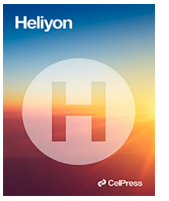
Heliyon 10 (2024) e34882

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Deep neural networks for external corrosion classification in industrial above-ground storage tanks

Anibal Alviz-Meza [a,*], Leidy L. Hadechini-Meza [b], Darío Y. Peña-Ballesteros [c]

[a] Grupo de Investigación en Arquitectura, Diseño e Ingeniería (GIADI), Facultad de Ingeniería y Arquitectura, Institución Universitaria Mayor de Cartagena, Cartagena, Colombia
[b] Grupo de Investigación Ciudad, Educación y Sociedad (CEUS), Facultad de Ciencias Sociales y Educación, Institución Universitaria Mayor de Cartagena, Cartagena, Colombia
[c] Grupo de Investigaciones en Corrosión (GIC), Universidad Industrial de Santander, Parque Tecnológico Guatiguará, Piedecuesta, 681011, Colombia

ARTICLE INFO

Keywords:
Convolutional neural networks
Transfer learning
Fine-tuning
External corrosion
Above-ground storage tanks

ABSTRACT

This paper uses transfer learning and fine-tuning approaches to explore EfficientNet smaller models for a multiclassification task of corrosion types in above-ground storage tanks. Data augmentation was used to increment the data an oil and gas company provided, reaching a dataset of around 5000 images. The images were stored in Google Drive and imported into Colab to train the models using TensorFlow and Keras. After the hyperparameters' tuning a transfer learning model was selected and explored with fine-tuning. The EfficientNetB0 model delivered from fine-tuning accomplished 94 % performance. This work is the first attempt to deploy an artificial vision automatic tool for being implemented during tank inspection in the industrial sector. In further development, this model can be coupled with one based on object detection to remotely identify failures due to external corrosion issues, improving safety and reliability in oil and gas operations.

1. Introduction

Oil and gas industries are still the main pillars of the world economy. For the 2020–2030 decade, the maximum production of conventional oil is expected, around 4.5–4.8 billion tons per year [1]. These industries are facing challenges such as transitioning to more sustainable operations and energy sources [2–5], as well as incorporating artificial intelligence in their processes [6]. This last aspect has a huge niche and applicability in the external corrosion detection of metallic structures. For instance, in above-ground storage tanks containing hydrocarbons, pitting corrosion —the primary concern— can be identified through machine learning models to prevent the release of toxic and flammable compounds [7]. It is not only a matter of safety but also of productivity since without said models, industries must cease operations to conduct regular preventive inspections.

The external corrosion of above-ground storage tanks (ASTs) is initiated when the coating or paint covering the metallic structure fails. The most commonly associated failures are

Pitting, crevice, galvanic, blistering, and uniform corrosion [8]. Pitting is evidenced by leaks, holes, and gouges (Fig. 1a), but it can also be associated with microbiologically induced corrosion. Crevice corrosion results in the formation of crevices (Fig. 1b), while

* Corresponding author. Institución Universitaria Mayor de Cartagena, 130001, Cartagena, Colombia.
E-mail address: a.formulador@umayor.edu.co (A. Alviz-Meza).

https://doi.org/10.1016/j.heliyon.2024.e34882
Received 30 August 2023; Received in revised form 17 July 2024; Accepted 18 July 2024
Available online 20 July 2024

galvanic corrosion causes section loss (Fig. 1c), and blistering corrosion manifests as the emergence of bubbles beneath the coating (Fig. 1d). Uniform corrosion (Fig. 1e) is the consequence of homogeneous chemical reactions in conjunction with the surface extension of a component. Stress corrosion cracking is less prevalent in storage tanks, although it can occur at the tank's base when the storage medium becomes more saline, when the tank is approaching the end of its service life [9], or when ethanol is present [10].

Introducing corrosion modeling in industry has covered several issues such as detection, prediction, and classification [11,12]. In particular, Convolutional Neural Networks (CNNs) have become the most utilized models for corrosion detection. CNNs extract features from large image datasets without the need for manual extraction, which is a common limitation of shallow machine learning models [13]. The parallelization provided by GPU computing and the use of cloud computing has enabled convolutional neural networks (CNNs) to achieve human-equivalent performance in image classification and segmentation tasks [14]. The processes involved in developing CNN models include image acquisition and preprocessing, architecture selection, hyperparameter tuning, and evaluation.

Most studies available in the scientific literature of CNN's models for corrosion detection in the oil and gas industry have been developed for pipelines [6,15]. Custom algorithms detect corrosion and evaluate its magnitude as low, medium, and high. This is achieved following a training process with a dataset of 140 K images [16]. The algorithms have been used in object detection for pitting, galvanic, and intergranular corrosion risk assessment in oils and gas facility systems. This was based on 36 images, using data augmentation, and Faster R–CNN with ResNet 50 architecture [17]. Offshore pipelines have been studied to develop models for corrosion detection and classification according to their grade, based on 4000 underwater images, achieving an accuracy of 81 % [18]. Other approaches for corrosion assessment on metallic surfaces have reached 98.5 % performance using more than 67 K images for corrosion detection through fine-tuning the VGG-16 model [19]. The structural damage detection method based on Faster R–CNN can also detect concrete cracks, steel high and medium corrosion, bolt corrosion, and steel delamination based on 2.3 K images, reaching 89.7 % testing performance. Furthermore, semantic segmentation by PSPNet and Mask R–CNN has led to corrosion detection with a performance of 73.2 %, which has been implemented in industrial inspections [20]. Similarly, tools such as RustSEG have emerged for segmenting images for automated corrosion detection without the requirement of per-pixel labeled datasets for training, achieving a confidence of 99 % [21].

Despite the above studies, there is limited information regarding CNN models for corrosion classification in ASTs. The closest solved corrosion classification problems found in the literature are related to metallic corrosion [22,23]. These studies trained custom convolutional neural network (CNN) models. One model was trained to classify corrosion grades, achieving an accuracy of 93.8 % [23], while the other was trained to detect external corrosion, achieving an accuracy of 85 % [22]. However, none of these models were designed to specifically address the external corrosion issues commonly found in ASTs. The objective of this research is to address the specific issue of corrosion classification in oil and gas ASTs by exploring the potential of pre-trained tailored to small datasets CNN models. The developed model will be used to classify pitting corrosion, crevice corrosion, galvanic corrosion, blister corrosion, and uniform corrosion. The novelty of this work is the development of the first custom CNN model for external corrosion classification for oil and gas ASTs. This study contributes to remote monitoring and the development of safe operations in the oil and gas sector.

This document is structured as follows: Section 2 includes the methodological approaches and technical decisions to implement the

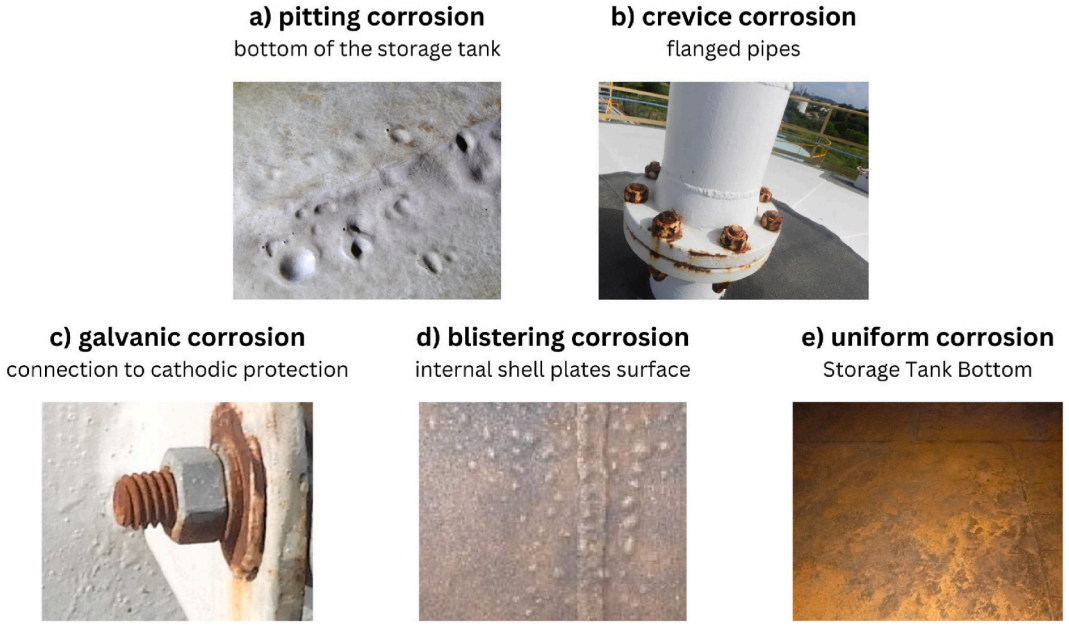


Fig. 1. External corrosion in above-ground storage tanks. a) pitting corrosion, b) crevice corrosion, c) galvanic corrosion, d) blistering corrosion, and e) uniform corrosion.

EfficientNet CNN model. Section 3 is dedicated to results, but it also comprises the customized CNN architectures proposed and the parameters, hyperparameters, and trials performed —experimental design. Section 4 is the discussion of results, and Section 5 is the conclusions.

2. Corrosion classification approach

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has facilitated the growth of artificial vision by enabling the deployment of powerful model architectures. These models are currently employed for transfer learning and fine-tuning in multiple artificial vision tasks. One of the most effective architectures was recently achieved by Efficient-Net models through novel scaling methods that uniformly scale all dimensions of depth, width, and resolution by employing constant ratios based on MobileNets and ResNet convolutional neural networks (CNNs).

2.1. Efficient-Net

In particular, EfficientNet-B7 achieves state-of-the-art 84.3 % in top-1 accuracy on ImageNet (see Fig. 2), while being 8.4× smaller and 6.1× faster on inference than the best existing convolutional neural networks (CNNs) [24]. The EfficientNet architecture employs the AutoML MNAS framework to optimize accuracy and efficiency. Furthermore, it utilizes the mobile inverted bottleneck convolution (MBConv) technique to enhance the floating-point operations per second (FLOP) budget. The fundamental model, designated as EfficientNetB0, is illustrated in Fig. 3, which was subsequently scaled up to yield the EfficientNetB7 current model [25].

2.2. Dataset acquisition and preprocessing

In the absence of freely available data sources on the internet, particularly in the case of ASTs, private companies represent one of the few avenues for obtaining images. The images in question were captured over a period of five years during tank inspections. The diversity of the images was sufficient to organize them into five files, each corresponding to a specific type of corrosion: galvanic, crevice, blistering, pitting, and uniform corrosion. However, due to the abundance of poor-quality, edited, and non-relevant images, the dataset was consolidated with only 800 images. Stress corrosion cracking and other types of corrosion were not identified in the tanks inspected from 2017 to 2023. All the images were labeled and resized to 224 × 224 pixels in Colab as required for transfer learning with EfficientNet. The five files containing images selected to illustrate each class were compressed and deposited as a single zip file in Google Drive for use in Google Colab notebooks.

