buildings

MDPI

Article

A Deep Learning Framework for Corrosion Assessment of Steel Structures Using Inception v3 Model

Xinghong Huang [1], Zhen Duan [1], Shaojin Hao [1], Jia Hou [2,3,*], Wei Chen [2,3] and Lixiong Cai [2,3]

[1] CSCEC 7th Division International Engineering Co., Ltd., Guangzhou 510080, China; 349228@whut.edu.cn (X.H.); 274419@whut.edu.cn (Z.D.); lh2691121083@126.com (S.H.)
[2] School of Civil Engineering and Architecture, Wuhan University of Technology, Wuhan 430062, China; iamhappychen@163.com (W.C.); cailixiong@whut.edu.cn (L.C.)
[3] Sanya Science and Education Innovation Park, Wuhan University of Technology, Sanya 572024, China
* Correspondence: houjia@whut.edu.cn

Abstract: Corrosion detection plays a crucial role in the effective lifecycle management of steel structures, significantly impacting maintenance strategies and operational performance. This study presents a machine vision-based approach for classifying corrosion levels in Q235 steel, providing valuable insights for lifecycle assessment and decision-making. Accelerated salt spray tests were performed to simulate corrosion progression over multiple cycles, resulting in a comprehensive dataset comprising surface images and corresponding eight loss measurements. A comparative evaluation with other architectures, namely, AlexNet, ResNet, and VggNet, demonstrated that the Inception v3 model achieved superior classification accuracy, exceeding 95%. This method offers an effective and precise solution for corrosion evaluation, supporting proactive maintenance planning and optimal resource allocation throughout the lifecycle of steel structures. By leveraging advanced deep learning techniques, the approach provides a scalable and efficient framework for enhancing the sustainability and safety of steel infrastructure.

Keywords: steel structure; corrosion level; machine vision; convolutional neural network; Inception V3

check for updates

Academic Editor: Grzegorz Ludwik Golewski

Received: 22 January 2025
Revised: 3 February 2025
Accepted: 5 February 2025
Published: 7 February 2025

Citation: Huang, X.; Duan, Z.; Hao, S.; Hou, J.; Chen, W.; Cai, L. A Deep Learning Framework for Corrosion Assessment of Steel Structures Using Inception v3 Model. *Buildings* **2025**, *15*, 512. https://doi.org/10.3390/buildings15040512

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Steel structures have gained widespread adoption due to their advantages, including lightweight design, high industrialization, environmental sustainability, energy efficiency, and recyclability [1]. Although steel structures offer numerous advantages, their susceptibility to corrosion in environments with high salt spray concentrations has multifaceted implications for long-term infrastructure planning [1,2]. Economically, corrosion significantly increases the full lifecycle cost of projects. During the construction phase, additional anti-corrosion measures must be implemented, raising the initial cost. In the operational phase, continuous capital investment is required for frequent inspections, maintenance, and repairs. In terms of safety, corrosion threatens the strength and stability of steel structures. A safety incident resulting from corrosion could lead to severe casualties, property damage, and social disruption. Therefore, in-depth research into the corrosion issues of steel structures and the exploration of effective solutions are crucial for ensuring the safe and stable operation of long-term infrastructure. Currently, manual visual inspection is the most commonly used method in engineering. This approach depends on the inspector's experience, leading to low efficiency, high subjectivity, elevated costs, and considerable safety risks [1–3]. Although non-destructive testing techniques such as ultrasonic testing,

eddy current testing, and infrared testing offer accurate results and are frequently used for factory testing of parts and components, they often incur prohibitive time and cost implications when applied to large spans and complex structures in engineering [3–6]. With the continuous advancements in image processing and neural network technology, machine vision detection has emerged as a new inspection method [7–10]. The advent of Convolutional Neural Networks (CNNs) has greatly enhanced the accuracy of image classification and object detection. They not only improve the accuracy and efficiency of inspections but also provide a more intelligent solution for the maintenance of steel structures [11–13].

Convolutional Neural Networks (CNNs) are capable of automatically extracting important features from images through multi-layer convolution operations and performing complex pattern recognition layer by layer during corrosion detection tasks [14–18]. This capability for automated feature extraction and learning enables CNNs to handle corrosion images under different scales and environmental conditions, thereby improving detection accuracy and consistency. Additionally, the deep learning structure of CNNs demonstrates better generalization ability when processing large volumes of data, allowing adaptation to various steel surface issues such as cracks and rust [9,13,19,20]. Faster R-CNN enhances detection accuracy by precisely locating corrosion regions using Region Proposal Networks (RPNs) and further improving it with a classification network [21–23]. U-Net and SegNet focus on image segmentation tasks; through encoder–decoder architectures and feature decoders, they can accurately segment corrosion areas and handle corrosion images with high complexity and noise [24–26]. AlexNet, as an early deep learning model, although having fewer layers, effectively extracts high-level features from images, laying the foundation for steel structure corrosion detection [27–29]. VGGNet, with its deep convolutional network structure, can identify complex corrosion patterns and perform fine analysis of subtle corrosion phenomena. ResNet addresses the gradient vanishing problem in deep network training by introducing residual blocks, resulting in higher precision and stability when processing complex corrosion images [30].

The Inception module in Convolutional Neural Networks (CNNs) enhances network expressiveness and efficiency by applying various convolution and pooling operations in parallel, enabling the extraction of features at multiple scales [31,32]. The introduction of depthwise separable convolutions and 1×1 convolutions reduces computational complexity, enabling the model to operate efficiently in environments with limited computational resources [33,34]. Inception v3 further optimizes this design by combining convolutional kernels of different sizes, improving its ability to recognize complex images. With its multi-scale feature extraction capabilities and efficient computational performance, Inception v3 provides accurate and real-time detection results for steel structure corrosion images. These characteristics make it particularly effective in addressing complex corrosion patterns [35–37].

In this study, weight loss and surface images of Q235 steel at various corrosion cycles were obtained through indoor accelerated salt spray corrosion experiments. A corrosion level standard was established to determine the approximate corrosion levels of the images, which were used as the dataset for the classification model. To validate the Inception v3 model's effectiveness for corrosion classification, three classic CNN architectures—AlexNet, ResNet, and VGGNet—were selected as control models. These models were compared with the lightweight Inception v3 model using the same dataset and accuracy (Acc) as the primary evaluation metric while keeping model training hyperparameters consistent. This comparison assessed the applicability and accuracy of the Inception v3 model for corrosion classification, offering a new technological approach for the efficient operation and maintenance of steel structure engineering.

2. Related Works

2.1. Inception

The Inception module in CNNs is an innovative architecture known for its modular design and multi-path parallel processing [38,39]. It combines convolutional kernels of different sizes with pooling layers, allowing the model to extract features at various scales while reducing computational complexity through 1 × 1 convolutions [40]. The core advantage of the Inception module lies in its flexible structure and efficient feature extraction capabilities, making it highly effective across multiple visual tasks and achieving a good balance in resource utilization [41].

The most direct way to enhance network performance is by increasing the depth of the convolutional network. However, as the number of layers increases, so do the parameters, which can lead to issues such as overfitting and gradient vanishing. Additionally, the explosion in parameter numbers causes a significant increase in memory usage during network training, leading to longer training times and suboptimal training results [42]. As shown in Figure 1, the core idea of the Inception module is to combine different convolutional layers in parallel. The result matrices processed by these convolutional layers are concatenated along the depth dimension to form a deeper matrix. The Inception module can be stacked repeatedly to form larger networks, allowing for efficient expansion in both depth and width. This approach enhances the accuracy of deep learning networks while preventing overfitting [43,44]. An advantage of the Inception module is its ability to perform dimensionality reduction on larger matrices while aggregating visual information at different sizes, facilitating feature extraction from various scales [45–47].

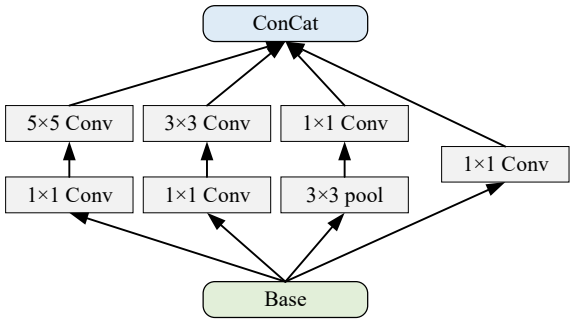

Figure 1. Inception module.

