INL/JOU-25-82915-Revision-0

Accurate segmentation of localized corrosion in structural alloys via deep learning

June 2025

Fei Xu, Xiaolei Guo, Jenifer Locke, Liang Zhao, Tiankai Yao

Changing the World's Energy Future

Idaho National Laboratory

INL/JOU-25-82915-Revision-0

Accurate segmentation of localized corrosion in structural alloys via deep learning

Fei Xu, Xiaolei Guo, Jenifer Locke, Liang Zhao, Tiankai Yao

June 2025

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

npj | materials degradation

Article

Published in partnership with CSCP and USTB

https://doi.org/10.1038/s41529-025-00633-3

Accurate segmentation of localized corrosion in structural alloys via deep learning

Liang Zhao[1], Jenifer Locke[2], Fei Xu[1], Tiankai Yao[1] ✉ & Xiaolei Guo[3] ✉

This study presents a deep learning-based approach for the automated segmentation of corrosion damage in scanning electron microscopy (SEM) images. The proposed method enables rapid and accurate segmentation of corrosion features in these SEM images, making it highly suitable for real-time applications such as automated microscopy. Specifically, a dedicated corrosion segmentation database tailored for this task is constructed. The newly constructed dataset, alongside data from two public databases, are employed to jointly train a deep learning-based model modified with a texture refinement module. Compared to the same model without the texture refinement module, the refined model substantially enhances the efficacy and efficiency of corrosion segmentation. Furthermore, the methodology developed here is extendable to segmentation tasks for other materials with similar resolution, texture, and contrast characteristics, thereby paving the way for accelerated and automated analysis in corrosion science and beyond.

Localized corrosion, such as pitting corrosion[1–4], crevice corrosion[5–8], and stress corrosion cracking (SCC)[9–11], is a leading cause of structural alloy failure in engineered systems. Compared to general corrosion that typically progresses uniformly, localized corrosion tends to initiate at microstructural features with higher susceptibility. These features include inclusions, secondary-phase particles, grain boundaries, and regions with compositional or structural heterogeneity, where the electrochemical conditions favor corrosion processes. Once initiated, often in hidden or difficult-to-access areas, localized corrosion could penetrate rapidly into the material, potentially causing catastrophic failure[12]. Therefore, reliable prediction of localized corrosion is crucial for ensuring the structural integrity of engineered systems such as oil pipelines, nuclear reactors, chemical processing equipment, marine vessels, and aerospace components.

For example, stainless steel (SS) 304 L, which is used as the canister material for the interim storage of hazardous spent nuclear fuel (SNF), is susceptible to pitting corrosion and subsequent SCCs[13,14]. Prior studies have shown that the morphology of pits forming on the surface of SS304 canisters is influenced by the relative humidity (RH)[15]. Specifically, hemispherical pits tend to develop on the SS surface at a high RH of ~75%, while pits with crosshatch morphologies and micro-fissures were observed at a lower RH of ~40%. The difference is largely attributed to the variations in salt chemistry that occur during deliquescence at different RHs. Notably, the crosshatch morphology and micro fissures can serve as stress concentrators on the SS, thereby transitioning pits into more hazardous cracks. These cracks can progress through-wall penetration, potentially leading to the release of hazardous radionuclides from the encapsulated SNF. Therefore, it is critical to identify pits on the surface of SS canisters and analyze their corresponding morphology. This approach could enable early prediction of pit-to-crack transition, informing timely inspection and mitigation to prevent more severe damage.

The goal of this study is to develop a computational model capable of segmenting corrosion morphology from two-dimensional (2D) SEM images collected from SS304L samples with complex corrosion damage. This study aims to lay the foundation for the follow-on segmentation of three-dimensional (3D) corrosion morphologies, enabling the extraction of critical features that may contribute to pit-to-crack transition.

In recent years, machine learning (ML) and artificial intelligence (AI) based computer vision (CV) methods have been widely adopted to automatically identify objects in natural images. However, their application to microstructural images in material science remains in its early stage[16]. Among these approaches, deep learning (DL)-based models have demonstrated significant improvements in object detection and instance segmentation tasks, making them particularly suitable for our objective of corrosion segmentation. Classical methods such as U-Net based architectures[17], You Only Look Once (YOLO) series[18], Mask Regional Convolutional Neural Network (Mask R-CNN)[19], and the Detection

[1]Idaho National Laboratory, Idaho Falls, ID, 83402, USA. [2]Department of Materials Science and Engineering, The Ohio State University, Columbus, OH, 43210, USA. [3]Department of Metallurgical and Materials Engineering, Colorado School of Mines, Golden, CO, 80401, USA. ✉e-mail: tiankai.yao@inl.gov; xiaolei.guo@mines.edu

TRansformer (DETR)[20] series are widely used, especially for segmentation tasks involving natural and medical objects.

Despite their success, most of these methods are developed exclusively for natural scenes or are further adapted for medical images, where each category of objects exhibits distinct attributes and requires a large amount of labeled data. For example, segmentation models have been effectively applied to tumor identification in medical computed tomography scans with low contrast, leveraging vast datasets representing various tumor types. However, corrosion represents a unique challenge as it does not align with the standard object categories found in natural object detection or segmentation frameworks. Furthermore, the segmentation of corrosion images is complicated by its inherent morphological diversity, high variance, and the scarcity of labeled data.

Several efforts have been made to identify pitting corrosion using ML/DL based approaches. Early works primarily focused on determining whether the corrosion damage was present. For example, Ahuja et al. presented an adaptive neural network-based approach for the classification of different types of damage in metals, with an ability to distinguish between images of corroded piplines and those with non-corrosion patterns resembling corrosion[21]. In addition, convolutional neural networks (CNN) based models were also applied for corrosion detection[22]. These models extended capabilities by identifying corrosion locations as bounding boxes, which are rectangular regions containing large areas of normal surfaces. In recent work, Katsamenis et al. applied advanced DL models, including the Fully Convolutional Network (FCN), U-Net, and Mask R-CNN, for corrosion detection[23]. However, the resulting images did not achieve the level of accuracy required for structural analysis and prefabrication, particularly in our specific application.

When we directly applied some existing models to our SEM data, the results did not show successful segmented corrosion regions. Even when corrosion was identified, the segmented regions exhibited arbitrary shapes and poorly defined boundaries that blended with their surroundings. There are two main reasons for this outcome. The first reason is that existing models were trained on datasets of natural or medical images, which lack features similar to corrosion or comparable objects. Although transfer learning technically can provide initialization for this specific task, its applicability here was limited. The closest dataset potentially of use is the Pothole Image Segmentation (PIS)[24] dataset, which was used for road holes detection in driving safety. However, the data differs substantially from our microstructural images obtained via SEM, which display low contrast and limited differentiability from the background. Some corrosions regions even exhibited attributes nearly indistinguishable from their background, even to the human eyes (Fig. 1a). We attempted to train YOLOv9[25] on this dataset without any modifications. The testing result is shown in Fig. 1b. Compared to the correct result in Fig. 1c, where we manually labeled the ground truth for this image, the YOLOv9 model that performed well in standard objects could not identify any corrosion in the image. This indicates that with existing data, the popular models alone may be incapable of identifying corrosions for our task. The second reason for the poor performance is the lack of model adaptation for our specific task. Addressing such challenges requires specialized approaches. For example, many methods used in medical image segmentation adopt boundary detection methods, which are tailored to their specific imaging requirements. Similarly, new strategies are essential for adapting models to our need for corrosion segmentation.