2.3. Data augmentation

Several techniques for data augmentation were employed in order to gather the largest possible image dataset, to avoid overfitting the customized EfficientNet-based models selected (those with the smaller number of parameters). These techniques included slight rotation (nearest fill mode), brightness, shear, and zoom changes, as well as minor channel/width/height shifts and horizontal/vertical flips. The resulting dataset was then distributed across three categories: training, validation, and testing. Table 1 shows the distribution of images for each of these categories after data augmentation. A total of 10–15 % of the original data was randomly designated for testing, while 15–30 % of the remaining data was allocated for validation, to stratify the data. It is also worth noting that, for all these experiments, a random seed was set in Python to obtain a similar data distribution across runs, in order to ensure comparable performances.

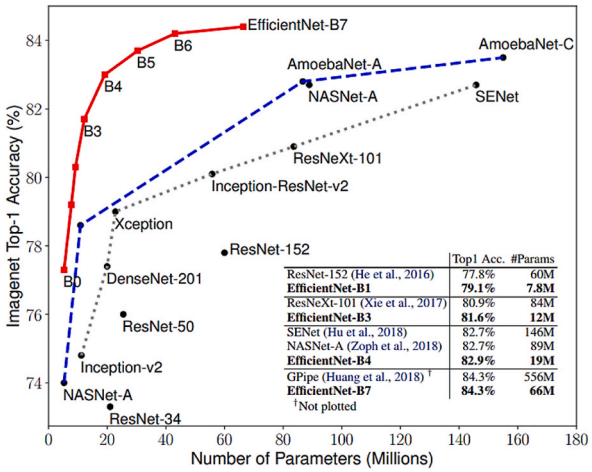


Fig. 2. Model Size vs. ImageNet Accuracy [24].

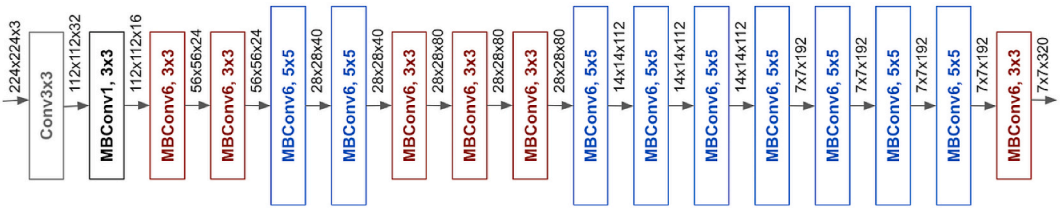


Fig. 3. The architecture of network EfficientNet-B0.

Table 1
Images of each type of corrosion are used for training, validation, and testing after data augmentation.

	Original	Training (85/70 %)	Validation (15/30 %)	Testing (15/10 %)
Pitting	200 - (25.0 %)	1296/1134	26/54	30/20
Crevice	156 - (19.5 %)	1017/882	20/42	23/16
Blistering	140 - (17.5 %)	918/792	17/38	21/14
Galvanic	154 - (19.2 %)	999/603	20/42	23/15
Uniform	150 - (18.8 %)	981/855	19/40	22/15

2.4. Evaluation metrics

The capacity of generalization of the model was evaluated using four metrics: accuracy, precision, recall, and F1 score (harmonic mean of precision and recall). However, to address the slight imbalance of classes in our dataset, we elected to concentrate our efforts on the weighted F1 score during the training phase. The mathematical expressions for such metrics are defined in Equations (1)–(3). In addition, we employed a confusion matrix to illustrate the performance of the trained models in testing with unseen images.

$$Precision = \frac{\sum true\ positives}{\sum true\ positives + \sum false\ positives} \tag{1}$$

$$Recall = \frac{\sum true\ positives}{\sum true\ positives + \sum false\ negatives} \tag{2}$$

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall + Presicion} \tag{3}$$

2.5. Implementation details

The Google Colab Notebook was utilized with a CPU virtual machine comprising 12.7 GB RAM and 107.7 GB of disk space, which is accessible at no cost. The experiments were conducted using Python (3.10.12), Keras (2.12.0), and Tensorflow (2.12.0). The complete scheme of the methodology used is depicted in Fig. 4.

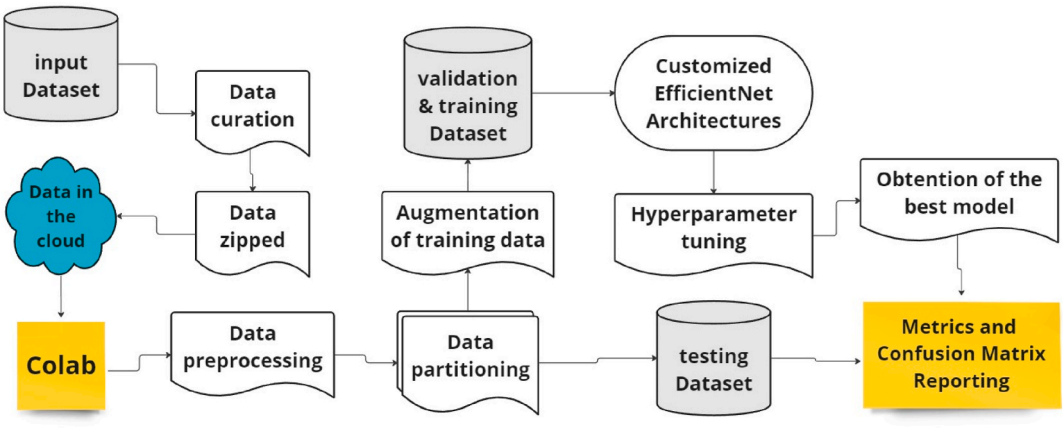


Fig. 4. Flowchart of the methodology approach used for this work.

3. Results

3.1. Data preprocessing

Images provided for ASTs of petrochemical industries are typically imbued with a surplus of information that is superfluous to the detection of corrosion (see Fig. 5a). In such instances, the images must be cropped to facilitate the training of CNN models [19]. In addition, as illustrated in Fig. 5b and c, some images were captured from a distance or encompass a range of corrosion types, respectively. These images were not utilized for the training of our model, as this can result in suboptimal performance when the data set is limited in size [16,26]. Consequently, one limitation of our model is that the input images must undergo the same preprocessing treatment, namely cropping, as illustrated in Fig. 1, in order to achieve optimal model performance. Conversely, the images fed to the models had dimensions comparable to those necessary for transfer learning, thus representing a relatively minor issue.

3.2. Data augmentation

This technique facilitated the acquisition of a more comprehensive dataset for training and validation. The data was augmented based on minor modifications to the original images to prevent the models from learning incorrect features from the training data (see Fig. 6). For example, the rotation of images generates a new space in the original frame that must be filled by the nearest pixels, constant pixels, reflected pixels, and so on. In addressing this specific issue, we opted to utilize the nearest pixel fill mode, as offered by Keras, given that the alternative options resulted in the creation of black spaces that were easily confused with black holes resulting from pitting corrosion.

3.3. Models proposed based on EfficientNet architectures

A decision was made to utilize the EfficientNetB0 and EfficientNetV2B0–B1 models for transfer learning. The aforementioned architectures belong to the novel EfficientNetV2 family of models, which are capable of training at a significantly accelerated pace in comparison to state-of-the-art models. Additionally, they are up to 6.8× smaller in size, and they employ a progressive learning approach that adjusts regularization along with image size. This is following the findings presented in Ref. [27]. These selections were based on the relatively small image dataset gathered by the petrochemical company, which reduced the possibility of overfitting when using large-scale architectures. For example, it has been demonstrated that models such as VGG16 and AlexNet, which have approximately 138 and 60 million parameters, respectively, exhibit suboptimal performance when applied to small data sets [28]. Notwithstanding, architectures such as GoogleNet, which possess a comparable number of parameters to EfficientNetB0, have been observed to exhibit fewer reported capabilities in small datasets [24]. The selection of EfficientNet architectures is also based on the preference for balanced models -small size and good performance for industrial implementation in devices such as mobile phones and tablets currently used for drone-aimed inspections. Moreover, to the best of our knowledge, these EfficientNet models have not been tested for corrosion classification tasks in the oil and gas industry. It is also noteworthy that we initially attempted to utilize MobileNetV2, a smaller model, but it yielded unsatisfactory results and was subsequently excluded from further analysis.

The experimental approach initially involved defining the optimal model architecture and hyperparameters through transfer learning (TL), followed by fine-tuning (FT). TL is a technique that has been shown to be effective in improving the performance of CNNs in image classification tasks [29]. However, FT has been demonstrated to be an effective approach for enhancing the top-1 performance of TL in practical problems related to artificial vision [30]. The specific details of the architecture are presented in Tables 2 and 3.

The number of intermediate layers to be introduced in our customized CNNs was determined through a series of experimental trials, as illustrated in Table 4. At the top of the frozen EfficientNet layers utilized for transfer learning (features extractor), we integrated a global_average_pooling2D. This reduced the computational complexity of subsequent layers. A sequence of inner dense layers (IL) was investigated to identify the optimal number of dense layers with a ReLU activation function for transfer learning. L1 (Lasso) and L2

Fig. 5. Illustration of some of the images received by the company. a) highlighted areas in images, b) distant images of the area of interest, and c) images with several corrosion types.

Fig. 6. Example of the images obtained after data augmentation.

Table 2
CNN's proposed models with 1 intermediate dense layer for TL.

Layers	Output shape	No. of parameters
EfficientNet (B0–V2B0–V2B1)	7 × 7 × 1280	0
Global_average_pooling2D	1280	0
Dense	50	64050
Batch_normalization	50	200
dropout	50	0
Dense_1	5	255
Total params: B0 = 4114076; V2B0 = 5983817; V2B1 = 6995629		
Trainable params: B0 = V2B0 = V2B1 = 66405		
Non-trainable params: B0 = 4049671; V2B0 = 5919412; V2B1 = 693.224		

Table 3
CNN's proposed models with 2 intermediate dense layers for FT.

Layers	Output shape	No. of parameters
EfficientNet (B0–V2B0–V2B1)	7 × 7 × 1280	0
Global_average_pooling2D	1280	0
Dense	100	128100
Batch_normalization	100	400
dropout	100	0
Dense_1	50	5050
Batch_normalization_1	50	200
Dropout_1	50	0
Dense_2	5	255
Total params: 4183576		
Trainable params: 133705		
Non-trainable params: 4049871		

Table 4
Test performed to explore the best CNN's model.