The Inception module is capable of flexibly selecting appropriate computational paths while maintaining high performance. The introduction of depthwise separable convolutions and 1 × 1 convolutions reduces computational complexity and parameter count, enabling the model to operate efficiently in resource-constrained environments [38–40]. Furthermore, parameter sharing and path optimization within the module further decrease computational overhead, making the Inception module particularly well suited for lightweight applications. It simultaneously extracts multi-scale and multi-level features, enhancing the network's expressive power [41–43]. The Inception module overcomes the computational bottlenecks associated with deep networks in traditional convolutional architectures and can effectively adapt to various complex tasks, as demonstrated by its widespread use in several CNN variants, including Inception v3 [48–50].

The Inception v3 model is an efficient architecture within CNNs, incorporating a "modular" design that combines convolutional kernels and pooling operations of different sizes to extract rich features. This approach allows the model to capture features at various scales while reducing computational burden [51,52]. By introducing combinations of multiple kernel sizes, the network can extract features across different scales, thereby improving its ability to recognize complex images. The Inception v3 model offers significant advantages

in the construction industry. Its high accuracy excels in classifying corrosion levels, where subtle differences are evident, and its deep network structure optimizes computational efficiency through branching mechanisms. This reduces the demand for computational resources, enabling real-time monitoring of large-scale building structures [53–55].

Inception v3 effectively extracts features at different scales, enhancing the ability to identify various building surface issues such as cracks and corrosion. The model performs stably under different conditions and adapts well to actual building inspection environments [48–50]. These advantages make Inception v3 highly promising for structural health monitoring and maintenance in construction. Through efficient image analysis, it can assist in identifying building types, detecting damage, monitoring construction quality, and aiding design optimization and automated measurement. These applications significantly enhance the management and maintenance efficiency of construction projects.

2.2. AlexNet

AlexNet marked the beginning of the era of deep convolutional networks. With its powerful image processing capabilities, AlexNet can accurately identify different types of buildings, monitor construction progress in real time, detect potential issues in building structures, and assist in optimizing architectural design, thereby enhancing management efficiency and quality control in construction projects [56–58]. Compared to LeNet, AlexNet utilizes ImageNet as the training dataset, resulting in a substantial increase in both the number of image categories and the amount of data. The incorporation of dropout, a regularization technique, mitigates the issue of network overfitting. Additionally, the use of the ReLU activation function, which has a faster convergence rate, improves model training efficiency [59–61].

The application of AlexNet in steel structure corrosion detection primarily leverages its powerful feature extraction capabilities provided by its deep convolutional neural network. As a classic convolutional neural network, AlexNet effectively extracts complex features from images of steel structure surfaces, capturing subtle corrosion details through its multi-layered network structure. Its robust image classification performance enables AlexNet to identify and classify varying degrees of corrosion, facilitating efficient and accurate detection of steel structure corrosion and providing a scientific basis for structural maintenance [62–66].

2.3. ResNet

ResNet builds upon the VGG architecture by introducing residual connections (skip connections), addressing the gradient vanishing problem in deep networks. This innovation enables the network to train deeper layers, improving both model performance and accuracy. In summary, ResNet enhances the depth and learning capability of the network based on VGG's design principles through its residual structure [67–69]. ResNet introduces a residual module named "residual", as shown in Figure 2. In a typical network module, the desired nonlinear mapping is F(x). With the introduction of an identity shortcut path, the desired mapping becomes H(x) = F(x) + x. The purpose of this adjustment is to combine the learned F(x) with the original x to better capture features. Compared to F(x), H(x) is easier to optimize. In extreme cases, if F(x) = 0, then H(x) = x indicates that the module has not learned any features and has only performed an identity mapping, which ensures that network performance is not adversely affected. In practice, the residual F(x) will not be zero, allowing the stacked layers to learn new features based on the input features, thereby achieving better performance.

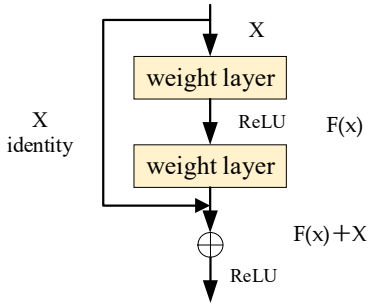

Figure 2. Residual module.

The results indicate that, although ResNet uses convolutional kernels of the same scale as VGG-19, its resolution of the network degradation problem allows it to construct deeper networks [70–72]. Compared to VGG, ResNet achieves lower training errors and higher accuracy on test sets. In the construction industry, ResNet's applications include building image classification, construction progress monitoring, and structural defect detection. Its innovative residual learning mechanism enables deep networks to effectively capture complex features in building images, accurately identify different types of buildings, detect issues such as cracks and corrosion in structural elements, and monitor construction quality and progress in real time. This enhances management efficiency and maintenance effectiveness for construction projects [71–75].

2.4. SegNet

ResNet has further advanced the exploration of network depth by addressing challenges in training deep networks [76,77]. SegNet improves upon the FCN's transmission logic by modifying the VGG-16 network as its backbone structure. During encoding, the model extracts features through convolution operations, with SegNet employing "same" convolutions to maintain the same dimensions as the original image after convolution. During decoding, the model also uses "same" convolutions, but these convolutions serve to enrich the information in the upsampled image, allowing information lost during pooling to be recovered through learning in the decoding process. SegNet's ability to recover information lost during pooling is particularly beneficial for construction applications. By maintaining spatial integrity and enhancing boundary detection, SegNet provides accurate and reliable segmentation masks, enabling precise structural detail extraction. This capability is crucial for tasks such as damage assessment, condition monitoring, and maintenance planning, ultimately contributing to safer and more efficient construction practices. In the decoding stage, the downsampled feature maps are upsampled, followed by convolutional processing of the upsampled images to refine the geometric shapes of objects in the image. This process restores the features obtained during encoding to the specific pixel locations in the original image [78–80].

SegNet is primarily used in the construction industry for image segmentation tasks, such as automated building outline recognition, structural detail extraction, and construction area marking. Its efficient encoder–decoder network structure allows SegNet to accurately segment different regions in building images, identify and label building components or damaged areas, and enhance the accuracy of design plans and the efficiency of construction process monitoring [81–83].

3. Methodology

3.1. Datasets

As the primary model for corrosion level assessment in this study, the classification model must ensure that the training dataset is large and representative. Q235 steel is a

commonly used carbon structural steel, known for its good machinability and weldability, and is widely applied in construction, bridges, and machinery. Its chemical composition and physical properties make it representative of the general corrosion behavior of steel, making it ideal for corrosion experiments. Using Q235 steel ensures that the experimental results are both representative and practical, providing valuable insights for corrosion protection in steel structures. Therefore, this study selects Q235 specimens, as shown in Figure 3, to investigate the corrosion behavior of Q235 steel in a salt fog environment. Physical data, including surface changes and mass loss across different corrosion cycles, are collected, and the corrosion level for each cycle is determined according to relevant standards [23]. Additionally, photographs are taken of each specimen for every cycle to capture images corresponding to the corrosion state, and these images, along with the corrosion levels, constitute the dataset for steel corrosion levels.

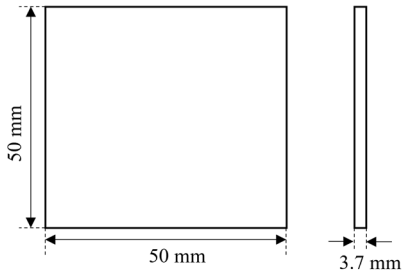

Figure 3. Specifications of experimental specimens.

3.1.1. Salt Fog Test

As shown in Figure 3, the specimens used in this experiment are 50 mm × 50 mm × 3.7 mm in size and made of Q235B steel. Figure 4 shows the most pristine polished steel sheet. The specific types of materials and the number of specimens used in the experiment are detailed in Table 1. Table 2 presents the list of experimental equipment.

