

## Article

# Rapid Detection of Corrosion on Steel Bridges Using a Deep Learning Method

Kemal Hacıfendioğlu <sup>1,2</sup> , Korhan Özgan <sup>1</sup>, Sara Mostofi <sup>1</sup>  and Ahmet Can Altunışık <sup>1,2,3,\*</sup><sup>1</sup> Department of Civil Engineering, Karadeniz Technical University, Trabzon 61080, Türkiye; kemalhaciefendioğlu@ktu.edu.tr (K.H.); saramostofi.4e@gmail.com (S.M.)<sup>2</sup> Earthquake and Structural Health Monitoring Research Center, Karadeniz Technical University, Trabzon 61080, Türkiye<sup>3</sup> Dynamic Academy Software, Construction Ind. Trade Limited Company, Trabzon 61081, Türkiye

\* Correspondence: ahmetcan@ktu.edu.tr

## Abstract

The corrosion of steel bridge structures, caused by anthropogenic or natural sources, can significantly impact the safety and integrity of these structures. The quick and accurate detection of corrosion is crucial for identifying areas that require strengthening and repair. This study proposes an image-based detection method, referred to here as CAM-K-OD, for identifying areas of corrosion in steel bridges using photographs captured with any device. The proposed method fuses a gradient-based class activation mechanism map (Grad-CAM) with K-means clustering applied to convolutional neural network (CNN) features and object localization modules to delineate corrosion zones. The detection pipeline leverages deep convolutional features, grouped through clustering, to extract attention-based visual patterns and identify defective areas. Labeled and masked image datasets were used to train and test the system, and its evaluation was conducted using IoU (the Intersection over Union metric used to measure the accuracy of object detection and segmentation algorithms). This method was tested alongside U-Net segmentation and EfficientDet detection models, which were used as benchmarks. The findings indicate that CAM-K-OD exhibits superior localization fidelity and robustness under varying imaging conditions. This model enables efficient and reliable corrosion identification, supporting real-world bridge maintenance by reducing inspection times and improving the targeting of repair efforts.

**Keywords:** deep learning; corrosion; Grad-CAM; K-means clustering; steel bridge

Academic Editor: Pedro Couto

Received: 4 March 2025

Revised: 16 April 2025

Accepted: 17 April 2025

Published: 10 November 2025

**Citation:** Hacıfendioğlu, K.; Özgan, K.; Mostofi, S.; Altunışık, A.C. Rapid Detection of Corrosion on Steel Bridges Using a Deep Learning Method. *Appl. Sci.* **2025**, *15*, 11929. <https://doi.org/10.3390/app152211929>

**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

The corrosion of steel structures, especially bridges, represents a considerable economic and safety concern worldwide. Annually, substantial amounts of iron and steel become unusable due to corrosion, leading not only to economic losses related to the material itself but also negative impacts for investment efficiency and broader economic stability. Structures such as buildings, bridges, power transmission lines, and marine vessels exposed to atmospheric conditions can rapidly deteriorate, significantly shortening their service life. Addressing this issue requires a thorough understanding of corrosion processes, effective inspection techniques, and appropriate structural designs that are aimed at the prevention and mitigation of corrosion.

Steel bridges, integral to transportation infrastructure, face ongoing threats due to corrosion, demanding their consistent monitoring and timely maintenance. Traditional

inspection methods, however, often prove costly, labor-intensive, and reliant on specialist expertise, making them inefficient and impractical. With recent technological advancements, particularly in artificial intelligence (AI), new automated inspection methods have emerged. AI systems that incorporate layered visual processing techniques such as convolutional neural networks (CNNs) have shown a strong performance in identifying and classifying visual patterns linked to damage. These methods are increasingly employed in structural health monitoring (SHM), leveraging high-quality visual data collected via devices such as smartphones and drones, thus facilitating rapid and accurate damage assessments.

In civil engineering, the evaluation of damage to buildings caused by various sources has been the subject of many studies. Conventional assessment techniques rely on physical measurements, modeling simulations, and detailed evaluations. While effective, such methods are labor-intensive, time-consuming, and expensive, limiting their practical use in continuous monitoring. Consequently, there has been a noticeable shift in the literature toward the adoption of automated damage detection methods such as deep learning-based frameworks [1–5]. It is essential to detect cracks in reinforced concrete structures quickly and effectively. Numerous studies have demonstrated the viability of using automated visual inspection methods that rely on trained deep neural networks and image analysis techniques to detect cracking and surface degradation with high accuracy [6–10].

This study introduces a novel corrosion detection technique, the CAM-K-OD model, which integrates gradient-weighted class activation maps (Grad-CAM), K-means clustering, and object detection through pre-trained CNN models. This integrated approach enables the precise and efficient identification of corrosion areas in steel bridge structures with minimal preprocessing requirements. The CAM-K-OD model's innovative combination of these methods addresses existing limitations by improving accuracy, reducing manual labor, and enhancing practical applicability. When evaluated against established methods like U-Net and EfficientDet, the proposed model demonstrates a superior detection performance, underscoring its ability to serve as a reliable tool for enhancing bridge inspection workflows and safety.

The enhancement in detection capability highlighted by this research holds significant implications for ensuring the structural reliability and long-term service life of steel bridges. The CAM-K-OD architecture offers a significant advantage through its effective corrosion identification mechanism, which operates with minimal dependence on preprocessing or manual labeling tasks. Its high precision and recall indicate its robust ability to distinguish between corroded and non-corroded areas, a crucial factor in the timely and efficient maintenance of bridges. Furthermore, its slightly higher average Intersection over Union (IoU) score compared to that of the U-Net and EfficientDet models suggests a more accurate alignment with ground truth detections, enhancing the reliability of its assessments.

## 2. Background and Related Work

### 2.1. Corrosion Detection in Structures

In recent years, there has been a notable surge in scholarly interest in the application of deep learning techniques for the detection of corrosion in structural materials. Nash et al. [11] developed a pixel-level segmentation model using a dataset of 225 images, employing Bayesian neural networks to generate uncertainty maps that assisted in assessing the reliability of each prediction. Recognizing the importance of labeled data in the training of deep learning models, Nash et al. [12] also introduced an online platform for image annotation to improve training efficiency and dataset volume. In another study, Atha and Jahanshahi [13] focused on the design of CNN architectures tailored to corrosion classification and identifying optimal input modalities across various corrosion types. Petricca et al. [14] compared deep learning-based classification with standard computer vision

techniques for rust identification, concluding that deep learning has a higher detection accuracy under realistic field conditions. Rahman et al. [15] introduced a deep learning framework for the semantic segmentation of corrosion regions that was integrated with an automated image labeling tool, enabling the rapid preparation of datasets and the improved classification of corrosion severity. Their approach used texture analysis and RGB-based classifier tuning to automate the annotation process. Forkan et al. [16] proposed CorrDetector, an automated approach for monitoring corrosion using image analysis, and reported significant improvements in the classification performance of this approach compared to previous models. Brandoli et al. [17] proposed a deep learning-based image analysis workflow for surface inspection, achieving a detection precision exceeding 93%, which helped mitigate the inconsistencies caused by inspector fatigue or insufficient training.

The object recognition and segmentation of visual data have been addressed using various deep learning frameworks. These include both rapid single-shot approaches and region-based models like You Only Look Once (YOLO) [18], faster R-CNN [19], SSD [20], and mask R-CNN [21]. While YOLO and SSD demonstrate a computational efficiency conducive to real-time video stream processing, their efficiency often comes with a trade-off in accuracy. Conversely, the mask R-CNN delivers high precision by performing instance segmentation at the pixel level, which makes it suitable for applications requiring detailed spatial mapping. However, creating a training dataset for these techniques is labor-intensive and time-consuming.

Recently, machine learning and vision-based approaches have been incorporated into the structural health monitoring (SHM) of steel bolted joints. Studies have developed vision-based techniques using region-based convolutional neural networks (R-CNN) or Faster R-CNN to monitor bolt looseness and detect multiple types of damage in civil structures [19,20]. However, the majority of the research in this field has concentrated on stainless steel bolts, while relatively little attention has been paid to the automated identification of corrosion-related damage in bolted connections.

The detection of incipient corrosion in bolted steel connections remains a critical yet underexplored challenge in SHM, thereby necessitating the development of refined, data-driven methodologies capable of addressing this need with greater accuracy and automation. In this context, instance segmentation architectures such as the advanced mask R-CNN model represent a technically viable solution. This model integrates object detection and segmentation into a single framework and is distinguished by its dual functionality, as it generates both bounding boxes and detailed instance-level segmentation masks, allowing for the precise identification of affected areas. This capacity to segment corrosion features down to the pixel level makes the model well suited for recognizing the onset of corrosion in bolted joints.

The novel CAM-K-OD object detection model presented in this study addresses a significant gap in the literature regarding corrosion detection in steel bridge structures. Existing methodologies often require considerable manual effort or do not offer sufficient accuracy. The methodological contribution of CAM-K-OD is particularly significant due to its multi-component architecture, which includes visual interpretability mechanisms, unsupervised pattern grouping, and object detection procedures, each derived from pre-trained CNNs. The model incorporates Grad-CAM-based saliency mapping [21], K-means clustering [22], and a deep object recognition engine. This combination allows for high-resolution corrosion localization with improved interpretability and accuracy. When benchmarked against existing models—U-Net, used for semantic segmentation [23], and EfficientDet, used for object recognition [24]—the CAM-K-OD model consistently demonstrates a higher performance.

The current literature reveals a growing trend in using deep learning techniques to detect structural damage, including cracks and corrosion. Despite significant progress, many challenges remain, particularly regarding dataset quality, image resolution, and the practical applicability of deep learning models. Existing corrosion detection methods, including the U-Net segmentation and EfficientDet object detection models, have had varying degrees of success but often require extensive manual interventions in terms of data preparation and labeling, which limits their efficiency and scalability.

To address these limitations, this work proposes the CAM-K-OD model, which will enhance the detection of corrosion on steel bridges through an innovative integration of deep learning and unsupervised segmentation techniques.

## 2.2. Proposed Framework

To provide a comprehensive understanding of the proposed CAM-K-OD detection framework, this section delineates the foundational components upon which the model is constructed. Core elements of the architecture include CNNs, which are extensively applied in SHM due to their capability to automatically identify damage; transfer learning, which serves to enhance the model's generalization and convergence efficiency under limited data regimes; Grad-CAM, which facilitates the interpretability of the model outputs through localized visual attribution; and K-means clustering, complemented by the elbow method, which is used for segmenting feature maps efficiently. Additionally, well-established object detection and segmentation models, namely U-Net and EfficientDet, are considered as comparative baselines for a performance evaluation. The methods used in this study, CNN, transfer learning, Grad-CAM, K-means clustering (K-means), and object detection–segmentation, are presented in Figure 1.

