Heliyon 10 (2024) e36754

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Corrosion image classification method based on EfficientNetV2

Ziheng Zhao, Elmi Bin Abu Bakar *, Norizham Bin Abdul Razak, Mohammad Nishat Akhtar **

School of Aerospace Engineering, Kampus Kejuruteraan, Universiti Sains Malaysia, 14300, Nibong Tebal, Pulau Pinang, Malaysia

ARTICLE INFO

Keywords:
Corrosion
CNN
EfficientNetV2
Fine-tuning
Image classification
Accuracy

ABSTRACT

Corrosion is one of the key factors leading to material failure, which can occur in facilities and equipment closely related to people's lives, causing structural damage and thus affecting the safety of people's lives and property. To identify corrosion more effectively across multiple facilities and equipment, this paper utilizes a corrosion binary classification dataset containing various materials to develop a CNN classification model for better detection and distinction of material corrosion, using a methodological paradigm of transfer learning and fine-tuning. The proposed model implementation initially uses data augmentation to enhance the dataset and employs different sizes of EfficientNetV2 for training, evaluated using Confusion Matrix, ROC curve, and the values of Precision, Recall, and F1-score. To further enhance the testing results, this paper focuses on the impact of using the Global Average Pooling layer versus the Global Max Pooling layer, as well as the number of fine-tuning layers. The results show that the Global Average Pooling layer performs better, and EfficientNetV2B0 with a fine-tuning rate of 20 %, and EfficientNetV2S with a fine-tuning rate of 15 %, achieve the highest testing accuracy of 0.9176, an ROC-AUC value of 0.97, and Precision, Recall, and F1-Score values exceeding 0.9. These findings can be served as a reference for other corrosion classification models which uses EfficientNetV2.

1. Introduction and background

Material corrosion has become one of the major problems of material failure, which refers to the deterioration of material qualities caused by interactions with their surroundings, corrosion is an unavoidable process that affects most metals and many other materials [1]. In this issue, either large facilities and equipment such as ships, oil and gas pipelines, aerospace, or small civilian items such as automobiles and household appliances, will face hazards caused by corrosion if they cannot be effectively detected. Corrosion has caused multiple problems in those aspects, In the case of ships, corrosion poses a significant safety risk and can result in the penetration of thickness, fatigue cracks, brittle fracture, and unstable failure [2]; for oil and pipelines, corrosion and high operating pressures can greatly diminish the lifespan and structural soundness of oil and gas pipelines, impacting their serviceability [3–5]. Considering the terrible effects caused by corrosion, it is necessary to design a high-efficiency method to determine which items have corrosion among a variety of equipment and products.

To detect corrosion, digital image processes have been widely used in this area where these non-destructive methods offer a cost-

* Corresponding author.
** Corresponding author.
E-mail addresses: zhaoziheng@student.usm.my (Z. Zhao), meelmi@usm.my (E.B.A. Bakar), nishat@usm.my (M.N. Akhtar).

https://doi.org/10.1016/j.heliyon.2024.e36754
Received 11 June 2024; Received in revised form 22 July 2024; Accepted 21 August 2024
Available online 24 August 2024

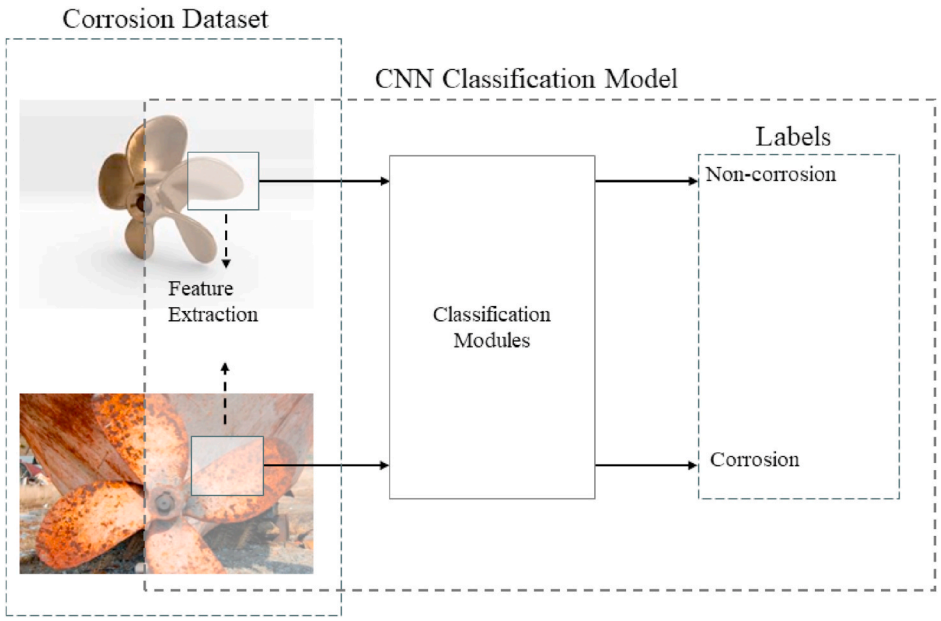

Fig. 1. A fundamental way of classifying corrosion images using CNNs.

effective, rapid, and reasonably precise outcome [6]. Nowadays, the utilization of convolutional neural networks (CNNs) led to an enhancement in the classification abilities of computers [7]. CNN, a major Deep Learning method used for image processing [8–10] including image classification, image semantic segmentation, object detection in images [11] has shown its ability in multiple fields [12–15]. Simultaneously, enhancing the capabilities of CNN [12,16] or integrating it with other technologies [14,15]may provide more substantial outcomes. Fırat, Hüseyin [12]. suggested a CNN method for classifying white blood cells. The CNN model is comprised of Inception module, Pyramid Pooling module, and Depthwise Squeeze-and-Excitation block. These modules can allow the model to simultaneously carry out several convolutions at various scales, collect data from different scales selectively, comprehend features that contain valuable information and discard irrelevant characteristics. To accurately localise and encode actions in facial expressions, Liu, Mengyi et al. [14] integrated a deformable parts learning component into the 3D CNN framework. This approach's evaluation of three datasets demonstrates its ability to reach the highest level of accuracy in recognising facial expressions. Khaki, Saeed et al. [15] introduced a novel approach for forecasting crop production by integrating CNN with Recurrent Neural Network (RNN). The CNN modules are employed to capture both the linear and nonlinear impacts of weather and soil data, while the RNN is utilized to capture the temporal patterns of crop yields over multiple years. The simulation results demonstrate that the suggested CNN-RNN model outperforms all other investigated approaches in terms of root mean square error. To enhance the accuracy of classification and reduce the computational complexity of CNNs, Qin, Jiaohua et al. [16] substituted the input of CNN from fixed-size images to suitable big-size images and replaced certain modules with an Inverted Residual Block module that has reduced computational cost and parameters. The simulation results demonstrate that this enhanced CNN exhibits superior performance in image categorization while simultaneously reducing network parameters and computational costs. In general, these methods exemplify the outstanding classification ability of CNNs and the variety that they can acquire through enhancements or amalgamation with other technologies.

Based on the advantages of CNNs, using them in the domain of corrosion detection is a favorable decision. Currently, there has been extensive research on applying CNNs for categorizing corrosion images. The fundamental CNN classification approach can be divided into three components. First, a dataset containing various categories of corrosion images is acquired. These categories can be divided into binary classification, where there are two options: corrosion and non-corrosion, or multivariate classification, where there are multiple levels of corrosion. Second, a CNN-based model is used to detect and categorize distinctive image characteristics, ultimately generating the matching category label for each image as the model's output. In this regard, Fig. 1 illustrates a fundamental method for classifying corrosion images. Multiple studies have demonstrated that CNN exhibits distinct advantages in processing corrosion images compared with other approaches [17–19] For instance, Malashin, Ivan et al. [17] used a custom-designed and optimized CNN for binary classification, utilizing a dataset consisting of 576,000 photos of pipes, both with and without corrosion. The results showed that the efficiency of the proposed CNN exceeded that of most current classification methods. To identify novel forms of corrosion in water pipeline-based images, Ramkumar, G [18]. employed CNN and support vector machine (SVM) techniques. These methods were applied to a dataset consisting of 40 low-resolution image samples to detect instances of corrosion. The results demonstrated that the CNN approach achieved an accuracy that was 15 % superior to that of SVM. Given that convolutional neural networks offer the benefit of not requiring prior knowledge or human effort in feature design, Atha, Deegan J [19]. presented various approaches for assessing metal surface corrosion using CNNs. The results demonstrated that CNNs outperformed the state-of-the-art vision-corrosion detection approach which was built utilizing texture and colour analysis with a simple multi-layer perceptron network. Thus, all the

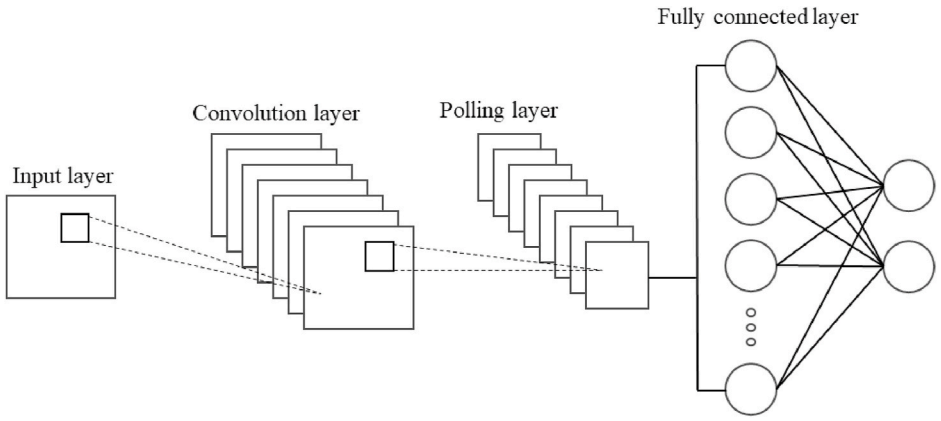

Fig. 2. Basic structure of a CNN model.

aforementioned methods demonstrated the validity of utilizing CNN in the domain of corrosion. It is evident that not only it can attain higher classification accuracy, but it can also surpass the performance of many other classification models.

Nevertheless, although numerous scholars have used CNNs on corrosion image classification and obtained good results, however there are certain issues and complexities which needs to be addressed to get enhanced performance. Bastian, Blossom Treesa et al. [20] utilized a dataset comprising 14,000 optical images of pipelines exhibiting various corrosion levels. They proposed a custom-designed CNN for classification, achieving a high accuracy of 98.8 %. However, this high accuracy is closely tied to the large dataset size. Idusuyi, Nosa et al. [21] used two image datasets produced by different devices, selecting the ResNet50 CNN model for classification, and obtained an accuracy above 80 % for both models. Nevertheless, they did not compare their results with other CNN models. Yao, Yuan et al. [22] introduced an innovative method for identifying and categorizing corrosion damage in hull structural plates using artificial intelligence and CNNs. However, their approach required 15,000 iterations for training, which demonstrated limited efficiency.

In analyzing the above literature, we also considered potential improvements by modifying certain layers within the model or freezing some layers while training others. For instance, Idusuyi, Nosa et al. [21] achieved only 80 % accuracy. By adjusting parameters or altering specific layers, the results might be enhanced. Yao, Yuan et al. [22] used over 10,000 iterations; training only the top layer could significantly speed up the process. Based on the identified shortcomings in the existing research, this paper proposes the following optimizations:

1. Selecting a moderately sized dataset to emphasize the impact of the CNN model itself on training results.
2. Adopting more effective training methods to optimize outcomes, analyzing results from multiple perspectives.
3. Using multiple CNN models for comparison to ensure more objective results.

In light of these improvements, this paper selects the highly efficient EfficientNetV2 model for training and explores simulations by selecting different layers and retraining varying numbers of top layers. Additionally, to enhance the model's practical value, we use a corrosion classification dataset covering a wide range of fields for training. The rest of the work in this manuscript is organized as: In Section 2, a brief of CNN structure will be introduced, and the character and advantages of EfficientNetV2 will be given. For better evaluation of the CNN models, the value and the meaning of F1-Score, Recall, and Precision will also be shown. In Section 3, this paper shows the design process and final structure of the model, which explains the role of adding a data augmentation layer and a global pooling layer. Section 4 shows the simulation which is divided into 2 parts and the best EfficientNetV2 model will be selected, and the analysis of the effect of choosing the number of retrain layers will also be given. Finally, Section 5 presents the conclusion and future work concerning the proposed study.