	Batch	Lr	α	γ	ESP	RP	RF	F1 Score	Val loss	IL	Mod	Test /Val
1	32	0.01	–	–	5	3	0.25	0.8320	1.1238	1	B0TL	15%T /(15%V)
2	64	0.01	–	–	5	3	0.25	0.7959	1.9799	1	B0TL	15%T /(15%V)
3	32	0.001	–	–	5	3	0.25	0.7984	1.2244	1	B0TL	15%T /(15%V)
4	64	0.001	–	–	5	3	0.25	0.7345	1.3761	1	B0TL	15%T /(15%V)
5	32	0.0001	–	–	5	3	0.25	0.7938	2.3206	1	B0TL	15%T /(15%V)
6	64	0.0001	–	–	5	3	0.25	0.7553	3.1431	1	B0TL	15%T /(15%V)
7	16	0.01	–	–	5	3	0.25	0.8234	0.8297	1	B0TL	15%T /(15%V)
8	16	0.01	–	–	10	5	0.25	0.7822	0.8272	1	B0TL	15%T /(15%V)
9	32	0.01	–	–	10	5	0.25	0.8135	1.1726	1	B0TL	15%T /(15%V)
10	32	0.01	–	–	10	5	0.25	0.8327	1.7406	2	B0TL	15%T /(15%V)
11	32	0.01	–	–	10	5	0.25	0.8132	2.7606	2	B0TL	15%T /(15%V)
12	32	0.01	–	–	10	5	0.25	0.8115	1.0490	1	V2B0 TL	15%T /(15%V)
13	32	0.01	–	–	10	5	0.25	0.8141	1.0323	1	B1TL	15%T /(15%V)
14	32	0.01	–	–	10	5	0.25	0.8153	0.7639	1	B0TL	10%T /(30%V)
15	32	0.01	–	–	10	5	0.50	0.8240	0.7204	1	B0TL	10%T /(30%V)
16	32	0.01	–	–	10	5	0.50	0.8152	0.7367	1	B0TL	10%T /(30%V)
17	32	0.01	0.25	2.0	10	5	0.50	0.8180	0.0702	1	B0TL	10%T /(30%V)
18	32	0.01	0.25	2.0	10	5	0.50	0.8057	0.0744	1	B0TL	10%T /(30%V)
19	32	0.01	0.25	2.0	10	5	0.25	0.8073	0.073	1	B0TL	10%T /(30%V)
20	32	0.01	0.5	2.0	10	5	0.50	0.7994	0.1533	1	B0TL	10%T /(30%V)
21	32	0.01	0.9	2.0	10	5	0.50	0.8060	0.2935	1	B0TL	10%T /(30%V)
22	32	0.01	0.9	1.5	10	5	0.50	0.7872	0.3724	1	B0TL	10%T /(30%V)
23	32	0.01	1.0	1.5	10	5	0.50	0.8039	0.4089	1	B0TL	10%T /(30%V)
24	32	0.01	1.0	1.0	10	5	0.50	0.7811	0.4971	1	B0TL	10%T /(30%V)
25	32	0.01	1.0	2.0	10	5	0.50	0.8087	0.3189	1	B0TL	10%T /(30%V)
26	32	0.01	1.0	3.0	10	5	0.50	0.8040	0.2364	1	B0TL	10%T /(30%V)
27	32	0.01	1.5	1.5	10	5	0.50	0.8029	0.6536	1	B0TL	10%T /(30%V)
28	32	0.01	1.5	2.0	10	5	0.50	0.7916	0.5371	1	B0TL	10%T /(30%V)
29	32	0.01	1.5	4.0	10	5	0.50	0.7987	0.2630	1	B0TL	10%T /(30%V)
30	32	0.01	2.0	2.0	10	5	0.50	0.8036	0.7107	1	B0TL	10%T /(30%V)
31	32	0.01	0.25	2.0	10	5	0.50	0.8886	0.0468	1	B0FT	10%T /(30%V)
32	32	0.01	0.25	2.0	10	5	0.50	0.8696	0.0469	1	B0FT	10%T /(30%V)
33	32	0.01	–	–	10	5	0.50	0.9021	0.4130	1	B0FT	10%T /(30%V)

(*continued on next page*)

Table 4 (*continued*)

	Batch	Lr	α	γ	ESP	RP	RF	F1 Score	Val loss	IL	Mod	Test /Val
34	32	0.01	–	–	10	5	0.50	0.8976	0.4302	1	B0FT	10%T /(30%V)
35	32	0.01	–	–	10	5	0.50	0.8519	0.7276	2	B0FT	10%T /(30%V)
36	32	0.01	–	–	10	5	0.50	0.8842	0.6677	2	B0FT	10%T /(30%V)
37	32	0.01	0.25	2.0	10	5	0.50	0.9082	0.0583	2	B0FT	10%T /(30%V)
38	32	0.01	0.25	2.0	10	5	0.50	0.8798	0.0543	2	B0FT	10%T /(30%V)

(Ridge) regularizations —Elastic Net— were added to this layer (fixed at 0.001), followed by a batch normalization layer, and a 20 % dropout layer. For both, TL and FT, the same procedure was followed. In the case of FT, the dropout layer was increased to 50 % and the intermediate layers were duplicated to reduce overfitting (see Table 3). In both cases, the output dense layer consisted of five neurons and a softmax activation function, thereby completing the artificial vision system for the classification of corrosion in ASTs.

3.4. Mathematical expression behind the proposed architecture

The most straightforward approach to understanding CNNs for a general audience is to conduct an experiment using the website https://poloclub.github.io/cnn-explainer. This interactive page elucidates the concepts of convolutions, strides, kernels, activation functions, pooling layers, input layers, and flatten layers [31]. For a more comprehensive understanding of feature extraction by CNNs, we recommend visiting the website https://microscope.openai.com/models/alexnet/, [32]. Also, revise Appendix A to observe how filters from CNNs aim for feature extraction. However, for the specific functions employed for regularization, including L1, L2, batch normalization, and dropout, their mathematical expressions are as follows. Equation (4) is associated with the categorical cross-entropy loss function, which is a widely used approach for multiclass classification problems. But, we also explored the categorical focal cross-entropy loss function to deal with hard to classy classes and imbalance issues in the dataset (see Equation (5), where $p_t = output\ if\ y_{true_i} == 1, else\ 1 - output$) [33]. Equations (6) and (7) are linked with L1 (Lasso) and L2 (Ridge) normalizations, which serve to penalize the complexity of the models. The objective of the batch normalization regulator is to address the issue of internal covariate shift. In Equation (8), μ_B is the batch mean, σ_B^2 the batch variance, and γ and β are learned during training. Equation (9) is the dropout, which selects neurons to be disabled based on the dropout probability p, rescaling the remained neurons' weights (w_i) to $1/(1\text{-}p)$.

$$CategoricalCE = -\sum_{i=1}^{N} y_{true_i} . \log\left(y_{pred_i}\right) \tag{4}$$

$$CategoricalFCE = = \alpha * (1 - p_t)^{\gamma} * CategoricalCE\left(y_{true_i}, y_{pred_i}\right) \tag{5}$$

$$loss\ fuction + \alpha . \frac{1}{N} \sum_{j=1}^{N} |w_i| \tag{6}$$

$$loss\ fuction + \alpha . \frac{1}{2N} \sum_{j=1}^{N} w_i^2 \tag{7}$$

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \in}} + \beta \tag{8}$$

$$w_i^* = \begin{cases} 0, with\ p \\ \dfrac{w_i}{1-p}, otherwise \end{cases} \tag{9}$$

3.5. Hyperparameter tuning

Concerning the hyperparameters that were not subjected to investigation within the model, the Adam optimizer and 150 epochs were employed. The batch size was investigated for 16, 32, and 64 values. The learning rate (Lr) was evaluated for three distinct magnitudes: 0.01, 0.001, and 0.0001. In the case of the categorical focal cross-entropy loss function, the values of gamma and alpha were tested in the ranges (1–4) and (0.25–2), respectively.

Furthermore, we employed the EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint callbacks from the Keras library to

mitigate the risk of overfitting. EarlyStopping with patience (ESP) was employed for training with a patience value of 5, while a patience value of 10 was utilized for the optimal models. For the ReduceLROnPlateau callback, patience (RP) of 3 and a factor (RF) of 0.25 were designated for training, while RP = 5 and RF = 0.50 were applied to the best models. The final hyperparameter adjustment was designed to reduce the learning rates of the model in incremental steps. Finally, the hyperparameter optimization was conducted manually to reduce the computational cost and time required when covering large ranges of values through Keras Tunner. Table 4 presents the experimental trials, which can also be found in the supplementary material. Two EfficientNetB0-TL models (15/16_replica and 17/18_replica) were selected as the most optimal models. One approach is based on categorical cross-entropy loss, while the other is based on categorical focal cross-entropy. Both models were evaluated with FT, indicating superior results when two intermediate dense layers were incorporated into the architectures, as this approach effectively mitigated the risk of overfitting (see section 3.6). Moreover, the data partition of 10 % for testing and 30 % of the remaining for validation was identified as the optimal distribution for reducing the loss values.

3.6. Training and validation results

The models that demonstrated the most efficacious performance during training attained an F1 score of 82 %, a figure that surpassed the performance of the remaining models by a margin of 2 % (see Table 4). This behavior can be attributed to the EfficientNet architecture utilized for transfer learning and the quality of the data. It is noteworthy that more advanced models, such as EfficientNetV2B0 and V2B1, did not demonstrate superior performance to that of the B0 model. This indicates that the pre-trained features learned by the simplest model align well with the classification problem. The optimal B0TL models exhibited the following hyperparameters: one intermediate dense layer with 50 units, a learning rate of 0.01, a batch size of 32, ESP = 10, RP = 5, RF = 0.5 (see runs 15–18 in Table 4). The distinguishing factor between the two models was the loss function. As indicated in Table 4, the categorical focal cross entropy function with alpha = 0.25 and gamma = 2 resulted in lower losses. Nevertheless, this does not necessarily imply an enhanced capacity for model generalization. When an alpha value of 0.25 was employed, the training focused on the minority class of the dataset, in this case, the blistering class. However, less attention was paid to the most abundant class, pitting [34]. However, if the performance of the model is constrained by the majority class, this loss function, with these hyperparameters, does not represent the optimal solution for the desired task. Consequently, the efficacy of the focal loss in achieving zero may be a mere illusion. Fig. 7 illustrates the limitations of this model for classifying corrosion types, despite its final loss of 0.0702. In contrast, the model that focused on the categorical cross-entropy function demonstrated comparable outcomes (see Fig. 8), with a final loss value of 0.7204. Consequently, the final loss value of the aforementioned model, which is approximately ten times higher than that of the model based on focal loss, is more aligned with the model's performance when tested with previously unseen images.