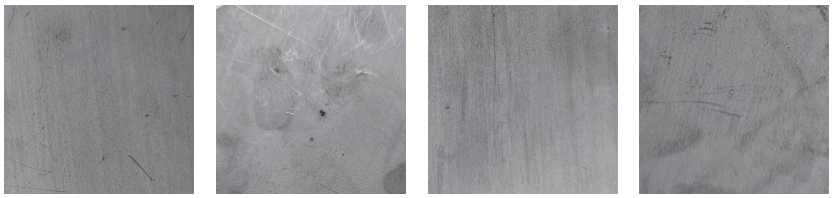

Figure 4. Experimental specimen setup.

Table 1. Experimental material setup.

Experiment Number	Specimen Material	Quantity (Pieces)
1	Q235	40
2	Q235	40
3	Q235	40
4	Q235	40
5	Q235	40

The salt fog test was conducted over 8 cycles, with each cycle lasting 4 h. A 5% NaCl (w/v) solution was used to generate the salt fog, and the experiment followed the American Society for Testing and Materials (ASTM) standard ASTM B117-19 [84]. The experimental steps are as follows:

(1) Prepare the reagents and add a sufficient amount to the salt fog chamber.
(2) Open the salt fog chamber, place the specimens inside as shown in Figure 5, and adjust the working environment (temperature, air pressure, etc.).

(3) After the first experimental cycle, remove the fog from the salt fog chamber, take out the specimens, and rinse the surface of the specimens with distilled water to remove salt.
(4) Place all specimens in the constant temperature drying oven to dry.
(5) Take photographs to obtain images of the corrosion on the specimens' surfaces.
(6) Remove the specimens (do not return them to the salt fog chamber), use an ultrasonic acid cleaning machine to remove surface corrosion, and dry them.
(7) Weigh the cleaned specimens, record the data, and determine the corrosion level based on mass loss and the proportion of the corrosion area.
(8) Repeat steps (2)–(7) until the full cycle experiment is completed.

Table 2. Experimental equipment list.

Equipment Name	Model	Purpose
Electronic Salt Fog Chamber (JITAI Corporation, Beijing, China)	JiTai 90	Indoor accelerated salt fog laboratory
Silent Air Compressor (AOTUSI Corporation, Taizhou, China)	ZhiPu 1490W-30L	Supplying air pressure to the salt fog chamber
Constant Temperature Drying Oven (SUBO Corporation, Shaoxing, China)	SuBo 101-00B	Drying specimens
Ultrasonic Cleaning Machine (SHANGPU Corporation, Shanghai, China)	ShangYi SN-QX-08E	Cleaning corrosion sites
Precision Electronic Scale (YOUKEWEITE Corporation, Kunshan, China)	LeQi 0.001 g	Weighing specimens

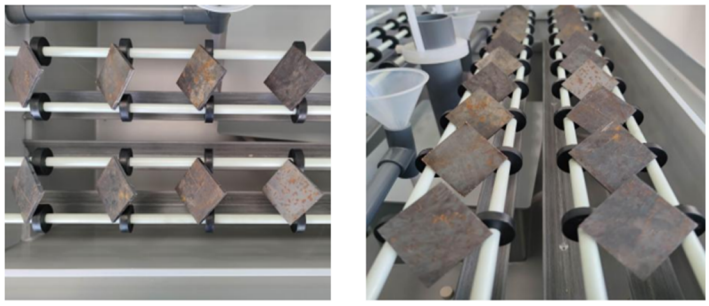

Figure 5. Placement of specimens in the salt fog chamber.

A preliminary experiment is first conducted to observe and determine the corrosion cycles and characteristics of the material in the salt fog environment, preventing potential data gaps in the formal experiments due to an overly large cycle span. The preliminary experiment on Q235 steel revealed the following patterns: initially, the corrosion rate of the specimens is high, and surface changes are clearly noticeable. However, once extensive corrosion coverage is reached, the corrosion rate decreases, and surface changes become less observable to the naked eye. In response to this, non-uniform intervals for the experimental cycles were established based on the preliminary experiment. Table 3 and Figure 6 illustrate the time intervals and corresponding corrosion conditions for each cycle, using Q235 steel specimens as an example. The formal experiment was then carried out in accordance with the corrosion patterns outlined in Table 3 to collect image and mass loss data for each specimen.

Table 3. Time corrosion cycles and surface conditions.

Title 1	Title 2	Title 3	Title 4
1	4 h	4 h	Pitting corrosion observed
2	8 h	4 h	Pitting corrosion intensifies and spreads further
3	12 h	4 h	Spot corrosion appears
4	24 h	12 h	Extensive corrosion observed
5	48 h	24 h	Corrosion almost fully covered
6	72 h	24 h	Corrosion fully covered with deepened color
7	96 h	24 h	Deepened corrosion color and beginning to flake off

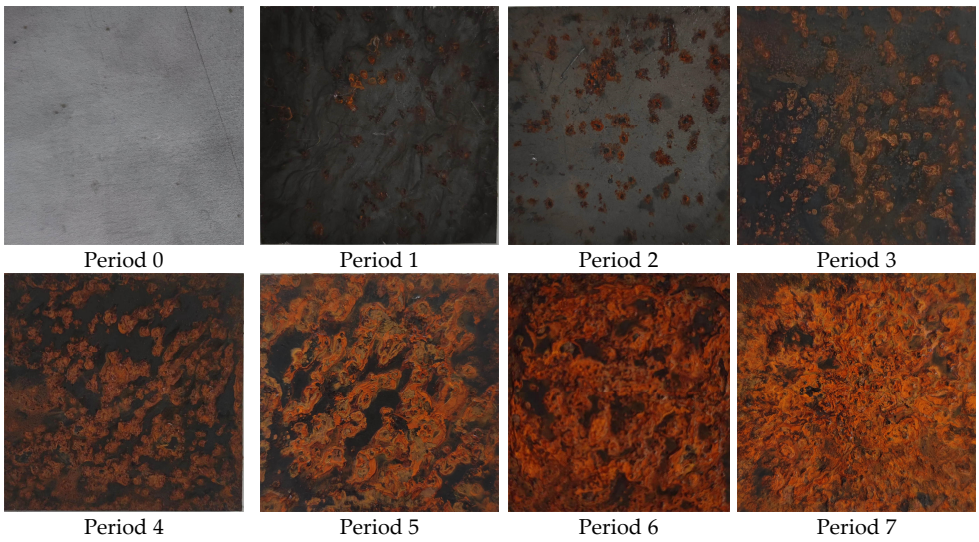

Figure 6. The surface of each corrosion period of the specimen.

3.1.2. Classification Dataset

1. Judgment Criteria

Corrosion level refers to the extent of corrosion on metal specimens and is another key aspect of this experiment. It has been found that surface observation alone is insufficient for accurately determining the corrosion level, especially when the specimen is fully covered by corrosion in the later stages, making surface changes less noticeable. Therefore, it was decided to use the weight loss method in conjunction with surface observation to determine the corrosion level. To ensure consistency and synchronization, we strictly control the time intervals between surface imaging and weight loss measurements in each experiment. Standardized sample handling procedures are employed to minimize error sources. During preliminary experiments, it was observed that the mass of the specimens might initially increase before decreasing over time due to corrosion loss. This makes weight gain calculations inadequate for accurately reflecting the corrosion extent, particularly in the later stages of the experiment. Consequently, in the formal experiment, the weight loss method (mass loss method) was adopted to calculate the mass lost due to corrosion oxidation. According to the American Society for Testing and Materials (ASTM) standard ASTM G1-03 [85] for the preparation, cleaning, and evaluation of corrosion specimens, the formula for calculating the corrosion rate using weight loss is given in Equation (1).

$$\eta = (W_0 - W_1)/W_0 \quad (1)$$

In the formula, η represents the corrosion rate, W_0 is the initial mass of the specimen, and W_1 is the mass of the specimen measured during the experiment.

The final corrosion levels, corresponding corrosion cycles, mass loss, and surface corrosion conditions are summarized in Table 4.

Table 4. Corrosion level system.