2. Research method

2.1. Brief introduction CNN

CNN is a deep learning model specifically built to analyze data with a grid pattern, such as photographs. Its purpose is to automatically and adaptively learn spatial hierarchies of characteristics [23]. In general, a CNN model at least should have three types of layers or their alternatives. They are convolutional layers, pooling layers and fully connected layers [24]: The convolution layer aims to obtain the feature information of the image, including shape, texture, etc. through convolution, and generate a corresponding number of feature maps containing effective information of the image according to the number of convolution kernels; the pooling layer is to prevent over-fitting, by replacing a certain area of the feature maps with the maximum value (max pooling layer) or average value (average pooling layer) to achieve dimensionality reduction and reduce the amount of calculation; the fully connected layer aims to integrate the previous feature maps to generate a one-dimensional vector, which will be converted into the probability of a specific

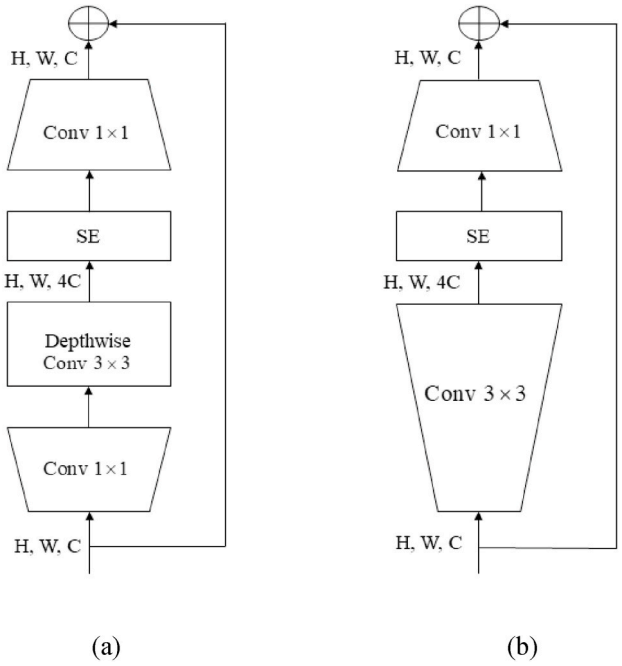

Fig. 3. The structure of (a) MBConv and (b) Fused-MBConv.

classification. The basic structure of a CNN model can be shown in Fig. 2.

By adding different numbers and sizes of convolution-pooling layers and fully connected layers, or improving/alternative their structures, the operating results of the model can be effectively improved. Simonyan, K. et al. [25] examine the impact of the depth of a convolutional network on its accuracy in the context of large-scale image recognition. They evaluate networks with increasing depth using an architecture that employs small (3x3) convolution filters. Their findings demonstrate that by increasing the depth to 16–19 wt layers (VGG16 and 19), significant enhancements can be achieved compared to previous configurations. Given that convolutional networks can achieve greater depth, accuracy, and training efficiency when they incorporate shorter connections between layers between the input and output, Huang, G. et al. [26] proposed the Dense Convolutional Network (DenseNet), where each layer is connected to every other layer in a feed-forward manner. This model can offer substantial enhancements compared to the current best practices in most cases, while also using less computational resources to attain better performance.

2.2. *Brief introduction of EfficientNetV2*

Considering that as the number of model layers increases, the training parameters will increase significantly. Although the complexity of the model can improve the accuracy of model training, it also places higher requirements on the performance of the training equipment. Therefore, it is necessary to find a model that is accurate and can achieve greater effectiveness within limited parameters. Thus, this paper chooses EfficientNetV2 for training. This method is an improved version of EfficientNetV1 which was proposed by Mingxing Tan [27] and inspired by the finding that carefully balancing network depth, width, and resolution can lead to better performance. The main block of EfficientNetV1 is the Inverted Residual Block (MBConv) block [28] and squeeze-and-excitation (SE) optimization [29], whereas EfficientNetV2 uses Fused-MBConvs which replaces the depthwise conv3x3 and expansion conv1x1 in MBConv with a single regular Conv3x3 [30] to replace some early stages of the MBConvs. After combining MBConv and Fused-MBConvs in the best way, the training speed of the model will be improved. Fig. 3 shows the structure of MBConv and Fused-MBConv.

In the above Fig. 3, MBConv (a)The i/p parameter is in HWC layout and it consist of a convolution (Conv) 1×1 (kernel size) block for increasing feature dimension, a Depthwise Conv 3×3 (or 5×5) block carrying out convolutions where each kernel only carries out convolution on the channel that corresponds to it, a SE block for enhancing the ability of a network to represent information by enabling it to dynamically adjust and recalibrate features on a per-channel basis [29], and a Conv 1×1 block to decrease dimension. Fused-MBConv (b) also takes the i/p parameter in HWC layout, however it replaces the first two modules in (a) with Conv 3×3 to increase the training speed.

Besides this, there are also 3 other differences between EfficientNetV2 and EfficientNetV1 [30] which are highlighted as:

a. EfficientNetV2 favours a smaller expansion ratio for MBConv because smaller expansion ratios generally result in lower memory access overhead.

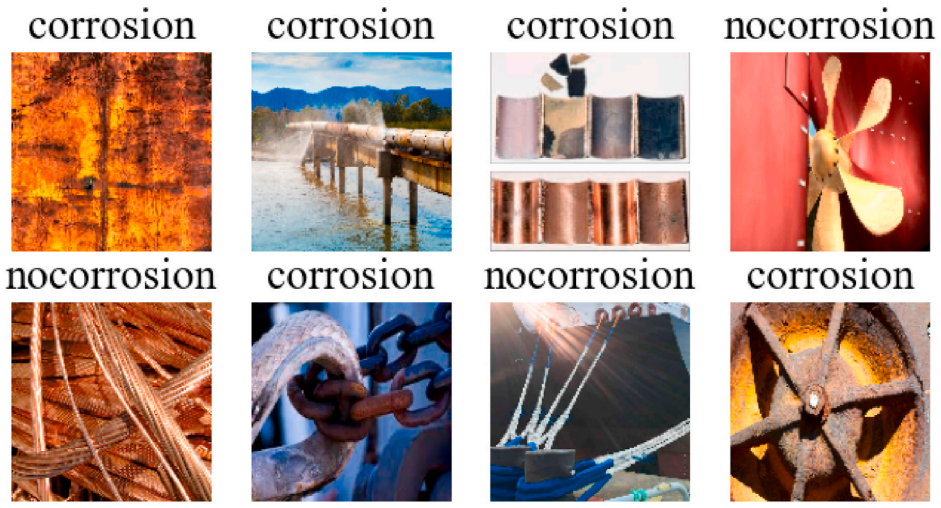

Fig. 4. Random 8 images from the training dataset.

b. EfficientNetV2 prioritises using smaller 3x3 kernel sizes, however it includes additional layers to make up for the smaller receptive field caused by the smaller kernel size.
c. EfficientNetV2 eliminates the last stride-1 step used in the original EfficientNet, likely because of its significant parameter size and the complexity associated with memory access.

By critically assessing the above points, EfficientNetV2 is deemed fit to be an excellent CNN model. In this regard, some of the researchers have applied EfficientNetV2 into their application. Sunil, C. K. et al. [31] proposed a cardamom plant disease detection approach using the EfficientNetV2 model wherein they can obtain an accuracy of 98.26 %. Moreover, Liu, Dingming et al. [32] compared the results of different CNN models to predict the probability of cancer by taking into consideration 9,109 microscopic images, and the results show that EfficientNetV2-b0 to b2 performs at par with MobilenetV2 and InceptionResnetV2.

After having assessed the accuracy quality of EfficientNetV2, this manuscript implements the same on the corrosion image dataset and chooses the metrics in the next sub-section to verify the performance.

2.3. Indicators to evaluate model performance

When using neural networks for classification problems, F1-Score, Recall, Precision, and Accuracy and ROC curve are important indicators to evaluate its performance. These indicators evaluate the model from different aspects according to the definition of their formulas.

For binary classification problems, the final prediction of the model will produce four results:

a. True Positive (TP), correctly predicts positive examples (correctly predicting the result "corrosion").
b. False Positive (FP), incorrectly predicts positive examples ("non-corrosion" is predicted as "corrosion").
c. True Negative (TN), counterexamples of correct predictions ("non-corrosion" is correctly predicted).
d. False Negative (FN), counterexamples of incorrect predictions ("Corrosion" predicted as "non-corrosion").

According to the above four samples, the expressions of Accuracy, Precision and Recall can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

From the above equations, it is to be noted that Accuracy measures the overall situation of the model, that is, whether it can make correct predictions; Precision focuses on the accuracy of the model when the prediction results are positive; Recall focuses on the accuracy when the true results are positive; and F1-score is the weighted average of Precision and Recall, which is a comprehensive

Fig. 5. Data augmentation in one random image.

consideration.

Besides these four parameters, the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) is also commonly used for assessing the discriminative ability of prediction models [33]. The Roc curve can be expressed as a dynamic relationship curve that reflects True Positive Rate (TPR) and False Positive Rate (FPR). If the curve is closer to the upper left corner, that is, the higher the AUC, the better the classification performance of the model, where TPR and recall rate have the same definition, and FPR is defined as the probability of incorrect prediction when the true value is negative:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

Compared with the previous four indicators, this method will more intuitively reflect the predictive ability of the model.

After providing fundamental theory behind EfficientNetV2 and the parameters for evaluating model's performance, the next section will apply the aforementioned theory to a corrosion dataset to formalize the final CNN corrosion image classification model.

3. Model description

For building a widely used classification model, the collection of a corrosion dataset containing a variety of materials is necessary, this dataset includes steels, ships, ship propellers, cars, oil and gas pipelines, concrete rebars, water storage tanks, and stainless steels corrosion and no-corrosion images [34]. This dataset is divided into 70 % training set, 20 % validation set, and 10 % testing set. Fig. 4 shows 8 images from the training dataset.

Through the above description of the dataset, it can be seen that the dataset covers not only a wide range of facilities and equipment but also includes oil and gas pipelines and ship facilities that are susceptible to corrosion, etc. For oil and gas pipelines, the portion of the pipeline that is buried in the soil is susceptible to microbial corrosion, and for ships, because they are in a liquid such as seawater or river water for a long period of time, the dissolved oxygen, the microbial content, and the chemical composition of the liquid may lead to corrosion of the metals inside ships. Therefore, the classification model developed using this dataset is not only versatile but also effective in identifying facilities and equipment that are prone to corrosion.

Certainly, this dataset primarily focuses on collecting images of corrosive and non-corrosive products in the areas mentioned above. As a result, its scope is restricted. Nevertheless, the aforementioned fields are intricately linked to both people's productivity and daily existence. The negative effects of corrosion in these areas not only impact the profitability of production and manufacturing but also risk the safety and well-being of individuals. For instance, the corrosion of oil and natural gas pipelines hinders the efficient transportation of resources resulting in decreased output and affecting the petrochemical industry's production. As an example, the ship and vehicle will see a decrease in their service life due to corrosion. If not addressed promptly, this might potentially lead to safety risks. Hence, despite the dataset's restricted coverage, the development of an appropriate corrosion prediction model using this dataset would nevertheless have significant practical use.

Furthermore, the utilization of transfer learning in this study serves as a validation of the system's ability to generalize. Specifically, the majority of the model's parameters remain unchanged during training, with only the top layers being repeatedly trained. As a result, the model's reliance on the dataset is somewhat limited, allowing it to possess a certain level of generalization ability.

Considering there are 1273 images in the dataset for training which still have room to be enhanced and decrease the overfitting problem, a data augmentation layer containing horizontal and vertical flips, random rotations, random contrast, random brightness and random translations is used. Fig. 5 shows the data augmentation in one random image.