On the other hand, blistering and pitting corrosion were identified as the hardest classes for the classification problem addressed. As seen in Fig. 8, the blisters formed below a coating can be easily confused with the pits caused by microorganisms at the bottom of the

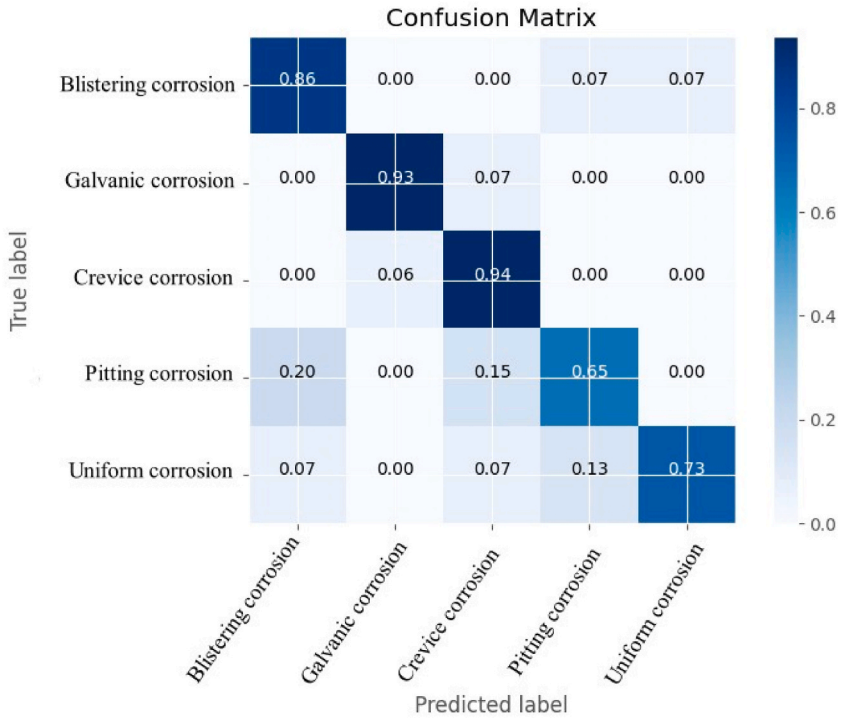


Fig. 7. Confusion matrix for the TL model trained using a focal function with the lower loss value.

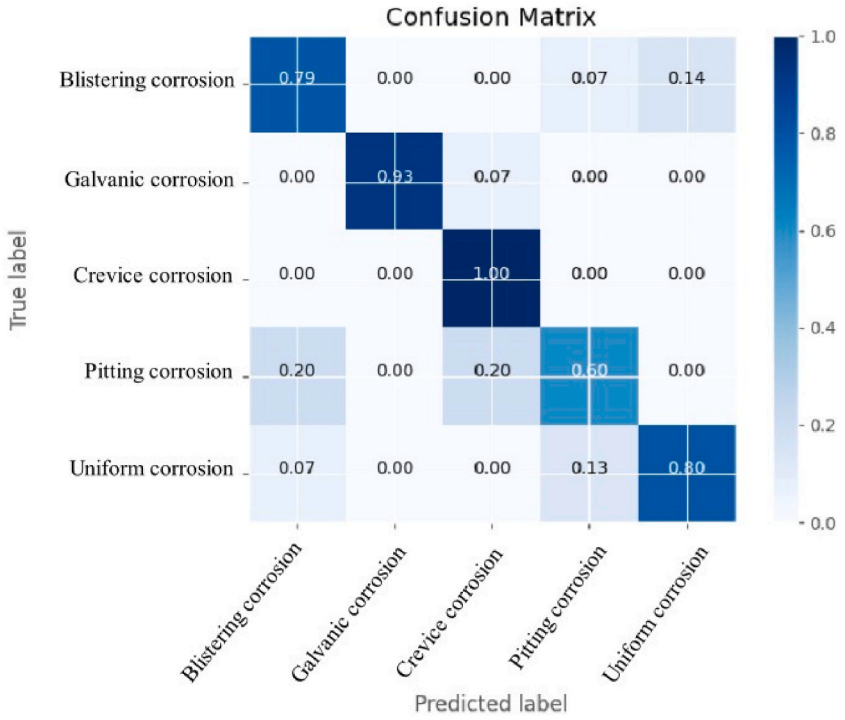


Fig. 8. Confusion matrix for the TL model trained categorical cross-entropy function with the lower loss value.

tank. Figs. 1a and 6 show how hard it can be, even for a human eye, to distinguish between these two classes. This limitation can also be observed in the confusion matrix of other runs in supplementary material. Hence, it is not only a problem of the loss function used but also of data scarcity to train the models, since microbial corrosion is a not frequent type of pitting in the studied ASTs −25 % of the pitting dataset.

In light of the aforementioned limitations, we elected to select the model with the highest final loss value, namely the one that employed the categorical cross-entropy function. Fig. 9 depicts the corresponding loss and F1-score curves for the B0TL model during training and validation. The optimal model performance was identified at epoch 51, as subsequent epochs exhibited signs of overfitting. Upon reaching convergence, the final learning rate of the B0TL model was found to be 6.25e-04, which was achieved through a series of four decreasing steps from 1e-2 (see Supplemental Material S39). The uniform and galvanic corrosion categories deliver the best performances while blistering and pitting the worst (see Fig. 10) in coherence with previous statements. These trends are also

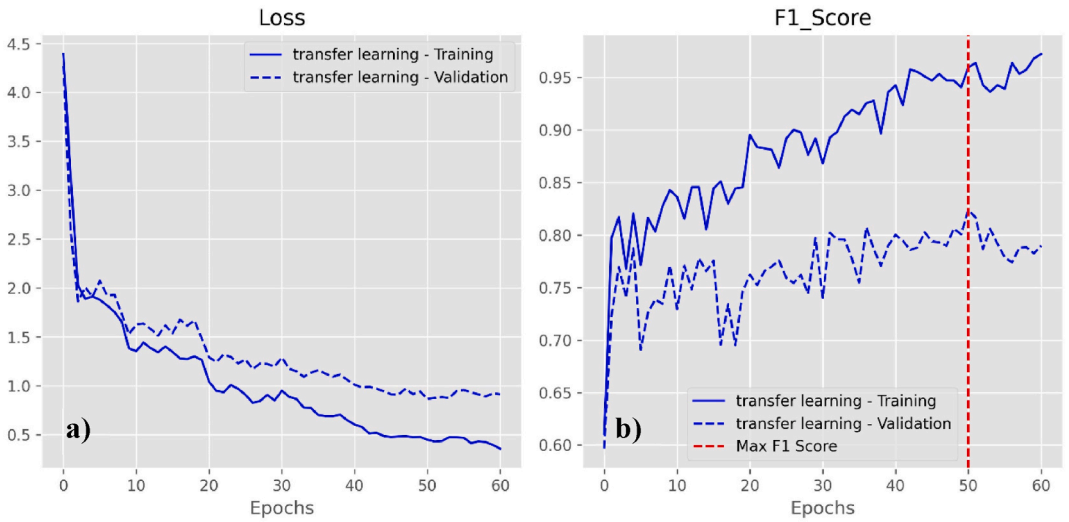


Fig. 9. B0TL a) loss and b) F1-score curves during training and validation.

observed in the class-wise Recall and Precision plots (see Figures B1 and B2). As much as Fig. 9, these class-wise metrics increased over epochs, as proof of the incremental learning capacity of our proposed model architecture for corrosion classification in ASTs.

Following the introduction of the TL model, the FT was employed to investigate potential enhancements in performance. This approach comprises two distinct phases: a preliminary phase, designated as "warming up," and a subsequent phase, designated as "adjustment." In the initial phase, all layers were trained with a learning rate of 1e-2. As the objective of this study differs from that of ImageNet, it was deemed appropriate to learn features at a relatively rapid pace for over 10 epochs, without freezing any layers. In the subsequent phase, the lower layers (feature extraction layers) were frozen, while the upper layers (task-specific layers) were adjusted to the data using a learning rate of 1e-3. The same hyperparameters employed to train the B0TL model (see runs 31–38 in Table 4) were used. Two intermediate dense layers with 100 and 50 units, respectively, were found to be optimal in order to avoid overfitting.

Fig. 11 depicts the corresponding loss and F1-score curves during the training and validation of the B0FT model. The optimal performance of the model was observed to occur around epoch 57. Similar to the transfer learning analysis, blistering and pitting classes release the worst performances (see Fig. 13). While, the class-wise learning process was as well incremental, as proved in Fig. 12, B3, and B4. In certain instances, crevice corrosion is misclassified due to its resemblance to galvanic corrosion when dealing with bolted joints or pitting when coating fails around the junctions (see Supplemental Material S42). Nevertheless, the final loss value attained for this model was somewhat lower than that observed after training the B0TL model (see supplemental material S36 and S41). As previously argued for the B0TL model, the performance of the B0FT model indicates that additional data is necessary to overcome the limitations of high similarity between some of the classes. Upon reaching convergence, the final learning rate for the B0TL model was 2.5e-04, representing a decrease of 1e-3 from the initial learning rate (see supplemental material S40).

3.7. Model testing and inference

The testing results were obtained by employing the B0TL and B0FT models, with categorical cross-entropy loss functions, due to their exemplary performances in the training and validation stages. As illustrated in Table 5, the B0TL model achieved a weighted mean F1 score of 85 %, representing a 10 % improvement over the TL model. These values exceed those observed in the validation phase, which may be attributed to the particular distribution of the validation dataset, which favored the learning of important patterns. Furthermore, the pitting and blistering corrosion classes exhibited the lowest performance, as anticipated based on the training process.

The B0FT model was employed to facilitate inferences and observations regarding its functionality. Fig. 14 illustrates how pitting induced by microbial corrosion is misinterpreted as blisters when the corrosion is less uniform, due to the homogeneous background. Furthermore, Fig. 15 indicates an accurate classification of blistering with a minor proportion of pitting, which may be present in the image. Similarly, an image displaying multiple types of corrosion was analyzed (see Fig. 16). This analysis demonstrated that the model's responses are contingent upon the predominant type of corrosion. In this instance, it is proposed that crevice corrosion should be selected as the predominant corrosion type, with galvanic corrosion and pitting also present. Blistering and pitting corrosions represent a bottle next to our developed CNNs architecture; which, may be solved by getting more diverse data for training and/or using ensemble model strategies. Even more, these results serve as a reminder that it is imperative to recognize that artificial vision should be employed in conjunction with an expert eye to corroborate and augment the inferences derived from the models.

4. Discussion and comparison with the state-of-the-art

While numerous authors have addressed the use of deep learning algorithms for corrosion detection [16,22,35–39], relatively few scientists have focused on the classification of corrosion types in different environments [14,17,23,40,41]. Moreover, no prior work has been conducted to classify corrosion types for ASTs using EfficientNet architectures. However, we compared our results with those of available models trained through CNNs on metallic infrastructure (see Table 6).

