|    |    └─ReLU: 3-22        [128, 18, 18]                             --
|    |    └─Conv2d: 3-23      [128, 18, 18]   3x3      1x1      (36,992)
|    |    └─ReLU: 3-24        [128, 18, 18]                            --
|    └─MaxPool2d: 2-9         [256, 9, 9]     3x3      2x2       --
|    └─Fire: 2-10             [384, 9, 9]                              --
|    |    └─Conv2d: 3-25      [48, 9, 9]      1x1      1x1      (12,336)
|    |    └─ReLU: 3-26        [48, 9, 9]                               --
|    |    └─Conv2d: 3-27      [192, 9, 9]     1x1      1x1      (9,408)
|    |    └─ReLU: 3-28        [192, 9, 9]                              --
|    |    └─Conv2d: 3-29      [192, 9, 9]     3x3      1x1      (83,136)
|    |    └─ReLU: 3-30        [192, 9, 9]                              --
|    └─Fire: 2-11             [384, 9, 9]                              --
|    |    └─Conv2d: 3-31      [48, 9, 9]      1x1      1x1      (18,480)
|    |    └─ReLU: 3-32        [48, 9, 9]                               --
|    |    └─Conv2d: 3-33      [192, 9, 9]     1x1      1x1      (9,408)
|    |    └─ReLU: 3-34        [192, 9, 9]                              --
|    |    └─Conv2d: 3-35      [192, 9, 9]     3x3      1x1      (83,136)
|    |    └─ReLU: 3-36        [192, 9, 9]                              --
|    └─Fire: 2-12             [512, 9, 9]                              --
|    |    └─Conv2d: 3-37      [64, 9, 9]      1x1      1x1      (24,640)
|    |    └─ReLU: 3-38        [64, 9, 9]                               --
|    |    └─Conv2d: 3-39      [256, 9, 9]     1x1      1x1      (16,640)
|    |    └─ReLU: 3-40        [256, 9, 9]                              --
|    |    └─Conv2d: 3-41      [256, 9, 9]     3x3      1x1      (147,712)
|    |    └─ReLU: 3-42        [256, 9, 9]                              --
|    └─Fire: 2-13             [512, 9, 9]                              --
|    |    └─Conv2d: 3-43      [64, 9, 9]      1x1      1x1      (32,832)
|    |    └─ReLU: 3-44        [64, 9, 9]                               --
|    |    └─Conv2d: 3-45      [256, 9, 9]     1x1      1x1      (16,640)
|    |    └─ReLU: 3-46        [256, 9, 9]                              --
|     |     └─Conv2d: 3-47      [256, 9, 9]      3x3      1x1      (147,712)
|     |     └─ReLU: 3-48        [256, 9, 9]                          --
├─Sequential: 1-2               [2, 1, 1]                            --
|     └─Dropout: 2-14           [512, 9, 9]                          --
|     └─Conv2d: 2-15            [2, 9, 9]        1x1      1x1       1,026
|     └─ReLU: 2-16               [2, 9, 9]              --
|     └─AdaptAvgPo2d: 2-17       [2, 1, 1]              --
============================================================================
```

Table10. Proposed Squeezenet architecture

III.3.3 Color spaces

RGB color is not always the best color space to analyze color since the three RGB channels are not independent .Furuta *et al*.[40] have shown in their paper that HSV could be a good alternative , since Hue Channel ranges from red(0),to yellow(44), and Value channel is dependent not only on the nature of the defected area but also on the intensity of light shed.

In our approach we use many color space RGB, HIS, HSV and YCbCr. Grayscale as been evaluated in image processing methods , so no need to be experimented in deep learning.

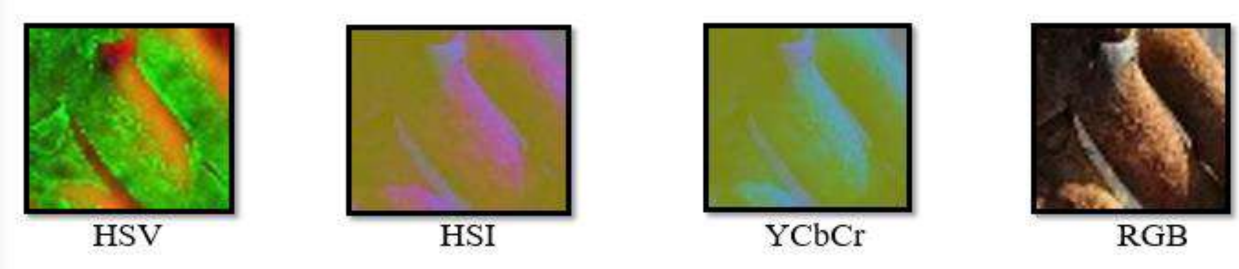


Fig30.Diffirent Color spaces investigated in the expriment

III.3.4 Model Ensemble of CNN

CNNs are nonlinear deep neural networks that learn through stochastic training error, they are capable to acquaint complex relationship between variables and targets and approximate functions that can solve a given problem. All this is done through updating weights each time models are trained. The drawback of this process is that is prone to the stochastic learning nature which results different predictions at each time of training and also to the random initialization of training data,therefore,it may produce different weights each time it is trained specially in relatively small datasets. In other words, the training data is not enough to generalize with high variance in data. Ensemble learning seeks to address these issues by combining predictions of multiple models, see Fig37, and resulting in a better performance compared to that of any individual constituent model.

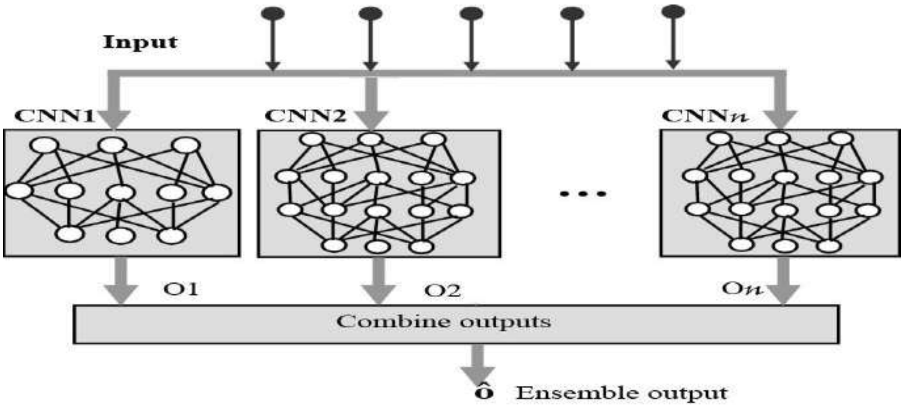


Fig34. An ensemble of convolutional neural networks

There are several ensemble approaches such as majority voting, simple averaging, weighted averaging.

Hard voting is the simplest case of majority voting. Here, we predict the class label ý via majority (plurality) voting of each classifier C_j .

$$\acute{y} = \text{mode}\{C_1(x), C_2(x), \ldots, C_m(x)\} \qquad (28)$$

We can compute a weighted majority vote by associating a weight W_j with classifier C_j:

$$\acute{y} = \text{argmax}_{i\,=1m} w_j \chi_A (C_j(x)=i) \qquad (29)$$

In soft voting, we predict the class labels based on the predicted probabilities pp for classifier -- this approach is only recommended if the classifiers are well-calibrated.

$$\acute{y} = \text{argmax}_i \ \sum_{j=1}^{m} W_j \ P_{ij} \qquad (30)$$

Where $w_j w_j$ is the weight that can be assigned to the jth classifier.

Assuming the example in the previous section was a binary classification task with class labels i∈{0,1}i∈{0,1}.

III.3.4.1 Weighted average outputs

In weighted ensemble, we optimized the weights for the model predictions that minimized the total logarithmic loss. This loss decreases as the prediction probabilities converge to true labels. We used a neural network with the tuned number of hidden layer and hyper parameters, to perform several iterations to converge to the optimal weights for the model predictions.