Table 1
Basic information of these models.

Model	Number of layers	Total parameters/MB
B0	270	22.58
B1	334	26.44
B2	349	33.45
B3	409	49.33
S	513	77.56
M	740	202.75
L	1028	449.17
V1B0	238	15.45
V1B1	340	25.08
V1B2	340	29.63
DenseNet121	427	26.85
DenseNet169	595	48.23

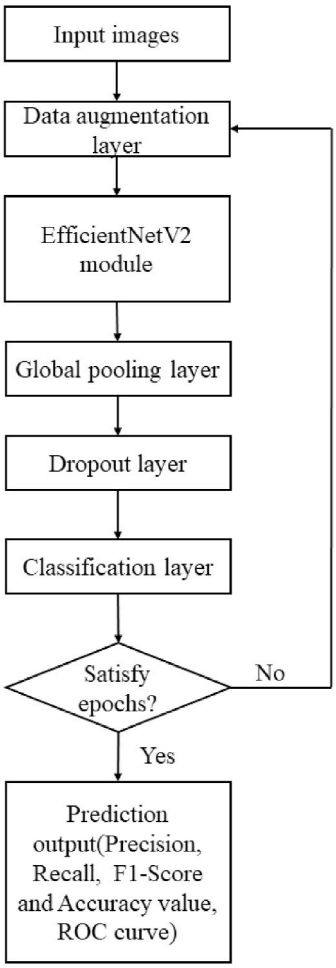

Fig. 6. The overall flowchart.

Vertical and horizontal flipping, random angle rotation, and random translation are methods that enable the model to learn image features from various angles and positions. This enhances the model's ability to generalize. For instance, if corrosion is present only in the centre of an image, the model will not be able to accurately identify corrosion located at the image's edges without these methods. By applying random contrast and random brightness, the model can effectively simulate various lighting conditions. This enables the model to learn how to differentiate corrosion under different image appearances. Additionally, it reduces the system's reliance on a single contrast and brightness setting, thereby enhancing the system's robustness. For instance, the corroded area of the image typically appears darker in colour, and reduced brightness can cause it to blend with the background, making it challenging to

Table 2
Testing accuracy and loss of these models.

Model	Global Average Pooling		Global Max Pooling	
	Accuracy	Loss	Accuracy	Loss
B0	0.8846	0.3228	0.8626	0.3404
B1	0.9011	0.3178	0.8626	0.2894
B2	0.8736	0.3434	0.8516	0.3206
B3	0.8736	0.3342	0.8571	0.3078
S	0.8901	0.3340	0.8626	0.3088
M	0.8132	0.4472	0.8077	0.4439
L	0.8022	0.4570	0.7967	0.4387
V1B0	0.8681	0.3332	0.8462	0.3229
V1B1	0.8571	0.3344	0.8297	0.3510
V1B2	0.8352	0.3917	0.7747	0.4896
DenseNet121	0.8462	0.3329	0.8076	0.4728
DenseNet169	0.8791	0.2810	0.8846	0.3823

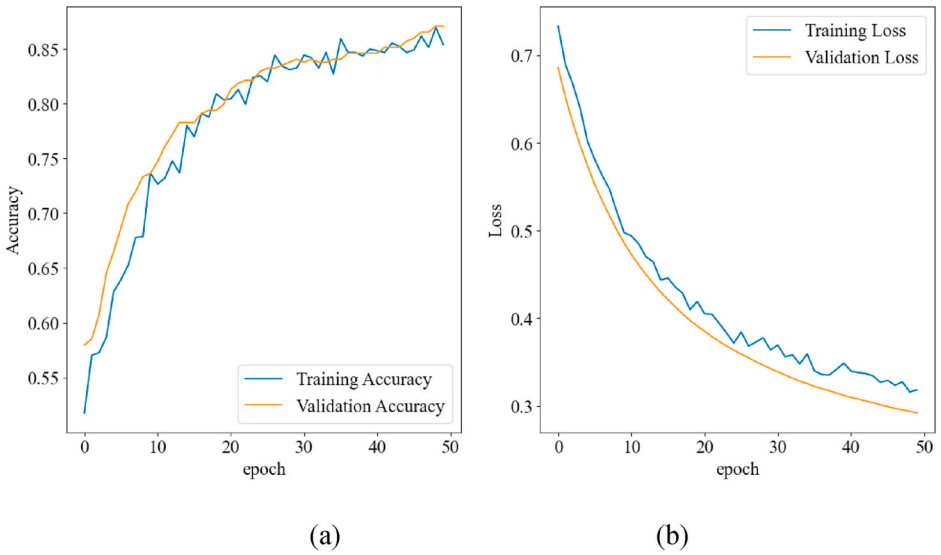

Fig. 7. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2B0.

distinguish between them. whereas increased brightness may also result in the loss of image details. Insufficient or excessive contrast in a picture might lead to the loss of some features, hence impacting the ability to identify corrosion. To summarize, applying flip, rotate, and translation operations to the image will increase the dataset and improve the model's ability to generalize. Similarly, introducing random changes in contrast and brightness can be seen as adding interference terms to enhance the model's resistance to interference.

Post image enhancement, the next part was choosing different sizes of EfficientNetV2 including EfficientNetV2 B0-B3, S, M, and L for training, and adding the EfficientNetV1 B0-B2 and DenseNet121, DenseNet169 for comparison. Table 1 shows their basic information.

From Tables 1 and it is evident that although B0-L are all EfficientNetV2 models, their sizes are different which will affect the training effect. For example, too few training layers may allow the models to be optimized to a limited extent, whereas too many training layers may make the model insufficiently trained with a limited number of samples and iterations, while at the same time, it needs to be verified whether the V2 models have significant advantages over the EfficientNetV1 models or other models like DenseNets mentioned in the previous section. Thus, it is necessary to discuss the effectiveness of the above models for the selected datasets. Therefore, based on the discussion in Section 1, this paper will consider using the original weights of each model in Table 1 to train the dataset, and filtering out the more suitable models based on the results, and then gradually expanding the proportion of retraining layers to the total number of layers, to explore the impact of the gradual expansion of the training parameters on the training results.

To better use the above feature blocks for the binary classification of this dataset, this paper used a Global Pooling layer to replace the fully connected layer. The Global Pooling Layer is used to reduce the dimensionality of data, whereas the Fully Connected layer can add a large number of parameters and then increase the complexity of the model and the degree of overfitting. For the pooling layer, this paper chooses Global Average Pooling and global Max Pooling layer for simulation, where Global Average Pooling can take the average of the feature maps, and the resulting vector is fed directly into the classification layer [35], whereas Global Max Pooling choose the maximum value of the feature maps [36]. Current research shows that the two methods perform differently when dealing

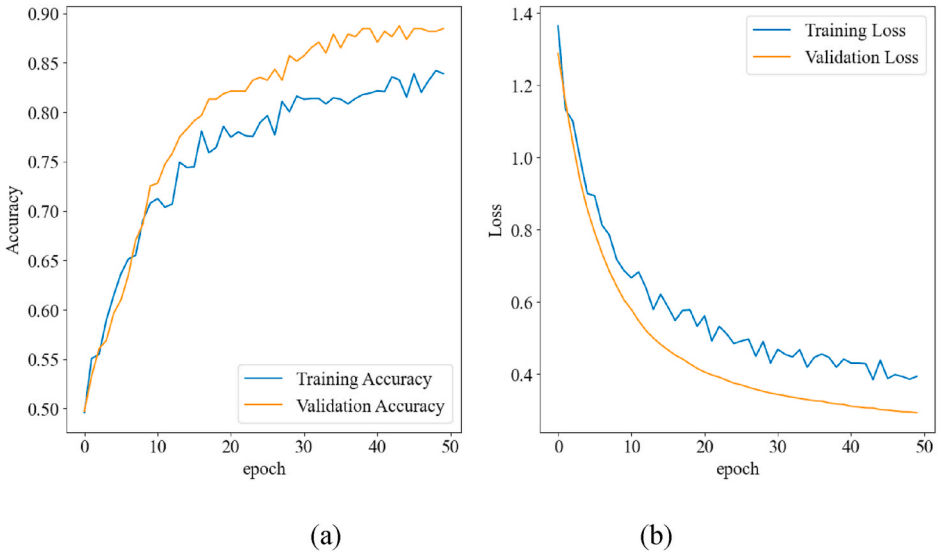

Fig. 8. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2B0.

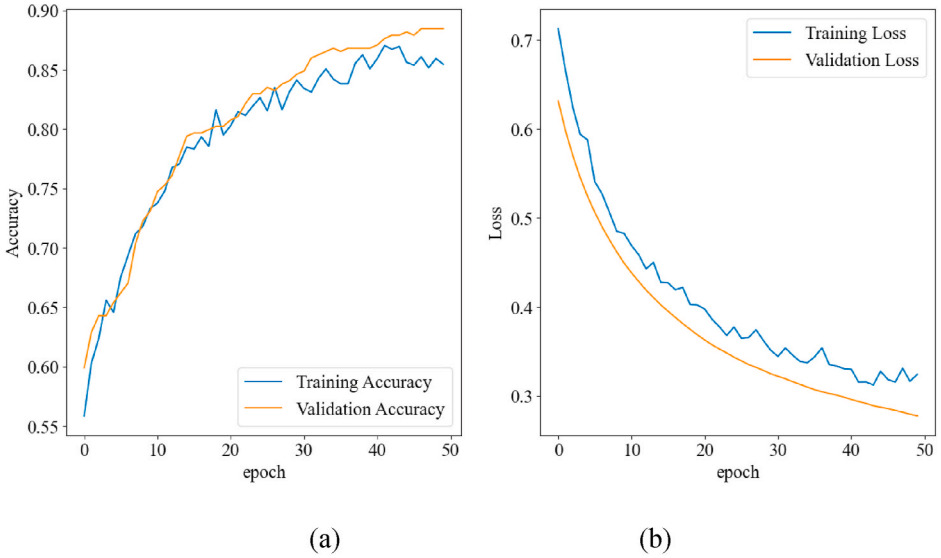

Fig. 9. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2B1.

with different data sets: Zhi Li. et al. [37] use these methods to classify the teeth category into four classes, and the results show max pooling performs better results than average pooling; Bieder et al. [38] compared different pooling methods by training on a large dataset of natural high-resolution images and the results indicate that none of the more advanced techniques exhibit a notable improvement over traditional Max- or Average-Pooling in this classification test. Based on these results, this paper will train models by using these two global pooling methods. In addition, considering the size of the testing dataset is not large which would also cause overfitting issues, a dropout layer is added between the Global Pooling layer and the classification layer to prevent complex co-adaptations [39]. The overall flowchart is shown in Fig. 6.

The final classification model is constructed by the layers: data augmentation layer, EfficientNetV2 module (or other CNN module), global pooling layer, dropout layer, classification layer. From the structure of Fig. 4, this model fully encompasses all the elements discussed in Sections 2 and 3.

Furthermore, the formalized image classification model is examined thoroughly with fine-tuning and adjusting scales which is discussed in the next Section.

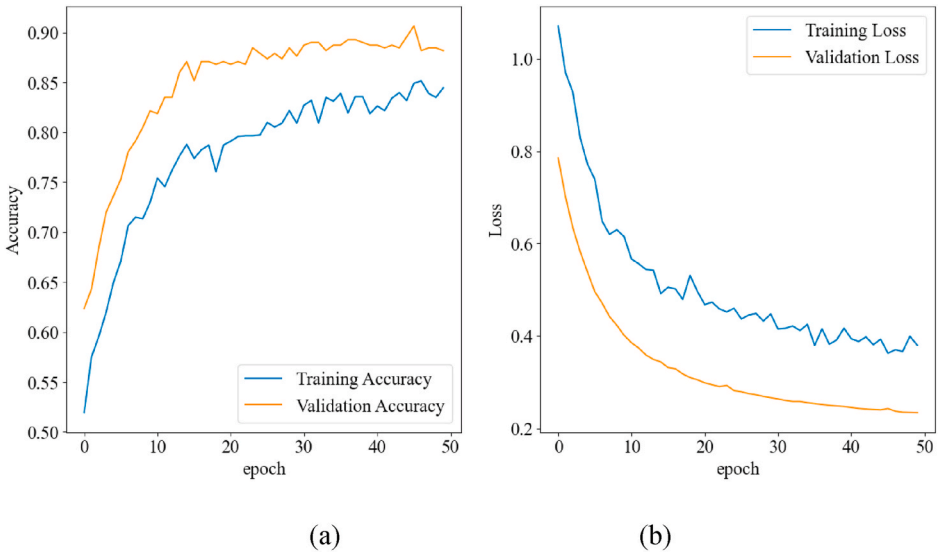

Fig. 10. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2B1.