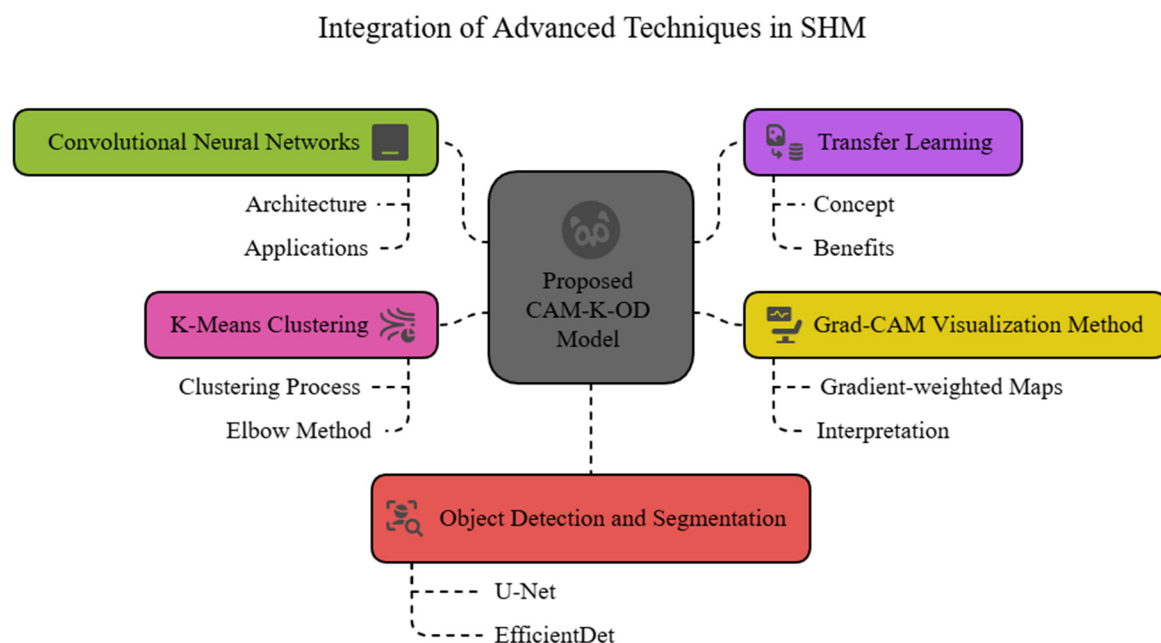


**Figure 1.** Conceptual diagram of the CAM-K-OD method for locating corrosion.

### 2.2.1. Transfer Learning

Transfer learning [25] is a deep learning strategy that allows the use of trained parameters from one problem domain to be the starting point for solving a new but related task. This approach is frequently employed in cases where data limitations and model complexity make training from scratch impractical. Instead of designing and training large models with extensive data requirements, transfer learning provides a faster and more

accurate alternative by reusing learned features. This is particularly beneficial for visual recognition tasks where labeled datasets are limited or difficult to collect. When applied effectively, transfer learning can yield high accuracy with significantly less training time and a reduced data overhead.

Instead of developing models specific to each new image classification problem, this technique repurposes the structure and weights from previously validated networks. These pre-trained networks serve as a baseline for further customization and fine-tuning. Since deep learning models generally require large datasets to avoid overfitting and ensure reliable convergence, transfer learning addresses this issue by reducing the need for vast amounts of data. By starting from previously learned weights rather than initializing a network with random values, the training process becomes more stable and converges faster [26,27]. This technique efficiently trains CNN topologies using minimal data.

### 2.2.2. ResNet50 Model

ResNet50 [28] is a deep convolutional neural network architecture composed of approximately 23 million trainable parameters distributed across 152 distinct layers. This model is specifically crafted to tackle the complexity of training very deep networks and is frequently employed in image classification tasks. Its core concept hinges on adding the original input (“residual”) to the result of two subsequent convolution operations. This method helps prevent the training difficulties that arise as networks grow deeper and more complex. Figures 2 and 3 provide a visual explanation of the ResNet architecture and the operation of its residual block.

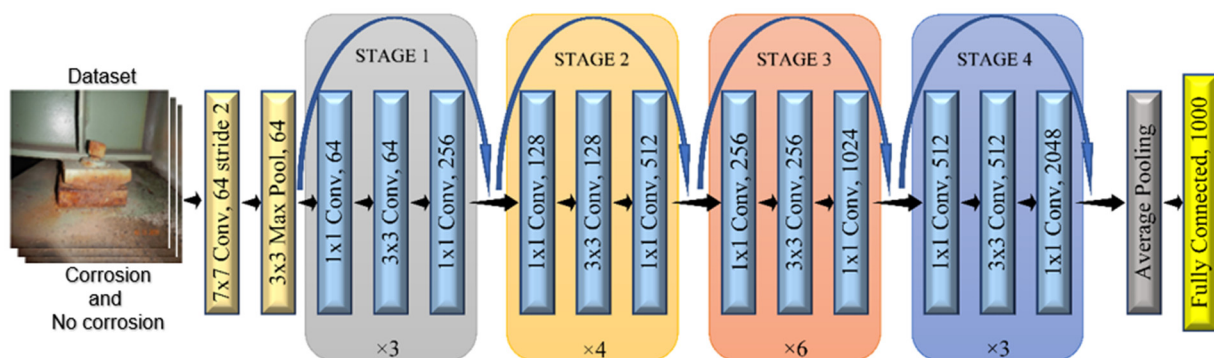


Figure 2. The network architecture of ResNet50.

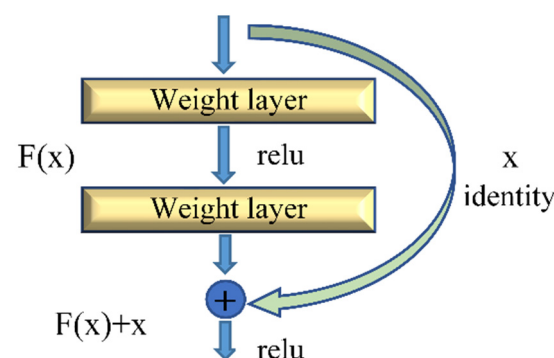


Figure 3. The residual block of a deep residual network.

A key element of ResNet50 is its bottleneck design, which reduces its computational load without compromising performance. The fundamental relationship is captured by Equation (1):

$$H(x) = F(x) + x \quad (1)$$

Here,  $H(x)$  represents the final output, while  $F(x)$  denotes the function learned from the input  $x$ . Because it is impractical to learn  $H(x)$  directly when the depth increases, the network employs “skipping links”. Consequently,  $F(x)$  is recognized as the “residual,” representing the part added back to the input to generate the final output.

### 2.2.3. Grad-CAM Method

Grad-CAM [21] provides a clear way of visualizing the parts of an image that influence a specific class score. It works by tracing gradients from the output of the last convolutional layer back to the input and then creating a spatial map that identifies which pixels matter most for that class. The resulting map, denoted as  $L_{Grad-CAM}^c$ , has dimensions  $u \times v$ , which align with the width and height of the last convolutional layer’s dimensions, respectively. The process involves computing the partial derivatives of the score  $y^c$  (prior to softmax) with respect to each feature map  $A^k$ . These gradients are then averaged over the entire spatial extent of the feature map, yielding a scalar weight  $W_k^c$  for each channel  $k$  through

$$W_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (2)$$

where  $Z$  represents the total count of the spatial positions over which the gradients are aggregated ( $Z = u \times v$ ). Through this averaging process, the magnitude of  $W_k^c$  captures the degree of importance feature map  $k$  holds with respect to class  $c$ . Subsequently, a weighted combination of the forward activation maps  $A^k$  is summed using these weights, and a Rectified Linear Unit (ReLU) function is applied:

$$L_{Grad-CAM}^c = \text{ReLU} \left( \sum_k W_k^c A^k \right) \quad (3)$$

This produces a coarse heatmap that mirrors the resolution of the convolutional feature maps. Applying ReLU effectively filters out negative contributions, ensuring the map highlights areas that positively affect  $y^c$ . Conversely, signals from non-contributing or negatively contributing regions are inherently reduced. Without this activation function, the localization map can become overly broad, lowering its ability to distinguish the actual region of interest.

### 2.2.4. K-Means Clustering

The segmentation of Grad-CAM-generated images commences with the determination of an optimal  $K$  value via the elbow method. Figure 4 illustrates a typical elbow graph. This method computes the within-cluster sum of squares (WCSS) as a function of the number of clusters and identifies the point at which additional clusters yield diminishing returns, thereby achieving compact cluster.

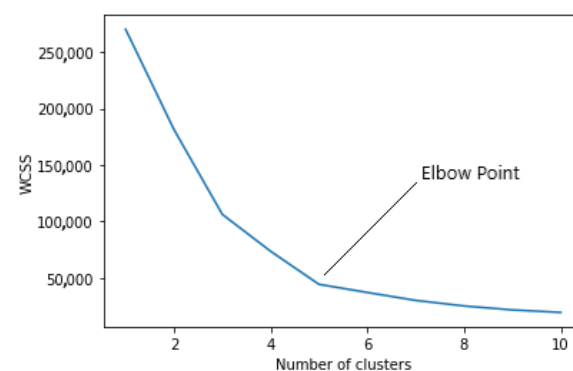


**Figure 4.** Elbow graph.

This unsupervised, center-based algorithm groups image pixels or features into clusters by minimizing the variance within clusters while maximizing the variance between them. K-means begins by randomly selecting  $K$  initial centroids. Each data point is then assigned to the nearest centroid, and the centroids are updated by averaging the points assigned to each cluster. This iterative procedure continues until the centroids stabilize, meaning that subsequent iterations produce minimal changes in their positions. The objective of this algorithm, as encapsulated in Equation (4) below [29], is to minimize the sum of the squared distances between each data point and its respective cluster centroid, thereby ensuring minimal intra-cluster variance while maximizing inter-cluster separability:

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (4)$$

where  $J$  is the overall error to be minimized,  $n_j$  is the number of data points in cluster  $j$ ,  $x_i^{(j)}$  represents the  $i$ th data point in cluster  $j$ , and  $c_j$  denotes the centroid of cluster  $j$ . The term  $\left\| x_i^{(j)} - c_j \right\|^2$  represents the squared Euclidean distance between the data point and its centroid.