In this study, we aim to address these challenges by constructing a new database from our experimental data and developing an efficient instance segmentation method tailored for the segmentation of materials corrosion. Firstly, we manually labeled the data with pixel-level annotations (an example is shown in Fig. 1c), which could be used for general segmentation of corrosion morphologies, such as pitting damage, in SEM images. Secondly, compared to large models with extensive parameters that could be under-parameterized to limited data, the YOLOv9 model is efficient in both the performance and computational costs, although it cannot solve our corrosion segmentation problem completely. This model effectively preserves input information through a lightweight deep neural network architecture. Furthermore, we innovatively adapted a texture refinement module, which was integrated into the network. This module encoded the internal structures of corrosion morphology relative to their surroundings, thereby improving the segmentation performance. Overall, the model can automatically and accurately segment the corrosion morphology, as shown in Fig. 1d, using SEM images.

Results

This work utilized SEM images of SS 304 L samples with localized corrosion. The details of corrosion experiments were described in a prior study[1]. In brief, SS304L samples were ground to 60 to 120 grit during materials fabrication at the manufacturer plant instead of by hand on a laboratory polishing wheel. The ground materials were then loaded with 300 μg/cm^2 ASTM D1141 sea salt. The salt loaded specimens were subsequently placed in a corrosion exposure chamber with controlled temperature and RH. During the corrosion study, the RH was maintained at a constant value of 75% at room temperature for a period of 1 to 52 weeks. Before imaging and analysis, corrosion products from the exposed samples were removed by scrubbing the sample with a nylon bristle brush in running tab water, followed by sonication in a 10 vol.% HNO_3 solution for 30 to 120 min. The specimens were finally rinsed with abundant deionized water and then air dried for SEM observation.

This section includes the results of experiments in two stages: initial corrosion segmentation and final results improved by the developed method. Comprehensive results and quantitative statistics are provided below for detailed evaluations.

Initial segmentation

The initial corrosion segmentation results were obtained from the model that was trained on the three datasets mentioned above but without incorporating the proposed texture refinement module. The purpose of this step was to identify the improvement brought by the texture refinement module we proposed here. As shown in Fig. 2, an example input was fed into

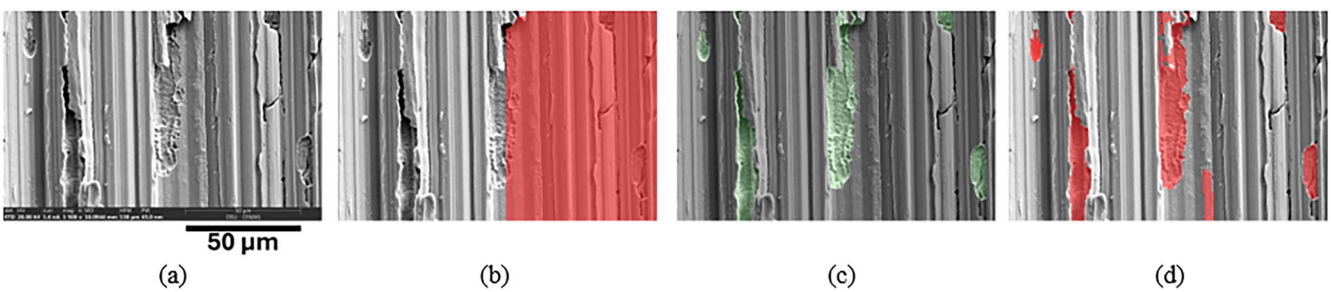

Fig. 1 | Illustration of instance segmentation of corrosions on an example SEM image. a An input SEM image as collected. **b** Segmentation results generated using a widely used method[18] trained on a pothole dataset[24]. The red regions represent the segmented areas. **c** Labeled data from our manual annotations, shown in green color. **d** Results obtained using our deep learning-based model.

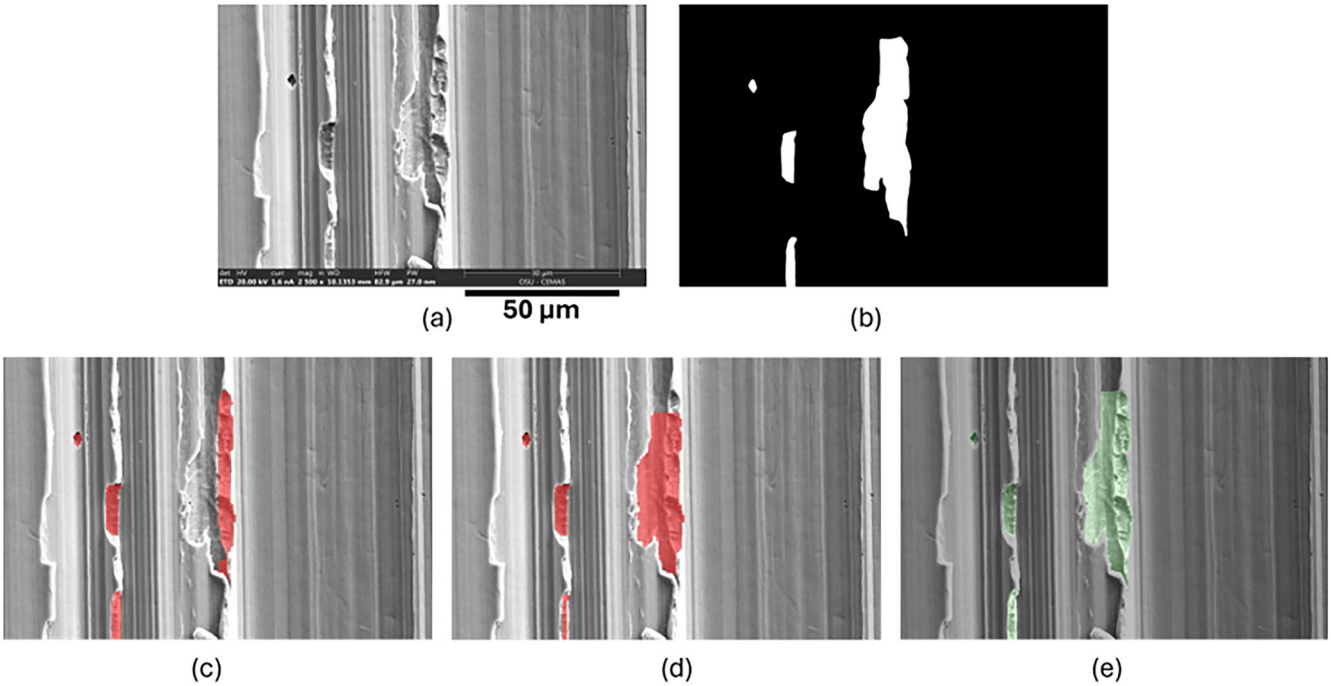

Fig. 2 | Comparison of results. a Input example. **b** Manually labeled ground truth. **c** Initial segmentation results without the proposed texture refinement module. **d** Final segmentation results with the texture refinement module. **e** Ground truth visualized with the original input SEM image.

the model, and the initial result is depicted in Fig. 2c. Compared with the ground truth, some corrosion regions were not captured due to the similar texture and low contrast of the image. The low-contrast texture was a major challenge in our data, making it difficult for the model to distinguish the corrosion damage from its surroundings. This in turn impacted the final accuracy of the model. Even with enough data to customize the model, it was insufficient to fully address the challenges presented by our task.