Corrosion Level	Corrosion Cycle	Mass Loss	Surface Corrosion Condition
0	Initial	0%	Smooth surface, no corrosion
1	1	0–0.5%	Pitting, corrosion at scratches
2	2, 3	0.5–2%	Patches of corrosion, strip corrosion
3	3, 4, 5	2–5%	Extensive corrosion, full coverage
4	5, 6, 7	>5%	Complete coverage, uneven surface, deepened color

2. Dataset

The corrosion level dataset consists of surface images of specimens at various corrosion cycles obtained from the experiments, along with the corresponding corrosion levels determined based on mass loss and surface corrosion conditions. The samples are organized into folders according to their corrosion levels, with images of the same corrosion level stored in the same folder and named accordingly. These images are then stored in the training files for the steel corrosion level classification model. Due to time constraints, a total of 651 valid corrosion images were collected and classified. The classification was based on image quality, clarity of corrosion features, consistency across experimental cycles, and relevance to the experimental conditions. Each experimental cycle provided approximately 150 valid images. To mitigate the issue of limited data, data augmentation techniques were employed, as shown in Figure 7. These techniques, including rotation and flipping, were applied to generate additional images within each category, thus expanding the dataset. After augmentation, a total of 2000 images were randomly selected from each category, with 400 images per category. The dataset was then split into training, validation, and testing sets, as shown in Table 5.

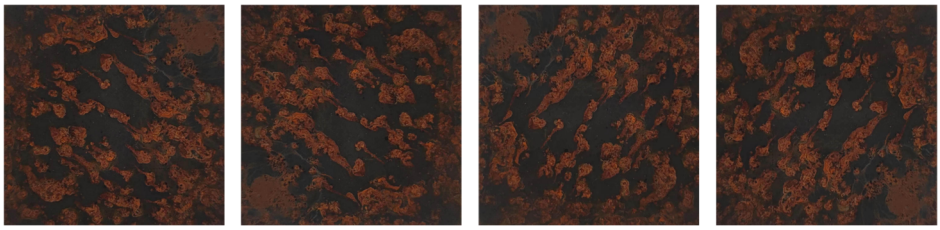

Figure 7. Data enhancement.

Table 5. Dataset partitioning for classification training.

Dataset	Level 0	Level 1	Level 2	Level 3	Level 4	Total
Training Set	320	320	320	320	320	1600
Validation Set	40	40	40	40	40	200
Testing Set	40	40	40	40	40	200

3.2. Inception Module Lightweight

The Inception module's lightweight design significantly reduces computational complexity and model size, thereby enhancing computational efficiency and real-time performance. This is achieved through several key optimizations: reducing convolutional kernel sizes, incorporating depthwise separable convolutions, and employing parameter-sharing techniques. These strategies enable substantial reductions in computational demands while preserving robust feature extraction capabilities [40–43]. As a result, the model becomes more suitable for deployment in devices with limited computational resources. Larger convolutional kernels, such as 5×5 or 7×7, incur disproportionately higher memory usage and computational costs. For example, a 5×5 convolution operation requires 2.78 times more computation than a 3×3 convolution with an equal number of filters. Since Inception networks are fully convolutional, each weight corresponds to a multiplication for each activation. Therefore, any reduction in computational cost directly translates to a decrease in the number of parameters. Through appropriate decomposition, more disentangled parameters can be obtained, which accelerates training speed and improves overall model efficiency [48–50]. The modified Inception module structure is illustrated in Figures 8 and 9. These modules will be directly applied in the subsequent Inception v3 model.

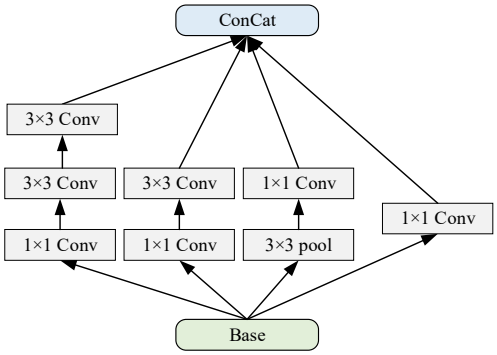

Figure 8. Replacement of 5 × 5 convolution with 3 × 3 convolutions.

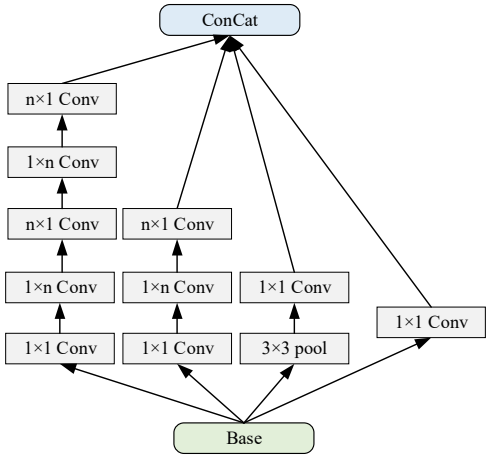

Figure 9. Replacement of high-dimensional convolutions with 1 × n and n × 1 convolutions.

3.3. Inception v3

The 3 × 3 convolution module, as shown in Figure 8, is utilized as the initial Inception module. The 1 × n module, depicted in Figure 9 with n set to 7, is applied to the intermediate Inception layers. Convolutions through this module effectively simulate a 17 × 17 convolution operation, as illustrated in Figure 10. Additionally, a high-dimensional separable Inception module is implemented, as shown in Figure 11, before the final output classification layer. This module expands feature dimensions to generate high-dimensional sparse features, thereby increasing the number of features. The specific structure of the Inception v3 model is detailed in Table 6.

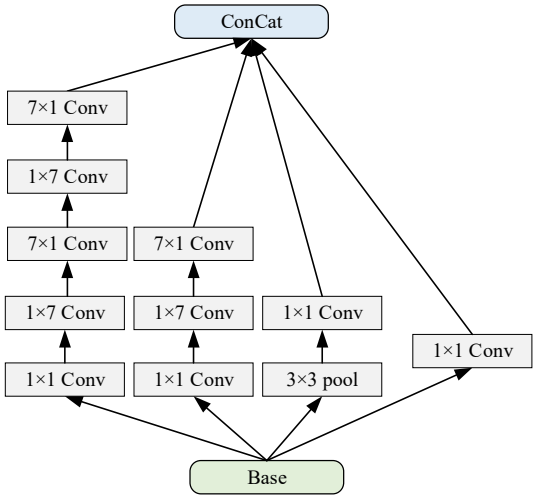

Figure 10. Replacing high-dimensional convolutions with 1 × 7 and 7 × 1 convolutions.

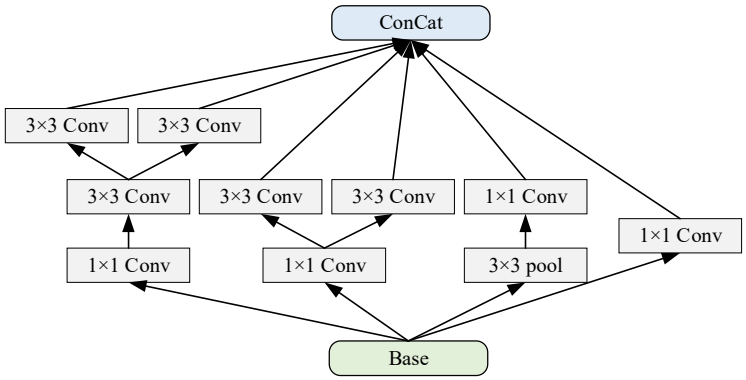

Figure 11. High-dimensional separable Inception module.

Table 6. Inception v3 architecture.

Type	Receptive Field/Stride	Input Size
conv	$3 \times 3/2$	$299 \times 299 \times 3$
conv	$3 \times 3/1$	$149 \times 149 \times 32$
conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
pool	$3 \times 3/2$	$147 \times 147 \times 64$
conv	$3 \times 3/1$	$73 \times 73 \times 64$
conv	$3 \times 3/2$	$71 \times 71 \times 80$
conv	$3 \times 3/1$	$53 \times 53 \times 192$
$3 \times$ Inception	Figure 8	$35 \times 35 \times 288$
$5 \times$ Inception	Figure 10	$17 \times 17 \times 768$
$2 \times$ Inception	Figure 11	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

4. Experiments and Results

4.1. Training and Finetuning

Although the model has undergone lightweight improvements, issues such as a high number of training parameters and significant data memory usage still persist. To ensure smooth model training, computing devices with higher configurations that meet the training requirements were selected. The software and hardware configurations used for model training are outlined in Table 7.