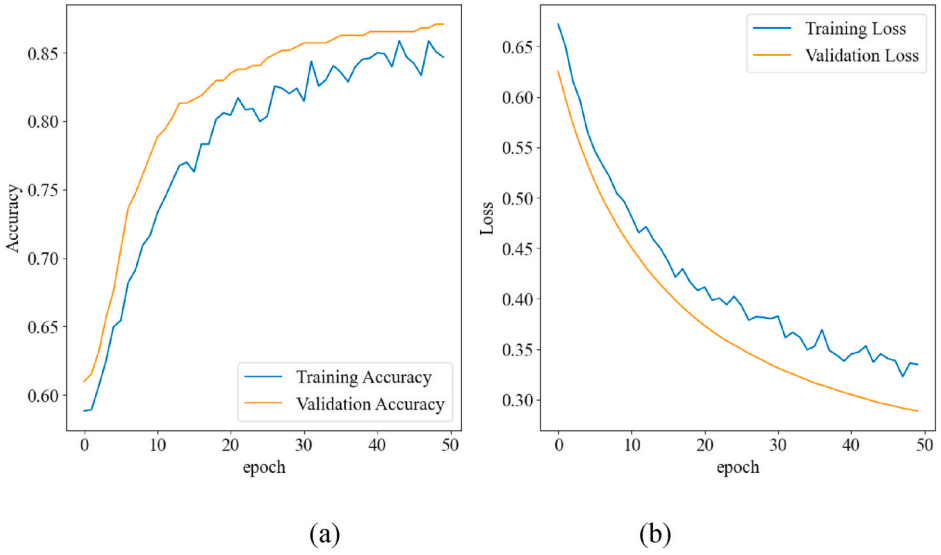

Fig. 11. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2B2.

4. Simulation analysis

This simulation is divided into two parts: one without fine-tuning and one with fine-tuning. The first part involves using 50 epochs, setting the learning rate to 0.0001, a batch size of 32, and a dropout value of 0.2 to train different EfficientNetV2 models. This is done by freezing the layers of EfficientNetV2s and only training their classification layer. The performance is tested using the Global Average Pooling layer and Global Max Pooling layer, providing the training and loss curves along with the testing accuracy value. This part aims to choose suitable EfficientNetV2 models and global pooling layers.

The second part uses 25 epochs to train the classification layer and another 25 epochs to retrain the top layers of EfficientNetV2 along with the classification layer. During this retraining, the learning rate is adjusted to 0.00001 to better update the parameters. This part aims to analyze the impact of the ratio of the number of retrained top layers to the total number of layers on the training results, using the Confusion Matrix, ROC curve, and values of Precision, Recall, and F1-score.

Additionally, to use the model more efficiently, the proposed methodological approach is computed using a 13th Gen Intel(R) Core (TM) i5-13500HX CPU and compiled using Python 3.12. The simulation results of part 1 are shown in Table 2 and Figs. 7–20. Table 2 shows the testing accuracy and loss of these models with different global pooling layers, while Figs. 7–20 illustrate their training and loss curves.

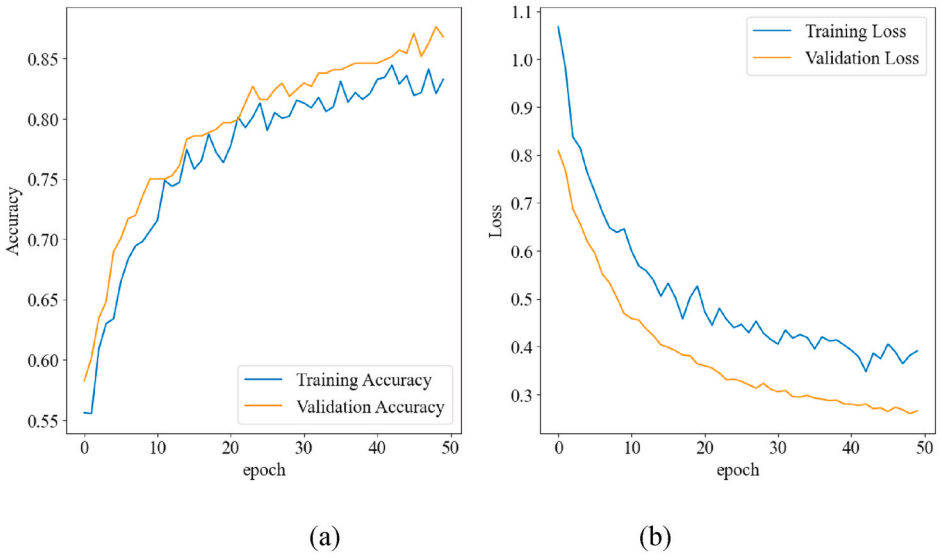

Fig. 12. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2B2.

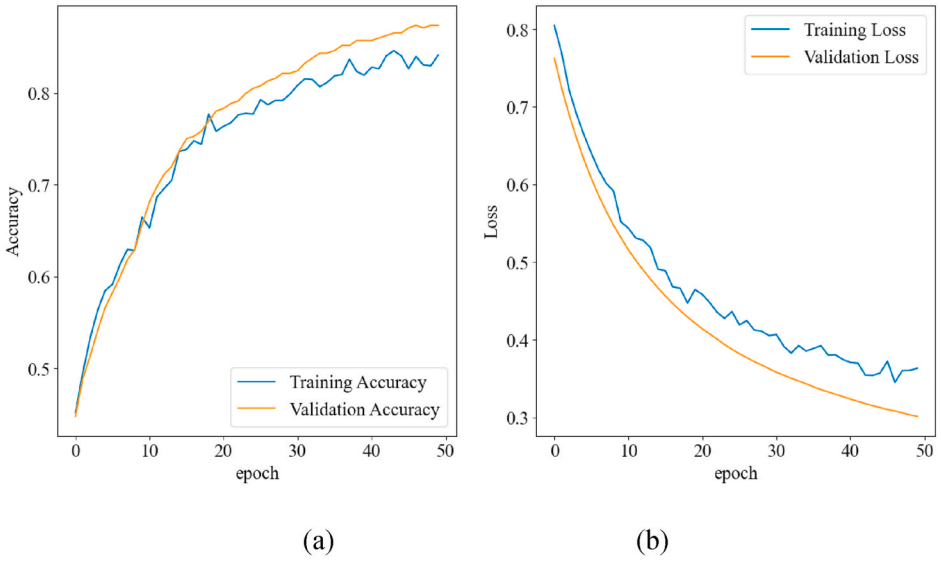

Fig. 13. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2B3.

From these simulation results, several conclusions can be drawn. Firstly, in terms of model selection, according to the results in Tables 1 and 2, we can conclude that this dataset achieves better results on smaller models. As shown in Table 2, the running results of B0, B1, B2, B3, and S can reach an accuracy of more than 85 %, with B0 and S approaching 90 %, and B1 exceeding 90 %, while the accuracy of M and L is above 80 %. Based on these results, B0, B1, and S will be chosen for the next simulations.

Secondly, the results of these models on different global pooling layers varies. According to Table 1, except for DenseNet169, these models achieve higher testing accuracy by applying the Global Average Pooling layer. Their accuracies can be improved by up to 4 % compared to the use of a Global Max Pooling layer. Additionally, according to Figs. 7–20, although the use of both Global pooling layers results in high training and validation accuracies and the training and validation loss curves converge efficiently, the accuracy curves of the training and validation datasets are relatively stable when using the Global Average Pooling layer. In contrast, the validation accuracies are higher than the training accuracies when using the Global Max Pooling layer. Meanwhile, the Global Average Pooling layer yielded essentially the same validation accuracies as the Global Max Pooling layer, while the training accuracies derived from the Global Max Pooling layer were generally lower than those using the Global Average Pooling layer. Therefore, considering the overall accuracies of the models, combining the results obtained from Figs. 7–20 and Table 2, the Global Average Pooling layer will be chosen for the next part of the simulation.

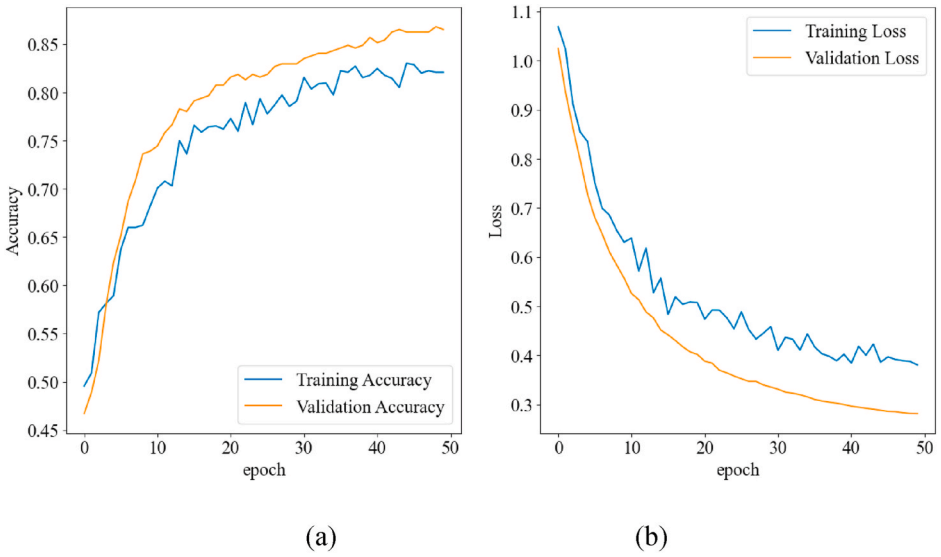

Fig. 14. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2B3.

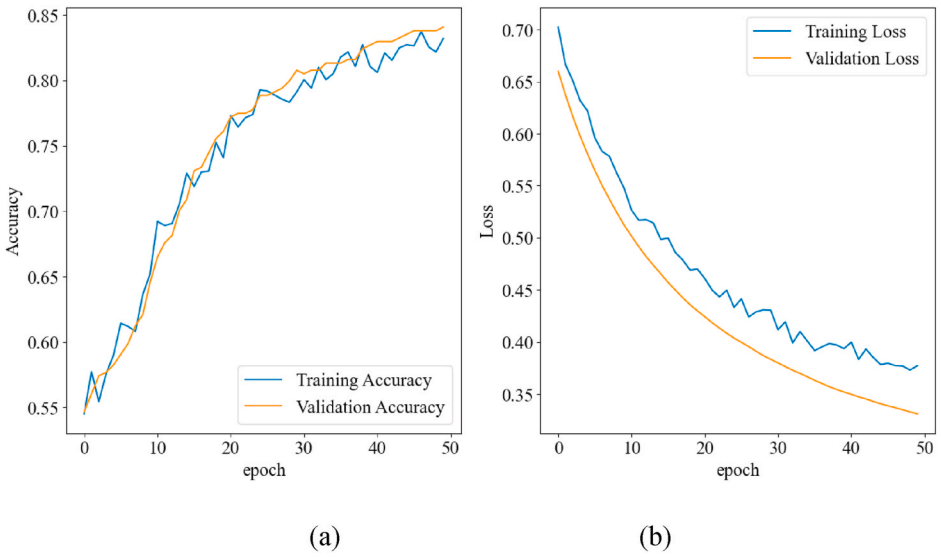

Fig. 15. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2S.

Thirdly, as evident from Figs. 7–20, these models do not show significant overfitting (i.e., the training loss is declining but the validation loss is decreasing until there is a sustained increase). This is because the dropout layer is added in addition to using data augmentation in order to increase the dataset's size, which thereby reduces overfitting by allowing some nodes to be set to zero values during the process of training. This allows the neurons to recognise more fundamental structures in the data rather than just passing on the positive outcomes of the previous layer with little or no changes [40]. As a result, even though the dataset used in this paper is small and a more complex training model is used, the training results do not show overfitting.