III.3.5 Network visualization

To have better understanding of how our networks behave, we demonstrate both the learned Convolutional filters and feature maps. The learned filters display visual patterns for each layer. The filters from the first layer show simple textures and colors, the deeper layers resemble textures found in the raw images.

The feature map visualization of each layer is seen in Figure 35. The deep layers tend to capture the texture information as we can see in two last convolution layers.

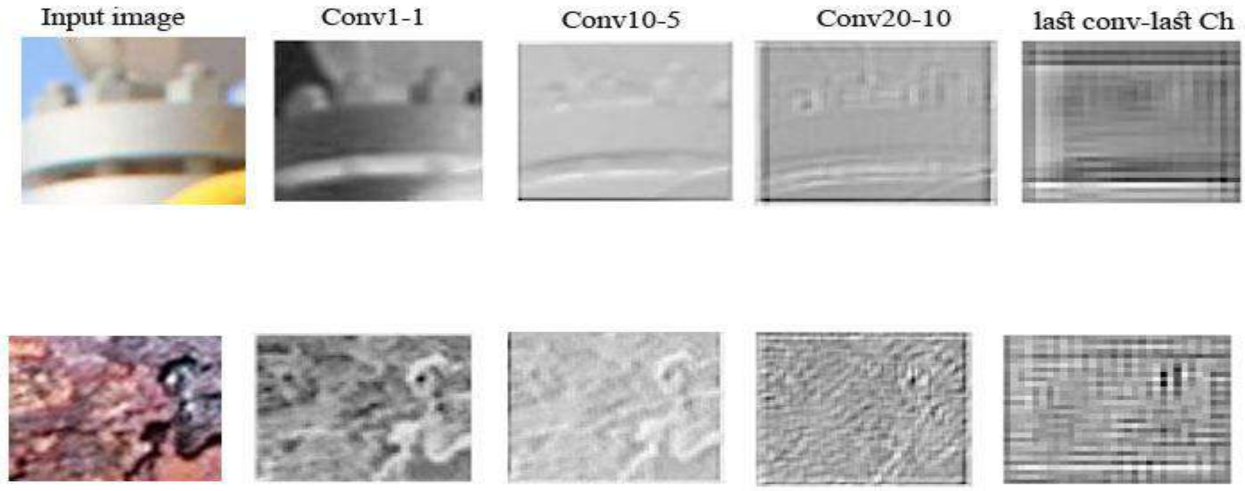


Fig35. Squeezenet Network Visualization of feature maps from four CONV blocks (denoted CONV1-1: channel number 1 block n1)

III.3.6 Sliding window

In the absence of labeling at the level of pixels , and in order to recognize the deteriorated area in images, a sliding window is utilized, see the Fig36. two sizes have been applied 64x64 and 128x128.We observed that the smaller sliding window size , the better corroded region can be localized. But , the smaller image ,the less number of features we would learn .This diminish could lead to a decrease in signal-to-noise ratio.however, the dataset image are spilt to 50x50 , so it is logical to get better result wieth 64x64 , rather than 128x128.

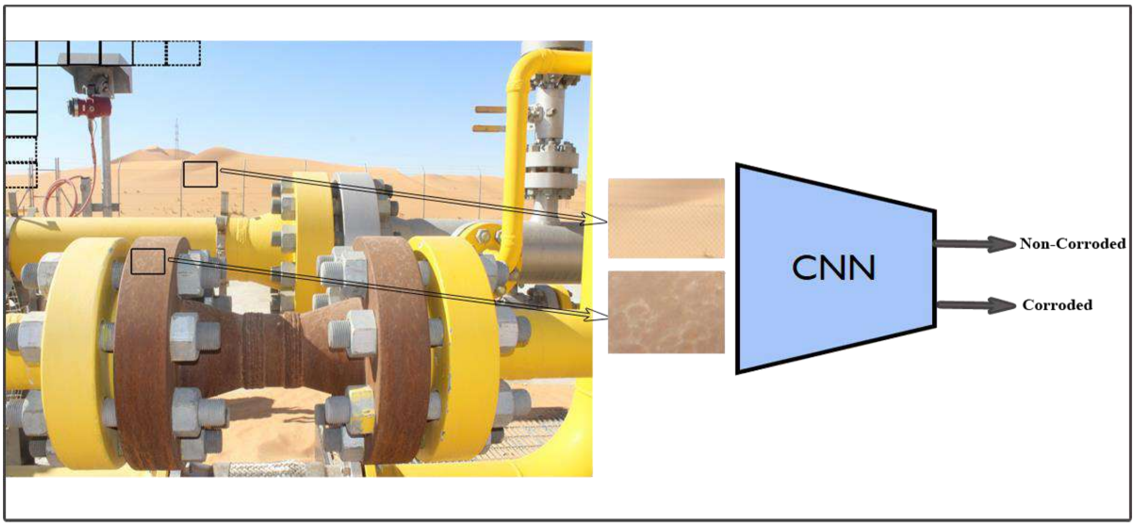


Fig36. Sliding window mechanism

III.4 Summary

In this chapter, we tried to present briefly several methods used in our experiments, as it is categorized previously , we introduced methods of handcrafted features mainly analyze color and texture , as well as, exploring the powerful tool deep learning including its techniques .

Thus, going further into details and explain them in few words and examine methods and approaches employed to tackle corrosion detection problem, to this end, we plotted some illustrations and statistics , especially when it comes to represent data or to map model's parameters, in order to get the intuition and better understanding or a given model detailed architecture , which is crucial to evaluate how deep and heavy the proposed model , and for the computational consumption aspect.

Finally, we have also presented the method we adopt in this study, and provide different stragiees used in this method.

task at hand. So for that we seek the data augmentation and transfer learning techniques. Data augmentation is a common pre-processing task which is used when there is limited training data .It can involve performing random rotations, shifts, shears, noising.. etc.

On the top of that, DL-based models takes a lot of time to train and to test also, even if we use machines endowed by high computational resources , still slower than any handcrafted feature extraction .

Summary

A lot of the CV techniques invented in the last decades, have become irrelevant and unused recently because of DL emergence. However, knowledge is never being obsolete and there is always something worth to learn from each generation of innovation. That knowledge can give you more intuitions and tools to use especially when you wish to deal with 3D CV problems for example. Knowing only DL for CV will dramatically limit the kind of solutions in a computer vision engineer’s arsenal.

In this literature review chapter , we have laid down many works for some traditional CV techniques that are still very much useful in corrosion detection problem , as well as in the age of DL there are new technique in the literature .

From the first impression, we can acquaint that DL is advantageous in many aspects, against Image-based processing, but, both approaches have weaknesses and strengths and it is too early to draw conclusions.

Chapter IV. RESULT AND DISCUSSION

IV.1 Introduction

In this chapter, we exhibit the result of different experiment conducted in proposed method chapter, we have two parts in the chapter, the first is devoted to experimental setup and the second to experimental results .the first part presents the tools and data on which we accomplish our evaluation .the second parts is consecrated to demonstrate results and analyze them in a discussion .Moreover ,We display the completion of proposed model on some images , to watch closely the performance of our production tool with new introduced data.

IV.2 Experimental setup

IV.2.1 Dataset establishment

To create a dataset from scratch , we have taken images from SHFCP (SHFCP is Oil and gas company has two facilities situated in the eastern south of Algeria about 300km from Hassi Messaoud wilaya of Ouargla.) facilities , we took photos from the structures that are in production .A very sophisticated and secured camera has been used to take high resolution photos in safe conditions ,the camera is constructed to be authorized and used in such facilities, as well , the facility has two manufactories , one for oil processing and the other for Gas.