Upon finalization of the clustering assignment and centroid stabilization, the resulting image is subsequently processed for segmentation by converting it into grayscale before transforming it into a binary format (referred to as bi-level). In this representation, the image contains only two pixel values corresponding to two possible intensity levels. These binary images store pixel information as either 0 or 1, facilitating clear separation between the background and key foreground objects. The OpenCV library, widely recognized for its efficiency in image manipulation tasks, was employed for executing image transformations.

### 2.3. Benchmark Models

#### 2.3.1. U-Net Segmentation Model

Semantic segmentation is a key technique used in computer vision that assigns class labels to each pixel in an image. This technique is crucial for accurately defining the boundaries of various objects or regions and has been used for identifying different anatomical structures in medical imaging scenarios [30]. The progression of CNN-based algorithms has significantly propelled advancements in this area. One of the most effective CNN-based models used for this task is U-Net, originally introduced by Ronneberger et al. [23]. The U-Net architecture employed for corrosion detection, as illustrated in Figure 5, is described as follows:

**Encoder:** This component utilizes a traditional layer sequence to derive features from multiple image resolutions. Each encoding block contains a pair of  $3 \times 3$  convolutions with a unit stride, followed by ReLU activations. To progressively reduce spatial dimensions and increase the representational depth, a  $2 \times 2$  max-pooling operation with a stride of 2 is applied after each convolutional block.

**Decoder:** The decoder mirrors the encoder's structure but uses transposed convolution layers ( $2 \times 2$ ) for up-sampling, reversing the down-sampling effect. These layers are augmented with feature maps from corresponding encoder blocks through concatenation, ensuring the detailed feature integration necessary for precise localization.

**Output:** The final stage consists of a  $1 \times 1$  convolution which is used to compress the feature maps into a single-channel output, resulting in a  $128 \times 128$  single-channel image indicating the presence of corrosion.

This robust architecture combines the strengths of deep convolutional networks with the precise localization capabilities offered by transposed convolutions and skip connections, making it highly effective for semantic segmentation tasks. U-Net's adaptability and

efficiency have led to its application beyond biomedical segmentation, and it has become a versatile tool used in various image processing domains.

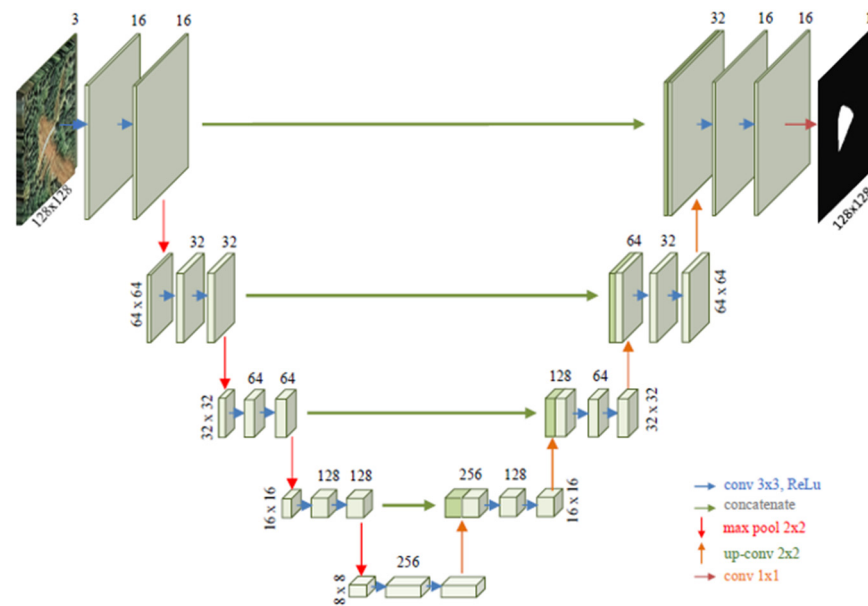


**Figure 5.** U-Net architecture.

### 2.3.2. EfficientDet Object Detection Model

EfficientDet [24] models are specifically designed for accurate object detection while maintaining efficiency in terms of computation. These models offer a balance between precision and resource constraints, making them suitable for various platforms. The model's architecture is made up three primary components: the backbone, the feature network, and the final class/box network.

The backbone is responsible for generating feature maps from the input image. It acts as the foundation for subsequent processing.

The feature network collects outputs from different backbone layers and integrates them using a Bi-directional Feature Pyramid Network (BiFPN). This design allows information to be propagated both from higher to lower layers and vice versa, improving the model's ability to detect objects at different scales. The BiFPN improves upon standard pyramid networks by incorporating learnable feature weighting and multiple fusion pathways.

The classification/box prediction network is the final stage. It processes the fused features to generate object class labels and their corresponding bounding boxes. This network is lightweight but effective, using compact layers to maintain speed.

Efficiency gains are achieved by using an EfficientNet backbone instead of other, less efficient options like ResNets or AmoebaNet. Furthermore, depthwise separable convolutions replace regular convolutions, reducing computational complexity. Figure 6 provides an overview of the architecture of EfficientDet.

EfficientDet improves the balance between detection accuracy and computational efficiency through a compound scaling technique. This method scales image resolution, network depth, and width together, using a heuristic-guided strategy to maintain an optimal balance for the model. The EfficientDet series includes models D0 through D7, with each successive model having a higher computational cost and offering greater accuracy.

When evaluated on the COCO benchmark dataset [31], the highest-tier model, EfficientDet-D7, achieved a mean average precision (mAP) of 52.2. This result represents a 1.5-point improvement over the previous state-of-the-art model, despite using only

one-quarter of the parameters and 9.4 times less computation. Additionally, these models are faster than other detectors implemented on both GPU and CPU.

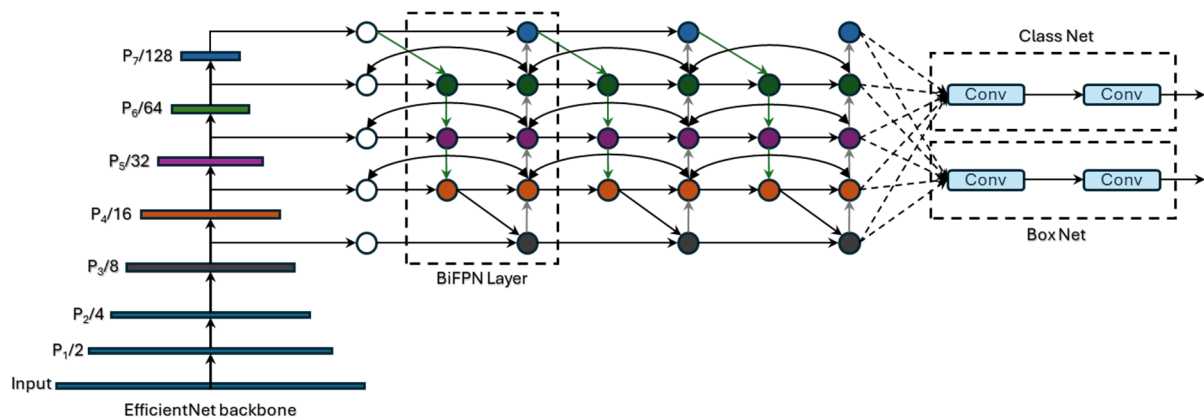


**Figure 6.** EfficientDet architecture.

The EfficientDet framework comprises a scalable and resource-conscious suite of object detectors capable of delivering competitive accuracy levels while significantly minimizing computational overhead. These models are adaptable to a wide range of resource constraints, making them suitable for real-world applications across various platforms.

Overall, the combination of a ResNet50 backbone, Grad-CAM visualization, and K-means clustering were chosen for their proven ability to extract robust deep features, provide clear interpretability of key regions, and effectively segment corroded areas. These components together ensure both the high classification accuracy and precise localization critical for corrosion detection on steel bridges.

### 3. CAM-K-OD Model

Building on the concepts outlined in Section 2, the proposed CAM-K-OD model integrates ResNet50, Grad-CAM, and K-means clustering to achieve high classification accuracy and precise corrosion localization. This framework combines deep learning and unsupervised segmentation methods into an automated framework for corrosion detection on steel bridge structures. The input data comprised 880 images from bridge inspection reports, with an equal number of corroded and non-corroded cases. To maintain consistency during training, all images were resized to a fixed resolution of  $224 \times 224$  pixels and labeled using a binary format that distinguished between corrosion conditions.

Feature extraction was performed using a pretrained ResNet50 network [28], where the initial layers were frozen to retain foundational feature learning. The model architecture was further improved by adding global average pooling, dropout, and fully connected layers to boost classification accuracy. Grad-CAM visualization was applied to the trained model to generate heatmaps, allowing for the identification of the regions most relevant to corrosion detection. These heatmaps enhanced both interpretability and localization accuracy.

The heatmaps were passed through a K-means clustering process to separate corrosion from background regions. The number of clusters was chosen using the elbow method, ensuring optimal grouping. After clustering, the output was converted to grayscale and binarized, which made it possible to separate corrosion areas clearly. Bounding boxes were then applied: red for predicted areas and green for ground truth masks.

The performance of the model was assessed using multiple evaluation metrics: accuracy, precision, recall, F1-score, area under the receiver operating characteristic curve (AUC), and IoU. The IoU metric compared predicted bounding boxes to the ground truth,

offering a direct measure of the accuracy of localization. The results showed that CAM-K-OD exceeded the performance of the benchmark models. Figure 7 outlines the processing flow of the proposed detection methodology.