Furthermore, compared with EfficientNetV2 models, EfficientNetV1 B0-B2 show lower accuracy at the same parameter levels, and DenseNet121 and 169 require more training time per epoch at the same parameter levels. Thus, these results demonstrate the advantages of EfficientNetV2.

For the simulation in Part 2, which focuses on choosing the number of EfficientNetV2 layers for fine-tuning, it is important to note that with a limited number of epochs and a limited dataset, the parameters will be degraded due to insufficient training if all or most of the EfficientNetV2 layers are retrained, thereby not achieving optimal results and affecting the overall outcomes. Thus, this article only considers simulating by unfreezing 5 %, 10 %, 15 %, and 20 % of the top layers.

The simulation results of V2B0 are shown in Figs. 21–24 and Table 3. The values in the confusion matrixes (subplot (a) in

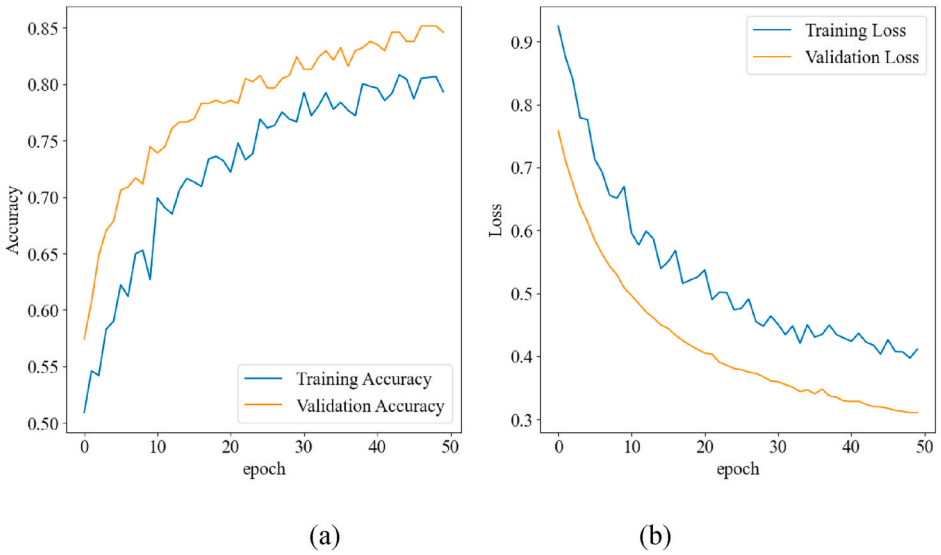

Fig. 16. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2S.

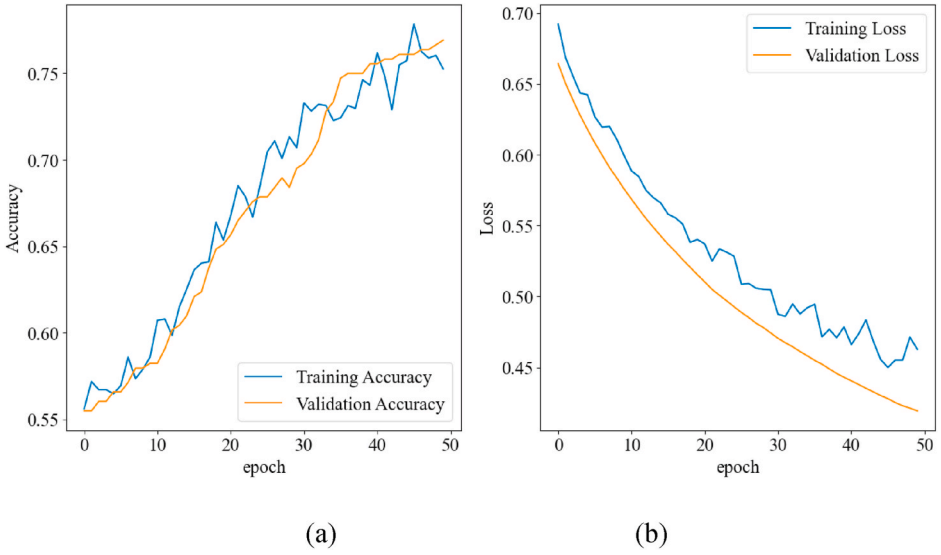

Fig. 17. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2M.

Figs. 21–24) are calculated according to Equations (1)–(5), resulting in the values of the four parameters in Table 3, as well as the ROC curves and AUC values in subplot (b) in Figs. 21–24.

From the simulation results of B0 in Figs. 21–24 and Tables 3 and it is evident that, except for the fine-tuning rate of 15 %, model fine-tuning can effectively improve performance. For instance, when the fine-tuning rate is 20 %, the four indicators in Table 3, except for the precision value, all exceed 0.91. The precision value also exceeds 0.9, and the ROC-AUC reached 0.97. Therefore, these results prove the effectiveness of fine-tuning.

Moreover, the stability and reliability of the model are enhanced with fine-tuning, as evidenced by the consistently high scores across multiple metrics. The fine-tuning process allows the model to adapt better to the specific features of the corrosion dataset, optimizing the parameters for improved detection and classification. This adaptability is crucial in real-world applications where variations in corrosion patterns can occur.

Additionally, fine-tuning at a 20 % rate demonstrates the model's ability to generalize well, reducing the risk of overfitting despite the limited number of epochs. This balance between generalization and precision is vital for deploying the model in practical scenarios, ensuring that it performs well on unseen data.

Therefore, these results not only prove the effectiveness of fine-tuning but also underline the importance of selecting appropriate fine-tuning ratios to achieve optimal model performance.

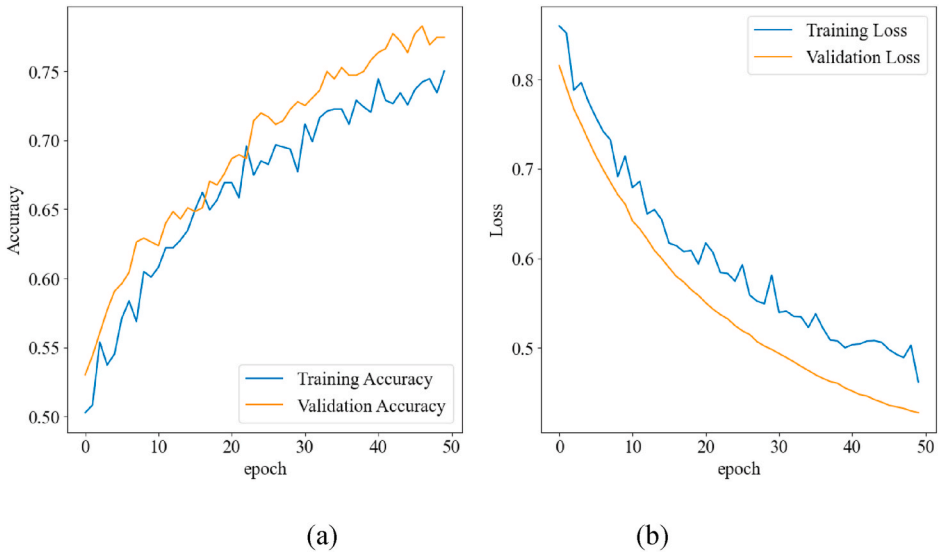

Fig. 18. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2M.

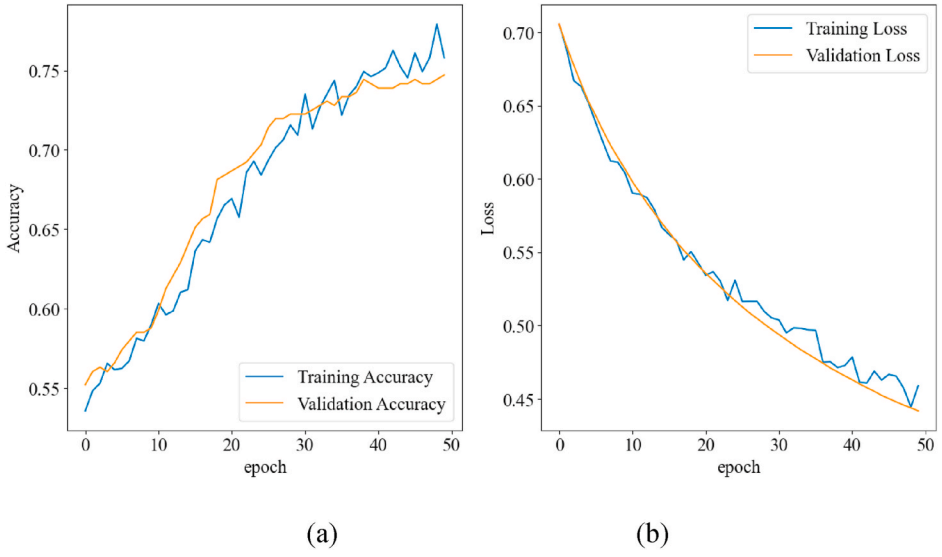

Fig. 19. (a) Training and validation accuracy and (b) Loss in Global Average Pooling of EfficientNetV2L.

The simulation results of V2B1 are shown in Figs. 25–28 and Table 4. The values in the confusion matrixes (subplot (a) in Figs. 25–28) are calculated according to Equations (1)–(5), resulting in the values of the four parameters in Table 4, as well as the ROC curves and AUC values in subplot (b) in Figs. 25–28.

From the simulation results of B1 in Figs. 25–28 and Tables 4 and it could be observed that except for the fine-tuning rate of 10 %, the improvement effect of fine-tuning on this model is not as good as B0, which reflects that its accuracy is lower than pre-fine-tuning.

Furthermore, the results show that fine-tuning at a 10 % rate yields a marginal improvement in performance metrics such as accuracy, precision, recall, and F1-Score. However, for other fine-tuning rates (15 %, 20 %), the performance either stagnates or slightly declines, suggesting a ceiling effect or overfitting when too many layers are retrained. This indicates that B1's architecture might require more sophisticated tuning strategies, such as layer-wise fine-tuning or progressive learning rates, to unlock its full potential.

The relatively lower improvement in B1's performance compared to B0 also underscores the importance of initial model selection. B0's architecture appears inherently more suited to the nuances of the corrosion dataset, allowing for more significant gains through fine-tuning. This observation highlights the necessity of empirical validation when choosing models for specific tasks, as theoretical suitability does not always translate into practical performance.

Additionally, these findings point to a potential area for further research. Exploring different pre-processing techniques,

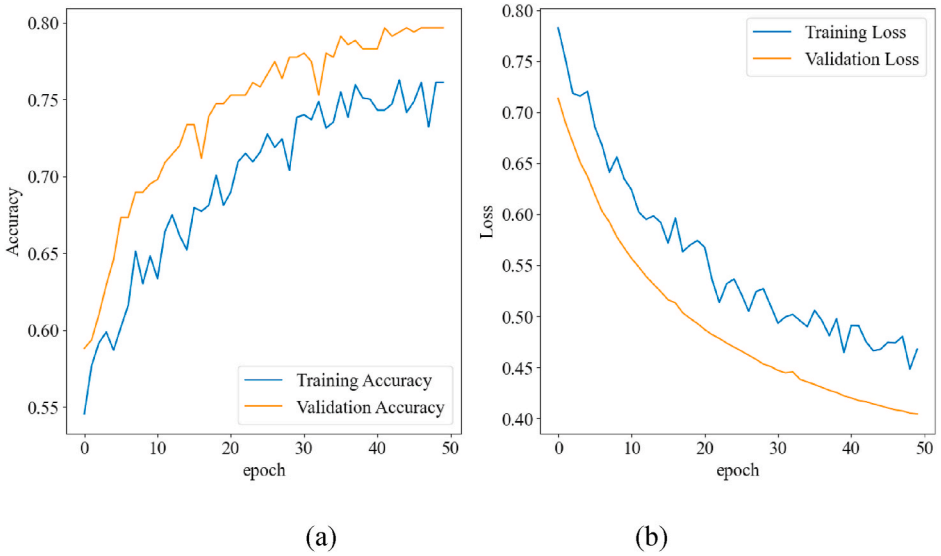

Fig. 20. (a) Training and validation accuracy and (b) Loss in Global Max Pooling of EfficientNetV2L.