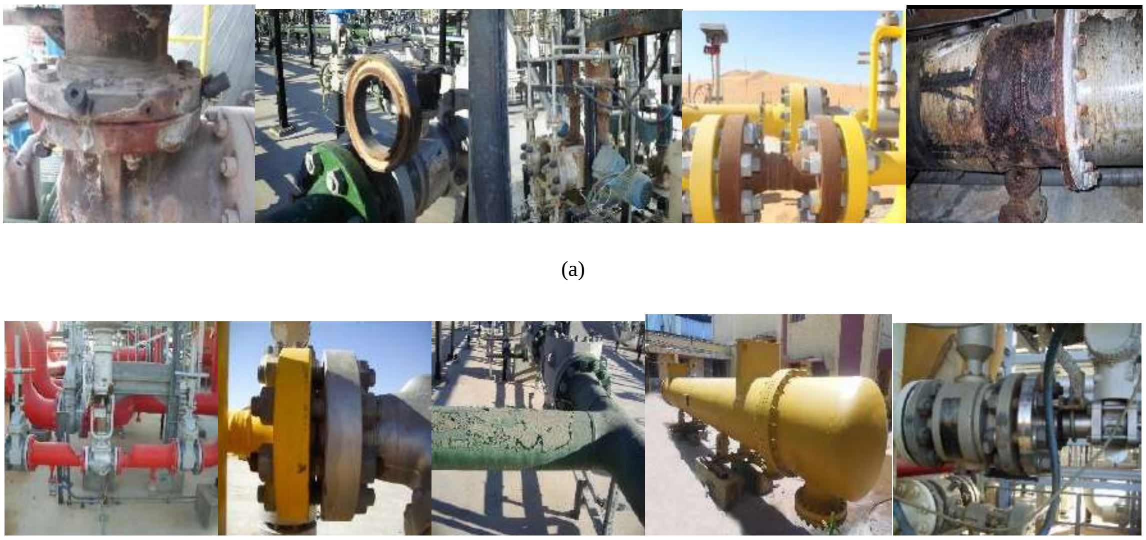
The camera reference is: ATEX / IECEx Certificate No. TRAC13ATEX0046X / IECEx TRC 13.0016x.

The detail properties are described in the table12.

General information	
Standard image capture resolution	16 Megapixel
Zoom	5 x optical, 4 x digital zoom
Screen size	2.7 inches
Additional features	Anti shake, auto focus, macro, face recognition
Tripod mount	1⁄4 inch 20 TPI

Table12: Camera ATEX proprieties

In the Fig26, you will see some examples of images that have been taken from the facilities.as we notice that the image of (a) has corroded regions ,whereas , the (b) images don’t have corrosion.



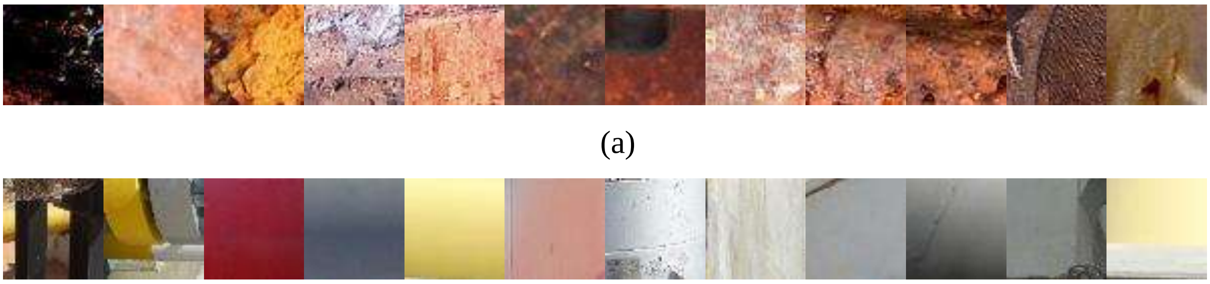
(a)

(b)

Fig37. (a) Images have corrosion zones (b) non corrosion images.

First, we sub-sampled images and split them into 50x50 pixel, Each pixel of a genuine-color image has three layers of red, green and blue. Hence, each pixel has three digital components. Overall we gathered 5000 images of corrosion ,and 5000 non corrosive , the total of our dataset is 10000 ,furthermore , any image that is corrosive only in edges have been excluded to avoid noisy data.

The final image samples are shown in fig27. After the split.



(a)

(b)

Fig39. (a) Corrosion images and (b) non corrosive images after split

IV.2.2 K-Fold Cross-Validation

In deep learning, it is vital to test models accurately , particularly, in a dataset that is relatively small , to this end, we applied 5-fold cross-validation. We randomly split the data into five partitions (20% of the training data) of equal number of images (k = 5). For each partition n, we trained the proposed model on the remaining four partitions, and tested it on partition n.

The final score was the average of all five scores obtained. The illustrative of 5-fold validation is shown in Fig28.

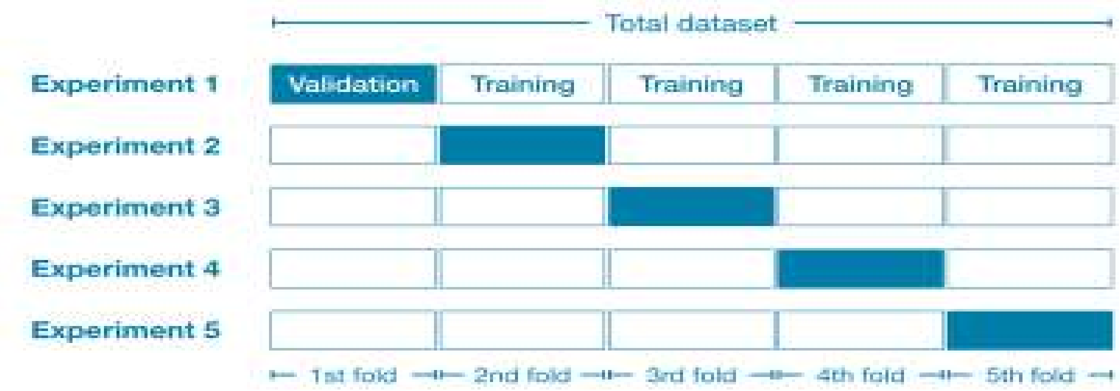


Fig40. 5-fold Cross validation applied on the dataset

IV.2.3 Performance metrics

In this thesis, we consider the following measures, precision (26),recall(25), F1 Score(27).True positives , False negatives , false positives are used to calculte Recall and Precision. Whereas ,F1 Score is measure of test's accuracy that consider both the precision and recall to compute the score.

$$Recall = \frac{\sum true\ positive}{\sum true\ positive + \sum false\ negative} \qquad (25)$$

$$Precision = \frac{\sum true\ positive}{\sum true\ positive + \sum false\ positive} \qquad (26)$$

$$F1\ Score = \frac{2\ X\ recall\ X\ precision}{recall + precision} \qquad (27)$$

IV.2.4 Developpement Envirement

The most of works have been done with a personal computer endowed by graphic processing .Generally ,Image processing techniques don’t need a lot of resources as the feature extraction consume only memory but sometimes need resources in processing dependent to image resolution and computational nature of some CV algorithms . However, deep learning requires huge processing resources and that can be met only if we have been helped by a GPU processor , the large amount of data in RGB images has to be treated and transferred to the GPU speedily , for that , it is also vital to have a large memory namely RAM.

It is worth to say that some workshops were achieved by a virtual machine hosted on a Server in my workplace , its specifications are described here :CPU Intel(R) Xeon(R)E5506 8 x 2.133GHz and Ram 64GB. As below you find the configuration used to complete the main work.

IV.2.4.1 Hardware Configuration.

Motherboard Gaming MSI b85-g43

CPU Intel i5-4590S @3.0 GHz

RAM 16 GB

GPU:Nvidia Geforce GTX-1060 6G .

IV.2.3.2 Software Configuration.

OS: Windows10 64x .

Python 3.6

CUDA NVIDIA CUDA Toolkit 10.0

Python Python 3.6.10 (Anaconda 3.17.8)

Pytorch 1.4.0

TorchVision 0.5.0

We have written the code of all our experiments with Python language which is a powerful language whether for image processing or Deep learning , because of its capabilities and scalability , it can be suitable for multitude works namely machine learning algorithms , these algorithms have been developed to be utilized so easily. The installation of the libraries are setup with Anaconda, this cloud platform helps to choose the right libraries matrices, it is also useful in creating virtual environments and therefore switch easily between them, to avoid any conflicts in frameworks of same nature.

