

Application of CNN for multiple phase corrosion identification and region detection

Oluseyi Ayodeji Oyedeji^{*}, Samir Khan, John Ahmet Erkoyuncu

School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Cranfield, United Kingdom

HIGHLIGHTS

- Implementation of a binary CNN model that can diagnose the presence of corrosion on alloy materials.
- Conceptualisation of stages of corrosion in alloy materials.
- Implementation of a multiclass CNN model that can diagnose corrosion defect stages.
- Implementation of corrosion severity detection model.
- Implementation of damage region detection model to complement the other stages.

ARTICLE INFO

Keywords:

Corrosion
Convolutional neural network (CNN)
Reliability
Maintenance
Degradation

ABSTRACT

Corrosion is a significant issue that contributes negatively to the degradation of materials most especially metals. To ensure proper maintenance, enhance reliability and prevent breakdown, it is very essential to not only effectively detect corrosion but to also understand its locations and distributions on the materials. A Multiple phase Convolutional Neural Network (CNN) model is created for this purpose. The Multiple phase CNN model consists of custom designed deep learning algorithms at various stages. This created the opportunity to make use of binary classification, multi-label classification and patch distribution algorithm to detect and identify corrosion regions on metallic materials. Six (6) different labels of corrosion were modelled to represent different levels of degradation using 600 anonymized images. The images were used in the various stages of the framework for training the respective models. Results at the binary level shows 94.87 % of corrosion detection. The multiclass stage of the Multiple phase CNN records the highest accuracy of 92.1 %. The patch distribution stage recorded a highest accuracy of 96.5 % and 94.6 % for the Average Image and Average Pixel ROCAUC (Region of Concentration Area Under Cover). It also shows a region segment average accuracy detection of 91.5 % (image level) and 89.2 % (pixel level) for 9 distinct regions. The research provides a comprehensive and detailed reliability and maintenance information for Aerospace, Transport and Manufacturing Materials experts and non-experts. The framework shows a robust approach to detecting corrosion which is essential for critical and safety applications as well as preventing economic loss due to corrosion. This can also be extended to other domains beyond the corrosion case study.

1. Introduction

Corrosion affects Aerospace, Transport and Manufacturing systems because a substantial constituent of these systems is the alloy material [19,23]. Ranging from aircraft fuselages [32], automobile engines, wheels, to manufacturing tools and equipment, alloy is a common denominator in all these systems [33], and it is usually susceptible to corrosion defect. Corrosion happens when metallic surfaces encounter

chemical substances such that a pure metal is converted into its more stable form. In other words, Corrosion is a mechanism that only affects the surface of materials [12]. Corrosion of metal parts of equipment will have a negative impact on its operating characteristics, which primarily include electrical or mechanical characteristics depending on the type of equipment [14].

The effect of corrosion cannot be over-emphasised in Aerospace and Automotives. In fact, corrosion is likely to develop in aircrafts as

^{*} Corresponding author.

E-mail address: ayodeji.oyedeji@cranfield.ac.uk (O.A. Oyedeji).

<https://doi.org/10.1016/j.asoc.2024.112008>

Received 21 October 2023; Received in revised form 21 February 2024; Accepted 6 July 2024

Available online 23 July 2024

1568-4946/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

protective coatings fail especially when aircraft fleets are operated past their original design life, which will likely result in higher corrosion-related expenditures [13]. Corrosion is a huge cause of economic loss and system breakdown. For example, corrosion of metals such as steel and cast iron has been proven to be a serious cause of pipe failures [9]. There is a direct relationship between corrosion and deterioration of mechanical property. This shows that it is indeed a Global problem. Depending on the nature and intensity of the corrosion as well as the materials involved, the effects of corrosion can be profound and wide-ranging. Damage to structures, equipment failure, loss of functionality, contamination, and aesthetic issues are some frequent outcomes but in general, corrosion's effects can be expensive, both in terms of monetary costs and dangers to people's safety, health, and the environment. It is therefore essential that drastic measures be taken to prevent or control the effect of corrosion on metals.

In terms of understanding the presence or absence of corrosion, due to subjection to certain conditions and exposures, components cannot exactly be placed within a static box when conceptualising their life-cycle. It is therefore not always accurate to assume that all new components would be free from corrosion while older components would be corroded. There might be situations where maintenance check based on expert information might not be accurate due to human error or when the damage is small. At other times, experts will have to go through rigorous processes to detect a potential risk. In other words, there is a huge limitation to the effectiveness of manual inspection of component for corrosion damage detection. Meanwhile, there has been a huge availability of image data that which can be leveraged upon to reduce the heavy dependence on expertise for identification and detection of corrosion timely and effectively.

This research therefore tends to investigate the health or condition of a mechanical component based on its current visual representation at a given time. This current condition is monitored using the image of that component. There is a dearth of research in combining different deep learning methods to provide multiple and robust corrosion information by simply visualising the surface of a component within the aerospace, transport, or manufacturing domain that shows an in-depth health information about the status, stage, severity, and exact region. Many of the related works have either made use manual inspection or the use of maintenance information in form of text to make important predictions, the latter is a limited option because it means that information not captured in the maintenance log would not be captured within the model. Other similar research that are image-based simply captured defects on the binary level and are not robust enough to accommodate other variables. Most of the works that explored multiple class of corrosion did not detect the location of the damage. Hence, the major contribution of this research is the multiple phase corrosion detection and specific region identification. In addition to the combination of multiple corrosion detection approach, this research also provides a robust approach that goes beyond a specific context but applicable all alloy based materials regardless of the domain or industry.

The knowledge of the condition of a component or material beyond the creation date or even in the absence of the creation date is a priceless information as it would not only tell the health of the component but also help to decide which decision to make in terms of maintenance at that point. Convolutional neural network (CNN) which is a type of neural network is known to be remarkably effective in creating image classification models and solving computer vision problems. CNN works by the principle of linear algebra extracting unique features in data to discover patterns that distinguishes distinct groups. CNN is therefore a viable key to this research work by helping to identify the visual characteristics of corrosion on the surface of alloy materials regardless of the size. The main aim of this paper is to create a multiphase corrosion detection framework that analyses the degradation state of a material or component and provide detailed and instant information needed during inspection and maintenance. The objectives are to conceptualize the corrosion damage stages, map a data set of corrosion images to the

conceptualized stages and use the images to train a 4-way model that detects damage status, stage, severity, and region. Lastly, the framework is validated for precision and accuracy.

2. Related works

2.1. Mechanical methods

Over the years, different methods have been used to prevent and control corrosion. Mechanical approaches have been used in previous research. Sacrificial anodes are more successful in cathodic prevention (i.e., preventing the onset of corrosion) than at controlling ongoing pitting corrosion, according to some chemical tests (i.e., in guaranteeing cathodic protection). To stop the development of pitting corrosion in the exposed portion of marine piles, submerged sacrificial anodes were used [5]. Previous studies have also employed fibre optic sensors in fleet management to identify corrosion [20]. Corrosion prevention methods also include the application of protective coatings. Entire length tube coatings have also been used to avoid corrosion of water boxes, tube sheets, and tube ends while being priced [37]. Titration practise, a wet-chemistry technique, aids in making sure the lubricant is doing its job of limiting the impacts of dangerous impurities and avoiding wear and corrosion [34]. Studies have also shown that One of the finest methods for shielding metals and alloys from harm is the use of corrosion inhibitors [1]. Besides being manual, these mechanical methods are mostly useful for prevention and not effective when corrosion is already present.

2.2. Deep generative methods

Deep generative methods have also been applied to solve corrosion problems. Two independent deep generative models, variation autoencoder (VAE) and generative adversarial network (GAN), were the subject of a recent study that looked at how they may generate artificial data to augment small experimental datasets. A small amount of stacked graphene over copper surfaces was part of the model experimental system. When compared to VAE-generated synthetic data (78–80 % accuracy), GAN-generated synthetic data demonstrated superior neural network system performance (83–85 % accuracy) [3]. In a different investigation, corrosion morphology was measured from optical micrographs of exposed metals using fractal dimension analysis and corrosion boundary length-to-area ratios. A convolutional neural network (CNN)-based pattern recognition algorithm was trained using the images, and it was then used to objectively determine which accelerated test bear more resemblance to the field exposures [42]. These methods were majorly focused on improving the datasets.

2.3. Transfer learning

Transfer learning has also been effective in handling corrosion. In contrast to ordinary VGG and AlexNet models, an upgraded VGG16 convolutional neural network classification has been applied [43] to effectively categorise and detect the various corrosion patterns on the inner walls of water supply pipes with varied morphologies and complicated and variable backdrops. Based on acoustic characteristics and artificial neural networks, researchers created a remote, non-contact, online health monitoring and warning system for wind turbine blades. Since each class only had 50 data, the testing results showed that the method's accuracy was 94.1 %, compared to classic ANN's accuracy of just 85 % [44]. Another study integrated a phase-field model with machine learning and uncertainty quantification to create a probabilistic computational analytical framework for corrosion-to-crack transitions [24]. Time-series analysis-based methodology based on data has also been investigated. To forecast the free corrosion potential (Ecorr) based on 150 days of immersion testing in saline solution, a corrosion propagation model was created utilising cobalt-based alloys

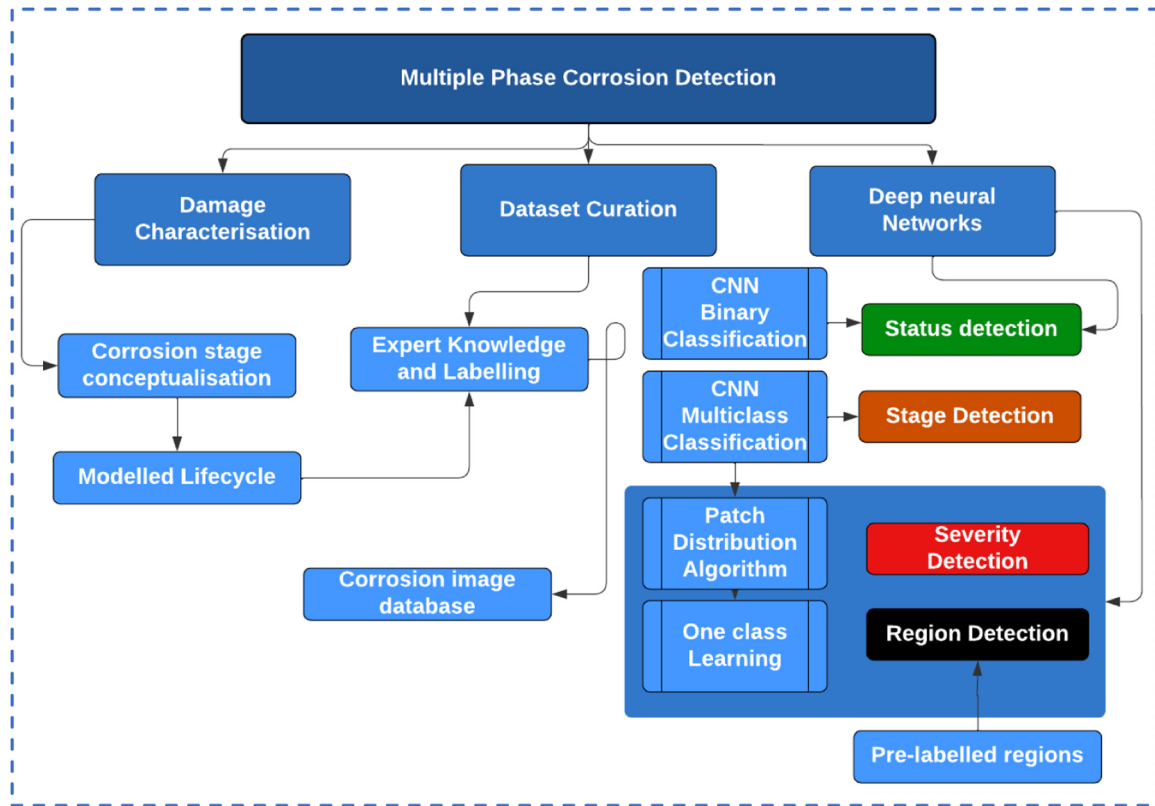


Fig. 1. Main methodology framework.

and duplex stainless steels as the case scenario [16]. The proposed method in our work extends beyond a single algorithm, it would encompass various methods at various stages for a robust corrosion information.

2.4. CNN related methods

The review of numerous works established that the use of convolutional neural network related methods. Recently, A probabilistic computational framework was developed for predicting corrosion-induced cracking in large structures [24] by integrating a phase-field model with machine learning and uncertainty quantification. The author made use of CNN and a Gaussian Process regression model with a nonlinear autoregressive exogenous model (NARX) architecture to facilitate the model. Abubakr et al [2] worked on image classification using Xception & Vanilla models based on convolutional neural networks (CNN). The result of the classification was compared using concrete defect bridge image (CODEBRIM) dataset, Xception gave an impressive accuracy of 94.95 % thereby outperforming Vanilla which gave an accuracy of 85.71 %. However, the former only focused more on solving crack defect caused by corrosion while the latter was focused on general defects in the context of Reinforced concrete (RC) bridges defects. There is the need to proffer solutions to corrosion inspection regardless of the context.

[38] postulated that the current image-based corrosion detection techniques are very time consuming and require bogus amount of workforce and finances. They therefore proposed an artificial intelligence method that makes use of CNN for hull structural plate corrosion damage detection and recognition based. A classifying model was generated from considerable number of images. Furthermore, an overlap-scanning window algorithm was used to identify location of damage. This research showed the enormous potential of artificial intelligence technology for corrosion detection; however, the findings tend to tilt majorly towards the field of naval architecture & ocean

engineering.

2.5. Domain-specific deep learning methods

A lot of previous deep learning works target specific domain such as civil engineering, especially in design, construction, and maintenance of roofing systems [18], corroded power line insulators due to air-borne contamination and weathering [21], structural identification using CorrDetector [11], non-destructive inspection in aerospace industries [45] in which findings show a 100 % detection only in moderate corruptions, weathering steel structure facilities [36], underwater exploration [26], intelligent computer vision is highly important because the metallic parts corrode when they encounter sea water which contains sodium chloride, and it is often not easy to inspect these parts as it would be done outside water. A deep learning based under water metal object detection that makes use of image data was proposed for an underwater autonomous operation [26]. The author also made use of pre-processing techniques for illumination and image enhancement as well as KFCM-Segmentation and DWT extraction. A classification accuracy of 98.83 % was recorded for this approach. In another research, an alternating current field measurement (ACFM) technique presented as an image pre-processing technique before using CNN to identify the defects and the classification was accurate [40].