Petrica and colleagues gathered 1300 images of 256 × 256 pixels for the "rust" class and 2200 for the "non-rust" class. They allocated 80 % of the images for training and 20 % for validation, aiming to solve a binary classification problem. The model *bvlc_reference_caffenet*, developed by the Berkeley Vision and Learning Center, was utilized for fine-tuning, resulting in an 88 % performance for a learning rate of 5e-5 [41]. In contrast, Ejimuda and colleagues employed convolutional neural networks to address a multiclassification problem with object detection, categorizing corrosion types including intergranular, galvanic, and pitting corrosion. In order to address the limited number of images available, data augmentation was employed, resulting in a total of 336 images. Seventy percent of the data was utilized for training, while the remaining 30 % was employed for validation. This approach yielded an 83.3 % performance rating [17]. The primary limitation of both models in comparison to the present research was their robustness and the scarcity of the data.

In a more detailed examination of the aforementioned work, Holm et al. successfully addressed a multiclass corrosion problem, categorizing it into three distinct types: non-corrosion, paint flaking, and two red corrosion types related to bridge constructions. A large image dataset was employed following data augmentation of 9300 images. The data partition was established at 80 %, 10 %, and 10 % for training, validation, and testing, respectively. Four pre-trained models were employed. The models employed were AlexNet, GoogLeNet, ResNet-50, and VGG-16. The final model achieved the highest F1 score, with a value of 95.53 % [40]. The most analogous research was conducted by Ahuja et al., who solved a corrosion detection and classification problem using Faster R–CNN for six metallic corrosion categories. Their approach differed from ours in that they did not consider blistering corrosion. Following data augmentation, the authors collected approximately 2000 images of 512 × 512 pixels. Additionally, a batch size of 8, an input size of 224 × 224, 10 epochs, an Adamax optimizer, and a learning rate of 1e-2 were implemented. A VGG-16 network, pre-trained and

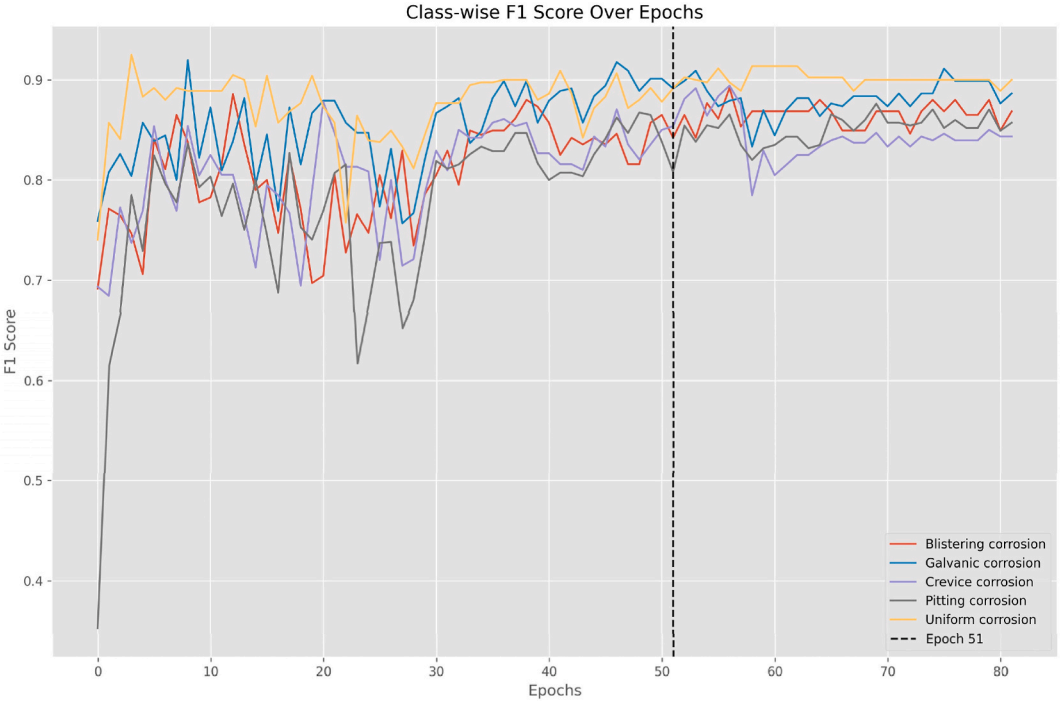


Fig. 10. Class-wise F1-score over epochs for the B0TL model.

Fig. 11. B0FT a) loss and b) F1-score curves during training and validation.

customized, was optimized to enhance classification accuracy, resulting in an F1 score of 93.8 % [23].

In contrast to our model, Holm's approach counted the double images to accomplish their goal [40]. While the report of Ahuja et al., lacks a deeper analysis and reliability since, despite employing half of our images and a model with 8 times more trainable parameters (VGG-16), they achieved a very high performance –overfitting alert– [23]. We addressed a more complex classification task, with two challenging classes -pitting and blistering corrosion-, and accomplished a 94 % performance in the best attempt, but using a clearer and more robust methodology than the work of Ahuja and coworkers. Then, EfficientNetB0 proved to be an excellent choice for corrosion classification tasks, allowing TL and FT approaches due to its small set of trainable parameters and lower computational cost.

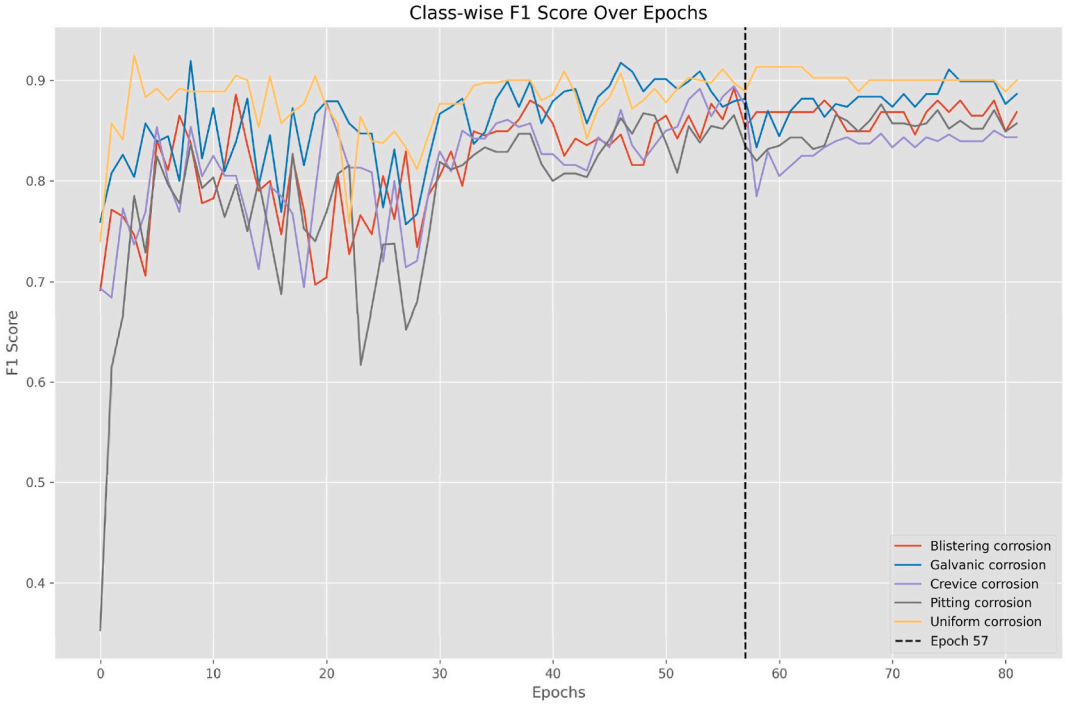


Fig. 12. Class-wise F1-score over epochs for the B0FT model.

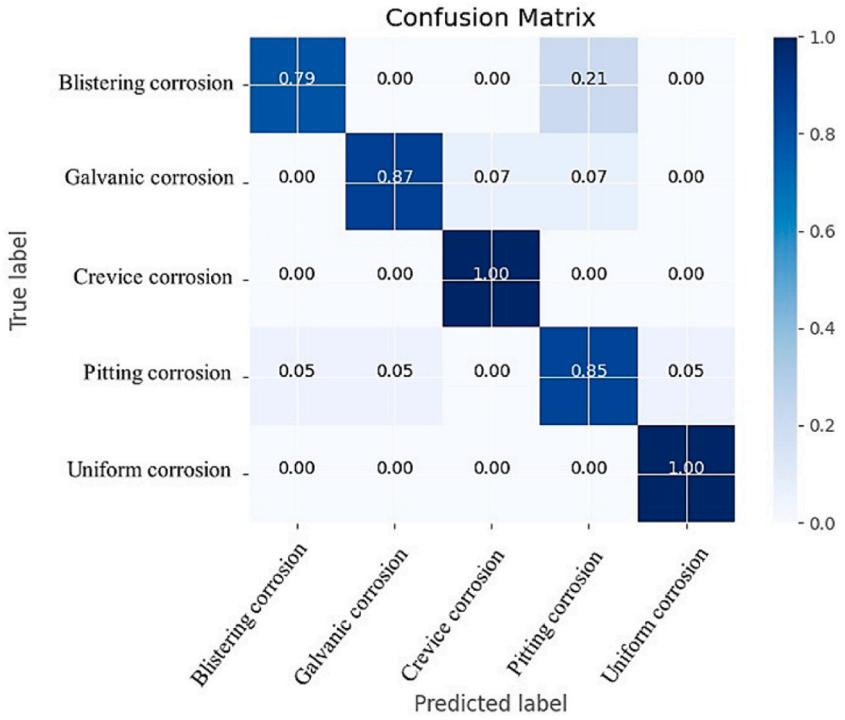


Fig. 13. Confusion matrix for the FT model trained with categorical cross-entropy function with the lower loss value.

Table 5
Average results of the testing step of the TL and FT models.