We used also Cuda to accelerate processing in the deep learning techniques , and Torchvision to download pre-trained models and datasets from the their location in the net.

IV.3 Experimental results

IV.3.1 Deep learning

This section details the average result of validation subsets with different color spaces, the recall , precision and F1score on the validation folds are evaluated.

Equations for recall, precision, and F1 score can be seen in equations (25) to (27), respectively. F1 measurement is an evaluation of classification performance that considers both the precision and recall. It is used generally in binary classification, the calculation is the harmonic mean of precision and recall where the arithmetic average on the reciprocal of precision and recall is performed.

We have also investigated the effect of color spaces, RGB ,HSV , HIS and YCbCr , we tried to watch closely If other space colors can get better results .

Color Space	**Recall**		**Precision**		**F1 Score**	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
RGB	**97.94**	1.56	**97.71**	1.65	**97.82**	0.62
HSV	93.98	2.45	94.45	2.07	94.18	2.89
HSI	96.43	1.67	96.21	1.87	96.31	2.09
YCbCr	92.50	3.45	92.56	3.38	92.52	3.98

Table13. fine-tuning results of the proposed model CROGF7.

Color Space	**Recall**		**Precision**		**F1 Score**	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
RGB	**98.37**	1.76	**98.43**	1.55	**98.39**	1.62
HSV	95.98	3.85	94.67	2.89	95.32	3.23
HSI	95.45	2.27	96.21	2.78	95.82	2.56
YCbCr	93.12	1.55	92.41	1.85	92.76	1.12

Table14. Fine-tuning results of the proposed model Squeezenet.

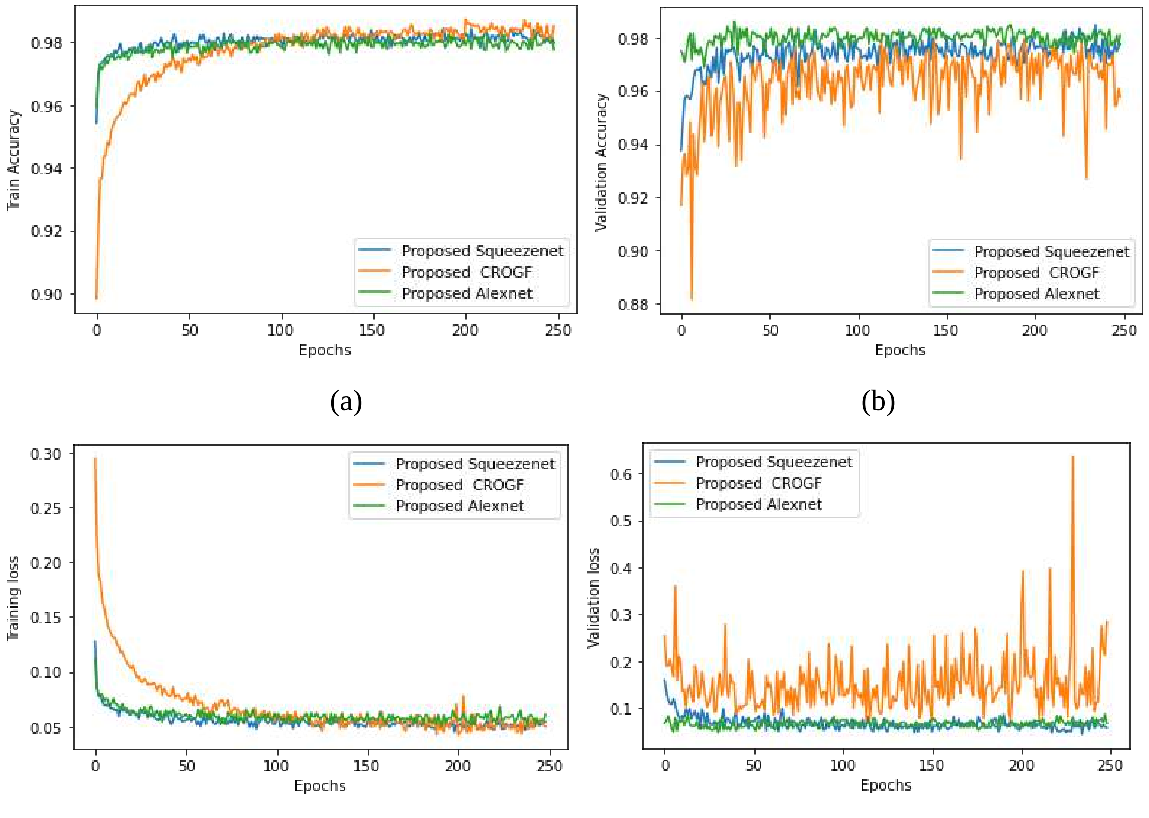
Color Space	**Recall**		**Precision**		**F1 Score**	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
RGB	**98.80**	0.78	**98.54**	0.84	**98.66**	1.01
HSV	95.18	2.13	95.45	2.59	95.31	1.19
HSI	96.03	1.34	96.39	1.13	96.20	1.32
YCbCr	93.40	3.89	93.56	3.90	93.48	2.83

Table15. Fine-tuning results of the proposed model Alexnet.

To compare the effect of CNN architecture the three models CROGF7 , Alexnet and Squeezenet have been employed .Fine tuning of Alexnet and Squeeznet gives better result than pretrained , because we have unleashed all weights in FC layers to be trained for the task given.CROGF7 has shown the less performance due to the number and thickness of convolution layers ,as we have set only 4 convolutional layers and the kernels are of 5x5 and 3x3 respectively. Alexnet with fine-tuning outperformed CROGF7 and Squeezenet with mean F1 score of 98.80

To evaluate the color spaces performance , we noticed that RGB color still the best for our task, HSI gives better result than the other color spaces but still doesn’t outperform RGB .we observe also that dispersion is large with HSV and YCbCr color spaces ,means the results are not steady during the test, in contrast of RGB and HIS which yield more convergent scores .

Alexnet has the better result among the three proposed models, and has the least dispersion measured by standard deviation.



(a) (b)

(c) (d)

Fig41.models performance against iterations: (**a**) Training accuracy plot; (**b**) Validation accuracy plot; (**c**) Training loss plot; (**d**) Validation loss plot.

Hyper-parameters in CNNs are vital to improve model’s performance. Optimizers are algorithms used to alter the attributes of your neural network such as weights and learning rate in order to reduce the losses, choosing the right optimizer could lead your model to better performance .In our evaluation, we employed three optimizers (Stochastic Gradient Descent ,Adamx and RMSprop). In the fig40. the effect of optimizer is seen clearly , SGD and Admax yield better performances in both training and validation , and their losses are lower ,however, RMSprop’s graph is closer in training and validation accuracy but shows more loss values in training and validation.

One of the most important hyper-parameters of neural networks is the learning rate of the optimizer , which define how quickly a neural network model learn a given task . Learning rate is crucial parameter in learning process , Fig41. shows the effect of the learning rate on the training and validation history. it is better to be a small value to avoid oscillating this phenomenon ,the latter occurs when the learning rate is large and consequently produces large weights, we see it clearly in the training and validation losses of the learning rate of 0.01. However when the learning rate is set to 0.001, the learning increases gradually and the oscillations are fewer. For learning rate of 0.0001 ,the training and validation is not oscillating but it shows slow learning and even the performance is not high as the learning rate of 0.001.