Table 7. Inception v3 training environment.

Component		Title 3
Hardware	CPU	Intel(R) Core(TM) i7-10700 CPU @290 GHz (Intel Corporation, Santa Clara, CA, USA)
	GPU	NVIDIA GeForce RTX 3060 (NVIDIA Corporation, Santa Clara, CA, USA)
	RAM	DDR4 16 GB (Samsung Electronics, Suwon, Republic of Korea)
	Cuda	Cuda 10.0
Software	IDE	PyCharm Community Edition 2017.2.2
	Programming Language	Python 3.8
	Framework	PyTorch 1.2.0

4.2. Results and Analysis

4.2.1. Accuracy Analysis

Accuracy (Acc), also known as the accuracy rate, is a crucial metric for evaluating the performance of a neural network. It measures the proportion of correctly predicted samples to the total number of predicted samples, regardless of whether the samples are

positive or negative. If only positive samples are considered in the model, the negative samples can be omitted from the numerator in the formula. Accuracy is one of the most commonly used metrics in neural network evaluations, with values ranging from 0 to 1. A value of 0 indicates that the model has classified all samples incorrectly, while a value of 1 signifies that all samples have been classified correctly. The formula for accuracy is given by Equation (2):

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

In the formula, *TP* represents the number of true positive samples (correctly classified positive samples); *TN* represents the number of true negative samples (correctly classified negative samples); *FP* denotes the number of false positive samples (incorrectly classified positive samples); and *FN* indicates the number of false negative samples (incorrectly classified negative samples).

To demonstrate the superior performance of the lightweight Inception v3 model for Q235 steel corrosion classification, comparative experiments were conducted using the same dataset to train AlexNet, ResNet, and VggNet as control models. All models were trained with consistent hyperparameter settings. The classification performance of each model was evaluated based on accuracy (Acc), as shown in Figure 12. It is evident that all four classification models were well trained, with accuracies exceeding 85% after 200 iterations. Among them, AlexNet exhibited the lowest classification performance, with an accuracy of only 87%. Both ResNet and VggNet showed comparable results, achieving accuracies of over 90%. The Inception v3 model demonstrated the highest performance for corrosion classification tasks, with an accuracy of 95.5%. Therefore, it can be concluded that the lightweight Inception v3 model is the most suitable for classifying the corrosion levels of steel.

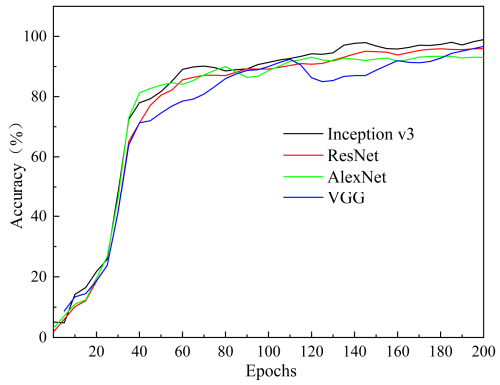

Figure 12. Classification accuracy of various models.

4.2.2. Error Analysis

A confusion matrix is a table used in machine learning and statistics to evaluate the performance of a classification model. It is commonly used to assess the quality of a classifier by providing a visual representation of classification results through matrix data. Its basic structure is illustrated in Table 8.

Table 8. Confusion matrix.

	Predicted Positive	**Predicted Positive**
Actual Positive	True Positive (*TP*)	False Negative (*FN*)
Actual Negative	False Positive (*FP*)	True Negative (*TN*)

The classification results of the lightweight Inception v3 network for a total of 200 test samples were exported into a confusion matrix, as shown in Figure 13. The

confusion matrix indicates that the model accurately classifies corrosion levels overall, with only a few misclassified images. Notably, for images with a corrosion level of 0, all classifications were correct. This high accuracy is due to the fact that level 0 images have not undergone corrosion, resulting in smooth, uniformly colored surfaces with fewer and more distinct features.

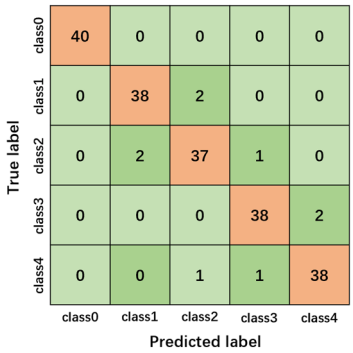

Figure 13. Confusion matrix for Inception v3.

The remaining misclassified test samples across the four corrosion levels were selected and analyzed to investigate the reasons behind these classification errors. Both the original images and the corresponding classification weights were examined, as shown in Figure 14. For Level 1, two test samples—sample "a" and its rotated version—were misclassified as Level 2. The primary reason was that sample "a" had minor surface scratches before the experiment, which, after corrosion, resulted in a yellowed appearance resembling patchy corrosion. This visual similarity likely led the model to classify it as Level 2. Among the 40 test samples for Level 2, three samples—b, c, and d—were misclassified. Sample "b" was incorrectly predicted as Level 1 with an 81.4% probability, possibly because it was captured at an intermediate stage between pitting and patchy corrosion, making it difficult to differentiate, even for human evaluators. Sample "c" was also misclassified as Level 1 due to its extensive corrosion coverage, which blended with the original surface color, preventing the model from effectively capturing corrosion edge features. Sample "d" was erroneously classified as Level 3, likely because the model perceived the corrosion as more severe or misinterpreted the darker corrosion color as a more advanced stage.

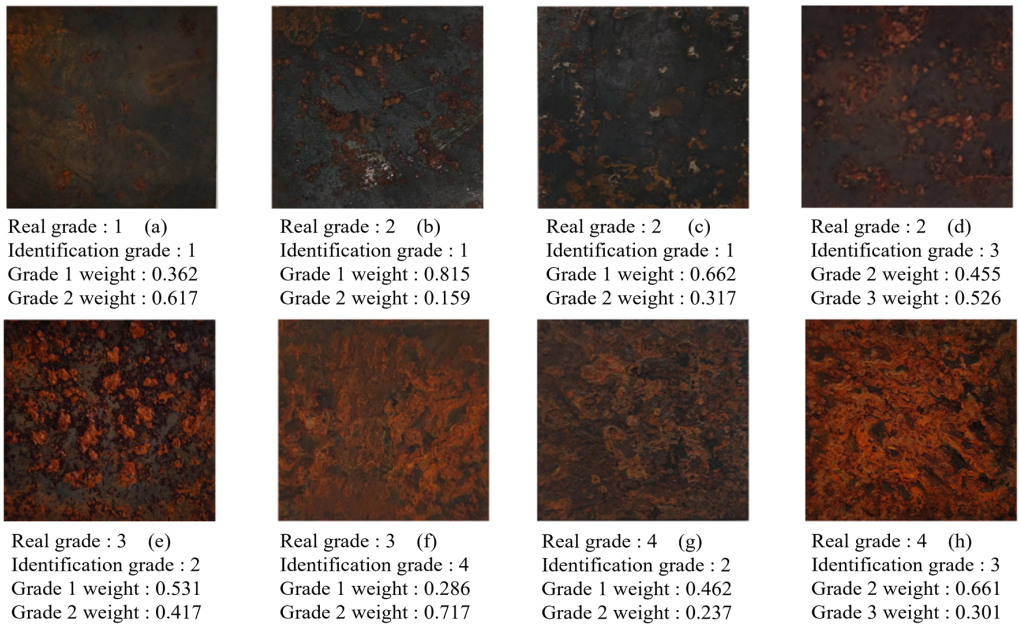

Figure 14. Classification weights for misclassified samples.

For Levels 3 and 4, four samples were misclassified. Sample "e" was misclassified due to pre-cleaning before imaging and poor lighting conditions during capture, which hindered the model's ability to extract corrosion features accurately. Sample "f" was predicted with over a 70% probability as the highest corrosion level (Level 4), while sample "h" was classified as Level 3 with a 66% probability. The challenge in distinguishing these samples arose because, as corrosion advanced, it accumulated and thickened, leading to minimal apparent changes that complicated classification. Sample "g" was misclassified as Level 2, likely because the corrosion had progressed to a stage where peeling occurred, darkening the surface. This visual shift caused the model to misinterpret the darkened regions as non-corroded areas, leading to an incorrect classification.