In some other context-specific corrosion research using CNN, pipelines were found to be exposed to various unfavourable environmental conditions that results into defects such as corrosion amongst others, a stereo camera-based monitoring system was therefore proposed for scanning and detecting these defects especially in cracks and leakages [29]. The solution made use of a stereo camera for extracting information and then using CNN to identify the defect classification. However, the focus was also majorly on generation of 3D point clouds from the result. This was more of a generative oriented solution. Also, in high voltage transmission line, it is essential to ensure that the anti-vibration hammer is in good condition to guarantee safety. A deep learning



Fig. 2. Methodology flowchart.

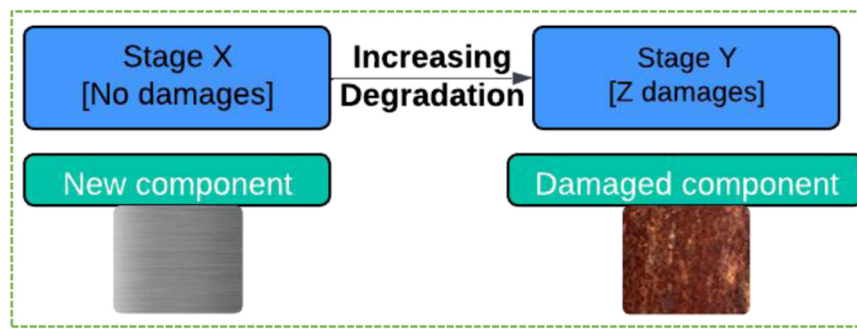


Fig. 3. Conceptualization of the polar end of the stages in the life of a component.

method, Faster Region-based CNN including Feature Pyramid Network (FPN) and a two-stage cascade Region Proposal Network (RPN) was proposed for identifying defects in the anti-vibration hammer [41]. This method focused majorly on identifying 4 kinds of defect in anti-vibration hammer.

2.6. Automated visual inspection

Lastly, some research made very excellent results in automatic visual inspection. In Thermo-electric pipes, automatic visual inspection of corroded metallic pipes that makes use of CNN is seen a better alternative to the handcrafted method [35]. The model made use of compact network which makes parameter manipulation extremely easy. An accuracy of 99.20 % was recorded in detecting various levels of corrosion in the pipe walls. The research made impact in the automatic inspection but only focused on thermos-electric pipes. CNN has been used in previous work to detect difference between surface protective coatings in metals [28]. Hybrid CNN-LSTM model has also been used in the detection of defects on welded joints in metallic pipelines [31]. Another research focused on the use of computer vision to detect corrosion in water, oil and gas pipelines with a custom CNN designed such that very few parameters are learned unlike most CNN [4]. The result shows an exceedingly high classification accuracy of 98.8 % which is quite impressive than most state-of-the-art classifier. However, none of these works investigated the identification of specific regions where the corrosion is detected.

3. Methodology

The multiphase CNN created in this work consists of four different deep learning methods that works separately but towards a main goal of providing a robust information about the corrosion features on an alloy material surface. The four different methods help to give corrosion information such as damage status, stage, severity, and region.

The overall methodology as seen in Fig. 1 begins with damage characterisation and this has to do with conceptualisation of the various stages in the lifecycle of a typical alloy material. The stages in the process are established and conceptualised with an equation which is discussed extensively in Section 3.1. After establishing a lifecycle for alloy materials in the context of corrosion, Dataset is curated to create an image database for the multiphase corrosion detection solution. The data consists of 600 anonymised images of alloy with varying levels of corrosion defects. The images were mapped to fit into the previously modelled lifecycle after which preprocessing, and augmentations are conducted. A simple CNN model is used to implement the corrosion status detector with a binary classification algorithm discussed extensively in section 3.3.2. The stage detector is implemented also with a CNN multiclass algorithm supported with a pretrained model known as VGG-16 and is explored extensively in section 3.3.3.

A one class training approach is used to create the severity detector. This method is achieved with Patch distribution algorithm to find any

level of anomaly by assigning anomaly scores to corrosion defects. In addition, learning from pretrained networks WideResnet and Resnet-18 were transferred to support the method. This is explored in section 3.3.4. Finally in section 3.3.5, the region detector is implemented by slightly modifying the dataset by categorising into a new set of categories of 9 damage locations. The patch distribution algorithm was applied for effective localisation and isolation of the defects.

The flowchart for the proposed methodology is fully represented in Fig. 2. It shows how the proposed method translates to execution in real life. The flowchart depicts a typical corrosion inspection activity which is usually conducted by experts in the field. Here, the proposed system automates the activity with the combination of various algorithms revolving around deep learning. Firstly, the image of an alloy is visualised and immediately there is a check for the presence or absence of corrosion.

According to the flow chart, if there are not traces of corrosion found, then it is imperative that resources are not used further in the inspection process and the inspection terminates after all parts have been examined. In a situation where corrosion check comes back as positive, then the system proceeds to carry out three other checks which involves determination of the corrosion stage, severity and the region and a detailed result is outputted. The process is repeated until all parts have been checked and the system terminates.

3.1. Damage characterization

Firstly, it is particularly important to characterize the expected category that images would belong to from a supervised point of view. Here, a conceptual approach is used to model the lifecycle of a mechanical component in the context of deterioration, defect, damage, or degradation. This is basically because the process of creating states within the lifecycle requires a minimalistic level of specification, the literature review suggests that the idea of corrosion damage is a similar concept across various context. Certain factors will determine why an image belongs to a certain state, why a state exists or why a state precedes another, but corrosion is the same across board. This applies to all damage and defect types, and it provides a filter for each stage and to decide where each image belongs to.

Fig. 3 depicts how the beginning and end of the lifecycle of a typical component can be conceptualized. The component basically starts from a stage X where it is still very brand new and can be described as not having any defect or anomaly. On the other end of the conceptualization is the final stage where the component has completely failed or can be described to have the highest amount of defect or anomaly during its lifecycle. Since this research work is focused on making use of automated visual recognition to solve the problem of defect and anomaly detection, it means stage X and Y for any given component has a visual identity. The various stages in between X and Y will also have an identity.

To streamline the scope of the research work, it is essential to not only view the defect progression from Stage X to stage Y from the

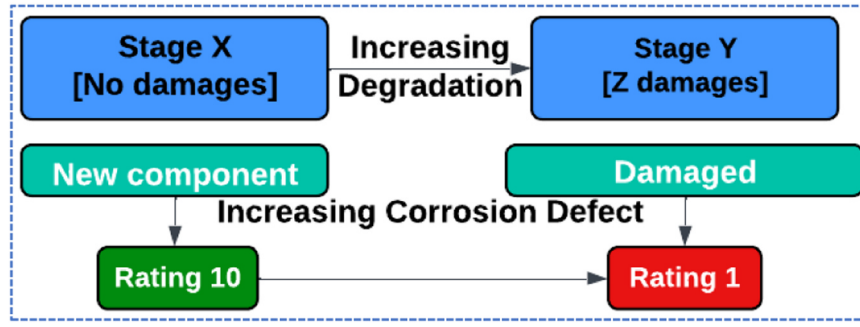


Fig. 4. Mapping collected data to modelled lifecycle.

Table 1
Final categories and sample of collected images.

Stage in lifecycle	Rating	Sample
Stage a	10	
Stage b	9	
Stage c	8	
Stage d	7	
Stage e	6	
Stage f	5	

perspective of the cause. A component may become faulty based on different reasons such as corrosion, cracks, fracture etc. However, since we are not focused on finding the cause of the defect, it is more productive to focus on the result of the action, in this case, corrosion to be specific. Regardless of the cause of the defect, the term used to describe the resulting effect of a defected component or material is Degradation. The stages in the lifecycle can therefore be quantified using a degradation coefficient as seen in Eq. 1.

$$Corr(t) = \sum_{i=X}^Z Deg_x . Stage_i \quad (1)$$

Eq. 1 depicts the conceptualisation of the cumulative amount of corrosion, $Corr(t)$ that is present in each component or material at a given point in time, t when it is being inspected. Deg_i represents the coefficient of degradation that the material has gone through while $stage_i$

refers to the corrosion stage. Summation of all the damages from labelled Stage X to Z and their respective degradation coefficient exemplifies the damage progress and thereby depicts the status of the material or component at every given point in time when it is being inspected.

$$Corr(t) = Deg_x . Stage_x + Deg_{N1} . Stage_{N1} + \dots + Deg_N . Stage_N + Deg_{N+1} . Stage_{N+1} + \dots + Deg_Z . Stage_Z \quad (2)$$

A more precise version of the conceptualised corrosion equation is given in Eq. 2 to show specific stages and damages. Here, Deg_x depicts the coefficient of degradation at the first stage $Stage_x$. Deg_N and $Stage_N$ depicts the coefficient of degradation and corrosion stage at intermediate stages respectively with $N1, N2, \dots, N$ representing the stage labels. In conclusion, the final degradation coefficient and last stage of corrosion is given by Deg_Z and $Stage_Z$ respectively.

3.2. Image database & expert-knowledge

Secondly, corrosion images are collected, and expert knowledge is used to index these images. These images adhere to the model in the previous stage. This also includes methods that involves pre-processing these images and make them suitable for the next phase. The specific material chosen is the alloy family because they are present in a large amount of mechanical component.

Corrosion image data set was obtained from ARL website [6] for BMVC (British Machine Vision Conference) release 2021. This consists of 600 anonymized image data and using ASTM(American Society for Testing and Materials) standards where material panels were exposed to NaCl, $NaHCO_3$ and $CaCl_2$, these are the constituents that are normally present in seawater which causes corrosion in real world scenarios. The panels were also exposed to varying levels of temperature and humidity through the laboratory experiment cycles.

The images were labelled and annotated correctly by experts using ratings that conforms to the standardized laboratory environments and procedures. The dataset is interesting and chosen due to its uniqueness and realistic depiction, for example the images have different background colours, contains noisy areas as well as single or double scribes to give the impression of actual corrosion, also, some of the corruptions are very minute and cannot be seen with ordinary eye. The rating starts from 0 and ends with 10. 10 is the rating for images with minimal amount of defect of 0 mm corrosion while 1 is the rating for images with the highest amount of defect of 16 mm+ corrosion. Hence, the modelled lifecycle can be modelled to the images collected where rating-10 is mapped to Stage X while rating-1 is mapped to Stage Y as seen in Fig. 4. All other rating values falls 10 and 1 falls between stages X and Y in the order of increasing stages within the modelled lifecycle [6].

The final label of the collected data falls within 6 categories as seen in Table 1.

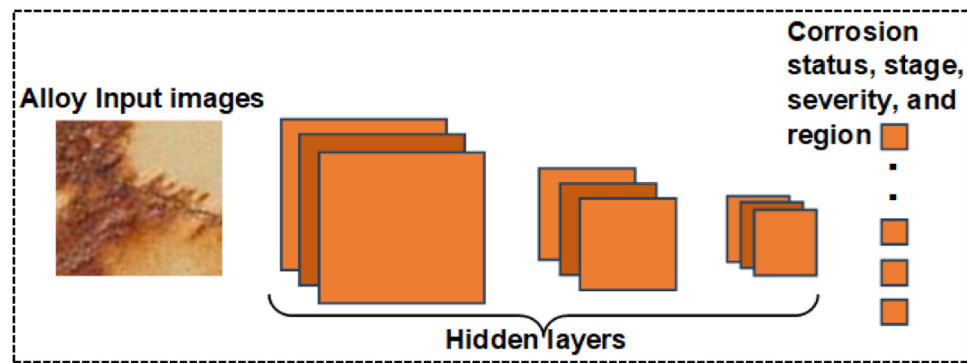


Fig. 5. Overview of the convolutional neural network for identifying degradation stages in the lifecycle of components.

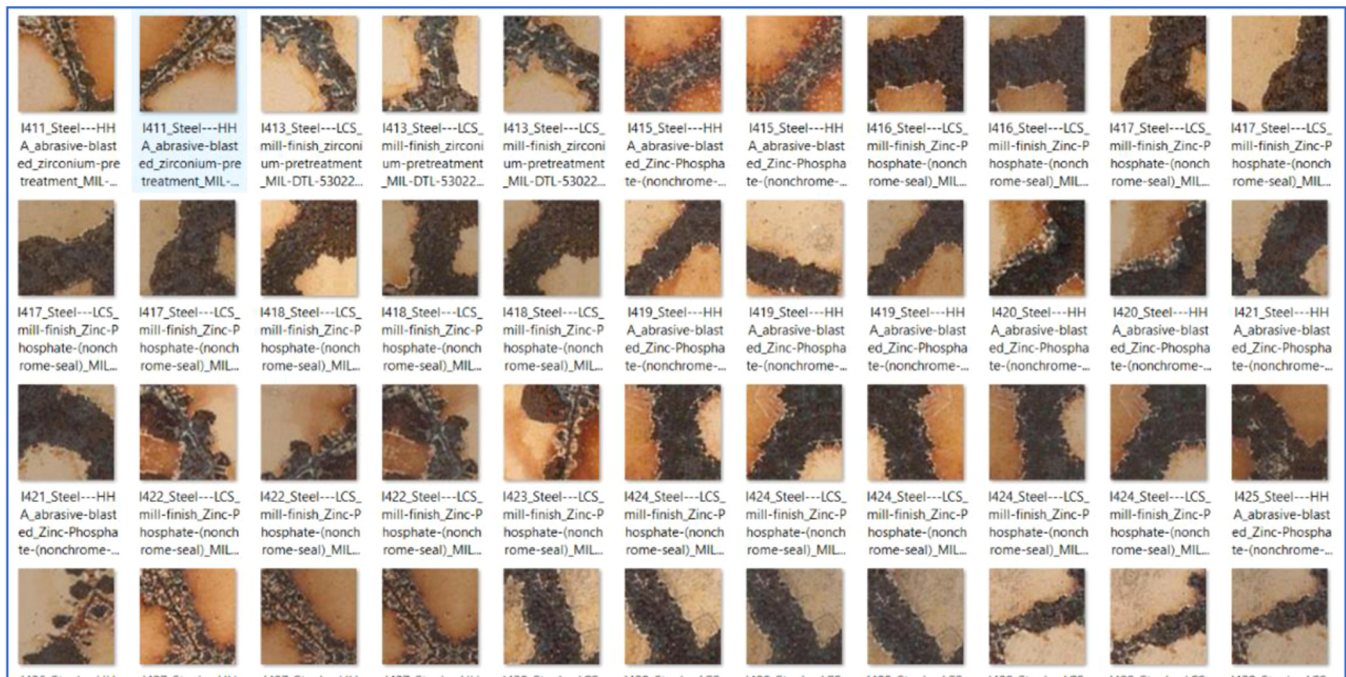


Fig. 6. Noise reduction: damage zoomed images.