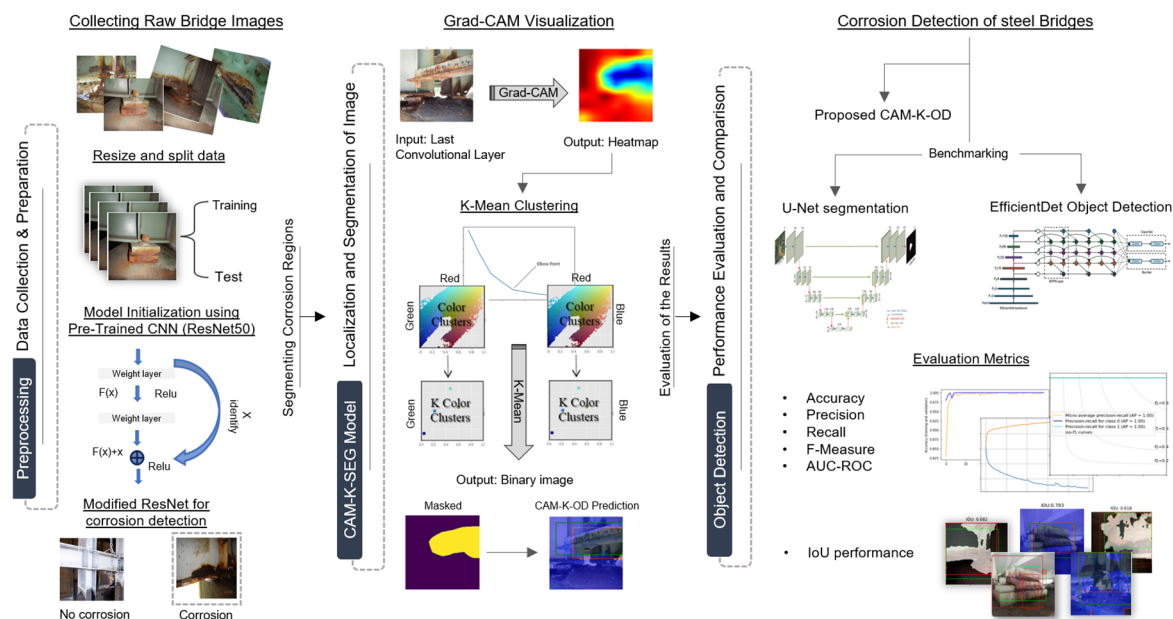


**Figure 7.** Workflow of the proposed corrosion detection methodology.

In this framework, a CNN was initialized with pretrained weights from ImageNet. The top classification layers of the base models were removed and replaced with custom-designed components suitable for corrosion detection. The model was enhanced through several modifications, including zero-padding for spatial integrity,  $3 \times 3$  convolutional filters, and a high-dimensional convolutional layer consisting of 1024 output maps. To reduce overfitting and compress feature representations, the network included a global average pooling layer followed by a dropout layer set at a 0.5 rate. The final classification layer employed a Softmax activation function applied to a dense layer for probabilistic output. The detailed structure of this adapted CNN is depicted in Figure 8.

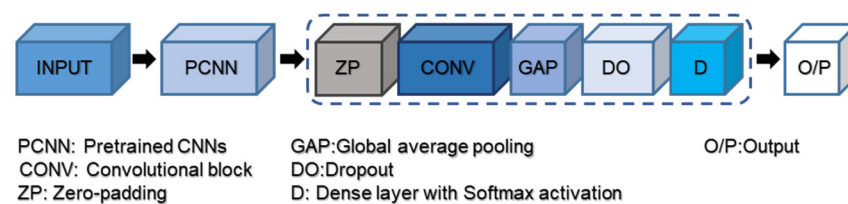


**Figure 8.** Structural layout of the CNN.

Two models were employed to classify the dataset, one built using a CNN trained from scratch, and the other based on a transfer learning approach. With the CNN, a success rate (IoU performance) of 80% ( $\pm 2$ ) was attained, while the transfer learning model had a success rate of 95% ( $\pm 1$ ). These outcomes are significant because they demonstrate the value of the transfer learning strategy. The structure of the transfer learning model applied in this analysis is presented in Figure 9.

Pretrained convolutional models were adapted by freezing their lower layers, preserving their previously learned weights to retain their general feature extraction capabilities. To enhance the capacity of the model for domain-specific adaptation, the conventional fully connected classification head was supplanted by a global average pooling operation applied to the final convolutional layer. This operation computes the mean value of each

feature map, summarizing spatial features into a representative value. The output from this stage was then processed through a dense layer, followed by a dropout mechanism set at 0.5 to reduce overfitting, and then passed through a second dense layer before its final classification using the Softmax function.

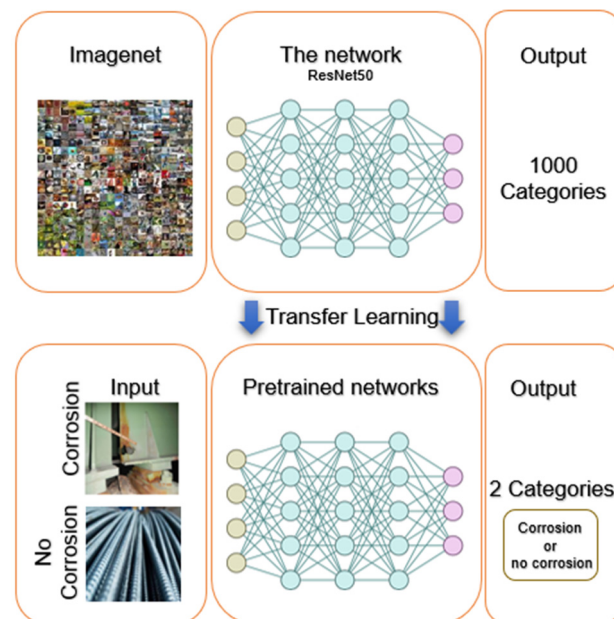


**Figure 9.** Structural configuration of the applied transfer learning model.

For training, categorical cross-entropy was used to handle multiple output classes. The Adam optimizer was employed, with a learning rate of 0.0001, a batch size of 32, and a training cycle of 30 epochs. Among the pretrained networks tested, ResNet50 achieved the highest performance, outperforming other configurations in terms of its classification reliability. All training processes were GPU-accelerated via the CUDA and cuDNN libraries, using the TensorFlow backend in conjunction with the Keras API. The training and validation of the models were conducted on a Windows 11 computer with 32 GB of RAM and an NVIDIA Quadro RTX 4000 GPU.

The CAM-K-OD framework integrates the Grad-CAM interpretability mechanism with convolutional feature representations derived from the ResNet50 architecture, specifically from the “conv5\_block3\_3\_conv” layer. This layer was found to produce the most relevant features for distinguishing between corroded and non-corroded surfaces. Its high-level feature representations enabled a better spatial identification of damaged regions. When integrated with the Grad-CAM visualization algorithm, this layer facilitates the generation of highly localized class activation maps that accurately highlight regions of interest associated with corrosion.

Grad-CAM’s performance has been assessed using the corrosion class test set and ResNet50 model that performed best in detecting corrosion spots. The resulting visualizations, shown in Figure 10, highlight the effectiveness of Grad-CAM in locating corroded regions within various samples. This approach provided clear, interpretable heatmaps that allowed for a spatial understanding of the model’s decision-making process. Among all visualization methods tested, Grad-CAM produced the most accurate localization results, supporting its integration within the CAM-K-OD framework for practical corrosion assessments.

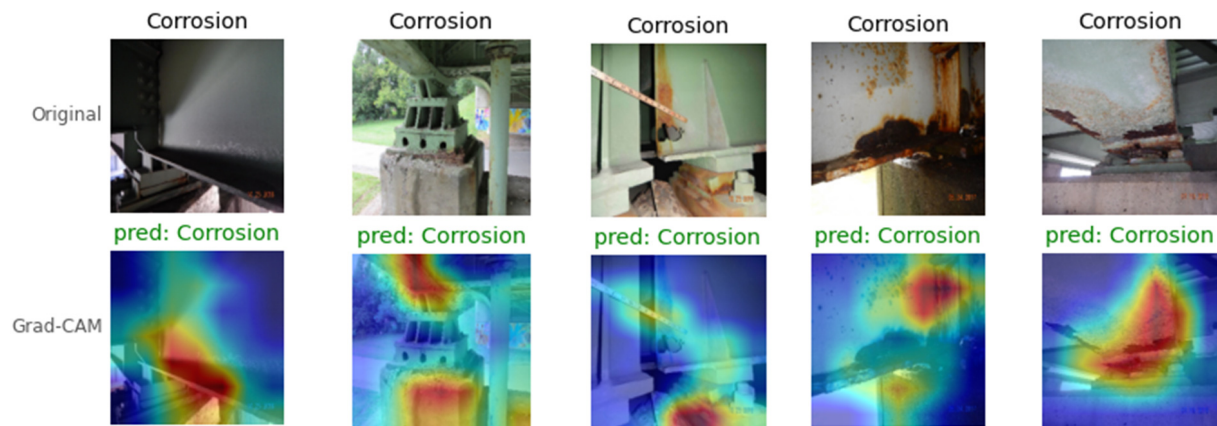


**Figure 10.** Examples of corrosion region localization using Grad-CAM.

The CAM-K-OD pipeline can be broken down into a series of ordered steps, as illustrated in Figure 11. The procedure began with (a) capturing the input image. In step (b), a heatmap was constructed using Grad-CAM applied to the ResNet50 network. The resultant heatmap was superimposed onto the original image in step (c). In the next phase, (d), clustering was executed through the K-means algorithm to extract color groups that represented key regions. Step (e) included conversion to grayscale and the removal of minor artifacts through filtering. The image was subsequently transformed into a binary format in step (f). Finally, in step (g) the segmented image was produced and presented as the final result.

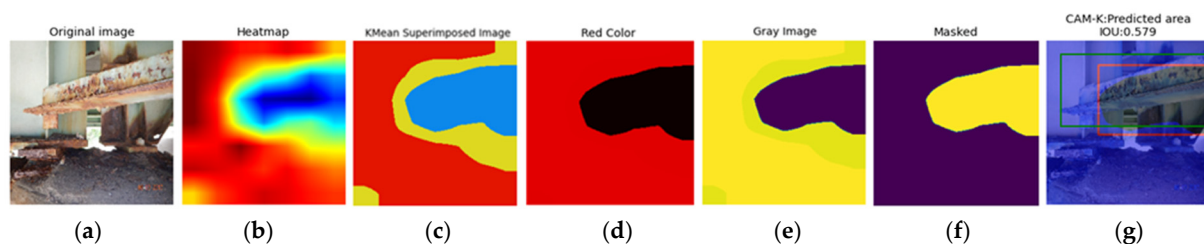


**Figure 11.** The CAM-K-OD corrosion detection pipeline.

### 3.1. Dataset

In this study, the Virginia Department of Transportation (VDOT)'s bridge inspection reports provided information about steel bridge inspections for structural health monitoring [32]. According to the corrosion status requirements outlined in the Bridge Inspectors' Reference Manual and the American Association of State Highway and Transportation Officials (AASHTO), the data are semantically tagged (BIRM). The original dataset included four corrosion levels: fully intact (no corrosion), minor surface degradation, moderate corrosion, and extensive corrosion. These four categories were consolidated into a single corrosion label to meet the objective of the study, which was to identify only the corroded areas of bridges. The dataset consists of 440 photos with corrosion and 440 images without corrosion. Some samples are shown in Figure 12. Each of the two classes' images (with and without corrosion) were divided into 396 training photos and 44 test images at random. For training and testing, all images were resized to  $224 \times 224$ .