Texture refinement module

To address the problems in our corrosion segmentation task and overcome the limitations, we purposefully integrated the texture refinement module into the entire training process. With the same procedures, the final results are given in Fig. 2d. Compared with the initial results illustrated in Fig. 2c, the refined result accurately identified the correct corrosion regions in the center with a well-defined boundary. To facilitate visual analysis, the predicted masks were overlaid on the original SEM images, enabling a direct comparison of the performance against the ground truth. Compared with the ground truth in Fig. 2e, the model with the texture refinement module segmented nearly all the corrosion regions in the micrograph. This demonstrates the validity of our texture refinement module, which can help the model to discern the texture variations and improve the segmentation of corrosion morphologies.

Overall results

The refined model was applied to all SEM images in the testing group, with four examples shown in Fig. 3. The top two SEM images in Fig. 3a collected at higher magnifications showed typical localized corrosions mainly characterized as pits with varying morphologies. Due to differences in lighting and contrast levels, the surrounding regions also exhibited various features. The ground truth for the pit morphology was labelled in Fig. 3b. For the initial segmentation results shown in Fig. 3c, most of the large corrosion areas were segmented. However, more accurate shapes were captured in the final results in Fig. 3d. Comparing Fig. 3c and 3d, it is evident that some corrosion areas were incorrectly labelled in Fig. 3c, which was obtained without the proposed texture refinement module. In contrast, the segmented corrosion areas shown on Fig. 3d with the texture refinement module closely matched the ground truth. The last two rows of SEM images in Fig. 3a were collected at low magnification, showing a distant view of multiple corrosion areas with reduced contrast than the top two rows of Fig. 3a. These corrosion areas are relatively smaller and display less morphological variations compared to the SEM images collected at higher magnifications. Further, some corrosion areas lack clear boundaries. Compared to the results in Fig. 3c, the improved results (in Fig. 3d) show more accurate shapes of the corrosion areas and less false positives. Despite the improved accuracy, the segmentation of some corrosion areas was confounded by the uncorroded area. This needs to be improved further in future studies. One typical example, shown in the last row of Fig. 3, highlights how the corrosion damage in the center of the image gradually transitions into uncorroded areas. While humans can discern the levels of corrosion, the model struggled with such subtle distinction. Nonetheless, the developed model identified most corrosion details and generated shapes that are highly consistent with the ground truth.

The results shown above demonstrate that our method is effective in automatically segmenting corrosion features in SEM data. Although minor discrepancies exist compared to the ground truth, these errors mainly are localized to small regions with a high density of corrosion variations. Such challenges arise due to low contrast and overlapping areas, especially for the areas with fragmented corrosion features. Additionally, there were varying degrees of corrosion with similar but subtly different morphologies within the same regions, as well as tiny corrosion areas that may not be captured in low magnification images.

The statistics of the segmentation results were also generated and presented in Table 1. The metrics used in the experiments include precision, recall, and mean average precision (m-AP) for pixel-level segmentation. Conventionally, True Positives (TP) means the number of instances where the model predictions match correctly the ground truth. True Negatives (TN) denotes the number of instances predicted by the model that do not match the ground truth. False Positives (FP) presents the number of instances from model prediction that do not match the ground truth. False Negatives (FN) is the number of instances in the ground truth that the model fails to predict. The formulas for the metrices are: $Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$, and $mAP = 1/N\sum_{i=1}^{N} AP_i$, where $AP = \int_0^1 P(R)dR$ is the area under the precision-recall curve $P(R)$, and N is the number of images. These

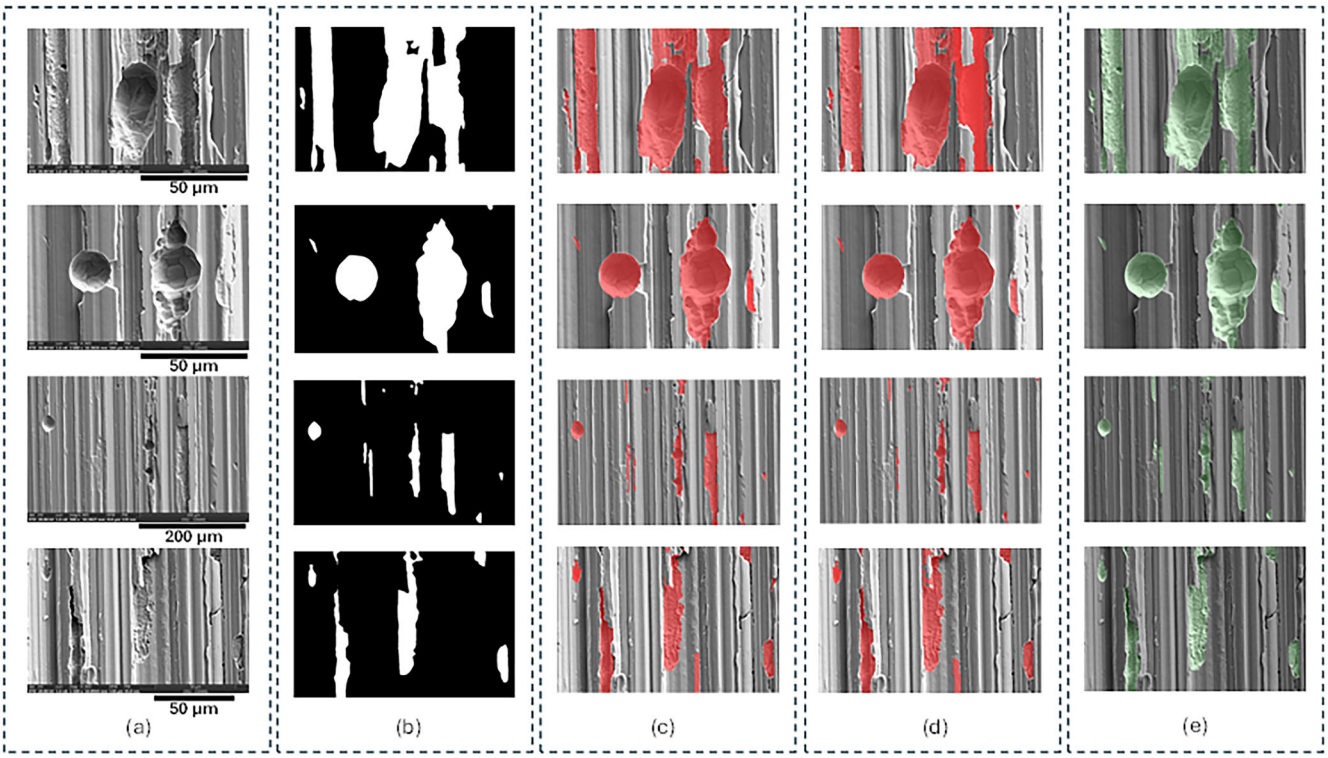

Fig. 3 | Corrosion segmentation results. a Input examples. **b** Manually labeled ground truth. **c** Initial corrosion segmentation without the texture refinement module. **d** Final segmentation results with texture refinement module. **e** Ground truth visualized in original SEM images.