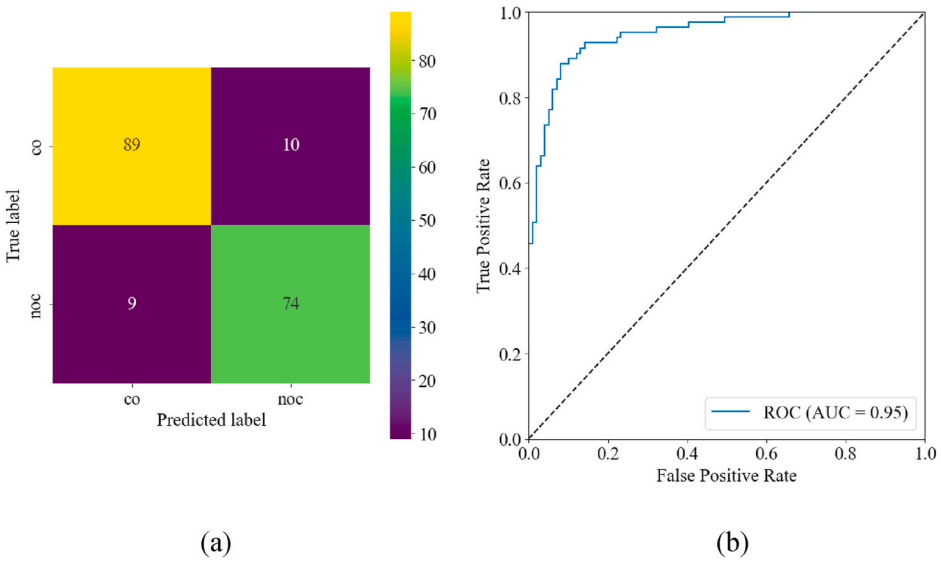

Fig. 21. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B0 unfreezing 5 % top layers.

augmentation strategies, or even hybrid models that combine the strengths of both B0 and B1 could yield better results. Moreover, leveraging transfer learning with more specific domain-related datasets before applying fine-tuning might enhance B1's performance.

The simulation results of V2S are shown in Figs. 29–32 and Table 5. The values in the confusion matrixes (subplot (a) in Figs. 29–32) are calculated according to Equations (1)–(5), resulting in the values of the four parameters in Table 5, as well as the ROC curves and AUC values in subplot (b) in Figs. 29–32.

From the simulation results of S in Figs. 29–32 and Tables 5 and it is evident that when the fine-tuning ratio is 15 %, the model has the best performance: its F1-Score and Recall exceed 0.9, Accuracy exceeds 0.91, the precision value reaches 0.91, and the AUC value reaches 0.97. This indicates that this model, along with B0 fine-tuned at 20 %, can be approximated as a nearly perfect classifier with predictive value compared to the relatively mediocre model predictions at other fine-tuning ratios.

In general, comparing the fine-tuning results of the four models, it is clear that fine-tuning is more effective for B0 and S, which can be used for classification in other datasets of similar size. Although fine-tuning doesn't effectively prove the result in B1, it achieves the highest precision score of 0.9351, indicating its effectiveness in predicting the category "Corrosion." Additionally, model fine-tuning can significantly reduce training time compared to training all model parameters and also reduce the risk of system overfitting. Therefore, model fine-tuning has a significant degree of extensibility. However, it is important to note that the model's performance in

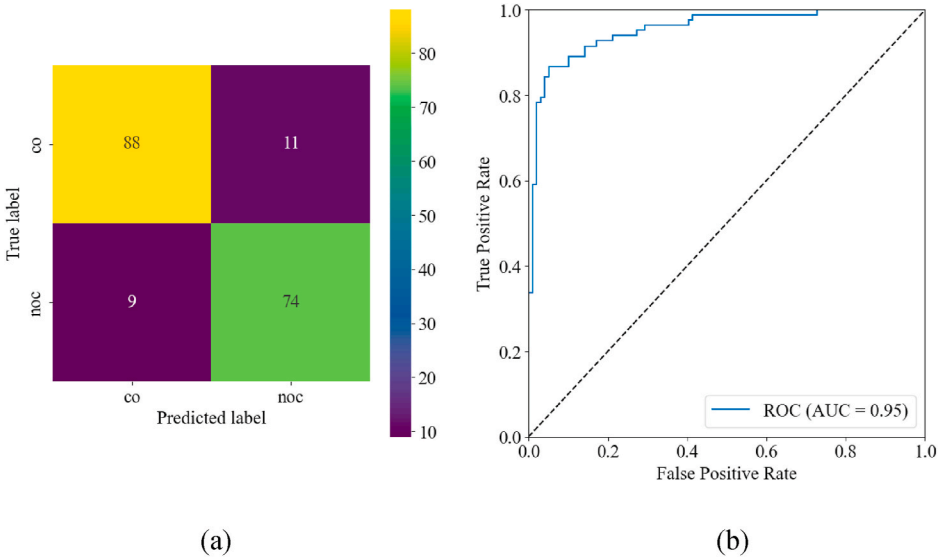

Fig. 22. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B0 unfreezing 10 % top layers.

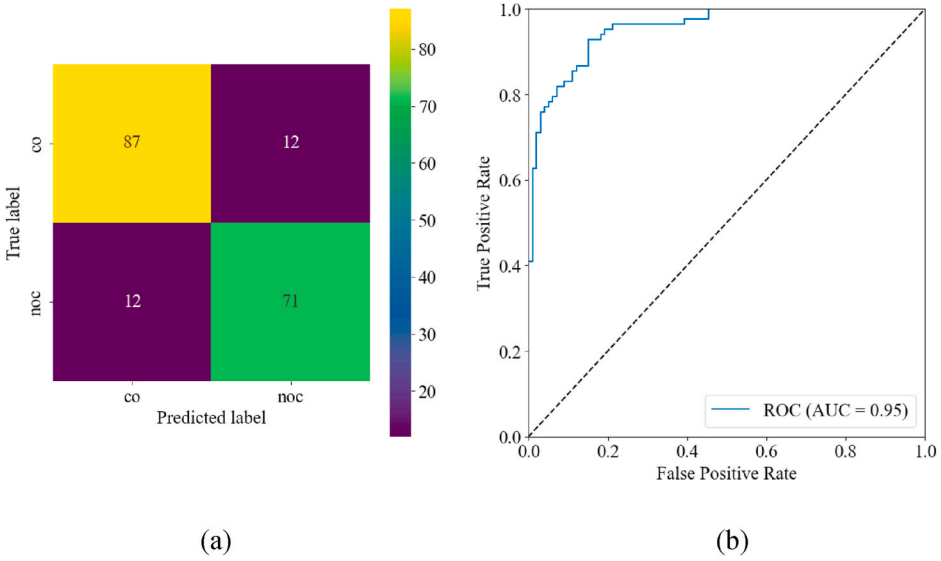

Fig. 23. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B0 unfreezing 15 % top layers.

a controlled test environment may not accurately reflect its performance in a real-world scenario where corrosion detection conditions can vary significantly. To address this, we can initially use a smaller fine-tuning ratio in the controlled environment to obtain the best model, and then further fine-tune it in conjunction with the real scenario. This approach ensures that the resulting model is able to adapt to the real environment.

In particular, even though both V2B0 (fine-tuning in 20 %) and V2S (fine-tuning in 15 %) can achieve high accuracy, the total training time for V2B0 is significantly smaller than for V2S. V2B0 takes about 5–6 s per epoch during the first 25 epochs of training, whereas V2S takes about 12–14 s. During the second 25 epochs of fine-tuning, V2B0 takes about 7–8 s per epoch, while V2S takes about 21–23 s. Therefore, if time cost is a consideration, training and fine-tuning V2B0 would be a good choice.

Furthermore, due to the limited performance of the hardware equipment used in this paper, only 50 epochs were selected for training. As can be seen from the results in Figs. 7–20, even the training curves with better performance (i.e., the results obtained using the Global Average Pooling layer) exhibit some fluctuations. These fluctuations are caused partly by the addition of the data augmentation layer, which introduces a certain degree of interference (see Section 3 for details), and partly due to the insufficient number of total training epochs, preventing the curves from converging more completely. Thus, the model can be further optimized by increasing the training time when equipment conditions allow it.

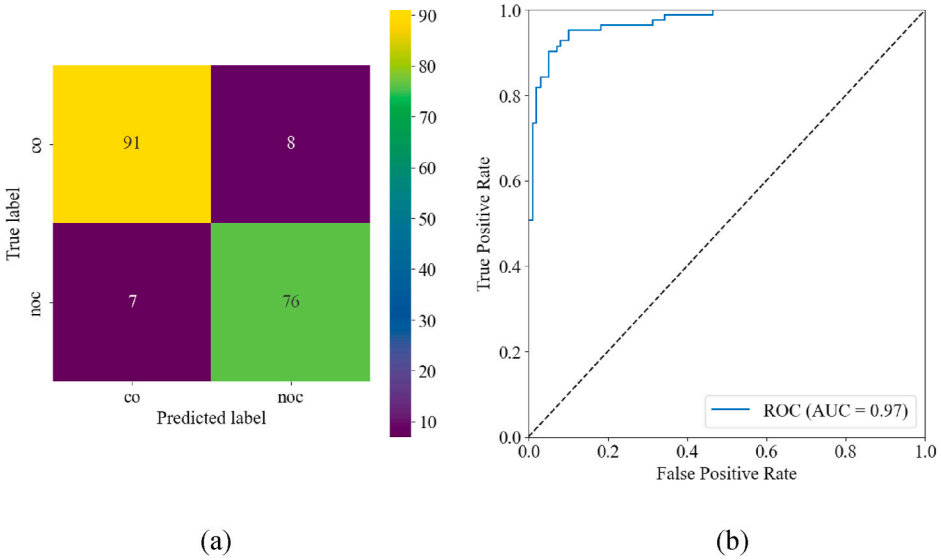

Fig. 24. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B0 unfreezing 20 % top layers.

Table 3
Testing results of EfficientNetV2B0.

Rate	precision	Recall	F1-score	Accuracy
5 %,	0.8810	0.8916	0.8862	0.8956
10 %	0.8706	0.8916	0.8810	0.8901
15 %,	0.8554	0.8554	0.8554	0.8681
20 %	0.9048	0.9157	0.9102	0.9176

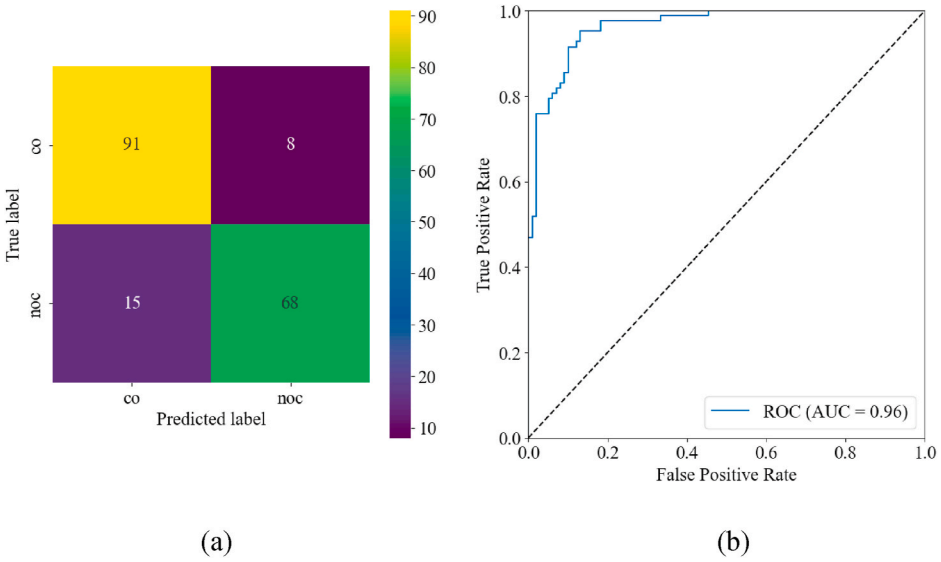

Fig. 25. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B1 unfreezing 5 % top layers.