Overall, the errors encountered by the model are within a controllable range. Most of the misclassified samples result from issues during dataset creation, such as specimen wear, ambiguous classification, and poor lighting conditions during image capture. This underscores the importance of constructing a high-quality dataset before model training. For future classification model research, it is crucial to ensure that the dataset is clearly categorized and features are distinct. A well-constructed dataset is essential for training an effective neural network model.

5. Conclusions

In conclusion, this study demonstrates the effectiveness of a machine vision-based approach for corrosion level classification in Q235 steel, offering a robust tool for lifecycle management and maintenance optimization of steel structures. By integrating accelerated salt spray tests with advanced deep learning techniques, we successfully generated a comprehensive dataset of corrosion progression and achieved a classification accuracy exceeding 95% using the Inception v3 model. This approach not only outperforms traditional methods and other architectures such as AlexNet, ResNet, and VggNet but also provides a scalable and efficient framework for real-world applications. The results highlight the potential of this method to enhance the sustainability, safety, and cost-effectiveness of steel infrastructure through proactive maintenance planning and data-driven decision-making.

While machine vision detection provides significant advantages over traditional inspection methods, large-scale implementation faces challenges such as deployment costs, adaptation to varying environmental conditions, and ensuring reliability across diverse corrosion patterns. Future research should focus on improving model generalization and developing cost-effective deployment strategies to enhance practical applicability. By addressing these challenges, machine vision-based corrosion detection can contribute to more efficient and intelligent maintenance strategies for steel structures, particularly in high-risk environments.

Author Contributions: Conceptualization, X.H.; methodology, X.H.; software, X.H.; validation, S.H.; formal analysis, Z.D.; investigation, Z.D.; resources, L.C.; data curation, J.H.; writing—original draft preparation, J.H.; writing—review and editing, S.H.; visualization, W.C.; supervision, W.C.; project administration, W.C.; funding acquisition, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Special research and development plan of China Construction Seventh Engineering Division Co., Ltd. (Grant No. CSCEC7b-2023-Z-19).

Data Availability Statement: The original contributions presented in this study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Zhen Duan, Xinghong Huang, and Shaojin Hao are working for the company CSCEC 7th Division International Engineering Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Khayatazad, M.; Honhon, M.; De Waele, W. Detection of corrosion on steel structures using an artificial neural network. *Struct. Infrastruct. Eng.* **2023**, *19*, 1860–1871. [CrossRef]
2. Khayatazad, M.; De Pue, L.; De Waele, W. Detection of corrosion on steel structures using automated image processing. *Dev. Built Environ.* **2020**, *3*, 12. [CrossRef]
3. Han, Q.H.; Liu, X.; Xu, J. Detection and Location of Steel Structure Surface Cracks Based on Unmanned Aerial Vehicle Images. *J. Build. Eng.* **2022**, *50*, 14. [CrossRef]
4. Imran, M.M.H.; Jamaludin, S.; Ayob, A.F.M.; Ali, A.; Ahmad, S.; Akhbar, M.F.A.; Suhrab, M.I.R.; Zainal, N.; Norzeli, S.M.; Mohamed, S.B. Application of Artificial Intelligence in Marine Corrosion Prediction and Detection. *J. Mar. Sci. Eng.* **2023**, *11*, 25. [CrossRef]
5. Han, Q.H.; Zhao, N.; Xu, J. Recognition and location of steel structure surface corrosion based on unmanned aerial vehicle images. *J. Civ. Struct. Health Monit.* **2021**, *11*, 1375–1392. [CrossRef]
6. Han, S.X.; Li, B.D.; Li, W.; Zhang, Y.; Liu, P.Y. Intelligent analysis of corrosion characteristics of steel pipe piles of offshore construction wharfs based on computer vision. *Heliyon* **2024**, *10*, 14. [CrossRef]
7. Xu, Z.C.; Cai, B.Y.; Yan, L.C.; Pang, X.L.; Gao, K.W. Statistical analysis of metastable pitting behavior of 2024 aluminum alloy based on deep learning. *Corros. Sci.* **2024**, *233*, 14. [CrossRef]
8. Yang, L.Y.; Huang, X.B.; Ren, Y.C.; Han, Q.; Huang, Y.C. Steel plate surface defect classification technology based on image enhancement and combination feature extraction. *Eng. Comput.* **2023**, *40*, 1305–1329. [CrossRef]
9. Yu, Q.F.; Han, Y.D.; Lin, W.G.; Gao, X.J. Detection and Analysis of Corrosion on Coated Metal Surfaces Using Enhanced YOLO v5 Algorithm for Anti-Corrosion Performance Evaluation. *J. Mar. Sci. Eng.* **2024**, *12*, 19. [CrossRef]
10. Liu, T.T.; Kang, K.; Zhang, F.; Ni, J.L.; Wang, T.Y. A Corrosion Detection Algorithm Via the Random Forest Model. In Proceedings of the 17th International Conference on Optical Communications and Networks (ICOCN), Zhuhai, China, 16–19 November 2018.
11. Atha, D.J.; Jahanshahi, M.R. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct. Health Monit.* **2018**, *17*, 1110–1128. [CrossRef]
12. Chen, S.K.; Huang, I.F.; Chen, P.H. Applying fully convolutional neural networks for corrosion semantic segmentation for steel bridges: The use of U-Net. In Proceedings of the 10th International Conference on Bridge Maintenance, Safety and Management (IABMAS), Sapporo, Japan, 11–18 April 2021; pp. 341–346.
13. Xu, J.; Gui, C.Q.; Han, Q.H. Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Comput.-Aided Civil Infrastruct. Eng.* **2020**, *35*, 1160–1174. [CrossRef]
14. Barakbayeva, T.; Demirci, F.M. Fully automatic CNN design with inception and ResNet blocks. *Neural Comput. Appl.* **2023**, *35*, 1569–1580. [CrossRef]
15. Li, X.; Hao, T.X.; Li, F.; Zhao, L.Z.; Wang, Z.H. Faster R-CNN-LSTM Construction Site Unsafe Behavior Recognition Model. *Appl. Sci.* **2023**, *13*, 16. [CrossRef]
16. Mannem, K.R.; Mengiste, E.; Hasan, S.; de Soto, B.G.; Sacks, R. Smart audio signal classification for tracking of construction tasks. *Autom. Constr.* **2024**, *165*, 13. [CrossRef]
17. Meng, Q.H.; Zhu, S.Y. Construction Activity Classification Based on Vibration Monitoring Data: A Supervised Deep-Learning Approach with Time Series RandAugment. *J. Constr. Eng. Manag.* **2022**, *148*, 11. [CrossRef]
18. Zou, G.F.; Fu, G.X.; Gao, M.L.; Shen, J.; Yin, L.J.; Ben, X.Y. A novel construction method of convolutional neural network model based on data-driven. *Multimed. Tools Appl.* **2019**, *78*, 6969–6987. [CrossRef]
19. Jiang, X.; Qi, H.; Qiang, X.H.; Zhao, B.S.; Dong, H. A Convolutional Neural Network-Based Corrosion Damage Determination Method for Localized Random Pitting Steel Columns. *Appl. Sci.* **2023**, *13*, 21. [CrossRef]
20. Li, Z.J.; Shao, P.; Zhao, M.H.; Yan, K.; Liu, G.X.; Wan, L.; Xu, X.L.; Li, K.L. Optimized deep learning for steel bridge bolt corrosion detection and classification. *J. Constr. Steel Res.* **2024**, *215*, 13. [CrossRef]
21. Cui, Y.; Lei, D.F. Optimizing Internet of Things-Based Intelligent Transportation System's Information Acquisition Using Deep Learning. *IEEE Access* **2023**, *11*, 11804–11810. [CrossRef]
22. Panmatharit, A.; Jiraraksopakun, Y.; Siripanichgorn, A.; Siricharoen, P. Bolt Looseness Identification using Faster R-CNN and Grid Mask Augmentation. In Proceedings of the 14th Annual Summit and Conference of the Asia-Pacific-Signal-and-Information-Processing-Association (APSIPA ASC), Chiang Mai, Thailand, 7–10 November 2022; pp. 1632–1637.
23. Zhao, D.B.; Li, H. Forward Vehicle Detection Based on Deep Convolution Neural Network. In Proceedings of the 3rd International Conference on Advances in Materials, Machinery, Electronics (AMME), Wuhan, China, 19–20 January 2019.