Table 1 | Quantitative results of testing samples

Experiments	Images (#)	Instances (#)	Precision (%)	Recall (%)	mAP (%)	Time (seconds)
1	9	51	28.8	14.8	13.8	5
2	9	51	70.9	68.5	71.2	5
3	9	51	76.8	84.3	85.2	8

The metrics were presented as percentage values with numerical counts denoted by "#" and time represented in seconds. Index number 1, 2, and 3 indicate results obtained without our dataset, without the proposed texture refinement module, and with the proposed module, respectively.

quantitative evaluations of the data show the robustness of our method. The metrics were calculated by considering three situations: (1) the model trained without the newly constructed dataset, (2) the model trained with the newly constructed dataset but without the texture refinement module, and (3) the model with both newly constructed dataset and the proposed texture refinement module. All models were trained in the same environment and conditions. The quantitative improvements achieved by our model are shown in Table 1. Notably, the overall accuracy of corrosion segmentation improved significantly from 28.8% to 76.8%. Similarly, the mAP improved from 13.8% to 85.2%. While the time cost increased from 5 s to 8 s, the difference is relatively small compared to the substantial improvement in accuracy. This trade-off highlights the efficiency of the proposed method, as the improvement in performance outweighs the marginal increase in computational time. Such enhancements make the approach practical for real-world applications where both accuracy and time efficiency are critical.

In Table 2, we present cross validation experiments for the proposed method. Experiment 1 corresponds to the final results reported in Table 1 and is included here for comparison. Experiments 2 to 7 are six additional trials using different random splits of the dataset, each with 75 images for training and 9 images for testing. Although the number of images remains constant, the specific images differ across splits, leading to variations in the total number of corrosion instances. Note that testing images of our Experiment 1 are manually selected and they are the most challenging samples in the data, which is to validate the effectiveness of our method. When difficult samples are randomly selected into training sets and easy samples are divided into testing sets for Experiment 2 to 7, the performance is increased in many metrics. As a result, some fluctuations are observed in precision, recall, and mAP metrics. Nevertheless, the results consistently demonstrate the validity of the proposed method.

Discussion

The SEM images used in this work presents significant challenges due to dense, complex, and highly interconnected corrosion morphologies with a contrast significantly lower than that of common natural objects used in CV and ML. Although many manual methods have achieved promising results on high quality SEM/TEM data, the popular models perform poorly on our SEM images. Compared to previous step-by-step processing approaches, we developed a model in an end-to-end framework without human intervention at each step. Instead of relying on simple CNNs to detect corrosion, we introduced a DL-based method incorporating a texture refinement module specifically designed for this task. The model was trained end-to-end on existing data, enabling it to directly segment corrosion instances without additional preprocessing or postprocessing steps. Moreover, for these challenging datasets, existing studies mainly rely on manual labeling of several micrographs and perform analysis on a sample-by-sample basis. In contrast, our method can generate pixel-level segmentation results for new

Table 2 | Cross validation of the proposed method

Experiments	Images (#)	Instances (#)	Precision (%)	Recall (%)	mAP (%)	Time (seconds)
1	9	51	76.8	84.3	85.2	8
2	9	58	94.5	70.7	82.9	8
3	9	66	94.9	84.3	88.0	8
4	9	69	91.8	92.8	94.2	8
5	9	58	97.2	84.5	87.3	8
6	9	47	100.0	95.0	95.6	8
7	9	45	94.2	72.8	89.2	8
Average	9	56	92.8	83.5	88.9	8

The metrics are presented as percentage values with numerical counts denoted by "#" and time represented in seconds. Indices 1 through 7 indicate results obtained from randomly splitting the training and testing sets across seven independent runs.

SEM data within seconds, offering a significant improvement in efficiency and scalability.

From this study, we also identified some limitations in the proposed method. One major challenge is the lack of sufficient training data specific to our target task, which exacerbates the domain gap problem during testing. This issue is common in the field of materials science, where obtaining adequate training data with manual labeling is often difficult. Furthermore, the domain of material micrographs is vastly different from the natural images typically used in general models, making it harder to achieve optimal performance. In this paper, we addressed this challenge by developing a texture refinement module that focused on the continuity problem in corrosion texture changes. In future work, we aim to further solve the domain gap problem to reduce its impact on the performance of our model with SEM data. By doing so, we can further enhance the efficiency and accuracy of corrosion segmentation and contribute to advancing research in material degradation.

Another limitation is the potential for human bias and errors during the manual labeling of SEM data. Although we initially identified corrosion damage with a tool model and consulted corrosion experts on the team for verification, the results can still vary due to the difference in image resolution and human visual perception. For manual annotations, discernible corrosion areas were labelled by humans based on SEM images, but these may deviate from appearances on SEM images, such as darker or brighter on different corrosion areas, blurred on corrosion pitting boundaries. The procedure of identifying and labelling corrosion needs to be standardized as much as possible to improve consistency. For example, when multiple corrosion areas are connected, clear criteria for subtle classifications need to be established. This would help avoid inconsistencies in learning and prevent non-robust performance in corrosion segmentation. Further improvement is particularly important for SEM data containing severe corrosion damage and interconnected morphology with subtle difference compared to the surrounding uncorroded area.

This work introduces a deep learning-based end-to-end method for automatically segmenting corrosion damage in SEM data. Without requiring any manual preprocessing, the developed model can directly perform corrosion segmentation on as-collected SEM data, making it suitable for real-time applications. Two main contributions are developed in the method. Firstly, the creation and development of a corrosion segmentation dataset specifically for this task provides a valuable resource that can broadly support research advancements in the field of material and corrosion science. Secondly, the end-to-end DL-based model integrated with a texture refinement module greatly improves the performance of corrosion segmentation in SEM data. This approach effectively addresses the challenges posed by texture variations in low-contrast corrosion images collected through SEM. The results demonstrate the robustness of the developed model in handling corrosion segmentation tasks for SEM images. Moreover, the model can be extended to segmentation tasks for general materials with similar resolution, texture, and contrast. This advancement holds the potential to accelerate automated processes and further propel research in the field of material science.

Methods

DL methods of the YOLO series represent state-of-the-art techniques for real-time object detection in natural image datasets such as MS COCO[26]. These models have consistently achieved significant advancements in efficiency, accuracy, and adaptability. To solve our problem of corrosion segmentation, we developed an instance segmentation method based on YOLOv9[25], a recently released version of the series. This version introduced groundbreaking techniques such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). Figure 4 shows the entire architecture of the model. The input micrograph was fed into a main branch (the upper part) to extract multiscale features and also processed through auxiliary branch (the bottom part) for additional refinement. The features were then concatenated into the prediction head to obtain segmentation results. To adapt the model to our specific tasks, we replaced the original detection head with a segmentation head and adopted a texture refinement module to enhance the texture details of corrosion morphology. In the following sections, we provide a detailed explanation of the architecture of the modules, then introduce the data used in our methods, and finally outline the training and testing processes.

Architecture

As shown in Fig. 4, the input micrograph X was fed into the main branch. It consists of multiple modules, including the Convolution, Batch Normalization, and Swish Activation (CBS), GELAN, Average Down-sample (ADW), Spatial Pyramid Pooling (SPP), Up-sampling (UP), and Concatenation (CON) blocks. Specifically, the UP and CON blocks represent up-sampling and concatenation layers, respectively. The other blocks contain multiple layers as shown in Fig. 5. The basic block CBS consists of a 2D convolutional layer equipped with a batch normalization layer and a Sigmoid Linear Unit (SiLU)[27] activation layer. Together with layers of the average pooling, maximum pooling, and concatenation as shown in Fig. 5e, the ADW module could gradually down-sample the features to extract multi-scale features. Equipped with maximum pooling layers, the SPP block (Fig. 5d) outputs all the down-sample features. The GELAN block (Fig. 5a) consists of RepNCSP[28] blocks (Fig. 5b) and BottleNetck blocks[29] (Fig. 5c). The lightweight GELAN is based on gradient path planning, which can achieve better parameter utilization than the traditional depth-wise convolution. It further analyses computational complexity, accuracy, and inference speed.