In addition, given the substantial computing requirements of EfficientNetV2, it may not be feasible to deploy it in conditions with low computational resources. Therefore, it is necessary to use appropriate hardware and software strategies to maximize the algorithm's value for deployment. From a hardware perspective, GPUs (Graphics Processing Units) are capable of carrying out computational tasks due to their architectural design. While CPUs are known for their speed and versatility in completing a series of tasks that involve numerous interactions, however they are also composed of a limited number of cores and a large number of caches [41]. As a

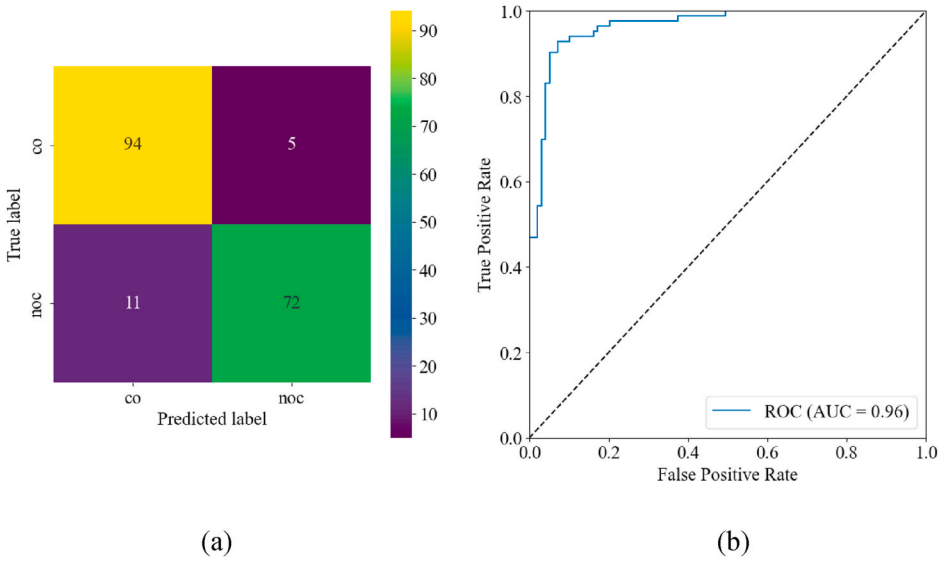

Fig. 26. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B1 unfreezing 10 % top layers.

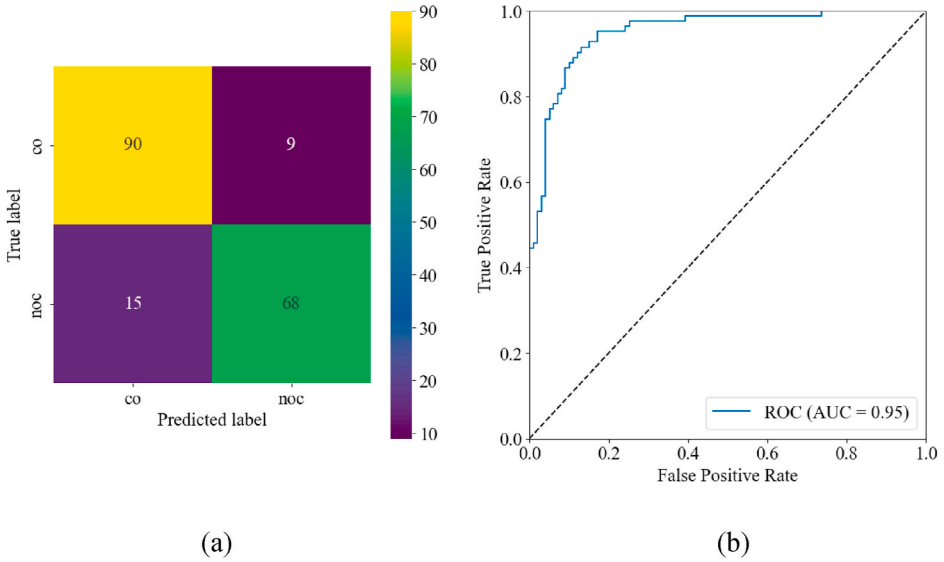

Fig. 27. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B1 unfreezing 15 % top layers.

result, CPUs can only handle a few software threads at a time, wherein due to simultaneous execution of high number of threads in respective cores, there is also a delay in task execution due thread context switching and core context switching. In contrast, GPUs are equipped with hundreds of cores and can handle thousands of threads simultaneously [41]. Utilizing GPUs for processing helps to efficiently reduce the computational expenses. Utilizing transfer learning on the software side can significantly decrease the number of parameters needed for training. Given that the V2B0 and V2S models in this study are versions of EfficientNetV2 with fewer parameters, therefore performing transfer learning on this foundation is a more efficient approach for saving computational resources. Thus, based on the preceding explanation, enhancing EfficientNetV2 in terms of both hardware and software enables its deployment in scenarios with restricted computing resources.

5. Conclusion

To address the current frequent material corrosion problem, we proposed a corrosion image classification method based on EfficientNetV2, utilizing a binary corrosion image dataset containing various corrosion materials and facilities. This method focuses on two main parts: first, testing the accuracy to determine which Global Pooling Layer and EfficientNetV2 module is most suitable for the

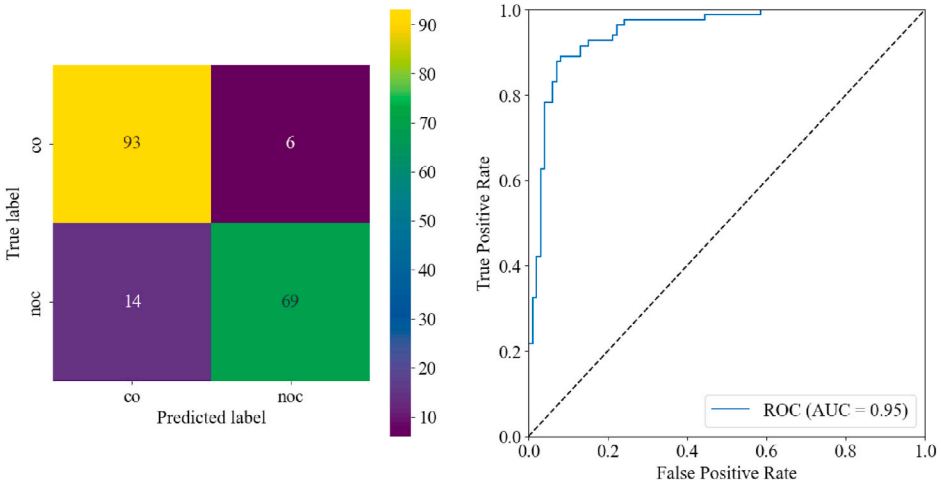

Fig. 28. (a) Confusion matrix and (b) ROC curve of EfficientNetV2B1 unfreezing 20 % top layers.

Table 4
Testing results of EfficientNetV2B1.

Rate	precision	Recall	F1-score	Accuracy
5 %,	0.8947	0.8193	0.8553	0.8736
10 %	0.9351	0.8675	0.9	0.9121
15 %,	0.8831	0.8193	0.85	0.8681
20 %	0.92	0.8313	0.8734	0.8901

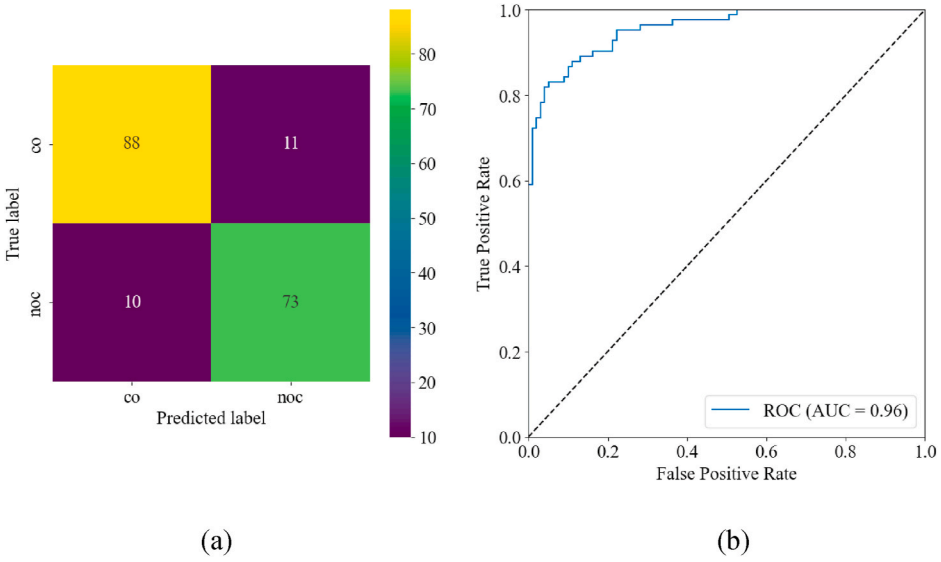

Fig. 29. (a) Confusion matrix and (b) ROC curve of EfficientNetV2S unfreezing 5 % top layers.

dataset, and second, setting different fine-tuning ratios to evaluate the classification performance based on Accuracy, Precision, Recall, F1-Score, and ROC-AUC values.

The simulation results indicate that the classification model using the Global Average Pooling layer instead of the fully connected layer and selecting B0, B1, and S modules can achieve higher accuracy. Specifically, better results can be obtained by using the B0 module with a fine-tuning rate of 20 % and the S module with a fine-tuning rate of 15 %. Additionally, compared to EfficientNetV1 and

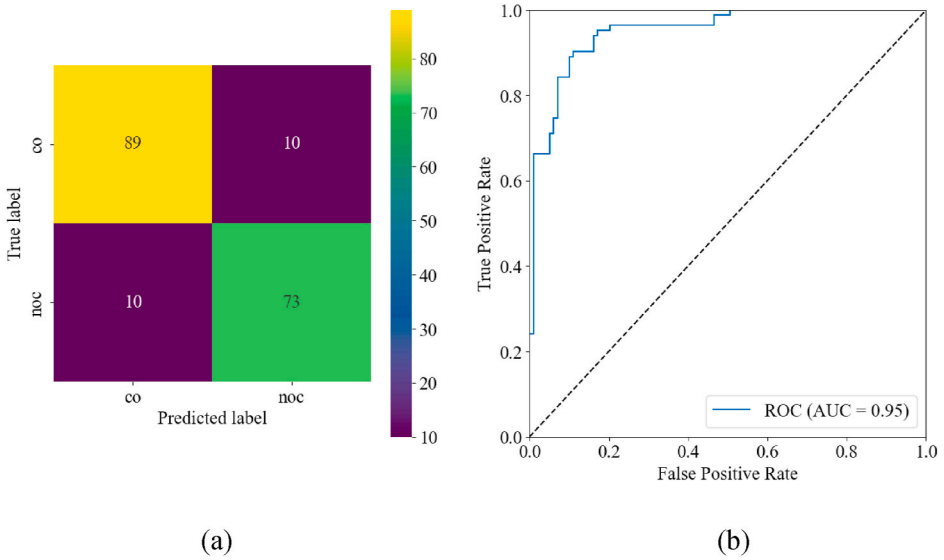

Fig. 30. (a) Confusion matrix and (b) ROC curve of EfficientNetV2S unfreezing 10 % top layers.