**Figure 12.** Dataset examples of the (a) corrosion and (b) no-corrosion classes.

### 3.2. Evaluation Metrics

The effectiveness of the models was assessed using a set of performance indicators, including classification accuracy, AUC, precision, recall, and F1-score, which are defined in Equations (5)–(8).

$$\text{Accuracy} = \frac{\text{Correct classification}}{\text{Total classification}} \quad (5)$$

$$\text{Precision} = \frac{\text{Correctly classified actual positives}}{\text{Everything classified as positive}} \quad (6)$$

$$\text{Recall} = \frac{\text{Correctly classified actual positives}}{\text{All actual positives}} \quad (7)$$

$$\text{F1 - score} = \frac{2(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (8)$$

The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) to evaluate the classification performance (corroded and non-corroded zones) of the models under different thresholds. A model with a curve closer to the top-left corner has a better classification capability. The AUC summarizes this curve into a single number: higher AUC values indicate stronger class separability. The FPR is calculated using the following equation:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (9)$$

## 4. Results

### 4.1. Evaluation of the CAM-K-OD Model

Figure 13 shows the graphs of the change in accuracy and loss over 50 epochs. The model reached high performance after approximately 10 epochs. At the end of 50 epochs, 100% accuracy and 0.0001 loss values are obtained.

The learning rate, momentum, and L2-decay hyperparameters of the ResNet50 pre-trained CNN model are  $1 \times 10^{-4}$ , 0.95, and  $1 \times 10^{-6}$ , respectively. Evaluation of the model on the corrosion-specific test dataset revealed a perfect classification performance across all

considered metrics, each achieving a value of 1.00. Using the combination of deep feature extraction using ResNet50 and interpretability via Grad-CAM visualization as the basis of the proposed framework proved highly effective in locating corrosion.

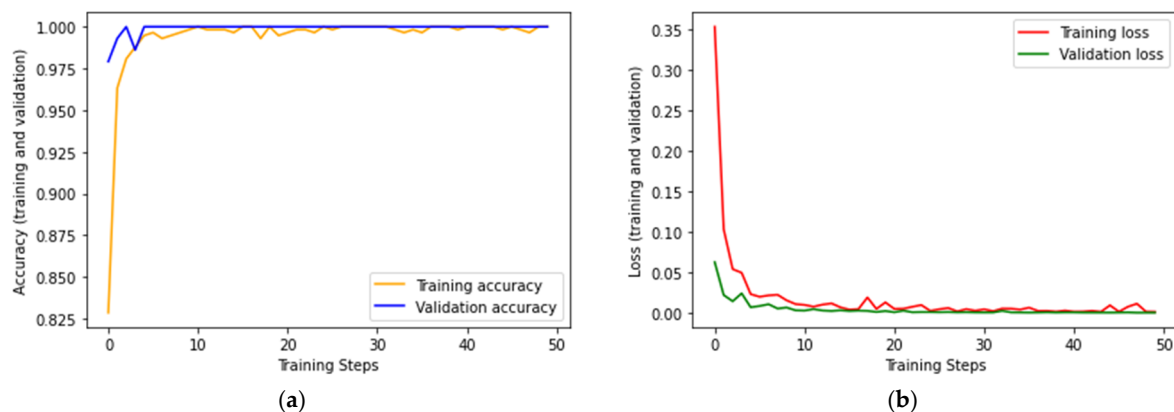


**Figure 13.** Training and validation accuracy (a) and loss (b) curves over 50 epochs.

In this context, precision refers to how many of the predicted corrosion instances were correctly identified. Recall captures the proportion of actual corrosion instances that were successfully detected. The F1-score provides a combined measure of both, offering a single value that accounts for both types of classification errors. As shown in Table 1, all three metrics, across both classes, achieve the maximum score of 1.00, indicating perfect classification. The macro and weighted averages, referred to here as the unweighted and proportional class means, also reached 1.00, and the overall accuracy was 100%. These values indicate that the model consistently and accurately classified both corrosion and non-corrosion images. Figure 14 complements these results with the ROC and precision–recall (PR) plots, both of which confirm the superior discriminative power of the classifier.

**Table 1.** Performance evaluation of Resnet-50.

|                         | Precision | Recall | F1-Score |
|-------------------------|-----------|--------|----------|
| Corrosion               | 1.00      | 1.00   | 1.00     |
| No corrosion            | 1.00      | 1.00   | 1.00     |
| Accuracy                |           |        | 1.00     |
| Unweighted class mean   | 1.00      | 1.00   | 1.00     |
| Proportional class mean | 1.00      | 1.00   | 1.00     |

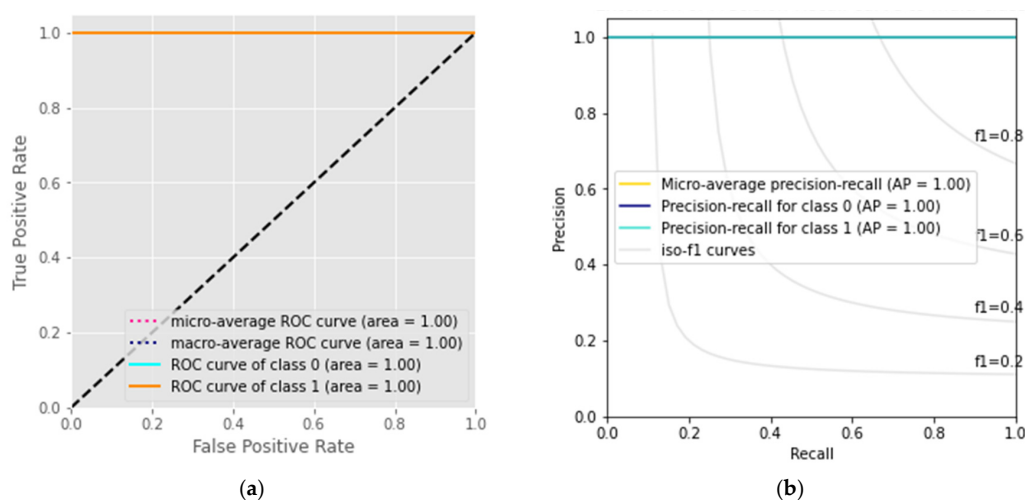


**Figure 14.** Extension of (a) ROC and (b) PR plots for ResNet50-based corrosion detection.

#### 4.2. Evaluation of the U-Net Segmentation

The performance of the U-Net model was evaluated by tracking both its training loss and accuracy across multiple iterations. These trends are plotted in Figure 15, demonstrating the learning progress of the model over time. The U-Net segmentation model achieved a loss value of 0.0728 and an accuracy value of 0.9684. These results indicate that U-Net successfully learned to distinguish and segment relevant regions in the corrosion images. Its steady improvement in accuracy and reduction in loss suggest that the model efficiently captured the spatial patterns necessary for accurate segmentation. Training convergence was observed after approximately 100 optimization iterations, at which point the performance metrics plateaued, indicating model stabilization. This suggests that the model was able to identify the relevant features in the images and perform segmentation tasks reliably.

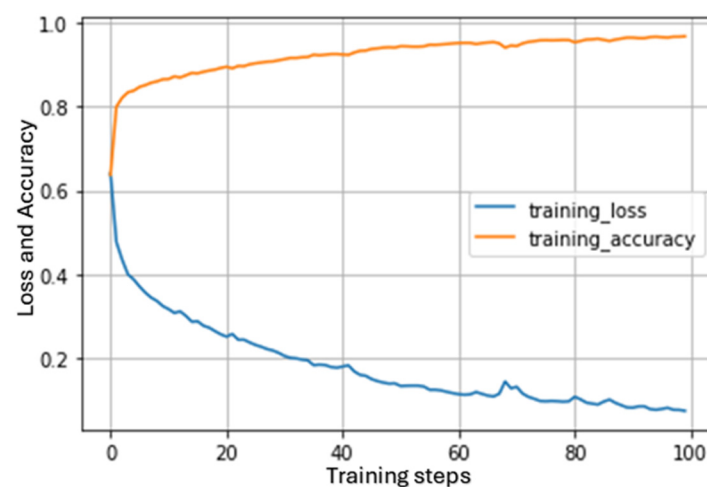


**Figure 15.** Learning progress of the U-Net segmentation model.

#### 4.3. Evaluation of the EfficientDet Object Detection Model

The evolution of the total training loss, classification loss, and regression loss over the course of the model's optimization is illustrated in Figure 16, providing a graphical depiction of the training dynamics of the EfficientDet architecture. The values obtained for the training loss curve, classification, and regression at the 5800th step were 0.0634, 0.0183, and 0.0322, respectively, indicating sufficient training. The analysis of the results revealed that the validation mean average precision (mAP) value was 43.53%.

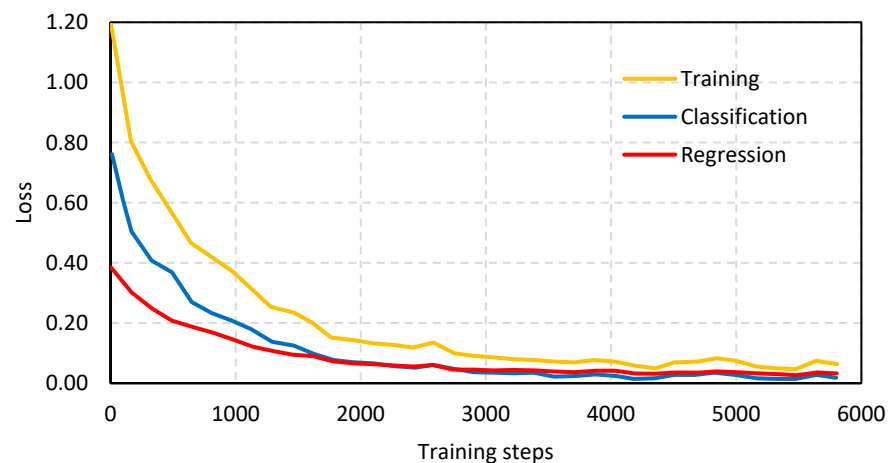


**Figure 16.** Training, classification, and regression loss curves.

In this section, the term “regression data” specifically refers to the bounding box regression component of the EfficientDet object detection model. Object detection tasks typically involve two primary objectives:

- Classification—determining what type of object is present (in this case, “corrosion” vs. “no corrosion”).
- Regression—predicting the precise coordinates (bounding box) that the object is localized to in the image.

In this model, the regression loss reflects the accuracy of predicted bounding box coordinates in detecting localized objects. EfficientDet simultaneously solves a classification task (identifying whether the object is “corrosion” or “no corrosion”) and a regression task, which involves predicting the spatial boundaries of the object. By minimizing the regression loss, the model refines its localization capability, ensuring that the bounding boxes tightly encompass the corroded areas. Thus, the regression data shown in this section demonstrate how well the model is learning to position bounding boxes around corrosion, separately from simply classifying whether corrosion is present.