	TL Precision	TL Recall	TL F1Score	FT Precision	FT Recall	FT F1Score	Img.
Blistering Corrosion	0.78 ± 0.07	0.75 ± 0.04	0.76 ± 0.05	0.93 ± 0.07	0.86 ± 0.00	0.89 ± 0.03	14
Galvanic corrosion	0.93 ± 0.01	0.84 ± 0.04	0.88 ± 0.02	0.94 ± 0.06	0.93 ± 0.00	0.94 ± 0.04	15
Crevice corrosion	0.81 ± 0.02	0.94 ± 0.00	0.87 ± 0.01	0.92 ± 0.03	0.97 ± 0.03	0.94 ± 0.00	16
Pitting corrosion	0.92 ± 0.03	0.78 ± 0.03	0.84 ± 0.01	0.89 ± 0.02	0.98 ± 0.03	0.93 ± 0.00	20
Uniform corrosion	0.80 ± 0.01	0.94 ± 0.07	0.86 ± 0.02	1.00 ± 0.00	0.87 ± 0.00	0.93 ± 0.00	15
Accuracy			0.85 ± 0.01			0.93 ± 0.02	80
Macro Avg.	0.85 ± 0.01	0.85 ± 0.01	0.84 ± 0.01	0.93 ± 0.02	0.92 ± 0.01	0.93 ± 0.02	80
Weighted Avg.	0.85 ± 0.01	0.85 ± 0.01	0.85 ± 0.01	0.94 ± 0.02	0.93 ± 0.02	0.93 ± 0.02	80

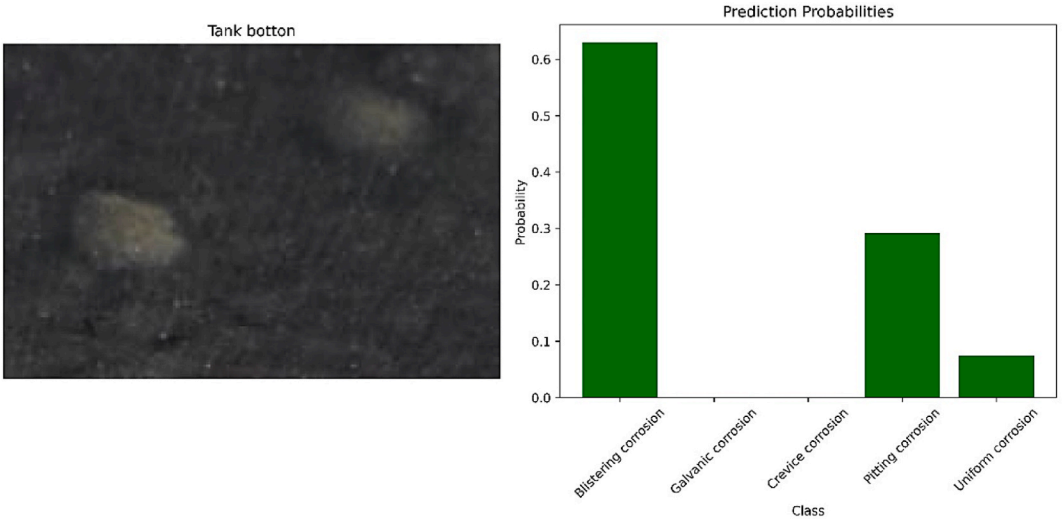


Fig. 14. Inferences of the B0FT model for an image with pitting corrosion.

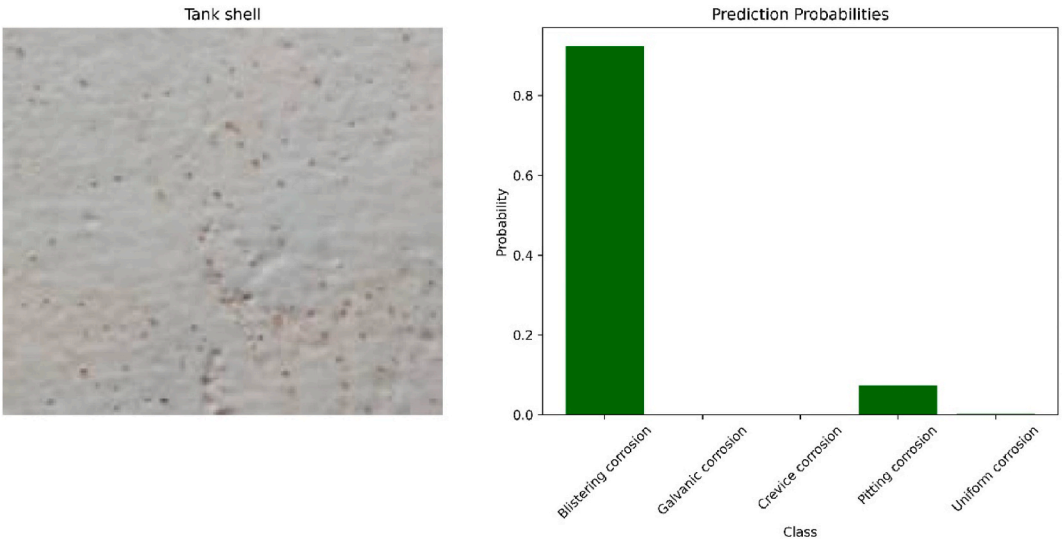


Fig. 15. Inferences of the B0FT model for an image with blistering corrosion.

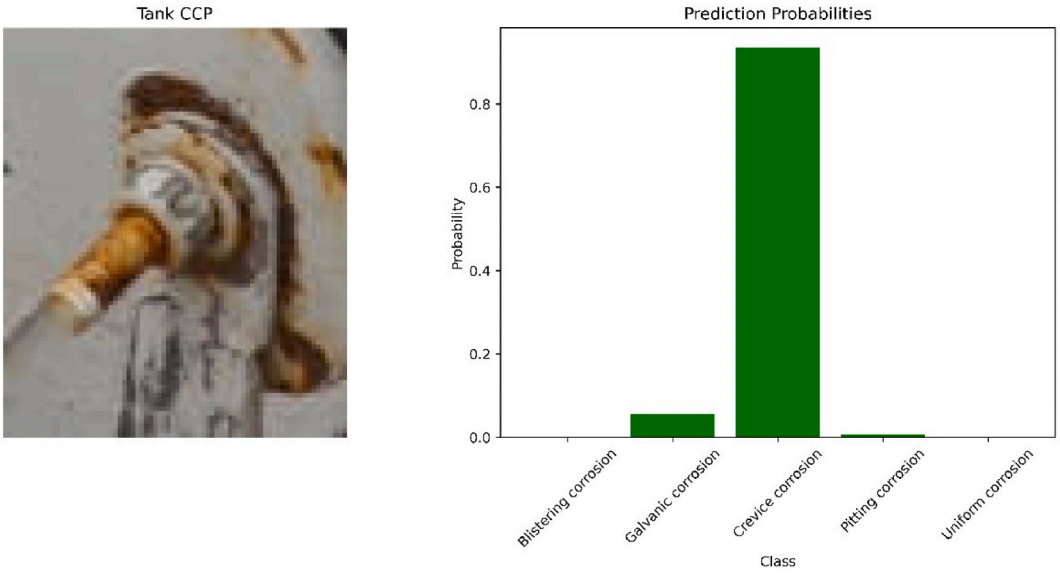


Fig. 16. Inferences of the B0FT model for an image with crevice corrosion in a cathodic current protection (CCP) connection system.

Table 6
Comparison with similar approaches.

	No. of Images	Model	Best F1 score
Holm et al. [40]	9300	VGG-16	95.5 %
Ahuja et al. [23]	2000	VGG-16	93.8 %
This work	4562	EfficientNetB0	94.0 %

5. Conclusions

This article presents an investigation into the performance of EfficientNet models B0, V2B0, and V2B1 for corrosion image classification in above-ground storage tanks. The images used were obtained from a petrochemical company and augmented to create a dataset of approximately 5000 images. The types of corrosion of interest identified in the company were galvanic, interstitial, uniform, pitting, and blistering corrosion. The model with the best performance was obtained by fine-tuning EfficientNetB0, which yielded an F1-score of 94 % in the best of cases. Data scarcity and the need for a more diverse database of corrosion issues, in particular for pitting and blistering classes, are the main limitations of the present work, while the main assumptions concern the preferential use of EfficientNet architectures to obtain the best performances and the use of weighted F1_score as the principal metric to solve the slight class imbalance of the training data. Blistering and pitting posed a barrier to the trained model due to the limited data used. These two types of corrosion were particularly challenging because they have similar shapes, colors, and textures when microbial pitting is present. The trained B0FT model proved its potential to be applied in automatic inspections of metallic infrastructures. This model can be extended to include other types of corrosion, taking advantage of its low computational cost and reduced number of trainable parameters compared to other models available on ImageNet. Future work should focus on ensembling this model with a lightweight object detection architecture to improve the robustness and performance of corrosion classification. This would lead to corrosion detection and classification for drone-based inspections. While focusing more on early detection of coating peeling, it is suggested to regularly monitor the metallic parts with an artificial vision model trained to detect coating imperfections that may lead to corrosion issues.

Data availability

The data employed is not available. The data that has been used is confidential.

Funding

No funding to declare.

CRediT authorship contribution statement

Anibal Alviz-Meza: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Leidy L. Hadechini-Meza:** Writing – review & editing, Visualization, Validation. **Darío Y. Peña-Ballesteros:** Validation, Supervision, Investigation, Formal analysis.

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.

Acknowledgments

We thank Dr. Valencia-Arias Alejandro from the Instituto Tecnológico Metropolitano (Medellin) for proofreading the manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.heliyon.2024.e34882.

Appendix A

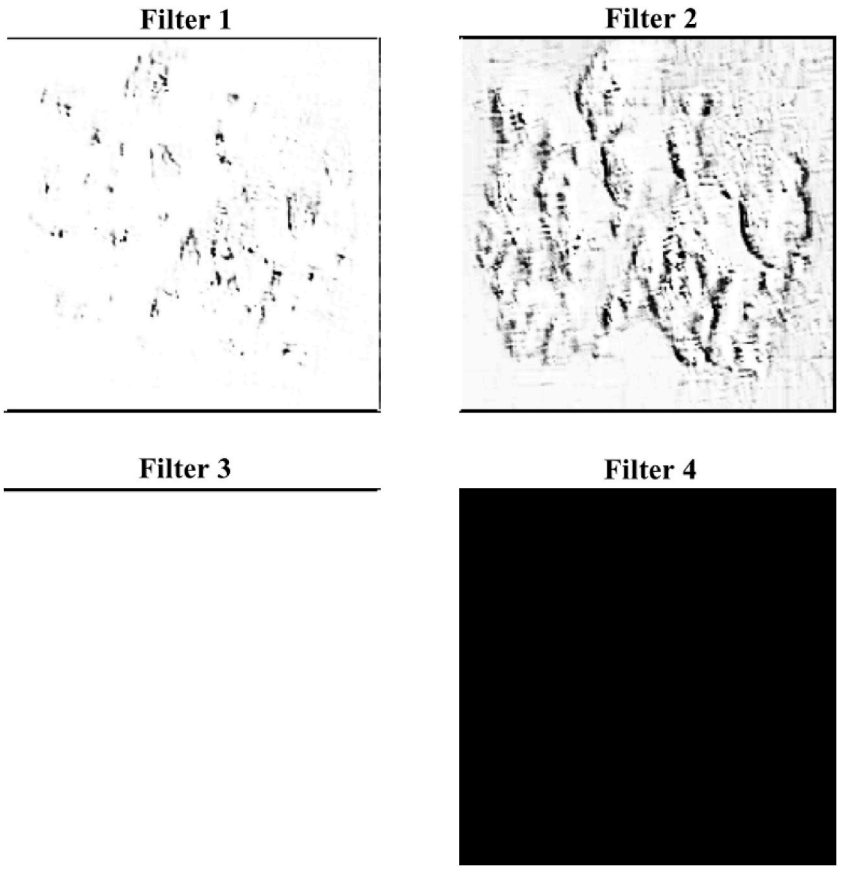


Fig. A1. Feature extraction performed by some CNN filters to a pitting corrosion image.