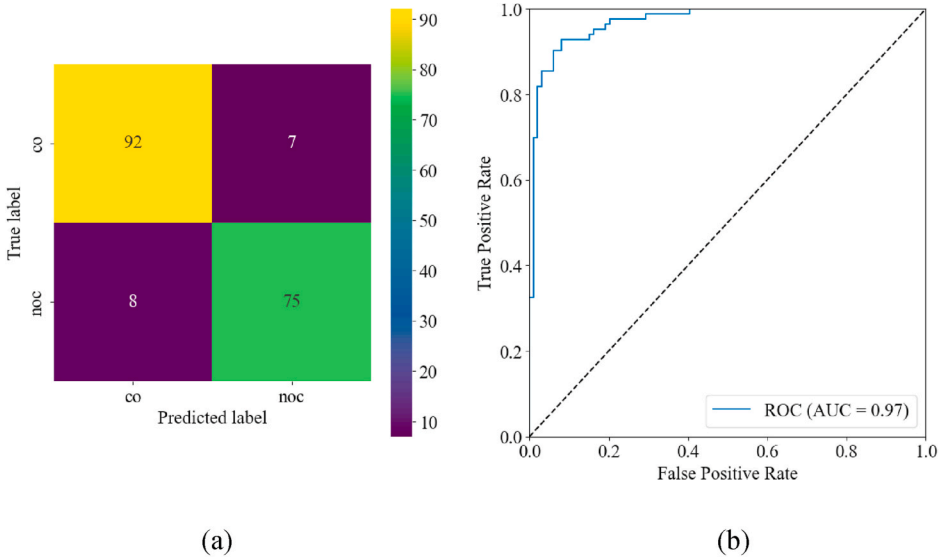

Fig. 31. (a) Confusion matrix and (b) ROC curve of EfficientNetV2S unfreezing 15 % top layers.

DenseNet, EfficientNetV2 achieves high accuracy and faster training speeds. The findings of this study can provide an effective reference for subsequent corrosion research.

Although this method achieves high accuracy, there are still areas for improvement. For example, increasing the number of training iterations and fine-tuning ratios could reduce fluctuations in the training curve and allow more parameters to be fully optimized, thereby expanding the scope for further analysis. Moreover, this paper did not modify the structure of the EfficientNetV2 module itself, so future research will explore the impact of replacing and optimizing certain components within EfficientNetV2 on the simulation results.

Data availability statement

The data that support the findings of this study are available in [Deep Learning for Automated Corrosion Detection] at https://github.com/pjsun2012/Phase5_Capstone-Project/tree/main/data, reference number [34]. These data were derived from the following resources available in the public domain: https://github.com/pjsun2012/Phase5_Capstone-Project.git.

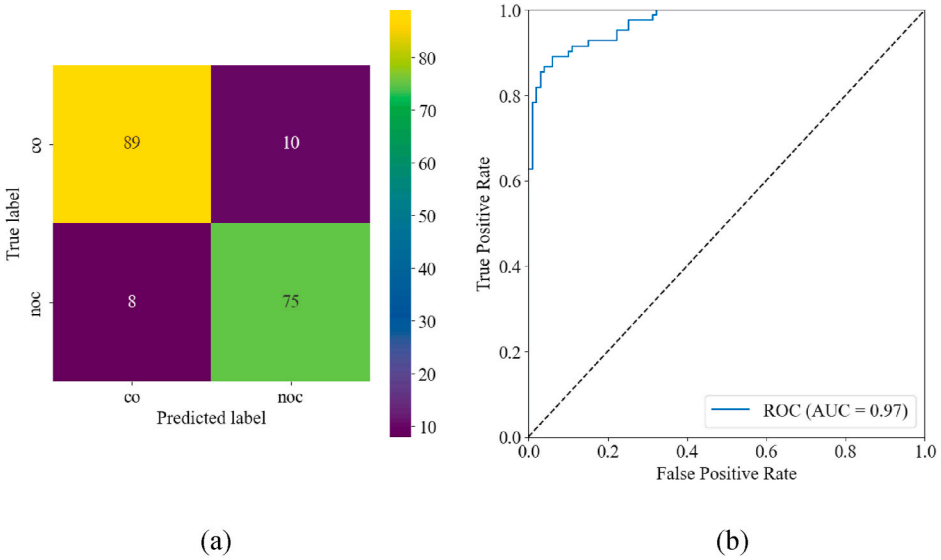

Fig. 32. (a) Confusion matrix and (b) ROC curve of EfficientNetV2S unfreezing 20 % top layers.

Table 5
Testing results of EfficientNetV2S.

Rate	Precision	Recall	F1-score	Accuracy
5 %,	0.8690	0.8795	0.8743	0.8846
10 %	0.8795	0.8795	0.8795	0.8901
15 %,	0.9146	0.9036	0.9091	0.9176
20 %	0.8824	0.9036	0.8929	0.9010

CRediT authorship contribution statement

Ziheng Zhao: Software, Methodology, Investigation, Formal analysis, Conceptualization. **Elmi Bin Abu Bakar:** Resources, Project administration, Funding acquisition. **Norizham Bin Abdul Razak:** Writing – review & editing, Visualization, Validation, Formal analysis. **Mohammad Nishat Akhtar:** Supervision, Project administration, Investigation, Funding acquisition, Conceptualization.

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.

Acknowledgement

The authors would like to acknowledge Research and Creativity Management Office (RCMO) Universiti Sains Malaysia for supporting this research. The authors also acknowledge the grant 304/PAERO/6315761 provided by RCMO, Universiti Sains Malaysia.

References

[1] Barbara Shaw, Robert Kelly, What is corrosion? Electrochem. Soc. Interface 15 (1) (2006) 24, https://doi.org/10.1149/2.F06061IF.
[2] C. Guedes Soares, et al., Influence of environmental factors on corrosion of ship structures in marine atmosphere, Corrosion Sci. 51 (9) (2009) 2014–2026, https://doi.org/10.1016/j.corsci.2009.05.028.
[3] Pierre R. Roberge, Handbook of Corrosion Engineering, vol. 1128, Mcgraw-hill, New York, 2000.
[4] Redvers N. Parkins, A Review of Stress Corrosion Cracking of High Pressure Gas Pipelines, NACE CORROSION. NACE, 2000.
[5] G.H. Koch, et al., Chapter 1-Cost of Corrosion in the United States Handbook of Environmental Degradation of Materials, William Andrew Publishing, Norwich, NY, 2005, https://doi.org/10.1016/B978-081551500-5.50003-3.
[6] Sanjay Kumar Ahuja, Manoj Kumar Shukla, A survey of computer vision based corrosion detection approaches, Information and Communication Technology for Intelligent Systems (ICTIS 2017)- 2 2 (2018) 55–63, https://doi.org/10.1007/978-3-319-63645-0_6.
[7] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012), https://doi.org/10.1145/3065386.
[8] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT press, 2016, https://doi.org/10.1007/s10710-017-9314-z.
[9] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.
[10] Yanming Guo, et al., Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48, https://doi.org/10.1016/j.neucom.2015.09.116.

[11] Jianxin Wu, Introduction to Convolutional Neural networks." National Key Lab for Novel Software Technology 5.23, Nanjing University, China, 2017, p. 495.
[12] Hüseyin Fırat, Classification of microscopic peripheral blood cell images using multibranch lightweight CNN-based model, Neural Comput. Appl. 36 (4) (2024) 1599–1620, https://doi.org/10.1007/s00521-023-09158-9.
[13] Wadhah Ayadi, et al., Deep CNN for brain tumor classification, Neural Process. Lett. 53 (2021) 671–700, https://doi.org/10.1007/s11063-020-10398-2.
[14] Mengyi Liu, et al., Deeply learning deformable facial action parts model for dynamic expression analysis. Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12, Springer International Publishing, 2015, https://doi.org/10.1007/978-3-319-16817-3_10.
[15] Saeed Khaki, Lizhi Wang, Sotirios V. Archontoulis, A CNN-RNN framework for crop yield prediction, Front. Plant Sci. 10 (2020) 492736, https://doi.org/10.3389/fpls.2019.01750.
[16] Jiaohua Qin, et al., A biological image classification method based on improved CNN, Ecol. Inf. 58 (2020) 101093, https://doi.org/10.1016/j.ecoinf.2020.101093.
[17] Ivan Malashin, et al., Deep learning approach for pitting corrosion detection in gas pipelines, Sensors 24 (11) (2024) 3563, https://doi.org/10.3390/s24113563.
[18] G. Ramkumar, Hybrid model for detection of corrosion in water pipeline images using CNN and comparing accuracy with SVM, ECS Trans. 107 (1) (2022) 13861, https://doi.org/10.1149/10701.13861ecst.
[19] Deegan J. Atha, Mohammad 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] Blossom Treesa Bastian, et al., Visual inspection and characterization of external corrosion in pipelines using deep neural network, NDT E Int. 107 (2019) 102134, https://doi.org/10.1016/j.ndteint.2019.102134.
[21] Nosa Idusuyi, et al., Corrosion classification study of mild steel in 3.5% NaCl using convolutional neural networks, FUOYE J Eng Technol 7 (2022) 61–64, https://doi.org/10.46792/fuoyejet.v7i1.773.
[22] Yuan Yao, et al., Artificial intelligence-based hull structural plate corrosion damage detection and recognition using convolutional neural network, Appl. Ocean Res. 90 (2019) 101823, https://doi.org/10.1016/j.apor.2019.05.008.
[23] Rikiya Yamashita, et al., Convolutional neural networks: an overview and application in radiology, Insights into imaging 9 (2018) 611–629, https://doi.org/10.1007/s13244-018-0639-9.
[24] Keiron O'shea, Ryan Nash, An introduction to convolutional neural networks, arXiv preprint arXiv:1511.08458 (2015), https://doi.org/10.48550/arXiv.1511.08458.
[25] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, https://doi.org/10.48550/arXiv.1409.1556, 2014.
[26] Gao Huang, et al., Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, https://doi.org/10.48550/arXiv.1608.06993.
[27] Mingxing Tan, Quoc Le, Efficientnet: rethinking model scaling for convolutional neural networks. International Conference on Machine Learning, PMLR, 2019, https://doi.org/10.48550/arXiv.1905.11946.
[28] Mark Sandler, et al., Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, https://doi.org/10.48550/arXiv.1801.04381.
[29] Jie Hu, Li Shen, Gang Sun, Squeeze-and-excitation networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, https://doi.org/10.1109/CVPR.2018.00745.
[30] Mingxing Tan, Quoc Le, EfficientNetV2: smaller models and faster training. International Conference on Machine Learning, PMLR, 2021, https://doi.org/10.48550/arXiv.2104.00298.
[31] C.K. Sunil, C.D. Jaidhar, Patil Nagamma, Cardamom plant disease detection approach using EfficientNetV2, IEEE Access 10 (2021) 789–804, https://doi.org/10.1109/ACCESS.2021.3138920.
[32] Dingming Liu, et al., EfficientNetV2 model for breast cancer histopathological image classification. 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), IEEE, 2022, https://doi.org/10.1109/IWECAI55315.2022.00081.
[33] A. Janssens, J.W. Cecile, Forike K. Martens, Reflection on modern methods: revisiting the area under the ROC Curve, Int. J. Epidemiol. 49 (4) (2020) 1397–1403, https://doi.org/10.1093/ije/dyz274.
[34] Pengju Sun, Deep learning for automated corrosion detection. https://github.com/pjsun2012/Phase5_Capstone-Project/tree/main/data, 2021.
[35] Min Lin, Qiang Chen, Shuicheng Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013), https://doi.org/10.48550/arXiv.1312.4400.
[36] Hannah Kim, Young-Seob Jeong. "Sentiment classification using convolutional neural networks.", Appl. Sci. 9 (11) (2019) 2347, https://doi.org/10.3390/app9112347.
[37] Zhi Li, et al., Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling, Int. J. Imag. Syst. Technol. 29 (4) (2019) 577–583, https://doi.org/10.1002/ima.22337.
[38] Florentin Bieder, Robin Sandkühler, Philippe C. Cattin, Comparison of methods generalizing max-and average-pooling, arXiv preprint arXiv:2103.01746 (2021), https://doi.org/10.48550/arXiv.2103.01746.
[39] Geoffrey E. Hinton, et al., Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, https://doi.org/10.48550/arXiv.1207.0580, 2012.
[40] What is the dropout layer?, Data Basecamp, 2023. https://databasecamp.de/en/ml/dropout-layer-en.
[41] Brian Caulfield, What's the Difference between a CPU and a GPU? NVIDIA, 2009. https://blogs.nvidia.com/blog/whats-the-difference-between-a-cpu-and-a-gpu/.