24. Wang, H.; Cao, G.M.; Liu, J.J.; Wu, S.W.; Li, Z.F.; Liu, Z.Y. Development and application of automatic identification methods based on deep learning for oxide scale structures of iron and steel materials. *J. Mater. Sci.* **2023**, *58*, 17675–17690. [CrossRef]
25. Zhang, M.Y.; Wang, W.L. Deep learning-based extraction and quantification of features in XCT images of steel corrosion in concrete. *Case Stud. Constr. Mater.* **2024**, *20*, 18. [CrossRef]
26. Zhou, G.; Sun, H.Y. Defect Detection Method for Steel Based on Semantic Segmentation. In Proceedings of the IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 975–979.
27. Jafari, F.; Dorafshan, S.; Kaabouch, N. Segmentation of fatigue cracks in ancillary steel structures using deep learning convolutional neural networks. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA, 28–30 June 2023; pp. 872–877.
28. Santos, R.; Ribeiro, D.; Lopes, P.; Cabral, R.; Calcada, R. Detection of exposed steel rebars based on deep-learning techniques and unmanned aerial vehicles. *Autom. Constr.* **2022**, *139*, 17. [CrossRef]
29. Zhu, Z.L.; Liang, Y.L. Prediction of Residual Stress of Carburized Steel Based on Machine Learning. *Appl. Sci.* **2020**, *10*, 16. [CrossRef]
30. Bao, N.X.; Zhang, T.; Huang, R.Z.; Biswal, S.; Su, J.Y.; Wang, Y. A Deep Transfer Learning Network for Structural Condition Identification with Limited Real-World Training Data. *Struct. Control Health Monit.* **2023**, *2023*, 8899806. [CrossRef]
31. Ali, R.; Cha, Y.J. Subsurface damage detection of a steel bridge using deep learning and uncooled micro-bolometer. *Constr. Build. Mater.* **2019**, *226*, 376–387. [CrossRef]
32. Chi, Y.L.; Cai, C.Z.; Ren, J.H.; Xue, Y.F.; Zhang, N. Damage location diagnosis of frame structure based on wavelet denoising and convolution neural network implanted with Inception module and LSTM. *Struct. Health Monit.* **2024**, *23*, 57–76. [CrossRef]
33. Liu, Z.; Wang, X.S.; Chen, X. Inception Dual Network for steel strip defect detection. In Proceedings of the 16th IEEE International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 9–11 May 2019; pp. 409–414.
34. Ren, J.H.; Cai, C.Z.; Chi, Y.L.; Xue, Y.F. Integrated Damage Location Diagnosis of Frame Structure Based on Convolutional Neural Network with Inception Module. *Sensors* **2023**, *23*, 418. [CrossRef]
35. Bouguettaya, A.; Zarzour, H. CNN-based hot-rolled steel strip surface defects classification: A comparative study between different pre-trained CNN models. *Int. J. Adv. Manuf. Technol.* **2024**, *132*, 399–419. [CrossRef]
36. Ivo, R.F.; Rodrigues, D.D.; Bezerra, G.M.; Freitas, F.N.C.; de Abreu, H.F.G.; Rebouc, P.P. Non-grain oriented electrical steel photomicrograph classification using transfer learning. *J. Mater. Res. Technol-JMRT* **2020**, *9*, 8580–8591. [CrossRef]
37. Sundarrajan, K.; Rajendran, B.K. Explainable efficient and optimized feature fusion network for surface defect detection. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 1–18. [CrossRef]
38. Ahmed, S.; Cho, S.H. Hand Gesture Recognition Using an IR-UWB Radar with an Inception Module-Based Classifier. *Sensors* **2020**, *20*, 18. [CrossRef] [PubMed]
39. Baixo, S.; Ribeiro, T.; Lopes, G.; Ribeiro, A.F. 3D Face Recognition using Inception Networks for Service Robots. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Santa Maria da Feira, Portugal, 29–30 April 2022; pp. 47–52.
40. Chavan, S.; Nair, L.; Nimbalkar, N.; Solkar, S. Karyotyping of human chromosomes in metaphase images using faster R-CNN and inception models. *Int. J. Imaging Syst. Technol.* **2024**, *34*, 24. [CrossRef]
41. Huang, Q.Q.; Cai, Q.; Chen, Y.; Huang, J.B. Single image dehazing network based on inception module. In Proceedings of the 3rd International Conference on Electronics and Communication; Network and Computer Technology (ECNCT), Xiamen, China, 3–5 December 2021.
42. Liu, F.L.; Qin, D.B.; Yang, S.; Du, R.Y. WS-ICNN algorithm for robust adaptive beamforming. *Wirel. Netw.* **2024**, *30*, 5201–5210. [CrossRef]
43. Wang, J.K.; He, X.H.; Faming, S.; Lu, G.L.; Cong, H.; Jiang, Q.Y. A Real-Time Bridge Crack Detection Method Based on an Improved Inception-Resnet-v2 Structure. *IEEE Access* **2021**, *9*, 93209–93223. [CrossRef]
44. Yong, L.; Bo, Z. An Intrusion Detection Model Based on Multi-scale CNN. In Proceedings of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 214–218.
45. Zhai, Z.L.; Feng, S.; Yao, L.Y.; Li, P.H. Retinal vessel image segmentation algorithm based on encoder-decoder structure. *Multimed. Tools Appl.* **2022**, *81*, 33361–33373. [CrossRef]
46. Zhang, C.Z.; Zhang, Y.Z.; Huang, Z.Y.; Lv, C.; Hao, D.; Liang, C.; Deng, C.H.; Chen, J.R. Real-Time Optimization of Energy Management Strategy for Fuel Cell Vehicles Using Inflated 3D Inception Long Short-Term Memory Network-Based Speed Prediction. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1190–1199. [CrossRef]
47. Zheng, G.Y.; Han, G.H.; Soomro, N.Q. An Inception Module CNN Classifiers Fusion Method on Pulmonary Nodule Diagnosis by Signs. *Tsinghua Sci. Technol.* **2020**, *25*, 368–383. [CrossRef]
48. Chandankhede, C.; Sachdeo, R. Offline MODI script character recognition using deep learning techniques. *Multimed. Tools Appl.* **2023**, *82*, 21045–21056. [CrossRef]