Appendix B

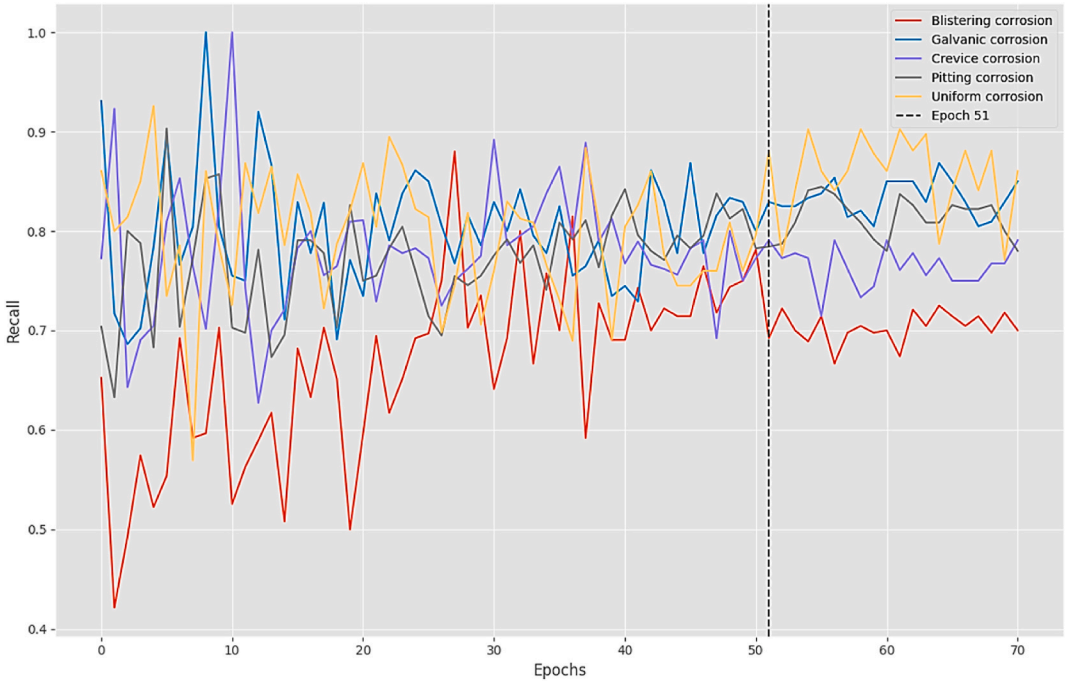


Fig. B1. Class-wise Recall over epochs for the B0TL model.

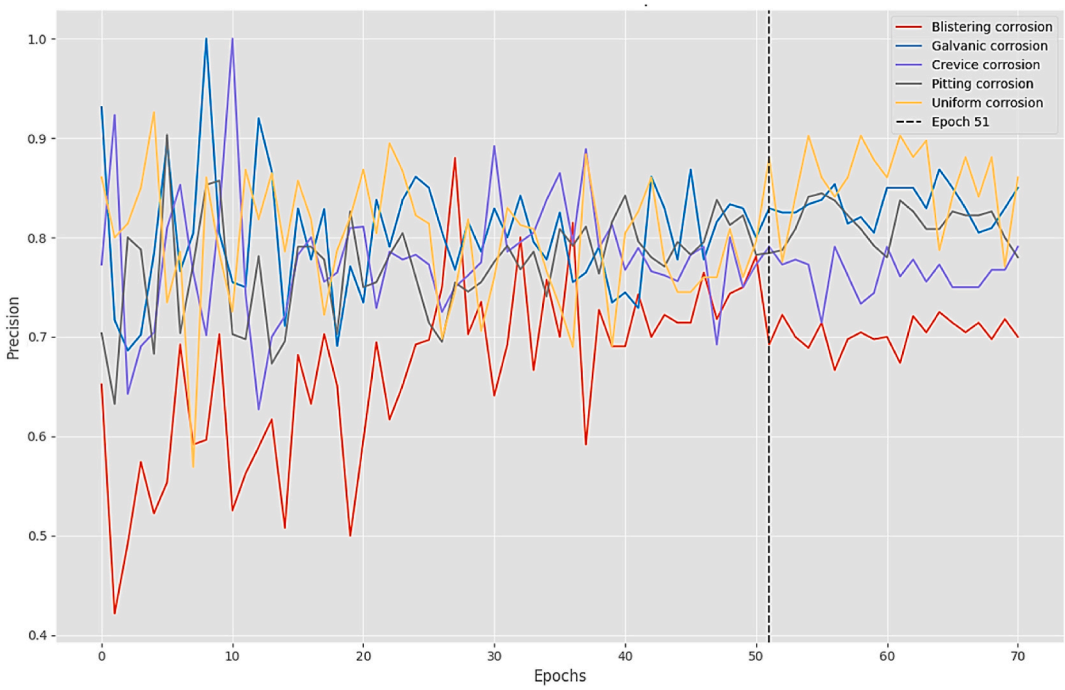


Fig. B2. Class-wise Precision over epochs for the B0TL model.

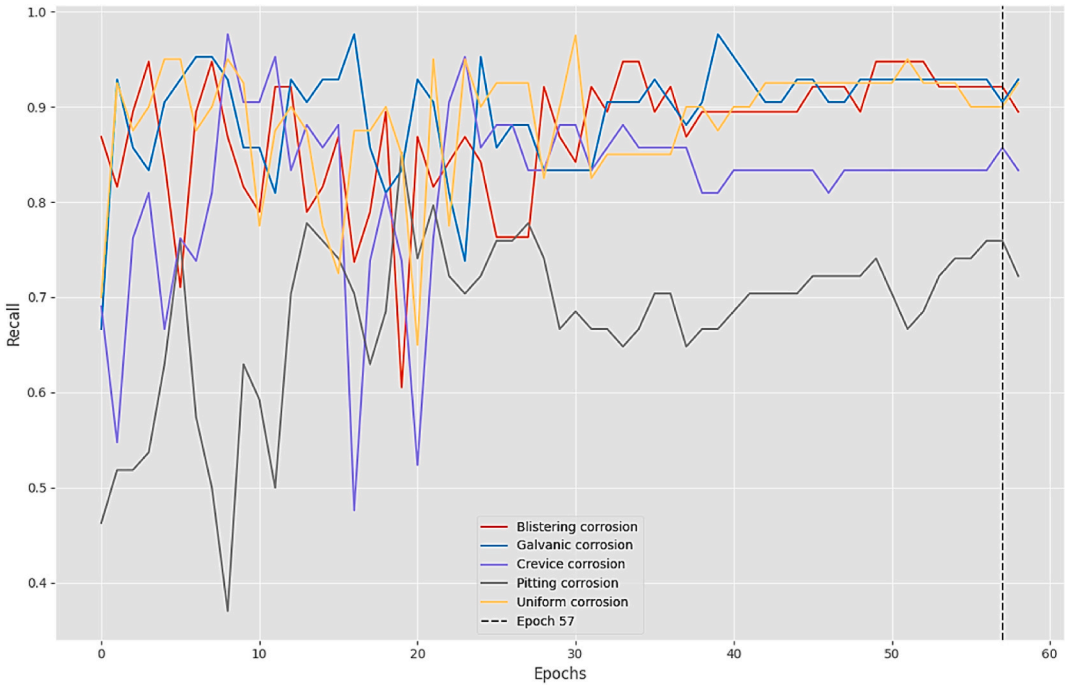


Fig. B3. Class-wise Recall over epochs for the B0FT model.

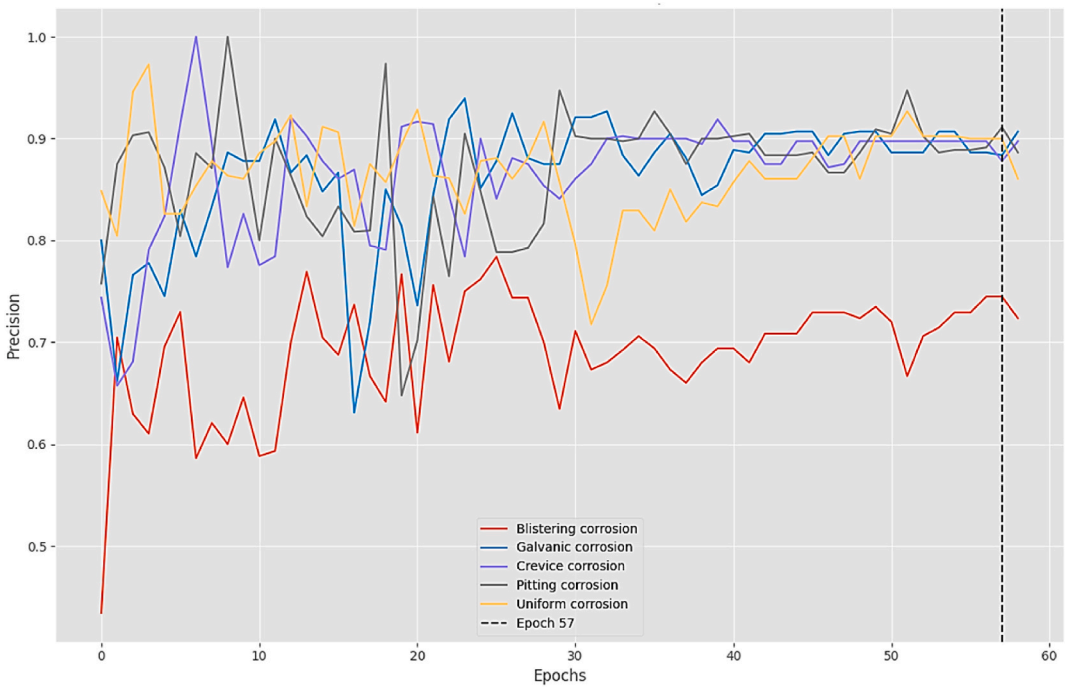


Fig. B4. Class-wise Precision over epochs for the B0FT model.