3.3. Convolutional neural network (CNN)

The CNN is used to create a model and trained with the images to understand the lifecycle structure and architecture. This is done with the help of Keras, TensorFlow, NumPy, Scikit and other useful library within the python programming language domain.

The basic working principle of the convolutional neural network makes it especially useful in a lot of image recognition problems due to its ability to process data with grid-like topology. This is remarkably similar to the way the human brain decodes images that are being viewed with the human eye. The layers of the CNN are arranged such that they can detect various parts of images in each of the stages in the lifecycle or classes defined for the collected images (Fig. 5).

Invariably, there is an input layer where the complete image is fed into the neural network. There are series of hidden layers between the original image and the final layer where the complete recognition occurs to classify the image into the appropriate category to provide information about the status, stage, severity, and region of the inspected alloy material.

3.3.1. Data preparation and augmentation

To create an effective model, it is especially important to prepare the

data needed. This involves:

- **Data augmentation:** Data generation is needed to support the low availability of images representing the corrosion stage. The existing images are used as an input to generate variations using methods such as Rotation, Shifting, Shearing, Zooming and horizontal flipping. The augmentation was implemented using Keras pre-processing class for generating data sets to support the already available datasets. **Data cleaning:** Corrupted Images, Images with wrong format (not JPEG), images with wrong size (not 512×512) and images that do not conform with the standard of the data requirement are eliminated from the data sets.
- **Noise reduction:** an essential approach to ensuring that the model is focused on necessary aspects of the image and that key features are extracted is to reduce noise. In this case, the background is reduced to focus on the damage itself as seen below while in other cases the image is converted into grayscale.

(Figs. 6 and 7) One approach towards applying convolutional neural network is to train the network from scratch using the images. Different training methods are taken for this training based on the data manipulation; they are:

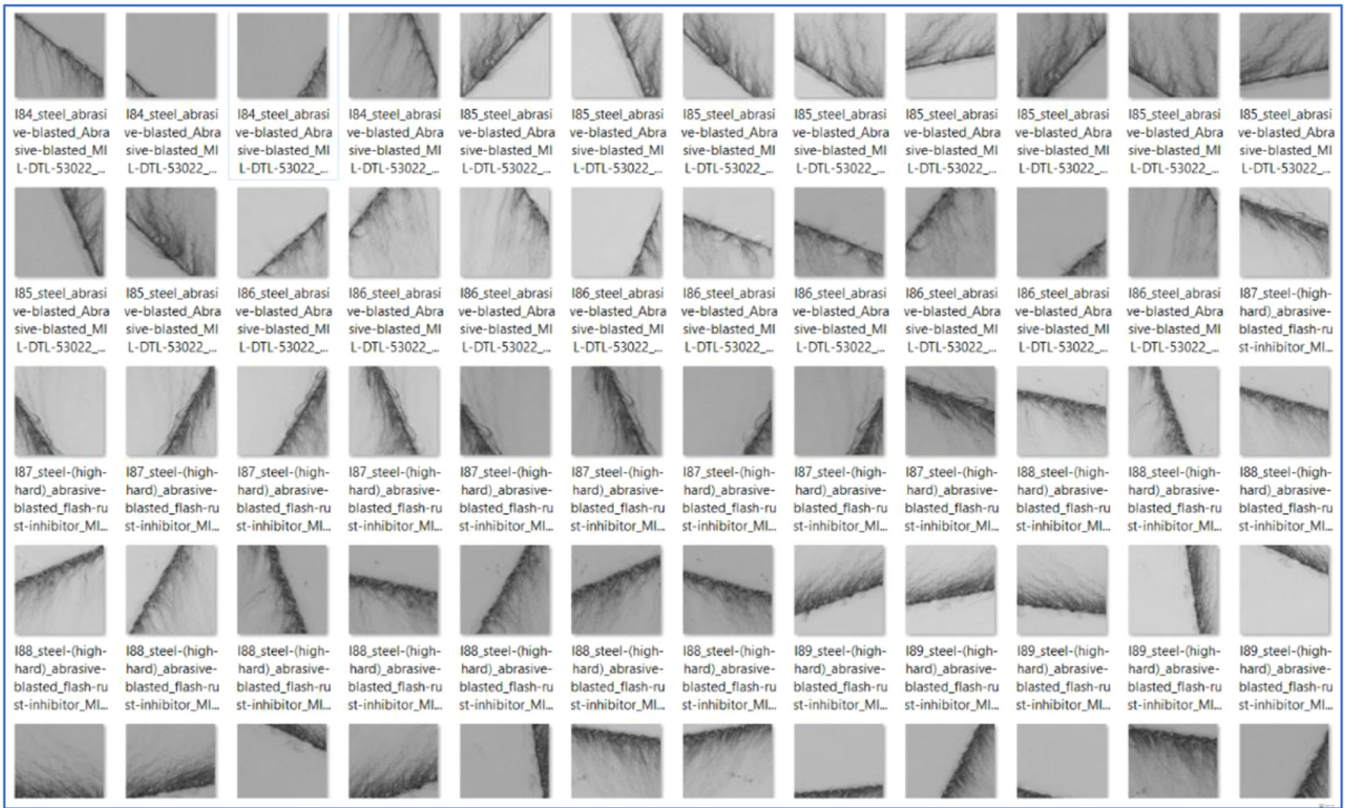


Fig. 7. Noise reduction: colour removed (grayscale).

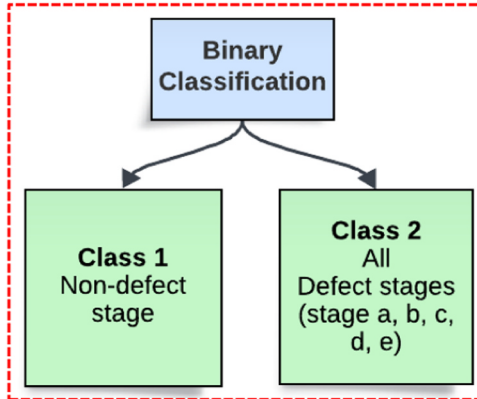


Fig. 8. Binary training using a single class for all defect stages.

- Binary training of all defection stages as a single class against non-defected materials
- Multi-class training of all defection stages and non-defection as separate classes

The major difference between the two methods described above is the difference in the data. Whilst binary training consists of data grouped into two categorical labels, the multi class method involves data grouped and identified by multiple labels. This is also a logic-oriented method to put less load on the framework in case the status shows that there is no degradation detected, then the flow of the framework can exit and not proceed to check for damage stage, severity and region.

3.3.2. CNN level 1 (one) – damage status detector

Here, all the various defection stages are collapsed into one category.

Since all the stages apart from the non-defection have a form of defect no matter how small, it is important to first train the model to understand the clear boundary between a brand-new material and a defected material within the context of the lifecycle of that material (Fig. 8).

The data structure for this training involves the following (Fig. 9):

The total number of data used for this training is 600 for each of the defect class and non-defect class. The data was further split into 2 and a group of 300 was applied for training while the other 300 was applied to testing or validation. Since majority of the non-defect class were augmented, the order of distribution into train/test group doesn't matter. However, the defect group was evenly split into the train/test group so that each of the five defect stages are equally represented. The model would be ineffective if specific stages are used for training while a different stage is used for validating since the aim is to have a robust model that can identify different stages for this training set (Figs. 10 and 11).

Convolutional neural network is quite appropriate for this classification problem. There are unique 600 images available for the defect stages, hence there should not be so much problem with overfitting. The major area where augmentation is applied at this point is in the non-defect stage, this is not much of a problem because most brand-new materials would look the same while still in their brand-new form. Also, the images are left at the coloured form and not converted to grayscale to ensure that as many features are retained as possible for robustness of the model, hence potentially modulating the entropic capacity of the model along with other method such as weight regularization.

The network is built from scratch and is intentionally designed to have quite a small number of layers as well as filters in each of these layers. Data augmentation and dropout is also applied because it is important to ensure that trained patterns are unique. In summary, the network consists of:

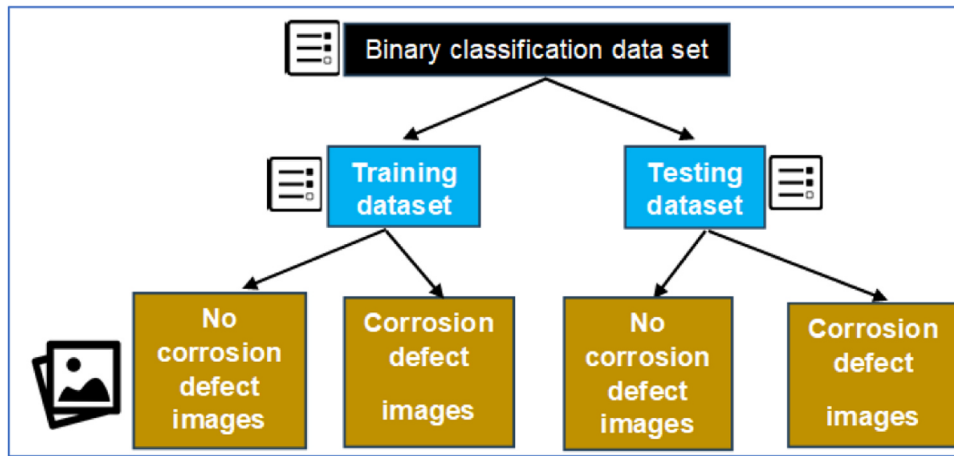


Fig. 9. Data structure for binary training using a single class for all defect stages.



Fig. 10. Preview of dataset in defect category.

- Three convolution layers stacked on each other.
- RELU activation
- Max pooling layers

The neural network is built on top of Keras open-source library. This is implemented in a python environment 3.7 to be specific. Various classes within the Keras library such as `Imagedatagenerator`, `Sequential`, `Conv2D`, `MaxPooling2D`, `Activation`, `Dropout`, `Flatten`, `Dense` and `Backend` are firstly imported to make use of them within the network.

The image dimensions are specified which is 512×512 , all images within the train and test folder have been processed and all resized to this dimension. Also, this dimension is used as an argument for the image generator class as well as setting the image data format. The path for the location of the training data as well as that of the validation data is specified in the script. The path is needed during training to load the images. The number of training and validation samples are also specified. The last set of necessary parameters to be specified are the epoch and batch size. These parameters are needed during the fitting stage of the model and needed to be defined earlier.

The Image data format is originally specified in the `keras.json` file of

the host machine where the training is being conducted. However, to ensure that the dimension of image data being used across the multi-layer of the CNN is consistent, this parameter is reset in the script to give TensorFlow an idea of what to expect in each layer, hence preventing the model from being trained in unexpected ways. The layer within the network consists of three convolution layers each of which consists of an activation layer with a max-pooling layer.

Due to the nature of the image classification problem which is binary i.e., classification of two classes of defect and non-defect stages of materials. There is the need for a sigmoid activation; this is attached to a single unit at the end of the model. The model also consists of two fully connected layers.

To prevent overfitting, data augmentation is implemented within the network. By doing this, we ensure a proper modulation of the entropic value such that the model stores as much essential feature as possible. Data augmentation is used by applying image data generator from the Keras library to both training data and test data. The image dimension that was defined earlier in the script is used here to ensure that generated images conform to the dimension of the original images.

The training of the model is conducted using the `model_fit` class from

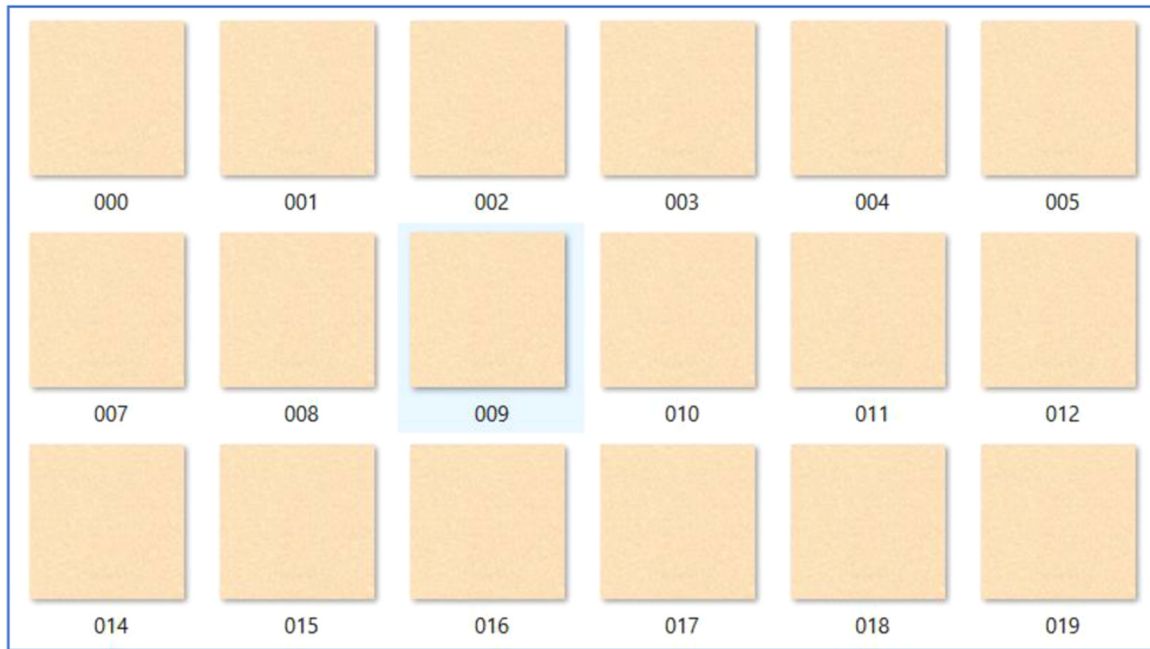


Fig. 11. Preview of dataset in no-defect category.