49. Jeyakumar, J.P.; Jude, A.; Henry, A.G.P.; Hemanth, J. Comparative Analysis of Melanoma Classification Using Deep Learning Techniques on Dermoscopy Images. *Electronics* **2022**, *11*, 11. [CrossRef]
50. Liu, K.; Yu, S.T.; Liu, S.D. An Improved InceptionV3 Network for Obscured Ship Classification in Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *13*, 4738–4747. [CrossRef]
51. Mputu, H.S.; Abdel-Mawgood, A.; Shimada, A.; Sayed, M.S. Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers. *IEEE Access* **2024**, *12*, 8283–8295. [CrossRef]
52. Pravin, S.C.; Rohith, G.; Kiruthika, V.; Manikandan, E.; Methelesh, S.; Manoj, A. Underwater Animal Identification and Classification Using a Hybrid Classical-Quantum Algorithm. *IEEE Access* **2023**, *11*, 141902–141914. [CrossRef]
53. Singh, N.; Tripathi, P. An efficient model for detecting real-time facemask based on different Classification Algorithms. *Multimed. Tools Appl.* **2023**, *83*, 55175–55198. [CrossRef]
54. Sriraam, N.; Srinivasulu, A. Performance evaluation of convolution neural network models for detection of abnormal and ventricular ectopic beat cardiac episodes. *Multimed. Tools Appl.* **2024**, *83*, 65149–65188. [CrossRef]
55. Thentu, S.; Cordeiro, R.; Park, Y.; Karimian, N. ECG biometric using 2D Deep Convolutional Neural Network. In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 10–12 January 2021.
56. Abd Almisreb, A.; Jamil, N.; Din, N.M. Utilizing AlexNet Deep Transfer Learning for Ear Recognition. In Proceedings of the 4th International Conference on Information Retrieval and Knowledge Management (CAMP), Sabah, Malaysia, 26–28 March 2018; pp. 8–12.
57. Revathi, M.; Raghuraman, G. Kidney Stone Detection from CT Images Using ALEXNET and Hybrid ALEXNET-RF Models. *J. Circuits Syst. Comput.* **2024**, *33*, 16. [CrossRef]
58. Toliupa, S.; Tereikovskyi, I.; Tereikovskyi, O.; Tereikovska, L.; Nakonechnyi, V.; Akov, Y.K. Keyboard Dynamic Analysis by Alexnet Type Neural Network. In Proceedings of the 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv, Ukraine, 25–29 February 2020; pp. 416–420.
59. Xiao, L.S.; Yan, Q.; Deng, S.Y. Scene Classification with Improved AlexNet Model. In Proceedings of the 12th International Conference on Intelligent Systems and Knowledge Engineering (IEEE ISKE), Nanjing, China, 24–26 November 2017.
60. Yang, M.Y.; Xie, K.; Li, T.; Ye, Y.H.; Yang, Z.P. Color Constancy Using AlexNet Convolutional Neural Network. In Proceedings of the 6th International Workshop on Pattern Recognition (IWPR), Beijing, China, 25–27 June 2021.
61. Zhang, Y.M.; Chang, F.L.; Li, N.J.; Liu, H.B.; Gai, Z.D. Modified AlexNet for Dense Crowd Counting. In Proceedings of the 2nd International Conference on Computer Engineering, Information Science and Internet Technology (CII), Sanya, China, 11–12 November 2017; pp. 351–357.
62. Ge, Y.F.; Liu, G.; Tang, H.M.; Zhao, B.B.; Xiong, C.R. Comparative analysis of five convolutional neural networks for landslide susceptibility assessment. *Bull. Eng. Geol. Environ.* **2023**, *82*, 26. [CrossRef]
63. Lu, G.Y.; Cao, B.; Zhu, X.D.; Lin, Z.S.; Bai, D.X.; Tao, C.Y.; Li, Y.N. Identification of rock mass discontinuity from 3D point clouds using improved fuzzy C-means and convolutional neural network. *Bull. Eng. Geol. Environ.* **2024**, *83*, 18. [CrossRef]
64. Wang, Z.J.; Zhao, W.L.; Du, W.H.; Li, N.P.; Wang, J.Y. Data-driven fault diagnosis method based on the conversion of erosion operation signals into images and convolutional neural network. Process Saf. *Environ. Protect.* **2021**, *149*, 591–601. [CrossRef]
65. Xie, G.B.; Shi, B.H.; Su, Y.X.; Wu, X.R.; Zhou, G.; Shi, J.F. Research on the Vanishing Point Detection Method Based on an Improved Lightweight AlexNet Network for Narrow Waterway Scenarios. *J. Mar. Sci. Eng.* **2024**, *12*, 17. [CrossRef]
66. Yuan, X.; Ren, J.W.; Cheng, G.F.; Xu, J. Toward Alleviating the Data Sparsity Problem of Deep Learning Based Underwater Target Classification. In Proceedings of the OCEANS Conference, San Diego, CA, USA, 20–23 September 2021.
67. Yang, S.; Xue, L.Z.; Hong, X.; Zeng, X.Y. A Lightweight Network Model Based on an Attention Mechanism for Ship-Radiated Noise Classification. *J. Mar. Sci. Eng.* **2023**, *11*, 17. [CrossRef]
68. Zecchetto, S.; Zanchetta, A. Validation of high resolution SAR winds fields obtained by Deep Learning. In Proceedings of the IEEE International Workshop on Metrology for the Sea Learning to Measure Sea Health Parameters (MetroSea), Milazzo, Italy, 3–5 October 2022; pp. 501–505.
69. Zhang, P.Y.; Jiang, W.L.; Zheng, Y.F.; Zhang, S.Q.; Zhang, S.; Liu, S.Y. Hydraulic-Pump Fault-Diagnosis Method Based on Mean Spectrogram Bar Graph of Voiceprint and ResNet-50 Model Transfer. *J. Mar. Sci. Eng.* **2023**, *11*, 31. [CrossRef]
70. Dai, Z.Z.; Liang, H.; Duan, T. Small-Sample Sonar Image Classification Based on Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 22. [CrossRef]
71. Luo, X.W.; Zhang, M.H.; Liu, T.; Huang, M.; Xu, X.G. An Underwater Acoustic Target Recognition Method Based on Spectrograms with Different Resolutions. *J. Mar. Sci. Eng.* **2021**, *9*, 20. [CrossRef]
72. Xie, J.L.; Shi, W.F.; Xue, T.; Liu, Y.H. High-Resistance Connection Fault Diagnosis in Ship Electric Propulsion System Using Res-CBDNN. J. *Mar. Sci. Eng.* **2024**, *12*, 17. [CrossRef]
73. Chen, Y.N.; Tian, Z.G.; Wei, H.T.; Dong, S.H. Reliability analysis of corroded pipes using MFL signals and Residual Neural Networks. Process Saf. *Environ. Protect.* **2024**, *184*, 1131–1142. [CrossRef]

74. Lin, K.S.; Zhao, Y.C.; Wang, L.A.; Shi, W.J.; Cui, F.F.; Zhou, T. MSWNet: A visual deep machine learning method adopting transfer learning based upon ResNet 50 for municipal solid waste sorting. *Front. Env. Sci. Eng.* **2023**, *17*, 12. [CrossRef]
75. Peng, L.G.; Zhang, J.C.; Lu, S.Q.; Li, Y.Q.; Du, G.F. One-dimensional residual convolutional neural network and percussion-based method for pipeline leakage and water deposit detection. Process Saf. *Environ. Protect.* **2023**, *177*, 1142–1153. [CrossRef]
76. Chen, L.S.; Peng, H.Y.; Yang, D.D.; Wang, T.Z. An attachment recognition method based on semi-supervised video segmentation for tidal stream turbines. *Ocean Eng.* **2024**, *293*, 17. [CrossRef]
77. Dong, X.R.; Li, J.S.; Li, B.; Jin, Y.Q.; Miao, S.F. Marine Oil Spill Detection from Low-Quality SAR Remote Sensing Images. *J. Mar. Sci. Eng.* **2023**, *11*, 20. [CrossRef]
78. Kim, K.; Kim, J. Semantic Segmentation of Marine Radar Images using Convolutional Neural Networks. In Proceedings of the OCEANS—Marseille Conference, Marseille, France, 17–20 June 2019.
79. O'Byrne, M.; Pakrashi, V.; Schoefs, F.; Ghosh, B. Semantic Segmentation of Underwater Imagery Using Deep Networks Trained on Synthetic Imagery. *J. Mar. Sci. Eng.* **2018**, *6*, 15. [CrossRef]
80. Yu, F.; He, B.; Li, K.G.; Yan, T.H.; Shen, Y.; Wang, Q.; Wu, M.H. Side-scan sonar images segmentation for AUV with recurrent residual convolutional neural network module and self-guidance module. *Appl. Ocean Res.* **2021**, *113*, 14. [CrossRef]
81. Nunes, A.; Gaspar, A.R.; Matos, A. Comparative Study of Semantic Segmentation Methods in Harbour Infrastructures. In Proceedings of the OCEANS Conference, Limerick, Ireland, 5–8 June 2023.
82. Nunes, A.; Matos, A. Improving Semantic Segmentation Performance in Underwater Images. *J. Mar. Sci. Eng.* **2023**, *11*, 26. [CrossRef]
83. Zhou, H.X.; Tao, G.X.; Nie, Y.X.; Yan, X.Y.; Sun, J. Outdoor thermal environment on road and its influencing factors in hot, humid weather: A case study in Xuzhou City, China. *Build. Environ.* **2022**, *207*, 15. [CrossRef]
84. *ASTM B117-19*; Standard Practice for Operating Salt Spray (Fog) Apparatus. ASTM International: West Conshohocken, PA, USA, 2019.
85. *ASTM G1-03*; Standard Practice for Preparing, Cleaning, and Evaluating Corrosion Test Specimens. ASTM International: West Conshohocken, PA, USA, 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.