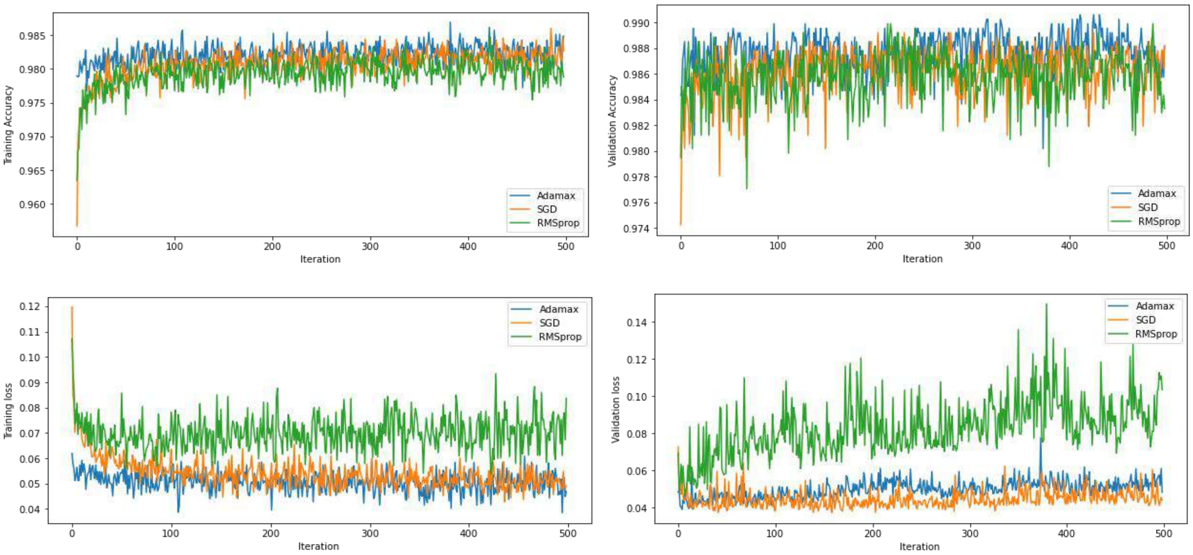


Fig40. Effect of optimizer on the individual model training and validation accuracy and loss

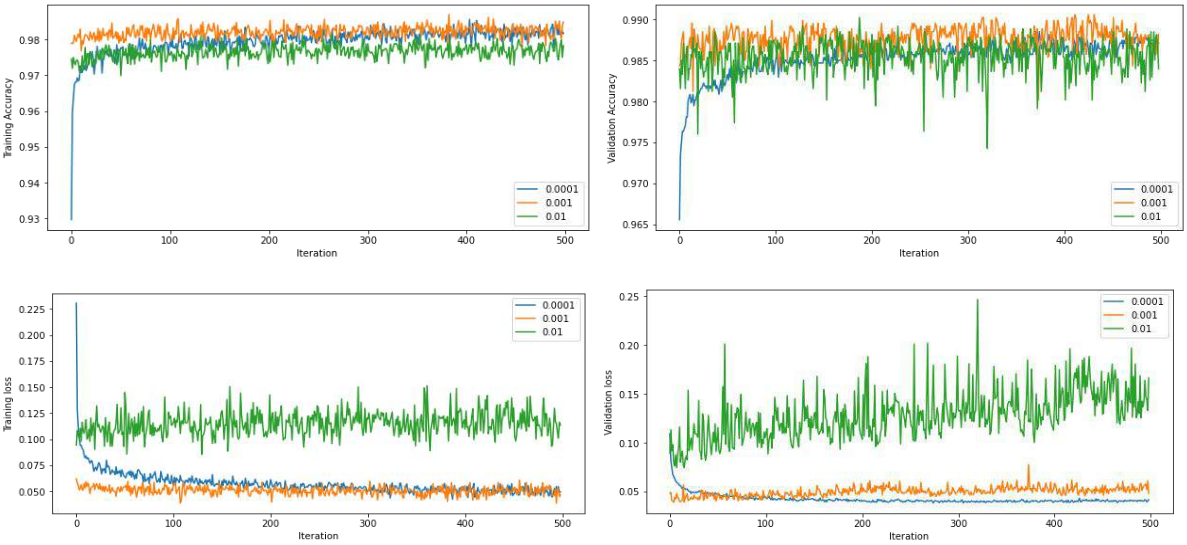


Fig42. Effect of learning rate on individual model training and validation accuracy and loss

In Fig42. We observe the effect of batch size on the training and validation history .With a big batch size and small one ,we can notice that there are close performance, as seen with batches of 8 and 32, the training and validation processes are smooth and converge quickly .Contrary to 8 and 32, the batch size of 16, the training and validation suffer drops and losses in training and validation are the bigger too .

In summary , it is known that smaller batch size produces faster learning process but it does not affect the accuracy, but in our case even batch size of 32 gives good results .

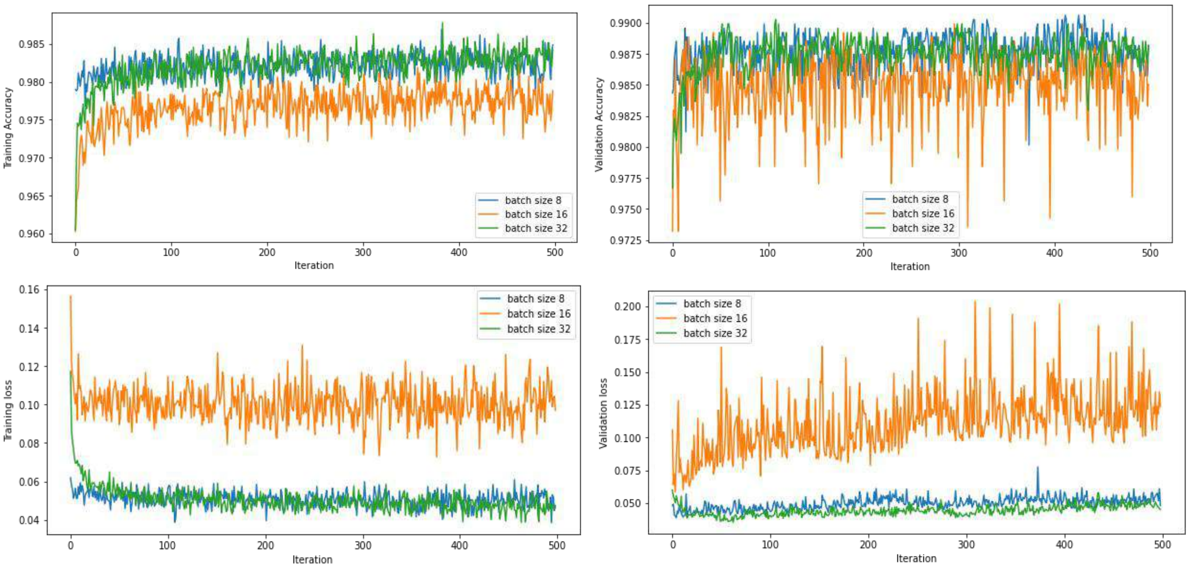


Fig43. Effect of batch size on individual model training and validation accuracy and loss

VI.3.2 Ensemble model

Combining multiple model’s outputs to build a powerful model is one of the techniques that shows better result than each models separately, besides, ensemble model can produce a varied and balanced weights during the train. Another reason is that our dataset is relatively small so that the predictions would be more accurate if we share the decision with other models . In other words, ensemble networks have a high variance than single ones ,in our case we tried to combine the three models , which are respectively ,CROGF7, Squeezenet and Alexnet.

We constructed ensembles of the top-3 performing fine-tuned CNNs to evaluate for an improvement in predicting the corrosion . We used majority voting, simple averaging, and weighted averaging strategies toward this task. But the best method reported is weighted averaging, we optimized the weights for the model predictions to minimize the total logarithmic loss. You can see results of the applied three methods in the table20.