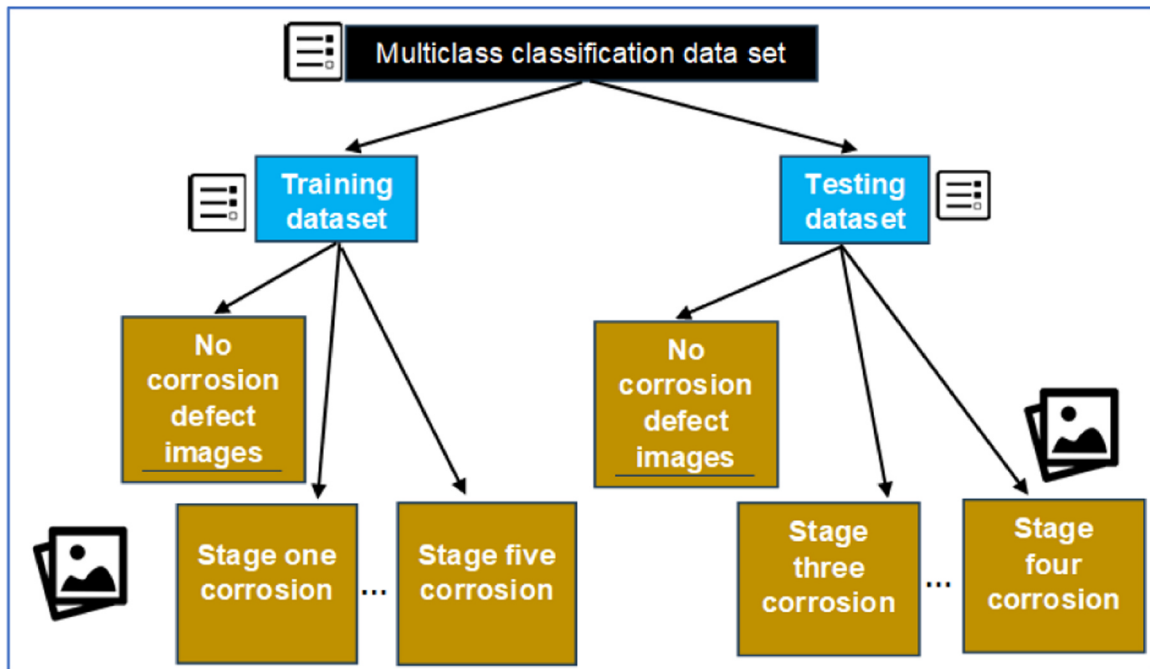


Fig. 12. Dataset arrangement for multistage defect classification.

the Keras Library with parameters such as the training and test generators from the previous stage. The number of epochs defined is 30 using the complete number of training (300) in each of the epochs. Since the number of samples is not extremely large, the training is easily done using CPU power.

Finally, the model is saved after training into a file. The saved format is .h5 which is a h5 file format. This allows recreation of the model and a lightweight version of the model that can be used for immediate prediction in testing or production.

3.3.3. CNN level two – damage stage detector

This stage has to do with creating a model that can correctly identify the stage which a material component belongs within the degradation lifecycle. The data collected already represents the marked stage a-f which describes the increasing intensity of degradation on the material component. There are a total of 6 classes representing the stages.

The training and testing data for the multi class degradation stage is structured as seen in Fig. 12.

There is a total of 300 images in each category or class. Original images were re-generated and augmented by rotating, flipping, zooming etc. to generate unique images sufficient for the training and validation

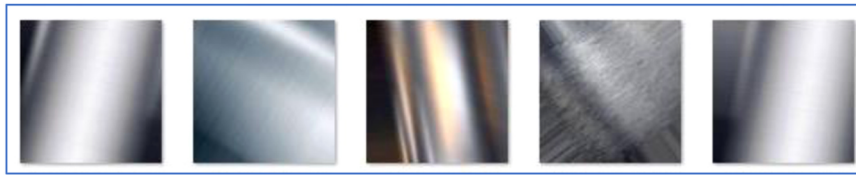


Fig. 13. Preview of no defect stage dataset.



Fig. 14. Preview of stage one defect dataset.



Fig. 15. Preview of stage two defect dataset.



Fig. 16. Preview of stage three defect dataset.

set. Peek view of image data in each of the categories goes thus (Figs. 13–18):

To have a more effective model, it is sometimes preferable to make use of the knowledge of a model that has been trained before using other data or pretrained models as it is fondly called in the machine/deep learning domain. The pretrained model applied here is the VGG16 model. The VGG16 model is an object detection and classification algorithm which has already been trained to categorize a thousand images

of a thousand categories, it was trained on one of the large datasets in the world, ImageNet dataset. VGG16 already has already learned features used to classify different objects and even though the defect data is not present in the ImageNet data used to train VGG16, it is possible to apply it in such a way that it generalizes in this problem (Fig. 19).

VGG16 is a neural network that consists of 6 layers, 5 convolution blocks and a single fully connected layer. Implementing the CNN for this degradation lifecycle identification will involve the convolution aspect



Fig. 17. Preview of stage 4 defect dataset.



Fig. 18. Preview of stage 5 defect dataset.

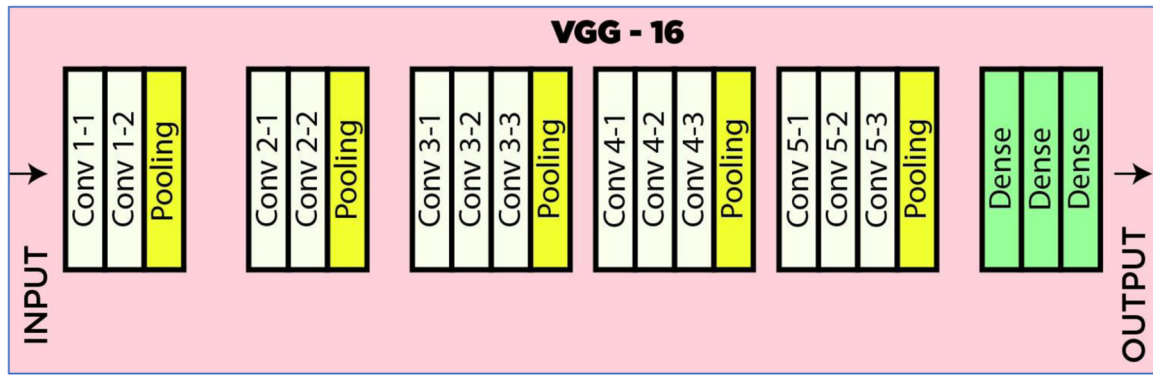


Fig. 19. Architectural view of the VGG-16 models.

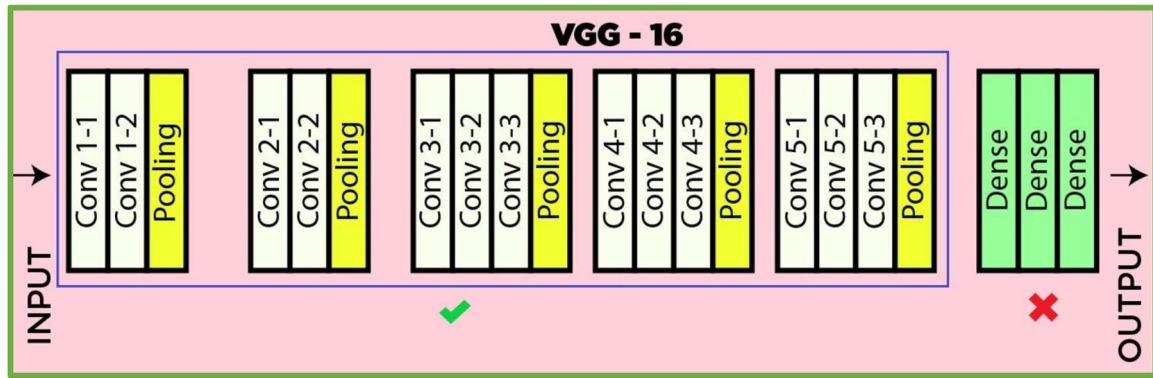


Fig. 20. Adopting the convolutional layers of the VGG-16 model.

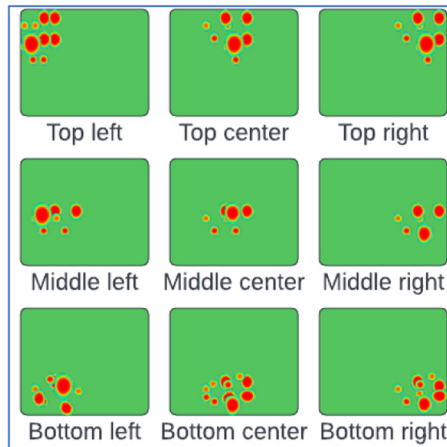


Fig. 21. Representation of Region of damage showing area of concentration.

of the model alone (Fig. 20).

The 6 classes of the defect stages dataset are divided into 2 groups of training and test split, each group containing 300 images, totalling 3600 images (1800 training and 1800 testing) is used to train the convolution layer of the VGG-16 model. This initial output is saved for computational efficiency, converted to NumPy arrays. Then, a fully connected layer is trained on top of the stored features. In the beginning of this model, the necessary Keras classes are imported for use. The NumPy library is also imported. The dimension of the images used in the training and testing stage are all of size 512×512 , this was ensured during the data cleaning stage earlier. It is therefore essential that this dimension is predefined within the global scope so that so can be referenced easily at different

points of the model script.

The path for saving the initial model is specified. The path for training and testing data is also specified as well as other parameters such as number of training samples, number of validation samples, epoch, and batch size. It is possible to use this information directly inside the model scripts as well but for clarity, consistency, and reusability, it is often better to define them globally.

As mentioned earlier, the first step is to make use of the convolution part of the VGG-16 model. A function *save_initial_features()* is created to conduct this process. Since this is a multiclass problem, the class mode is set to categorical as opposed to binary which was used in the previous classification involving two classes. The model is trained and validated using the whole dataset.

The other part of this approach is to create a brief layer of network that consists of a fully connected layer. A function is created to perform this process. By doing this, the stored features from the previous stage will be conveniently transferred. The activation function used is softmax while categorical_crossentropy is used for the loss function. Softmax activation is an ideal parameter with proven efficiency with multiclass image processing problems. Finally, the two separate functions are called in the right order.

3.3.4. Anomaly finder/severity detector

Another approach to using computer vision as a diagnostic tool to analyse degradation in material component is the concept of anomaly detection. It is quite convenient to classify defects based on a stage, however, there is no absolute certainty that every single stage of degradation will be modelled. There are certain patterns or stages of degradation that would not be capture in the modelling stage as well as images that would never be trained or assessed by the neural network in the previous methods used in this research work. It is therefore now that the concept of localization is important.



Fig. 22. Actual sample of various damage regions annotated.

Table 2

Corrosion stages and class mapping.

Stage	Defect category
Stage a	No defect
Stage b	Defect class 1
Stage c	Defect class 2
Stage d	Defect class 3
Stage e	Defect class 4
Stage f	Defect class 5

Anomaly detection simply involves the comparison of the image of a defected component material with the image of a normal material component, then analysing and assigning a score usually known as anomaly score to that defected material component based on the degree at which it differs from the normal material component. Anomaly localization involves assigning each pixel or each patch of pixel an anomaly score to output an anomaly map. A proven algorithm that is appropriate for this kind of problem is called the Patch Distribution Algorithm, usually known as the PaDim. This was first introduced by [8].

Patch distribution algorithm is uniquely known for its single class learning approach. It makes use of a pre-trained convolutional neural network for patch embedding and Multivariate Gaussian Distribution to compute a probabilistic representation of a normal class. PaDim

Table 3

Model summary of damage status detector.

Layer (type)	Output Shape	Param #
<i>conv2d (Conv2D)</i>	(None, 511, 511, 32)	416
<i>activation (Activation)</i>	(None, 511, 511, 32)	0
<i>max_pooling2d (MaxPooling2D)</i>	(None, 255, 255, 32)	0
<i>conv2d_1 (Conv2D)</i>	(None, 254, 254, 32)	4128
<i>activation_1 (Activation)</i>	(None, 254, 254, 32)	0
<i>max_pooling2d_1 (MaxPooling2D)</i>	(None, 127, 127, 32)	0
<i>conv2d_2 (Conv2D)</i>	(None, 126, 126, 64)	8256
<i>activation_2 (Activation)</i>	(None, 126, 126, 64)	0
<i>max_pooling2d_2 (MaxPooling2D)</i>	(None, 63, 63, 64)	0
<i>flatten (Flatten)</i>	(None, 254,016)	0
<i>dense (Dense)</i>	(None, 64)	16,257,088
<i>activation_3 (Activation)</i>	(None, 64)	0
<i>dropout (Dropout)</i>	(None, 64)	0
<i>dense_1 (Dense)</i>	(None, 1)	65
<i>activation_4 (Activation)</i>	(None, 1)	0
Total params: 16,269,953		
Trainable params: 16,269,953		
Non-trainable params: 0		

leverages on the correlations between the various semantic levels of CNN. It is also known for its Low complexity because it transfers knowledge from pre-trained network and the complexity that comes with training strong deep neural network or large amount of data is eliminated.

Table 4
Model Result comparison for damage status detector.

Data type	No of Epochs	Batch size	Training size	Test size	Accuracy	Average Precision
Colored	5	5	1500	600	88.60 %	0.50
Colored	10	10	1600	650	93.73 %	0.49
Grayscale	5	5	1500	1200	94.87 %	0.49
Grayscale	10	10	1600	650	93.53 %	0.50

Table 5
Classification report for damage status detector.

	Precision	Recall	F1-score	Support
Defect	0.51	0.47	0.49	1500
Non defect	0.51	0.55	0.53	1500
Accuracy			0.51	3000
Macro Avg	0.51	0.51	0.51	3000
Weighted Avg	0.51	0.51	0.51	3000

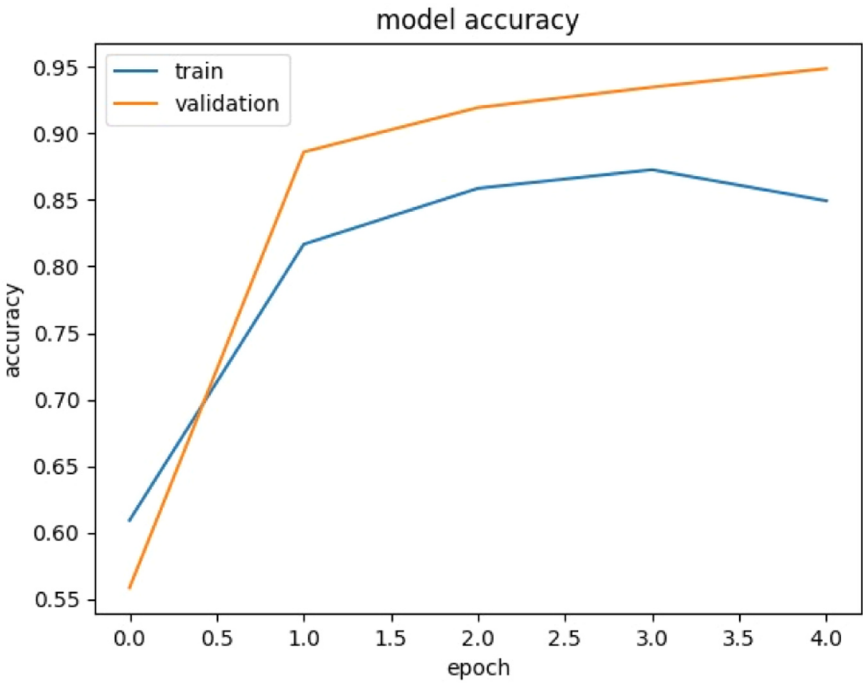


Fig. 23. Model accuracy graph during epoch for damage status detector.