During training, the input micrograph X was also fed into the auxiliary branch (the bottom part in Fig. 4), which is the reversible design of PGI to mitigate information loss. The CBL block was introduced to aggregate gradient information from the main branch. It consists of three 2D convolutional layers. Subsequently the CBF block combined the information with the auxiliary branch using one resizing and one pixel-level addition layer. The purpose of the auxiliary branch is to generate reliable gradients

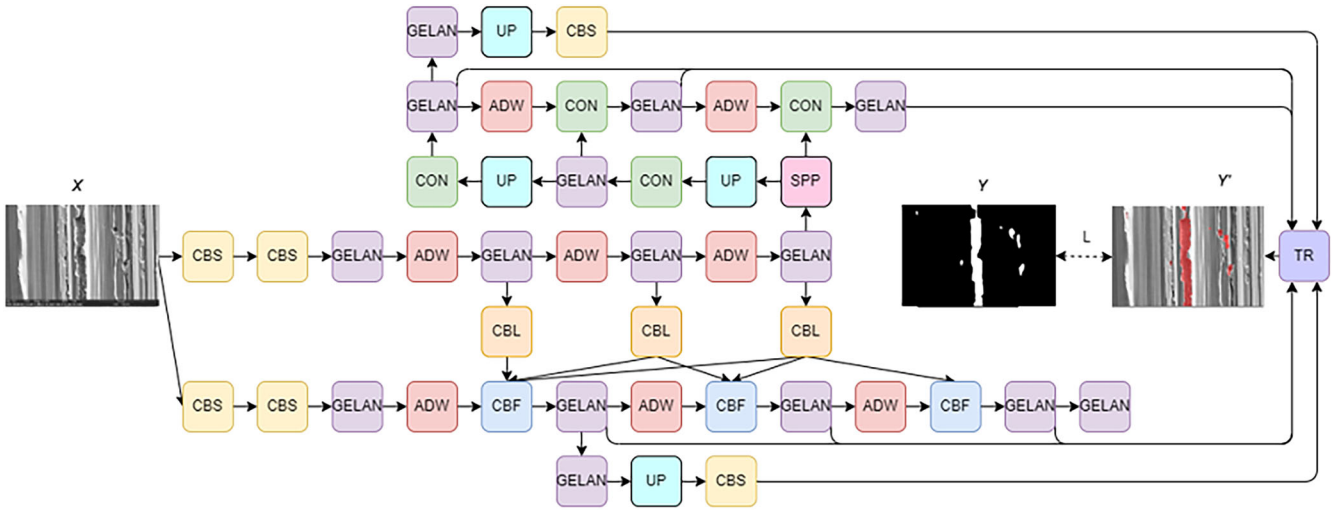

Fig. 4 | Overview of the architecture. The input SEM image X was fed into a YOLOv9-based framework for training. Each module consists of different convolutional layers and configurations, as detailed in Fig. 5. The predicted corrosion area Y′ were compared with the ground truth Y.

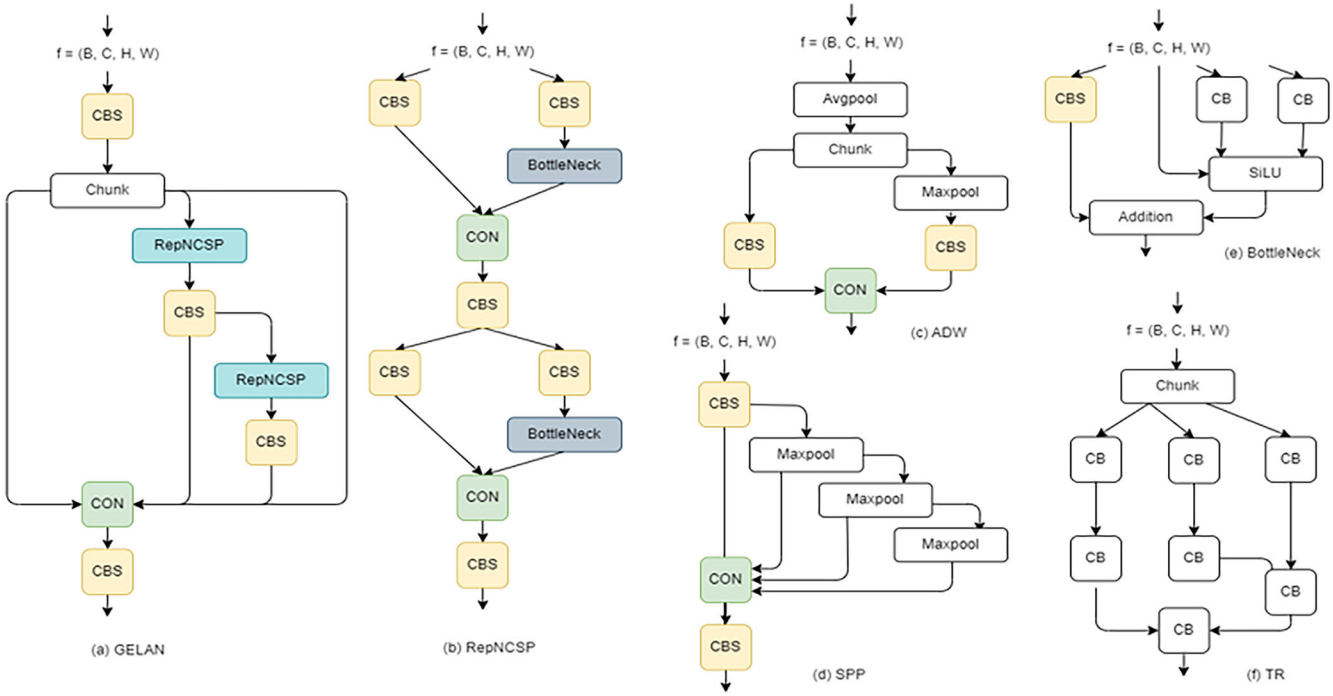

Fig. 5 | Illustration of multiple modules. a The GELAN module, which consists of CBS, CON, and RepNCSP blocks. **b** The configuration of the RepNCSP block. **c** The BottleNeck block. **d** The SPP block. **e** The ADW block. **f** The proposed texture refinement module. Most blocks shown in white color contain one single computational layer as indicated by their names, except for the CB block, which consists of one convolutional layer and one batch normal layer.

and maintain key characteristics within the deep representations. This ensures that features are more effective for the target task because each feature pyramid receives comprehensive information about all targets. It solves the issue of data loss for lightweight models, especially when the training data is limited. Notably, the issue of data loss often becomes severe as information propagates through deep networks. The problem arises when the convolutional network maps attributes between input and target. The loss of information results in incorrect gradient updates and consequently inaccurate predictions. According to the information theory[30], a reversible function $v_\tau(\cdot)$ is the inverse transformation of function $r_\omega(\cdot)$ if $X = v_\tau(r_\omega(X))$, where τ and ω are parameters of v and r, respectively. Thus, there is no information loss with reversible functions, as represented by $I(X,X) = I\big(X, v_\tau(r_\omega(X))\big)$, where I is mutual information. To mitigate information loss, the auxiliary branch was adopted to provide important information $I(Y,X)$, which maps data X to the target Y instead of relying solely on $I(X,X)$[25]. Formally, the relationship $I(Y,X) \geq I(Y, f_\theta(X)) \geq \cdots \geq I(Y, Y')$ holds, where $f_\theta(\cdot)$ is the transformation function or model with parameters θ, Y is the target, and Y' is the predicted result. Complementary information is combined with main branch information. When the objective function was calculated with more complete information, the prediction was improved.