References

[1] I.V. Morozov, Y.M. Potanina, S.A. Voronin, N.V. Kuchkovskaya, M.D. Siliush, Prospects for the development of the oil and gas industry in the regional and global economy, Int. J. Energy Econ. Pol. 8 (4) (Jul. 2018) 55–62 [Online]. Available: https://www.econjournals.com/index.php/ijeep/article/view/6523. (Accessed 15 August 2023).

[2] A. Alviz-Meza, S.I. Shah, V. Kafarov, D. Peña-Ballesteros, Fireside corrosion of 9Cr-1Mo steel at high temperatures, in the acid flue gas from an oil refinery, Corros. Sci. 193 (October) (2021) 109878, https://doi.org/10.1016/j.corsci.2021.109878.

[3] A. Alviz, V. Kafarov, L. Meriño, Methodology for evaluation of corrosion damage during combustion process in refinery and petrochemical industry. Case study: AISI 304 and ASTM A335 P5 steels, Chem. Eng. Trans. 61 (2017) 1315–1320, https://doi.org/10.3303/CET1761217.

[4] A. Alviz-Meza, J. Sanabria-Cala, V. Kafarov, D.Y. Peña-Ballesteros, Study of continuous corrosion on ASTM A335 P91 steel in an environment of CO2-O2-N2-H2O derived from the theoretical combustion products of a mixture of refining gases at high temperatures, Chem. Eng. Trans. 70 (2018) 1069–1074, https://doi.org/10.3303/CET1870179.

[5] A. Alviz Meza, V. Kafarov, D.Y. Pena Ballesteros, Study of the continuous corrosion in an oxidation environment derived from the theoretical combustion products in a refinery. Case study: ferritic steel ASTM A335 P91, J. Phys. Conf. Ser. 935 (2017) 012057, https://doi.org/10.1088/1742-6596/935/1/012057.

[6] A.A. Soomro, A.A. Mokhtar, J.C. Kurnia, N. Lashari, H. Lu, C. Sambo, Integrity assessment of corroded oil and gas pipelines using machine learning: a systematic review, Eng. Fail. Anal. 131 (October 2021) (2022) 105810, https://doi.org/10.1016/j.engfailanal.2021.105810.

[7] M.F. Milazzo, G. Ancione, P. Bragatto, E. Proverbio, A probabilistic approach for the estimation of the residual useful lifetime of atmospheric storage tanks in oil industry, J. Loss Prev. Process. Ind. 77 (Jul. 2022) 104781, https://doi.org/10.1016/J.JLP.2022.104781.

[8] M. Wasim, M.B. Djukic, External corrosion of oil and gas pipelines: a review of failure mechanisms and predictive preventions, J. Nat. Gas Sci. Eng. 100 (April) (2022), https://doi.org/10.1016/j.jngse.2022.104467.

[9] H. Bi, et al., Evaluation of the acoustic emission monitoring method for stress corrosion cracking on aboveground storage tank floor steel, Int. J. Pres. Ves. Pip. 179 (Jan. 2020) 104035, https://doi.org/10.1016/J.IJPVP.2019.104035.

[10] R.D. Kane, J.G. Maldonado, L.J. Klein, Stress corrosion cracking in fuel ethanol: a newly recognized phenomenon, Mar, OnePetro 28 (2004) [Online]. Available:. (Accessed 15 August 2023).

[11] B. Akhlaghi, H. Mesghali, M. Ehteshami, J. Mohammadpour, F. Salehi, R. Abbassi, Predictive deep learning for pitting corrosion modeling in buried transmission pipelines, Process Saf. Environ. Protect. 174 (Jun. 2023) 320–327, https://doi.org/10.1016/J.PSEP.2023.04.010.

[12] E. Arzaghi, B.H. Chia, M.M. Abaei, R. Abbassi, V. Garaniya, Pitting corrosion modelling of X80 steel utilized in offshore petroleum pipelines, Process Saf. Environ. Protect. 141 (Sep. 2020) 135–139, https://doi.org/10.1016/J.PSEP.2020.05.024.

[13] P.H. Chen, H.K. Shen, C.Y. Lei, L.M. Chang, Support-vector-machine-based method for automated steel bridge rust assessment, Autom. ConStruct. 23 (May 2012) 9–19, https://doi.org/10.1016/J.AUTCON.2011.12.001.

[14] N. Idusuyi, O.J. Samuel, T.T. Olugasa, O.O. Ajide, R. Abu, Corrosion modelling using convolutional neural networks: a brief overview, J. Bio- Tribo-Corrosion 8 (3) (2022) 1–8, https://doi.org/10.1007/s40735-022-00671-3.

[15] Q. Wang, et al., Evolution of corrosion prediction models for oil and gas pipelines: from empirical-driven to data-driven, Eng. Fail. Anal. 146 (Apr. 2023) 107097, https://doi.org/10.1016/J.ENGFAILANAL.2023.107097.

[16] B.T. Bastian, J. N, S.K. Ranjith, C.V. Jiji, Visual inspection and characterization of external corrosion in pipelines using deep neural network, NDT E Int. 107 (February) (Oct. 2019) 102134, https://doi.org/10.1016/j.ndteint.2019.102134.

[17] C.C. Ejimuda, C.M. Ejimuda, Using deep learning and computer vision techniques to improve facility corrosion risk management systems, SPE West. Reg. Meet. Proc. 2018-April (Apr. 2018), https://doi.org/10.2118/190036-MS.

[18] S. Bhowmik, Digital twin for offshore pipeline corrosion monitoring: a deep learning approach, Proc. Annu. Offshore Technol. Conf. (Aug. 2021), https://doi.org/10.4043/31296-MS.

[19] D.J. Atha, M.R. Jahanshahi, Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection, Struct. Health Monit. 17 (5) (2018) 1110–1128, https://doi.org/10.1177/1475921717737051.

[20] S.K. Fondevik, A. Stahl, A.A. Transeth, O.O. Knudsen, Image segmentation of corrosion damages in industrial inspections, Proc. - Int. Conf. Tools with Artif. Intell. ICTAI 2020-Novem (2020) 787–792, https://doi.org/10.1109/ICTAI50040.2020.00125.

[21] B. Burton, W.T. Nash, N. Birbilis, RustSEG: Automated Segmentation of Corrosion Using Deep Learning, 2022, pp. 1–30.

[22] 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. Hum. Comput. 14 (7) (2023) 8087–8098, https://doi.org/10.1007/s12652-021-03580-4.

[23] S.K. Ahuja, M.K. Shukla, K.K. Ravulakollu, Neural network-based surface corrosion classification on metal articles, Lect. Notes Data Eng. Commun. Technol. 62 (2021) 115–124, https://doi.org/10.1007/978-981-33-4968-1_10/COVER.

[24] M. Tan, Q.V. Le, EfficientNet: rethinking model scaling for convolutional neural networks, 36th Int. Conf. Mach. Learn. ICML 2019 2019-June (May 2019) 10691–10700 [Online]. Available: https://arxiv.org/abs/1905.11946v5. (Accessed 16 August 2023).

[25] T. Mingxing, EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling, Google Research Blog, 2019. https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html. (Accessed 16 August 2023).

[26] M. Khayatazad, M. Honhon, W. De Waele, Detection of corrosion on steel structures using an artificial neural network, Struct. Infrastruct. Eng. 19 (12) (2023) 1860–1871, https://doi.org/10.1080/15732479.2022.2069272.

[27] M. Tan, Q.V. Le, EfficientNetV2: smaller models and faster training, Proc. Mach. Learn. Res. 139 (Apr. 2021) 10096–10106 [Online]. Available: https://arxiv.org/abs/2104.00298v3. (Accessed 16 August 2023).

[28] X. Zhang, The AlexNet, LeNet-5 and VGG NET applied to CIFAR-10, Proc. - 2021 2nd Int. Conf. Big Data Artif. Intell. Softw. Eng. ICBASE 2021 (2021) 414–419, https://doi.org/10.1109/ICBASE53849.2021.00083.

[29] M. Shaha, M. Pawar, Transfer learning for image classification, Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018 (Sep. 2018) 656–660, https://doi.org/10.1109/ICECA.2018.8474802.

[30] M.S. Tanveer, M.U.K. Khan, C.M. Kyung, Fine-tuning DARTS for image classification, Proc. - Int. Conf. Pattern Recognit. (2020) 4789–4796, https://doi.org/10.1109/ICPR48806.2021.9412221.

[31] Z.J. Wang, et al., CNN explainer: learning convolutional neural networks with interactive visualization, IEEE Trans. Vis. Comput. Graph. 27 (2) (Feb. 2021) 1396–1406, https://doi.org/10.1109/TVCG.2020.3030418.

[32] OpenAI, "Microscope,". https://microscope.openai.com/models, 2024. (Accessed 25 April 2024).

[33] H. Phan, K. Yamamoto, T.H. Phan, K. Yamamoto, Resolving class imbalance in object detection with weighted cross entropy losses [Online]. Available: https://arxiv.org/abs/2006.01413v1, Jun. 2020. (Accessed 21 August 2023).

[34] M. Kerkhof, L. Wu, G. Perin, S. Picek, Focus is key to success: a focal loss function for deep learning-based side-channel analysis, Lect. Notes Comput. Sci. 13211 (LNCS) (2022) 29–48, https://doi.org/10.1007/978-3-030-99766-3_2/TABLES/8.

[35] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, Comput. Civ. Infrastruct. Eng. 33 (9) (2018) 731–747, https://doi.org/10.1111/mice.12334.

[36] Q. Zhou, S. Ding, Y. Feng, G. Qing, J. Hu, Corrosion inspection and evaluation of crane metal structure based on UAV vision, Signal, Image Video Process 16 (6) (2022) 1701–1709, https://doi.org/10.1007/s11760-021-02126-7.
[37] 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 (April 2018) (2019) 101823, https://doi.org/10.1016/j.apor.2019.05.008.
[38] S. Zhang, et al., A channel attention based deep neural network for automatic metallic corrosion detection, J. Build. Eng. 42 (March) (2021) 103046, https://doi.org/10.1016/j.jobe.2021.103046.
[39] S.K. Ahuja, M.K. Shukla, K.K. Ravulakollu, Surface corrosion grade classification using convolution neural network, Int. J. Recent Technol. Eng. 8 (3) (2019) 7645–7649, https://doi.org/10.35940/ijrte.C6196.098319.
[40] E. Holm, A.A. Transeth, O.O.Ø. Knudsen, A. Stahl, Classification of corrosion and coating damages on bridge constructions from images using convolutional neural networks, Twelfth International Conference on Machine Vision (ICMV 2019) 11433 (11433) (Jan. 2020) 549, https://doi.org/10.1117/12.2557380.
[41] L. Petricca, T. Moss, G. Figueroa, S. Broen, Corrosion detection using A.I : 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, p. 99, https://doi.org/10.5121/csit.2016.60608.