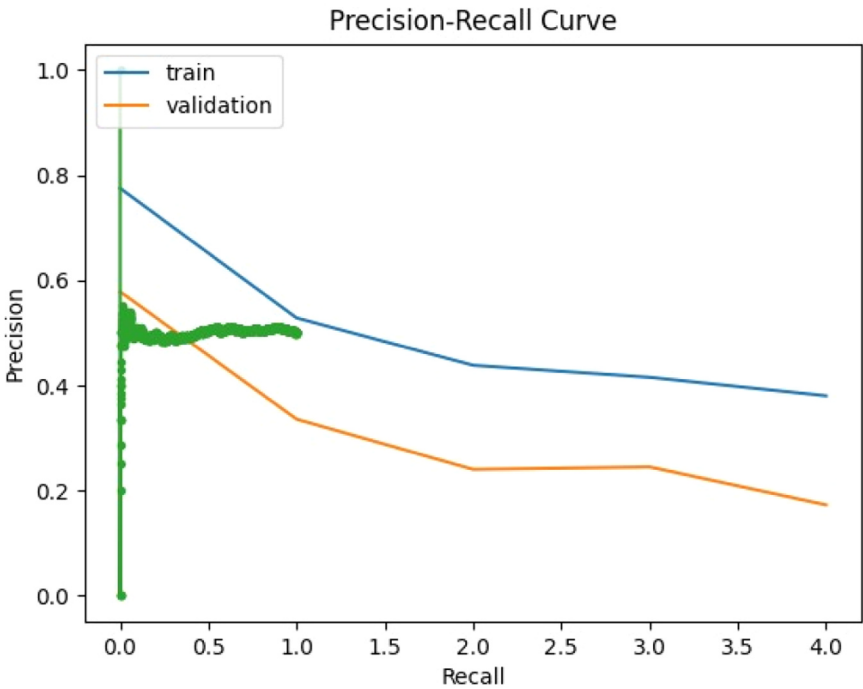


Fig. 24. Precision recall curve for damage status detector.

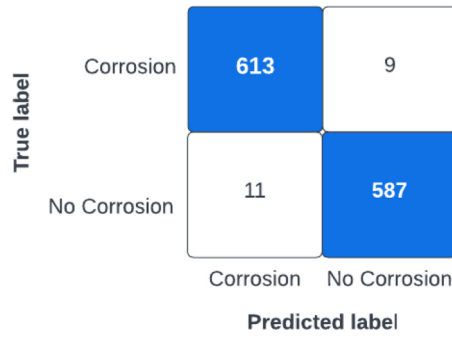


Fig. 25. Confusion matrix of the CNN model for damage status.

Table 6

Multiclass model summary comparison for damage stage detector.

Layer Type	Configuration	Output Shape	Parameters
Input	Input shape: (128, 128, 3)	(128, 128, 3)	0
Convolution 2D	Filters: 48, Kernel Size: (3, 3), Activation: ReLU	(126, 126, 48)	1344
MaxPooling	2D Pool Size: (2, 2), Strides: 2	(63, 63, 48)	0
Convolution 2D	Filters: 48, Kernel Size: (3, 3), Activation: ReLU	(61, 61, 48)	20,784
MaxPooling	2D Pool Size: (2, 2), Strides: 2	(30, 30, 48)	0
Convolution 2D	Filters: 32, Kernel Size: (3, 3), Activation: ReLU	(28, 28, 32)	13,856
MaxPooling	2D Pool Size: (2, 2), Strides: 2	(14, 14, 32)	0
Flatten		6272	0
Dense	Units: 128, Activation: ReLU	128	802,944
Dense	Units: 64, Activation: ReLU	64	8256
Dense	Units: 6, Activation: Softmax	390	390

Firstly, Padim algorithm works by using a multivariate gaussian distribution to analyse all sections of an image and then detecting the connections between the different semantic levels with help of a pre-trained Convolutional neural network such as Resnet, WideResnet or EfficientNet.

The stages of Patch Distribution algorithm include:

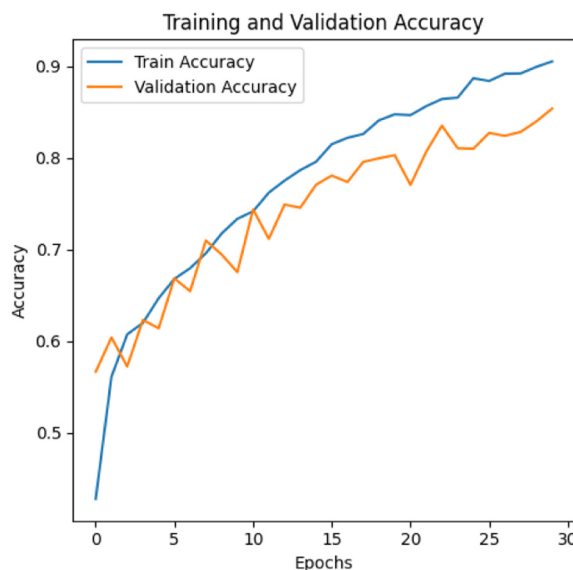
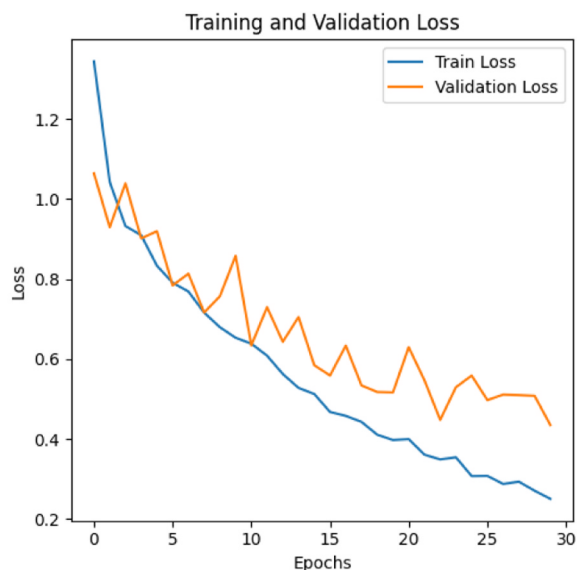


Fig. 26. Graphical representation of epochs showing training and evaluation of multi-class corrosion stages.

- **Extraction:** As earlier mentioned, the rigorous effort or computation power of training a neural network from scratch is eliminated by making use of a pre-trained CNN. Patch embedding vectors are generated from the pre-trained CNN. For every patch of the normal image, a spatially corresponding activation in the pretrained CNN activation map is associated with it.
- **Single class Learning:** The single class is the normal class. It is essential to learn characteristics of normal images since it consists of the features to be used as a standard for detecting anomaly
- **Making deductions:** Mahalanobis Distance is used to compute the anomaly map i.e., the distance between the test patch embedding and learned distribution. High scores in the map indicate high anomaly areas while low score indicates low anomaly areas. The final anomaly score for the entire image is chosen as the maximum score on the map.

There are two main metrics used for evaluating patch distribution algorithm namely:

- **Area Under Receiver Operating Characteristic Curve (AUROC):** True positive rate = Percentage (%) of Pixels correctly classified as anomalous
- **Per region overlap score (PRO-score):** This is a plotting for each connected component, curve for the mean values of the correctly classified pixel rates as a function of the false positive rate between 0 and 0.3. High pro-score indicates that both large and small anomalies are well localized.

The previous data set is used in implementing the patch distribution algorithm. However only the normal class is trained, the corrosion defect class is used for validation. The detailed process includes:

1. **Importing important classes and libraries:** This includes and not limited to NumPy, Sklearn, Scipy, matplotlib as well as the two pretrained Resnet models.
2. **Parsing arguments:** Certain command lines are needed to define important paths. The parser is used to inject such command lines into the code.
3. **Using Pretrained Model:** Here, The Pretrained models are loaded. However, because the experiment is designed to make use of a single pretrained model at a time, a logic is used to separate the models depending on which is loaded and therefore either of them can be used. Resnet and WideResnet were used separately.

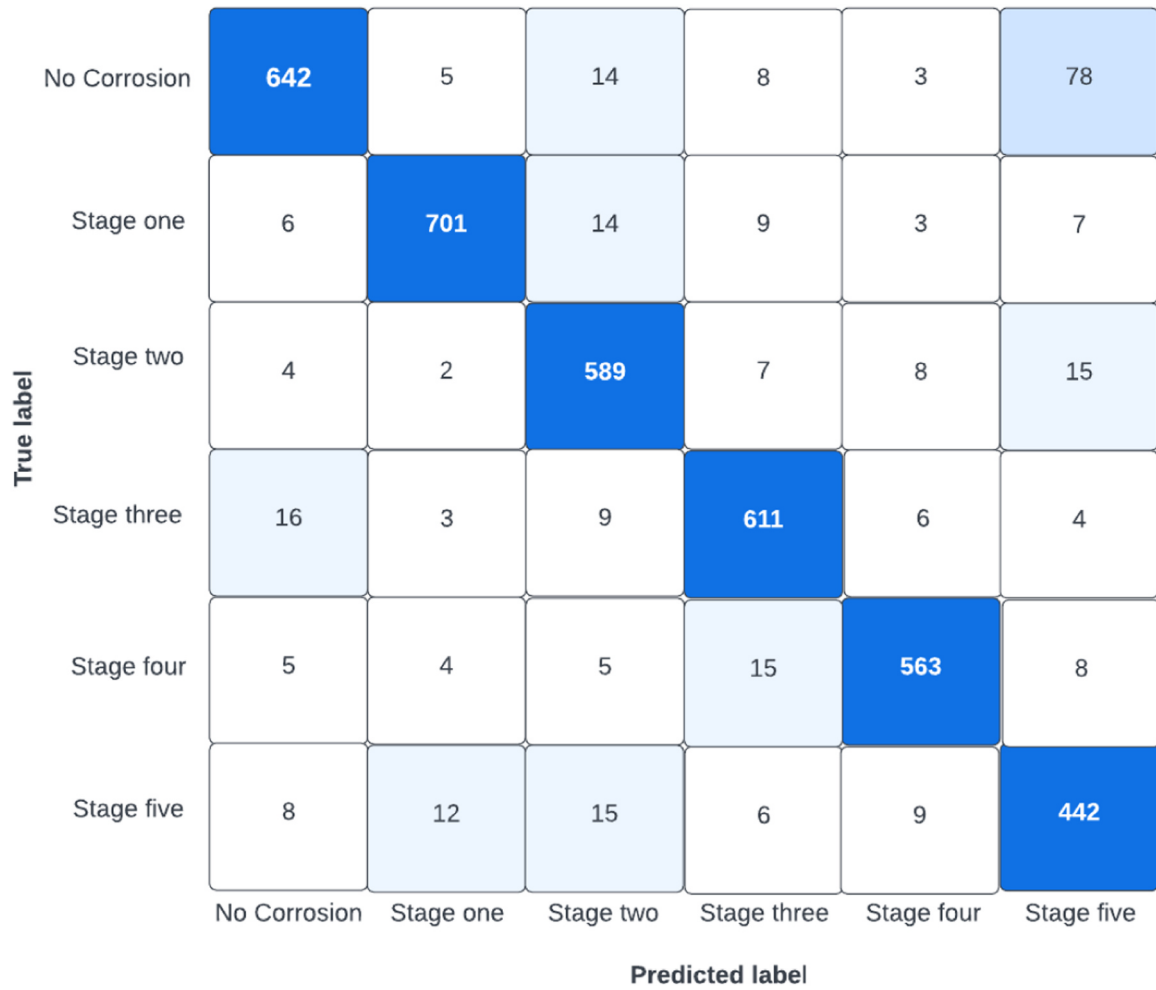


Fig. 27. Confusion matrix of the model for damage stage detector.

Table 7
Multiclass model classification report.

Corrosion stage	Precision	Recall	F1-score	Support
no-defect	0.80	0.85	0.82	1500
stage-one-defect	0.78	0.82	0.80	1500
stage-two-defect	0.82	0.79	0.80	1500
stage-three-defect	0.75	0.77	0.76	1500
stage-four-defect	0.79	0.81	0.80	1500
stage-five-defect	0.77	0.80	0.78	1500
Accuracy	-	-	0.80	9000
Macro Avg	0.79	0.81	0.80	9000
Weighted Avg	0.79	0.81	0.80	9000

Table 8
Pixel ROCAUC using both Resnet-18 and WideResnet-50 as pretrained network.

Corrosion stage	Resnet-18 Image ROCAUC	Wideresnet-50 Image ROCAUC
Stage one	91.7 %	93.2 %
Stage two	96.7 %	97.9 %
Stage three	96.5 %	97.4 %
Stage four	99.2 %	96.8 %
Stage five	98.4 %	96.8 %
Average	96.5 %	96.4 %

3.3.5. Damage region detector

The next phase within the Multi Phase CNN is the region detection model. This is used to detect the specific location of the corrosion

Table 9
Pixel ROCAUC using both Resnet-18 and WideResnet-50 as pretrained network.