#### 4.4. Comparisons

In this study, the CAM-K-OD model, incorporating a transfer learning-based ResNet50 CNN model, was used to locate and identify the corrosion in steel bridge structures that develops over time under external effects. The model identifies corrosion regions and localizes them using bounding boxes, which are then matched against the annotated ground truth to evaluate performance. The IoU, as a criterion for assessing the spatial alignment between predicted and actual object boundaries, was used as the primary evaluation standard. The IoU computes the ratio of the area of the intersection between the predicted bounding box and ground truth box ( $y_{true}$  and  $y_{pred}$ ) to the area of their union. Figure 17 provides an illustration of this process, showing how prediction overlaps are calculated relative to the ground truth.

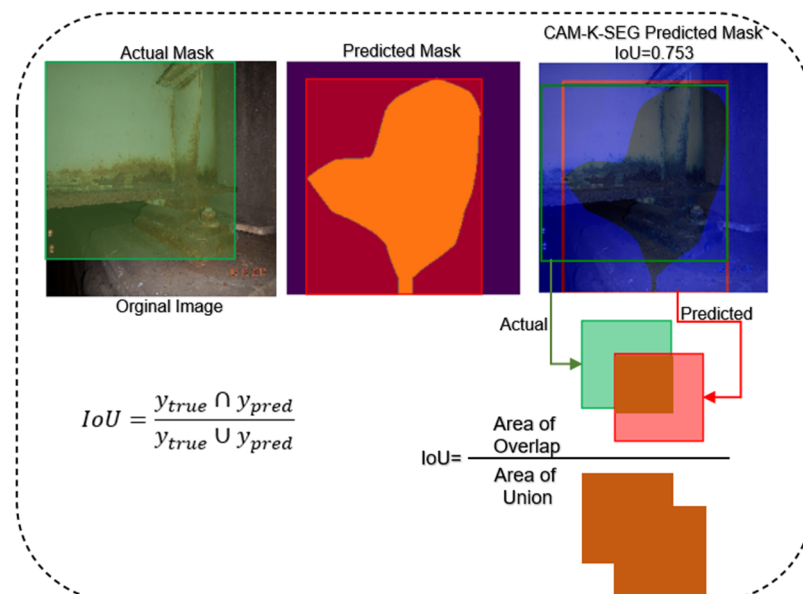


**Figure 17.** IoU calculation using predicted and reference bounding boxes.

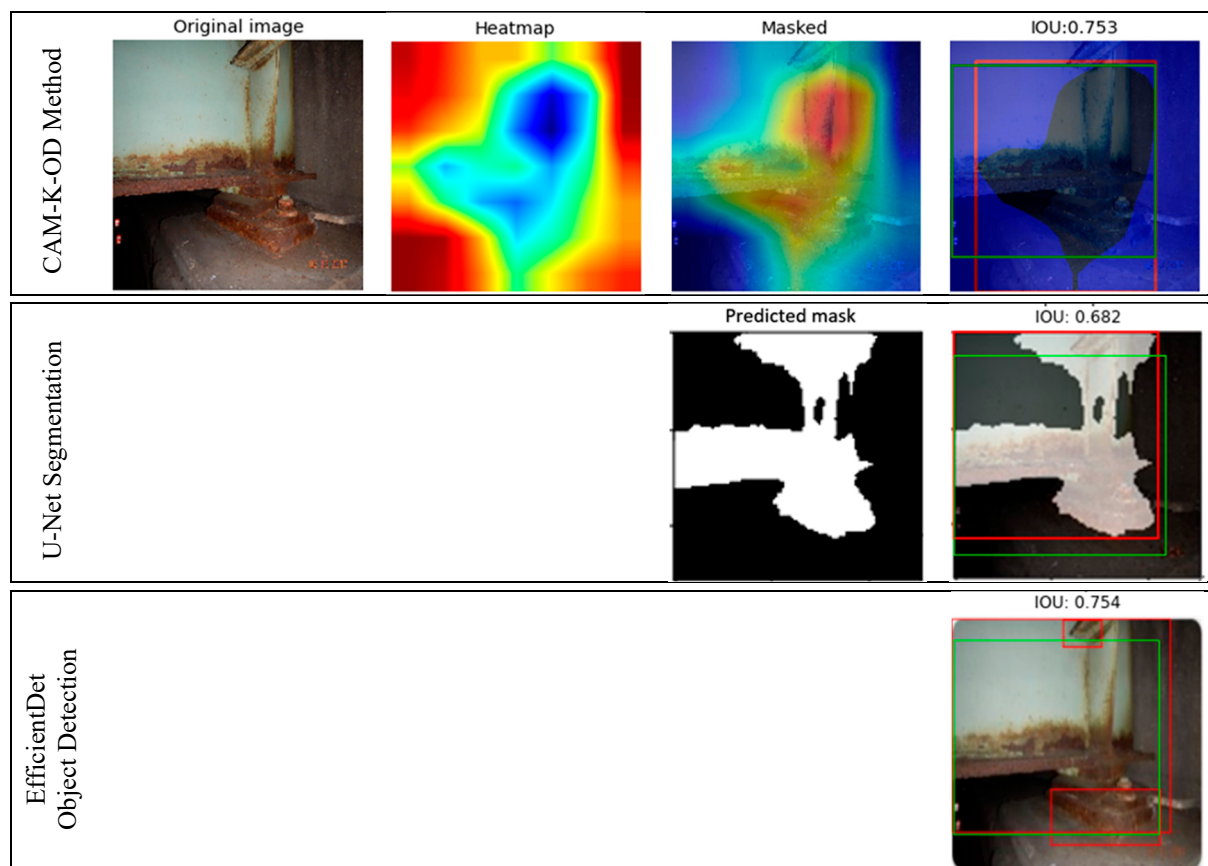
This section compares the proposed CAM-K-OD method with two other commonly used computer vision models: U-Net (a segmentation network) and EfficientDet (an object detection architecture). The primary purpose of these comparisons is to demonstrate the effectiveness of CAM-K-OD in detecting corrosion, as well as to highlight any no-

table performance differences among the three approaches when they are applied to the same dataset.

Figure 18 displays the expected areas of corrosion as determined by analyses of various steel bridges. In each subfigure (a–e), the green box shows the actual corrosion (ground truth), while the red box shows the area identified by the model. It can be seen that the bounding boxes for each analysis using CAM-K-OD identify areas near the bounding boxes of the actual corrosion areas.

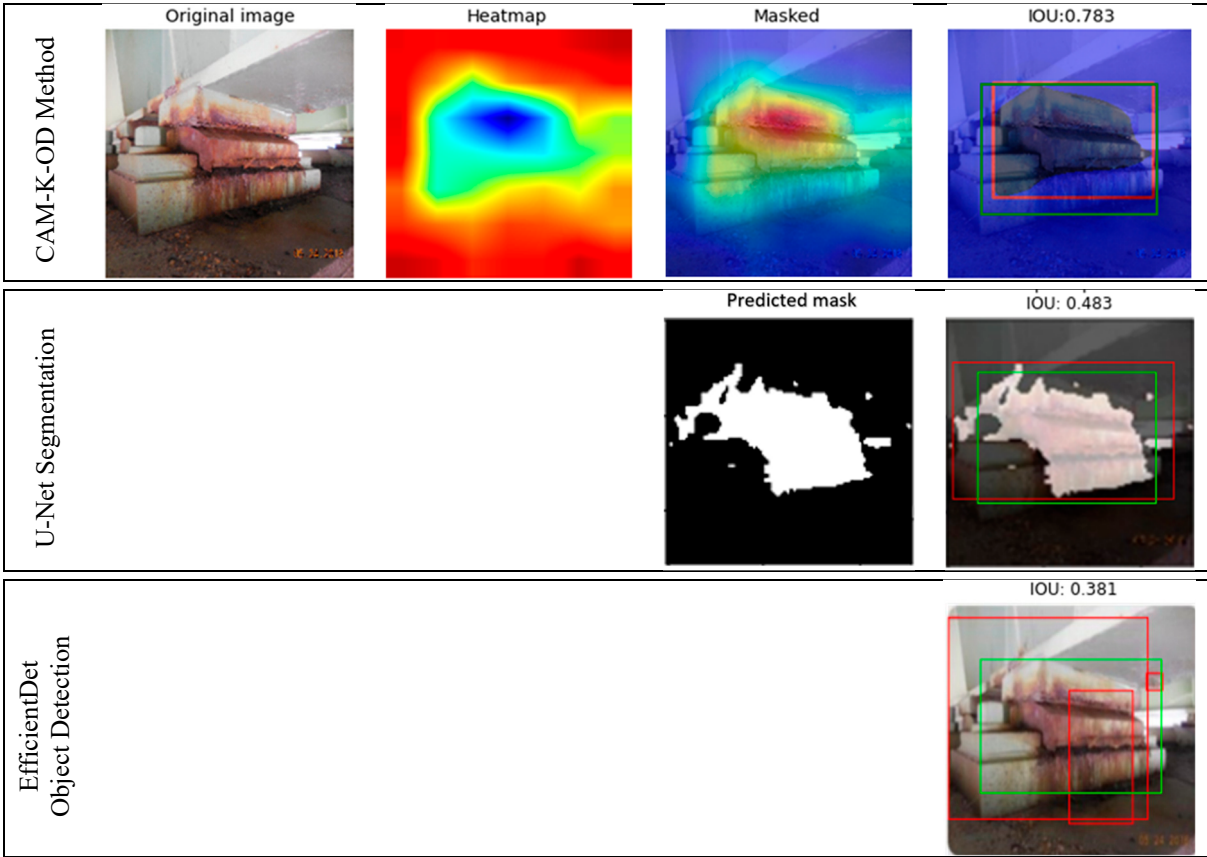
The results enable a comparison of the performance of three computer vision models: CAM-K-OD, U-Net, and EfficientDet. The models were evaluated on five different test images, labeled as Figure 18a through to Figure 18e, and their average IoU was also calculated. A higher IoU indicates a better performance. Of the three architectures, CAM-K-OD demonstrated superior spatial accuracy, yielding an average IoU of 0.731, outperforming EfficientDet (mean IoU = 0.640) and U-Net (mean IoU = 0.633). Looking at the individual test images, it can be seen that CAM-K-OD performs consistently well across all five images, with an IoU ranging from 0.635 to 0.783. U-Net performs well on some images (e.g., Figure 18c, where it had an IoU of 0.682) but not as well on others (e.g., Figure 18b, where it had an IoU of 0.483). EfficientDet performs similarly to U-Net, but with slightly higher IoU scores overall.