Here we propose an innovative approach incorporating a texture refinement module[31] that can be combined with the framework to guide the transfer learning of corrosion segmentation. The module, denoted as

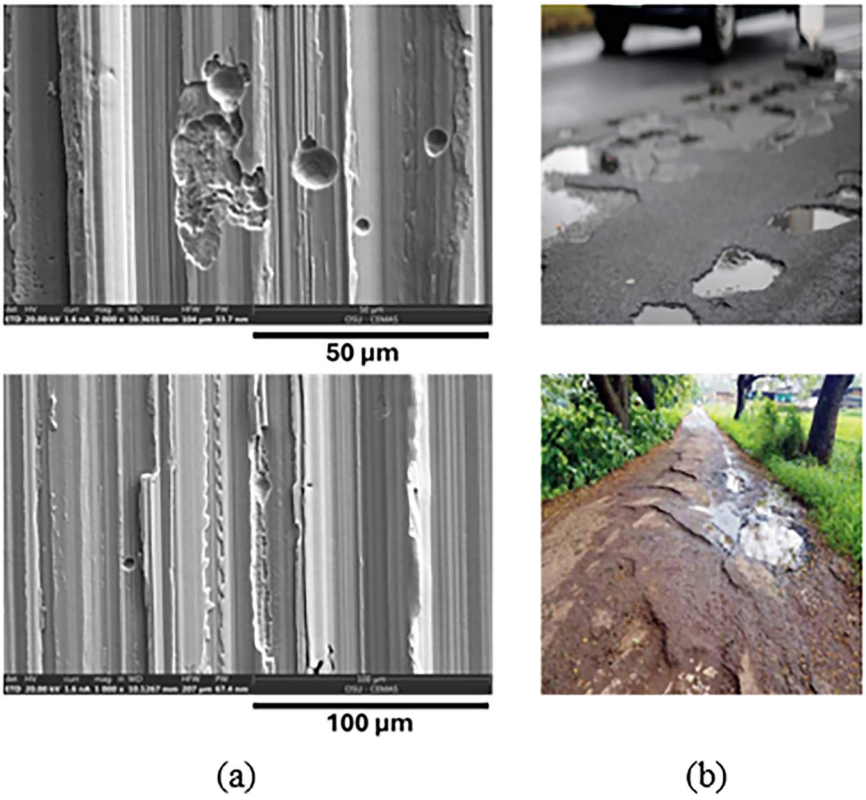

Fig. 6 | Comparison of data variance. Column (**a**) shows two examples from our dataset. Column (**b**) shows two examples from the PIS [24]dataset.

TR block, is integrated before the final prediction head. Based on fused features, the module extracted texture details, enabling the identification of corrosion attributes distinct from backgrounds. The module also consists of multiple CB blocks, each consisting of a sequence of convolutional layers and a batch normalization layer. In the TR block, the collected features were fed into different convolutional layers. Formally, $TR(f|\gamma, \beta) = \gamma \odot f + \beta$, where f denotes the collected feature maps, γ and β are modulation parameters learned during training, and $\odot$ represents element-wise multiplication or Hadamard product. The texture refinement module synthesized multiple feature maps with texture features to increase the accuracy in detecting subtle and indiscernible corrosion morphologies. In the task of corrosion segmentation, accurately identifying texture is important before the results are predicted in feature maps.

Finally, the multiscale features were concatenated and fed into the prediction head. It sequentially applied the classification and segmentation layers to predict the mask $Y\prime$. Specifically, the detection results were generated by the output convolutional layer on the fused features. It calculated the detection loss $\mathscr{L}_1$, which consisted of the Binary Cross Entropy (BCE) classification loss ℓ_{cls} and the box IoU loss ℓ_{iou} during the training. Formally, $\ell_{cls} = -\sum_{i=1}^{N} w_i \left[Y_i \log \sigma\left(Y_i'\right) + \left(1 - Y_i\right) \log\left(1 - \sigma\left(Y_i'\right)\right) \right] / N$, where N is the batch size, w is the weight, Y is the target label, Y' is the predicted category, and σ is the sigmoid function. $\ell_{iou} = 1 - \sum_{i=1}^{N} (I_i / U_i) / N$, where I_i is the intersection of target and prediction boxes, U_i is the union of them. Then, the segmentation candidates $Y\prime$ was predicted through the sigmoid layer. The prediction results were compared with the ground truth Y with pixel-level BCE segmentation loss $\mathcal{L}_2$. Formally, the objection is $\mathcal{L}_2 = -\sum_{i=1}^{HW} \left(Y_i \log Y_i' + \left(1 - Y_i\right) \log\left(1 - Y_i'\right) \right)$, where H is the height and W is the width of the image. The final optimization objection is the sum of two losses: $\mathscr{L} = \mathscr{L}_1 + \mathscr{L}_2$.

In the inference stage, the features only merged information from the main branch, without features from the auxiliary branch. Instead of losses, the Non-Maximum Suppression (NMS) is used to filter out redundant results on predictions.

Data

The model was trained on two public datasets and one newly constructed dataset. The original YOLOv9 was initialized on the public dataset, i.e., Microsoft Common Objects in Context (MS COCO) dataset[26], which are normally used for natural object detection and segmentation. The model developed in this study was trained on the existing Pothole Image Segmentation (PIS) dataset[24] and our newly constructed Corrosion Segmentation in Materials (CSM) dataset.

MS COCO[26] is a large-scale dataset for object detection and segmentation. In the 2017 released version, the dataset consists of 164 K natural scenes with training/validation/test split of 118 K/5 K/41 K. It contains 80 categories of objects, such as cars, bikes, airplanes, apples, boats, horses, kites, etc. However, it does not include categories for holes, corrosion, or similar objects. Thedataset was used to pre-train the model from scratch.

The PIS dataset[24] is composed of 720 training images, 60 validation images, and 0 test images. The dataset was specifically designed for driving safety and road maintenance. The potholes in roads are detected to prevent accidents, reduce repair costs, and ensure smooth traffic flow. However, a lot of these images in this dataset are road cavities filled with water, which exhibits minimum texture and high contrast compared to their surroundings, as shown in column b of Fig. 6. The model trained on this dataset performed poorly for our corrosion segmentation tasks, as shown on Fig. 1b. It failed to identify the targets entirely, suggesting a significant limitation. Note that the PIS dataset is quite different from our SEM data (Fig. 6a). Specifically, the PIS dataset contains clear sparse cavities while our SEM data contains dense and complex corrosion patterns.