Corrosion stage	Resnet-18 Pixel ROCAUC	Wideresnet-50 Pixel ROCAUC
Stage one	81.3 %	92.7 %
Stage two	73.8 %	92.5 %
Stage three	64.8 %	94.6 %
Stage four	53.3 %	95.8 %
Stage five	53.0 %	97.5 %
Average	65.0 %	94.6 %

damage on the surface of the analysed image. To have an effective and accurate result. The damage masks from the patch distribution algorithm are adopted for the specific regions of corrosion. Hence, the images are categorized into the following based on the location of damage concentration (Figs. 21 and 22):

1. Top right
2. Top left
3. Top centre
4. Middle right
5. Middle left
6. Middle centre
7. Bottom right
8. Bottom left
9. Bottom centre

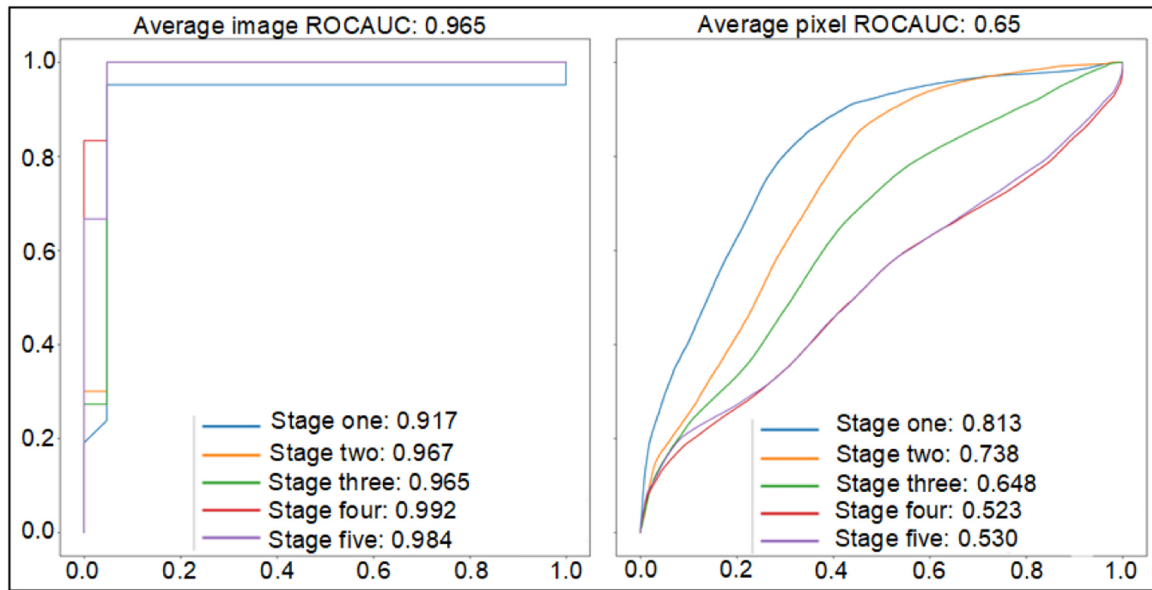


Fig. 28. Graphical representation of image and pixel ROCAUC using pretrained Resnet-18 by corrosion stage.

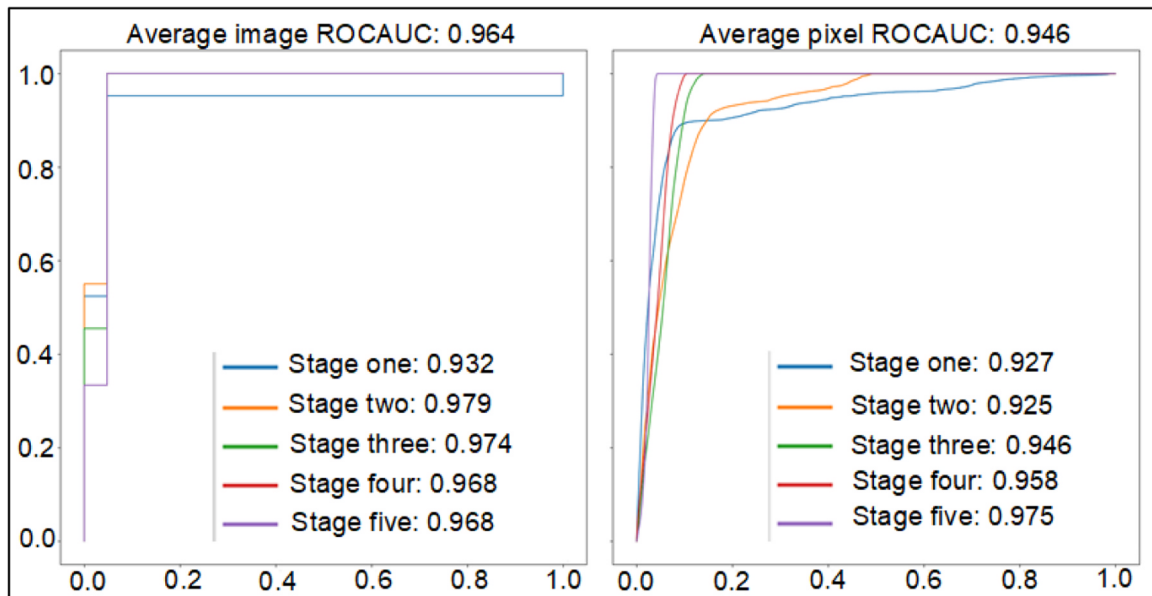


Fig. 29. Graphical representation of pixel ROCAUC using pretrained WideResnet-50 by corrosion stage.

4. Discussion of results

4.1. The damage status detector

A simple CNN binary model was used to implement the damage status detector. The summary of the model is seen in Table 2.

The result for the damage status stage was experimented by varying the image type between colored and grayscale, batch number and epoch. The accuracy and average precision were recorded and compared. The values as recorded in Table 4 showed the highest accuracy of 94.87 % when using an epoch of 5 and batch size of 5 on 1500 grayscale training images while the lowest accuracy of 88.6 % was recorded for colored images at 5 epochs. Also, a fair accuracy of 93.73 % and 93.53 % were recorded for colored and grayscale images respectively when epoch and batch sizes were increased to 10. The classification report of the best accuracy is seen in Table 3 while the epoch and precision graphs are seen

in Figs. 23 and 24, respectively (Table 5).

The training size of 1500 of grayscale images with 1200 test images provided the highest accuracy as seen in Table 3. Compared to other training and test size combinations, this provided the best result. The first justification that could be inferred from this is the use of grayscale images because the number of channels in the images has been reduced. Another justification is because the larger number of test set of 1200 is a true representation of the overall data distribution and a reduction in the variance of data. Finally, the training to assess data ration has helped the model to understand a wider range of pattern in the learned class, hence the possibility of better generalisation and lesser overfitting.

In addition, the saved model weight was evaluated separately after full epochs. This was done by predicting the damage status of sample images that has not been seen by the model before. The Keras library is modified to carry out this function. Firstly, the essential classes are imported from the Keras library. Especially the prediction class

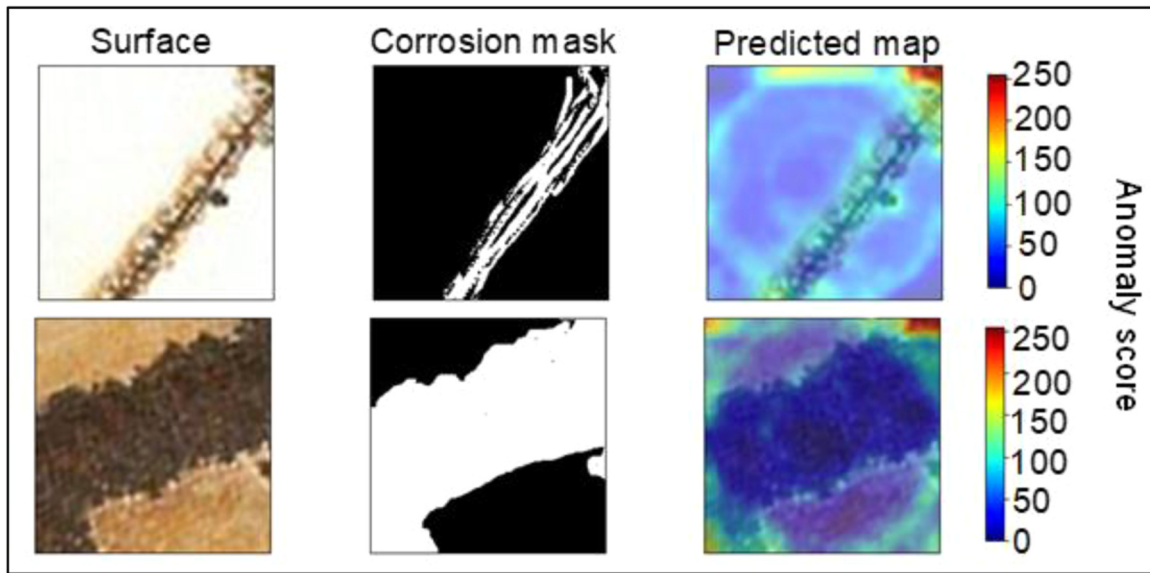


Fig. 30. Anomaly detection based on corrosion stage using pretrained Resnet-18.

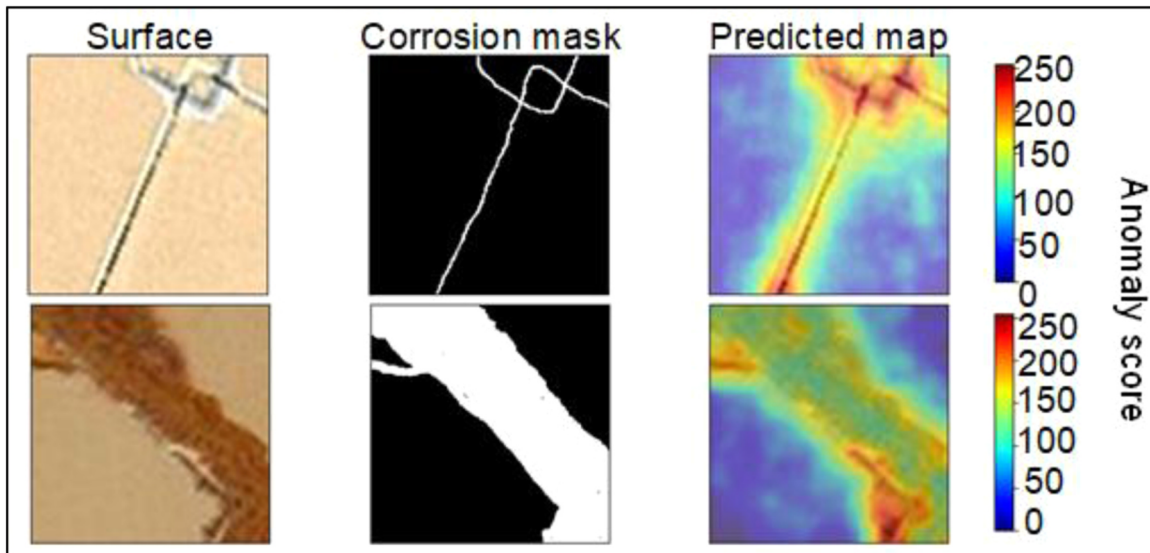


Fig. 31. Anomaly detection based on corrosion stage using pretrained Wideresnet-50.

Table 10

Damage region Image ROCAUC using both Resnet-18 and WideResnet-50 as pretrained network.

Damage region	Resnet-18 Image ROCAUC	Wideresnet-50 Image ROCAUC
Top left	91.8 %	92.3 %
Top right	86.7 %	89.3 %
Top centre	81.7 %	87.6 %
Middle left	86.8 %	95.0 %
Middle right	64.9 %	86.5 %
Middle centre	85.5 %	94.5 %
Bottom left	87.5 %	100.0 %
Bottom right	84.6 %	84.6 %
Bottom centre	87.5 %	93.8 %
Average	84.1 %	91.5 %

Table 11

Damage region Pixel ROCAUC using both Resnet-18 and WideResnet-50 as pretrained network.

Damage region	Resnet-18 Pixel ROCAUC	Wideresnet-50 Pixel ROCAUC
Top left	88.4 %	86.9 %
Top right	86.3 %	87.4 %
Top centre	78.2 %	84.1 %
Middle left	75.3 %	91.1 %
Middle right	67.9 %	88.3 %
Middle centre	72.1 %	92.3 %
Bottom left	88.4 %	96.1 %
Bottom right	86.0 %	84.1 %
Bottom centre	78.1 %	92.6 %
Average	80.1 %	89.2 %

necessary for making prediction.

Since the model has been saved into a file from the previous stage, the file is loaded using the exact path to which it was saved previously.

This is done using the `load_model` function. Prediction involves a whole new image that has not been seen by the model before. If a training image or test image is used during prediction, the accuracy of the model

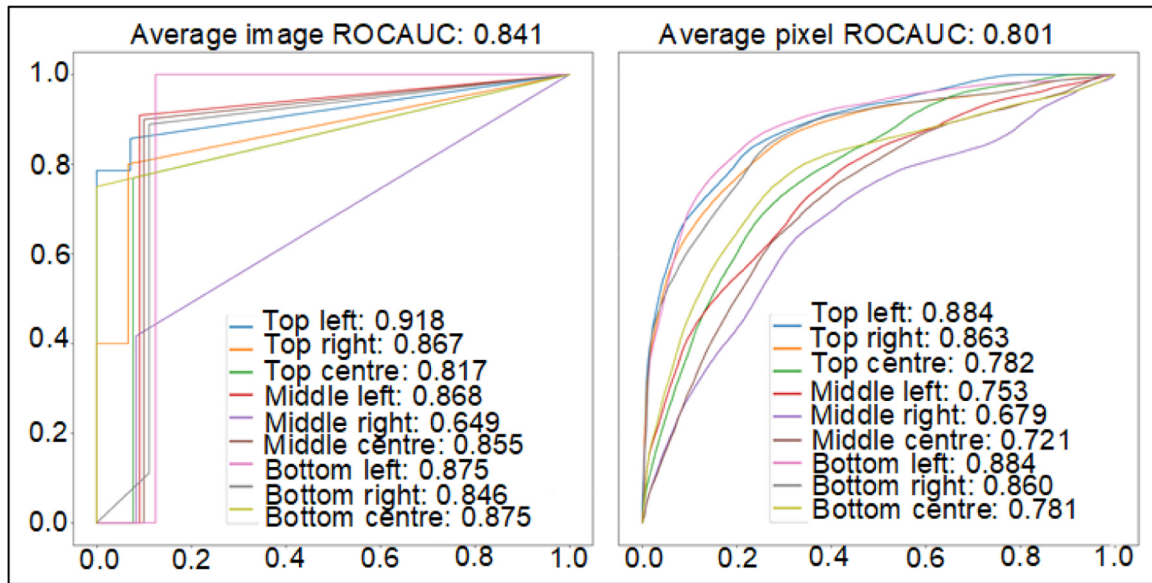


Fig. 32. Graphical representation of image and pixel ROCAUC using pretrained Resnet-18 by damage region.