Ensemble methods	Recall	precision	F1 Score
Majority voting	98.87	98.73	98.79
Simple averaging	98.74	98.77	95.75
Weighted averaging	**98.92**	**98.78**	**98.84**

Table16. Ensemble methods comparison

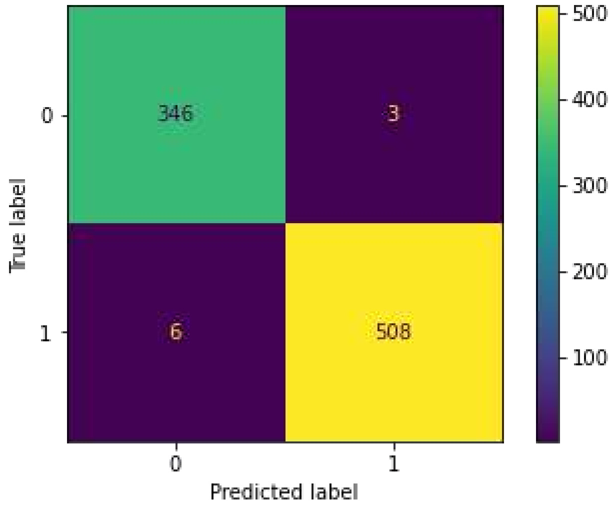


Fig44. Confusion matrix of weighted averaging ensemble method .

Below you find below the performance of the weighted averaging method over training and validation accuracy and loss, we have tuned our neural network to reduce the loss and therefore yield better result in fig 43.you find the confusion matrix which details the different true and false prediction , the results also proves that the false positive predictions of corrosion are only three and this is good precision.

IV.4 Comparison with image-based processing methods

IV.4.1 Color moment

Color moment experiment has produced a vector of 6 columns and the seventh is the label With the help of KNN (K-Nearest neighbors) to predict the class of the image, the result yielded is shown in the graph below Fig38.

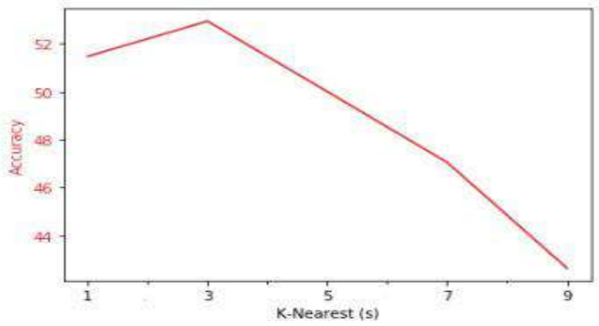


Fig45. The prediction of class accuracy compared to K-nearest neighbors (nearest neighbors)

K-neghbors	Score
1	0.8865
3	0.8326
5	0.8012
7	0.7762
9	0.7365

Table17. Color moments scores with KNN .

From the result shown in the graph Fig4, we can conclude that the color moment cannot be a classifier for the corrosion as the three-channel RGB properties have near values in both corroded and non-corrosive images. And KNN best result with three neighbors .

IV.4.2 Red Channel average

The table below Table12. Demonstrates s clearly the limit of red average ,and that might due to the nature of oil and gas equipment which can be found habitually coated by paint. Furthermore, Corrosion is not only characterized in red color but it ranges from bright orange to dark brown and sometimes black spots .

Penalty Coefficient(C)	degree	Gamma	kernel	Score
1	1	Scale	rbf	0.9014
5	2	Auto	linear	0.8925
10	3	Scale	poly	0.8735
15	4	Auto	sigmoid	0.8241

Table18. SVM scores against differentiating SVM hyper-parameters.

IV.4.3 Comparison with handcrafted features

IV.4.3.1 GLCM and GRLM

Corrosion detection based on the texture feature extraction with GLCM , GRLM and color properties proposed in the paper [23], for a small-seized image samples can achieve good result . Demonstration of the feature extraction phase which computes data texture information is provided in Table 3 , 4 .

Herein, for each image sample, 71 features representing the statistical measurements of image colors, GLCM, and GLRL are attained and used for data classification purpose. In addition, the evolutionary process of the DFP metaheuristic-based SVM model optimization which is replaced by SearchGrid of DEAP Sklearn library is illustrated in Figure 27 . It is noted that the corrosion detecteion with this proposed approach can achieve relatively good classification result. Nevertheless, several positive samples located in the boundary of the corroded area have not been identified correctly.

Table9. Below show the result of method of texture analysis , we can deduce that texture feature extraction with GLCM and GLRM and color properties can give good result compared to other approaches in image processing , the result of table13. Is for the validation data ,thus , 95.44 as F1-score is good result but we still need to try other texture descriptors.

	Precision	Recall	F1-score
Texture analysis	95.56	95.34	95.44

Table19. Below shows the result of other handcrafted features for texture analysis.

Other Texture descriptors have been employed on the same dataset , LBG ,HOG and GLCM .

IV.4.3.2 HOG and LBP

LBP and HOG show result underneath eighty , and combination of both descriptors shows 0.81 as result, but the best result recorded when GLCM is combined with LBP and HOG , besides the SVM was the top classifier against K-nearest neighbors and Gaussian naive bayes ,it is noted also that GridSearch has been used to get the best hyper-parameters of all the classifiers.

	LBP	HOG	LBP+HOG	GLCM+LBP+HOG
SVM	0.787879	0.739019	0.814346	0.899036

KNN	0.756897	0.706784	0.715687	0.759476
GNB	0.742998	0.718969	0.756881	0.736776

Table20. handcrafted ’ scores with different ML algorithms

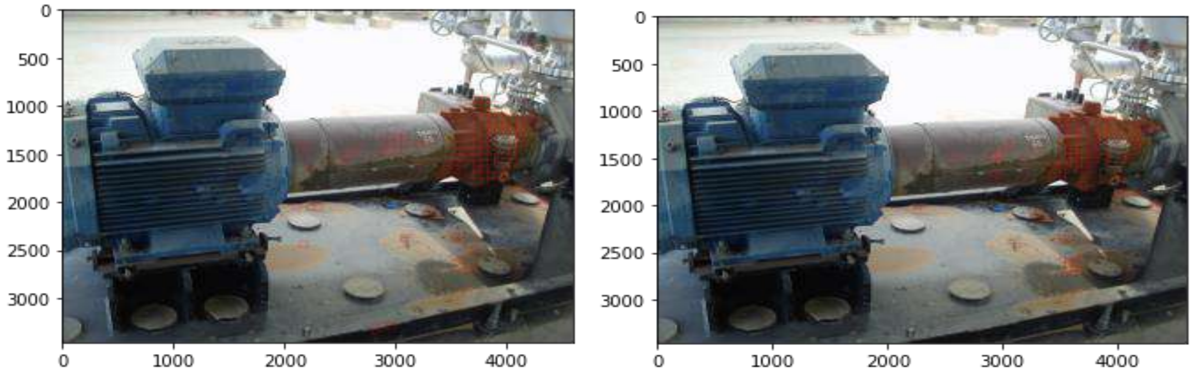
The texture analysis is a robust method in image processing, especially if we use GLCM , one of the best texture descriptor , however, it cannot resist the image variation nor data augmentation, we can clearly see that in the performance with image augmentation, the precision and recall regresses by two numbers in the corrosion images.

On the top of that, texture Analysis can yield very high results in training phase, but when it comes to test new data , it performs poorly .