Our data (as shown in column a of Fig. 6) presents greater complexity in corrosion segmentation. Obviously, the corrosion damage, specifically the pit morphologies, shows similar contrast to their surroundings, with subtle texture changes in localized regions. Additionally, some corroded areas are very small, and the texture is difficult to discern. In cases where no texture exists, the corrosion pattern could be easily confused with the uncorroded areas. Furthermore, the boundaries of corrosions are often not clear. Clearly, pure transfer learning does not work for our task. The CSM

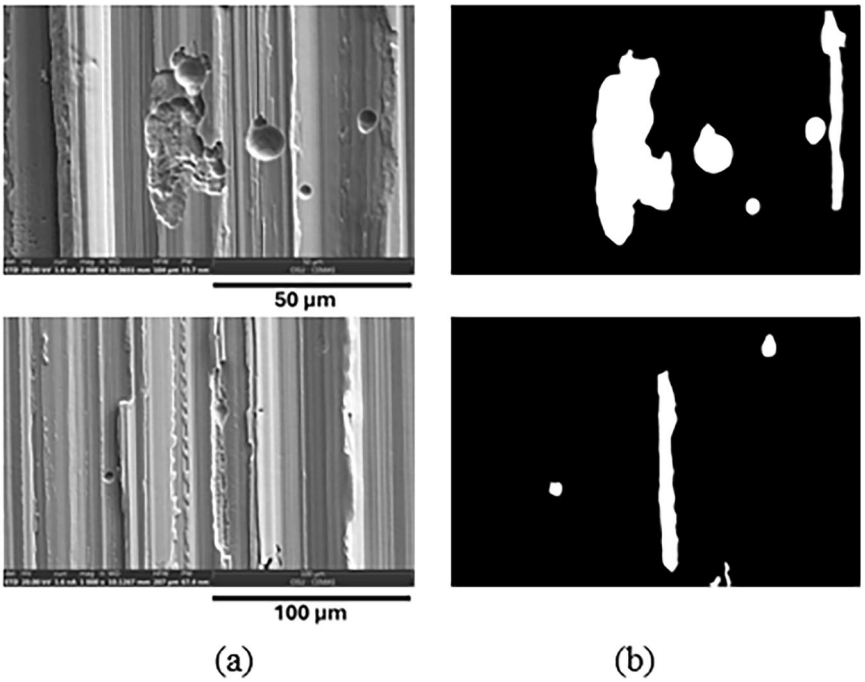

Fig. 7 | Illustration of data labeling. Column (**a**) shows examples of as collected SEM images. Column (**b**) displays the corresponding binary masks indicating the corrosion areas.

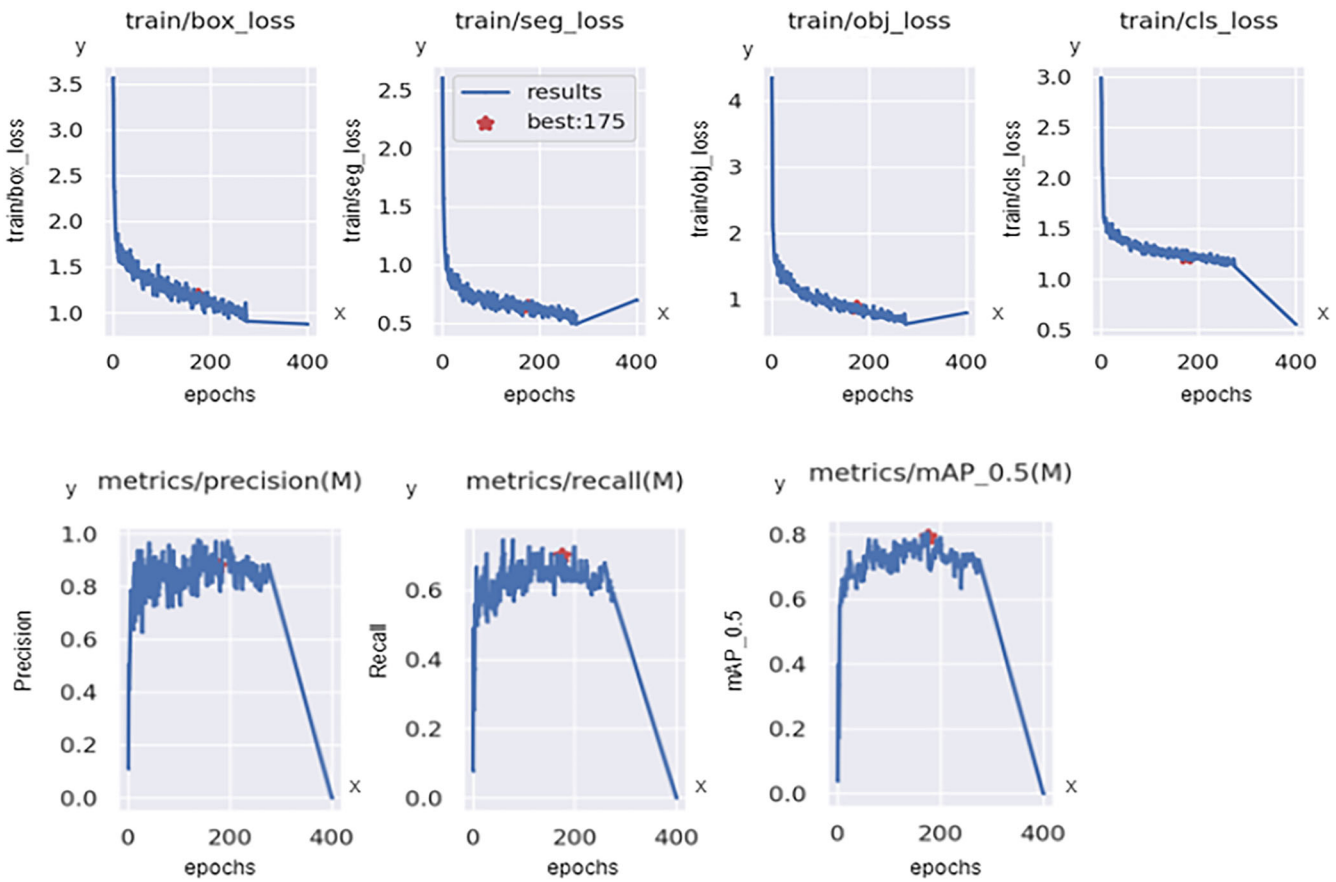

Fig. 8 | The training loss curves, along with performance metrics across epochs. The x axis indicates the epochs. The first row shows the training loss curves with training halted after 300 epochs. The "box_loss" and "cls_loss" refer box IoU loss and box classification loss. The "seg_loss" indicates instance segmentation loss. The "obj_loss" is the averaged total loss. The y axis is the value of loss. The second row illustrates the precision, recall and mAP performance curves on the training set. The y axis is the metric value of performance.

dataset, which consists of 75 training images and 9 testing images, was constructed primarily to address these challenges and adapt to the requirements of corrosion segmentation.

To construct the dataset from scratch, we first collected SEM micrographs from different areas on the SS304L. We then selected valid data for labeling through a comprehensive process, which included identifying duplicates in different conditions, removing images with slight changes, and filtering out images with little discernible attributes. After processing, we obtained 84 valid images, which were divided into 75 training images and 9 testing images. Finally, we manually labeled the ground truth for each micrograph. This step was, however, labor intense and expensive as the corrosion features were often difficult to identify due to the low contrast and diverse shapes. To improve the efficiency of this step, we adopted an interactive tool[32] to assist in drawing masks using human vision. We also consulted the team members with corrosion expertise to guide the labeling process when the corrosion morphologies were complicated to delineate. We interactively annotated each corrosion damage to achieve the most accurate shapes possible. We modified the annotations three times for all the micrographs. Finally, corrosion experts were again engaged to review and verify each annotation for correctness. Examples of the annotated images are shown in Fig. 7.

Implementation

The model was implemented using Pytorch[33]. The batch size was set to 8 for training. The parameters were updated by Adam optimizer[34]. The initial learning rate was 1×10^{-3}, and the weight decay was 5×10^{-4}. The analysis was conducted on a single Tesla V100 graphics processing unit. The training process was conducted in two steps. First, the model was trained on the PIS[24] dataset for 300 epochs, initialized with pre-trained weights on MS COCO[26]. In the second step, the model was further trained on the newly constructed CSM dataset for an additional 300 epochs. In the inference stage, we used the testing data from the CSM database and obtained the corrosion segmentation results from the model. Additionally, the quantitative evaluation metrics were also calculated by comparing the predicted results with the ground truth.