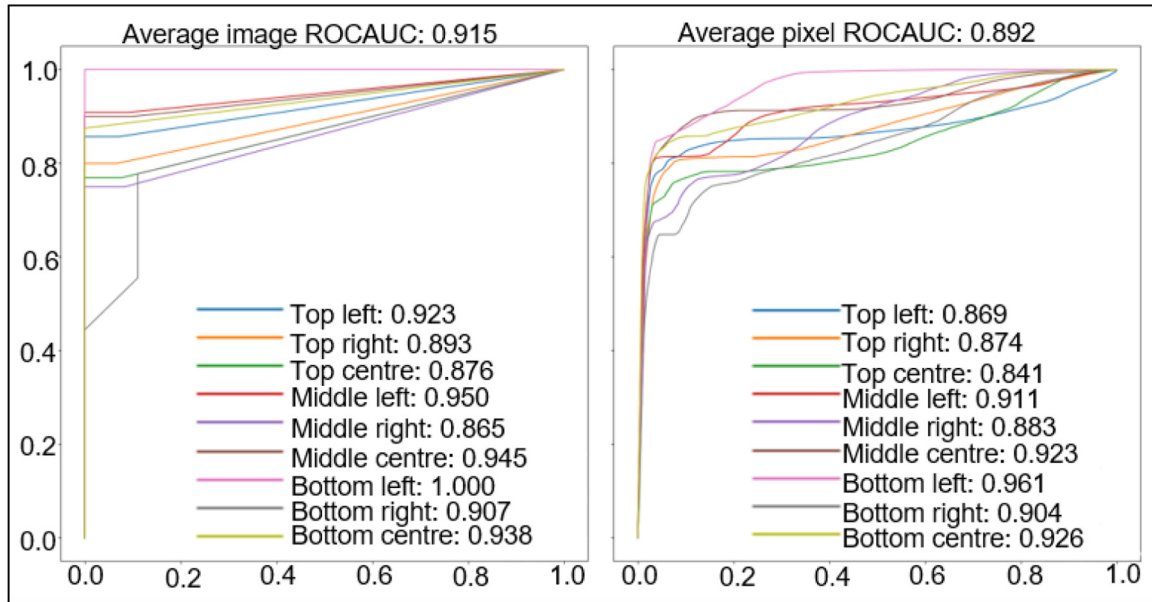


Fig. 33. Graphical representation of image and pixel ROCAUC using pretrained Wideresnet-50 by damage region.

cannot be ascertained since the model already knows the label for that image beforehand. The path of the new image was specified as a parameter for the `load_img` function.

The image was converted into an array using `NumPy` and `reshaped` as a pre-processing stage for the prediction process. The image was passed as a parameter to the `predict` function and then returns the learned label. In this case, the `predict` function returns a float value that dictates a number between 0 and 1 which shows the probability of the distribution towards a particular class.

The confusion matrix is represented by Fig. 25 and shows the model's performance in predicting positive and negative corrosion in the data. Even though there are very few false negatives and false positives, the model still shows a prominent level of accuracy in making correct predictions against the true labels.

After getting the label, as would be done in most application-level solutions, different forms of programming logic can be used in

modifying the final output of the result. In this case, an if condition is used to assign label values to an intuitive string value. Threshold was set to 0.9 such that values lesser than 0.9 refers to sample surfaces with defected damage status while values greater than 0.9 refers to sample surfaces with defected damage status.

4.2. Evaluation of stage detector

A simple Multiclass CNN model was used for the implementation of the corrosion stage detector. The model summary is seen in Table 6. The fully connected layer has an output of 6 values including no-defect and all the other 5 corrosion stages.

The training of the sample images was done with 1500 images and a batch size of 20 running at 30 epochs. The best accuracy was recorded at 92.1 %. The training and validation accuracy at 30 epoch is seen in Fig. 26 along with the training and validation loss. The classification

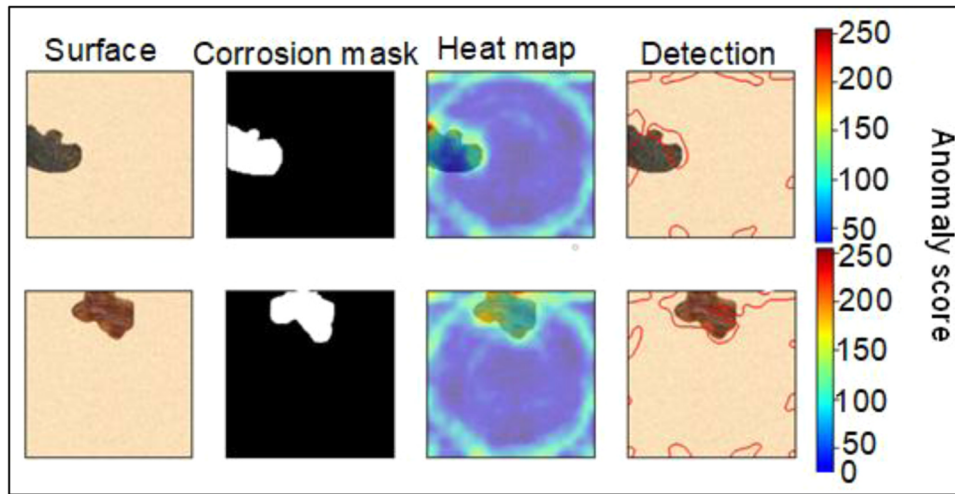


Fig. 34. Anomaly detection based on corrosion region using pretrained Resnet-18.

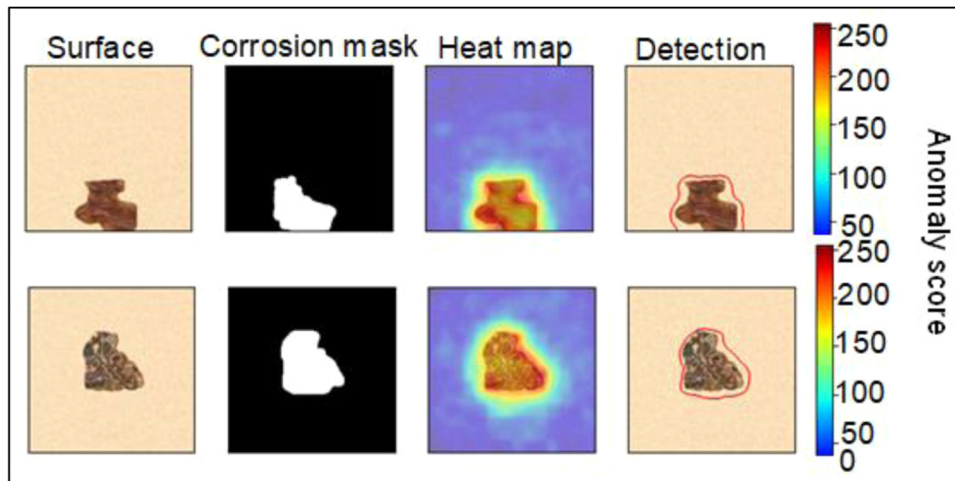


Fig. 35. Anomaly detection based on corrosion region using pretrained Wideresnet-50.

report is also seen in Table 6, showing the precision, recall and f1 score for the multiclass model. The confusion matrix is seen in Fig. 27 shows the true label against the predicted label, this indicates an impressive approach to how the model makes prediction across the different classes (Table 7).

4.3. Evaluation of the anomaly detection model

Anomaly detection stage is implemented with a patch distribution algorithm. This makes use of a pretrained convolutional neural network and then used to capture intermediate feature outputs during inference i.e., training is done on normal class and then anomalies are detected and estimated during testing. The patch distribution is evaluated for data at the distinct stages of corrosion separately. Due to the nature of the algorithm, only the normal class is used for trained but during the testing stage, the various stages are used to test to see how the algorithm performs on terms of detection of specific patches. The evaluation is done by calculating the anomaly score. The anomaly score was calculated on two levels. The first level was done on the image level while the second score is done on the Pixel level.

The classifier is looped during testing for detecting anomaly in various stages. The image ROCAUC and Pixel ROCAUC was calculated for each of them and compared. The accuracy for detecting anomaly in the various stages on the image level were recorded and compared for

both Resnet-18 and Wideresnet-50 as seen in Table 8. The best result was 99.2 % for stage four anomaly using Resnet-18 while the lowest score was 91.7 % for stage one also using Resnet-18. Meanwhile, the average Image ROCAUC score of all stages is 96.5 % and 96.4 % for pretrained Resnet-18 and Wideresnet-50 respectively showing Resnet-18 to have a slightly higher performance.

At the pixel level, pretrained Resnet-18 has a low performance with an average pixel ROCAUC of 65 % anomaly detection across all stages while pretrained wideresnet-50 had an excellent average ROCAUC performance of 94.6 % anomaly detection across all stages. Despite the low performance of pretrained Resnet-18, a significant 81.3 %-pixel ROCAUC anomaly detection was recorded for stage one which declines further to 53.0 % for stage five. Pretrained Wideresnet-50 on the other hand has a steady increase in anomaly detection at the pixel level across the stages. A full record of all pixel level anomaly detection for all stages is seen in Table 9.

The image and pixel ROCAUC for anomaly detection at various stages using pretrained Resnet-18 and wideresnet-50 is seen in Figs. 28 and 29, respectively.

The actual results as seen on tested images shows a corrosion mask which is a 2-channel image with the normal image as the black background while the anomalies are the white patches on the masks. In addition, a heat map is generated to clearly show the anomaly areas as it affects the pixel on the image surface. Depending on the anomaly score,

Table 12

Result comparison of proposed method with published results.

Deep learning algorithm	Dataset	Year	Author	Accuracy	Defect category
Residual neural networks(ResNet)	Electric Power Research Institute (EPRI) dataset	2021	Papamarkou et al.	79.60 % (F1 score)	Corroded and intact
Ensemble deep learning with convolutional neural networks (CNNs)	Real world drone images	2022	Forkan et al.	86.28 % (pixel level) 92.50 % (image level)	Corrosion and non-corrosion
Yolov3-tiny, DSConv layers, CBAM, spatial pyramid pooling (SPP)	Field inspection image dataset	2023	Yu et al.	84.96 %	Nubby corrosion, bar corrosion, exfoliation, and fastener corrosion
Dilated convolutional neural network (DiLCNN) and a capsule neural network for ToA (CapsToA)	Simulated dataset	2023	Donati, Zonzini and Marchi	70 %	Undefined
CNN	Random pitting data	2023	Jiang et al.	90.2 %	corrosion damaged and corrosion undamaged
3DCNN encoder, a ConvLSTM decoder, and a ConvLSTM future predictor	Pipe leaking dataset	2023	Karimanzira	96 %	Leaking and non-leaking
Improved Yolov5	Original Corrosion dataset	2023	Ren et al.	74.6 %	Undefined
PreActResNet18 Resnet50	Multimodal dataset	2023	Schafer, Faltings & Glaser	68.09 %	Varying, Imbalance
Proposed multi-phase corrosion detection and region identification method	Simple CNN	2024	Oyedeleji, Khan & Erkoyuncu	94.87 %	Positive corrosion and negative corrosion status
	Optimised VGG16			92.1 %	Corrosion Stages a - f
	Stage-based One class learning CNN with Patch distribution algorithm			96.5 % (Image level), 96.4 % (Pixel level)	1 normal class against Corrosion Stages a - f
	Region-based One class learning CNN with Patch distribution algorithm			91.5 % (Image level), 89.2 % (Pixel level)	1 normal class against 9 distinct regions

color red is used to depict the areas with highest damages on the surface. Fig. 30 shows some of these results as generated when using pretrained resnet-18 from stages one to five while Fig. 31 shows some of the results when using pretrained wideresnet-50.

4.4. Evaluation of region detection model

The region detection stage makes use of the patch distribution algorithm like the one used in the previous stage. However, since the aim of this stage is to provide specific information based on the exact region of damage on the surface of the material being inspected, the image data used was re-structured. The normal images are used in the initial stage of the model in addition to the pretrained rsnet-18 and wideresnet-50, however, images with specific regions are used in isolating the damages.

Results for all the nine named damage region were recorded with the two pretrained model at image and pixel level. During image level analysis, an average ROCAUC anomaly detection accuracy of 84.1 % and 91.5 % was recorded for pretrained resnet-18 and resnet-50, respectively. During pixel level analysis, 80.1 % was recorded for the average pixel ROCAUC anomaly detection using resnet-18 as pretrained model. 89.2 % was recorded for the average pixel ROCAUC anomaly detection using wideresnet-50 as pre-trained model. Full anomaly detection accuracy for the various regions is seen in Tables 10 and 11.

A graphical view of the anomaly detection accuracy across the nine damage regions is seen in Figs. 32 and 33 for resnet-18 and wideresnet-50 respectively showing the trend of the model performance. The actual results of the region detection are found in Figs. 34 and 35 clearly showing the damage masks, predicted heat map and detection points for sample images of different damage regions. As shown in the score tables, using WideResnet as a pre-trained network in the patch distribution algorithm has a better performance at detecting the exact region of damage, this is also corroborated by the results in Fig. 35 where the detection point fits the exact region of damage unlike resnet-18 in Fig. 34 where false detection occurs in some regions on the damage surface.

4.5. Result comparison

Results from the proposed methodology is compared with published results as displayed in Table 12. In terms of single neural networks, ResNet was used in 2021 by Papamarkou et al. [22] with the Electric Power Research Institute (EPRI) dataset to classify corroded and intact data and yielded an F1 score of 79.60 %. [15] made use of pure Convolutional neural network for binary corrosion classification with an accuracy of 90.2 %. Dataset of European scrap classes (DOES) research produced a dataset of corrosion images with varying levels of corrosion defects [30]. Our method produced a result of 94.87 % at the binary corrosion status detection level.

Improved versions of models such as Yolov5 [27], Yolov3-tiny[39] also yielded increased results of 74.6 % and 84.96 % respectively. Here, our proposed methodology yielded a result of 91.2 % with the improved VGG16.

With regards to hybridisation of deep learning models, Ensemble deep learning underpinned by convolutional neural networks (CNNs) achieved 86.28 % at pixel level and 92.50 % image level [11], Dilated convolutional neural network (DiLCNN) and a capsule neural network for ToA (CapsToA) achieved a result of 70 % [10]. 3DCNN encoder, a ConvLSTM decoder, and a ConvLSTM future predictor also yielded an accuracy of 96 % on a pipe leaking dataset [17]. Our proposed method achieved an accuracy of 96.5 % at Image level and 96.4 % at the Pixel level for the patch distribution algorithm.

5. Conclusions

In this study, a multi-phase corrosion identification and region detection was developed to tackle the problem of corrosion. The best way to identify corrosion is through visual mechanism, hence images of exposed alloy materials were used as data for the neural networks at different stages. The first phase also known as the binary test phase helps to identify the presence of corrosion on alloy and the simple binary CNN model has an excellent performance of 94.87 % when the image is first converted to grayscale while inspecting. The second phase also known as the multi class phase provides more information by identifying the stage

at which the damage is using the predefined stages modelled in the study and the simple multiclass CNN used here delivered a classification accuracy of 92.1 %. The third phase makes use of Patch distribution algorithm to determine the distribution of the damages on the surface of the metal by producing a heat map with the best average accuracy of 94.6 % and 96.5 % at Pixel and image level respectively. The last stage helps to isolate the exact region of corrosion on the material surface with a best average accuracy of 91.5 % and 89.2 % at Image and Pixel level, respectively.