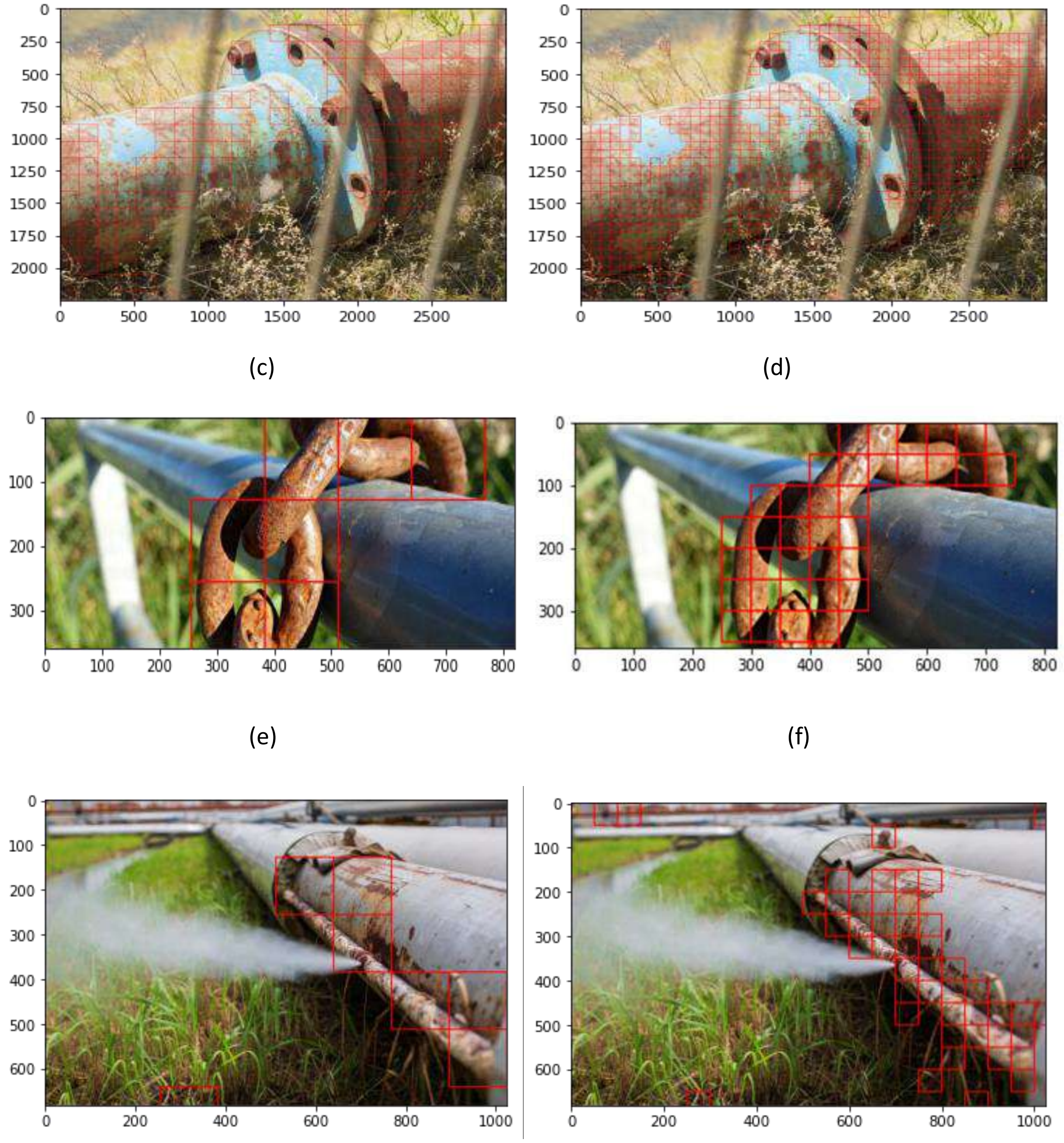
IV.5 Sliding Window performance

The Fig44. Shows the sliding window outputs on new introduced images to evaluate our model recognition of corrosion area in both image ,the first mage is taken from a similar environment of oil and gas equipment , however , we choose the second image because it has rough texture and so we see how powerful our model to resist in such image texture roughness . We observe also that the smaller sliding window we apply the best accuracy we get, 128x128 can perform well but it often mistaken some windows which have small spot of corrosion .

The third image with relatively low resolution, demonstrates that our model can be effective even on windows that are not fully covered by corroded regions , and also can resists the contrast which is clearly seen on the light of sun brighten the metal.



(a) (b)


0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
2500
0
100
200
300
0
100
200
300
400
500
600
700
800
0
100
200
300
400
500
600
0
200
400
600
800
1000

(c)

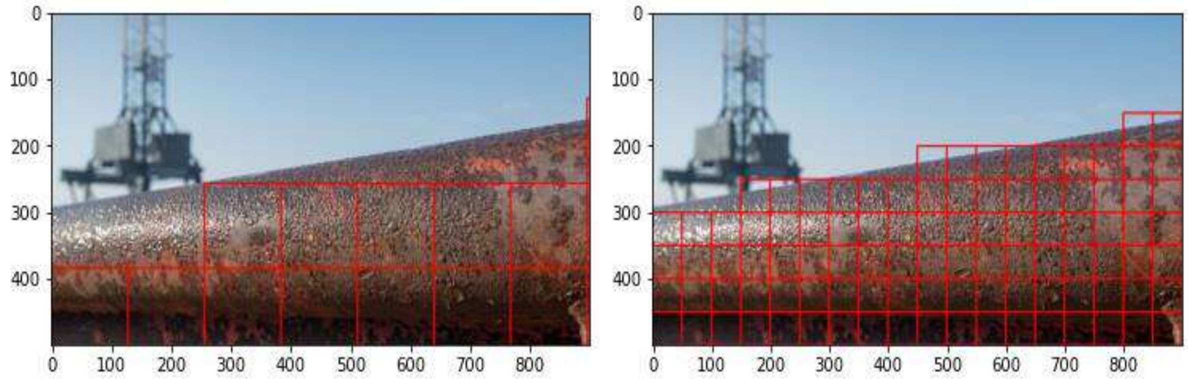
(d)

(e)

(f)

(g)

(h)



(i)

(j)

Fig46. (a) ,(c) ,(e) , (g) and (i) sliding window of 128x128 along with (b) ,(d) ,(f) , (h) and (i) sliding window of 50x50

The fourth image scanned by our algorithm signifies that it can detect corrosion pitting which causes the leakage of gas , as we know that pitting is the most dangerous form of corrosion . The fifth image is ordinary corrosion with the contrast of sun, the latter couldn't be noisy or confuse our model ,consequently, resisting variation in color or contrast and being so steady in different locations and situation is the purpose of building ensemble CNN .

IV.6 Final results and summary

By exploring several methods and techniques, Table19. Reports the comparison of the accuracy, recall and F1-score of several approaches and methods that have been studied Previousely to make the global comparison

Method	*Recall*	*Precision*	*F1 Score*
Color moment	54.32	53.52	53.91
Red average	63.35	63.41	63.37
Texture analysis	95.56	95.34	95.44
CNNs Fine-tuning	98.75	98.48	98.61
Our method(ensemble)	**98.92**	**98.78**	**98.84**

Table21. Previous methods comparison of results

The table18. above shows that ensemble model have the best result among the rest of methods, It outperforms the fine-tuning , and the other methods of image-based processing .

Model ensembles further improved qualitative and quantitative performance in corrosion detection. Ensemble learning tries to bridge the gap of the random initialization of individual models by combining their predictions and reduce prediction variance to the training data. Observed that the weighted averaging ensemble of the top-3 performing fine-tuned models delivered better performance compared to any individual constituent model.

The results in Table19. demonstrate the final results and the comparison of other methods .So the corrosion detection task benefits from an ensemble of repeated Squeeznent , Alexnet and the proposed CROGF fine-tuned models.Ensemble learning also compensates for the precision and recall in fine tuning of individual methods .

Empirical evaluation shows that ensemble models scores outperforms other models and it slightly outperforms individual CNN models .

To summarize, ensemble model is advantageous for corrosion detection , it profits from individual CNNs error rates and reduce them,for that end , we can deduce that the proposed model not only attains a high accuracy but is also robust , the weighted averaging strategy is good method to minimize the log loss and converge to the accuracy by balancing models with the learnt weights . Compared with all other previous methods, the proposed method achieved the best results.

GENERAL CONCLUSION

This thesis presented a comparison study of image processing methods and deep learning , we examined many methods in image processing , we found out that texture extraction with GLCM and GRLM is sophisticated method to recognize corrosion by analyzing texture discriminants, but as it is known generally in handcrafted features ,they have a lot of drawbacks due of their venerability to variation and contrast and it can't resist different image resolutions, furthermore, contrary to the training phase which yields high scores , in test data , texture analysis performs poorly .We also tried other texture descriptors , but no one outperforms the GLCM and GRLM .