To enhance the diversity of the training data and improve the predicting capability of the model, the data was augmented by rotating at 0°, 90°, 180°, 270° and flipping. The augmentation was implemented by a combination of rotations with horizontal, vertical, and dual axis flips at each angle. Additionally, random cropping, brightness adjustment, and resizing were adopted. Specifically, the augmentation process included: a 50% probability of horizontal flipping, random cropping within a range from 0 to 20%, random rotations from -15 to 15 degrees, random sharing from -5 to 5 degrees both horizontally and vertically, random brightness adjustment within a range from -25% to 25%, and random exposure adjustment within a range from -25% to 25%.

As shown in Fig. 8, the training and validation loss curves help assess the generalization performance of the model given the limited dataset of 84 labeled instances. While the dataset is relatively small, the model benefits from pretraining on a large external dataset, and the loss curves show no signs of overfitting. The training loss converges around epoch 300, and the validation loss approaches zero without divergence, indicating consistent learning rather than memorization. Furthermore, on the training set of 75 images, the precision, recall, and mAP metrics show stable and consistent performance, again supporting the robustness of the implementation under data-limited conditions.

Data availability

The dataset and model used in this work have been deposited into a public data repository [https://doi.org/10.5281/zenodo.15627907].

Received: 12 May 2025; Accepted: 18 June 2025;
Published online: 03 July 2025

References

1. Frankel, G. S. Pitting corrosion of metals: a review of the critical factors. *J. Electrochem. Soc.* **145**, 2186 (1998).
2. Li, T., Scully, J. R. & Frankel, G. S. Localized corrosion: passive film breakdown vs pit growth stability: Part II. A model for critical pitting temperature. *J. Electrochem. Soc.* **165**, C484 (2018).
3. Li, T. et al. Activation energy of metal dissolution in local pit environments. *Corros. Sci.* **193**, 109901 (2021).
4. Nyby, C. et al. Electrochemical metrics for corrosion resistant alloys. *Sci. data* **8**, 58 (2021).
5. Kelly, R. G. Crevice corrosion[M]//Corrosion: Fundamentals, Testing, and Protection. ASM International, 242–247 (2003).
6. Lillard, R. S. et al. Quantifying Alloy 625 crevice corrosion using an image differencing technique: part II. a diffusive transport model of crevice cation concentration using surface current density. *J. Electrochem. Soc.* **167**, 141503 (2020).
7. Guo, X. et al. Self-accelerated corrosion of nuclear waste forms at material interfaces. *Nat. Mater.* **19**, 310–316 (2020).
8. Guo, X. et al. Enhanced crevice corrosion of stainless steel 316 by degradation of Cr-containing hollandite crevice former. *Corros. Sci.* **205**, 110462 (2022).
9. Sieradzki, K. & Newman, R. C. Stress-corrosion cracking. *J. Phys. Chem. solids* **48**, 1101–1113 (1987).
10. Was, G. S. et al. Corrosion and stress corrosion cracking in supercritical water. *J. Nucl. Mater.* **371**, 176–201 (2007).
11. Jones, R. H. Stress-corrosion cracking[M]//Corrosion: fundamentals, testing, and protection. ASM international 346–366 (2003).
12. Chen, H. et al. Microbiologically influenced stress corrosion cracking responsible for catastrophic failure of cable bolts. *Eng. Fail. Anal.* **131**, 105884 (2022).
13. Qu, H. J., Tatman, J. & Wharry, J. P. Chloride-induced stress corrosion cracking in Austenitic steels for SNF storage canisters-Recent understanding and advances in mitigation and repair. *J. Nucl. Mater.* **596**, 155080 (2024).
14. Katona, R. M., Karasz, E. K. & Schaller, R. F. A review of the governing factors in pit-to-crack transitions of metallic structures. *Corrosion* **79**, 72–96 (2023).
15. Weirich, T. D. et al. Humidity effects on pitting of ground stainless steel exposed to sea salt particles. *J. Electrochem. Soc.* **166**, C3477–C3487 (2019).
16. Holm, E. A. et al. Overview: Computer vision and machine learning for microstructural characterization and analysis. *Metall. Mater. Trans. A* **51**, 5985–5999 (2020).
17. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, (2015): 234–241.
18. Redmon, J. You only look once: Unified, real-time object detection[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. (2016).
19. He, K. et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. (2017): 2961–2969.
20. Carion, N. et al. End-to-end object detection with transformers[C]// European conference on computer vision. Cham: Springer International Publishing, (2020): 213–229.
21. Ahuja, S. K., Shukla, M. K. & Ravulakollu, K. K. Neural Network-Based Surface Corrosion Classification on Metal Articles[C]//Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2020. Springer Singapore, (2021): 115–124.
22. Krysko, N. V. et al. Identification and localization of pitting corrosion on metallic surface using deep learning. (2024).
23. Katsamenis, I. et al. Pixel-level corrosion detection on metal constructions by fusion of deep learning semantic and contour

segmentation[C]//International Symposium on Visual Computing. Cham: Springer International Publishing, (2020): 160–169.
24. Pothole Image Segmentation Dataset sourced via Roboflow, URL: https://universe.roboflow.com/farzad/pothole_segmentation_yolov8.
25. Wang, C. Y., Yeh, I. H. & Mark Liao, H. Y. Yolov9: Learning what you want to learn using programmable gradient information[C] // European Conference on Computer Vision. Springer, Cham, (2025): 1–21.
26. Microsoft Common Objects in Context (MS COCO) dataset, URL: https://cocodataset.org/.
27. Elfwing, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **107**, 3–11 (2018).
28. Wang, C. Y. et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. (2020): 390–391.
29. He, K. et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778 (2016).
30. Tishby, N. & Zaslavsky, N. Deep learning and the information bottleneck principle[C]//2015 ieee information theory workshop (itw). IEEE, (2015): 1–5.
31. Zhao, L. et al. Background-insensitive scene text recognition with text semantic segmentation[C] // European Conference on Computer Vision. Cham: Springer Nature Switzerland, (2022): 163–182.
32. Sun, S. et al. Cfr-icl: Cascade-forward refinement with iterative click loss for interactive image segmentation[C] // Proceedings of the AAAI Conference on Artificial Intelligence. (2024), 38: 5017–5024.
33. Imambi, S. & Prakash, K. B. & Kanagachidambaresan, G. R. PyTorch. Programming with TensorFlow: solution for edge computing applications, (2021): 87–104.
34. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, (2014).

Acknowledgements

We acknowledge the financial support from the U.S. Department of Energy, MFC (MFC) of the Nuclear Technology Research and Development program in the office of Nuclear Energy, as well as the Nuclear Energy University Program through Award No. NE0009388. This research made use of the resources of the High-Performance Computing Center (HPC) at Idaho National Laboratory (INL), which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities (NSUF). This manuscript has been authorized by Battelle Energy Alliance, LLC under Contract No. DE-AC07–05ID14517 with the U.S. Department of Energy. the U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript or al- low others to do so, for U.S. Government purposes. And the authors greatly acknowledge all IMCL facility support regarding sample preparation and handling, and data collection.

Author contributions

L.Z.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing—original draft, review and editing. J.L.: Investigation, Data collection, Writing—review & editing. F.X.: Investigation, Data collection, Writing—review & editing, Resources, Supervision. T.Y.: Investigation, Supervision. X.G.: Investigation, Data collection, Writing—review & editing, Resources, Funding acquisition, Supervision, Project administration—review & editing.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Tiankai Yao or Xiaolei Guo.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.