The proposed methodology is justified by the performance. However, it is highly recommended that the method be implemented for a real time corrosion system. There is ongoing research on the use of the methodology for detecting corrosion and further research is also encouraged to investigate and observe the multi-phase corrosion detection and region identification in real time. Also, the introduction of data from other domain for defect detection is encouraged for future works to evaluate the performance of the method in different scenarios other than corrosion detection. Other techniques such as the stochastic model [7] and BAM neural networks [25] can also be used in future works to manage noise in the corrosion images and improve the quality of the classification model .

CRedit authorship contribution statement

Samir Khan: Methodology, Project administration, Supervision, Writing – review & editing. **John Ahmet Erkoyuncu:** Project administration, Supervision, Writing – review & editing, Investigation, Methodology. **Oluseyi Ayodeji Oyediji:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

References

- [1] F.E. Abeng, B.E. Nyong, M.E. Ikpi, M.E. Obeten, The role of computational chemistry in corrosion inhibition: a review, *Port. Electrochim. Acta* 40 (4) (2022), <https://doi.org/10.4152/pea.2022400402>.
- [2] M. Abubakar, M. Rady, K. Badran, S.Y. Mahfouz, Application of deep learning in damage classification of reinforced concrete bridges, *Ain Shams Eng. J.* (2023), <https://doi.org/10.1016/j.asej.2023.102297>.
- [3] C. Allen, S. Aryal, T. Do, R. Gautum, M.M. Hasan, Bharat K. Jasthi, E. Gnimpeba, V. Gadhamshetty, Deep learning strategies for addressing issues with small datasets in 2D materials research: microbial Corrosion, *Front. Microbiol.* 13 (2022), <https://doi.org/10.3389/fmicb.2022.1059123>.
- [4] B.T. Bastian, N. J, S.K. Ranjith, C.V. Jiji, Visual inspection and characterization of external corrosion in pipelines using deep neural network, *NDT E Int.* 107 (2019), <https://doi.org/10.1016/j.ndteint.2019.102134>.
- [5] L. Bertolini, M. Gastaldi, M.P. Pedferri, E. Redaelli, Prevention of steel corrosion in concrete exposed to seawater with submerged sacrificial anodes, *Corros. Sci.* 44 (7) (2002) 1497–1513, [https://doi.org/10.1016/S0010-938X\(01\)00168-8](https://doi.org/10.1016/S0010-938X(01)00168-8).
- [6] Biao Yin, Nicholas Josselyn, Thomas Considine, John Kelley, Berend Rinderspacher, Robert Jensen, James Snyder, Ziming Zhang, and Elke Rundensteiner. Corrosion image data set for automating scientific assessment of materials, in: Proceedings of the British Machine Vision Conference (BMVC 2021). (<https://www.bmvc2021-virtualconference.com/assets/papers/1544.pdf>).
- [7] Yang Cao, A. Chandrasekar, T. Radhika, Velusamy Vijayakumar, Input-to-state stability of stochastic Markovian jump genetic regulatory networks, *Math. Comput. Simul.* (2023), <https://doi.org/10.1016/j.matcom.2023.08.007>.
- [8] Defard, Thomas & Setkov, Aleksandr & Loesch, Angelique & Audigier, Romaric. (2021). PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization. https://doi.org/10.1007/978-3-030-68799-1_35.
- [9] F. Defforian, Y. Hou, D. Lei, S. Li, W. Yang, C. Li, Experimental investigation on corrosion effect on mechanical properties of buried metal pipes, *Int. J. Corros.* 2016 (2016) 5808372, <https://doi.org/10.1155/2016/5808372>.
- [10] G. Donati, F. Zonzini, L. de Marchi, Tiny deep learning architectures enabling sensor-near acoustic data processing and defect localization, *Computers* 12 (7) (2023), <https://doi.org/10.3390/computers12070129>.
- [11] A.R.M. Forkan, Y.Bin Kang, P.P. Jayaraman, K. Liao, R. Kaul, G. Morgan, R. Ranjan, S. Sinha, CorrDetector: a framework for structural corrosion detection from drone images using ensemble deep learning, *Expert Syst. Appl.* 193 (2022), <https://doi.org/10.1016/j.eswa.2021.116461>.
- [12] S. Harsimran, K. Santosh, K. Rakesh, Overview of corrosion and its control: a critical review, *Proc. Eng. Sci.* 3 (1) (2021), <https://doi.org/10.24874/PES03.01.002>.
- [13] Hinton, B.R.W. & Trueman, Antony & Galea, Silvia & Cole, Ivan & Miah, S. & Davidson, L. & Liu, Qianchu (2008). Corrosion control on Australian Defence Force aircraft - future challenges, in: Proceedings of the Forty Eighth Annual Conference of the Australasian Corrosion Association 2008: Corrosion and Prevention 2008.
- [14] Z. Huang, C. Feng, J. Wang, W. Liu, Y. Xie, Research on metal corrosion mechanism and inhibition measures of transmission and distribution equipment in Hunan, *IOP Conf. Ser. Earth Environ. Sci.* 621 (1) (2021), <https://doi.org/10.1088/1755-1315/621/1/012035>.
- [15] X. Jiang, H. Qi, X. Qiang, B. Zhao, H. Dong, A convolutional neural network-based corrosion damage determination method for localized random pitting steel columns, *Appl. Sci. (Switz.)* 13 (15) (2023), <https://doi.org/10.3390/app13158883>.
- [16] X. Jiang, Y. Yan, Y. Su, Data-driven pitting evolution prediction for corrosion-resistant alloys by time-series analysis, *npj Mater Degrad* 6 (2022) 92, <https://doi.org/10.1038/s41529-022-00307-4>.
- [17] D. Karimanzira, Simultaneous pipe leak detection and localization using attention-based deep learning autoencoder, *Electronics* 12 (22) (2023), <https://doi.org/10.3390/electronics12224665>.
- [18] R. Lemos, R. Cabral, D. Ribeiro, R. Santos, V. Alves, A. Dias, AutomatiC Detection of Corrosion in Large-scale Industrial Buildings Based on Artificial Intelligence and Unmanned Aerial Vehicles, *Appl. Sci. (Switz.)* 13 (3) (2023), <https://doi.org/10.3390/app13031386>.
- [19] S. Li, X. Yue, Q. Li, H. Peng, B. Dong, T. Liu, Q. Jiang, Development and applications of aluminum alloys for aerospace industry, *J. Mater. Res. Technol.* 27 (2023) 944–983, <https://doi.org/10.1016/j.jmrt.2023.09.274>.
- [20] G. McAdam, P.J. Newman, I. McKenzie, C. Davis, B.R.W. Hinton, Fiber optic sensors for detection of corrosion within aircraft, *Struct. Health Monit.* 4 (1) (2005), <https://doi.org/10.1177/1475921705049745>.
- [21] S. Panigrahy, S. Karmakar, R. Sahoo, Condition assessment of high voltage insulator using convolutional neural network, in: roceedings of CONECT 2021: Seventh IEEE International Conference on Electronics, Computing and Communication Technologies (2021), <https://doi.org/10.1109/CONECT52877.2021.9622361>.
- [22] T. Papamarkou, H. Guy, B. Kroencke, J. Miller, P. Robinette, D. Schultz, J. Hinkle, L. Pullum, C. Schuman, J. Renshaw, S. Chatzidakis, Automated detection of corrosion in used nuclear fuel dry storage canisters using residual neural networks, *Nucl. Eng. Technol.* 53 (2) (2021) 657–665, <https://doi.org/10.1016/j.net.2020.07.020>.
- [23] O. Prentkovskis, P. Maruschak, S. Panin, F. Berto, Application of alloys in transport, *Metals* 13 (2023) 31, <https://doi.org/10.3390/met13010031>.
- [24] G. Qian, K. Tantratian, L. Chen, Z. Hu, M.D. Todd, A probabilistic computational framework for the prediction of corrosion-induced cracking in large structures, *Sci. Rep.* 12 (1) (2022), <https://doi.org/10.1038/s41598-022-25477-8>.
- [25] T. Radhika, A. Chandrasekar, Velusamy Vijayakumar, Quanxin Zhu, Analysis of Markovian jump stochastic cohen–grossberg bam neural networks with time delays for exponential input-to-state stability, *Neural Process. Lett.* 55 (2023) 1–18. (<https://doi.org/10.1007/s11063-023-11364-4>).
- [26] D. Rajendran, T. Sasilatha, S.A.D.H. Mary, S.S. Rajendran, C. Lacnjevac, G. Singh, Deep learning based underwater metal object detection using input image data and corrosion protection of mild steel used in underwater study-A case study Part A-Deep learning based underwater metal object detection using input image data, *Mater. Prot.* 63 (1) (2022) 5–14, <https://doi.org/10.5937/zasmat2201005R>.
- [27] Ren, X., Hu, S., Hou, Y., & Ke, Y. (2023). Morphology and Transformer-Based YOLOv5 for Workpiece Surface Crack Detection. <https://doi.org/10.3233/fia230835>.
- [28] A. Samide, C. Stoean, R. Stoean, Surface study of inhibitor films formed by polyvinyl alcohol and silver nanoparticles on stainless steel in hydrochloric acid solution using convolutional neural networks, *Appl. Surf. Sci.* 475 (2019) 1–5, <https://doi.org/10.1016/j.apsusc.2018.12.255>.
- [29] A.S. Saragih, F. Aditya, W. Ahmed, Defect identification and measurement using stereo vision camera for in-line inspection of pipeline, in: Proceedings of the Advances in Science and Engineering Technology International Conferences, ASET 2022 (2022), <https://doi.org/10.1109/ASET53988.2022.9735082>.
- [30] M. Schäfer, U. Faltings, B. Glaser, DOES - A multimodal dataset for supervised and unsupervised analysis of steel scrap, *Sci. Data* 10 (1) (2023), <https://doi.org/10.1038/s41597-023-02662-6>.
- [31] L. Shang, Z. Zhang, F. Tang, Q. Cao, N. Yodo, H. Pan, Z. Lin, Deep learning enriched automation in damage detection for sustainable operation in pipelines with welding defects under varying embedment conditions, *Computation* 11 (11) (2023), <https://doi.org/10.3390/computation11110218>.
- [32] E.A. Starke, J.T. Staley, Application of modern aluminum alloys to aircraft, *Prog. Aerosp. Sci.* 32 (2) (1996) 131–172, [https://doi.org/10.1016/0376-0421\(95\)00004-6](https://doi.org/10.1016/0376-0421(95)00004-6).
- [33] Tanwir Alam, M., & Husain Ansari, A. (2017). REVIEW ON ALUMINIUM AND ITS ALLOYS FOR AUTOMOTIVE APPLICATIONS Maulana Mukhtar Ahmad Nadvi

- Technical Campus Malegaon. (<https://www.researchgate.net/publication/317075488>).
- [34] N. Truong, Using back titration as a time-saving technique, *Pract. Oil Anal.* 10 (2008) 1–2. (<https://www.scopus.com/inward/record.uri?eid=2-s2.0-39749185436&partnerID=40&md5=40a1e174aaefd2879116ea1321981206>).
- [35] D. Vriesman, A. Britto Junior, A. Zimmer, A. Lameiras Koerich, Texture CNN for thermoelectric metal pipe image classification, in: *Proceedings of the International Conference on Tools with Artificial Intelligence, ICTAI, 2019-November (2019)* 569–574, <https://doi.org/10.1109/ICTAI.2019.00085>.
- [36] Y. Wang, X. Shen, K. Wu, M. Huang, Corrosion grade recognition for weathering steel plate based on a convolutional neural network, *Meas. Sci. Technol.* 33 (9) (2022), <https://doi.org/10.1088/1361-6501/ac7034>.
- [37] B. Woodruff, Coatings in power plants controlling fouling and corrosion in tubing, *J. Prot. Coat. Linings* 22 (2005) 11. (<https://www.scopus.com/inward/record.uri?eid=2-s2.0-33750040851&partnerID=40&md5=dedaea45a47159059d722427070f35e4>).
- [38] Y. Yao, Y. Yang, Y. Wang, X. Zhao, Artificial intelligence-based hull structural plate corrosion damage detection and recognition using convolutional neural network, *Appl. Ocean Res.* 90 (2019), <https://doi.org/10.1016/j.apor.2019.05.008>.
- [39] L. Yu, E. Yang, C. Luo, P. Ren, AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection, *J. Ambient Intell. Humaniz. Comput.* 14 (7) (2023) 8087–8098, <https://doi.org/10.1007/s12652-021-03580-4>.
- [40] Xin'an Yuan, Wei Li, Guoming Chen, Xiaokang Yin, Xiao Li, Jie Liu, Jianchao Zhao, Zhao Jianming, Visual and intelligent identification methods for defects in underwater structure using alternating current field measurement technique, *IEEE Trans. Ind. Inform.* (2021) 1–1, <https://doi.org/10.1109/TII.2021.3117034>.
- [41] K. Zhang, Houqiang, W. Huang, Defect Detection of Anti-vibration Hammer Based on Improved Faster R-CNN, in: *Proceedings of the Seventh International Forum on Electrical Engineering and Automation, IFEEA 20200 (2020)* 889–893. (<https://doi.org/10.1109/IFEEA51475.2020.00186>).
- [42] D. Zhang, J. Srinivasan, J.S. Locke, Quantitative evaluation of the ability of accelerated tests to simulate on-road corrosion morphology of automotive aluminum alloys, *Corrosion* 78 (12) (2022), <https://doi.org/10.5006/4174>.
- [43] Q. Zhao, L. Li, L. Zhang, Identification of corrosion on the inner walls of water pipes using a vgg model incorporating attentional mechanisms, *Appl. Sci.* 12 (24) (2022), <https://doi.org/10.3390/app122412731>.
- [44] Y. Zhu, X. Liu, S. Li, Y. Wan, Q. Cai, Wind turbine blade defect detection based on acoustic features and small sample size, *Machines* 10 (12) (2022), <https://doi.org/10.3390/machines10121184>.
- [45] K. Zuchniak, W. Dzwiniel, E. Majerz, A. Pasternak, K. Dragan, Corrosion detection on aircraft fuselage with multi-teacher knowledge distillation, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2021) 318–332, https://doi.org/10.1007/978-3-030-77980-1_25.



DSPACE

<https://dspace.org/>

Application of CNN for multiple phase corrosion identification and region detection

Oyedeji, Oluseyi Ayodeji; Khan, Samir; Erkoyuncu, John Ahmet

2024-10-30

Elsevier

<https://doi.org/10.1016/j.asoc.2024.112008>

Oyedeji OA, Khan S, Erkoyuncu JA. (2024) Application of CNN for multiple phase corrosion identification and region detection. Applied Soft Computing, Volume 164, October 2024, Article number 112008