On the other hand ,we designed a CNN model with a suitable number of convolutional and fully connected layers. We also tried out fine-tuning of two successful networks (Squeezenet , Alexnet) , they have given close results , our proposed model is the best in training but Alexnet has the best performance in new introduced data . we bid K-fold cross validation on our dataset, to have more balanced evaluation, as the training and test data differentiates five times in iterations .

We applied a model average ensemble, which separately trained several models with the same architecture and then combined their predictions for testing. The reason why the model average ensemble achieves good performance is that different CNN models , usually doesn'nt make all the same errors on the test set. So multiple models compensate for errors of the other ones. Therefore, a model average ensemble of CNN models performs better than individual ones.

We experimented with a new created dataset, on which we created from scratch .This dataset is relatively small compared to the standard in deep learning however it gave us good performance once it is tested with new introduced images . We proposed a tool for corrosion recognition, this is very useful to spotlight defected region in given image. We noticed that smaller window could lead to better accuracy.

Corrosion of Oil and gas equipment and pipelines hiders the production process of the industry, hence, the early detection could be advantageous in way to prevent accident and treat metal with proper ways to mitigate rusting process.

In this end. We suggest an automatic system for corrosion detection by using CCTV , drones or unmanned vehicles , to replace visual human inspections, this latter is costly and routine and time-consuming.

Defect detection in computer vision is vast field, it can include the detection of defects and damages (cracking,pitting, spalling, defective joints, corrosion, etc.) existing on civil infrastructure or industry plants such as oil and gas facilities, buildings, bridges, roads, pipes and tunnels, and the defects' dimensions (number, width, length, etc.), or manufacturing defects for instance (plastic ,cars…etc) . The state of inspected object assessment results are very useful to predict future conditions, to support investment planning, and eventually, allocate limited maintenance and repair resources.

Future perspectives

Any study in the field of industry can bring an added value, however, this value would be multiplied if we realize a system that replaces human inspection in the field.

There are a lot of ways to implement our system in the real world, for example use Raspberry Pi board , attached with high resolution camera , we can also connect it to the network and start dealing with taken image in real time .Furthermore, we can use an existed CCTV system by sending image periodically afterward ,AI model will treat sent images .

Below , we have some notes as suggestions :

- As we declare no conflict of interest, our dataset can be a benchmark for other studies, we will let it available for the community .Our dataset establishment respects standards in this term, and we are open to critics .
- With the support of my supervisor Doctor Oussama Aiadi , and Doctor Khaldi Belal our thesis can be summarized in a paper and sent to the scientific community for approval , and we hope that this paper finds its way in an esteemed journal.
- Scientifically speaking, the problem of corrosion still has a lot aspects to be researched on , for instance, we can define other classes of corrosion, the level of defect high , medium, shallow and fine .
- Other forms of defects can occur in oil and gas structures such pitting corrosion, erosion and deforms. each aspect of the previous defects can be alone a field of study.

REFERENCES

1. Koch G, Varney J, Thompson N, et al. International measures of prevention, application, and economics of corrosion technologies study. Houston, TX: NACE International, 2016.
2. WCO Shanghai 2019 declaration pubished on the website of the WCO orgabnization.

3. Handbook of Environmental Degradation of Materials

4. Master thesis , « Scénarii d'Incendie - Explosion au niveau du complexe pétrochimique de Skikda ». Etude de cas: Unité éthylène. Présenté par Chafik Guerzi » , Master thesis .

5. Master thesis , « Scénarii d'Incendie - Explosion au niveau du complexe pétrochimique de

   Skikda ». Etude de cas: Unité éthylène. Présenté par Chafik Guerzi » page 67.

6. Motamedi, M., F. Faramarzi, and O. Duran. *New concept for corrosion inspection of urban pipeline networks by digital image processing*. in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. 2012. IEEE.
7. Bondada, V., D.K. Pratihar, and C.S. Kumar, *Detection and quantitative assessment of corrosion on pipelines through image analysis.* Procedia Computer Science, 2018. **133**: p. 804-811.
8. Ji, G., Y. Zhu, and Y. Zhang, *The corroded defect rating system of coating material based on computer vision*, in *Transactions on edutainment VIII*. 2012, Springer. p. 210-220.
9. Jiménez–Come, M., I. Turias, and F. Trujillo, *An automatic pitting corrosion detection approach for 316L stainless steel.* Materials & Design (1980-2015), 2014. **56**: p. 642-648.
10. Nash, W.T., et al., *Automated Corrosion Detection Using Crowdsourced Training for Deep Learning.* Corrosion, 2020. **76**(2): p. 135-141.
11. Petricca, L., et al. *Corrosion detection using AI: a comparison of standard computer vision techniques and deep learning model*. in *Proceedings of the Sixth International Conference on Computer Science, Engineering and Information Technology*. 2016.
12. Yao, Y., et al., *Artificial intelligence-based hull structural plate corrosion damage detection and recognition using convolutional neural network.* Applied Ocean Research, 2019. **90**: p. 101823.

13. Atha, D.J. and M.R. Jahanshahi, *Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection.* Structural Health Monitoring, 2018. **17**(5): p. 1110-1128.
14. Gonzalez, R.C. and R.E. Woods, *Digital Image Processing, Global Edition*. 2017, Pearson Education Limited.
15. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. 2005. IEEE.

16. Acosta, M.R.G., J.C.V. Díaz, and N.S. Castro, *An innovative image-processing model for rust detection using Perlin Noise to simulate oxide textures.* Corrosion science, 2014. **88**: p. 141-151.
17. Choi, K.-Y. and S. Kim, *Morphological analysis and classification of types of surface corrosion damage by digital image processing.* Corrosion Science, 2005. **47**(1): p. 1-15.
18. Bastian, B.T., et al., *Visual inspection and characterization of external corrosion in pipelines using deep neural network.* NDT & E International, 2019. **107**: p. 102134.
19. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine learning, 1995. **20**(3): p. 273-297.
20. Humeau-Heurtier, A., *Texture feature extraction methods: A survey.* IEEE Access, 2019. **7**: p. 8975-9000.
21. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee.
22. Haralick, R.M. and L.G. Shapiro, *Computer and robot vision*. Vol. 1. 1992: Addison-wesley Reading.
23. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International journal of computer vision, 2015. **115**(3): p. 211-252.
24. Jahanshahi, M.R. and S.F. Masri, *Parametric performance evaluation of wavelet-based corrosion detection algorithms for condition assessment of civil infrastructure systems.* Journal of computing in civil engineering, 2013. **27**(4): p. 345-357.
25. Galloway, M.M., *Texture analysis using grey level run lengths.* STIN, 1974. **75**: p. 18555.
26. Ejimuda, C. and C. Ejimuda. *Using Deep Learning and Computer Vision Techniques to Improve Facility Corrosion Risk Management Systems 2.0*. in *SPE Nigeria Annual International Conference and Exhibition*. 2019. Society of Petroleum Engineers.
28. O'Mahony, N., et al. *Deep learning vs. traditional computer vision*. in *Science and Information Conference*. 2019. Springer.
27. Hoang, N.-D. and V.-D. Tran, *Image processing-based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach.* Computational intelligence and neuroscience, 2019. **2019**.
29. Fukushima, K., *Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron.* IEICE Technical Report, A, 1979. **62**(10): p. 658-665.
30. LeCun, Y., et al., *Backpropagation applied to handwritten zip code recognition.* Neural computation, 1989. **1**(4): p. 541-551.
31. Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *ICML*. 2010.
32. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.
33. Iandola, F.N., et al., *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size.* arXiv preprint arXiv:1602.07360, 2016.
34. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
35. Huang, G., et al. *Densely connected convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
36. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.

37. Canziani, A., A. Paszke, and E. Culurciello, *An analysis of deep neural network models for practical applications.* arXiv preprint arXiv:1605.07678, 2016.

38. Hinton, G.E., et al., *Improving neural networks by preventing co-adaptation of feature detectors.* arXiv preprint arXiv:1207.0580, 2012.

39. Vapnik, V.N., *An overview of statistical learning theory.* IEEE transactions on neural networks, 1999. **10**(5): p. 988-999.