The figures also show the percentage difference in the average IoU between CAM-K-OD and U-Net and between CAM-K-OD and EfficientDet. CAM-K-OD outperforms U-Net by 13% and EfficientDet by 12%. Overall, the results suggest that CAM-K-OD is the best-performing model among the three for this particular corrosion detection task.

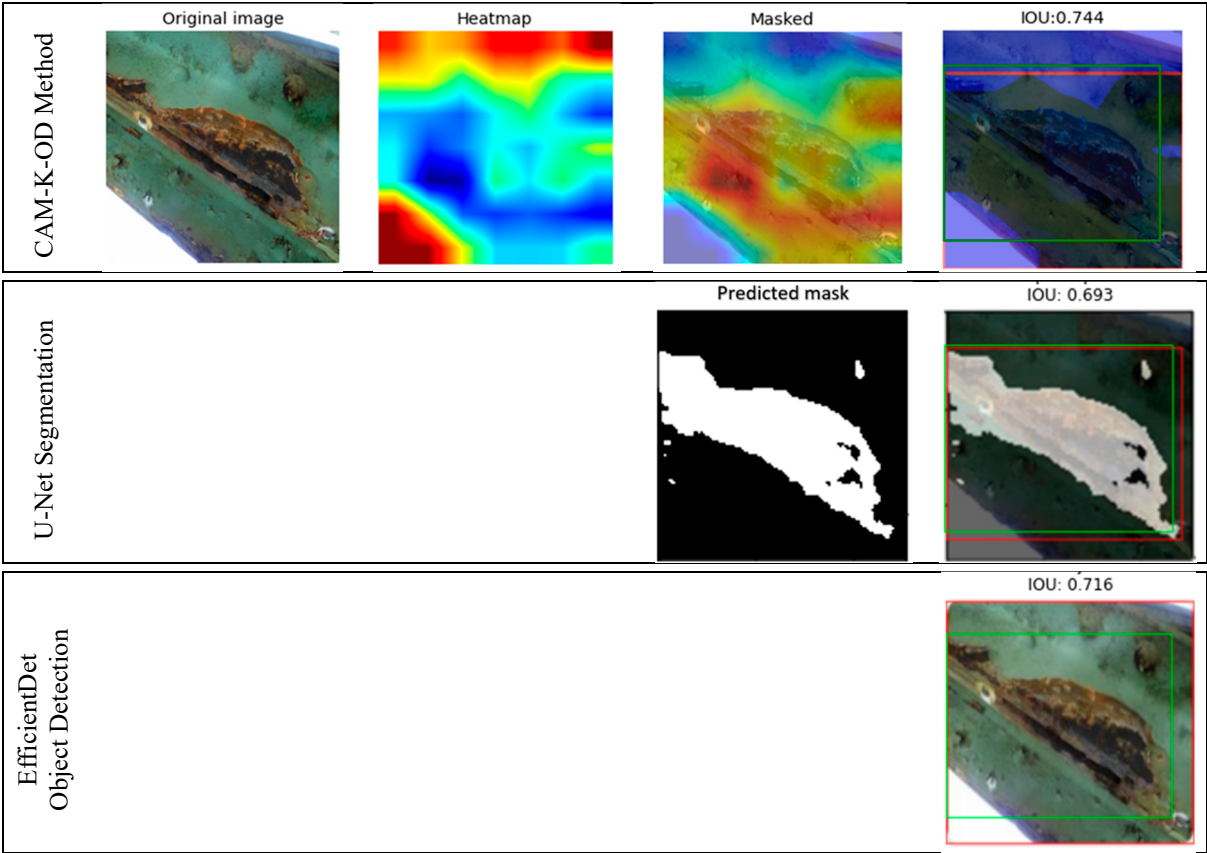


(a)

Figure 18. Cont.

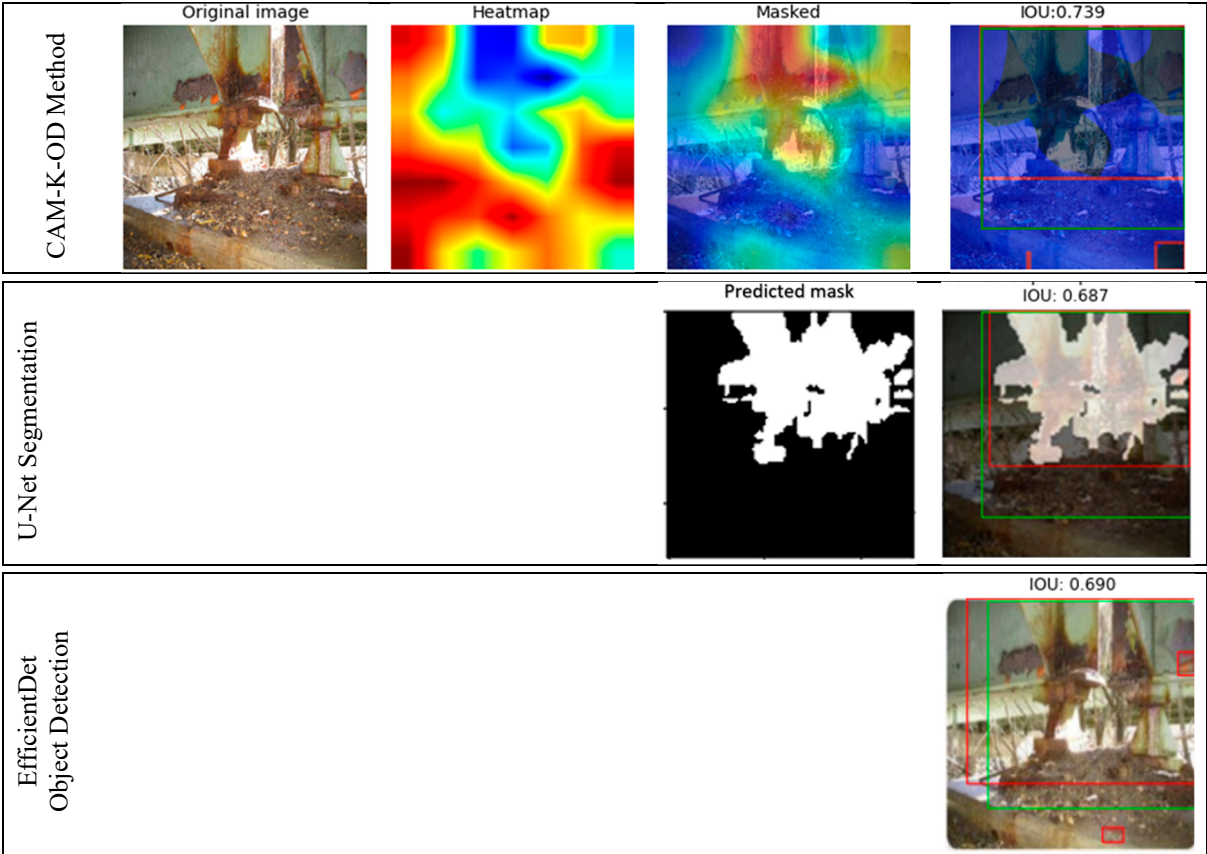


(b)

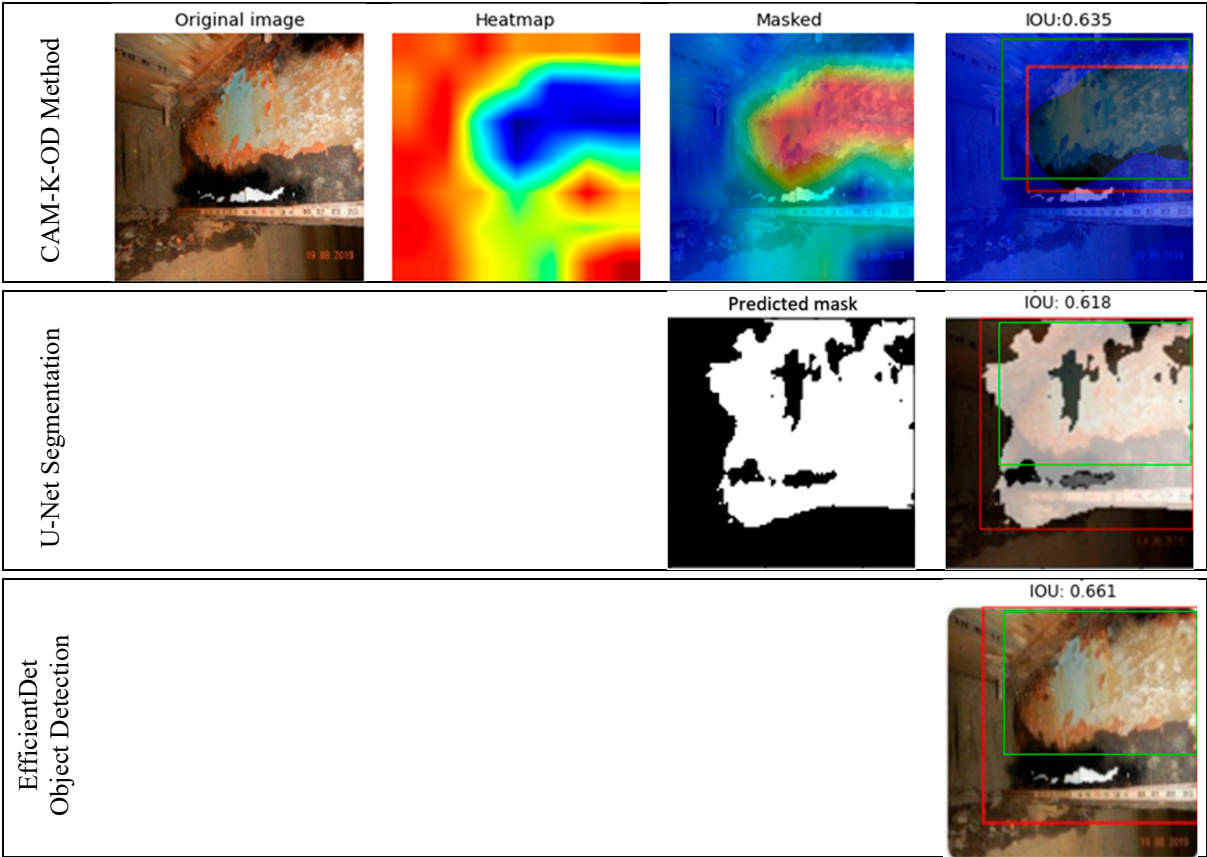


(c)

Figure 18. Cont.



(d)



(e)

Figure 18. Examples of using CAM-K-OD to determine the corroded area.

## 5. Discussions

In this study, a corrosion detection system for steel bridge structures was proposed using the CAM-K-OD object detection model and compared its performance to that of the U-Net and EfficientDet models. The proposed model accurately located corrosion in steel bridge structures and demonstrated a superior performance on certain test sets compared to the U-Net and EfficientDet models.

A possible limitation of this model is the need to use multiple classes to detect areas of corrosion. That is, to better define corrosion, non-corrosive objects should be defined as a separate class. However, the CAM-K-OD model has the advantage of being able to quickly and practically locate corrosion areas in all images when provided with only the input images, while the U-Net model requires the masking of all training images and the EfficientDet model the enclosing of the corrosion areas in all training images.

Overall, the results suggest that the proposed framework offers considerable potential as a corrosion identification and localization tool for the SHM of steel bridges. However, while initial results underscore the superiority of CAM-K-OD over both U-Net and EfficientDet within the evaluated dataset, future work should evaluate their performances across different test sets and identify potential areas for improvement. It should also be acknowledged that the proposed framework is one of several possible approaches to corrosion detection, and there may be other approaches that could yield even better results.

The lighting conditions in which vision-based methods are developed and tested can significantly impact their performance, particularly when it comes to the inspection of bridge components. The design and development of such methods must account for the variations in lighting conditions that may occur in real-world settings. This necessitates laboratory-scale evaluations of the proposed method under controlled lighting conditions to identify potential issues and refine the system accordingly. Subsequently, the method must undergo field tests to evaluate its efficacy in practical scenarios. By addressing the impact of lighting conditions and conducting rigorous testing, vision-based methods for bridge component inspection with improved performance and reliability can be developed. In this study, test analyses were carried out on images taken in the most suitable lighting conditions. To achieve more accurate and sensitive determinations, a dataset should be created that takes into account all light conditions (shade, sunset, daylight, cloudy weather, rainy weather, etc.).

## 6. Conclusions

This work has presented a novel corrosion detection system that uses a CNN-based pre-trained model for deep learning, the CAM-K-OD object detection model, which can be used for the detection of corrosion occurring in steel bridge structures. Built using transfer learning and visual localization techniques, the system was evaluated alongside U-Net and EfficientDet to determine its effectiveness in identifying corroded regions.

In binary classification tasks, the model achieved 1.00 in all three key classification metrics, indicating its excellent ability to accurately identify instances of “corrosion” and “no corrosion” with negligible false detections. This is a crucial aspect of this corrosion detection system, as accurately identifying the presence or absence of corrosion is essential for accurately assessing the condition of steel bridge structures.

Subsequently, the performance of the three models were assessed using the IoU. The CAM-K-OD model achieved the highest mean IoU score of 0.731, slightly higher than the EfficientDet, with an average IoU of 0.640, and the U-Net, with an average IoU of 0.633.

Overall, the results suggest that the CAM-K-OD object detection model performs well in detecting corrosion in steel bridge structures, with a strong classifier and a slightly higher average IoU compared to the U-Net and EfficientDet models. Its perfect classifier scores

ensure that corrosion is correctly identified, while the slightly higher average IoU of the CAM-K-OD model suggests that it can accurately detect corrosion areas, with a slightly higher overlap with the ground truth compared to the U-Net and EfficientDet models.

**Author Contributions:** Conceptualization, K.Ö. and A.C.A.; Methodology, K.H., K.Ö. and A.C.A.; Software, K.H.; Validation, K.H.; Formal analysis, K.H.; Investigation, K.H. and K.Ö.; Resources, A.C.A.; Data curation, K.H.; Writing—original draft, K.H. and K.Ö.; Writing—review & editing, S.M. and A.C.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author due to privacy.

**Conflicts of Interest:** Author Ahmet Can Altunışık is the founder and CEO of Dynamic Academy Software, Construction Ind. Trade Limited Company; the company had no role in funding, data collection, analysis, or publication. All authors declare no commercial or financial conflicts of interest.

## References

1. Avci, O.; Abdeljaber, O.; Kiranyaz, S.; Hussein, M.; Gabbouj, M.; Inman, D.J. A Review of Vibration-Based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and Deep Learning Applications. *Mech. Syst. Signal Process.* **2021**, *147*, 107077. [\[CrossRef\]](#)
2. Shihavuddin, A.S.M.; Chen, X.; Fedorov, V.; Christensen, A.N.; Riis, N.A.B.; Branner, K.; Dahl, A.B.; Paulsen, R.R. Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis. *Energies* **2019**, *12*, 676. [\[CrossRef\]](#)
3. Nex, F.; Duarte, D.; Tonolo, F.G.; Kerle, N. Structural Building Damage Detection with Deep Learning: Assessment of a State-of-the-Art CNN in Operational Conditions. *Remote Sens.* **2019**, *11*, 2765. [\[CrossRef\]](#)
4. Wang, N.; Zhao, X.; Zhao, P.; Zhang, Y.; Zou, Z.; Ou, J. Automatic Damage Detection of Historic Masonry Buildings Based on Mobile Deep Learning. *Autom. Constr.* **2019**, *103*, 53–66. [\[CrossRef\]](#)
5. Lin, Y.Z.; Nie, Z.H.; Ma, H.W. Structural Damage Detection with Automatic Feature-Extraction through Deep Learning. *Computer-Aided Civil. Infrastruct. Eng.* **2017**, *32*, 1025–1046. [\[CrossRef\]](#)
6. Yang, C.; Chen, J.; Li, Z.; Huang, Y. Structural Crack Detection and Recognition Based on Deep Learning. *Appl. Sci.* **2021**, *11*, 2868. [\[CrossRef\]](#)
7. Hu, G.X.; Hu, B.L.; Yang, Z.; Huang, L.; Li, P. Pavement Crack Detection Method Based on Deep Learning Models. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5573590. [\[CrossRef\]](#)
8. Park, S.E.; Eem, S.H.; Jeon, H. Concrete Crack Detection and Quantification Using Deep Learning and Structured Light. *Constr. Build. Mater.* **2020**, *252*, 119096. [\[CrossRef\]](#)
9. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil. Infrastruct. Eng.* **2017**, *32*, 361–378. [\[CrossRef\]](#)
10. Choudhary, G.K.; Dey, S. Crack Detection in Concrete Surfaces Using Image Processing, Fuzzy Logic, and Neural Networks. In Proceedings of the 2012 IEEE 5th International Conference on Advanced Computational Intelligence, ICACI, Nanjing, China, 18–20 October 2012; pp. 404–411. [\[CrossRef\]](#)
11. Nash, W.; Zheng, L.; Biribilis, N. Deep Learning Corrosion Detection with Confidence. *npj Mater. Degrad.* **2022**, *6*, 26. [\[CrossRef\]](#)
12. Nash, W.T.; Powell, C.J.; Drummond, T.; Biribilis, N. Automated Corrosion Detection Using Crowd Sourced Training for Deep Learning. *Corrosion* **2019**, *76*, 135–141. [\[CrossRef\]](#) [\[PubMed\]](#)
13. 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. [\[CrossRef\]](#)
14. Petricca, L.; Moss, T.; Figueroa, G.; Broen, S. Corrosion Detection Using A.I: A Comparison of Standard Computer Vision Techniques and Deep Learning Model. *Comput. Sci. Inf. Technol.* **2016**, 91–99. [\[CrossRef\]](#)
15. Rahman, A.; Wu, Z.Y.; Kalfarisi, R. Semantic Deep Learning Integrated with RGB Feature-Based Rule Optimization for Facility Surface Corrosion Detection and Evaluation. *J. Comput. Civil. Eng.* **2021**, *35*, 04021018. [\[CrossRef\]](#)
16. Forkan, A.R.M.; Kang, Y.B.; Jayaraman, P.P.; Liao, K.; Kaul, R.; Morgan, G.; Ranjan, R.; Sinha, S. CorrDetector: A Framework for Structural Corrosion Detection from Drone Images Using Ensemble Deep Learning. *Expert Syst. Appl.* **2022**, *193*, 116461. [\[CrossRef\]](#)

17. Brandoli, B.; de Geus, A.R.; Souza, J.R.; Spadon, G.; Soares, A.; Rodrigues, J.F.; Komorowski, J.; Matwin, S. Aircraft Fuselage Corrosion Detection Using Artificial Intelligence. *Sensors* **2021**, *21*, 4026. [CrossRef]
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 17–30 June 2016; pp. 779–788. [CrossRef]
19. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput.-Aided Civil. Infrastruct. Eng.* **2018**, *33*, 731–747. [CrossRef]
20. Yoshida, S.; Feng, G.-L.; Xu, J.; Grazzini, A.; Piana, G.; Ta, Q.-B.; Huynh, T.-C.; Pham, Q.-Q.; Kim, J.-T. Corroded Bolt Identification Using Mask Region-Based Deep Learning Trained on Synthesized Data. *Sensors* **2022**, *22*, 3340. [CrossRef]
21. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
22. Ahmad, A.; Dey, L. A K-Mean Clustering Algorithm for Mixed Numeric and Categorical Data. *Data Knowl. Eng.* **2007**, *63*, 503–527. [CrossRef]
23. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 234–241.
24. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2019; pp. 10778–10787. [CrossRef]
25. Khandelwal, R. Deep Learning Using Transfer Learning. Towards Data Science 2019. Available online: <https://medium.com/data-science/deep-learning-using-transfer-learning-cfbce1578659> (accessed on 3 March 2025).
26. Hacıfendioğlu, K.; Demir, G.; Başağa, H.B. Landslide Detection Using Visualization Techniques for Deep Convolutional Neural Network Models. *Nat. Hazards* **2021**, *109*, 329–350. [CrossRef]
27. Hacıfendioğlu, K.; Başağa, H.B.; Ayas, S.; Karimi, M.T. Automatic Detection of Icing Wind Turbine Using Deep Learning Method. *Wind. Struct.* **2022**, *34*, 511–523.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
29. Höppner, F.; Klawonn, F.; Kruse, R.; Runkler, T. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis, and Image Recognition*; John Wiley & Sons: New York, NY, USA, 2000; ISBN 0471988642.
30. Liu, L.; Cheng, J.; Quan, Q.; Wu, F.X.; Wang, Y.P.; Wang, J. A Survey on U-Shaped Networks in Medical Image Segmentations. *Neurocomputing* **2020**, *409*, 244–258. [CrossRef]
31. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3686–3693.
32. Bianchi, E.; Hebdon, M. Corrosion Condition State Semantic Segmentation Dataset. Available online: [https://figshare.com/articles/dataset/Corrosion\\_Condition\\_State\\_Semantic\\_Segmentation\\_Dataset/16624663](https://figshare.com/articles/dataset/Corrosion_Condition_State_Semantic_Segmentation_Dataset/16624663) (accessed on 12 October 2022).

**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.